WINDING NUMBERS FOR NETWORKS WITH WEAK ANGULAR DATA

ROBERT GHRIST

ABSTRACT. We consider a network of nodes in the plane whose locations are unknown but for which the nodes establish communication links based on proximity, as in an ad hoc wireless network. Given a node in the network and a cycle disjoint from the node, we want the network to compute the *winding number* and determine, in particular, whether the cycle surrounds the node in the plane. We assume that nodes possess unique IDs, a sharp notion of proximity, and, to various degrees, a weak form of angular ordering between neighbors.

We give algorithms for computing winding numbers in networks with varying sensing capabilities and architectures with regards to angular data of neighbors. In all instances, we are motivated to compute the *minimal sensing* required to compute winding data rigorously. This takes the form of investigating varying amounts of uncertainty in angular readings.

1. Introduction

This paper considers a discrete, network-theoretic version of a simple classical topology problem: given a point x_0 in the plane $\mathbb{R}^2$ and a simple closed curve $\mathcal{L}$, determine whether or not $\mathcal{L}$ surrounds the point (see Fig. 1[left]). This question leads one to the development of *winding numbers* [**14**]. Recall that the winding number of a planar cycle $\mathcal{L}$ about a point $x_0 \in \mathbb{R}^2$ is, roughly speaking, the number of times the cycle wraps around the point. Winding numbers can be computed in a number of ways, including:

1. **analytically:** via integrating the tangent vector to $\mathcal{L}$ about $\mathcal{L}$.
2. **topologically:** via computing the homology class of $\mathcal{L}$ in $H_1(\mathbb{R}^2 - \{x_0\})$.
3. **combinatorially:** via computing the algebraic intersection number of $\mathcal{L}$ with a transverse ray based at x_0 in $\mathbb{R}^2$.

As the winding number can be defined as the degree of the inclusion map $\iota : \mathcal{L} \hookrightarrow \mathbb{R}^2 - \{x_0\}$, it is a homotopy invariant, meaning that deforming the loop $\mathcal{L}$ does not change the winding number (so long as the deformed curves never intersect x_0).

In this setting, the winding problem is very easily solved using any of the methods listed above. We consider a network-theoretic version of the problem, in

Key words and phrases. winding number, sensor network, degree, unit disk graph.

which x_0 is a node in an abstract network graph Γ whose vertices represent sensors or stationary robots in the plane and whose edges encode proximity. The loop $\mathcal{L}$ is in this setting an abstract cycle in this graph (Fig. 1[right]). The challenge is to compute winding number data without recourse to coordinates, distances, or angles: the input to the problem is the abstract network graph. We develop an approach to computing winding numbers in networks that differs from all three mathematical approaches listed above.

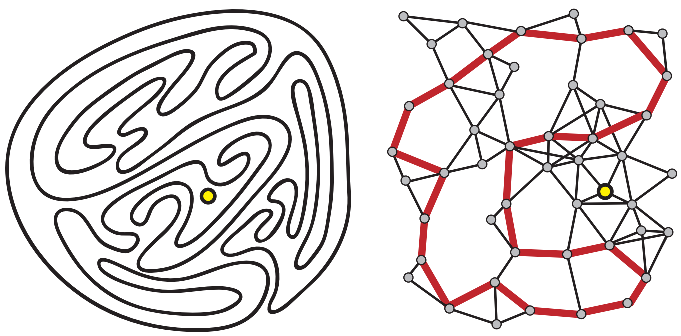

FIGURE 1. [left] Is the node inside the curve or outside? The network-theoretic version [right] can be challenging.

The are numerous motivations for this problem, mostly coming from sensor networks and multi-agent robotics. One concrete example would be a networked collection of accelerometers or acoustic sensors distributed in a planar domain and networked via proximity-based wireless signals. Given a certain node x_0 which registers an important reading (an alarm), one problem relevant to security applications is to determine whether the alarm has occurred within a region of particular importance whose perimeter is defined by a cycle in the network. Or, in environmental monitoring, one might wish to determine whether an animal (tagged with an RFID chip) is presently within a certain electronic 'fence' represented by a cycle in the monitoring network.

If one has sufficient data to localize nodes — that is, to determine local coordinates — then all such problems about winding numbers are trivial to solve. The assumption of coordinatized nodes is reasonable for systems with stationary intentionally place nodes. Such would include systems of video cameras or sensors mounted on fixed towers. However, in the case of nodes which are distributed in an unpredictable and non-uniform manner, or in which the nodes are mobile, then localized nodes are no longer *a priori* natural. Robotics, in particular, presents a natural setting in which mobile devices communicating via an ad hoc wireless network can provide localization challenges.

1.1. Related Work. There is a substantial and growing literature on geometric properties of ad hoc networks in which localization is weakened or not assumed at all. The recent work on routing without localization initiated by [**21**] uses a heat-flow to determine virtual coordinates for a non-localized network for applications to weighted routing problems. In many cases [**21, 10**] a set of known landmarks is used to estimate system geometry. All these methods are effective, but, with a few exceptions [**15**], non-rigorous. Recent work of Fekete et al. [**12**] gives a distributed algorithm for rigorous topology exploration and boundary detection: the algorithm is complete when the nodes are sufficiently dense.

There is a large body of work on coarse distance estimation in ad hoc networks augmented with angular data in the form of the angle of separation between a node's neighbors. This arises in the the paper [**9**], which uses a network graph along with exact angular measures of neighbors to detect holes in the physical network and perform routing. The work detailed in [**20**] gives criteria for ensuring coverage in a sensor network using bounds on separation angles among neighbors.

The classical work on *rotation systems* — planar graphs encoded with cyclic ordering of neighbors — would appear to be very relevant (see, e.g., [**8**] and subsequent work). However, the graph theory literature consistently assumes that the graphs in question are (1) embedded on an oriented surface, and (2) non-geometric, in the sense that there is no information about bounds on lengths of edges. In this paper, the entire difficulty lies in the fact that one begins with the abstract network graph and must deduce its canonical physical projection to the plane. This projection is *not* necessarily an embedding, and one must use constraints on the geometry of the graph (all edges are of at most unit length in the plane) to compensate for the lack of information about node placement.

