Aligning Coordinate Frames in Multi-Robot Systems with Relative Sensing Information

Sasanka Nagavalli, Student Member, IEEE, Andrew Lybarger, Lingzhi Luo, Student Member, IEEE, Nilanjan Chakraborty, Member, IEEE, Katia Sycara, Fellow, IEEE

Abstract—In this paper, we present both centralized and distributed algorithms for aligning coordinate frames in multi-robot systems based on inter-robot relative position measurements. Robot orientations are not measured, but are computed by our algorithms. Our algorithms are robust to measurement error and are useful in applications where a group of robots need to establish a common coordinate frame based on relative sensing information. The problem of establishing a common coordinate frame is formulated in a least squares error framework minimizing the total inconsistency of the measurements. We assume that robots that can sense each other can also communicate with each other. In this paper, our key contribution is a novel asynchronous distributed algorithm for multi-robot coordinate frame alignment that does not make any assumptions about the sensor noise model. After minimizing the least squares error (LSE) objective for coordinate frame alignment of two robots, we develop a novel algorithm that outperforms state-of-the-art centralized optimization algorithms for minimizing the LSE objective. Furthermore, we prove that for multi-robot systems (a) with redundant noiseless relative sensing information, we will achieve the globally optimal solution (this is non-trivial because the LSE objective is non-convex for our problem), (b) with noisy information but no redundant sensing (e.g., sensing graph has a tree topology), our algorithm will optimally minimize the LSE objective. We also present preliminary results of the real-world performance of our algorithm on TurtleBots equipped with Kinect sensors.

I. INTRODUCTION

In applications of multi-robot systems including search and rescue operations and environmental monitoring, robots need to communicate and/or interpret information in a common reference frame. For some application scenarios like mapping, robots need to maintain a consistent reference frame throughout the temporal extent of the operation to build a globally consistent map (for example, see [1], [2]). In other applications, robots only need to maintain a common reference frame at particular instances of time to exchange information. For example, consider a search and rescue operation where a group of robots enter a disaster zone and have to provide an assessment of the positions and number of victims in the zone. In this scenario, each robot on the team can operate without a common reference frame until one robot finds a victim. It is only at this point that a common reference frame must be temporarily established, so that the victim’s location can be communicated to the other robots. This periodic establishment of a common reference frame is sufficient for the robots to cooperatively identify the positions and number of victims found in the zone, while avoiding the overhead and difficulty of continuously maintaining their noisy sensor data in a common frame. Consider another scenario where robots are dropped in a GPS-denied environment. In such a situation, the robots begin without an obvious common reference frame and must establish one before starting the mission.

We refer to the problem of establishing a common reference frame at certain instances of time as the coordinate frame alignment problem, whereas the problem of maintaining a consistent reference frame over time is called the global localization or simply localization problem. In practice, estimation errors are usually much larger in global localization due to the accumulation of errors resulting from noisy sensor data, actuation errors and drift caused by interaction with the environment (e.g., wheel slip, autonomous underwater vehicle drift) over time. In contrast, for coordinate frame alignment, estimation errors are only due to sensory noise at a given time instant, resulting in significantly lower estimation error. Thus, in applications where the history of motion is not important for the interpretation of information among robots, one should use coordinate frame alignment algorithms. In this paper, we study the coordinate frame alignment problem.

Specifically, we consider a collection of robots where each robot can obtain noisy position measurements of other robots within its field of view in its own reference frame. We do not assume the knowledge of any sensor noise model, since it is usually difficult to get reliable sensor noise models. We assume that each robot can only communicate with robots it can sense. The coordinate frame alignment problem is to establish a common reference frame for all the robots using this relative sensing information. This problem is mathematically equivalent to the network localization problem in the context of static sensor networks [3], [4], [5].

In the literature, many versions of the localization problem have been studied for both mobile multi-robot systems and static sensor networks based on the sensing information available to the robots. The different sensing information considered in the literature are range (or inter-robot distance), bearing, position, orientation or any combination of these quantities. It has been established that the localization problem in the presence of noise is NP-hard in general [6], [3]. Distributed algorithms have been proposed in the literature for solving different versions of both the multi-robot localization problem [7], [8], [9], [10] and the sensor network localization problem [4], [5], [11]. In the context of the
problem studied in this paper, when only relative positions can be measured, existing distributed algorithms [5] that consider measurement error can get stuck in local minima even when the error is zero (i.e. their performance does not improve gracefully with increased accuracy of measurement). In contrast, algorithms [11], [12] that can establish a common reference frame in the absence of noise have not been extended to handle noise.

II. RELATED WORK

Localization problems have been studied both in the robotics literature, where the agents to be localized are assumed to be mobile, and the sensor networks literature where the agents to be localized are assumed to be static. As stated before, different problems have been formulated based on the types of sensor information available to the robot. Broadly speaking, the existing methods can be classified into two approaches: filtering based approaches (see [13], [14], [15] and references therein) and pose-graph based approaches (see [9], [7], [16] and references therein). The problem studied in this paper is mathematically equivalent to the localization problem in sensor networks with position measurements and therefore we discuss it in more detail here. A thorough discussion of the multi-robot localization literature can be found in [7], [10].

