Collaborative localization with heterogeneous inter-robot measurements by Riemannian optimization

Joseph Knuth and Prabir Barooah

***Abstract*—We propose a distributed algorithm for collaborative localization of multiple autonomous robots that fuses inter-robot relative measurements with odometry measurements to improve upon dead reckoning estimates. It is an extension of our previous work [7], in which a method for fusing inter-robot *pose* measurements was presented. In this paper we extend the method to fuse any type of inter-robot measurements (distance, bearing, relative position, relative orientation, and any combination thereof), thus increasing the applicability of the method. The proposed method is posed as an optimization problem in a product Riemannian manifold; and is solved by gradient descent without performing a parameterization of the orientations. The proposed distributed algorithm allows each robot to compute its own pose estimate based on local measurements and communication with its neighbors. Simulations - as well as experiments with a pair of ground robots - show that the proposed distributed algorithm significantly improves localization accuracy over the case of no-collaboration. Simulations show that, in some situations, the proposed distributed algorithm outperforms two competing methods - an Extended Kalman Filter-based algorithm as well as a distributed pose graph optimization method that relies on an Euclidean parameterization of orientations.**

I. Introduction

Interest in utilizing teams of autonomous mobile robots has grown in recent years, since multi-robot teams can provide increased functionality over a single robot. In search and rescue operations, a group of robots can cover a larger area than a single robot. In hazardous conditions, the innate redundancy of a group of robots may be necessary to prevent catastrophic loss of mission capability. Irrespective of the application, localization is a crucial capability for any mobile robot.

Though GPS provides global measurements of a robots position, GPS measurements may not be available in many situations, or may be only intermittently available. For example, a group of unmanned aerial vehicles (UAVs) operating in an urban environment may temporarily lose GPS measurements when the signal is blocked by large buildings. In such a situation, the absolute pose (position and orientation) can be obtained by integrating over the relative pose measurements obtained using IMUs or vision based sensors and using a known initial pose. This method of localization is referred to as *dead reckoning*. Localization through dead reckoning can lead to a rapid growth in localization error [15]. Within a team of collaborative robots, though, relative measurements between pairs of robots - such as bearing, distance etc. - may be available. These provide additional information on the robots' pose that can be used to improve localization accuracy over dead reckoning. This has led to research into collaborative localization, which is the topic of this paper.

We assume all robots are equipped with proprioceptive sensors (vision, IMU, etc.) allowing each robot to measure its change in pose. We refer to these noisy measurements as *inter-time relative pose measurements*. Using these noisy measurements, each robot can perform localization through dead reckoning. In addition, we assume each robot is equipped with exteroceptive sensors, allowing intermittent noisy measurements of *inter-robot relative measurements* between pairs of robots, which can be one or more of the following: relative bearing, relative distance, relative orientation, relative position, and relative pose.

We propose a distributed algorithm to perform collaborative localization by fusing the inter-time and inter-robot relative measurements to obtain an estimate of the absolute pose of every robot that improves over dead-reckoning estimates. With this algorithm, which we call the D-RPGO (Distributed Riemannian Pose Graph Optimization) algorithm, communication is only necessary between pairs of robots for which an inter-robot relative measurement has been obtained. The method behind the algorithm is an extension of the method in our previous work [7], which was limited to inter-robot relative pose measurements. In this paper we extend it to handle any type of relative measurement, and provide performance comparison of the D-RPGO algorithm with two alternate collaborative localization algorithms.

A. Related work

Most of the work on collaborative localization can be roughly classified as probabilistic; they provide estimates of the robots' poses as well as some measure of the associated uncertainty. These include [3, 10], where the pdf (probability density function) of robots' positions (or poses) are improved by fusing inter-robot measurements with inter-time measurements. Quite a few works are based on the extended Kalman filter (EKF); see [18, 17, 13, 5, 14] and references therein. The papers [16, 4], which are based on the particle filter, also belong to this category.

In *single* robot localization and mapping, an alternative to filtering is the pose graph based approach. This can be extended to multi-robot localization with map merging among robots [2] or without [6]. In the pose graph approach the problem is described in terms of a graph whose nodes correspond to robot poses and edges represent measurements that provide constraints between certain pairs of poses. The best set of pose estimates are found by minimizing a cost function that quantifies how well a given set of poses explain the measurements. Under certain assumptions on the measurement noise, the solution is the maximum likelihood or maximum a-posteriori estimate of the poses. Without any assumption, they still yield a least squares solution. Pose graph optimization methods used to compute the non-linear least squares solution in SLAM and bundle adjustment (see [19, 9] and references therein) can be directly applied to the multi-robot case, though distributing the computations among the robots present additional challenges [6]. The pose graph based methods typically only estimate the poses and not a measure of the associated uncertainty; though recent work has made progress in this direction; see [20] and references therein.

B. Contributions

In our earlier paper [7], we proposed a method for collaborative localization that required relative *pose* measurements between robots. In this paper we extend the method so that any type of inter-vehicle measurement (distance, bearing, relative position, relative orientation, relative pose, or some combination thereof) between pairs of robots can be fused. This increases the practical applicability of the method.

The method proposed here, as well as that in [7], can be classified as a pose graph based approach: the estimation problem is formulated as an optimization defined by a graph, where nodes represent robot poses at various times and edges represent inter-time and inter-robot measurements. A major - but subtle - distinction between our method and existing pose graph optimization methods is that while existing methods use a vector space parameterization of orientation (such as the complex part of the unit quaternion representation) and then search for the minima in that space, we perform the optimization directly on the product Riemannian manifold in which the problem is naturally posed, without relying on a specific parameterization. A gradient-descent method on the Riemannian manifold is used for searching for the optima that is independent of the parameterization as well; any parameterization of the orientations can be used for numerical implementation. The advantages of doing so are discussed in Section III (see Remark 1). Simulations show that the proposed method based on Riemmanian manifold pose graph optimization (RPGO) outperforms the traditional pose graph optimization performed by vector-space based methods, which we call Euclidean pose graph optimization (EPGO).

