A DISTRIBUTED ALGORITHM FOR COOPERATIVE MULTI-AGENT LOCALIZATION WITH RELATIVE MEASUREMENTS

Prabir Barooah Wm. Joshua Russell João P. Hespanha

Technical Report
Dept. of Mechanical and Aerospace Engineering, Univ. of Florida

***Abstract*—We propose a distributed algorithm for collaborative localization of a number of mobile agents in the absence of GPS. We assume that individual agents can obtain their own odometry measurements, and certain agents can measure their relative positions with other agents. All these are instances of noisy "relative measurements" between nodes in a measurement graph whose nodes correspond to past and current positions of the agents. The centralized optimal estimator (best linear unbiased estimator) of the agent positions given these measurements can produce estimates with much higher accuracy than what is possible by integrating odometry measurements alone. The proposed algorithm is designed to approximate the centralized optimal estimates, and it is distributed in the sense that every agent only needs to communicate with its neighbors. The proposed algorithm also produces a smoothed estimate of past positions of the agents. Effectiveness of the proposed algorithm is illustrated through simulations.**

I. INTRODUCTION

Mobile autonomous agents such as unmanned ground robots and unmanned aerial vehicles (UAV)s that are equipped with on-board sensing, actuation, computation and communication capabilities hold great promise for applications such as surveillance, disaster relief, and scientific exploration. Irrespective of the application, their successful use generally requires the ability for the mobile agents to obtain accurate estimates of their positions as well as the positions of landmarks, events, or targets that they detect. Although typically position information is provided by GPS, in many scenarios GPS may be available only intermittently, or sometimes not available at all. GPS denial may occur due to jamming (in hostile scenarios) or operation in urban canyons. In such situations, agents can estimate their current positions by integrating measurements of relative positions between current and past instants, which are obtained by IMUs (inertial measurement units) or/and vision sensors [1–3]. Such measurements are generally referred to as *odometry* measurements.

Recently, there has been intensive effort in fusing the odometry information from both IMUs and vision sensors to obtain accurate position estimates in GPS-denied scenarios. Although considerable success has been reported in developing such systems, due to the inherently high noise levels in the raw odometry measurements, estimation errors grow with time in the absence of GPS fixes [3–5].

It is possible to reduce localization errors if information on relative positions between pairs of robots are available. Such measurements can be obtained by vision-based sensing (cameras, LIDARs) and RF sensing (angle of arrival and received signal strength). These measurements, although noisy, furnish information about the agent's locations in addition to that provided by odometry measurements. The problem of fusing information on relative positions between mobile agents for estimating their locations is commonly known as *cooperative localization* in the robotics literature [6–8]. The typical approach is to use an extended Kalman filter to fuse both odometry data and robot-to-robot relative *distance* measurements [8, 9]. In the sensor networks and control literature, the problem of distributed Kalman filtering in multi-agent systems has drawn quite a bit of attention [10–13]. The papers [10, 11] in particular present a distributed Kalman filtering techniques for multi-agent localization. The paper [14] addresses cooperative localization in mobile sensor networks with intermittent communication, in which an agent updates its prediction based on the agents it encounters.

In this paper we examine the problem of cooperative localization from *relative position* measurements as a parameter estimation problem, where the parameters to be estimated at time t are the unknown robot positions up to that time, and the measurements available are the measurements of relative positions between certain pairs of robots as well as odometry measurements. The optimal linear estimator, the so-called BLUE (best linear unbiased estimator), has a convenient structure that can be described in terms of a graph consisting of nodes (agent locations) and edges (relative measurements). This structure can be exploited to distribute the computations by using parallel iterative methods of solving linear equations. We assume that every agent has an on-board compass so that all relative position measurements are expressed in a global Cartesian reference frame.

The optimal estimator decreases the error variance significantly compared to what an individual agent can obtain by integrating own odometry measurements. However, computing the optimal estimates requires that all the measurements

---

This material is based upon work supported by the Institute for Collaborative Biotechnologies through grant DAAD19-03-D-0004 from the U.S. Army Research Office and by the National Science Foundation under Grant No. CCR-0311084.

The first author is with the Dept. of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL. The other authors are with the Dept. of Electrical and Computer Engineering, Univ. of California, Santa Barbara, CA 93106.

(agent-to-agent as well as odometry measurements of all the agents) are made available to a central processor. Such a centralized approach requires routing data though an ad-hoc network of mobile agents, which is a difficult problem [15]. Therefore, we are interested in a distributed scheme that allows agents to estimate their positions accurately through local computation and communication instead of relying on a central processor.

This paper proposes an algorithm that can be employed by the agents in a robotic team to estimate their positions by fusing robot-to-robot relative position measurements with odometry measurements. The algorithm is distributed, in the sense that each agent estimates its own position, and all the information needed to carry out the computation is obtained from local sensor measurements and communication with nearby agents. We assume that every agent can exchange messages with another robot if it can obtain a relative position measurement with respect to that robot. The amount of information that agents have to exchange and store is also limited. The algorithm builds upon our earlier work on distributed localization of static agents [16, 17][1].

We show that a significant improvement in localization accuracy can be achieved by employing the proposed algorithm, compared to pure dead reckoning. The estimates produced by the algorithm (estimated from Monte-Carlo simulations) are seen to be quite close to the variance of the optimal estimator. In addition, the proposed method provides *smoothed* estimates of past positions of agents in real time. Accurate past position estimates of agents are needed, for example, to geotag data collected by their sensors, or to geolocate events detected. To the best of our knowledge, our algorithm is the first to achieve distributed smoothing for multi-agent localization.

The centralized BLUE estimator is closely related to the Kalman filter estimates; in fact the two are the same under certain circumstances. This is due to the fact that the linear minimum mean-square error (LMMSE) estimator is the same as the BLUE when the prior covariance is infinite [18], while the Kalman filter computes the LMMSE in a recursive way. The proposed algorithm can therefore be thought of as a distributed Kalman filter for agent localization (see [10–13] for related work). An additional benefit of the proposed method is that it provides a smoothed estimate of past agent locations.