The work on sensor network localization can be organized according to the type of sensing information available to the sensor nodes, the dimension of the space (two-dimensional or three-dimensional) in which localization is performed and the presence or absence of environmental features (like beacons or anchors) that the sensor nodes can sense. In [3], the authors give a thorough foundation for the network localization problem where the sensors know the distance (or range) measurements to some other sensors. Other papers discuss algorithms for situations where the sensor nodes have angle of arrival measurements or bearing measurements to other sensor nodes [17], [18]. Piovan et al. consider the problem of localization with position information [4]. However, the proposed algorithm is not truly distributed in the sense that the communication is not between direct neighbor nodes only. In [5], the authors present a consensus-based solution to the problem of localizing a three-dimensional camera network using image data and assuming that there are common points in the environment that the cameras can see. In this paper, we consider a two-dimensional system analogous to [4] and design a distributed algorithm that requires communication between direct neighbors only. Our problem formulation is closely related to the problem formulation in [5]. We consider a planar scenario, whereas [5] considers a three-dimensional system. However, we do not assume that the robots can sense any environmental features. Each robot can only sense its neighbors’ relative positions in its own reference frame. Furthermore, in contrast to [5], we can guarantee that our algorithm gives the optimal estimate of the relative transformation between the robot frames when there is no measurement error.

Fig. 1: Coordinate frames and relative position vectors for a two-robot system. Frame $\mathcal{M}$ is the unknown common frame to be established. Frame $\mathcal{R}_i$ and frame $\mathcal{R}_j$ are the frames attached to robot $v_i$ and robot $v_j$ respectively.

Consensus-based methods have also been developed in [11], [12] for estimating the centroid of a multi-agent system in the absence of a common reference frame based on only relative position measurements between the agents. As the authors indicated in [11], this algorithm can also be used by the agents to agree on a common reference frame. The authors use a gossip-based asynchronous message-passing algorithm and it is not assumed that the agents have unique identifiers. However, their algorithm implicitly assumes perfect sensing information. In this paper, we consider a system where the robots have unique identifiers and our algorithm does not assume perfect sensing information, so it can be used with noisy measurements.

III. PROBLEM STATEMENT

Consider a multi-robot system where there is no common reference frame for all of the robots. Each robot is equipped with limited range communication devices and relative position sensors. We define a communication graph $\mathcal{G}_c = (\mathcal{V}, \mathcal{E}_c)$ and a sensing graph $\mathcal{G}_s = (\mathcal{V}, \mathcal{E}_s)$ where each node $v \in \mathcal{V}$ represents a robot. If robot $v_i \in \mathcal{V}$ can communicate with robot $v_j \in \mathcal{V}$, then edge $(v_i, v_j) \in \mathcal{E}_c$. If robot $v_i \in \mathcal{V}$ can sense the relative position of robot $v_j \in \mathcal{V}$, then edge $(v_i, v_j) \in \mathcal{E}_s$. In this paper, we assume that if robot $v_i$ can sense robot $v_j$, then robot $v_j$ can also sense robot $v_i$ (i.e. the sensing graph is undirected) and that if two robots can sense each other, they can also communicate (i.e. $\mathcal{G}_s = \mathcal{G}_c$). In addition, we assume that the sensing robot can identify the sensed robot. However, the robots do not have any model of their sensing error.

For each robot $v_i \in \mathcal{V}$, let $\mathcal{R}_i$ be the robot reference frame for measurements, which is rigidly attached to robot $v_i$. There is also a world reference frame $\mathcal{M}$, which is a common frame of reference for all of the robots, but initially unknown to any of them. The position of robot $v_i$ is defined as the displacement $r^i_\mathcal{M} \in \mathbb{R}^2$ between the origin of frame $\mathcal{R}_i$ and the origin of frame $\mathcal{M}$. The orientation of robot $v_i$ is defined as the rotation matrix $\Omega_i \in \text{SO}(2)$ that transforms a vector in $\mathcal{R}_i$ to a vector in $\mathcal{M}$. Note that $\mathbb{R}^2$ is the two-dimensional Euclidean space and $\text{SO}(2)$ is the special orthogonal group of dimension 2. The goal is to establish the common reference frame $\mathcal{M}$, using only relative position measurements and communication between
Orientation of the robots is not measured, but computed as part of the process.

Each robot \( v_i \) has an initial estimate of its position \( \hat{r}_i^M \) and orientation \( \hat{O}_i \) in the unknown common reference frame \( M \). If no estimate is available, these two variables can be initialized to \( \hat{r}_i^M = 0 \) and \( \hat{O}_i = I_2 \) where \( I_2 \) is the \( 2 \times 2 \) identity matrix. If \( (v_i, v_j) \in E \), then robot \( v_i \) can measure the relative position \( r_{ij}^M \) of robot \( v_j \) in frame \( R_i \). Let \( \hat{r}_{ij}^M \) be the true relative position of robot \( v_j \) in frame \( R_i \) and \( \hat{r}_{ij} \) be the measurement with error \( \epsilon_{ij} \). (i.e. \( \hat{r}_{ij}^M = r_{ij}^M + \epsilon_{ij} \)).