In comparison with the EKF-based collaborative localization methods described above, our method offers a distinct advantage. The linearization involved in the EKF requires a small angle approximation to hold at all times. Unless the time interval between two successive inter-vehicle measurements is extremely small, which is unlikely in many practical situations, the small angle approximation is violated, which is likely to lead to poor covariance updates and poor pose estimates. This is verified through simulations presented here.

The EKF estimator we use is inspired by that in [14], while the Euclidean pose graph optimization method we use is an implementation of standard approaches [9].

II. Problem statement

A. The Collaborative Localization Problem

Consider a group of r mobile robots indexed by $i = 1, \ldots, r$. Time is measured by a discrete counter $k = 0, 1, 2, \ldots$. Each robot i is equipped with a local, rigidly attached reference frame, called *frame* i. Localization of robot i consists of estimating the Euclidean transformation $\mathbf{T}_i \in SE(3)$ that relates a robot's local reference frame to an absolute reference frame common to all robots. This transformation is referred to as the robot's *absolute pose*.

Measurements of a robot's absolute pose from GPS and compass are either not available or only rarely available. Instead, we assume that each robot is equipped with sensors (such as inertial sensors or vision based sensors) and perhaps sensor fusion algorithms such that, at every time k, a robot is able to obtain a inter-time relative pose measurement: a measurement of the transformation between the previous and the current pose. In addition, each robot is equipped with exteroceptive sensors so that the robot is able to uniquely identify each robot it can "see" (within some sensing radius), and obtain a relative measurement for each such robot with respect to itself. When robot i collects a measurement of robot j, it can be one of the following:

- **Relative pose**: The Euclidean transformation between the reference frame attached to robot i and the reference frame attached to robot j, expressed in robot i's reference frame. Denoted by the symbol $\mathbf{T}$.
- **Relative Orientation**: The element of $SO(3)$ that describes the change in orientation between frame i and frame j, expressed in frame i. Denoted by the symbol $\mathbf{R}$.
- **Relative position**: The vector in $\mathbb{R}^3$ that describes the change in position between robot i and robot j, expressed in frame j. Denoted by the symbol $\mathbf{t}$.
- **Relative bearing**: The vector of unit length that points from robot i to robot j, expressed in frame i. Denoted by the symbol $\boldsymbol{\tau}$.
- **Relative distance**: The distance between robot i and robot j. Denoted by the symbol δ.

These are called inter-robot relative measurements. The *collaborative localization problem* is to estimate the absolute pose of every robot by utilizing both the inter-time and inter-robot measurements. The situation above is best described in terms of a directed, time-varying, fully-labeled *graph* $\mathcal{G}(k) = (\mathcal{V}_0(k), \mathcal{E}(k), \ell(k))$ that shows how the noisy relative measurements relate to the absolute pose of each robot at every time step. The graph is defined as follows. For each robot $i \in \{1, \ldots, r\}$ and each time $t \leq k$, a unique index (call it u) is assigned to the pair (i, t). How this indexing is done is immaterial. The set of these indices $\{1, \ldots, rk\}$

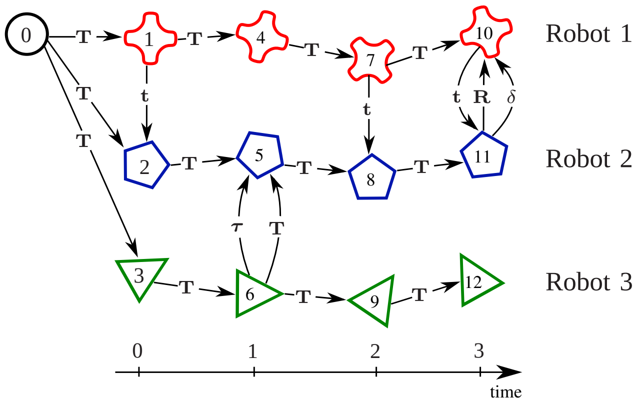

Fig. 1. Snapshot of a measurement graph at time $k = 3$ for a group of three robots. Each (robot,time) pair is labeled with the corresponding node index from $\mathcal{V}_0(3)$. Arrows indicate edges, i.e., relative measurements, in $\mathcal{E}(3)$. Each edge is labeled to indicate the type of measurement. Robots 1 and 3 had GPS measurements at the initial time $k = 0$. Thereafter, no other GPS measurements were available.

define the set $\mathcal{V}(k)$ and the node set of the graph is defined as $\mathcal{V}_0(k) := \mathcal{V}(k) \cup \{0\}$. We then refer to the reference frame attached to robot i at time t as *frame* u. Node u is associated with the absolute pose of robot i at time t relative to the common reference frame, i. e., the Euclidean transformation between the common reference frame and frame u expressed in the common frame. We call these poses *node variables* and denote them $\mathbf{T}_u$. The common reference frame with respect to which all node variables are expressed is associated with node 0. If the absolute pose of at least one robot is known at time 0, perhaps through the use of a GPS and compass, then node 0 can be associated with the common reference frame. When absolute pose measurements are not available, node 0 could correspond to the initial reference frame of one of the robots. In either case, estimating the node variables is equivalent to determining the robots' poses with respect to frame 0 (the reference frame associated with node 0). Node 0 is therefore called the *reference node*.