The rest of the paper is organized as follows. Section II describes the estimation problem precisely, and Section III describes centralized optimal estimation in measurement graphs and provides an example that illustrates the benefits of optimal estimation in multi-agent localization. Section IV describes the proposed algorithm, and simulations are presented in Section V.

II. PROBLEM DESCRIPTION

Consider a group of n mobile agents that need to estimate their own positions with respect to a geostationary coordinate frame, whose origin is denoted by x_0. In the absence of GPS, we arbitrarily fix the initial position of one of the agents, say, the first agent, as the origin. We assume that at certain discrete time instants $t \in \{1, 2, \dots\}$ an agent measures its relative position with respect to a set of nearby agents. Denoting by $x_j(t)$ the position of the j^{th} entity at time t, and we can associate the positions $\{x_0\} \cup \{x_j(\tau)\}$, $j \in \{1, \dots, n\}$, $\tau \in \{1, \dots, t\}$ of the entities until time t with the *nodes* of a *measurement graph* $\mathcal{G}(t) = (\mathcal{V}(t), \mathcal{E}(t))$. The *edges* $\mathcal{E}(t)$ of the graph correspond to the relative position measurements between the nodes $\mathcal{V}(t)$. In this problem we have four types of relative measurements:

1) *inter-agent measurements:* measurements of the relative position between two mobile agents at a given time τ, of the form $\zeta_{i(\tau),j(\tau)} := x_i(\tau) - x_j(\tau) + \epsilon$, where ϵ denotes some measurement error.
2) *GPS measurements:* measurements of the relative position of one mobile agent with respect to a global reference, of the form $x_i(\tau) - x_0 + \epsilon$, where ϵ denotes the error in GPS measurements.
3) *odometry measurements:* measurement of the displacement of a mobile agent between two consecutive time instants, of the form $\zeta_{i(\tau),i(\tau-1)} := x_i(\tau) - x_i(\tau-1) + \epsilon$.

We assume that the noise sequences affecting the measurements listed above (though we have suppressed time dependency etc. for simplicity) are white, zero mean, uncorrelated with one another, and are possibly non-stationary. We assume that agent has an on-board compass so that these relative position measurements are expressed in a common Cartesian reference frame.

All measurements mentioned above are of the type

$$\zeta_e = x_u - x_v + \epsilon_e \tag{1}$$

where u and v are nodes of the measurement graph $\mathcal{G}(t)$ and $e = (u, v)$ is an edge (an ordered pair of nodes). In particular, an edge exists between a node pair u and v if and only if a relative measurement of the form (1) is available between the two nodes. Since a measurement of $x_u - x_v$ is different from that of $x_v - x_u$, the edges in the measurement graph are directed. The edge directions are arbitrary.

We assume that if an agent can measure its position relative to another agent, it can also exchange messages with that agent, and that the resulting *communication graph*, as well as the measurement graph $\mathcal{G}(t)$ is connected at every time t. Figure 1 shows an example of a measurement graph and a communication graph created by the motion of four mobile agents.

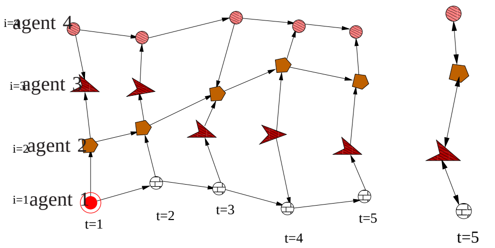
The problems of interest are: (1) to accurately estimate the positions of the agents and targets $x_j(t)$, $j \in \{1, \dots, n\}$ at the current time t, and (2) to accurately estimate the positions of agents and targets at some specific time instant τ in the past (i.e., $\tau < t$), from the noisy relative position measurements available until the current time t. The estimation scheme is to be distributed in the sense described earlier.

III. BLU ESTIMATION FROM RELATIVE MEASUREMENTS

We briefly review BLU estimation from relative measurements for graphs that do not change with time. The BLUE is optimal (minimal variance) among all linear unbiased estimators. Consider a fixed measurement graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$,

---

[1] *Draft note: Put more citations.*



(a) The measurement graph $\mathcal{G}(5)$ at time 5. (b) Comm. at time 5.

Fig. 1. (a) An example of a measurement graph generated as a result of the motion of a group of four mobile agents. The graph shown here is $\mathcal{G}(5)$, i.e., the snapshot at the 5^{th} time instant. The unknown variables at current time $t = 5$ are the positions $x_i(\tau)$, $i \in \{1, 2, \dots, 4\}$, at the time instants $\tau \in \{1, 2, \dots, 5\}$, except for the initial position of robot 1: $x_1(1)$, which is taken as the reference. (b) The communication graph at time 5.

where the nodes in $\mathcal{V}$ correspond to variables and edges in $\mathcal{E}$ correspond to relative measurement between node variables of the form (1). Let $V_r \subset \mathcal{V}$ denote the non-empty subset of nodes whose variables are known, which are called *reference variables* , and $n = |\mathcal{V} \setminus \mathcal{V}_r|$ be the number of unknown variables that are to be estimated. Let k be the dimension of each variable (e.g., $k = 2$ if only x- and y-coordinates are of interest). Let $\mathbf{x} \in \mathbb{R}^{nk}$ be the vector obtained by stacking together the unknown variables. As described in [17], given a measurement graph with n unknown variables, the BLU estimate $\hat{\mathbf{x}}^*$ is given by the solution of a system of linear equations