As a problem in computational geometry, networks with only proximity measurements arise in the literature on *unit disc graphs*: abstract graphs whose vertices correspond to a set of nodes in the plane and whose edges are determined by nodes within unit distance. Clearly, not all graphs are realizable as a unit disc graph. Recognizing whether a graph is a realizable unit disc graph is NP-complete [**3**]. It follows that finding some embedding of an abstract unit disc graph into the plane for which the graph is the unit disc proximity network is also NP hard. Even finding an 'approximate' embedding which realizes a unit disc graph up to local errors is NP hard [**17**]. But, using angular data, [**4**] gives an algorithm for finding a realization of a spanner of the unit disc graph, which enables one to compute virtual coordinates and approximate some locations.

This paper is part of a growing body of work in topological techniques for multi-agent systems, including coordinated robots and sensor networks. Contemporary topology is finding applications in several robotics contexts, including motion planning [**11**], pursuit-evasion problems [**2**], programmable self-assembly [**16**], and reconfiguration [**1**]. Very recently, algebraic topology has been recognized as a novel tool for coverage, hole repair, and pursuit problems in sensor and ad hoc networks [**5, 6, 7**]. It is this perspective that inspires this paper.

1.2. Outline. The goal of this paper is to delineate when one can and cannot compute winding number information. For simplicity, we choose to restrict attention to the problem of *separation*, that is, whether the winding number is zero or nonzero.

There are many parameters one can vary in answering the feasibility question. Our parameter of choice is, roughly speaking, the *sensing complexity* — the amount of sensing capability required. For example, we show in §3 that separation is solvable given the ability to sense cyclic orientation of neighbors. In §4 we consider uncertainty in cyclic orientation data, and demonstrate that there are certain models of uncertainty which do not diminish the ability to compute winding numbers.

The results are complete, and require no nondegeneracy assumptions. However, it is true that the existence of an algorithm which works for 'typical' or 'sufficiently dense' graphs requires a much lower sensing complexity. In §5 we present an algorithm for systems which possess no cyclic orientation data whatsoever. The algorithm (one of many possible) is non-complete and provides certificates for separation. We close in §6 with open questions.

2. Background

2.1. Assumptions. The following assumptions will be in force throughout this paper:

- **P** (Planar) Nodes with unique labels lie in the Euclidean plane $\mathbb{R}^2$.
- **N** (Network) Nodes form the vertices of a connected unit disc network graph Γ of sufficiently large diameter.

This is the only information available to the network. There are no coordinates, no node localization, and no assumptions about node density or distribution other than sufficient extent. Assumptions **P** and **N** imply that nodes can broadcast their unique IDs and these can be detected by any neighboring nodes within unit distance. This creates a network graph whose vertices correspond to the labeled nodes and whose edges correspond to communication links. There is no metric information encoded in an edge beyond the coarse datum that the distance between the nodes in the plane is no more than one.

Definition 2.1. The *hop distance* on a network graph Γ, denoted $d(\cdot,\cdot)$, is the metric on nodes $V(\Gamma)$ given by the number of edges in a (combinatorially) shortest path in Γ. A *k-hop neighborhood* of a subgraph $\Gamma' \subset \Gamma$, denoted $\mathcal{N}^k(\Gamma')$, is the set $\{x \in V(\Gamma) : d(x, \Gamma') \leq k\}$.

We will also have occasion to assume the following:

- **O** (Ordering) Each node can determine the clockwise cyclic ordering of its neighbors in the plane.

Assumption **O** means that each node can perform a cyclic "sweep" of its neighborhood and determine the cyclic order in which neighbors appear. Absolute angular coordinates are assumed unknown and thus the ordering is defined up to a cyclic permutation. There is also no precise relative angular data assigned to the ordering: an oriented pair of neighbors may form an arbitrary (nonzero) angle with the central node x_0 without changing the angular ordering.

This type of coarse angular data is a well-studied subtopic of topological graph theory (where it is called a *rotation system* [**8**]). It is not too uncommon in robotics contexts, especially in primitive landmark vision systems, radar networks, and robots with gap sensors.

We assume that each node's cyclic ordering data is consistent: that each nodes sweeps its neighbors in a fixed orientation, either clockwise or counterclockwise. It is not necessary to know what this orientation is, only that it is consistent among all nodes.

Assumption **O** is equivalent to the ability to compute the following index:

Definition 2.2. Let x_0 be a node and $(x_i)_1^3$ be an ordered triple of distinct nodes in $\mathcal{N}^1(x_0)$. Define the index $\mathcal{I}_{x_0}(x_1, x_2, x_3)$ to be $+1$ if the ordering (x_1, x_2, x_3) agrees with the cyclic ordering data at x_0. If the cyclic ordering data does not agree, set $\mathcal{I}_{x_0}(x_1, x_2, x_3) = -1$.

REMARK 2.3. This index has a homological interpretation. The choice of uniform 'direction' for the cyclic orientation of neighbors is really a choice of orientation for $\mathbb{R}^2$, hence, a consistent generator for the local homology $H_2(\mathbb{R}^2, \mathbb{R}^2 - x; \mathbb{Z})$ for each $x \in \mathbb{R}^2$. The ordered triple (x_1, x_2, x_3) determines a generator for the local homology $H_2(\mathbb{R}^2, \mathbb{R}^2 - x_0; \mathbb{Z})$ as follows: project $\mathbb{R}^2 - x_0$ radially to obtain a unit circle S^1 with $H_2(\mathbb{R}^2, \mathbb{R}^2 - x_0) \cong H_2(\mathbb{R}^2 - x_0) \cong H_1(S^1)$. The sequence (x_1, x_2, x_3) projects to three distinct points in S^1 and determines a loop in this S^1 by concatenation of simple paths in the order specified. The index $\mathcal{I}_{x_0}(x_1, x_2, x_3)$ is equal to the value of this local homology class in the preferred generator determined by the orientation class.

For the present, we assume that nodes are in a 'general position' so as to possess a positive lower bound on angles between neighbors. This is, of course, completely unrealistic in practice. In §5, we abandon the assumption on perfect angular ordering information. A primary result of this paper is the demonstration that there is a surprisingly large tolerance for angular ordering blindness.

2.2. Problem statement. The difficulties of this problem lie in the relationship between the abstract network graph and its projection into the plane. The input data for the problem is the network graph Γ. When Assumption **O** is in place, the graph has vertices augmented with the cyclic ordering type of its immediate neighbors.

