Scale-Free Coordinates for Multi-Robot Systems with Bearing-only Sensors

Alejandro Cornejo, Andrew J. Lynch, Elizabeth Fudge
Siegfried Bilstein, Majid Khabbazian, James McLurkin

Abstract We propose *scale-free coordinates* as an alternative coordinate system for multi-robot systems with large robot populations. Scale-free coordinates allow each robot to know, up to scaling, the relative position and orientation of other robots in the network. We consider a weak sensing model where each robot is only capable of measuring the angle, relative to its own heading, to each of its neighbors. Our contributions are three-fold. First, we derive a precise mathematical characterization of the computability of scale-free coordinates using only bearing measurements, and we describe an efficient algorithm to obtain them. Second, through simulations we show that even in graphs with low average vertex degree, most robots are able to compute the scale-free coordinates of their neighbors using only 2-hop bearing measurements. Finally, we present an algorithm to compute scale-free coordinates that is tailored to low-cost systems with limited communication bandwidth and sensor resolution. Our algorithm mitigates the impact of sensing errors through a simple yet effective noise sensitivity model. We validate our implementation with real-world robot experiments using static accuracy measurements and a simple scale-free motion controller.

1 Introduction

Large populations of robots can solve many challenging problems such as mapping, exploration, search-and-rescue, and surveillance. All these applications require robots to have at least some information about the *network geometry*: knowledge about other robots positions and orientations relative to their own [1]. Different approaches to computing network geometry have trade-offs between the amount of information recovered, the complexity of the sensors, the amount of communications required and the cost. A GPS system provides each robot with a global position, which can be used to derive complete network geometry, but GPS is not available in many environments: indoors, underwater, or on other planets. The cost and complexity of most vision- and SLAM-based approaches makes them unsuitable for large populations of simple robots.

This work proposes *scale-free coordinates* as a slightly weaker alternative to the complete network geometry. We argue that scale-free coordinates provide sufficient information to perform many canonical multi-robot applications, while still being implementable using a weak sensing platform. Informally, scale-free coordinates provide the complete network geometry information up to an unknown scaling factor *i.e.* the robots can recover the shape of the network, but not its scale.

Formally, the scale-free coordinates of a set of robots S is described by a set of tuples $\{(x_i, y_i, \theta_i) \mid i \in S\}$. The relative position of robot $i \in S$ is represented by the coordinates (x_i, y_i) which match are correct up to the same (but unknown) multiplicative constant α. The relative orientation of robot $i \in S$ is represented by θ_i. Of particular interest to us are the *local scale-free coordinates* of a robot, which are simply the scale-free coordinates of itself and its neighbors, measured from its reference frame.

We consider a simple sensing model in which each robot can only measure the angle, relative to its own heading, to neighboring robots. These sensors are appropriate for low-cost robots that can be deployed in large populations [2]. Our approach allows each robot to use the bearing measurements available in the network to determine the relative positions and orientations of any subset of robots up to scaling. We remark that in this work we make no assumptions on the relationship between the Euclidean distance between two robots and presence of an edge in the communication graph between them. In particular, *we do not* assume the communication graph is a unit disk graph, or any other type of geometric graph.

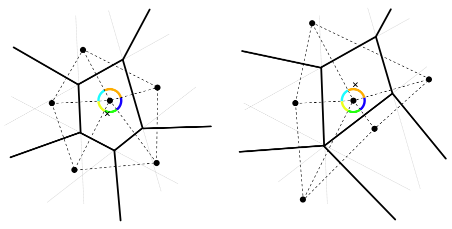


Fig. 1: Two distinct Voronoi cells with the same angle measurements. A robot cannot distinguish these cells using only the angle measurements to its neighbors.

With only local bearing measurements, a robot has the capability to execute a large number of algorithms [3, 4, 5], but this information is insufficient to directly compute all the parameters of its network geometry. For instance, consider the canonical problem of controlling a multi-robot system to a centroidal Voronoi configuration [6]. This is straightforward to solve with the complete network geometry, but it is not possible to using only the bearing measurements to your neighbors. Figure 1 shows two configurations with the same bearing measurements that produce very different Voronoi cells (in this diagram we assume robots at the center of adjacent Voronoi cells are neighbors in the communication graph). Local scale-free coordinates are sufficient for each robot to compute the shape of its Voronoi cell. However, since scale-free distances have no units, the robot cannot distinguish between 3 m or 3 cm distance to the centroid. This presents challenges to algorithms, in particular to motion control, which we consider in our experiments in Section 5.3.

There are three main contributions in this work. Section 3 presents the theoretical foundation for scale-free coordinates, and proves the necessary and sufficient conditions required to compute scale-free coordinates for the entire configuration of robots. We then generalize this approach to compute the scale-free coordinates of any subset of the robots. Section 4 shows through simulations, that in random configurations most robots are able to compute their local scale-free coordinates in only 3 or 4 communication rounds, even in in networks with low average degree. Section 5 presents a simplified algorithm, tailored for our low-cost multi-robot platform [7], to compute local scale-free coordinates using information from the 2-hop neighborhood around each robot. The 2-hop algorithm computes scale-free coordinates efficiently with a running time that is linear in the number of angle measurements. Our platform is equipped with sensors that only measure coarse bearing to neighboring robots, we mitigate the effect of this errors through a noise sensitivity model. We show accuracy data from static configurations, and implement a simple controller to demonstrate the feasibility of using the technique for motion control.

1.1 Related Work

Much of the previous work on computation of coordinates for multi-robot systems focuses on computing coordinates for each robot using beacon or anchor robots (or landmarks) with known coordinates [8, 9, 10]. There are also distributed approaches, which do not require globally accessible beacon robots, but instead use multi-hop communication to spread the beacon positions throughout the network [11]. Generally, these approaches do not scale for large swarms of simple mobile robots. Moreover, these approaches are generally based on some form of triangulation. In contrast, the approach proposed in this paper can be used to compute the scale-free coordinates even in graphs where there does not exist a single triangle.