$$\mathrm{L}\,\mathbf{x} = \mathbf{b}, \tag{2}$$

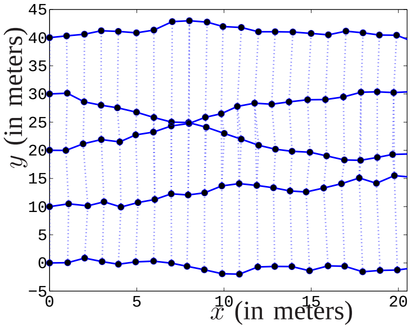
where $\mathrm{L} \in \mathbb{R}^{nk \times nk}$ and $\mathbf{b} \in \mathbb{R}^{nk}$ depends on the measurement graph $\mathcal{G}$, the measurement error covariance matrices $P_e, e \in \mathcal{E}$, the measurements $\zeta_e, e \in \mathcal{E}$ and the reference variables $x_r, r \in \mathcal{V}_r$. The matrix L is invertible (so that BLU estimate $\hat{\mathbf{x}}^*$ exists and is unique) if and only if for every node, there is an undirected path between the node and at least one reference node [19]. Under this condition, the covariance matrix of the estimation error $\Sigma := cov(\hat{\mathbf{x}}^*, \hat{\mathbf{x}}^*)$ is given by

$$\Sigma = \mathrm{L}^{-1}\,. \tag{3}$$

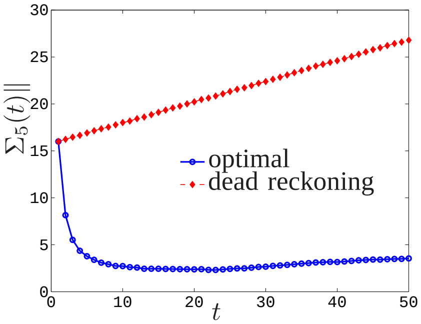
The reader is referred to [17, 19] for the details.

A. BLU estimation reduces error propagation

We first provide an example of how the growth of error in localization from odometry can be reduced by using the optimal estimation method described in the previous section when multiple agents are present. Figure 2(a) shows an example of the measurement graph that is created by the motion of 5 autonomous agents. The agents are initially arranged in a line along the y-axis with $10m$ separation between pairs. Relative position measurements (both odometry and agent-to-agent) are obtained by the agents from measurements of range and angle. We assume the range and bearing measurements are affected by a additive white Gaussian noise with standard deviations $\sigma_r = 0.5m$ and $\sigma_\theta = 11^o$, respectively. The measurement error covariances are estimated from the range and bearing measurements and these two parameter values, as explained in [17]. GPS is assumed not available, so the initial position of agent 1 is fixed as the global reference. The covariance of the optimal estimation error (based on the relative measurements) of agent 5 is compared with that obtained from dead reckoning in Figure 2(b). The plot shows that the optimal estimator that combines both forms of relative measurements achieves a much lower error than what is achieved by dead reckoning.



(a) The measurement graph at time $t = 20$ with 5 agents, initially arranged along the y axis, with agent 1 being the "bottom-most" and agent 5, the "top-most". Dashed lines represent relative position measurements between agents and solid lines represent odometry measurements.



(b) Error in the position estimate of agent 5 with optimal estimation and dead reckoning. $\Sigma_5(t)$ is the optimal error covariance of the position estimate of agent 5 at time t.

Fig. 2. The measurement graph $\mathcal{G}(t)$ as a function of time until $t = 20$ created by the motion of 5 mobile agents (Figure (a)), and the error covariances of their position estimates with two different methods - dead reckoning and optimal estimation from the relative measurements (Figure (b)). Dead reckoning does not use the relative position measurements between agents. As the plot in (b) shows, the optimal estimator reduces the error growth significantly by using the agent-to-agent relative position measurements.

This example demonstrates the benefits of collaborative localization (using optimal estimation from relative measurements), even with a small number of agents.

The proof is provided in the Appendix.

Proposition 1. *Let n agents move in a way that the resulting measurement graph at time t is an $t \times n$ rectangular grid. If the error covariance of each relative measurement is uniformly bounded, then the error covariance of the (centralized) BLUE estimate of a robot's position at time is $\Theta(t/n) + O(\log(nt))$. If each robot estimates its position by integrating its odometry measurements alone without using inter-robot relative mea-*

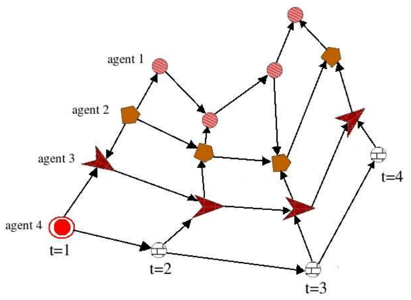


Fig. 3. Motion of four agents that produces a measurement graph at time t which is a $t \times 4$ 2D grid.

surements, the error in its position estimate at time t is $\Theta(t)$.

Note that the notation $g(x) = \mathcal{O}(p(x))$ means that there exist positive constants x_o, a such that $g(x) \leq ap(x)$ for all $x > x_o$. Similarly, $g(x) = \Omega(p(x))$ means there exist a positive constants x_o, b such that $g(x) \geq bp(x)$ for all $x > x_o$. The notation $g(x) = \Theta(p(x))$ means that both $g(x) = \mathcal{O}(p(x))$ and $g(x) = \Omega(p(x))$ are true.

Remark 1. *A measurement graph of the kind specified in the proposition results when the agents move in a way that the neighbor relationships do not change with time (e.g., the one shown in Figure 3). This result shows that asymptotically (t large), BLUE estimation with inter-robot measurements reduces the error variance by a factor of n, the number of robots.*

IV. A DISTRIBUTED ALGORITHM FOR DYNAMIC LOCALIZATION

Our goal is to devise a distributed algorithm to obtain position estimates of the mobile agents/targets that are close to the centralized BLU estimator described in the previous section. By distributed we mean every agent should be able to estimate its own position, and all the information needed to carry out the computation should come from local sensing and communication with its neighboring agents. Two agents are said to be neighbors if they are able to exchange messages wirelessly. Here we describe such an algorithm.

A. Infinite memory

We first describe the algorithm by assuming that every agent can store and broadcast an unbounded amount of data. We will relax this assumption later.