The set of *directed edges* at time k, denoted $\mathcal{E}(k)$, corresponds to the noisy inter-time and inter-robot measurements collected up to time k. That is, suppose robot i is able to measure robot j's relative pose at time $\bar{k}$, and let u, v be the nodes corresponding to robots i, j at time $\bar{k}$, respectively. Then the edge $e = (u, v)$ will be in $\mathcal{E}(k)$ for all $k \geq \bar{k}$. Similarly, each inter-time relative pose measurements of a robot also creates an edge in the graph. To delineate the type of measurement, a label from the set { $\mathbf{T}$ (pose), $\mathbf{R}$ (orientation), $\mathbf{t}$ (position), $\boldsymbol{\tau}$ (bearing), δ (distance) } is attached to each edge. The map from the set of edges to the set of labels is given by $\ell(k)$. Recall our example in which the edge $e = (u, v)$ is associated with a relative pose measurements. The label for edge e is then given by $\ell(k)(e) = \mathbf{T}$ (pose). The noisy relative measurement associated with each edge $e = (u, v) \in \mathcal{E}(k)$ is denoted by $\hat{\mathbf{T}}_{u\,v}, \hat{\mathbf{R}}_{u\,v}, \hat{\mathbf{t}}_{u\,v}, \hat{\boldsymbol{\tau}}_{u\,v},$ or $\hat{\delta}_{u\,v}$ for $\ell(k)(e) = \mathbf{T}$ (pose), $\mathbf{R}$ (orientation), $\mathbf{t}$ (position), $\boldsymbol{\tau}$ (bearing), and δ (distance) respectively.

The graph $\mathcal{G}(k)$ is called the *measurement graph* at time k; see Figure 1 for an example. We make the following assumption to ensure at least one estimate exists for every robot at each time k:

Assumption 1. *All inter-time relative measurements are of the full relative pose and each robot has access to an estimate of its current absolute pose at time* 0.

This assumption will often hold in practice, as any robot capable of localizing itself using dead-reckoning must estimate the full relative pose between each time step. Under this assumption, an estimate for the pose of robot i at time time k (or equivalently, the value of the node variable $\mathbf{T}_u$, where node $u \in V(k)$ corresponds to the pair (i, k)) can be computed by composing the inter-time relative pose measurements obtained by robot i up to time k. This estimate is equivalent to robot i performing dead-reckoning. *The goal of collaborative localization is to fuse information available from all edges in the graph $\mathcal{G}(k)$ to obtain estimates of the robots' poses at time k that is more accurate than that possible from dead reckoning, which only uses single paths from the reference to the current robot pose.*

In section III we propose a *centralized algorithm* to solve the collaborative localization problem that uses all the measurements in the graph $\mathcal{G}(k)$ to compute the "best" estimate for each node variable. In Section IV, we propose a *distributed algorithm* to compute an estimates of each robot's current pose. The distributed algorithm has the additional constraints that (i) at every time step, each robot must only be required to communicate with other robots it can "see" or that can "see" it, and (ii) memory, processor power, and communication bandwidth are limited. For the distributed algorithm, *we assume that the communication range is greater then the measurement range*. It is always possible to satisfy this assumption by dropping any measurements between robots that are unable to communicate.

III. Centralized Algorithm

In the centralized case, we assume that the relative measurements are instantly available to a central processor at each time k. The problem of estimating the robots' poses at time k is embedded in the problem of estimating all the node variables of the measurement graph $\mathcal{G}(k)$

$$\{\mathbf{T}\}_{\mathcal{V}(k)} := \{\mathbf{T}_u \in SE(3) : u \in \mathcal{V}(k)\} \tag{1}$$

using the robots' past noisy relative measurements (both inter-time and inter-robot). We estimate the node variables by minimizing a cost function of these variables that measures how well a given set of node variables (absolute poses) explains the noisy measurements. The initial guess for each node variable $\mathbf{T}_u$, $u \in \mathcal{V}(k)$ is taken to be the dead-reckoning estimate, whose existence is guaranteed by Assumption 1.

The cost function we propose is of the form

$$f(\{\mathbf{T}\}_{\mathcal{V}(k)}) := \frac{1}{2} \sum_{(u,v)=e \in \mathcal{E}(k)} g_e(\mathbf{R}_u, \mathbf{t}_u, \mathbf{R}_v, \mathbf{t}_v) \tag{2}$$

where the scalar, non-negative valued function $g_e(\mathbf{R}_u, \mathbf{t}_u, \mathbf{R}_v, \mathbf{t}_v)$ is designed to be a measure of how well the noisy relative measurement associated with the edge $e = (u, v)$ fit estimates of the relevant node variables $\mathbf{R}_u, \mathbf{t}_u, \mathbf{R}_v, \mathbf{t}_v$. Here $\mathbf{R}_u \in SO(3)$ and $\mathbf{t}_u \in \mathbb{R}^3$ denote the rotation and translation components that make up the node variable $\mathbf{T}_u$. For an illustrative example, consider the relative rotation measurement $\hat{\mathbf{R}}_{u\,v}$ associated with the edge $e = (u, v)$. By definition, $\hat{\mathbf{R}}_{u\,v}$ is a measurement of the rotation between frame u and frame v, expressed in frame u. This same rotation is also equal to $\mathbf{R}_u^T \mathbf{R}_v$, where $\mathbf{R}_u^T$ is the adjoint of the operator $\mathbf{R}_u$. Therefore, when no noise is present in the measurement, $\hat{\mathbf{R}}_{u\,v}$ is equal to $\mathbf{R}_u^T \mathbf{R}_v$. When measurements are corrupted by noise, the distance between the two quantities, with $\mathbf{R}_u$ and $\mathbf{R}_v$ replaced by their estimates, measured by a suitable metric on $SO(3)$, provides a measure of how well the estimates of $\mathbf{R}_u$ and $\mathbf{R}_v$ fit the noisy measurement. Therefore, a suitable cost function for a relative orientation measurement $\hat{\mathbf{R}}_{u\,v}$ associated with the edge $e = (u, v)$ is given by