The literature presents multiple approaches to network geometry such as pose in a shared external reference frame [12], pose in a local reference frame [1], distance-only [13, 14, 15] bearing-only [16, 17, 18], sorted order of bearing [19], or combinatorial visibility [20].

The closest in spirit to our work is the "robust quads" work of Moore et. al. [13]. Using inter-robot distance information, they find *robust quadrilaterals* in the network around each robot and combine them to recover the positions of the robot's neighbors. Our work is in the same vein, except that we use inter-robot bearing information instead, which allows us to also recover relative orientation. In the error-free case, we present localization success rates which are comparable to the Moore results. However, our approach has less requirements on the graph; scale-free coordinates can be extracted in a graph formed by robust quadrilaterals, but there are graphs without even a single robust quadrilateral where scale-free coordinates are computable.

From the computational geometry literature, the closest work to ours it that of Whiteley [21], who studied directional graph rigidity using the tools of matroid theory. This paper follows a simpler alternative algebraic characterization that allows us to directly compute the scale-free coordinates of any subset of robots. In addition, Bruck [22] addresses the problem of finding a planar spanner of a unit disk graph by only using local angles. The Bruck work is similar to our approach of forming a virtual coordinate system, but their focus delves into routing schemes for sensor networks.

Bearing-only models are more limited than range-bearing models and the type of problems to solve is reduced. In addition, the amount of inter-robot communication often increases greatly. The inter-robot communication requirement is often overlooked in the literature. However, algorithms that require large amounts of information from neighboring robots or many rounds of message passing are impractical on systems with limited bandwidth. This work uses the bearing-only sensor model with scale-free coordinates to balance the trade-off between cost, complexity, communications, and capability.

2 System Model and Definitions

We assume each robot is deployed at an arbitrary position in the Euclidean plane and with an arbitrary orientation unit vector. The communication network is modeled as an undirected graph, $G = (V, E)$, where every vertex in the graph represents a robot, and $N(u) = \{v \mid \{u, v\} \in E\}$ denotes the neighbors of robot u. We consider a synchronous network model, where the execution progresses in synchronous lock-step rounds. During each round every robot can send a message to its neighbors, and receive any messages sent to it by its neighbors. Moreover we assume that when node u receives a message from node v, it also measures the angle $\theta(u, v)$, relative to its own orientation, from u to v. These assumption greatly simplifies the analysis, and can be implemented easily in a physical system via synchronizers [1].

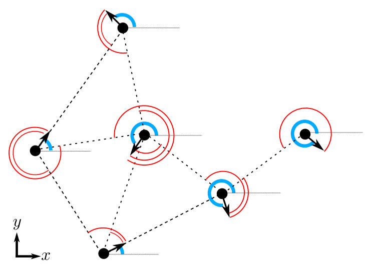


Fig. 2: The position of each robot is depicted by a black disk, and the orientation by a thick (blue) arc. Thin dotted lines connect neighboring robots. Thin (red) arcs represent the angle measurements.

We define the *realization* of graph G as a function $p : V(G) \to \mathbb{R}^2$ that maps each vertex of G to a point in the Euclidean plane. We use p_0 to denote the ground truth realization of G, specifically $p_0(v)$ is the position of robot v in a fixed global coordinate system. The function $\phi : V(G) \to [0, 2\pi)$ maps each robot v to its *orientation* $\phi(v)$, which is defined as the counter-clockwise angle between the $\hat{x}$-axis of the global coordinate system and v's orientation unit vector. For neighboring robots $\{u, v\} \in E(G)$ the angle measurement $\theta(u, v)$ from u to v is the counter-clockwise angle between the orientation unit vector of u and the vector from u to v (see Fig 2). We emphasize that at the beginning of the executions each robot knows only its own unique identifier. We do not assume a global coordinate system; the only way for robot u to sense other robots is by measuring the angles $\theta(u, v)$ for each of its neighbors $v \in N(u)$.

We define the function $\psi(\theta) := \left[\cos\theta \; \sin\theta\right]^T$ that maps an angle θ to the $\hat{x}$-axis when anticlockwise rotated θ radians. Analogously ψ^{-1} receives a unit vector and returns an angle via *atan*2 (i.e., $\psi(\psi^{-1}(\hat{n})) = \hat{n}$).

For each undirected edge we consider its two directed counterparts, specifically we use $\overleftrightarrow{E}(G) = \{(u, v), (v, u) \mid \{u, v\} \in E(G)\}$ to denote the directed edges present in G. The function $\ell : \overleftrightarrow{E}(G) \to \mathbb{R}^+$ represents a set of *length-constraints* on the graph, and associates to each directed counterpart of every edge $\{u, v\} \in E(G)$ a "length" such that $\ell(u, v) = \ell(v, u)$ (i.e. ℓ is symmetric). Similarly, the function $\omega : \overleftrightarrow{E}(G) \to [0, 2\pi)$ represents a set of *angle-constraints* on the graph, and associates to each directed counterpart of every edge $\{u, v\} \in E(G)$ an "angle" such that $\omega(u, v) = (\omega(v, u) + \pi) \mod 2\pi$ (i.e., ω is antisymmetric, and this implies $\psi(\omega(u, v)) = -\psi(\omega(v, u))$). Observe that if all the robots had the same orientation then the set of all angle measurements would describe a set of angle-constraints on G.