For every agent j, let $\mathcal{V}_j(t)$ contain all the nodes that correspond to the positions of itself and the positions of the agents with whom j has had relative measurements, up to and including time t. Let $\mathcal{E}_j(t)$ be the subset of edges in $\mathcal{G}(t)$ that are incident on nodes that correspond to j's current or past positions. By assumption, the relative measurements $\zeta_e, e \in \mathcal{E}_j(t)$ are available to agent j at time t. We now define the *local subgraph* of agent j at time t as $\mathcal{G}_j(t) = (\mathcal{V}_j(t), \mathcal{E}_j(t))$. Figure 5 shows the subgraphs of agents 1 and 2 at time $t = 4$ corresponding to the measurement graph shown in Figure 1.

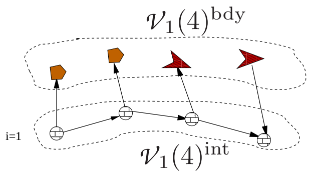


Fig. 4. The subgraph $\mathcal{G}_1(4)$ of agent 1 at time 4, for the measurement graph shown in Figure 1.

The nodes in a local subgraph $\mathcal{G}_j(t)$ are divided into two categories - the *internal nodes* $\mathcal{V}_j(t)^{\text{int}}$ and the *boundary nodes* $\mathcal{V}_j(t)^{\text{bdy}}$. The internal nodes are the nodes that correspond to the positions of the agent up and including time t. The boundary nodes consist of the nodes in the local subgraph that correspond to the positions of the neighboring agents. Thus, $\mathcal{V}_j(t) = \mathcal{V}_j(t)^{\text{int}} \cup \mathcal{V}_j(t)^{\text{bdy}}$ and $\mathcal{V}_j(t)^{\text{int}} \cap \mathcal{V}_j(t)^{\text{bdy}} = \emptyset$.

To explain the algorithm, imagine first that the agents have stopped moving at time t_{stop}. We have proposed a distributed algorithm in [17, 20] for localization of static agents that is based on the Jacobi iterative method of solving linear equations [21]. If agents stop moving, they can use the Jacobi algorithm to compute the optimal estimate of its entire position history in distributed manner. We describe the procedure briefly, which will serve as a stepping stone into developing the proposed algorithm.

1) Static estimation at t_{stop}: Let o be the global reference node. Consider the set of past positions of agent j until time t_{stop}, i.e., $\{x_v, v \in \mathcal{V}_j(t_{\text{stop}})^{\text{int}} \setminus \{o\}\}$. The vector of the node variables in this set is denoted by $\mathbf{x}_{j,t_{\text{stop}}}$. The set of unknown node positions at time t_{stop} is $\cup_j \mathbf{x}_{j,t_{\text{stop}}}$. Let $\hat{\mathbf{x}}_{j,t_{\text{stop}}}(\tau)$ be the estimate of $\mathbf{x}_{j,t_{\text{stop}}}$ obtained by agent j at time τ, where $\tau > t_{\text{stop}}$. These estimates are obtained and improved using the following distributed algorithm, starting with an arbitrary initial condition:

At time τ ($\geq t_{\text{stop}}$), every agent j does the following.

1) It broadcasts the current estimate $\hat{\mathbf{x}}_{j,t_{\text{stop}}}(\tau)$ to all of its neighboring agents. Consequently, it also receives the current estimates $\hat{\mathbf{x}}_i(\tau)$ from each of its neighboring agent i.
2) It (agent j) assigns the boundary nodes $\mathcal{V}_j(t_{\text{stop}})^{\text{bdy}}$ as the reference nodes of its local subgraph $\mathcal{G}_j(t_{\text{stop}})$ and sets the reference variables to be the estimates of those node variables that it has recently received from its neighbors. With this assignment of reference node variables and with the relative measurements $\{\zeta_e, e \in \mathcal{E}_j(t_{\text{stop}})\}$, agent j then sets up the system of linear equations (2) for its local subgraph $\mathcal{G}_j(t_{\text{stop}})$, and solves these equations to obtain an updated estimate of $\hat{\mathbf{x}}_{j,t_{\text{stop}}}(\tau + 1)$ of its "internal" node variables.

The procedure above is repeated by all the agents at every time instant $\tau = t_{\text{stop}}, t_{\text{stop}} + 1, \ldots$. $\square$

The following result about the behavior of the estimates follows from the convergence property of the Jacobi algorithm (see [17, 20]).

Proposition 2. *The estimates of all the node variables $\mathbf{x}(t_{\text{stop}})$ (i.e., all agents' past positions up to time t_{stop}) converge to* *its centralized optimal estimate:* $\hat{\mathbf{x}}_{j,t_{\text{stop}}}(\tau) \to \hat{\mathbf{x}}^{\text{BLUE}}_{j,t_{\text{stop}}}$ *as* $(\tau - t_{\text{stop}}) \to \infty$. □

As a result, if agents stop moving, by communicating with its neighbors and updating sufficiently many times, an agent can obtain an estimate of its entire position history that is arbitrarily close to the optimal estimates.

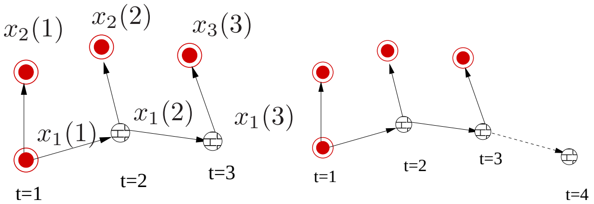
Now we are ready to describe the algorithm when agents are mobile. In the description of the algorithm that follows, T_{dr} is a (integer) design variable that is provided to all the agents initially.

