Distributed collaborative localization of multiple vehicles from relative pose measurements

Joseph Knuth and Prabir Barooah

Abstract— **We propose a distributed algorithm for estimating the poses (positions and orientations) of multiple autonomous vehicles in GPS denied scenarios when pairs of vehicles can measure each other's relative pose in their local coordinates. Currently, navigation of an autonomous vehicle in GPS denied scenarios is achieved by integrating relative pose measurements between successive time instants that are obtained from on-board sensors, such as cameras and IMUs. However, this suffers from a high rate of error growth over time. We seek methods to ameliorate this error growth by using cooperation among a group of vehicles. Measurements of relative pose between certain pairs of vehicles provide extra information on their poses, which can be used for improving localization accuracy. We designed a distributed algorithm to fuse all the relative pose measurements to compute a more accurate estimate of all the vehicles' poses than what is possible by the vehicles individually. The algorithm is fully distributed since only neighboring vehicles need to exchange information periodically. Monte Carlo simulations show that the error in the location estimates obtained by using this algorithm is significantly lower than what is achieved when vehicles estimate their poses without cooperation.**

I. Introduction

Recent advances in several technological fronts have led to a heightened interest in autonomous vehicles, such as Unmanned Ariel Vehicles, Unmanned Underwater Vehicles and ground robots. These vehicles can be used to perform missions in which it is too dangerous for human operators to be present (fires, battle zones, areas contaminated by CBRN agents), or when space and weight limitations make presence of human operators difficult. Ability for precise navigation or localization is a crucial capability for autonomous vehicle operation.

Current localization systems for autonomous vehicles typically consist of an Inertial Measurement Unit (IMU) and Global Positioning System (GPS). In the absence of GPS, use of IMUs alone leads to a high rate of growth of localization error with time. When only IMU measurements are available, a vehicle has to integrate the relative pose (position and orientation) measurements between its frame at two successive time instants obtained from the IMUs to compute its current pose estimates. This integration accumulates error over time, which is the reason for the high growth of localization error with IMUs.

Joseph Knuth and Prabir Barooah are with the Dept. of Mechanical and Aerospace Engineering, Univ. of Florida, Gainesville, FL. This material is based upon work supported by by the Institute for Collaborative Biotechnologies through grant W911NF-09-D-001 (subcontract KK9149) from the U. S. Army Research Office.

Yet there are many situations when GPS is either not available or only intermittently available, such as in urban canyons, hilly terrains, tunnels, and under water. Extensive research is being conducted on fusing additional sensor measurements with IMU signals to reduce localization error. In particular, vision-based sensors such as scanning laser, infrared sensors, and cameras, have attracted a lot of attention for autonomous vehicle navigation. However, these sensors also face the same issue of error growth over time when no GPS measurements are available.

In this paper we investigate the potential for using cooperation among a group of vehicles to reduce the growth of localization error in autonomous vehicle navigation. In particular, we examine the situation when certain pairs of vehicles can measure their relative pose (position and orientation), albeit with some error. Such measurements can be obtained by a pair of cameras [1], with a single camera if prior knowledge about the measured vehicle's geometry is available to the measuring vehicle [2], with laser scanning sensors [3], and a combination of RF-based distance measurements and vision-based orientation measurements [4], [5]. A noisy relative pose measurement between two vehicles provides additional information on the absolute positions (and in fact, orientations) of the two vehicles. If successfully fused with the measurements of the vehicle's motion from IMUs (or from cameras), these inter-vehicle relative measurements should help in estimating the positions more accurately.

There are several challenges in using the redundant information on vehicle positions provided by the vehicle-to-vehile relative pose measurements. One is how to compute the appropriate average of poses. The relationship between the pose of a vehicle in a global reference frame and relative pose measurements between vehicle pairs is non-linear: rotation and Euclidean transformation matrices are not elements of a linear vector space. This non-linearity makes the question of "averaging" these measurements tricky. The other issue is distributing the computations involved, so that sending all measurements to a central processor is not necessary. For a team of vehicles communicating over an unreliable communication channel, distributed processing is useful in making the computations robust to the failure of team members.

Roumeliotis and Becky considered the problem of cooperative localization problem for a group of ground robots; the proposed method used a Kalman filter to fuse all measurements with the predictions of a single-integrator dynamic (linearized) model of the robots [4]. The issue of how to average poses was avoided by decomposing the dynamics along x- and y-axis and using an Euler angle representation of the rotations. Sharma and Taylor examined cooperative localization in a group of MAVs when relative orientation measurements between vehicles are available [6]. Again, all vehicles were assumed to be coplanar, so that relative orientations could be expressed in terms of a scalar angle.

The algorithm we propose in this paper to average relative poses is based on an idea first proposed by Govindu in [7] in connection with estimating camera motions. We utilize Govindu's method to construct a centralized algorithm for multi-vehicle localization. The centralized algorithm is able to improve estimates of vehicle positions significantly over the case when there is a single vehicle. However, the algorithm requires all measurements to be transmitted to a single processor, and its memory requirements grow without bound over time. We then develop a distributed algorithm that requires only limited local communication among nearby vehicles, and has bounded memory requirements. In simulations, the distributed algorithm is seen to perform almost as well as the centralized algorithm.

The rest of the paper is organized as follows. We pose the problem of cooperative localization precisely in Section II. A centralized algorithm for averaging relative pose measurements to obtain global pose estimates is described in Section III. A distributed algorithm to perform the same function is described in Section IV. Simulation results with the proposed algorithm is presented in Section V to illustrate its effectiveness.

II. Problem description

Consider a group of n vehicles moving in 3-dimensional Euclidean space. Time is measured in a discrete counter $k = 0, 1, 2, \ldots$. Each vehicle has a Cartesian frame attached to its body. The pose of a vehicle at time k is the position (of the origin of its local frame) and orientation of its local frame at time k with respect to a global (common Cartesian) coordinate frame. The global reference can be the reference that GPS uses, if GPS measurement is available at time 0, or it can simply be the initial frame of vehicle 1. In the latter case, all vehicle positions are to be estimated with respect to the initial frame of the first vehicle.