The measurement and the error expressed in frame \( M \) is

\[
\hat{r}_{ij}^M = O_i \hat{r}_{ij}^R, \quad \epsilon_{ij} = O_i \epsilon_{ij} \tag{1}
\]

The magnitude of the measurement error is given by \( \| \epsilon_{ij} \| = \| O_i \hat{r}_{ij}^R - r_{ij}^M \| \). The total measurement error is

\[
\sum_{(v_i,v_j) \in E} \| \epsilon_{ij} \| = \sum_{(v_i,v_j) \in E} \| O_i \hat{r}_{ij}^R - r_{ij}^M \| \tag{2}
\]

The objective function to minimize the total measurement error in the unknown common frame \( M \) is

\[
\min \sum_{(v_i,v_j) \in E} \left\| O_i \hat{r}_{ij}^R - (\hat{r}_j^M - r_i^M) \right\|^2 \tag{3}
\]

where \( \hat{O}_i \in SO(2) \) and \( \hat{r}_i^M \in R^2 \) are the optimization variables. Notice that when there is no measurement error, the objective function is minimized when the estimates are equal to the true values (i.e. \( \hat{O}_i = O_i, \hat{r}_i^M = r_i^M \)). The objective function is non-convex.

IV. ALGORITHMS FOR NOISELESS MEASUREMENTS

Before solving the more general case with sensing noise, we look at the Problem (3) assuming that there is no sensing noise. Even in this case, existing algorithms [5] can get stuck in local minima. Our goal in this section is to present algorithms that guarantee that the global minimum of the problem in Equation (3) is achieved in the absence of noise. In Section V, we will show that the algorithms for the noiseless case are useful for solving the problem with noise.

Let us first consider a system consisting of only two robots. By assumption, if robot \( v_i \in V \) can measure the position of robot \( v_j \in V \) in frame \( R_i \), then robot \( v_j \) can measure the position of robot \( v_j \) in frame \( R_j \). The two robots can align their coordinate frames as follows:

1. Robot \( v_i \) communicates its estimate of its own position \( \hat{r}_i^M \) and the measurement \( \hat{r}_{ij}^M = O_i \hat{r}_{ij}^R \) to robot \( v_j \).
2. Robot \( v_j \) finds rotation \( R \in SO(2) : \hat{r}_{ij}^M = -R \hat{r}_i^M \) and updates its orientation \( O_j \) to \( R \hat{O}_i \).
3. Robot \( v_j \) updates its position \( \hat{r}_j^M = \hat{r}_i^M + \hat{r}_{ij}^M \).

We use this two robot alignment procedure described above as a basic building block for algorithms to align the coordinate frames of all robots in the system. Assume that all robots in the system have a unique identifier (UID) with \( i = UID(v_i) \) for all robots \( v_i \in V \).

Algorithm 1 Centralized Alignment of Coordinate Frames

1. procedure \text{ALIGNCOORDINATEFRAMES}(G = (V, E))
2. for all \( v_i \in V \) do
3. \( \hat{O}_i \leftarrow I_2, \hat{r}_i^M \leftarrow 0 \)
4. for all \( v_j : (v_i, v_j) \in E \) do
5. \( \hat{r}_{ij}^M \leftarrow \text{GETMEASUREMENT}() \)
6. end for
7. end for
8. \( Q \leftarrow \emptyset, A \leftarrow \emptyset \)
9. \( l \leftarrow \min \{ i : \forall v_i \in V \} \)
10. \( A \leftarrow A \cup \{ l \} \)
11. \( \text{ENQUEUE}(Q, l) \)
12. while \( Q \neq \emptyset \) do
13. \( m \leftarrow \text{DEQUEUE}(Q) \)
14. for all \( s : ((v_m, v_s) \in E) \wedge (s \notin A) \) do
15. \( R \leftarrow J \in SO(2) : \hat{O}_m \hat{r}_{ms}^M = -J \hat{O}_s \hat{r}_{sm}^M \)
16. \( \hat{O}_s \leftarrow R \hat{O}_s \)
17. \( \hat{r}_s^M \leftarrow \hat{r}_m^M + \hat{O}_m \hat{r}_{ms}^M \)
18. \( A \leftarrow A \cup \{ s \} \)
19. \( \text{ENQUEUE}(Q, s) \)
20. end for
21. end while
22. end procedure

A. Centralized Algorithm

The algorithm may be summarized as a BFS starting at the leader node, where every visited node aligns its local frame with the local frame of the first neighbour from which it is visited. One can envision that the alignment of frames of nodes in the sensing graph occurs in the same order in which nodes would be added to a spanning tree rooted at the leader created via BFS through the graph. The computational complexity of the algorithm is the same as BFS, with a worst case time complexity of \( O(|V| + |E|) \) [19] and a worst case space complexity of \( O(|V|) \). This is the worst case space complexity for maintaining the queue, which assumes that the input graph \( G \) and measurements \( \hat{r}_{ij}^M : \forall (v_i, v_j) \in E \) are already known and have separate storage allocated for them.