$$g_e(\mathbf{R}_u, \mathbf{t}_u, \mathbf{R}_v, \mathbf{t}_v) = d^2(\hat{\mathbf{R}}_{u\,v}, \mathbf{R}_u^T \mathbf{R}_v), \tag{3}$$

where $d(\,\cdot\,,\cdot\,)$ is the Riemannian distance: $d(A, B) = \sqrt{-\frac{1}{2}\operatorname{Tr}\left(\log^2(A^T B)\right)}$, $A, B \in SO(3)$, where $\operatorname{Tr}()$ denotes trace. Using arguments similar to the one presented above for orientation measurements, appropriate cost functions for all measurement types are constructed:

$$g_e(\mathbf{R}_u, \mathbf{t}_u, \mathbf{R}_v, \mathbf{t}_v) = \begin{cases} d^2(\hat{\mathbf{R}}_{u\,v}, \mathbf{R}_u^T \mathbf{R}_v) + \|\hat{\mathbf{t}}_{u\,v} - \mathbf{R}_u^T(\mathbf{t}_v - \mathbf{t}_u)\|^2 & \text{if } \ell(k)(e) = \mathbf{T} \\ d^2(\hat{\mathbf{R}}_{u\,v}, \mathbf{R}_u^T \mathbf{R}_v) & \text{if } \ell(k)(e) = \mathbf{R} \\ \|\hat{\mathbf{t}}_{u\,v} - \mathbf{R}_u^T(\mathbf{t}_v - \mathbf{t}_u)\|^2 & \text{if } \ell(k)(e) = \mathbf{t} \\ \|\left(\hat{\boldsymbol{\tau}}_{u\,v}\|\mathbf{t}_v - \mathbf{t}_u\|\right) - \mathbf{R}_u^T(\mathbf{t}_v - \mathbf{t}_u)\|^2 & \text{if } \ell(k)(e) = \boldsymbol{\tau} \\ \|\left(\hat{\delta}_{u\,v} - \|\mathbf{t}_v - \mathbf{t}_u\|\right)\|^2 & \text{if } \ell(k)(e) = \delta \end{cases} \tag{4}$$

The "best" set of pose estimates is a solution to the following optimization problem:

$$\{\mathbf{T}^*\}_{\mathcal{V}(k)} = \min_{\{\mathbf{T}\}_{\mathcal{V}(k)} \in (\mathbb{R}^3 \times SO(3))^{n(k)}} f(\{\mathbf{T}\}_{\mathcal{V}(k)}), \tag{5}$$

where $n(k)$ is the cardinality of the set $\mathcal{V}(k)$. Finding the minimum of a function defined over a vector space has been studied extensively. However the function $f(\,\cdot\,)$ in (2) is defined on a curved surface, specifically, the product *Riemannian Manifold* $(SO(3) \times \mathbb{R}^3)^{n(k)}$. We search for the minima by using recently developed techniques for optimization over Riemannian manifolds [1].

Remark 1 (Relation to Euclidean pose graph optimization)**.** *The problem* (5) *is very similar to the least squares problem that appears in pose graph optimization used for SLAM and bundle adjustment [19, 9]. In existing pose graph optimization methods, rotations are first parameterized and then edge-costs are defined as squared-error terms defined over the vector space of those parameters. The cost on an edge* $e = (u, v)$ *would be of the form* $\mathbf{e}_e^T Q_e \mathbf{e}_e$*, where* Q_e *is a covariance matrix and the vector* $\mathbf{e}_e$ *is an error. The cost function to minimize is then defined as*

$$f(\mathbf{p} \in \mathbb{R}^{D(k)}) := \frac{1}{2} \sum_{e \in \mathcal{E}(k)} \mathbf{e}_e^T Q_e \mathbf{e}_e, \tag{6}$$

where $\mathbf{p}$ *is a vector of Euclidean parameterization of the* $n(k)$ *poses to be determined, whose dimension* $D(k)$ *is* $6n(k)$ *with minimal parameterization. The cost function is then minimized by methods such as gradient descent, Levenberg-Marquardt, etc. on the real coordinate space* $\mathbb{R}^{D(k)}$*. The problem* (5) *differs from* (6) *both in formulation and the solution method we employ. The orientations (such as* $\mathbf{R}_{(\,\cdot\,)}$*) that appear in our problem formulation are abstract rotation operators, or elements of* $SO(3)$*, and not rotation matrices or quaternions or any parameterization thereof. In contrast,* (6) *is defined in terms of a specific parameterization. For instance, if the measurement on* $e = (u, v)$ *is that of the relative orientation, one possible choice of the error is* $\mathbf{e}_e = q(\mathbf{R}_u^{-1}) \otimes q(\mathbf{R}_v) - q(\hat{\mathbf{R}}_{uv})$*, where* $q(\,\cdot\,)$ *denotes the unit-quaternion representation of its argument and* $\otimes$ *represent quaternion multiplication. Optimization is performed, e.g., over the first three entries of the unit-quaternion parameterization, which makes the cost defined over a vector space. Another possibility is* $\mathbf{e}_e = vec(M(\mathbf{R}_u^{-1}) \otimes M(\mathbf{R}_v) - M(\hat{\mathbf{R}}_{uv}))$*, where* $M(\,\cdot\,)$ *is the* 3×3 *matrix representation, and* $vec(M)$ *is the stacked vector of all the columns of the matrix* M*. Each such choice of the parameterization and the error – and there are many – will change the local minima as well the difficulty of searching for the minima.* □

In light of the remark above, we defined the cost (2) in the natural space of the variables (the product manifold $(\mathbb{R}^3 \times SO(3))^{n(k)}$) that is independent of any parameterization of rotations. Our next goal is to find a provably correct algorithm to solve (5) that utilizes the geometry of the space without relying on any particular parameterization. We accomplish this through use of a gradient descent algorithm on the product manifold.