DEFINITION 2.4. The projection map $\Gamma \mapsto \overline{\Gamma} \subset \mathbb{R}^2$ maps vertices of Γ to the position of the corresponding node in the Euclidean plane and edges of Γ to the line segment connecting the nodes.

We solve the following problem concerning winding numbers of cycles:

Separation: Given a cycle $\mathcal{L}$ in the network graph Γ and a node $x_0 \in V(\Gamma)$ which is disjoint from the nodes of $\mathcal{L}$, determine whether the projected cycle $\overline{\mathcal{L}}$ surrounds x_0.

The image of the cycle $\overline{\mathcal{L}}$ in the plane is a closed piecewise-linear curve. If the curve is *simple* (that is, not self-intersecting), then the Jordan Curve Theorem implies that the cycle separates the plane into two connected components, only one of which is bounded. We will restrict attention to simple cycles, using network criteria to satisfy this condition (Corollary 2.7).

2.3. Basic lemmas. The separation problem is less complicated when the projected cycle $\overline{\mathcal{L}}$ is a simple closed curve in $\mathbb{R}^2$. The easiest way to guarantee such a cycle is to choose a cycle which is 'minimal' with respect to communication between nodes.

DEFINITION 2.5. For any subgraph $\Gamma' \subset \Gamma$, let $\langle \Gamma' \rangle$ denote the maximal subgraph of Γ spanned by the vertices of Γ'. Say that Γ' is *chord-free* if $\langle \Gamma' \rangle = \Gamma'$.

The simplest criterion for a cycle $\mathcal{L}$ to have a simple projection to the plane is that $\langle \mathcal{L} \rangle = \mathcal{L}$. The following lemma is both trivial and well-known [**12, 18, 5**].

LEMMA 2.6. *If the projections of two edges of a unit disc graph Γ intersect in $\mathbb{R}^2$, then these span a subgraph of Γ containing a cycle of three edges.*

Proof. Consider the two edges whose projection forms an 'X' in the plane with four segments. At least two of these segments must have length no greater than one-half, since each full edge is of at most unit length. Two applications of the triangle inequality to the shorter of these segments and its neighbors completes the proof. □

Corollary 2.7. Any path (or cycle) $\mathcal{P}$ in a unit disc graph Γ satisfying $\langle\mathcal{P}\rangle = \mathcal{P}$ has image $\overline{\mathcal{P}}$ a non self-intersecting (closed) curve in $\mathbb{R}^2$.

Later in the paper, we will have need to estimate angular data from the unit disc graph. The following proofs are trivial but included for completeness (cf. [**9**]):

Lemma 2.8. *Let* L *be a connected unit disc graph on three vertices with one vertex of degree two. Then the angle at the central vertex of* L *in* $\mathbb{R}^2$ *exceeds* $\pi/3$*.*

Proof. Assume the edge lengths are $a, b \leq 1$, and assume the central angle does not exceed $\pi/3$. Then, by the Law of Cosines, the distance c between the two endpoints of L satisfies

$$c^2 \leq a^2 + b^2 - ab \leq a^2 + b^2 - \min\{a^2, b^2\} \leq 1.$$

Contradiction. □

Lemma 2.9. *Let* Y *be a connected unit disc graph on four vertices with one vertex of degree three. Then all three pairwise angles at the central vertex of* Y *in* $\mathbb{R}^2$ *exceed* $\pi/3$*.*

Proof. Apply Lemma 2.8 three times. □

Lemma 2.10. *Let* $\top$ *be a connected unit disc graph with four vertices and one cycle of three edges, as in Fig. 2. If the projection* $\overline{\top}$ *is not embedded, then, at the degree four vertex, both angles of the fourth edge with the triangle are at least* $\pi/3$*. In addition, the two acute angles of this triangle are each strictly less than* $\pi/6$*.*

Proof. We coordinatize the problem as in Fig. 2[left], with the vertex associated to the obtuse angle located at the point $(0, 1)$. The two other vertices of the triangle must lie outside the closed unit disc in the plane. Upon minimizing the obtuse angle, the 4-dimensional space of locations of these vertices reduces to a 1-dimensional space, where all three vertices lie on the unit circle and the two acute angle vertices are a unit distance apart. Straightforward optimization reveals that the smallest obtuse and largest acute angles arise in the limit of the anti-symmetric configuration as in Fig. 2[right]. □

3. Computing winding number with precise angular ordering

In this section, we consider the separation problem for node x_0 and cycle $\mathcal{L}$ in a network Γ satisfying Assumptions **P**, **N**, and **O**.

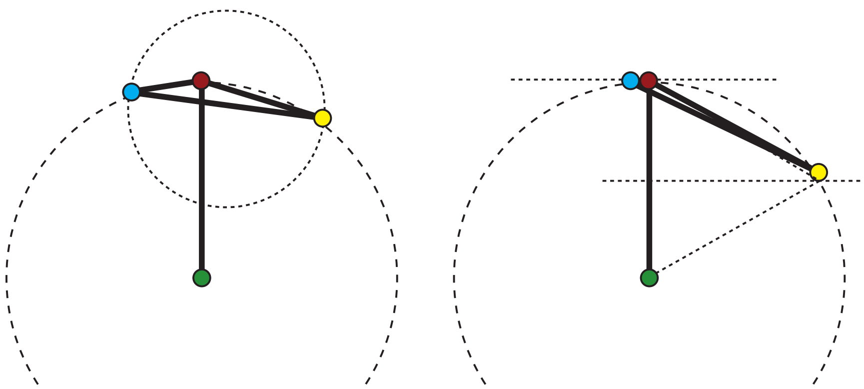

FIGURE 2. [left] The smallest obtuse and largest acute angles for the triangle of $\overline{\top}$ arise when all three vertices of the triangle lie on a unit circle. [right] In the critical case, these angles tend to $5\pi/6$, $\pi/6$, and 0 respectively.

3.1. Restrictions. We impose the restriction that $d(x_0, \mathcal{L}) > 1$, meaning that the shortest path from x_0 to a vertex of $\mathcal{L}$ in Γ requires more than one 'hop' or edge. Both this and Assumption **O** are necessary for a complete solution. Critical examples are illustrated in Fig. 3 of labeled graphs in $\mathbb{R}^2$ with isomorphic network graphs and identical cyclic orientation data. The target node x_0 lies on opposite sides of the cycles illustrated. These examples can be easily modified to possess infinite diameter, in accordance with Assumption **N**.