B. Distributed Algorithm

Since the requirement for a central computer with knowledge of the topology of the entire sensing graph is restrictive, a distributed algorithm is often preferable. Call the robot with the lowest UID the "leader". If every robot in the system, except the leader, can sense and communicate with at least one robot with a lower UID, then the coordinate frames of all robots can be aligned by the following procedure. Every robot \( v_i \in V \) communicates the position estimate \( \hat{r}_i^M \) and relative measurement \( \hat{r}_{ij}^M \) to all neighbours \( v_j \in V \) where \( UID(v_j) < UID(v_i) \). Every robot \( v_j \) then aligns its frame with the neighbour with the lowest UID. If the sensing graph is connected, when this procedure is applied iteratively. In a system with \( N \) robots, the algorithm will result in the coordinate frames of all robots being aligned with that of the leader within at most \( N - 1 \) iterations.
Fig. 2: Simple Case that is Not Alignable Using UIDs Alone

As shown in Figure 2, there are many cases where robots other than the leader are not connected to any robots with lower UIDs. Based on the procedure above, robot \( v_2 \) will not align its frame with robot 3. The incorporation of hop IDs (HID) can solve this problem. Starting from the leader, perform a BFS through the graph and assign every node an HID. HIDs are not unique. An HID represents the minimum number of hops from that node to the leader. The leader has an HID of 0. In this way, a spanning tree for the graph rooted at the leader can be implicitly established and every node except the leader is connected to at least one other node with a lower HID. Every robot \( v_i \in \mathcal{V} \) communicates the position estimate \( \tilde{r}_M \) and relative measurement \( \tilde{r}_{ij} \) to all neighbours \( v_j \in \mathcal{V} \) where HID \((v_i) < \text{HID}(v_j)\). Every robot \( v_j \) then aligns its frame with its lowest HID neighbour. If multiple neighbours have the same HID, then of those neighbours it picks the one with the lowest UID.

Algorithm 2 Distributed Alignment of Coordinate Frames

```plaintext
1: procedure ALIGNCOORDINATEFRAME\((u, \mathcal{N})\)
2: \( \hat{O}_u \leftarrow I_2, \hat{r}_M \leftarrow 0 \)
3: \( I \leftarrow u, h \leftarrow 0, m \leftarrow \text{NIL} \)
4: for all \( n \in \mathcal{N} \) do
5: \( \hat{r}_M \leftarrow \text{GETMEASUREMENT()} \)
6: \( \hat{r}_{un} \leftarrow \hat{O}_u \hat{r}_{un} \)
7: \( \text{SENDMSG}(n, u, h, l, \hat{r}_u, \hat{r}_M) \)
8: end for
9: while \( \{n, h', l', \hat{r}_M, \hat{r}_{un}\} \leftarrow \text{RECVMSG()} \) do
10: if \((l > l') \lor (l = l' \land h > h' + 1)\) then
11: if \((l > l') \lor (l = l' \land h > h' + 1)\) then
12: \( I \leftarrow l', h \leftarrow h' + 1, m \leftarrow \text{NIL} \)
13: \( \text{end if} \)
14: if \((m = \text{NIL}) \lor (m \geq n)\) then
15: \( R \leftarrow J \in SO(2): \hat{r}_M = -JR_M \)
16: \( \hat{O}_u \leftarrow R \hat{O}_u \)
17: \( \hat{r}_M \leftarrow \hat{r}_M + \hat{r}_M \)
18: \( m \leftarrow n \)
19: for all \( i \in \mathcal{N} \) do
20: \( \hat{r}_{ui} = \hat{O}_u \hat{r}_{ui} \)
21: \( \text{SENDMSG}(i, u, h, l, \hat{r}_u, \hat{r}_M) \)
22: end for
23: \( \text{end if} \)
24: \( \text{end if} \)
25: \( \text{end while} \)
26: end procedure
```

Algorithm 2 is a distributed algorithm that uses the above ideas to align the local frames of all robots in a system. Every robot in the system executes the algorithm and initially only has knowledge of its own UID and the UIDs of its direct neighbours. No robot in the system ever requires or acquires any more knowledge of the graph topology.

The algorithm takes the UID \( u \) of the robot executing the algorithm and the set \( \mathcal{N} \) of UIDs of the robot’s direct neighbours as input. On line 2, the robot initializes its own pose. This line is optional if the robot has a different initial estimate of its pose. On line 3, the robot initializes its belief of the leader’s UID \( l \), its belief of its own HID \( h \) and the UID \( m \in \mathcal{N} \) of the neighbour with which it is currently aligned (i.e. its master). Initially, every robot assumes the leader is itself, which means its current HID is 0 and it has no master. On lines 4–8, the robot \( v_u \) measures the relative position \( \tilde{r}_{un} \) of each neighbour robot \( v_n : n \in \mathcal{N} \) in frame \( \mathcal{R}_u \). It then uses an estimate of its orientation to express the measurement in frame \( \mathcal{M} \) and uses SendMsg\((p_1, p_2, p_3, p_4, p_5, p_6)\) to send a message to the neighbour whose position it measured. The message contains the following items: \( p_1 \) is the destination robot’s UID, \( p_2 \) is the source robot’s UID, \( p_3 \) is the source robot’s current HID, \( p_4 \) is the source robot’s current leader’s UID, \( p_5 \) is the source robot’s position in its own local frame, \( p_6 \) is the destination robot’s relative position in the source robot’s local frame. RecvMsg\(()\) unpacks a message sent from a neighbour via SendMsg\((p_1, p_2, p_3, p_4, p_5, p_6)\). Only the last 5 parameters are returned. On lines 9–25, the robot receives and processes messages from its neighbours. On lines 11–13, the robot updates its belief of its leader’s UID, its own HID and its master’s UID if it discovers that either its neighbour’s leader’s UID \( l’ \) is lower than its leader’s UID \( l \) or its own HID is incorrect. On lines 15–17, the robot \( v_u \) aligns its frame with its neighbour \( v_n \) only if \( v_u \) has the same leader, a lower HID than \( v_n \) and the UID of \( v_n \) is the same or lower than the UID of the current master of \( v_u \). If aligned with \( v_n \), on line 18, \( v_u \) sets the UID of its master to the UID of \( v_n \). On lines 19–22, the robot uses SendMsg\((p_1, p_2, p_3, p_4, p_5, p_6)\) to send messages to its neighbours containing their relative positions in its updated local frame.