We assume that at each time step k, each vehicle r ($r = 1, \ldots, n$) can measure the Euclidean transformation between its pose at previous time $k-1$ and current time k, expressed in the coordinate frame that was attached to its body in the previous time instant. This transformation is denoted by $M_{r(k),r(k-1)}$. Note that an Euclidean transformation $M \in SE(3)$ between two Cartesian coordinate frames is a combination of rotation $R \in SO(3)$ and translation $t \in \mathbb{R}^3$:

$$M = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \tag{1}$$

We assume that every vehicle r can obtain a noisy measurement of $M_{r(k),r(k-1)}$, which is denoted by $\hat{M}_{r(k),r(k-1)}$, for $k = 1, \ldots$. It follows from (1) that the ability to measure $M_{r(k),r(k-1)}$ is equivalent to the ability to measure the rotation and translation undergone by the vehicle in the time interval between $k-1$ and k. These can be obtained by IMUs and stereo-imaging systems, and in the case of ground vehicles, from differential wheel odometry. The measurements of the form $\hat{M}_{r(k),r(k-1)}$ are called *inter-time* measurements.

We denote the common global reference frame by the symbol $\mathbf{0}$, and assume that at least one of the vehicles, say vehicle 1, can measure its initial pose with respect to the global reference. That is, $\hat{M}_{1(0),\mathbf{0}}$ is available to vehicle 1. When no GPS measurement is available, the frame attached to an arbitrary vehicle, say vehicle 1, at $k = 0$ can be taken as the global reference. In that case, $\hat{M}_{1(0),\mathbf{0}} = \hat{M}_{\mathbf{0},\mathbf{0}} = I$, and all poses are to be estimated with respect to the initial pose of the that vehicle. If GPS measurement is available to at least one vehicle at time $k = 0$, then the GPS reference frame is taken as the global reference frame.

When there is only one vehicle, say vehicle 1, the measurements available to it at time k are $\hat{M}_{1(0),\mathbf{0}}$, $\hat{M}_{1(1),1(0)}$, $\hat{M}_{1(2),1(1)}$, ..., $\hat{M}_{1(k),1(k-1)}$. The vehicle can estimate its pose at time k with respect to the global reference frame $\mathbf{0}$ by concatenating its intra-vehicle relative pose measurements:

$$\hat{M}_{1(k),\mathbf{0}} = \hat{M}_{1(0),\mathbf{0}}\hat{M}_{1(1),1(0)} \ldots \hat{M}_{1(k-1),1(k-2)}\hat{M}_{1(k),1(k-1)}. \tag{2}$$

This method of pose estimation is referred to as dead reckoning. Due to the large errors that typically affect intra-vehicle relative pose estimates, the error in the resulting global pose estimate from dead reckoning grows with k extremely fast [8], [9]. A Kalman filter is often used that fuses prediction of the vehicle's pose from some nominal motion model with the odometry measurements to estimate the pose. Though Kalman filtering is seen to reduce the growth rate of localization error, it does not eliminate it.

When multiple vehicles are present ($n > 1$) as shown in Figure 1, it may be possible for a vehicle to measure the relative pose between itself and a set of nearby vehicles, as described earlier. The relative pose of vehicle p with respect to vehicle r at time k is denoted by $M_{p(k),r(k)} \in SE(3)$, and its noisy measurement collected by vehicle r is denoted by $\hat{M}_{p(k),r(k)}$. We assume that each vehicle can recognize each of the other vehicles.

The situation when both inter-time and inter-vehicle relative pose measurements are available is best described in terms of a graph defined as a set of nodes and edges. The distinct frames attached to vehicle r till time k are denoted by $r(0), r(1), r(2), \ldots, r(k)$. Consider the set $\mathcal{V}(k)$, which consists of all the frames of the type $r(\ell)$ for $r = 1, \ldots, N$, and $\ell = 0, 1, \ldots, k$. Similarly, the set $\mathcal{E}(k)$ represents the relative pose measurements collected till k (both inter-time and inter-vehicle). More specifically, $\mathcal{E}(k)$ consists of ordered pairs of nodes of the following two types: (i) $(r(\ell-1), r(\ell))$ that corresponds to an inter-time measurement $\hat{M}_{r(\ell-1),r(\ell)}$ obtained by vehicle r at time ℓ ($\ell \leq k$), and (ii) $(r(\ell), p(\ell))$ that corresponds to a inter-vehicle relative pose measurement between the vehicles r and p at time ℓ ($\ell \leq k$). We call $\mathcal{G}(k) = (\mathcal{V}(k), \mathcal{E}(k))$ the *measurement graph* at time k.

In the example shown in Figure 1, vehicle 2 measures the pose of its neighbors 1 and 3 with respect to its local coordinate frame at time 0, but measures the relative pose of only vehicle 1 at time 1. The directed edge $(1(1), 2(1))$ represents the noisy measurement $\hat{M}_{1(1),2(1)}$ of the Euclidean transformation from the frame attached to vehicle 1's body at time 1 to the frame attached to vehicle 2's body at the same time.

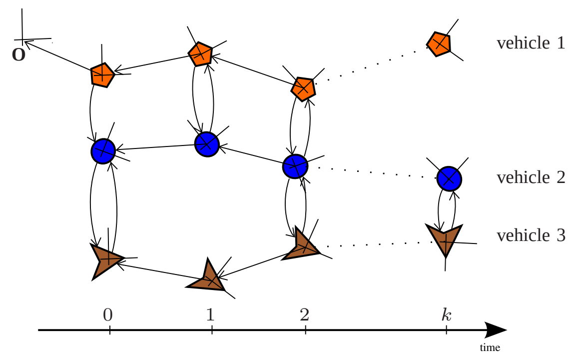


Fig. 1. A group of three vehicles moving to the right, with inter-time and inter-vehicle relative pose measurements. Vehicle 1 had a GPS measurement at initial time $k = 0$. Thereafter, no vehicle receives GPS measurements.