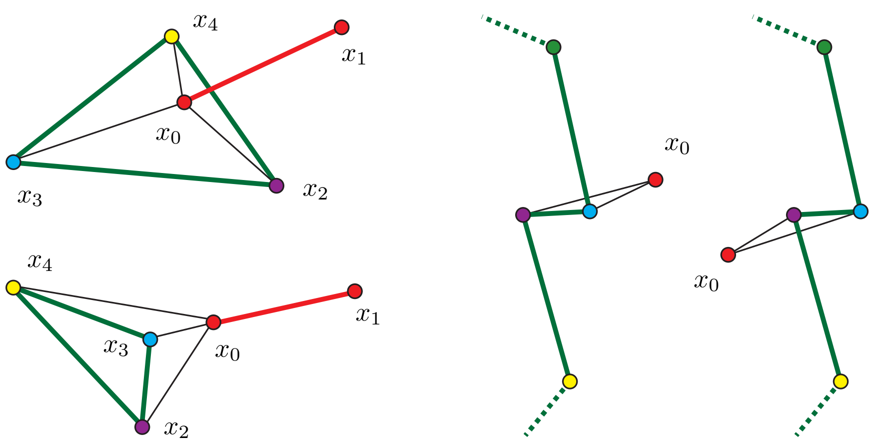

FIGURE 3. Two pairs of network graphs with identical network and cyclic orientation data, but for which the node x_0 lies on different sides of the illustrated cycle. That on the right demonstrates that cyclic orientation data can be incomplete even for arbitrarily long cycles $\mathcal{L}$.

3.2. Algorithm. In differential topology, the way one decides whether a loop in the plane encloses a point is to choose a path from the point which terminates sufficiently far from the starting point as to be definitely outside the loop. For a 'generic' choice of such a path, the path and the loop intersect transversally (i.e., without tangencies), and the number of intersection points counted mod 2 is zero if and only if the loop does not surround the point [**14**].

The obvious generalization of this strategy is to choose any path $\mathcal{P}$ in Γ from x_0 to a terminal point x_∞ which is sufficiently far away from $\mathcal{L}$ to guarantee that it is outside the cycle in $\mathbb{R}^2$, and then count intersections. However, this counting is not always easy or even possible. The inspiration for our method is nearly opposite to that coming from differential topology. Instead of trying to force intersections to be a discrete set of points, one thinks of manipulating the path so as to *maximize* the amount of intersection with the cycle with the result of having a single connected component in the intersection. Then, one computes whether the endpoints of $\mathcal{P}$ lie on the same side of $\overline{\mathcal{L}}$. This is what we do, using the orientation data in the final step.

Orient the cycle $\mathcal{L}$ in Γ and order the nodes (ℓ_i) of $\mathcal{L}$ cyclically. Choose a node x_∞ sufficiently far from $\mathcal{L}$ in Γ (see Lemma 3.1). Generate chord-free paths $\mathcal{P}_0$ and $\mathcal{P}_\infty$ from x_0 and x_∞ respectively to nodes on $\mathcal{L}$. The projections of these paths to $\mathbb{R}^2$ are not self-intersecting, and can only intersect $\overline{\mathcal{L}}$ at most once at the last segment of the path.

The crucial step is to determine whether the projected paths $\overline{\mathcal{P}}_0$ and $\overline{\mathcal{P}}_\infty$ lie on the same side of $\overline{\mathcal{L}}$ or different sides. In the simplest configuration, the final point on a path is connected to $\mathcal{L}$ in Γ by only one edge, as in Fig. 4[left]. The angular orientation data then suffices to determine on which side of $\overline{\mathcal{L}}$ the path lies. However, the situation may be more complicated, as in Fig. 4[right]. A more subtle manipulation of orientation data is required in this case. Details are presented in Algorithm IndexCheck.

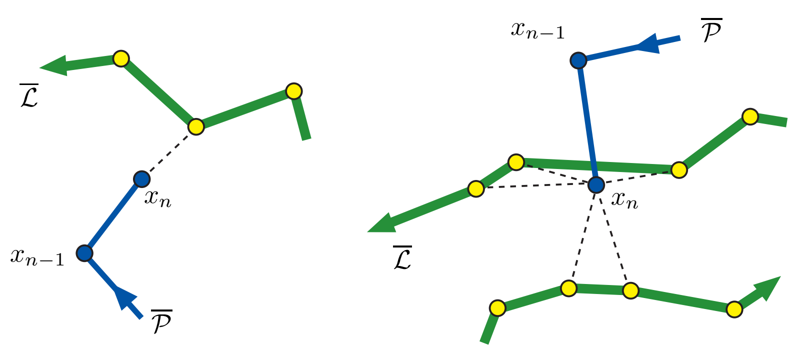

FIGURE 4. Determining whether an oriented path with terminal node x_n approaches the projected oriented cycle $\overline{\mathcal{L}}$ from the left or from the right can be simple [left] or complicated [right] depending on the number of communication links between x_n and $\mathcal{L}$.

3.3. Proofs.

Algorithm 1 $\mathcal{I} = \mathsf{IndexCheck}(x_0, \mathcal{L}, \Gamma)$

Require: $\Gamma = (V, E)$ is a graph satisfying **P**, **N**, and **O**
Require: $x_0 \in V(\Gamma)$, $\mathcal{L}$ is an oriented cycle of Γ, $\langle \mathcal{L} \rangle = \mathcal{L}$, and $d(x_0, \mathcal{L}) > 1$
choose a path $\mathcal{P}$ in $\Gamma - \mathcal{N}^1(\mathcal{L})$ with $\langle \mathcal{P} \rangle = \mathcal{P}$ and $|\mathcal{P}| > 2|\mathcal{L}|^2/\pi^2$
choose x_∞ any vertex on $\mathcal{P}$
return $\mathcal{I} \Leftarrow \mathsf{IndexVertexLoop}(x_0, \mathcal{L}, \Gamma) - \mathsf{IndexVertexLoop}(x_\infty, \mathcal{L}, \Gamma)$