We showed in [7] that given a point $\mathbf{p}_m = (\mathbf{R}_1, \mathbf{t}_1, \ldots, \mathbf{R}_n, \mathbf{t}_n)$ in the product manifold, the point $\mathbf{p}_{m+1}$ resulting from one step of the gradient descent algorithm is given by

$$\begin{aligned} \mathbf{p}_{m+1} = (&\mathbf{R}_1 \exp(\mathbf{R}_1^T \xi_{\mathbf{R}_1}), \mathbf{t}_1 + \xi_{\mathbf{t}_1}, \ldots, \\ &\mathbf{R}_n \exp(\mathbf{R}_n^T \xi_{\mathbf{R}_n}), \mathbf{t}_n + \xi_{\mathbf{t}_n}) \end{aligned} \tag{7}$$

where $\exp(\,\cdot\,)$ is the Lie-group exponential map [11], $\xi_{\mathbf{R}_i} = -\eta_m \sum_{e \in \mathcal{E}(k)} grad\, g_e(\mathbf{R}_i)$, and $grad\, g_e(\mathbf{R}_i)$ denotes the gradient of the edge cost g_e with respect to the node variable $\mathbf{R}_i$. The positive scalar η_m is the step-size, determined using a line search algorithm. Similarly, $\xi_{\mathbf{t}_i} = -\eta_m \sum_{e \in \mathcal{E}(k)} grad\, g_e(\mathbf{t}_i)$.

The gradient with respect to relative pose measurements were derived in [7]. The innovation in this paper is computation of gradients for all types of measurements. When edge $e = (u, v)$ corresponds to a bearing measurement, it can be shown after some tedious calculations that $grad\ \ g_e(\mathbf{R}_h) =$

$-2\mathbf{R}_h\Big(\mathbf{R}_h^T(\mathbf{t}_v - \mathbf{t}_h)\hat{\boldsymbol{\tau}}_{uv}^T\|\mathbf{t}_u - \mathbf{t}_v\| - \hat{\boldsymbol{\tau}}_{uv}\|\mathbf{t}_v - \mathbf{t}_u\|(\mathbf{t}_v - \mathbf{t}_u)^T\mathbf{R}_h\Big)$ if $h = u$, and 0 otherwise. Similarly, $grad\ g_e(\mathbf{t}_h) = -4I_{uv}(h)[(\mathbf{t}_v - \mathbf{t}_u) - \|\mathbf{t}_v - \mathbf{t}_u\|\mathbf{R}_u\hat{\boldsymbol{\tau}}_{uv}]$. where $I_{uv}(h)$ equals 1 if $h = u$, -1 if $h = v$, and 0 otherwise. The formulas for the gradients for all other types of measurements can found in [8]; we do not provide them here due to lack of space. Gradient descent is performed using the update law given in (7), terminating when the norm of the gradient falls below some user specified threshold. Theorem 4.3.1 in [1] guarantee that this algorithm converges to a critical point of the cost function f defined in (2).

The algorithm presented above, which we call the RPGO algorithm, is independent of the parameterization used to represent rotations. One could use any parameterization during numerical implementation, and the choice of parameterization does not affect the minimum obtained.

IV. Distributed Algorithm

The method used to compute estimates in a distributed way is the same as the one proposed in [7]. Therefore we describe the distributed algorithm very briefly; the interested reader is referred to [7] for details.

Two robots are called neighbors at time k if at least one of them obtains a relative measurement of the other at that time. Let $N_i(k)$ be the *neighbors* of robot i at time k. We assume that each robot can communicate with its neighbors during each time step. Robot i then forms a *local* measurement graph $\mathcal{G}_i(k) = (\mathcal{V}_i(k), \mathcal{E}_i(k), \ell_i(k))$, whose node set is simply the neighbors of i at time k along with the reference node 0 and i itself. The edges of $\mathcal{G}_i(k)$ correspond to the inter-robot measurements at time k between i and its neighbors, along with an edge $e = (0, j)$ for each $j \in \mathcal{V}_i(k)$. Each robot j obtains its current estimate $\hat{\mathbf{T}}_{0j}$ at time k by concatenating the robot's pose estimate obtained at time $k-1$ with the noisy inter-time relative pose measurement describing the robots motion from $k-1$ to k. This estimate is then used as the measurement associated with edge $e = (0, j)$. At each time k, every robot $i \in \{1, \ldots, r\}$ uses the estimate $\hat{\mathbf{T}}_{0i}$ so obtained as an initial pose estimate and then updates it by using the RPGO algorithm on its local measurement graph at that time. After the computation, each robot transmits the updated absolute pose estimate to each of its current neighbors. The process keeps repeating as the robots keep moving.

After the computations at time k, a robot only stores the estimated value its own current pose. Note that if robot i has no neighbors at time k, the distributed collaborative localization algorithm is equivalent to performing self-localization from inter-time relative measurements. Since the distributed algorithm is simply the centralized algorithm applied to a local measurement graph, it inherits the correctness property of the centralized algorithm as well.

The proposed distributed algorithm is called the D-RPGO algorithm, since it is a distributed version of the RPGO algorithm. The D-RPGO algorithm only requires measurements that the robot can collect with on-board sensors and communication with nearby robots. Moreover, computational complexity of the algorithm grows only with the number of neighbors, since it depends on the size of the local subgraph, and not with the total number of neighbors. This makes the algorithm highly scalable to large teams of robots.