We say p is a *satisfying realization* of (G, ℓ) iff every edge $(u, v) \in \overleftrightarrow{E}(G)$ satisfies $\|p(v) - p(u)\| = \ell(u, v)$. Realizations are *length-equivalent* if one can be obtained from the other by a translation, rotation or reflection (distances are invariant to these operations). A length-constrained graph (G, ℓ) has a *unique realization* if all its satisfying realizations are length-equivalent. Similarly, we say p is a *satisfying realization* of (G, ω) iff every edge $(u, v) \in \overleftrightarrow{E}(G)$ satisfies $p(v) - p(u) = \psi(\omega(u, v)) \|p(v) - p(u)\|$. Realizations are *angle-equivalent* if one can be obtained from the other by a translation or uniform-scaling (angles are invariant to these operations). An angle-constrained graph (G, ω) has a *unique realization* if all its satisfying realizations are angle-equivalent.

3 Theoretical Foundation for Scale-Free Coordinates

This section develops a mathematical framework that characterizes the computability of scale-free coordinates and outlines an efficient procedure to compute them. The full procedure derivation with proofs appears in a tech report [23]. Here we omit some intermediate results and present a self-contained summary. First all the robots in the network to agree on a common reference orientation. In a connected graph this is accomplished by having each robot propagating orientation offsets to the entire network with a broadcast tree. As a side-effect of this procedure every robot can compute the relative orientation of every other robot. The details of this distributed algorithm, along with proofs of correctness, appear in [23]. In the rest of the paper we assume that all angle measurements are taken with respect to a global $\hat{x}$-axis and therefore constitute a valid set of angle-constraints on the graph.

Given an angle-constrained graph, the task of computing scale-free coordinates for every robot is equivalent to finding a unique satisfying realization of the graph. If such a realization does not exist, then either there is no set of scale-free coordinates consistent with the angle-measurements, (perhaps due to measurement errors), or there are multiple distinct sets of scale-free coordinates which produce the same angle measurements, and it is impossible to know which one of them corresponds to the ground truth. We note that every realization of a graph induces a unique set of length- and angle-constraints which are simultaneously satisfied by that realization:

Proposition 1. *A realization p of a graph G induces a unique set of length- and angle-constraints ℓ_p and ω_p which are simultaneously satisfied by p.*

However, the converse does not hold, since there are length- and angle-constraints that do not have a realization which satisfies them simultaneously.

The necessary and sufficient conditions that determine if a set of angle-constraints have a satisfying realization are captured by the cycles of the graph. In particular, given any realization p of G, traversing a directed cycle C of G and returning to the starting vertex there will be no net change in position or orientation. Formally:

$$\sum_{(u,v)\in E(\mathbf{C})} (p(v) - p(u)) = \sum_{(u,v)\in E(\mathbf{C})} \ell_p(u,v)\psi(\omega_p(u,v)) = \mathbf{0}. \tag{1}$$

Since by definition $\ell_p(u,v) = \ell_p(v,u)$ and $\psi(\omega_p(u,v)) = -\psi(\omega_p(v,u))$ we can verify that the direction in which we traverse an undirected cycle is not relevant, since both directions produce the same equation. Since the terms of the equations are two-dimensional vectors, each cycle generates two scalar equations for the x- and y-component. If the realization p of G is unknown, but we know both G and a set of angle-constraints ω of G, then equation 1 represents two linear restrictions on the length of the edges of any realization p which satisfies (G,ω).

The number of cycles in a graph can be exponential, however we show it suffices to consider only the cycles in a *cycle basis* of G. For a detailed definition of a cycle basis we refer the interested reader to [24]. Briefly, a cycle basis of a graph is a subset of the simple undirected cycles present in a graph, and a connected graph on n vertices and m edges has a cycle basis with exactly $m-n+1$ cycles. A cycle basis of G can be constructed in $\mathcal{O}(m\cdot n)$ time by first constructing a spanning tree T of G. This leaves $m-n+1$ non-tree edges, each of which forms a unique simple cycle when added to T. Let $\mathscr{C} = \{C_1,\ldots,C_q\}$ be any cycle basis of G.

It will be useful to represent the length of the edges of a realization as a real vector. Let $\mathcal{E} = \{e_1,\ldots,e_m\}$ be any ordered set of directed edges that cover all the undirected edges in $E(G)$. Specifically for every undirected edge in $E(G)$ one of its directed counterparts (but not both) is present in $\mathcal{E}(G)$, conversely if a directed edge is present in $\mathcal{E}(G)$ then its undirected version is in $E(G)$. Let $\mathbf{x}$ be an $m\times 1$ column vector whose i^{th} entry represents the length of the directed edge $e_i \in \mathcal{E}$ of *any* satisfying realization of (G,ω).

Applying equation 1 to a cycle basis of G results in the following:

$$\begin{array}{c} \\ C_1 \\ \vdots \\ C_q \end{array} \underbrace{\begin{array}{c} \begin{array}{ccc} e_1 & \cdots & e_m \end{array} \\ \begin{bmatrix} a_{11} & \ldots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{q1} & \ldots & a_{qm} \end{bmatrix} \end{array}}_{A_{(G,\omega)}} \underbrace{\begin{bmatrix} \ell_p(e_1) \\ \vdots \\ \ell_p(e_m) \end{bmatrix}}_{\mathbf{x}} = \mathbf{0}. \tag{2}$$

Here $A_{(G,\omega)}$ is a $2q\times m$ matrix constructed using G, $\mathscr{C}$ and ω. Row i corresponds to a cycle $C_i \in \mathscr{C}$, and column j corresponds to an edge $(u,v)\in\mathcal{E}$. If $(u,v)\in E(C_i)$ then $a_{ij} = \psi(\omega(u,v))$, if $(v,u)\in E(C_i)$ then $a_{ij} = -\psi(\omega(u,v)) = \psi(\omega(v,u))$, otherwise $a_{ij} = 0$. Since these are vector equations, there are two scalar rows in $A_{(G,\omega)}$ for every cycle in $\mathscr{C}$ – one equation for the x-components and one for the y-components of each cycle.