2) Estimation with mobile agents:

1) If GPS is not available to every agent at $t = 1$, one agent's initial position serves as the global reference. Every other agent starts with the initial estimate that is obtained by adding the relative measurements on a path from itself to the agent whose initial position is taken as the global reference. For example, when agent 1 is the global reference and relative position measurements are available between agents with successively increasing indices, we have $x_j(1) := \zeta_{j(1),j-1(1)} + \zeta_{j-1(1),j-2(1)} + \cdots + \zeta_{2(1),1(1)}$. We assume that these measurements are transmitted to the agents initially before they start moving. From $t = 1$ till $t = T_{\text{dr}}$, every agent estimates its position based on dead reckoning from the initial estimate : $\hat{x}_j(t+1) = \hat{x}_j(t) + \zeta_{j(t+1),j(t)}$.
2) At every time instant t that is an integer multiple of T_{dr}, i.e., when $t = mT_{\text{dr}}$ for some positive integer m, each agent j broadcasts all the measurements it collects between $(m-1)T_{\text{dr}}$ and mT_{dr}, i.e., $\{\zeta_e, e \in \mathcal{E}_j(mT_{\text{dr}}) \setminus \mathcal{E}_j((m-1)T_{\text{dr}})\}$ to its neighboring agents.
3) At every time instant τ such that $mT_{\text{dr}} \le \tau \le (m+1)T_{\text{dr}}$ for some positive integer m, every agent updates its position estimate thus:
   Update past position estimates: agent j updates the estimate of its past position history $\hat{\mathbf{x}}_{j,t_{\text{stop}}}(\tau)$ with $t_{\text{stop}} = mT_{\text{dr}}$, by performing the computations described in Section IV-A1. The agent whose initial position is taken as the reference (in the absence of GPS) always uses its initial position as one of the reference node variables.
   Estimate current position: after past position estimates are updated, agent j estimates its current position by dead reckoning with respect to its most recently updated position estimate of $x_j(mT_{\text{dr}})$:
   $$\hat{x}_j(\tau) = \hat{x}_{j,mT_{\text{dr}}}(\tau) + \zeta_{j(mT_{\text{dr}}+1),j(mT_{\text{dr}})} + \dots + \zeta_{j(\tau),j(\tau-1)}$$
   Concurrently, agent j also obtains new relative measurements (between itself and its nearby agents and between its current and previous positions) and stores them in local memory.

The algorithm continues as long as the agents continue to move. If all entities cease to move, the measurement graph stops changing with time. In that case, the algorithm is terminated after sufficiently many iterations in step 3 are performed after motion ceases.

It follows from Proposition 2 that if T_{dr} is large, the estimates at current time τ, of the *past* positions of the agents (until time mT_{dr}), will be close to their optimal estimates in the graph $\mathcal{G}(mT_{\text{dr}})$ as $t - \tau \to \infty$. However, the current estimate's accuracy will suffer with a large T_{dr}, since that error is a linear function of T_{dr}. Hence there is a trade-off in choosing the value of T_{dr}.



(a) $\mathcal{G}_1(3)$: the local subgraph of agent 1 at time 3.

(b) At time 4, agent 1 still uses subgraph $\mathcal{G}_1(3)$ to improve its past positions.

Fig. 5. Subgraphs of agent 1 at two consecutive time instants for the measurement graph shown in Figure 1. **(a)** The local subgraph of agent 1 at time 3 with the reference nodes marked by red concentric circles. The unknown variables in this subgraph are $x_1(2)$ and $x_1(3)$, its positions at time $t = 2$ and $t = 3$ (current position). Its initial position is also used as a reference node since it is the global reference. Agent 1 uses the estimates of the variables $x_2(1)$, $x_2(2)$, and $x_3(3)$ it receives from its neighbor 2 at time 3 as reference variables to set up the estimation problem. **(b)** The situation at time 4 when $T_{\text{dr}} = 3$. In the interval between $t = 3$ and $t = 4$, agent 1 and 2 have exchanged one round of messages, so that agent 1 has updated estimates of all the node variables in $\mathcal{G}_1(3)$. It recomputes the optimal estimates in $\mathcal{G}_1(3)$ by using the new estimates of $x_2(1)$, $x_2(2)$, and $x_3(3)$ it received in this time interval to update the estimates of its past positions $x_1(2)$ and $x_1(3)$. It then estimates its current position $x_1(4)$ by adding the appropriate odometry measurement to the most recent estimate of $x_1(3)$, i.e., $\widehat{x_1(4)}^{(4)} = \widehat{x_1(3)}^{(4)} + \zeta_{1(4),1(3)}$, where the superscript refers to the time at which the estimate is obtained.

B. Finite memory

In the description above it was assumed that every agent can keep in its local memory and broadcast an infinite amount of data. In practice an agent will have a finite amount of memory and the amount of data that can be broadcast and received will be limited. It turns out that the algorithm can be modified to handle these constraints with limited loss of performance.

Depending on the amount of data agents can keep in their local memory (and broadcast), we choose an integer T_m and require that, at time t, every agent keeps in local memory the following data: (i) measurements gathered at times $t-1, t-2, \dots, t-T_m$, (ii) the most recent estimates of its own and its neighbors' positions at those times. All previous data is discarded. We call T_m the *memory length* of the agents. We choose T_m to be an integer multiple of T_{dr}. Consider now the truncated local subgraph $G_j(t, T_m)$ that is constructed by truncating all the nodes and edges from $\mathcal{G}_j(t)$ that corresponds to agents' positions before $t - T_m$. Figure 6 shows an example. The agent whose initial position is taken as the reference (in the absence of GPS) always uses its current estimate of its earliest position in the truncated local subgraph as one of the reference node variables. The rest of the algorithm stays the same.

1) Communication and computation cost:

The amount of data an agent needs to store and broadcast depends on the "size" of the truncated local subgraph that the agent keeps in local memory. In the nominal condition when the neighbors

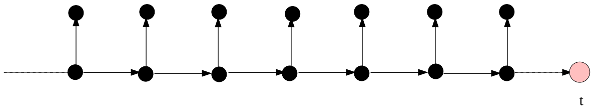


Fig. 6. Truncated subgraph $\mathcal{G}_1(t,6)$ of agent 1 for the measurement graph shown in Figure 7. The parameters in this case are $T_{\rm dr} = 3$, $T_m = 2T_{\rm dr}$, so that the number of time steps for which data is kept is $3T_{\rm dr}$. The nodes and edges shown in solid colors constitutes the local subgraph that is used for the iterative updates. As the current time increases to reach the next multiple of $T_{\rm dr}$, the local subgraph is updated by adding the extra measurements, but the tail of the subgraph is truncated at the same time so that the next set of iterations are done again with a graph that contains only the data from the most recent $T_{\rm m}$ instants.