V. Simulation Results

We now present simulations (i) that show the improvement in localization accuracy with the proposed D-RPGO algorithm over self-localization, and (ii) comparison with alternate collaborative localization algorithms. We first define some performance metrics. The position estimation error of robot i is defined as $\mathbf{e}_i(k) := \hat{\mathbf{t}}_i(k) - \mathbf{t}_i(k)$, where $\mathbf{t}_i(k)$ is its absolute position at k and $\hat{\mathbf{t}}_i(k)$ is the estimate. The bias in the position estimation error of robot i is defined as $\|\mathrm{E}[\mathbf{e}_i(k)]\|$, where $\|\cdot\|$ is the 2-norm and E denotes expectation. The standard deviation is defined as $\sqrt{\mathrm{Tr}\left(Cov(\mathbf{e}_i(k), \mathbf{e}_i(k))\right)}$, where $Cov(\cdot)$ stands for covariance. In each scenario described below, the bias and variance in position estimation error is estimated through the use of a Monte Carlo simulation with $1,000$ sample runs.

A. Performance with various relative measurements

A group of robots are simulated traveling along randomly generated distinct zig-zag paths in 3-D space, so that all translational and rotational coordinates vary along time for each robot. Two robots can obtain relative pose measurements at time k if the Euclidean distance between them at that time is less than $7\ m$. Furthermore, 25% of these potential measurements were dropped, simulating random failure and insuring the measurement graph would not be symmetric. Error in measurements of the relative orientation are induced by composing $C(\mathbf{q})$ with the rotation corresponding to the true orientation, where $C(\mathbf{q})$ is the rotation operator corresponding to the unit quaternion $\mathbf{q}$ drawn from a Von Mises-Fisher distribution centered about the identity [12]. Similarly, noisy relative bearing measurements are generated by applying the random $C(\mathbf{q})$ to the vector in unit sphere in 3-D that describes the true bearing. Measurements of relative distance and position are normally distributed with mean corresponding to the true values.

The bias and standard deviation in position estimation error for robot 1 for different scenarios, each with a distinct type of noisy inter-robot measurement, are reported in figure 2. We see from the figure that improvement over single robot localization in both bias and standard deviation occurs for all inter-robot measurement types with one exception. Though distance measurements improve the standard deviation of the position estimates, they have little effect on the bias. As expected, full relative pose provides the most benefit to localization accuracy. However, other measurement types also leads to improvement over self-localization. For the scenario and time duration considered, the three types of relative measurements - bearing, position and orientation - seem to have somewhat similar degree of benefit.

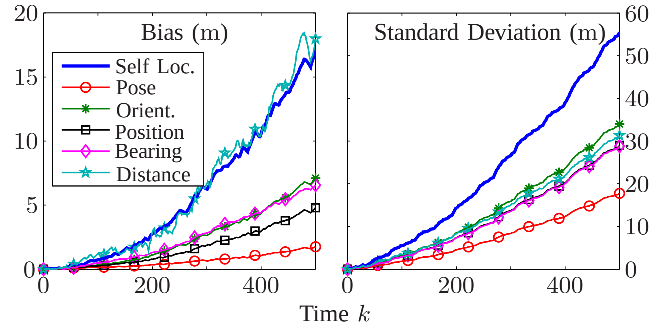

Fig. 2. Simulation: bias and standard deviation in robot 1's position estimation error, when each robot in a group of 5 robots use the D-RPGO algorithm to estimate their poses. Each curve corresponds to the robots using a distinct type of inter-robot relative measurements (indicated by the labels). The label "Self Loc." refers to a robot using dead reckoning alone, without using any inter-robot relative measurements.

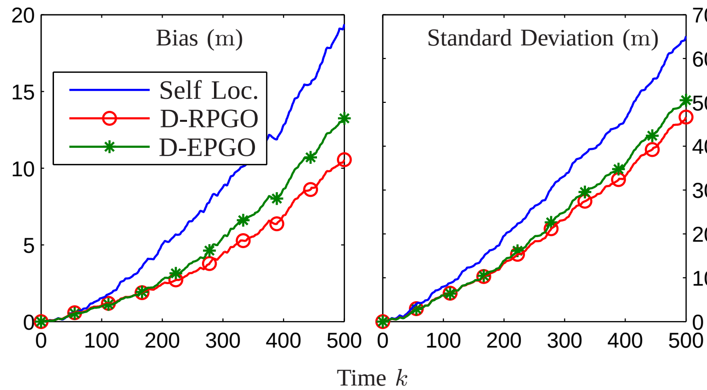

Fig. 3. Simulation: comparison between D-RPGO and D-EPGO algorithms. The position estimation error of robot 1 (in a group of 5 robots utilizing noisy inter-robot relative pose measurements), computed with both algorithms, with the same input data are shown. The label "Self Loc." refers to a robot localizing by dead reckoning alone.

B. *Comparison with alternate methods of collaborative localization*

We now present simulations that provide some insight into how the D-RPGO algorithm performs when compared with two state-of-the-art collaborative localization algorithms. All inter-robot relative measurements are of the relative pose for these simulations.

The first alternative we consider is a standard pose graph formulation, in which optimal robot poses are computed by minimizing the cost function (6). Rotations are parameterized by the complex part of the corresponding unit-quaternion, and the optimization problem is set up as in [9]. Searching for the optima is performed by Levenberg-Marquardt algorithm. When the optimal poses of a measurement graph are obtained in this way, we call it a solution from a EPGO algorithm. To maintain comparability, we provide the same local measurement graph to both the D-RPGO algorithm as well as a EPGO algorithm, which we call the D-EPGO algorithm.

A group of 5 are robots are simulated to move along the 3-D path described above. Error in the pose measurements were induced as in simulations in Section V-A. Simulations were performed varying the concentration parameter K in the Von Mises-Fisher distribution from which the noisy rotations (quaternions) used to corrupt the inter-robot orientation measurements are drawn. These simulations show that, when K is very large, that is, the variance is very low, D-EPGO does very well, even outperforming the D-RPGO algorithm. However, when K is small, that is, the noise variance is large, D-RPGO outperforms D-EPGO. Due to a lack of space, only the results for $K = 100$ are shown; see Figure 3. For this specific case, it is clear that the proposed D-RPGO algorithm outperforms the D-EPGO algorithm. A detailed comparison is a topic of future work.