Notice that no more messages are sent after every robot (1) agrees on the same leader, (2) acquires the correct HID and (3) aligns its frame with the neighbour with lowest HID and UID in respective order. Since no more messages will be sent, the algorithm will terminate implicitly. Any robot can initiate the algorithm by sending out the first message, but no robot is ever explicitly aware when the algorithm has terminated. However, when the algorithm does terminate, if the sensing graph is connected, the coordinate frames of all robots in the system will be aligned. The algorithm has a space complexity of \( O(1) \) because every robot only needs to store five persistent values \((l, h, m, \hat{O}_u, \hat{r}_M)\) and some temporary storage for received messages. Since no robot will ever need to realign itself more times than there exist edges in the sensing graph, the algorithm has a worst case time complexity of \( O(|\mathcal{E}|) \).

Assuming that every robot has a queue to store incoming messages for retrieval and processing via RecvMsg\(()\), the algorithm has the benefit of being fully asynchronous. Messages can be sent and received at any time. The only requirement is messages with both the same source and the same destination must be processed in order. For example, if robot \( v_i \) sends message \( m_1 \) and message \( m_2 \) to robot \( v_j \) in
that order, then robot \( v_j \) must process message \( m_1 \) first and message \( m_2 \) second. However, if robot \( v_i \) sends message \( m_1 \) to robot \( v_k \) and robot \( v_j \) sends message \( m_2 \) to robot \( v_k \), then robot \( v_k \) can process \( m_1 \) and \( m_2 \) in any order.

V. ALGORITHMS FOR NOISY MEASUREMENTS

The basic algorithms in the previous section can be modified to handle noisy measurements. Note that the measurement error in position is composed of two components: distance measurement error and bearing measurement error. Consider a system consisting of two robots, \( v_i \) and \( v_j \). Assume that UID \( (v_j) < \) UID \( (v_i) \). In this case, the objective function to be minimized is

\[
\left\| \hat{O}_i \tilde{F}_{ij} - (\tilde{F}_i^M - \tilde{F}_j^M) \right\|^2 + \left\| \hat{O}_j \tilde{F}_{ji} - (\tilde{F}_i^M - \tilde{F}_j^M) \right\|^2
\]

As before, if the robots localize themselves with respect to the robot with the lowest UID, then \( \hat{O}_j = I_2 \) and \( \tilde{F}_j^M = 0 \). Therefore, the objective function reduces to

\[
\min \left( \left\| \hat{O}_i \tilde{F}_{ij} + \tilde{F}_i^M \right\|^2 + \left\| \tilde{F}_{ji} - \tilde{F}_j^M \right\|^2 \right)
\]

(4)

**Lemma 1.** For any two non-zero constant vectors \( \tilde{y}, \tilde{z} \in \mathbb{R}^2 \), the objective function

\[
\min \left( \left\| \tilde{y} - \tilde{x} \right\|^2 + \left\| \tilde{z} + \tilde{x} \right\|^2 \right)
\]

is minimized by \( \tilde{R} \in SO(2) \) and \( \tilde{x} \in \mathbb{R}^2 \) such that

\[
\tilde{R} \tilde{z} = -\frac{\| \tilde{y} \|^2}{2\| \tilde{y} \|} \tilde{y}, \quad \tilde{x} = \frac{\| \tilde{y} + \tilde{R} \tilde{z} \|}{2}\tilde{y}
\]

(6)

As a result of Lemma 1 (proof in Appendix), the optimal solution to the objective function in Equation (4) is

\[
\hat{O}_i \tilde{F}_{ij} = -\frac{\tilde{F}_{ij}^T}{\| \tilde{F}_{ij} \|^2} \hat{O}_i \tilde{F}_{ij}^T, \quad \tilde{F}_i^M = \tilde{F}_i^M + \frac{\| \tilde{F}_{ij} \|^2}{\| \tilde{F}_{ij} \|^2} \tilde{F}_{ij}
\]

(7)

where \( \hat{O}_i \in SO(2) \) and \( \tilde{F}_i^M \in \mathbb{R}^2 \). The objective function’s value at the optimal solution is \( \frac{1}{2} \left( \| \tilde{F}_{ij} \|^2 - \| \tilde{F}_{ij}^M \|^2 \right) \). Note that this error is zero whenever the magnitude of the two measurements is equal (i.e. \( \| \tilde{F}_{ij} \| = \| \tilde{F}_{ij}^M \| \)), which means we can only minimize the effects of distance measurement error and not bearing measurement error.