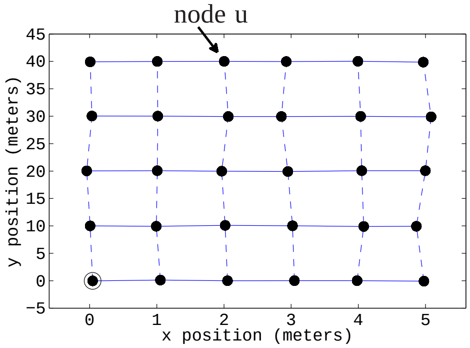


Fig. 7. A snapshot of the measurement graph $\mathcal{G}(t)$ at time $t = 6$ created by the motion of 5 mobile agents, for which the simulations reported here are conducted. The agents are initially arranged along the y-axis with $10m$ separation and they move approximately $1m$ to the right during every discrete time instant. Their motion is constrained to be such that the neighbor relationships don't change with time. The node u marked in the figure is the position of agent 5 at time 3.

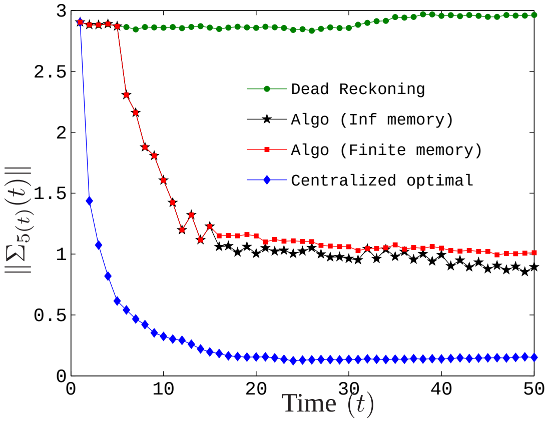


Fig. 8. **Agent 5:** Covariance of the estimate of the current position of agent 5 (of Figure 7) as a function of time. Agent 5 is the one farthest from agent 1, whose initial position being the reference node. The algorithm is run with $T_{\rm dr} = 5$. "Finite memory" is with $T_{\rm m} = 10$, which means only data for the most recent 10 time steps are kept.

of an agent do not change with time, the number of nodes in its local truncated subgraph at any given time will be $T_m + N_{\rm nbr}T_m$, where $N_{\rm nbr}$ is the number of neighbors of the agent. The number of edges in the truncated local subgraph is at most $T_m + T_m N_{\rm nbr}$ (the first term is the number of odometry measurements and the second term is the number of relative measurements between the agent and its neighboring agents that appear as edges in the subgraph). Additionally, a maximum of $T_{\rm dr}N_{\rm nbr}$ measurements may be collected during the iterative computation that are concatenated to the subgraph data at the end of every $T_{\rm dr}$ time steps. Therefore, an agent needs a local memory large enough to store $k[2T_m(1+N_{\rm nbr})+T_{\rm dr}N_{\rm nbr}]$ floating-point numbers.

An agent has to broadcast the current estimates of its interior node variables at every time instant between subgraph updates (i.e., at time instants τ such that $(\ell-1)T_{\rm dr} < \tau < \ell T_{\rm dr}$ for some ℓ). Therefore every agent has to broadcast kT_m numbers.

V. SIMULATIONS

We illustrate the algorithm's performance by numerical simulations. For these simulations, 5 agents are initially arranged in a straight line along the y direction with an initial separation of $10m$ between agents. The agents move toward the right approximately $1m$ at every time instant, but in such a way that after about $t = 8$, agents 3 and 4 cross each other (the situation is shown in Figure 2). Figure 7 shows a snapshot of the agent positions (and the measurement graph) at $t = 6$. GPS information is assumed not available, so the initial position of agent 1 (bottom left node in Figure 7) is taken as the reference. Every measurement of $x_u - x_v$ is obtained from noisy measurements of the distance $\|x_u - x_v\|$ and the angle between x_u and x_v. The distance and angle measurements are corrupted with additive Gaussian noise, with $\sigma_r = 0.05m$ and $\sigma_\theta = 5^o$. The measurement error covariances are estimated from the range and bearing measurements and the parameters σ_r, σ_θ (as explained in [17]), which makes the covariances of the errors on relative position measurements on distinct edges distinct.

Covariances of agent position estimates produced by the proposed algorithm are estimated from 250 Monte-Carlo runs. Figure 8 shows the covariance of the estimate of $x_5(t)$, the position of agent 5, as a function of t. Note that agent 5 is the one farthest away from agent 1. The figure shows that the algorithm (both with infinite and finite memory) performs much better than dead reckoning. It is seen from the plot that the estimation error covariance of the algorithm is close to that of the centralized optimal estimator. Moreover, comparing the plots for the finite and infinite memory cases, one sees that the performance of the proposed algorithm with a small memory of 10 time steps is quite close to that with infinite memory. This shows that the algorithm can perform well even with limited agent memory, which makes it advantageous for agents with limited computational and processing power. Figure 9 plots these variables for the second agent.