Note that there corresponds a unique *node variable* to every node $i(\ell) \in \mathcal{V}(k)$, namely, the absolute pose $M_{i(\ell),\mathbf{0}}$ of vehicle i at time ℓ (where $i = 1, \dots, n$, $\ell = 0, 1, \dots, k$). The problem is to estimate these node variable of $\mathcal{G}(k)$ as accurately as possible from the measurements corresponding to the edges $\mathcal{E}(k)$. In addition, the computation should be distributed, meaning that each vehicle should be able to gather the information it needs to compute its own pose estimate at time k by communicating with a limited number of neighboring vehicles.

Even without the constraint on distributed processing, the problem is challenging. Since Euclidean transformations (and rotation matrices) are not elements of a linear vector space, it is not clear how to "average" them to reduce noise and compute more accurate estimates. A restricted version of this problem that has been examined in the literature is the problem of averaging multiple measurements of an unknown rotation matrix R [10]. However, the available approaches to that problem are not applicable to the problem under study in this paper. Our problem data does not consist of measurements of the node variables to be estimated. Instead, the available measurements are noisy estimates of the transformations that relate one node variable to another.

III. A CENTRALIZED ALGORITHM FOR MULTI-VEHICLE LOCALIZATION

To avoid making the notation too cumbersome, we re-index all the nodes of the graph $\mathcal{G}(k)$ from 0 to N_k, where $N_k := |\mathcal{V}(k)| - 1$, where $|\mathcal{V}(k)|$ is the cardinality of the set $\mathcal{V}(k)$. The index 0 corresponds to the pose of the global reference, i.e., node **0**. Other than that, the manner of indexing is immaterial. The problem posed in the previous section can be restated as estimating the node variables M_i, where $M_i = M_{\mathbf{0}\,i}$, $i = 1, |\mathcal{V}(k)| - 1$ from the relative pose measurements $\hat{M}_{i\,j}$, for all $(i, j) \in \mathcal{E}(k)$. A solution to this problem, if found, can be used to estimate vehicle poses by applying the solution to the graph $\mathcal{G}(k)$ at every time k. In practice, such a batch processing approach may be computationally burdensome. However, the purpose in posing and solving this problem is to help us develop a distributed algorithm for vehicle pose estimation.

The algorithm we propose to solve this centralized pose estimation problem is inspired by an algorithm that Govindu proposed in [7] for computing the relative pose between a sequence of images taken by a camera. When there are common feature points between non-consecutive images in the sequence, redundant pose measurements become available between the first and the last frame of the image sequence than what is nominally required to estimate the pose between the first and the last image pair. The algorithm proposed in [7] works by converting the measurements into the Lie algebra $se(3)$ associated with the Lie group $SE(3)$, using a least squares approach in $se(3)$, and then converting the result back to $SE(3)$. We follow a similar approach here, but for the measurement graph $\mathcal{G}(k)$ instead of a sequence of images.

We denote elements of $SE(3)$ (or $SO(3)$) by M and elements of $se(3)$ (or $so(3)$) by m, which are related by $m = log(M)$ and $M = exp(m)$. Since $se(3)$ and $so(3)$ are non-commutative Lie groups, for $x, y \in se(3)$ (or $so(3)$), we have $e^x e^y = e^{BCH(x,y)}$ (*not* e^{x+y}, in general), where the mapping $BCH(\cdot, \cdot)$ is defined by the Baker-Campbell-Hausdorff formula: [11]

$$BCH(x, y) = x + y + \frac{1}{2}[x, y] + \frac{1}{12}[x - y, [x, y]] + O(|x, y|^4),$$

where $[x, y]$ is the Lie bracket:

$$[x, y] = xy - yx. \tag{3}$$

It follows from the BCH formula that we can *approximate* the matrix exponential product, when exponents are elements in $se(3)$, as:

$$e^x e^y = e^{BCH(x,y)} \approx e^{x+y} \tag{4}$$

Since $\hat{M}_{ij} \in SE(3)$ is a noisy measurement of M_{ij}, we write

$$\hat{M}_{ij} = M_j M_i^{-1} E_{ij} \tag{5}$$

where $E_{ij} \in SE(3)$ models the error between $\hat{M}_{ij}$ and M_{ij}. Taking logarithm and using the approximation (4) twice, we get

$$\hat{m}_{ij} \approx m_j - m_i + e_{ij}$$

where $m_{(\cdot)} = \log(M_{(\cdot)})$, which leads to

$$\hat{m}_{ij} = m_j - m_i + \delta_{ij}$$

where δ_{ij} captures the additional error between $m_j - m_i$ and $\hat{m}_{ij}$ not captured by e_{ij}. By using the vec- operation, the equation above can be written equivalently as

$$z_{ij} = v_j - v_i + \epsilon_{ij} \tag{6}$$

where $z_{ij} = vec(\hat{m}_{ij})$, $v_i = vec(m_i)$, and $\epsilon_{ij} = vec(\delta_{ij})$. Recall that $vec(X)$ for a matrix X returns a column of parameters extracted from the input matrix X. Now define $\mathbf{Z}$ as the vector of all z_{ij}'s stacked together, $\mathbf{V}$ as the vector of all v_i's stacked together ($i = 1, \ldots, N_k$), and $\boldsymbol{\epsilon}$ is the vector of all ϵ_{ij}'s stacked together. It is straightforward to show that

$$\hat{\mathbf{Z}} = (A_b \otimes I)^T \mathbf{V} + \boldsymbol{\epsilon} \tag{7}$$

where $\otimes$ denotes the Kronecker product, I is an identity matrix, and A_b is the *basis incidence matrix* of the graph $\mathcal{G}(k)$ obtained by taking the row of the incidence matrix out that corresponds to the reference node 1. The incidence matrix A of a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a matrix with as many rows as nodes and as many columns as edges that is defined as $A_{ij} = 1$ if $(i, j) \in \mathcal{E}$, $A_{ij} = -1$ if $(j, i) \in \mathcal{E}$, and $A_{ij} = 0$ otherwise.