A. Tree Topologies

When the graph structure is a tree, Theorem 1 implies that Algorithm 1 and Algorithm 2 will compute the optimal pose estimates in frame \( M \) for every robot. Lines 15–17 of Algorithm 1 and lines 15–17 of Algorithm 2 must be replaced to perform the computation in Equation (7) to minimize error pairwise between each robot and its neighbour.

**Theorem 1.** If the undirected sensing graph \( G = (V, E) \) is a tree, then the coordinate frame alignment objective function in Equation (3) has a globally minimal solution

\[
\hat{O}_s \in SO(2) : \hat{O}_s \tilde{F}_{sm} = \frac{-\tilde{F}_{sm}}{\| \tilde{F}_{sm} \|^2} \hat{O}_m \tilde{F}_{ms}
\]

\[
\hat{F}_s^M \in \mathbb{R}^2 : \hat{F}_s^M = \tilde{F}_m^M + \frac{\| \tilde{F}_{sm} \|^2}{\| \tilde{F}_{sm} \|^2} \hat{O}_m \tilde{F}_{ms}
\]

(8)

for all \( v_m \in V \) and \( v_s \in V \) such that \((v_m, v_s) \in E\).

B. General Graph Topologies

**Corollary 1.** In any general sensing graph \( G = (V, E) \), both Algorithm 1 and Algorithm 2 will globally minimize the objective function in Equation (3) when there is no measurement error.

1) Centralized Error Refinement: We will now present a generalization of Algorithm 1 that can reduce the effects of measurement error in sensing graphs that are not necessarily trees. First, we apply Algorithm 1 as presented in Section IV-A. Second, we apply Newton’s method to optimize the objective function in Equation (3) using the estimates obtained from applying Algorithm 1 as an initial guess. This will cause Newton’s method to converge to the same or more optimal local minimum than directly applying Newton’s method, as verified experimentally by simulation results in Figure 3 (compare Figure 3a and Figure 3c).

2) Distributed Error Refinement: In the distributed scenario, to generalize Algorithm 2 to handle measurement error, we will use the same intuition as in the centralized case. First, we need to define an objective function for each robot based on the objective function in Equation (3). For a robot \( v_i \in V \), let \( \mathcal{N} (i) \) be the set of the robot’s neighbours’ UIDs, and

\[
\mathcal{L} (i) = \{ A \subseteq \mathcal{N} (i) \mid j \in A : \text{LID} (v_j) = \text{LID} (v_i) \}
\]

be the set of UIDs of the robot’s neighbours that believe in the same leader. The objective function \( v_i \) must minimize is

\[
\sum_{j \in \mathcal{L} (i)} \left( \left\| \hat{O}_i \tilde{F}_{ij} - \hat{F}_j^M + \hat{F}_i^M \right\|^2 + \left\| \hat{F}_{ji} + \hat{F}_j^M - \hat{F}_i^M \right\|^2 \right)
\]

(9)

where \( \hat{O}_i \) and \( \hat{F}_i^M \) are the optimization variables and everything else is constant. Note that robot \( v_i \)’s individual objective function in Equation (9) is derived from all the terms related to the variables \( \hat{F}_i^M \), \( \hat{O}_i \) in the objective function in Equation (3). When robot \( v_i \) tries to optimize its objective function, it assumes that the state of each neighbour \( v_j : j \in \mathcal{L} (i) \) is fixed at the most recently received values. Algorithm 2 should be modified as follows:

1) When a robot discovers it should realign with its master, it should apply the computation in Equation (8).

2) For any message received from its neighbours that does not cause it to realign with its master, each robot should minimize its individual objective function in Equation (9) using Newton’s method.

The robot should then send its updated position and measurements expressed in frame \( M \) to its neighbours, repeating this procedure until the change in values obtained from the optimization is below a predefined threshold. True to the nature of Algorithm 2, this procedure refines the pose estimates of each robot in a distributed manner with each robot only using local information and communication. As shown in Figure 3, the performance of this distributed refinement procedure was verified in simulation to approximately match the performance of centralized refinement after
centralized coordinate frame alignment (compare Figure 3c with Figure 3d).

The distributed error refinement procedure presented above interleaves error refinement with alignment with a master. In the distributed setting, this interleaving is necessary because no robot is ever explicitly aware when Algorithm 2 has converged. Note that this interleaved method will produce exactly the same result as if Algorithm 2 is first executed until convergence and then the distributed error refinement is applied. This is due to the fact that any error refinements performed in step 2 are discarded every time step 1 is performed. Step 1 is only performed until all robots in the system agree upon the leader and each robot has identified its master. After this point, only step 2 is performed.

In the absence of measurement error, it is important to note that the additional error refinement procedures do not affect the ability of Algorithm 1 or Algorithm 2 to globally minimize the objective function in Equation (3). This is due to the fact that the alignment procedure is performed before the error refinement and from Corollary 1, the algorithms in Section IV will generate estimates that globally minimize the objective function. The following error refinement via Newton’s method does not change the estimates when they are already optimal.