Equation 2 is a homogeneous system, therefore the solution space is precisely the null space of $A_{(G,\omega)}$, denoted by $\mathsf{null}(A_{(G,\omega)})$. Our main result relates the null space of $A_{(G,\omega)}$ to the space of satisfying realizations of (G,ω).

Let $P_{(G,\omega)}$ be a set of realizations that satisfy (G,ω), where all equivalent realizations are mapped to a single realization that "represents" its equivalence class. We define the function $f_\omega : P_{(G,\omega)} \to \mathbb{R}^m$ that maps a realization in $P_{(G,\omega)}$ to a (positive) m-dimensional real vector which contains in its i^{th} entry the length of the directed edge $e_i\in\mathcal{E}$. Therefore $f_\omega(p)$ is simply a vector representation of the set of length-constraints ℓ_p satisfied by p. Observe that proposition 1 implies that when the domain of f_ω is restricted to $P_{(G,\omega)}$ then f_ω has an inverse f_ω^{-1}. We now state the main theorem of this section:

Theorem 2. *$p \in P_{(G,\omega)}$ if and only if $f_\omega(p) \in \mathsf{null}(A_{(G,\omega)})$.*

This theorem implies each column in the null space basis of $A_{(G,\omega)}$ corresponds to a distinct satisfying realization of (G,ω), and therefore a distinct set of scale-free coordinates. If the nullity of $A_{(G,\omega)}$, the number of columns of its null space basis, is zero no set of scale-free coordinates is consistent with the angle-measurements. If the nullity is one, then there is a single set of scale-free coordinates which are consistent with the angle-measurements. If on the other hand the nullity of $A_{(G,\omega)}$ is greater than one, then there are multiple distinct sets of scale-free coordinates consistent with the angle measurements and its impossible to know which one corresponds to the ground truth. We summarize this in the following corollary.

Corollary 1. *(G,ω) has a unique satisfying realization $\iff$ the nullity of $A_{(G,\omega)}$ is one $\iff$ the scale-free coordinates of every robot in G are computable.*

3.1 Local Scale-Free Coordinates

This subsection describes a procedure that uses the null space basis of $A_{(G,\omega)}$ to compute the scale-free coordinates of any subset of robots. Of particular interest to us is computing the scale-free coordinate of a specific robot and its neighbors (i.e., its local scale-free coordinates). From corollary 1 it follows that if the null space basis of $A_{(G,\omega)}$ has a single column, then we can compute the scale-free coordinates of any subset of robots, since we can compute scale-free coordinates for all robots simultaneously. However, it might be the case that the null space basis of $A_{(G,\omega)}$ has more than one column, but it is still possible to compute the scale-free coordinates of some subset of the robots.

For a set $S \subseteq V(G)$ of vertices, let $G[S]$ be the subgraph of G induced by S, and let $\ell[S]$ and $\omega[S]$ be the length- and angle-constraints that correspond to the edges in $G[S]$. We say an angle-constrained (G,ω) has a unique S-subset realization iff when restricted to the vertices of S all realizations of (G,ω) projected to the vertices in S are equivalent. From this definition we can see that the scale-free coordinates of the subset of robots in S are computable iff (G,ω) has a unique S-subset realization. Using these definitions we can prove the following.

Lemma 3. *The angle-constrained graph (G,ω) has a unique S-subset realization iff there is a superset $S' \supseteq S$ such that $(G[S'],\omega[S'])$ has a unique realization.*

The FixedTree algorithm leverages this lemma to compute scale-free coordinates for any subset of robots. The FixedTree algorithm receives as input a graph G, a subset of vertices $S \subseteq V(G)$, and a null space basis $\mathcal{N}$ of $A_{(G,\omega)}$. If there exists a superset $S' \supseteq S$ such that $(G[S'],\omega[S'])$ has a unique realization it will return this set. From corollary 1 it follows that we can use this set S' and the null space basis $\mathcal{N}$ to compute the unique satisfying realization, and therefore the scale-free coordinates, of $S' \supseteq S$.

Algorithm 1 FixedTree($G, S, \mathcal{N}$)

```
Pick $w \in S$ arbitrarily.
for each $\{w, v\} \in E(G)$ where $\{w, v\}$ is not degenerate in $\mathcal{N}$ do
    $\mathcal{N}' \leftarrow$ Fix edge $\{w, v\}$ in $\mathcal{N}$
    $T \leftarrow$ BFS traversal of $G$ rooted at $w$ using only edges fixed in $\mathcal{N}'$.
    if $T$ spans all vertices in $S$ then
        return $(\mathcal{N}, T)$
end for
return NoSolution
```

Recall that each row in the null space basis $\mathcal{N}$ corresponds to an edge of G, we define a labeling of the edges in G using $\mathcal{N}$. Fix an edge $e \in G$ and let j be the row in $\mathcal{N}$ that corresponds to e, (1) if there are both zero and non-zero entries in row j then e is *degenerate*, (2) if all entries in row j are the same then e is *fixed*, (3) otherwise e is *flexible*.

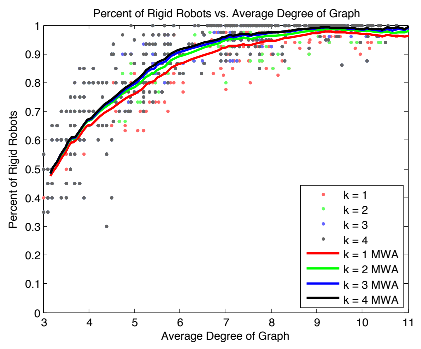
The Fix transformation –which relies on elementary column operations– receives an edge e and a null space basis $\mathcal{N}$ where e is labeled as flexible, and returns a null space basis $\mathcal{N}'$ where edge e is labeled as fixed. Specifically to Fix an edge e, which corresponds to a row j in a null space basis $\mathcal{N}$, it suffices to multiply each column i of $\mathcal{N}$ by the reciprocal of element n_{ij} in that column.