The least squares estimate $\hat{\mathbf{V}}$ of $\mathbf{V}$ is given by

$$\hat{\mathbf{V}} = (\mathbf{D}^T \mathbf{D})^{-1} \mathbf{D}^T \mathbf{Z} \tag{8}$$

where $\mathbf{D} := (A_b \otimes I)^T$. Estimates of m_i's can now be recovered from the entries of $\hat{\mathbf{V}}$. Because of the approximation (4), these estimates need to be refined. To do so, the problem is changed slightly, into one where the variables to be estimated are the errors in pose measurements rather than the poses themselves. More specifically, since the relation $M_i^{-1} M_{ij} M_j = I$ will not hold when $M_{(\cdot)}$ is replaced by its noisy counterpart $\hat{M}_{(\cdot)}$ (estimates or measurements). We define $\Delta M_{ij} := \hat{M}_j^{-1} \hat{M}_{ij} \hat{M}_i$ and treat $\delta m_{ij} := \log(\Delta M_{ij})$ as a prediction error of M_{ij}'s for a given estimate of M_i and M_j. Starting from initial arbitrary estimates $\hat{M}_i$ of the global poses M_i's, the algorithm iteratively improves the estimates $\hat{M}_i$ using the least squares approach outlined above until the updated estimates of the global poses explain the relative pose measurements sufficiently well. The method is summarised in Algorithm 1.

Note that $\mathbf{D}$ should be full column rank so that the least squares estimate (8) is well defined. The column rank of $\mathbf{D}$ is the same as the row rank of A_b, where the basis incidence matrix of $\mathcal{G}$, which is full row rank if and only if $\mathcal{G}(k)$ is weakly connected [12]. The only situation in which $\mathcal{G}(k)$ is not weakly connected is when the vehicles can be divided into a number of groups so that there is no inter-vechicle relative pose meas between two vehicles belonging to distinct groups up to time k.

At each time k, if all the relative measurements are available to a central processor, Algorithm 1 can be used to estimate the poses $\hat{M}_i$, for $i = 1, \ldots, |\mathcal{V}(k)|$. Note that $\mathcal{V}(k)$ contains all the frames of the vehicles from time 0 to the current time k. The estimate of $\hat{M}_{r(\ell),\mathbf{0}}$, where $\ell < k$, uses measurements gathered after time ℓ. Therefore, the algorithm produces not only estimates of current poses of the vehicles but also the *smoothed* estimates of all the past poses of the vehicles. The price it pays for carrying out both estimation and smoothing is the high computational cost: the number of measurements that the algorithm has to process at time k can be as large as $O(N^2 k)$.

Algorithm 1: **IntrinsicPoseAvg:** Algorithm for pose estimation on a measurement graph $\mathcal{G}(k)$ at time k, based on Govindu's algorithm in [7].

Input: Measurements of relative poses $\hat{M}_{ij}$, $(i, j) \in \mathcal{E}(k)$, initial guesses for the node poses $\hat{M}_i, i \in \mathcal{V}(k)$, residual threshold ε.
Output: $\hat{M}_i$, $i \in \mathcal{V}(k)$ estimates of the node poses
Construct $\mathbf{D}$ in (8)
while $\|\Delta \hat{\mathbf{V}}\| > \varepsilon$ **do**
  foreach $\hat{M}_{ij} \in \{\hat{M}_{ij}\}_n$ **do**
    $\Delta \hat{M}_{ij} = \hat{M}_j^{-1} \hat{M}_{ij} \hat{M}_i$
    $\delta \hat{m}_{ij} = log(\Delta \hat{M}_{ij})$
    $\delta \hat{z}_{ij} = vec(\delta \hat{m}_{ij})$
  end
  $\Delta \hat{\mathbf{Z}} = [\delta \hat{z}_{ij1}^T, \delta \hat{z}_{ij2}^T, \ldots; \delta \hat{z}_{ij|\mathcal{E}_k|}^T]^T$
  $\Delta \hat{\mathbf{V}} = \mathbf{D}^{\dagger} \Delta \hat{\mathbf{Z}}$
  $[\delta \hat{v}_2^T, \delta \hat{v}_3^T, \ldots; \delta \hat{v}_{N_k}^T] = \Delta \hat{\mathbf{V}}$
  foreach $\hat{M}_i \in \{\hat{M}_i\}_{N_k}$ **do**
    $\hat{M}_i = \hat{M}_i exp(\delta \hat{v}_i)$
  end
end

We will show through simulations in Section V that the algorithm proposed above succeeds in giving more accurate global pose estimates than what would result if intervehicular measurements are not used. However, the algorithm comes with an number of disadvantages. First of all, all the measurements have to be communicated to a central processor, and once computed, the estimates of all vehicles' poses have to be communicated back to the vehicles. Due to unreliable nature of wireless communication and bandwidth limitations, this becomes infeasible even for moderately large group of vehicles. The memory requirement of the algoirthm also increases without bound as time k increases. In addition, centralized processing makes the system highly vulnerable to failure of the central processor.

For the reasons mentioned above, a distributed computation scheme is preferred. We call an algorithm distributed if every vehicle needs to communicate with only a small number of nearby vehicles (neighbors), where the number of neighbors is upper bounded by a constant that is independent of the total number of vehicles. In addition, we want the memory requirements for each vehicle's processor to be upper bounded by a constant that does not depend on time.

We propose such a distributed algorithm in the next section.