Algorithm 2 $\mathcal{I} = \mathsf{IndexVertexLoop}(y, \mathcal{L}, \Gamma)$

Require: $\Gamma = (V, E)$ is a graph satisfying **P**, **N**, and **O**
Require: $y \in V(\Gamma)$, $\mathcal{L} = (\ell_i)$ is an oriented cycle of Γ, $\langle \mathcal{L} \rangle = \mathcal{L}$, and $d(y, \mathcal{L}) > 1$.
choose a path $\mathcal{P} = (y_i)_0^n$ in G with $y_0 = y$, $\langle \mathcal{P} \rangle = \mathcal{P}$, and $d(y_i, \mathcal{L}) = 1$ iff $i = n$.
if at y_n, y_{n-1} separates ℓ_j and ℓ_{j+1} in the cyclic order for some j **then**
    return $\mathcal{I} \Leftarrow \mathcal{I}_{\ell_j}(\ell_{j-1}, y_n, \ell_{j+1}) \cdot \mathcal{I}_{\ell_{j+1}}(\ell_{j+2}, \ell_j, y_n) \cdot \mathcal{I}_{y_n}(y_{n-1}, \ell_{j+1}, \ell_j)$
else
    choose any ℓ_j with $d(\ell_j, y_n) = 1$.
    return $\mathcal{I} \Leftarrow \mathcal{I}_{\ell_j}(y_n, \ell_{j+1}, \ell_{j-1})$
end if

3.3.1. *Finding the outside.*

Lemma 3.1. *Let $\mathcal{L}$ be a chord-free cycle in Γ and $\mathcal{P}$ a chord-free path in the complement of the 1-hop neighborhood of $\mathcal{L}$ of length at least $2|\mathcal{L}|^2/\pi^2$. Then $\overline{\mathcal{P}}$ lies in the region of $\mathbb{R}^2$ outside of $\overline{\mathcal{L}}$.*

Proof. The Isoperimetric Inequality says that the area A enclosed by the simple closed curve $\overline{\mathcal{L}}$ in $\mathbb{R}^2$ is bounded above by $1/(4\pi)$ times the square of the perimeter of $\overline{\mathcal{L}}$. This perimeter is bounded above by $|\mathcal{L}|$. As $\mathcal{P}$ is a chord-free path, placing a ball of radius $\frac{1}{2}$ about every other vertex of $\mathcal{P}$ yields disjoint balls of total area $\frac{1}{8}\pi|\mathcal{P}|$. A path avoiding the 1-hop neighborhood of the chord-free cycle $\mathcal{L}$ satisfies $\overline{\mathcal{P}} \cap \overline{\mathcal{L}} = \emptyset$ thanks to Lemma 2.6. Any such path of length at least $2|\mathcal{L}|^2/\pi^2$ violates the area constraint: $\overline{\mathcal{P}}$ is thus not surrounded by $\overline{\mathcal{L}}$. □

Better constants may be estimated with finer techniques from packing theory; however, a sharp lower bound in length must be quadratic in $|\mathcal{L}|$, since a chord-free path in the interior of $\overline{\mathcal{L}}$ can fill up the area bound by $\overline{\mathcal{L}}$, which can be quadratic in the perimeter.

3.3.2. *Determining cyclic orientation.* We next show how to pass local cyclic orientation data to larger neighborhoods. For the following, let $\triangle$ denote a unit disc graph having an oriented cycle (x_1, x_2, x_3), with each such 'inner node' x_i connected to one 'outer node' y_i (as in Fig. 5). For such a graph, we show how to compute the cyclic orientation of the three outer nodes (y_1, y_2, y_3). This cyclic orientation is not obviously well-defined, since the outer nodes could presumably be colinear in $\mathbb{R}^2$. However, no outer node lies within the triangle T defined as the convex hull of the inner nodes. Thus, the cyclic triple (y_1, y_2, y_3) defines a cyclic triple in the complement of T and thus a generator for $H_1(\partial T) \cong H_2(\mathbb{R}^2, \mathbb{R}^2 - T) \cong H_2(\mathbb{R}^2, \mathbb{R}^2 - x_i)$, where ∂T denotes the boundary of T.

Lemma 3.2. *For $\triangle$ a unit disc graph as above, the cyclic orientation of the outer nodes $(y_i)_1^3$ with respect to any inner node is equal to $\mathcal{I}_{x_j}(x_{j-1}, y_j, x_{j+1})$, where j is such that $\angle x_{j-1}x_jx_{j+1}$ is maximal.*

Recall that subscript indices are taken cyclically.

Proof. Let T denote the (closed, possibly degenerate) triangle in $\mathbb{R}^2$ given by the convex hull of the inner nodes. By definition, the cyclic orientation of the outer nodes $(y_i)_1^3$ can be computed by projecting each y_i to the nearest point on the boundary ∂T of the triangle T, the projection being performed along the line segments $\overline{y_ix_i}$.

Since there are no edges between the outer nodes $\{y_i\}_1^3$, Lemma 2.6 implies that at most one edge $\overline{y_ix_i}$ can cross T in $\mathbb{R}^2$. Lemma 2.9 implies that, if an edge $\overline{y_ix_i}$ crosses T in $\mathbb{R}^2$, then the angle $\angle x_{i-1}x_ix_{i+1}$ subtended by T at x_i is at least $2\pi/3$, and it is therefore the largest angle of T.

Assume that x_j is the inner node of largest angle in T. Since neither $\overline{y_{j-1}x_{j-1}}$ nor $\overline{y_{j+1}x_{j+1}}$ crosses T, the projection of y_{j-1} and y_{j+1} to ∂T is x_{j-1} and x_{j+1} respectively. Thus, the cyclic orientation of the outer nodes is given as $\mathcal{I}_{x_j}(x_{j-1}, y_j, x_{j+1})$. □

Lemma 3.3. *For $\triangle$ a unit disc graph as above, the cyclic orientation of the outer nodes $(y_i)_1^3$ with respect to any inner node is equal to*

$$\prod_{i=1}^{3} \mathcal{I}_{x_i}(x_{i-1}, y_i, x_{i+1}). \tag{3.1}$$

Proof. If $\overline{\triangle}$ is embedded, then all three indices are equal and the result holds. If $\overline{\triangle}$ is not embedded, then the indices at the two acute-angle inner nodes are equal, and the sign of the product of their indices is $+1$. Lemma 3.2 completes the proof. □

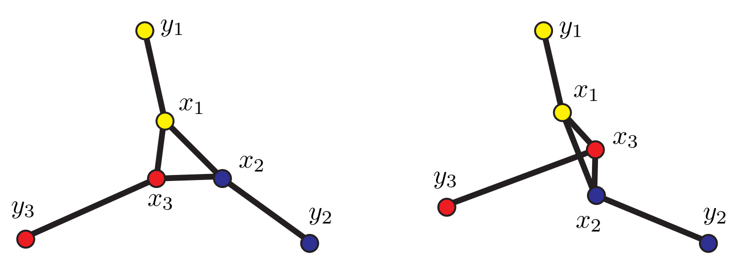

Figure 5. The cyclic orientation of the outer nodes can be derived from the indices of the inner nodes in the above cases by computing the product of indices.