The algorithm uses the Fix transformation to finds a tree in G (if it exists) that spans the vertices in S and whose edges can be simultaneously fixed in the null space basis $\mathcal{N}$. In other words, the Fix algorithm finds a projection of the null space basis $\mathcal{N}$ which is of rank 1 and spans all the vertices in S. The proof of correctness algorithm follows from lemma 3 and theorem 2.

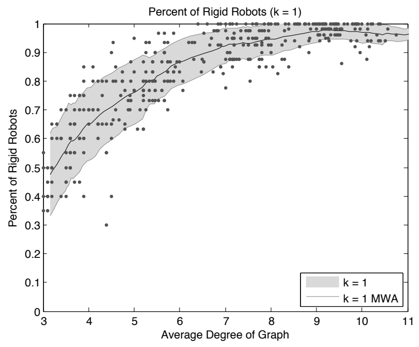
4 Simulation

Here we show that in the robots are deployed in random positions, it is feasible for each robot to compute the local scale-free coordinates of its neighbors using only the angle-measurements taken by other near-by robots. The simulation uses the FixedTree algorithm presented in the previous section. We use G_u^k to denote the k-neighborhood of robot u, which is the set of nodes at k or less hops away from u and the edges between these nodes. In practice to obtain its k-neighborhood G_u^k and the corresponding angle measurements, robot u will need $k+1$ communication rounds, using messages of size at most $O(\Delta^k)$ where Δ is the maximum degree of the graph. To compute the local scale-free coordinates for robot u using only its k-neighborhood we let $G = G_u^k$, $S = \{u\} \cup N(u)$ and $\mathcal{N}$ be the null space basis of $A_{(G_u^k, \omega)}$. In other words, we use only the null space of the matrix associated with the k-neighborhood of each node and not the entire graph. The computational complexity of the whole procedure is dominated by computing the null space basis. This was implemented using singular value decomposition requiring $O(m^3)$ time where m is the number of edges in G_u^k.

We ran simulations to determine how useful the algorithm would be in random graphs of various average degrees. Each robot is modeled as a disk with a 10 *cm* diameter and a communication range of 1 *m*. For each trial, we consider a circular environment with a 4 *m* diameter. We assume lossless bidirectional communication and noiseless bearing-only sensors. To be consistent with our hardware platform, we used the same sensing range as the communication range. We considered stationary configurations, but the results



(a) All Graphs



(b) Data for $k = 1$

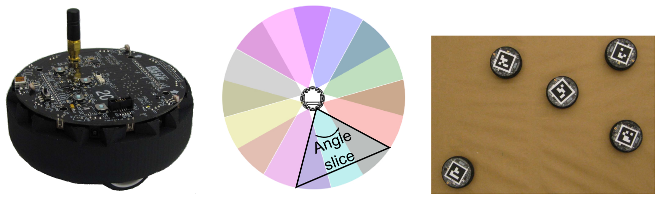
Fig. 3: Simulation data from 500 random graphs. **a:** Percentage of robots with scale-free coordinates vs. average degree of graph. A moving window average for each communication depth is overlaid on the plot. **b:** A closer look at $k = 1$ data in all generated graphs. The shaded area represents one standard deviation away from the moving window average.

presented are applicable while the system is in motion as long as the physical speed of the robots is negligible compared to the speed of communication and computation [1]. Random connected geometric graphs were generated by placing robots uniformly at random in the environment and discarding disconnected graphs. The parameters in the experiments are the population size and the communication depth k (or hop count). The population size controls the density of the graph and the communication depth controls the amount of information available to each robot. In real systems the communication depth will be limited by bandwidth constraints.

We carried out simulations using populations of 20, 30, 40, and 50 robots, running 100 experiments for each population size. Our results show that in 43% of these graphs all robots can successfully compute scale-free coordinates for every other robot, if they use a communication depth $k = diam(G)$, i.e. $G_u^k = G$. Since in practice bandwidth will be limited we are more interested in the percentage of robots which were able to compute local scale-free coordinates using small constant communication depths.

Our results shown in Figure 3a are encouraging; even for graphs with an average degree as low as 4, we can expect at least half of robots to successfully compute their local scale-free coordinates. For more typical graphs with an average degree of 6-7, on average 90% of robots can compute their local scale-free coordinates with a communication depth of only $k = 1$. The $k = 1$ depth is the most practical in our experimental platform, because only two communication rounds are required with messages of size $O(\Delta)$. Figure 3b shows a closer look at the data of Figure 3a for $k = 1$. For graphs of degree 6, approximately 80% of the robots can compute their local scale-free coordinates using $k = 1$, indicating that this is a feasible technique for bandwidth-limited platforms.

Figure 3b also allows for a direct comparison to the robust quad results of Moore [13]. Our localization success rates are somewhat better than robust quad results, with the same communication depth $k = 1$, around 10% more



(a) r-one robot (b) IR regions (c) APRIL tags

Fig. 4: **a:** The r-one robot for multi-robot research was designed by MRSL group at Rice University. **b:** Top view of the r-one's IR receiver detection regions. Each receiver detects an overlapping 68°, allowing a robot to determine the neighbor angle within 22.5°. **c:** An image from the overhead camera from our data collection system. The robots are outfitted with APRIL tags for detection of ground-truth 2D position and orientation.

nodes with low degree can localize using our algorithm with bearing-only measurements than the robust-quads algorithm with distance-only measurements. In addition, increasing communication depth from $k = 1$ to $k = 2$ increases likelihood of a given robot being able to compute its local scale-free coordinates. Subsequent increases in k have diminishing returns.