IV. A DISTRIBUTED ALGORITHM FOR COOPERATIVE GLOBAL POSE ESTIMATION

To describe distributed computation, first we describe a *communication graph* that specifies which vehicles can directly communicate information among one another. The node set of communication graph $\mathcal{G}_c(k) = (\mathcal{V}_c(k), \mathcal{E}_c(k))$ consists of all the n vehicles. However, To maintain consistency with notation used for the measurement graph, we write $\mathcal{V}_c = \{1(k), \dots, n(k)\}$. The time-varying edge set $\mathcal{E}_c(k)$ of the communication graph consists of the unordered pairs of nodes $(r(k), p(k))$ such that vehicles r and p can directly exchange information with each other at time k. In this paper we assume that if r can send data to p, p can also send data to r. The vehicles that r can communicate directly to at time k constitute the *communication neighbors* $\mathcal{N}_r(k)$ of r at k: $\mathcal{N}_r(k) = \{p \in \mathcal{V} | (r(k), p(k)) \in \mathcal{E}_c(k)\}$.

Notice that the measurement graph is directed whereas the communication graph is undirected. We assume that if vehicle r can measure its relative pose with respect to a vehicle p at time k, then r and p can also exchange information between them at that time. In other words, if either $(r(k), p(k)) \in \mathcal{E}(k)$ or $(p(k), r(k) \in \mathcal{E}(k))$, then $(r(k), p(k)) = (p(k), r(k)) \in \mathcal{E}_c(k)$, where $\mathcal{E}(k)$ is the set of edges of the measurement graph $\mathcal{G}(k)$. The reason is that for a vehicle r to measure p's relative pose, say with vision based sensors, the vehicles have to be close enough *and* r's on-board sensor has to have p in its field of view. On the other hand, for r and p to communicate, usually it is enough for the distance between them to be smaller than a communication radius. Therefore, conditions for successful communication are less stringent than the conditions for a inter-vehicle pose measurement. Note that the neighbor set $\mathcal{N}_r(k)$ is allowed to be empty. Figure 2 shows an example of a measurement graph and communication graph consisting of 4 vehicles at time $k = 2$.

At each time step every vehicle retains only (i) the estimated global pose for the previous time step and (ii) the local relative measurements with respect to its neighbors collected at the previous time step. It discards all measurements collected prior to that. With these, a *local measurement graph* $\mathcal{G}_r(k) = (\mathcal{V}_r(k), \mathcal{E}_r(k))$ for the vehicle r can now be defined, whose nodes corresponds to the frames of vehicle r and its neighbors at time k as well as the global reference frame $\mathbf{0}$. Specifically, $\mathcal{V}_r = \{\mathbf{0}, r(k)\} \cup \mathcal{N}_r(k)$. The edges of this graph corresponds to the relative poses whose estimates or measurements r has access to at k (except the ones it dropped). Specifically, $\mathcal{E}_r(k) = \{\hat{M}_{r(k-1),\mathbf{0}}, \hat{M}_{r(k),r(k-1)}\} \cup \{\hat{M}_{p(k),r(k)} | p(k) \in \mathcal{N}_r(k)\}$. Figure 3 shows an example of a local measurement graph.

Communication and processing of measurements collected at time k takes place in the interval between k and $k+1$. We denote this time interval by the symbol k^+. At the end

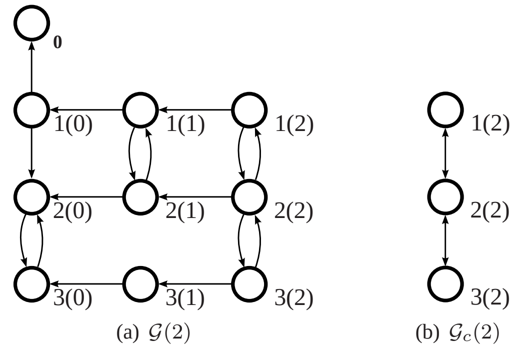


Fig. 2. The measurement graph and the communication graph at $k = 2$ for the multi-vehicle group shown in Figure 1.

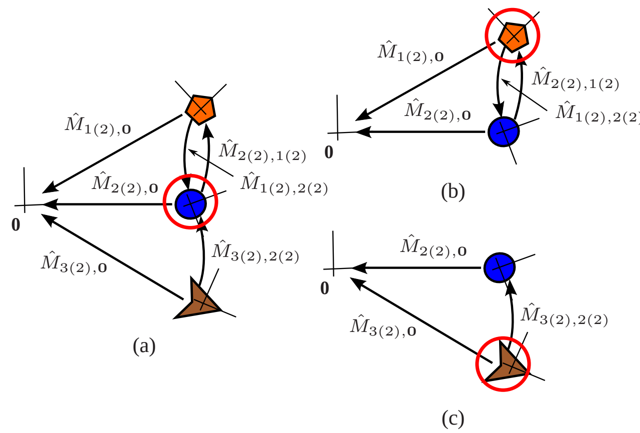


Fig. 3. The local measuremnt graphs for the vehicles shown in Figure 1: (a) $\mathcal{G}_2(2)$, (b) $\mathcal{G}_1(2)$, and (c) $\mathcal{G}_3(2)$.

time step k, each vehicle r broadcasts the estimate of its current global pose $\hat{M}_{r(k),\mathbf{0}}$ and the inter-vehicle relative pose measurements it obtained at k, i.e. $\{\hat{M}_{p(k),r(k)} | p \in \mathcal{N}_r(k)\}$. We assume that each vehicle completes broadcasting and receiving during $k+$, for each k. Therefore, at some time during k^+ after communication with its neighbors is over, vehicle r has an estimate of the poses of all the nodes in its local measurement graph $\mathcal{G}_r(k)$. It can now apply Algorithm 1 to the measurement graph $\mathcal{G}_r(k)$ to obtain an improved estimate of the poses of the nodes in $\mathcal{G}_r(k)$ than what it started with. The improvement comes from fusing the constraints on the poses that come from the inter-vehicle measurements with its neighbors. We assume that the computations by all are vehicles are completed during k^+.