Finally, note that our algorithms only use one set of measurements. Every robot $v_i$ has one measurement $\tilde{r}_{ij}$ for each of its neighbors $v_j$. When there are multiple measurements available (e.g. multiple sensors, multiple measurements over time) for the same vector $\tilde{r}_{ij}$, the application of our algorithms does not preclude the pre-processing of measurements using filtering techniques (e.g. Kalman filtering) to improve the quality of the relative position measurements.

VI. RESULTS

A. Simulation Results

Algorithm 1, Algorithm 2 and the generalizations presented in Section V were implemented in MATLAB and their performance was compared. As a benchmark, several conventional centralized optimization techniques (Newton’s method, Levenberg-Marquardt, trust region methods) were applied to the objective function in Equation (3). For our problem, Newton’s method had the best performance. Figure 3 shows the results for 1000 trials with a group of 10 robots with random initial poses and where the system’s sensing graph is a single connected component with a general graph topology and a random symmetric adjacency matrix. Inter-robot distance and bearing measurements were sampled from Gaussian distributions where the true value was the mean (i.e. no bias) and the standard deviations were 20% of the true value for distance and $\frac{\pi}{18}$ radians for bearing. This is a very large amount of error (i.e. what one would expect from a system with very poor sensors). Figure 3a shows the results of applying Newton’s method to the problem directly. Figure 3b shows the results of applying Algorithm 2 (without error refinement). As shown in Figure 3c and Figure 3d, the objective function was best minimized by first applying coordinate frame alignment followed by the error refinement process described in Section V.
Bearing Measurement Error
Distance Measurement Error
Distance Estimation Error
(System)
Orientation Estimation Error
(System)

In this paper, we assumed that the sensing and communication graphs of the multi-robot system are identical. In preliminary experimental results were reported.

The measurement and estimation errors for the mixed formation are shown in Figure 6. The results for the ordered formation are similar. As shown in Figure 6a and Figure 6b, the histograms for measurement error are not centered around zero, indicating a bias in the measurements. As shown in the Figure 6c and Figure 6d, the bias in measurements results in a bias in estimates. In the test setup, the robots are 1 meter away from each of their sensing graph neighbours, so a distance estimation error of 0.03 meters (mode of histogram) indicates the estimates are accurate to ±3%. The estimation error histograms for individual robots are shown in Figure 6e and Figure 6f. As a result of the bias in measurements, in Figure 6e, robot 4 appears to have a more accurate estimate of its position than robot 2, despite being more hops away from robot 1 in the sensing graph. The variance of the estimation error distributions is a better indicator of error propagation in the frame alignment algorithm. For example, in Figure 6f, robots 3 and 5 are both one hop away from robot 1 in the sensing graph and have similar variance for their error distributions, while robot 4 is more hops away from robot 1 in the sensing graph and has a wider, more irregular error distribution.

B. Experimental Results

To validate our algorithms under conditions with realistic, possibly non-Gaussian measurement error, experiments were conducted using TurtleBot robotic platforms running ROS (Robot Operating System). Each robot was labelled with a QR code containing its UID. Using image and depth data from a Kinect sensor, each robot identified its neighbours and measured their relative position in its reference frame. The performance of the distributed algorithm (i.e. Algorithm 2) was tested on the two formations shown in Figure 5. In both formations, all robots aligned their reference frames with that of the robot with the lowest UID (i.e. robot 1).

In the ordered formation, 1777 trials were conducted with each trial consisting of a set of relative position measurements. In the mixed formation, 776 trials were conducted. For each trial, the robots used Algorithm 2 to align their coordinate frames, which is optimal because the graph has a tree topology. The ground truth for the test setup was established through manual measurement and the test setup was not changed across trials for each test formation. For each measurement (i.e. two measurements per undirected edge in the sensing graph) in each trial, the distance measurement error \( \delta_{ij} \) and bearing measurement error \( \alpha_{ij} \) were computed. Similarly, for each robot \( v_i \) the distance estimation error \( \Delta_i \) and orientation estimation error \( \beta_i \) were computed. Note that every 2-dimensional rotation matrix \( R_i(\theta) \) can be represented using a single angle \( \theta \) as a parameter.

\[
\delta_{ij} = \left\| \hat{r}_{ij} - \tilde{r}_{ij} \right\|, \quad \hat{r}_{ij} = R_i(\alpha_{ij}) \tilde{r}_{ij}
\]

\[
\Delta_i = \left\| \hat{r}_i^M - \tilde{r}_i^M \right\|, \quad \hat{O}_i = R_i(\beta_i) \tilde{O}_i
\]

VII. Conclusion and Future Work

We designed algorithms for coordinate frame alignment in robotic teams whose members have noisy, range-limited relative position sensing and communication with no model of sensing error. In coordinate frame alignment, unlike global localization, a common reference frame is established at a particular instance in time, so only sensory noise at that instant needs to be considered (and not accumulated errors over time). Thus, for applications where a common reference frame is only necessary at some instances in time, coordinate frame alignment would be more accurate and relevant than global localization. We first presented centralized and distributed algorithms for coordinate frame alignment assuming no noise and proved that our algorithms will achieve a globally optimal solution. In the distributed setting, each robot updates its estimate of its pose within the unknown common reference frame by reconciling its measurements of neighbouring robots with their measurements of it. We presented modifications of our algorithms to handle measurement noise. In the distributed setting, each robot minimizes the terms of the overall objective function whose variables it controls. Simulation results demonstrated our distributed algorithm performs better than centralized Newton’s method. Finally, our algorithms were implemented on TurtleBots and preliminary experimental results were reported.