3.3.3. *Correctness and completeness.*

Theorem 3.4. *In any network satisfying Assumptions* **P**, **N**, *and* **O**, *let $\mathcal{L}$ be a cycle satisfying $\langle\mathcal{L}\rangle = \mathcal{L}$ and x_0 a node with $d(x_0, \mathcal{L}) > 1$. Algorithm* IndexCheck *returns $\mathcal{I} = 0$ iff the winding number of $\overline{\mathcal{L}}$ about the node $x_0 \in \mathbb{R}^2$ vanishes.*

PROOF. Corollary 2.7 implies that $\overline{\mathcal{L}}$ is embedded in $\mathbb{R}^2$. This simple closed curve separates the plane in two connected components, thanks to the Jordan Curve Theorem. Fixing an orientation on $\mathcal{L}$ induces an (unknown) orientation on $\overline{\mathcal{L}}$.

Choose a chord-free path $\mathcal{P}_0 = \{x_i\}_0^n$ from x_0 to x_n with $d(x_i, \mathcal{L}) = 1$ iff $i = n$. Via Corollary 2.7, the image of this path, $\overline{\mathcal{P}}_0$, is simple and the restriction of this path to the subpath between nodes x_0 and x_{n-1} lies entirely on one side of $\overline{\mathcal{L}}$ in $\mathbb{R}^2$. The edge from x_{n-1} to x_n may or may not cross $\overline{\mathcal{L}}$.

If there do not exist consecutive cycle nodes ℓ_j, ℓ_{j+1} incident to x_n, then choose any ℓ_j incident to x_n. In this case, the 'Y' graph connecting ℓ_j to x_n, ℓ_{j-1}, and ℓ_{j+1} has no additional connections between outer nodes, and the index $\mathcal{I}_{\ell_j}(x_n, \ell_{j-1}, \ell_{j+1})$ indicates on which side of $\overline{\mathcal{L}}$ the node x_n (hence x_0) lies.

If, however, consecutive cycle neighbors exist, one argues that the subgraph $\triangle$ consisting of the cycle $(x_n, \ell_j, \ell_{j+1})$ and the connections of these inner nodes to respective outer nodes $(x_{n-1}, \ell_{j-1}, \ell_{j+2})$ has image $\overline{\triangle}$ as in Fig. 5. A more complicated embedding cannot appear thanks to repeated application of Lemma 2.6[1]. Thanks to Lemma 3.3, the product of the three indices $\mathcal{I}_{\ell_j}(\ell_{j-1}, x_n, \ell_{j+1})$, $\mathcal{I}_{\ell_{j+1}}(\ell_{j+2}, \ell_j, x_n)$, and $\mathcal{I}_{x_n}(x_{n-1}, \ell_{j+1}, \ell_j)$ gives the cyclic orientation of the ordered triple $(x_{n-1}, \ell_{j-1}, \ell_{j+2})$ of outer nodes. This tells whether x_{n-1} (and thus x_0) lies to the 'left' or to the 'right' of the embedded segment $(\ell_i)_{j-1}^{j+2}$ of $\overline{\mathcal{L}}$. However, it is possible that $\overline{\mathcal{L}}$ doubles back and crosses the segment between x_{n-1} and x_n, as in Fig. 4[right]. In this case, one needs to be sure to use the subgraph $\triangle$ generated by consecutive nodes $(\ell_i)_{j-1}^{j+2}$ where, from the vantage of x_n, x_{n-1} separates ℓ_j from ℓ_{j+1}, and no other ℓ_i separates.

There is an ambiguity in $\mathcal{I}$ resulting from the fact that we do not know if the orientation on $\overline{\mathcal{L}}$ is clockwise or counterclockwise; thus we do not know which sign for $\mathcal{I}$ (i.e., the 'left' or the 'right' side of $\overline{\mathcal{L}}$) corresponds to the bounded component of $\mathbb{R}^2 - \overline{\mathcal{L}}$. To determine this, choose any node x_∞ on a chord-free path in $\Gamma - \mathcal{N}^1(\mathcal{L})$ of length $2|\mathcal{L}|^2/\pi^2$. From Lemma 3.1, x_∞ lies within the unbounded component of $\mathbb{R}^2 - \overline{\mathcal{L}}$. That one can choose such a node and a chord-free path $\mathcal{P}_\infty$ from $\mathcal{L}$ to x_∞ is possible thanks to the diameter condition of Assumption **N**. Computing the index of x_∞ with respect to $\mathcal{L}$ and comparing it to that of x_0 as in IndexCheck determines whether x_0 and x_∞ are on the same or different sides of $\overline{\mathcal{L}}$. □

4. Angles and uncertainty

It is unrealistic to expect that Assumption **O** is a valid sensor modality for all circumstances. This assumption is motivated by robotic navigation using low-resolution omnidirectional cameras to track the angular ordering of a set of landmarks within visible range. Certainly, for vision-based angular data, uncertainties in the cyclic ordering will be unavoidable in configurations where two neighboring nodes are nearly colinear with the point-of-view.

4.1. Models of uncertainty. There are several possible ways to account for uncertainty in angular cyclic orientation. To specify these models, we consider the

[1]It may be possible that some inner node is connected to more than one outer node, as seen in Fig. 4[right]. However, since there are no edges between outer nodes, and no outer node can lie within the convex hull of the inner nodes, Lemma 3.3 still holds.

cyclic orientation sensor at a node x_0 as a function on ordered triples of neighbors, as in Definition 2.2.