At time $k+1$, vehicle r obtains the odometry measurement $\hat{M}_{r(k+1),r(k)}$ as well as a new set of relative pose measurements with its current neighbors: $\{\hat{M}_{p(k+1),r(k+1)} | p(k+1) \in \mathcal{N}_r(k+1)$. It fuses its current estimate of previous pose, $\hat{M}_{r(k),\mathbf{0}}$, with the odometry measurement $\hat{M}_{r(k+1),r(k)}$ to compute an initial estimate of its pose at $k+1$-th instant: $\hat{M}_{r(k+1),\mathbf{0}} = \hat{M}_{r(k+1),r(k)} \hat{M}_{r(k),\mathbf{0}}$. During the interval $(k+1)^+$, vehicle r repeats the communication and computation steps outlined above (but now for the new local measurement

Algorithm 2: **Distributed Algorithm for Multi-Vehicle Pose Estimation:** Each vehicle $r \in \{1, \ldots, n\}$ runs this algorithm at each time step k to estimate its current pose $M_{r(k),\mathbf{0}}$.

Input:
  $\hat{M}_{r(k-1),\mathbf{0}}$ (global pose estimate from previous time step)
  $\hat{M}_{r(k),r(k-1)}$ (relative pose measurement from k to k-1)
Output: $\hat{M}_{r(k),\mathbf{0}}$ (global pose estimate at current time step)
$\hat{M}_{r(k),\mathbf{0}} = \hat{M}_{r(k-1),\mathbf{0}} \hat{M}_{r(k),r(k-1)}$
$\overline{\mathcal{N}}_r(k) \leftarrow \{p(k)|\hat{M}_{p(k),r(k)}$ is measured by $r\}$
$\{\hat{M}_{i(k),j(k)}\} \leftarrow \{\hat{M}_{v(k),r(k)}| v \in \overline{\mathcal{N}}_r\}$
broadcast $\{\hat{M}_{i(k),j(k)}\} \cup \{\hat{M}_{r(k),\mathbf{0}}\}$
foreach *received transmission* $\hat{M}_{r(k),v(k)}$ **do**
  add $\hat{M}_{r(k),v(k)}$ to $\{\hat{M}_{i(k),j(k)}\}$
  add v to $\overline{\mathcal{N}}_r(k)$
end
foreach *received transmission* $\hat{M}_{v(k),\mathbf{0}}$ *s.t.* $v \in \overline{\mathcal{N}}_r(k)$ **do**
  add $\hat{M}_{v(k),\mathbf{0}}$ to $\{\hat{M}_{i(k),\mathbf{0}}\}$
end
$\{\hat{M}_i\} \leftarrow$ *reindexed* $\{\hat{M}_{i(k),\mathbf{0}}\} \cup \{\hat{M}_{r(k),\mathbf{0}}\}$
$\{\hat{M}_{ij}\} \leftarrow$ *reindexed* $\{\hat{M}_{i(k),j(k)}\}$
$\{\hat{M}_i\} \leftarrow$ IntrinsicPoseAvg$(\{\hat{M}_i\}, \{\hat{M}_{ij}\})$
$\hat{M}_{r(k),\mathbf{0}} \leftarrow \hat{M}_i \in \{\hat{M}_i\}$

graph $\mathcal{G}_r(k+1)$) to obtain an improved estimate of its current global pose $M_{r(k+1),\mathbf{0}}$. This process is repeated ad infinitum, as summarized in Algorithm 2. Note that the re-indexing the nodes corresponding to the global and local pose estimates into the form needed for Algorithm 1 is not specified since the manner in which they are indexed is immaterial.

With the distributed algorithm just described, the memory requirement for each vehicle r is proportional to the number of edges in its local graph $\mathcal{G}_r(k)$, which is upper bounded by a constant independent of time. An important aspect of Algorithm 2 is that the algorithm is robust to loss of communication between vehicles. If a vehicle cannot measure any neighbor's relative pose or cannot exchange data with any other vehicles, the distributed algorithm simply reduces to dead reckonging (see (2)).

V. Simulations

Simulations for vehicle teams with various configurations and trajectories were conducted to investigate the performance of the proposed algorithms. We conducted simulations for four case: (i) nominal motion of a group of vehicles in a straight line with fixed neighbor relationships over time and reliable communication, (ii) a variant of (i) with random communication failures, and (iii) a zig-zag motion of a group of vehicles that changes neighbor relationships over time. For each case, the bias and the variance of the errors in the estimated positions of the vehicles were empirically determined from Monte-Carlo simulations. The results for (ii) were found to be similar to those for (i), with a slight degradation of performance. Due to lack of space, we only present the results for cases (i) and (iii).

Noisy measurements of rotations between frames were generated as follows. Unit quaternions were generated from a Von Mises-Fisher distribution $f(x; \mu, \kappa)$ on the 3-sphere with concentration parameter $\kappa = 4000$ and $\mu = [0, 0, 0, 1]^T$, using the algorithm given in [13]. The Von Mises-Fisher distribution is a Gaussian-like distribution on hyper spheres that is symmetric about its mean and with concentration controlled by parameter k [14]. Those quaternions were then applied to the true rotations to generate the noisy measurements.