5 Hardware Experiments

For our experiments, we use the r-one robot shown in Figure 4a [2]. It is a 11 cm robot with a 32-bit ARM-based microcontroller running at 50 mhz with no floating point unit. The local IR communication system is used for inter-robot communication and localization. Each robot has eight IR transmitters and eight receivers. The transmitters broadcast in unison and emit a radially uniform energy pattern. The robot's eight IR receivers are radially spaced to produce 16 distinct detection regions (shown in Figure 4b). By monitoring the overlapping regions, the bearing of neighbors can be estimated to within $\approx \pi/8$. The IR receivers have a maximum bit rate of 1250bps. Each robot transmits $(\varDelta+1)$ 4-byte messages during each round, one a system announce message, and the others contain the bearing measurements to that robot's neighbors. The system supports a maximum $\varDelta = 8$, and we used a $\varDelta = 4$ for the motion experiments. A round period of 2500 ms was used to minimize the number of message collisions.

The APRIL tags software system [25] (shown in Figure 4c) is used to provide ground-truth pose information. The system provides a mean position error of $6.56mm$ and mean angular error of $9.6mrad$, which we accept as ground truth.

5.1 TwoHop Scale-Free Algorithm

Given the computational and bandwidth constraints of our platform, it is unfeasible to compute in real-time the null space of the system of cycle equations described in Section 4. However, our simulation results show that a system that uses only 2-hop angle measurements should work reasonably well in practice. In this section we describe a simple distributed algorithm that computes local scale-free coordinates using only 2-hop angle measurements, and which can be implemented easily and efficiently in hardware without a floating point unit (this corresponds to $k = 1$ in our simulation experiments, but as we mentioned earlier, this requires two communication rounds and angle measurements from 2-hops, hence the name). Later we describe how to modify the algorithm to deal with sensing errors.

The main insight behind our algorithm is that instead of considering an arbitrary cycle-basis, when restricted to a 2-hop neighborhood of u we can always restrict ourselves to a cycle-basis composed solely of triangles of which node u is a part of. This is a consequence of the following lemma.

Theorem 4. *Robot u can compute its local scale-free coordinates using 2-hop angle measurements if and only if the graph induced by the vertices in $N(u)$ is connected.*

The basic idea behind the TwoHop Scale-Free algorithm is to traverse a tree of triangles, computing the lengths of the edges of the triangles using the SineLaw . Specifically, SineLaw receives a triangle (u, z, w) in in the 2-hop neighborhood of u. It assumes the length ℓ_z of the edge (u, z) is known (up to scale), and uses the inner angles $\psi_z = \theta(z, u) - \theta(z, w)$ and $\psi_w = \theta(w, z) - \theta(w, u)$ to return the length (up to scale) of edge (u, v).

$$\textsc{SineLaw}(u, z, w) = \ell_z \left| \frac{\sin(\theta(z, u) - \theta(z, w))}{\sin(\theta(w, z) - \theta(w, u))} \right|$$

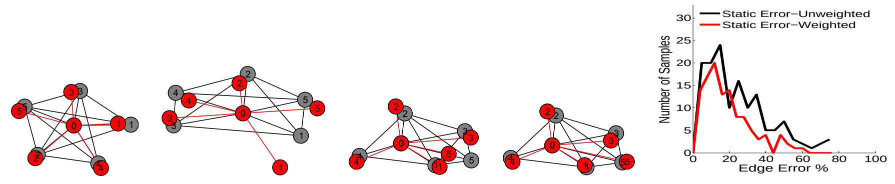
The following algorithm has a running time which is linear in the number of angle measurements in the 2-hop neighborhood of u.

Algorithm 2 TwoHop Scale-Free algorithm running at node u

```
1: Fix $v \in N(u)$
2: mark $v$ and set $\ell_v \leftarrow 1$
3: $Q \leftarrow queue(v)$
4: while $Q \neq \varnothing$ do
5:     $z \leftarrow Q.pop()$
6:     for each unmarked $w \in N(z) \cap N(u)$ do
7:         mark $w$ and set $\ell_w \leftarrow$ SineLaw $(u, z, w)$
8:         $Q.push(w)$
9:     end for
10: end while
```

Noise Sensitivity. To deal with coarse sensor measurements while preserving the computational efficiency (and simplicity) of the algorithm we introduce the concept of *noise sensitivity*. Informally, the noise sensitivity of a triangle captures the expected error of the lengths of a triangle when its angles are subject to small changes. For example, observe that given a triangle (u, v, z), as ψ_z gets closer to zero, the output of the SineLaw becomes more "sensitive to noise", since a small change in the angle measurements used to compute ψ_z translate to a potentially very large change in the computed length. Formally, the noise sensitivity of each triangle can be defined as a function of the magnitude of the vector gradient of SineLaw(u, v, z). This provides us with an approximation of the expected error in the computed length when using a particular triangle.

Hence, to reduce the effect of noisy measurements in the computed scale-free coordinates it suffices to find a spanning tree of triangles that has the smallest total noise sensitivity. This can be achieved by any standard minimum spanning tree algorithm at minimal additional computational cost. Specifically in our setting a minimum spanning tree of triangles can be found



(a) (b) (c) (d) (e)

Fig. 5: Scale-free coordinates plot as red nodes and ground-truth data as grey nodes. The IR communication links plot as black edges between grey nodes. The red lines depict the measured bearing between each robot, the block lines are the edges from the ground-truth positions. Four cases are presented: **a:** Accurate scale-free coordinates. **b:** Configuration with a bearing error to robot 1. **c:** Scale-free edge length error to robot 5. **d:** Scale-free edge length corrected with noise sensitivity. **e:** Edge error histograms with and without noise sensitivity for 28 robot configurations and 140 edges.

in $O(m \log m)$ time, where m is the number of angle measurements in the 2-hop neighborhood of u.