- **U1** (Single threshold) Any triple of neighbors can be cyclically ordered if none of the three pairwise angles is below the fixed threshold $\epsilon_{\max}$. We do not assume that the cyclic ordering measurement always fails whenever an angle is below $\epsilon_{\max}$, but rather that the reading either returns a true angular reading or an empty (i.e., uncertain) reading.
- **U2** (Dual threshold) Any triple of neighbors can be cyclically ordered if none of the three pairwise angles is below the fixed threshold $\epsilon_{\max}$. Furthermore, the cyclic ordering measurement always fails whenever some angle is below a smaller threshold $\epsilon_{\min} < \epsilon_{\max}$. All readings for systems with an angle in $[\epsilon_{\min}, \epsilon_{\max}]$ may return either true (certain) or empty (uncertain) results.

REMARK 4.1. One way to unify these (and many other) models is to consider a functional form of uncertainty. Fix a central node x_0 at which attention is focused and identify the unit tangent bundle to x_0 in $\mathbb{R}^2$ with a round circle S^1 of unit length. An ordered set of three neighbors of x_0 yields a point in $S^1 \times S^1 \times S^1$ given by the three angles at which these neighbors lie. (Of course, the sensor modality in place cannot detect the actual angles.) A *certainty function* could take the form of a multi-valued map $\Psi : S^1 \times S^1 \times S^1 \rightrightarrows [0, 1]$ which encodes the support of the range of certainties that a sensor may experience. If Ψ takes on the value 1, this means that the node x_0 can read the true cyclic orientation with full certainty. If Ψ takes on the value 0, this counts as an empty reading: there is no information on what the cyclic ordering might be. Other readings can be given a 'ranking' of certainty. Assumption **U1** means that $\Psi = \{1\}$ when all angles exceed $\epsilon_{\max}$. One can mimic **(U2)** by assuming that, in addition, $1 \notin \Psi$ when some angles are within $\epsilon_{\min}$.

4.2. Bounds on angles. The surprising fact is that for $\epsilon_{\max}$ as large as $\pi/3$, it is often possible to rigorously determine winding numbers. Some choices of Γ and $\mathcal{L}$ present too much uncertainty in the case of a single threshold uncertainty, but criteria for knowing when you can compute winding numbers are possible. For dual threshold models of certainty, we show that Algorithm IndexLoop is complete for $\pi/3 \geq \epsilon_{\max} > \epsilon_{\min} \geq \pi/6$. It is thus necessary to be able to distinguish sufficiently small angles from sufficiently large ones in order to use partial information to obtain winding data.

4.3. Case analysis for threshold uncertainty. We consider adapting Algorithm IndexVertexLoop to a network satisfying Assumptions **P**, **N**, and **U1**, with $\epsilon_{\max} \leq \pi/3$. Let $\mathcal{L}$ be a chord-free cycle and x a node with $d(x, \mathcal{L}) > 1$. Choose a path as in step 1: of the algorithm.

Case 1: Assume that x_n is not within one hop of a consecutive pair of nodes of $\mathcal{L}$ Choose any isolated incident cycle node ℓ_j of $\mathcal{L}$. Since $\mathcal{L}$ is chord-free, Lemma 2.9 implies that all three angles at ℓ_j to x_n, ℓ_{j-1}, and ℓ_{j+1} are greater than $\pi/3$ and thus $\epsilon_{\max}$. Therefore, by Lemma 2.9 $\mathcal{I}_{\ell_j}(x_n, \ell_{j-1}, \ell_{j+1})$ is certain and Algorithm IndexVertexLoop is successful.

Case 2: If x_n is connected in Γ to more than one incident node of $\mathcal{L}$, then there is a subgraph $\triangle$ of the form in Fig. 5. If there is precisely one such graph, proceed to the subcases below. However, there may be several such subgraphs available, precisely one whose image in $\mathbb{R}^2$ is not embedded: recall Fig. 4[right]. To apply Step 2: of IndexVertexLoop, we must determine which copy of $\triangle$ is non-embedded. We claim this is possible when $\epsilon_{\max} \leq \pi/3$.

Assume therefore that x_n is connected to two pairs of consecutive nodes of $\mathcal{L}$, say (ℓ_i, ℓ_{i+1}) and (ℓ_j, ℓ_{j+1}). (These may overlap.) It may be furthermore assumed from Lemma 2.10 that at least one of the indices $\mathcal{I}_{x_n}(x_{n-1}, \ell_i, \ell_{i+1})$ or $\mathcal{I}_{x_n}(x_{n-1}, \ell_j, \ell_{j+1})$ is certain. If both are certain, then, since $\mathcal{L}$ is chord-free, some pair of these four nodes, say ℓ_i and ℓ_{j+1} are not neighbors. Thus, the set of nodes $(x_n, x_{n-1}, \ell_i, \ell_{j+1})$ span a subgraph satisfying Lemma 2.9, and the index $\mathcal{I}_{x_n}(\ell_{j+1}, \ell_i, \ell_{i+1})$ is certain. This is sufficient to construct the entire cyclic ordering of $(\ell_i, \ell_{i+1}, \ell_j, \ell_{j+1}, x_{n-1})$ about x_n, which, in turn, identifies the appropriate copy of $\triangle$ in Γ.

Assume from now on that the appropriate copy of $\triangle$ in Γ has been identified. We consider cases based on how many of these three interior nodes admit a fully certain index.

Case 2.0: No indices are certain. This case is impossible, as argued above in Case 2.

Case 2.1: One index is certain. We claim that this single certain index is equal to the full index of $\mathcal{P}$ with respect to $\mathcal{L}$. This breaks into two cases, based on whether $\overline{\triangle}$ is embedded or not (cf. Fig. 5). In the embedded case, each vertex has the same index, and any one is the same as the triple product. In the non-embedded case, if only one index exists, it must be that associated to the vertex of largest angle. By Lemma 3.2, this index is equal to the triple-product and correctly computes angular ordering.

Case 2.2.1: Two indices are certain, and agree. One of these must be the index associated to the largest angle. By Lemma 3.2 either certain index correctly computes the triple index and thus angular ordering.