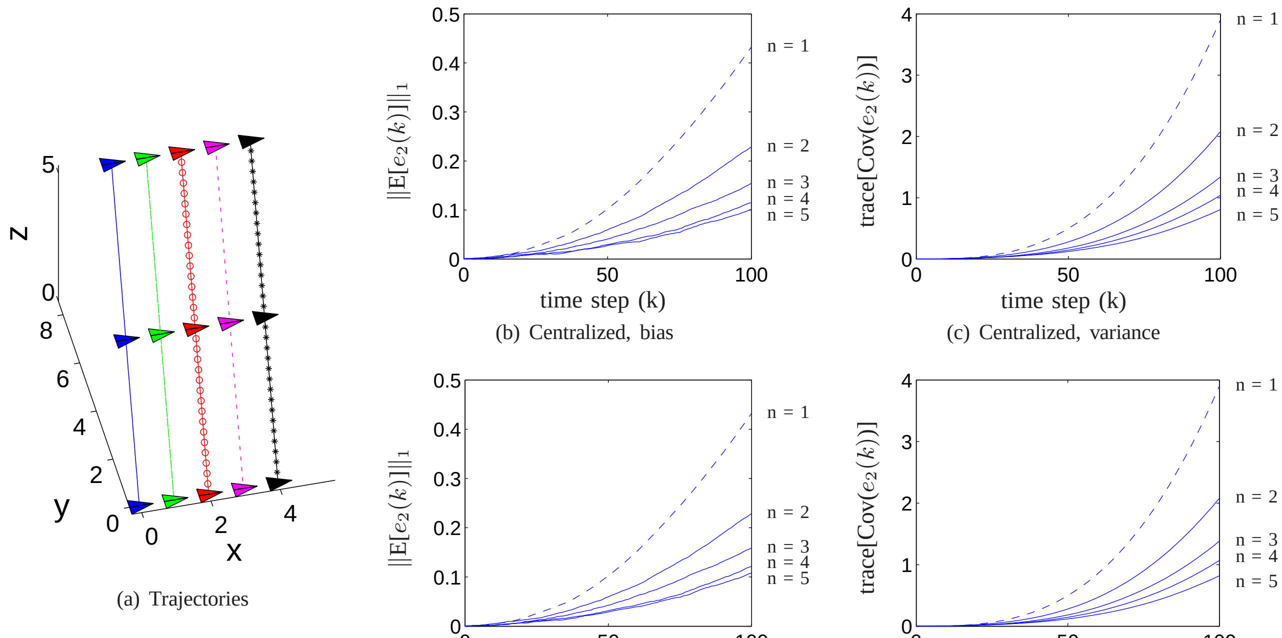
Case (i): Each vehicle in the group moves in a straight line through 3D Euclidean space and sees its two closest neighbors (or one if it is in the boundary of the group) at each time step. The position error of the r-th vehicle at time k is $e_r(k) := \hat{x}_r(k) - x_r(k)$, where $\hat{x}_r(k)$ is the estimated position, which is extracted from the estimated pose $\hat{M}_{r(k),\mathbf{0}}$. The bias and variance for the position error of vehicle 2 for this experiment is shown in Figure 4(b,c) for the centralized algorithm and Figure 4(d,e) for the distributed algorithm. The case $n = 1$ means vehicle 2 is the only vehicle in the team, and dead reckoning is used to estimate its position. Two conclusions are obvious from the figures:

1) collaborative estimation significantly improves location estimation accuracy compared to what is achieved by a single vehicle in self-location. The improvement in estimation accuracy is greater with increasing number of vehicles.
2) The centralized algorithm outperforms the distributed algorithm, but only by a small margin. In other words, the distributed algorithm performs almost as well as the centralized one. This is surprising, though certainly welcome.

Case(iii): Vehicles were now constrained to move along sinusoidal paths following a given vector through 3D Euclidean space. An example of the paths for each vehicle when $N = 6$ is given in Figure 5(a). In this experiment the neighbor relations were determined by distance. Each vehicle was able to see any vehicle whose distance was less then 5 units of measurement. The number of neighbors of vehicle 2 as a function of time is shown in Figure 5(b). The results for the Monte Carlo simulation of the distributed pose estimation algorithm are shown in Figure 5(c) and (d). The figure shows that even though the neighbor relationships were constantly changing, collaborative estimation using inter-vehicle relative pose measurements improve the pose estimation compared to that without collaboration.

VI. Summary and future work

In this paper we presented a distributed algorithm for collaborative localization of a multi-vehicle group in GPS denied scenarions when vehicle-to-vehile relative pose measurements may be available. By appropriately averaging the redundant information on the global poses of the vehicles



(a) Trajectories
(b) Centralized, bias
(c) Centralized, variance
(d) Distributed, bias
(e) Distributed, variance

Fig. 4. Improvement in the estimates of vehicles' location from collaborative estimation. The plots show the bias and variance of the error in a vehicle's location estimation, for both the centralized and distributed algorithms. Note that the case $n = 1$ (1 vehicle) means localization is performed without cooperation from other vehicles. The growth of estimation error (both bias and variance) with time slows down as the number of vehicles increases, showing the improvement brought about by averaging the inter-vehicle relative pose measurements.

provided by the inter-vehicle relative pose measurements, the algorithm reduces the error in the vehicle location estimates compared to what is achievable when vehicle estimate their own positions using only on-board sensors. In addition, the information needed by each vehicle can be obtained by communicating with only a limited number of neighboring vehicles.

In the current work, we have assume that the clocks of the vehicles are sufficiently synchornized so that that the inter-vehicle measurements can be accurately time-stamped. The sensitivity of the algorithm to inaccurate time synchronization remains to be investigated. Experimental testing of the algorithm using a network of ground vehicles is under way, and is expected to provide an answer to this sensitivity question.

In this work we do not assume any motion model of the vehicles. This makes the algorithm insensitive to discrepancy between actual vehicle model and that predicted by the model. At the same time, it is straightforward to incorporate a Kalman filter with an assumed motion model in each vehicle to generate the intra-vehicle relative pose measurements.

Providing any kind of analytical guarantees on the performance of both the algorithms (centralized and distributed), whether statistical or otherwise, is an open problem.

References

[1] E. Huber, "Object tracking with stereo vision," *CIRFFSS Conference on Intelligent Robotics in Field, Factory, Service and Space*, vol. 2, no. 1, pp. 763 – 767, 1994.

[2] D. Cruz, J. McClintock, B. Perteet, O. Orqueda, Y. Cao, and R. Fierro, "Decentralized cooperative control: A multivehicle platform for research in networked embedded systems," *IEEE Control Systems Magazine*, vol. 27, no. 3, pp. 58–78, June 2007.