The smoothing performance of the proposed algorithm is illustrated in Figure 10. Covariance of the estimates of agent 5's position at time 3 (i.e., of node u shown in Figure 7) produced by the algorithm is plotted as a function of time. As in the previous case, the covariance is estimated from 250 Monte-Carlo simulations. With infinite memory, the covariance of the current estimate of x_u decreases and approaches the value that the centralized BLU estimator achieves. The centralized estimate's covariance quickly settles down to a steady state

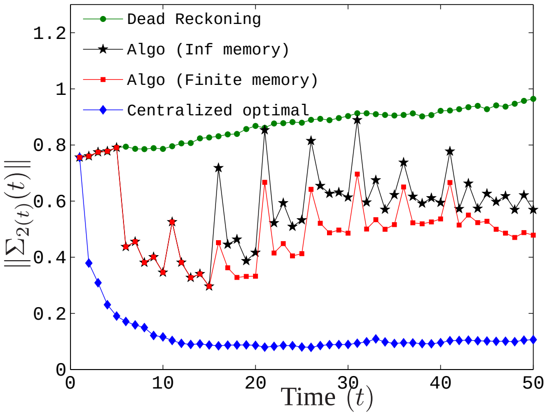


Fig. 9. **Agent 2:** Covariance of the estimate of the current position of agent 2 (of Figure 7) as a function of time. Agent 2 is the one next to agent 1. All parameters are the same as in Figure 8.

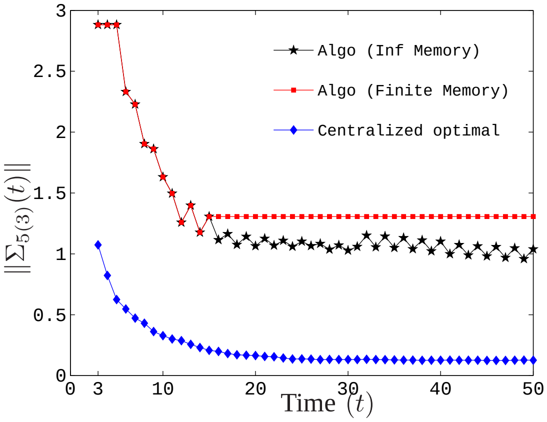


Fig. 10. **Smoothing performance:** Covariance of the estimate of $x_{5(3)}$ (the position of the agent 5 at time 3) produced by the algorithm as a function of time, as well as that produced by the centralized BLU estimator. The parameters for the simulation are $T_{\mathrm{dr}} = 5$ and $T_m = 10$. The covariance of the estimates with finite memory settles down to a steady state value larger than the centralized optimal, since past estimates are no longer updated after a while.

value, since adding more edges to a graph far away from a node does not change its BLUE covariance too much. The last statement follows from the results described in [17]. With finite memory, of course, the covariance of the estimates of x_u stops decreasing after a while, since past estimates are no longer updated after some time when agents have only finite memory.

VI. CONCLUSION

We presented a distributed algorithm for mobile agents to accurately estimate the current and past positions of the agents/targets by fusing odometry measurements with relative positions between agents/targets. The algorithm is distributed in the sense each agent can estimate its own position by communication only with nearby agents. Simulations show that the error covariances of the state estimates that the proposed distributed algorithm yield are close to that of the centralized optimal estimator, and are significantly lower than what is possible by odometry. Apart from estimates of current locations of the agents, the proposed algorithm also produces smoothed estimate of past positions of the agents.

There are several aspects of the proposed algorithm that need further investigation. Although numerical simulations show that the estimation errors are close to the centralized optimal (BLUE) estimation errors, analysis of how big the difference is, is lacking. For example, it will be useful to understand the affect of the parameters T_{dr} and T_m on the difference between the algorithm's estimates and the centralized BLUE. Moreover, the evolution error covariance will depend on the number of agents and the measurement graph, which is determined by agents' motion. The relationship between the covariance and agent motion is a subject of future research.

REFERENCES

[1] J. Borenstein, H. R. Everett, L. Feng, and D. Wehe, "Mobile robot positioning: Sensors and techniques," *Journal of Robotic Systems, Special Issue on Mobile Robots*, vol. 14, no. 4, pp. 231–249, April 1997.

[2] D. Nistér, O. Naroditsky, and J. R. Bergen, "Visual odometry," in *Conference on Computer Vision and Pattern Recognition (CVPR '04)*, 2004, pp. 652–659.

[3] C. F. Olson, L. H. Matthies, M. Schoppers, and M. W. Maimone, "Rover navigation using stereo ego-motion," *Robotics and Autonomous Systems*, vol. 43, no. 4, pp. 215–229, June 2003.

[4] A. Makadia and K. Daniilidis, "Correspondenceless ego-motion estimation using an imu," in *IEEE International Conference on Robotics and Automation*, 2005, pp. 3534–3539.

[5] T. Oskiper, Z. Zhu, S. Samarasekera, and R. Kumar, "Visual odometry system using multiple stereo cameras and inertial measurement unit," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR '07)*, 17-22 June 2007, pp. 1–8.

[6] R. Kurazume, S. Nagata, and S. Hirose, "Cooperative positioning with multiple robots," in *the IEEE International Conference in Robotics and Automation*, 1994, pp. 1250–1257.

[7] I. M. Rekleitis, G. Dudek, and E. E. Milios, "Multi-robot cooperative localization: a study of trade-offs between efficiency and accuracy," in *the IEEE/RSJ International Conference on Intelligent Robots and System*, vol. 3, 2002, pp. 2690–2695.

[8] A. I. Mourikis and S. I. Roumeliotis, "Performance analysis of multirobot cooperative localization," *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 666–681, August 2006.

[9] S. I. Roumeliotis and G. A. Bekey, "Distributed multi-robot localization," *IEEE Transactions on Robotics and Automation*, no. 5, pp. 781–795, October 2002.

[10] P. Alriksson and A. Rantzer, "Distributed kalman filtering using weighted averaging," in *17th International Symposium on Mathematical Theory of Networks and Systems (MTNS)*, 2006.

[11] ——, “Experimental evaluation of a distributed kalman filter algorithm,” in *46th IEEE Conference on Decision and Control*, December 2007.

[12] R. Olfati-Saber, “Distributed kalman filtering for sensor networks,” in *46th IEEE Conference on Decision and Control*, December 2007.