5.2 Static Evaluation

We generated 32 random configurations of six r-one robots. Four trials failed due to lost messages between robots, we discarded them and analyze the 28 successful trials. The configurations shown in Figure 5 illustrate some typical errors and the overall accuracy of our experiments. Ideally, the red nodes and edges will directly cover the black edges and grey nodes. The low resolution of the r-one localization system is the largest source of error. Lost messages between robots would occasionally remove edges from the local network, resulting in missing triangles.

To analyze each static configuration, we needed a way to compare scale-free edge lengths to ground-truth edge lengths. For each configuration, we computed an α_{opt} scaling factor that minimized the total edge length error when compared to ground truth. An example of a bearing inaccuracy is shown in Figure 5b for robot 1 to robot 0. Despite this error, our algorithm still effectively computes the edge coefficient. Bearing measurement errors cause the most significant problems in our scale-free coordinates. However, the majority of the bearing errors are still within the 22.5°designed tolerance of the robot. Figure 5c illustrates a scale-free edge length error to robot 5. In this case, the error was caused by a poor selection of triangles. We handled this scenario with noise sensitivity to select a better set of triangles. The corrected position of robot 5 is shown in Figure 5d. The summary error statistics are shown in Figure 5e. Running the algorithm without error sensitivity produced a mean error of 23.4%, and with sensitivity produced a mean error of 19.4%. Given our coarse bearing measurements, these results are reasonable, and are adequate for motion control.

5.3 Dynamic Evaluation: Real-time Centroid Behavior

This experiment measures the ability of the robot to move to a position specified by local scale-free coordinates, in this case, the centroid of a group of robots. Our controller is basic, it computes the centroid, rotates, and moves a fixed distance. This is intentional — the aim of these experiments is to illustrate the performance of scale-free coordinates algorithm, so we use unfiltered data. We also avoided using any odometry information to improve performance. Since our neighbor round is a (very long) 2500 ms, measuring neighbor bearings while moving can introduce errors, therefore robots remain stationary when measuring the neighbor bearings.

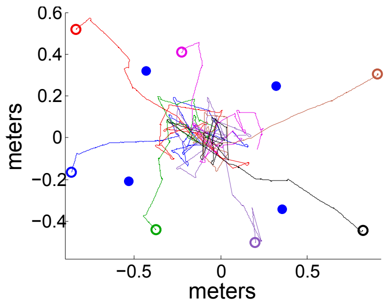
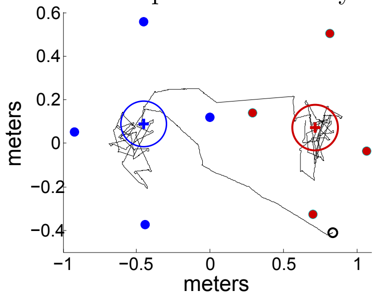


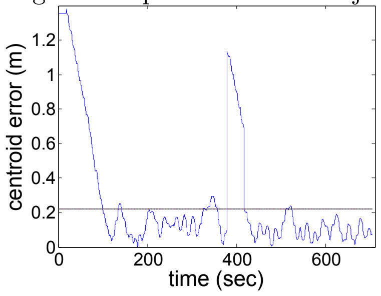
Fig. 6: Four static robots (blue dots) were placed in an arbitrary polygon. A mobile robot was placed in random locations outside the polygon (colored circles). Trajectories of the robot moving toward the centroid are represented by the different colored lines. The robot quickly reaches the centroid, but then oscillates because is does not know how far the goal is from its current position.

For the first experiment, four stationary robots were arranged in an arbitrary polygon and one moving robot is placed at random initial positions outside the polygon. At each iteration of the algorithm, the moving robot moves a distance of (d_{step}) towards the centroid. For this experiment, we used the robot diameter of 11 cm for (d_{step}). The trajectories of the moving robot converging to the centroid are shown in Figure 6. The robot continues to move around the centroid without settling because without knowing the distance to the centroid, the robot cannot know when to stop. We expect the diameter of the convergence region around the centroid to have a mean diameter of approximately $d_{step} = 0.11$ m, which is consistent with our measurement of 0.14 m.

The second experiment looks at the controller's response to a change in the goal position. The stationary robots start in the blue positions, then were moved to the red positions halfway through the experiment. The trajectory



(a) Centroid shift.



(b) Centroid shift error.

Fig. 7: **a:** This experiment shifts a group of robots to demonstrate a large shift in the centroid denoted by the blue plus sign. The four blue dots represent the initial polygon of static robots. The red dots represent the shifted group of robots. The black line trajectory shows the trajectory of the motion robot searching for the centroid. **b:** Corresponding error vs. time of trajectory shown in (a) between the motion robot position and the centroid. The robot begins at the black circle with significant error and then oscillates less than $2d_{step}$ around centroid. When the group is shifted the error spikes again and settles to another oscillation around the shifted centroid.

shown in Figure 7a show the moving robot successfully converging to the new position, and the size of the convergence region in Figure 7b is again around d_{step}, and mostly bounded by $2d_{step}$, which is shown as the circles in Figure 7a and the horizontal line in Figure 7b.

While the size of the convergence region is set by the step size, the time of convergence is limited by the communications bandwidth — more bandwidth can allow shorter rounds. This blurs the distinction between sensing and communication, but is consistent with the robot speed ratio [26].

6 Conclusion and Future Work

This paper presents local scale-free coordinates as an alternative coordinate system of intermediate power. Our noise sensitivity provided a computationally simple way to deal with sensor errors. However, in future work we will incorporate a full error model to provide superior performance.

In a separate project, we are studying the accuracy of a particle filter to estimate range using odometry and the bearing sensors [27]. This approach uses less communications and provides metrical estimates of range, but requires the robots to be moving, and remain neighbors long enough for the estimate to converge. On the other hand, the approach presented in this paper can be applied even if the robots are static (or to sensor networks). It is unclear which of these two approaches is the most powerful, in the sense proposed by O'Kane [28], which is an interesting question. We believe that for many applications, scale-free coordinates are a viable alternative for relative localization in multi-robot platforms with large populations of simple, low-cost robots.