[3] M. Hashimoto, Y. Matsui, and K. Takahashi, "Moving-object tracking with multi-laser range sensors for mobile robot navigation," *IEEE Conference on Robotics and Biomimetics*, pp. 399–404, 2007.

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

[5] X. Zhou and S. Roumeliotis, "Robot-to-robot relative pose estimation from range measurements," *IEEE Transactions on Robotics*, vol. 24, pp. 1379–1393, December 2008.

[6] R. Sharma and C. Taylor, "Cooperative navigation of MAVs in GPS denied areas," in *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, August 2008, pp. 481–486.

[7] V. Govindu, "Lie-algebraic averaging for globally consistent motion estimation," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1, June-2 July 2004, pp. I–684–I–691 Vol.1.

[8] J. Borenstein, "Internal correction of dead-reckoning errors with the smart encoder trailer," in *International Conference on Intelligent Robots and Systems*, September 1994, pp. 127–134.

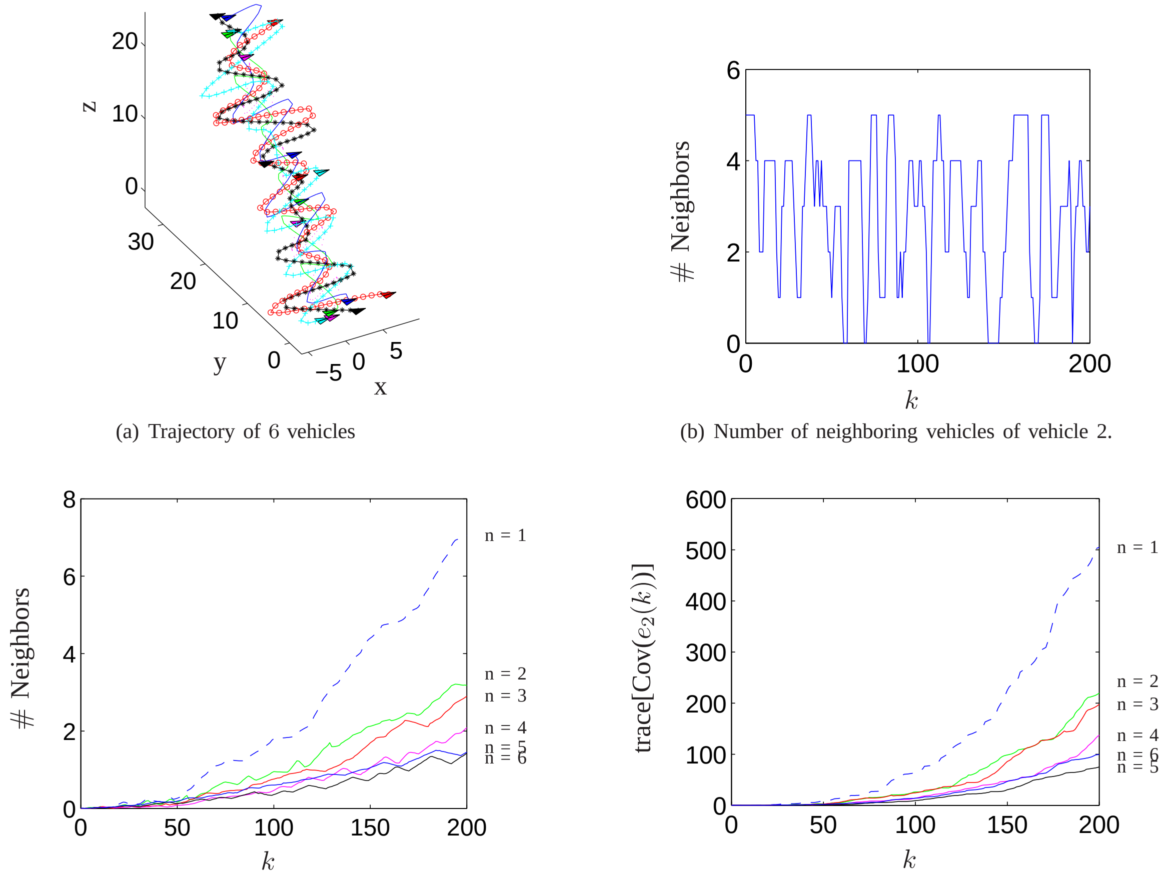
[9] C. F. Olson, L. H. Matthies, M. Schoppers, and M. W. Maimone, "Stero ego-motion improvement for robust rover navigation," in *IEEE International conference on Robotics and Automation*, March 2001.

[10] M. Moakher, "Means and averaging in the group of rotations," *SIAM Journal on Matrix Analysis and Applications*, vol. 24, no. 1, pp. 1–16, 2002. [Online]. Available: http://link.aip.org/link/?SML/24/1/1

[11] K. Engo, "On the BCH-formula in so(3)," *BIT Numerical Mathematics*, vol. 41, no. 3, pp. 629–632, June 2001.

[12] W.-K. Chen, *Applied Graph Theory*, H. A. Lauwerier and W. T. Koiter, Eds. North Holland Publishing Company, 1971.

[13] A. T. Wood, "Simulation of the von mises fisher distribution," *Communications in Statistics - Simulation and Computation*, vol. 23, pp. 157–165, 1994.



(a) Trajectory of 6 vehicles

(b) Number of neighboring vehicles of vehicle 2.

(c) Bias of estimation error

(d) Variance of estimation error

Fig. 5. Localization with distributed pose estimation Algorithm 2 for 6 vehicles moving in a zig-zag fashion. (a): vehicle trajectories in 3D, (b) the number of neighbors of vehicle 2 as a function of time, (c-d) bias and variance in the position estimates of vehicle 2.

[14] K. V. Mardia and P. E. Jupp, *Directional Statistics*, ser. Wiley series in probability and statistics. Wiley, 2000.