In this paper, we assumed that the sensing and communication graphs of the multi-robot system are identical. In
future work, we plan to investigate situations where each robot’s sensing capability might not match its communication capability. For example, the robots might have directional sensing and communication capabilities, each of which might have a different range.

APPENDIX

Proof of Lemma 1. Any choice of \( \hat{x} \) has a fixed length \( ||\hat{x}|| = L \). For any choice of \( \hat{x} \) with a constant length, the term \( ||\tilde{y} - \hat{x}||^2 \) is minimized by \( \hat{x} = \frac{||\tilde{y}||}{\sqrt{||\tilde{y}||^2 + 2}} \). In that case, \( \min ||\tilde{y} - \hat{x}||^2 = (||\tilde{y}|| - ||\hat{x}||)^2 \). For any choice of \( \hat{x} \) with a constant length, the term \( ||\tilde{R}z + \hat{x}||^2 \) is minimized by \( \tilde{R}z \) such that \( \tilde{R}z = -\frac{||\tilde{z}||}{||\tilde{z}||^2} \hat{x} \). In that case, \( \min ||\tilde{R}z + \hat{x}||^2 = (||\tilde{z}|| - ||\hat{x}||)^2 \). Thus, by setting \( \hat{x} = \frac{||\tilde{z}||}{||\tilde{z}||^2} y \) and finding \( \hat{R} \) such that \( \hat{R} = -\frac{||\tilde{z}||}{||\tilde{z}||} \tilde{y} \), both terms can be individually minimized. The only remaining issue is the fixed length \( ||\hat{x}|| = L \). The objective function when both terms are individually minimized is \( (||\tilde{y}|| - ||\hat{x}||)^2 + (||\tilde{z}|| - ||\hat{x}||)^2 \) for fixed length \( ||\hat{x}|| = L \). By substituting this solution into the solution for the individual terms for fixed length \( ||\hat{x}|| = L \), we get the same solution as in Equation (6). Therefore, the solution in Equation (6) minimizes Equation (5).

Proof of Theorem 1. For the tree \( G \), let \( v_e \in V \) be the root node (i.e. the leader node). For a leaf node \( v_s \) of \( G \) the relevant terms of the objective function in Equation (3) are

\[
\left\| \hat{O}_s \hat{R}^m_{ms} - (\hat{r}_m^M - \hat{r}_i^M) \right\|^2 + \left\| \hat{O}_s \hat{R}^c_{ms} - (\hat{r}_m^C - \hat{r}_i^C) \right\|^2
\]

where node \( v_m \in V : (v_m, v_s) \in E \). If \( \hat{O}_s \) and \( \hat{r}_m^M \) are fixed, then by Lemma 1, the solution to this two-term objective function is given by Equation (8). This indicates that the solution \( \hat{O}_s \) and \( \hat{r}_m^M \) for every leaf node \( v_s \) can be written in terms of the solution \( \hat{O}_m \) and \( \hat{r}_m^M \) of the parent node \( v_m \). Since the graph is a tree, it can be stated more generally that the solution for every child node \( v_s \) can be written in terms of the solution for the parent node \( v_m \). Every node in the tree has the leader as an ancestor and the solution for the leader is given by \( \hat{r}_1^M = 0 \) and \( \hat{O}_1 = I_2 \). This means that all nodes in the graph can write their solution in terms of the solution for the leader. Every child node only needs to minimize the two terms in the objective function that correspond to the undirected edge with its parent. Therefore, Equation (8) is the optimal solution to the problem in Equation (3). ■

Proof of Corollary 1. The graph \( \mathcal{G} \) is not necessarily a tree, but consider a spanning tree of \( \mathcal{G} \). From Theorem 1, applying Equation (8) will optimally minimize the error in this tree. Since there is no measurement error,

\[
O_{ij}^R = -\hat{O}_{ij}^R, \quad r_{ij}^M = -r_{ij}^M
\]

Equation (8) reduces to solving

\[
\hat{O}_s \hat{R}^c_{ms} = -\hat{O}_m \hat{R}^m_{ms}, \quad \hat{r}_s^M = \hat{r}_m^M + \hat{O}_m \hat{R}^c_{ms}
\]

for \( \hat{O}_s \) and \( \hat{r}_s^M \) which is the same as Lines 15–17 of Algorithm 1 and lines 15–17 of Algorithm 2. Substituting Equation (10) and estimates from Equation (12) into objective function in Equation (3), all terms of objective function evaluate 0. Thus, Equation (3) is globally minimized. ■

ACKNOWLEDGMENTS

This research has been sponsored in part by ONR Grant N0001409-10680.

REFERENCES