References

1. James McLurkin. *Analysis and Implementation of Distributed Algorithms for Multi-Robot Systems.* Ph.D. thesis, Massachusetts Institute of Technology, 2008.
2. James McLurkin, Andrew J. Lynch, Scott Rixner, Thomas W. Barr, Alvin Chou, Kathleen Foster, and Siegfried Bilstein. A low-cost multi-robot system for research, teaching, and outreach. *Proc. of the Tenth Int. Symp. on Distributed Autonomous Robotic Systems DARS-10, October*, 2010.
3. Ran Wei, R. Mahony, and D. Austin. A bearing-only control law for stable docking of unicycles. In *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 4, pages 3793–3798 vol.3, 2003.
4. T. Lemaire, S. Lacroix, and J. Sola. A practical 3D bearing-only SLAM algorithm. In *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 2449–2454, 2005.
5. S. Scheding, G. Dissanayake, E.M. Nebot, and H. Durrant-Whyte. An experiment in autonomous navigation of an underground mining vehicle. *Robotics and Automation, IEEE Transactions on*, 15(1):85–95, 1999.
6. J. Cortes, S. Martinez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, volume 2, pages 1327–1332, 2002.
7. J. McLurkin, A. Lynch, S. Rixner, T. Barr, A. Chou, K. Foster, and S. Bilstein. A low-cost multi-robot system for research, teaching, and outreach. *Proc. of the Tenth Int. Symp. on Distributed Autonomous Robotic Systems DARS-10, November*, page 200, 2010.
8. B. Hendrickson. The molecule problem: Exploiting structure in global optimization. *SIAM Journal on Optimization*, 5(4):835–857, 1995.
9. T. Eren, D.K. Goldenberg, W. Whiteley, Y.R. Yang, A.S. Morse, B.D.O. Anderson, and P.N. Belhumeur. Rigidity, computation, and randomization in network localization. In *Proc. 23rd IEEE Conference on Computer Communications*, volume 4, pages 2673–2684, 2004.

10. Kostas E. Bekris, A. A. Argyros, and L. E. Kavraki. Angle-based methods for mobile robot navigation: Reaching the entire plane. In *Proc. EEE International Conference on Robotics and Automation (ICRA)*, pages 2373–2378, 2004.
11. R. Nagpal, H. Shrobe, and J. Bachrach. Organizing a global coordinate system from local information on an ad hoc sensor network. *Proc. of Information Processing in Sensor Networks (IPSN)*, 2003.
12. Pradeep Ranganathan, Ryan Morton, Andrew Richardson, Johannes Strom, Robert Goeddel, Mihai Bulic, and Edwin Olson. Coordinating a team of robots for urban reconnaisance. In *Proceedings of the Land Warfare Conference (LWC)*, November 2010.
13. D. Moore, J. Leonard, D. Rus, and S. Teller. Robust distributed network localization with noisy range measurements. In *In Proc. 2nd international conference on Embedded networked sensor systems*, pages 50–61, 2004.
14. Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. The cricket location-support system. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 32–43, Boston, Massachusetts, United States, 2000. ACM.
15. Sooyong Lee, Nancy M. Amato, and James Fellers. Localization based on visibility sectors using range sensors. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 3505–3511, 2000.
16. B. Sundaram, M. Palaniswami, S. Reddy, and M. Sinickas. Radar localization with multiple unmanned aerial vehicles using support vector regression. In *Intelligent Sensing and Information Processing, 2005. ICISIP 2005. Third International Conference on*, pages 232 –237, 2005.
17. L. Montesano, J. Gaspar, J. Santos-Victor, and L. Montano. Cooperative localization by fusing vision-based bearing measurements and motion. In *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 2333 – 2338, August 2005.
18. S.G. Loizou and V. Kumar. Biologically inspired bearing-only navigation and tracking. In *Decision and Control, 2007 46th IEEE Conference on*, pages 1386 –1391, 2007.
19. Robert Ghrist, David Lipsky, Sameera Poduri, and Gaurav S. Sukhatme. Surrounding nodes in coordinate-free networks. In *Workshop on the Algorithmic Foundations of Robotics*, 2006.
20. Davide Bil, Yann Disser, Mat Mihalk, Subhash Suri, Elias Vicari, and Peter Widmayer. Reconstructing visibility graphs with simple robots. In *Structural Information and Communication Complexity*, pages 87–99, 2010.
21. W. Whiteley. Matroids from Discrete Geometry. *AMS Contemporary Mathematics*, 197:171–312, 1996.
22. Jehoshua Bruck, Jie Gao, and Anxiao (Andrew) Jiang. Localization and routing in sensor networks by local angle information. *ACM Transactions on Sensor Networks*, 5(1):7:1–7:11, February 2009.
23. A. Cornejo, M. Khabbazian, and J. McLurkin. Theory of scale-free coordinates for multi-robot system with bearing-only sensors. *Technical Report, http://mrsl.rice.edu/publications*, 2011.
24. J. D. Horton. A Polynomial-Time algorithm to find the shortest cycle basis of a graph. *SIAM Journal on Computing*, 16(2):358, 1987.
25. Edwin Olson. Apriltag: A robust and flexible multi-purpose fiducial system. Technical report, University of Michigan APRIL Laboratory, May 2010.
26. J. McLurkin. Measuring the accuracy of distributed algorithms on Multi-Robot systems with dynamic network topologies. *9th International Symposium on Distributed Autonomous Robotic Systems (DARS)*, 2008.
27. J. B. Rykowski. *Pose Estimation With Low-Resolution Bearing-Only Sensors*. M.S. thesis, Rice University, 2011.
28. J. M. O'Kane and S. M. LaValle. Comparing the power of robots. *The International Journal of Robotics Research*, 27(1):5, 2008.