Finally, we consider a method for collaborative localization using an Extended Kalman Filter (EFK), developed in a similar manner as the EKF observer for collaborative localization in [14]. An indirect form filter is used, with error between the true pose and the estimated pose (before fusing the current inter-robot measurements) being the filter state. A pair of robots is simulated traveling along distinct sinusoidal paths in 3-D space. Measurements are generated as described earlier.

The trends observed from extensive simulations can be summarized as follows. When the time interval between successive inter-robot measurements, call it ΔT, is small, the EKF performs as well, or better than, the D-RPGO algorithm. However, when the time between measurements is large, the D-RPGO algorithm provides significantly better estimates of the robots' poses compared to the EKF. Figure 4 provides numerical results for the case of a large ΔT (30 seconds in this example), when EKF performs poorly. How small ΔT has to be for EKF to perform well depends on many factors, including the motion of the robots, noise in the measurements, etc. For the parameters used in the simulations mentioned above, ΔT has to be smaller than 0.1 sec for the EKF to perform as well as the D-RPGO algorithm.

We believe the reason for this behavior of the EKF is the error introduced by the linearization involved in covariance propagation. The linearized state equations rely on the assumption that the angle between the true and estimated orientation is very small. When the time interval between inter-robot measurements is sufficiently small, this approximation holds. In that case the error in the covariance matrix due to linearization is small enough that it does not out weigh the added benefit of using covariance information. However, the small angle approximation is violated for large time intervals, leading to quite poor covariance estimates, which in turn lead to poor pose estimates.

VI. Experimental results

An experiment is conducted using two Pioneer P3-DX robots, shown in Figure 5. Each robot is equipped with a calibrated monocular Prosillica EC 1020 camera and wheel odometers. Measurements from these sensors were fused to

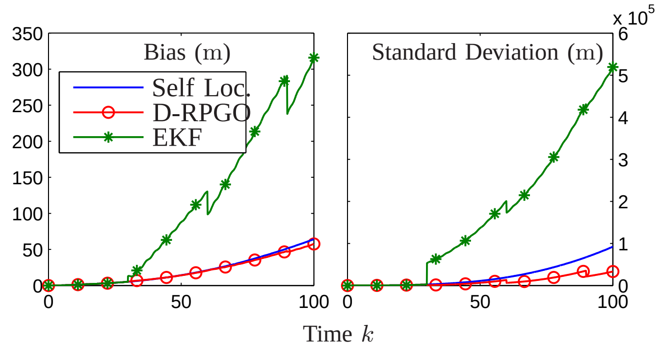

Fig. 4. Simulation: comparison between D-RPGO and EKF algorithms with $\Delta T = 30$ sec. The position estimation error of robot 1 (in a group of 2 robots utilizing noisy inter-robot relative pose measurements), computed with both algorithms. The label "Self Loc." refers to a robot localizing by dead reckoning alone.

obtain noisy inter-time relative pose measurements. Each robot is additionally equipped with a target allowing the on-board cameras to measure the inter-robot relative pose by exploiting the known geometry of each target. The true absolute pose of each robot is determined using an overhead camera capable of tracking each robot's target, providing ground truth. Noisy inter-robot relative pose measurements are obtained every 0.2 seconds.

Both robots move in straight lines that are approximately parallel. Six different pose estimates of the robots are obtained. The first estimate is obtained from dead reckoning. The remaining 5 estimates are obtained from the D-RPGO algorithm, each with a distinct type of inter-robot relative measurement: full pose, orientation only, position only, bearing only, or distance only. These measurements are obtained from the relative pose measurements by projection. Figure 6 shows the resulting position estimates obtained by the D-RPGO algorithm. As expected, a distinct improvement in localization accuracy is seen when collaborative localization is performed for all but the distance-only measurements. The experimental results reinforce the trends seen in Section V-A. Specifically, when D-RPGO algorithm is used to fuse inter-robot distance measurements, the bias in the position estimation error is similar to that without collaboration. For all other types of relative measurements, there is improvement with distributed collaborative localization over self localization. Because figure 6 is only a single realization of the estimate, the bias in position estimation error is visible, while the variance is not (cf. Figure 2).

VII. Summary and Future Work

In this paper we extended the algorithm introduced in [7] for distributed collaborative pose estimation of multiple robots. While the algorithm in [7] could only fuse relative pose measurements, which are difficult -if not impossible - to obtain in practice, the algorithm introduced here (D-RPGO) can fuse any type of relative measurements. Simulations and experiments show that distributed collaborative pose estimation with D-RPGO algorithm leads to significant improvement

Fig. 5. Two Pioneer P3-DX robots equipped cameras and targets. Robot 1 is shown on the left, while robot 2 is on the right.

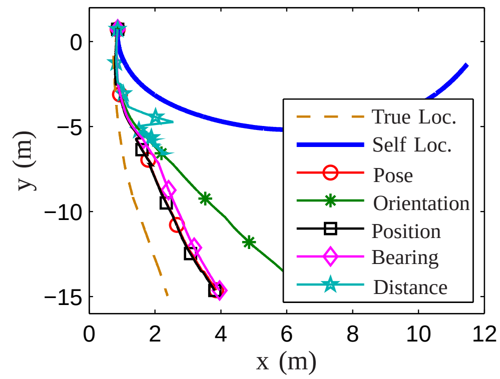

Fig. 6. Experimental: A plot of the location of robot 1 in the overhead camera reference frame when both robots move in a straight line. The true path (found using the overhead camera), estimated path using self localization, and estimated path using the distributed collaborative localization algorithm are all reported.

over self-localization. Improvements were largest when all inter-robot relative measurements are of the relative pose; but significant improvements were observed also with other types of measurements. Only distance measurements were observed to not have much benefit, especially in the bias. This is consistent with the trend observed in [13] for the 2-D case.