[13] D. P. Spanos, R. Olfati-Saber, and R. M. Murray, “Approximate distributed kalman filtering in sensor networks with quantifiable performance,” in *4th international symposium on Information processing in sensor networks (IPSN ’05)*, 2005.

[14] P. Zhang and M. Martonosi, “LOCALE: collaborative localization estimation for sparse mobile sensor networks,” in *International Conference on Information Processing in Sensor Networks (IPSN)*, 2008, pp. 195–206.

[15] S. Mueller, R. P. Tsang, and D. Ghosal, “Multipath routing in mobile ad hoc networks: Issues and challenges,” in *Performance Tools and Applications to Networked Systems, LNCS*. Springer Berlin/Heidelberg, 2004, vol. 2965, pp. 209–234.

[16] P. Barooah, N. M. da Silva, and J. P. Hespanha, “Distributed optimal estimation from relative measurements for localization and time synchronization,” in *Distributed Computing in Sensor Systems DCOSS*, ser. LNCS, P. B. Gibbons, T. Abdelzaher, J. Aspnes, and R. Rao, Eds. Springer, 2006, vol. 4026, pp. 266 – 281.

[17] P. Barooah and J. P. Hespanha, “Estimation from relative measurements : Algorithms and scaling laws,” *IEEE Control Systems Magazine*, vol. 27, no. 4, pp. 57 – 74, August 2007.

[18] J. M. Mendel, *Lessons in Estimation Theory for Signal Processing, Communications and Control*, A. V. Oppenheim, Ed. Prentice Hall, 1995.

[19] P. Barooah, “Estimation and control with relative measurements: Algorithms and scaling laws,” Ph.D. dissertation, University of California, Santa Barbara, July 2007.

[20] P. Barooah and J. P. Hespanha, “Distributed optimal estimation from relative measurements,” in *Proceedings of the 3rd International Conference on Intelligent Sensing and Information Processing (ICISIP)*, December 2005, pp. 226–231.

[21] G. H. Golub and C. F. van Loan, *Matrix Computations*, 3rd ed. The John Hopkins University Press, 1996.

[22] P. Barooah and J. P. Hespanha, “Estimation from relative measurements: Electrical analogy and large graphs,” *IEEE Transactions on Signal Processing*, vol. 56, no. 6, pp. 2181–2193, June 2008.

[23] P. Barooah, “The maximum effective resistance in 2-dimensional and 3-dimensional grids: Upper bounds from Wu’s formula.” University of Florida, Tech. Rep., February 2008. [Online]. Available: http://humdoi.mae.ufl.edu/~prabirbarooah/publications.html

APPENDIX A
PROOFS

Proof of Proposition 1: We assume without loss of generality that the measurement graph grows in length along the x-axis, while the width of the graph (along the y axis) stays constant as time progresses. We assume that the only reference node is the initial position of the first robot, which is the node at the lower left hand corner of the graph. The result does not change if GPS reference is available at $t = 0$, or if the position of any other robot is used as a reference. It was shown in [22] that the BLUE error covariance of a node in a measurement graph $\mathcal{G}$ is equal to the effective resistance between the node and the reference in the generalized electrical network constructed from the graph $\mathcal{G}$ by assigning to each edge e a matrix-valued effective resistance R_e that is equal to P_e, where P_e is the covariance matrix of the measurement noise on the edge e. Therefore the error covariance of robot i’s BLUE estimation error at time t, denoted by $P_i(t)$, is equal to the effective resistance between the node that corresponds to i’s current position in $\mathcal{G}(t)$ and the reference node o. The index of the node that corresponds to robot i’s position at time t in the graph $\mathcal{G}(t)$ is denoted by $u(i,t)$. Therefore

$$P_i(t) = R^{\text{eff}}_{u(i,t),o}(\mathcal{G}(t), P_e) = R^{\text{eff}}_{u(i,t),o}(\mathbb{Z}_{t\times n}, P_e)$$

where $R^{\text{eff}}_{u(i,t),o}(\mathcal{G}(t))$ denotes the effective resistance in the generalized electrical network $(\mathcal{G}(t), P_e)$, and $\mathbb{Z}_{a\times b}$ denotes a $a \times b$ rectangular grid. The last equality follows from the hypothesis about the motion of the robots. It was also shown in [22] that effective resistance obeys a monotonicity law, which states that is the edge resistances are increased, the effective resistance between any two nodes can only increase. Combining this with the equation above, we get

$$P_i(t) \leq R^{\text{eff}}_{u(i,t),o}(\mathbb{Z}_{t\times n}, \gamma I)$$

where $\gamma I \geq P_e$ is an upper bound on the measurement error covariances (uniform w.r.t. time t). Using Lemma 6 from [22], we get

$$R^{\text{eff}}_{u(i,t),o}(\mathbb{Z}_{t\times n}, \gamma I) \leq R^{\text{eff}}_{\max}(\mathbb{Z}_{t\times n}, 1)\gamma\, I$$

where $R^{\text{eff}}_{\max}(\mathbb{Z}_{t\times n}, \gamma)$ is the *scalar* maximum effective resistance in a $t \times n$ rectangular grid with $1\ \Omega$ resistance on every edge and I is an identity matrix of appropriate dimension. It was shown in [23] that the effective resistance between an arbitrary pair of nodes in a $N \times M$ rectangular grid of resistors with $1\ \Omega$ resistances on every edge is upper bounded by $M/N + N/M + 3\log(MN) + 6$. Using this result, we get

$$P_i(t) \leq \left(\frac{t}{n} + \frac{n}{t} + 3\log(nt) + 6\right)\, I$$

Since n, the number of robots is fixed, while t, the discrete time index, grows without bound, we get

$$P_i(t) = \Theta(\frac{t}{n}) + O(\log(nt))$$

When there is a single robot, the error covariance of the BLUE estimate of its position at time t is simply the effective resistance between two ends of a 1D graph of length t, which is $\Theta(t)$ [22]. This concludes the proof. ■