The proposed method is quite close in spirit to the pose graph optimization problems commonly encountered in mapping and localization. The distinction is that our cost function is defined on a Riemannian manifold and is minimized on that surface without converting the problem to a vector space optimization problem. While this distinction may seem minor, results indicate it leads to substantial performance benefit in some cases. In particular, the D-RPGO algorithm outperforms a distributed version of the traditional pose graph optimization (that uses a specific Euclidean parameterization of the robot orientations) in terms of accuracy when the noise in the inter-robot measurements is large.

It was found that the computation time of the D-RPGO algorithm is comparable to that of the D-EPGO algorithm in the simulations we performed. However, there are many ways to speed up the computations involved in Euclidean optimization that are not currently available for Riemannian optimizations. We suspect when applied to large graphs, Riemannian optimization will be slower than Euclidean optimization. Fortunately the measurement graphs that appear in distributed computation are small, since their size scales with the number of neighbors of a robot.

The EKF has been a popular tool in past work on collaborative localization as well as mapping. Simulation comparison of the proposed D-RPGO algorithm with the EKF showed that the EKF performs poorly compared to the proposed D-RPGO algorithm unless the time interval between successive inter-robot relative measurements is quite small. This has important implications for practical applications, since it is more likely that inter-robot measurements will be available quite infrequently. However, in those special cases when inter-robot measurements arrive frequently, the EKF performs just as well as the proposed method. One such scenario is the one considered in [14], where robots observe common feature points on the ground. In such a scenario the EKF maybe preferable over the proposed method.

A limitation of the proposed method over the EKF (and other filtering methods) is that the latter also provides a covariance estimate while the proposed method does not. A method to estimate the covariance, or some measure of estimation error, is an important future task. Although simulations reported here provide an indication of when the proposed method performs over existing state-of-the art methods, a more thorough study needs to be conducted to obtain a better understanding of the relative merits of the proposed method over alternatives. Future studies will also address issues such as identification of the neighboring robots, and rejection of outliers in the relative measurements.

References

[1] P. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, 2008.

[2] Lars A. A. Andersson and J. Nygards. C-SAM : Multi-robot SLAM using square root information smoothing. In *IEEE International Conference on Robotics and Automation*, Pasadena, California, USA, May 2008.

[3] Dieter Fox, Wolfram Burgard, Hannes Kruppa, and Sebastian Thrun. A probabilistic approach to collaborative multi-robot localization. *Autonomous robots*, 8(3):325–344, 2000.

[4] Andrew Howard. Multi-robot Simultaneous Localization and Mapping using Particle Filters. *The International Journal of Robotics Research*, pages 1–11, August 2011.

[5] Nadir Karam, Frederic Chausse, Romuald Aufrere, and Roland Chapuis. Localization of a group of communicating vehicles by state exchange. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, October 2006.

[6] Been Kim, M. Kaess, L. Fletcher, J. Leonard, A. Bachrach, N. Roy, and S. Teller. Multiple relative pose graphs for robust cooperative mapping. In *IEEE International Conference on Robotics and Automation*, Anchorage, Alaska, May 2010.

[7] Joseph Knuth and Prabir Barooah. Collaborative localization of multiple robots in 3-D from relative pose measurements. In *International conference on Robotics and Automation (ICRA)*, pages 1101–1106, 2012.

[8] Joseph Knuth and Prabir Barooah. Collaborative localization with heterogeneous inter-robot measurements by riemannian optimization. Technical report, University of Florida, 2012. URL http://plaza.ufl.edu/knuth/Publications/.

[9] Rainer Kummerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. g2o: A General Framework for Graph Optimization. *ICRA 2011*, pages 1–7, February 2011.

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

[11] Yi Ma, Jana Košecká, and Shankar Sastry. Optimization criteria and geometric algorithms for motion and structure estimation. *International Journal of Computer Vision*, 44(3):219–249, 2001.

[12] Kanti V. Mardia and Peter E. Jupp. *Directional Statistics*. Wiley series in probability and statistics. Wiley, 2000.

[13] A. Martinelli, F. Pont, and R. Siegwart. Multi-robot localization using relative observations. In *IEEE International Conference on Robotics and Automation*, Barcelona, Spain, April 2005.

[14] Igor V Melnyk, Joel A Hesch, and Stergios I Roumeliotis. Cooperative Vision-aided Inertial Navigation Using Overlapping Views. *IEEE International conference on Robotics and Automation*, pages 1–8, 2012.

[15] Clark F. Olson, Larry H. Matthies, Marcel Schoppers, and Mark W. Maimone. Rover navigation using stereo ego-motion. *Robotics and Autonomous Systems*, 43(4): 215–229, June 2003.

[16] I.M. Rekleitis, G. Dudek, and E.E. Milios. Multi-robot cooperative localization: a study of trade-offs between efficiency and accuracy. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, EPFL, Switzerland, September 2002.

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

[18] C. Sanderson. A distributed algorithm for cooperative navigation among multiple mobile robots. *Advanced Robotics*, 12(4):335 – 349, 1998.

[19] Gabe Sibley, Christopher Mei, Ian Reid, and Paul Newman. Adaptive Relative Bundle Adjustment. *Robotics Science and Systems*, pages 1–8, June 2009.

[20] Gian Diego Tipaldi, Giorgio Grisetti, and Wolfram Burgard. Approximate covariance estimation in graphical approaches to SLAM. In *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3460–3465.