Case 2.2.2: If two indices only are defined and the two certain indices differ, then we are in the case where $\overline{\triangle}$ is not embedded in the plane. However, if it is not possible to compute the third index and it is not possible to determine which of the two nodes has the larger subtended angle, then there is no information by which the index of this path can be determined. One may attempt to modify either $\mathcal{P}$ or $\mathcal{L}$ locally to remove the ambiguity, but there is no guarantee that this is possible for every Γ. Figure Fig. 6 gives an example of a graph which falls under this case, with zero and nonzero winding numbers returning exactly the same pair of certain indices.

This graph can be easily modified to satisfy the infinite-diameter requirement of Assumption **N**. We stress that in this (admittedly pathologically sparse) example, there is no recourse to using the homotopy invariance of winding number, or to choosing different paths to $\mathcal{L}$ in the hopes of reducing to another case.

Case 2.3: If all three indices are certain, there is no ambiguity, and the algorithm proceeds normally.

Thus, there is only one case in which uncertainty of type **U1** fails to solve the separation problem: this is Case 2.2.2 above. Ironically, having *more* uncertainty in that particular setting would lead to a certain solution to the winding problem. It

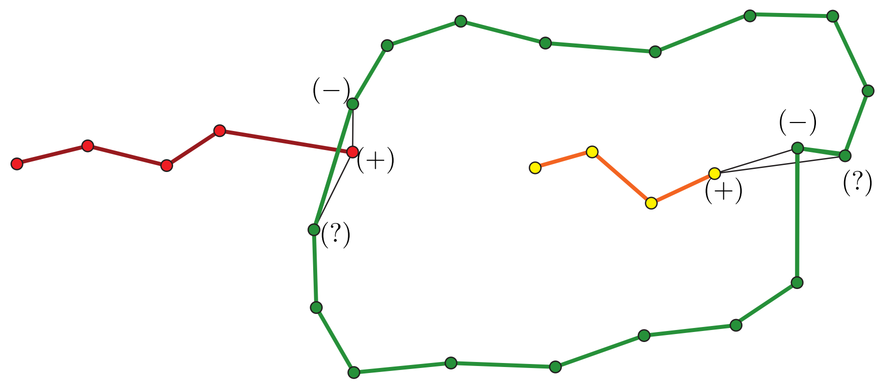

FIGURE 6. An example for which uncertainty in cyclic ordering of type **U1** leads to ambiguities in winding numbers.

is precisely when the system is inconsistent in its ability to distinguish small angles that ambiguity arises.

The above case analysis, combined with Lemma 2.10 shows that Algorithm IndexCheck is valid in a network replacing Assumption **O** with **U2**, where the certainty thresholds are $\epsilon_{\max} \leq \pi/3$ and $\epsilon_{\min} \geq \pi/6$:

THEOREM 4.2. *Consider a network satisfying Assumptions* **P**, **N**, *and* **U2** *with*

$$\frac{\pi}{3} \geq \epsilon_{\max} > \epsilon_{\min} \geq \frac{\pi}{6}.$$

Let $\mathcal{L}$ be a cycle satisfying $\langle\mathcal{L}\rangle = \mathcal{L}$ and x_0 a node with $d(x_0, \mathcal{L}) > 1$. Algorithm IndexCheck *returns $\mathcal{I} = 0$ iff the winding number of $\overline{\mathcal{L}}$ about the node $x_0 \in \mathbb{R}^2$ vanishes.*

5. Networks without cyclic orientation data

For systems which do not satisfy Assumption **O**, no solutions are possible which apply to arbitrary networks: it is easy to generate examples of very sparse graphs which can be embedded in the plane as a unit disc graph in multiple ways. There are non-complete algorithms which, upon successful termination, return rigorous winding number information. The 'flower' graphs of [**12**] provide one example of rigorous winding number computation without angular ordering, based on packing arguments for small graphs. We present a very different method for computing winding numbers based on simple plane topology. These methods are not applicable to all networks, but only to those which have some control over 'infinity.'

Consider a network satisfying **N**, **P**, and the following:

I There is a simply-connected domain $\mathcal{D} \subset \mathbb{R}^2$ and corresponding 'interior' graph Γ' whose vertices consist of all nodes of Γ in $\mathcal{D}$ and which satisfies $\overline{\Gamma'} \subset \mathcal{D}$.

We require additional restrictions. Assume that the inputs to the problem — the initial node x_0 and the cycle $\mathcal{L}$ both lie within this interior graph Γ'. Assumption **I** encodes winding number data in that $\partial\mathcal{D}$ has winding number ± 1 with respect to x_0. It is not necessary to know the precise geometry of $\mathcal{D}$ (cf. [**5**]). Furthermore, assume that the cycle $\mathcal{L}$ is at least R hops away from x, for some fixed $R > 1$.

Algorithm YWinding performs the following operations. From x_0, choose three chord-free paths $\{\mathcal{P}_i\}_1^3$ from x_0 to the 'exterior' graph $\Gamma - \Gamma'$ such that the 1-hop neighborhood of each $\mathcal{P}_i$ is disjoint from $\mathcal{P}_{i-1} \cup \mathcal{P}_{i+1}$ outside a neighborhood of x_0 (indices on the paths being computed cyclically). Of course, the existence of such paths is not guaranteed for all networks, and this is precisely where the algorithm fails to be complete for all networks. The algorithm sweeps along the cycle $\mathcal{L}$ and counts any arcs connecting $\mathcal{P}_i$ to $\mathcal{P}_{i+1}$ avoiding $\mathcal{P}_{i-1}$ for some fixed i. The obvious algebraic counting procedure on these arcs yields the degree of $\mathcal{L}$ in $\mathcal{D} - \{x_0\}$.

Algorithm 3 $\mathcal{I} = \mathsf{YWinding}(x_0, R, \mathcal{L}, \Gamma, \Gamma')$

Require: Γ satisfies **P**, **N**, and **I**
Require: x_0 is a vertex in Γ'
Require: $\mathcal{L}$ is a cycle in $\Gamma' - \mathcal{N}^R(x_0)$
1: $\{\mathcal{P}_i\}_1^3 \Leftarrow$ paths in Γ from x_0 to $\Gamma - \Gamma'$ satisfying $d(\mathcal{P}_i, \mathcal{P}_{j\neq i}) > 1$ outside $\mathcal{N}^R(x_0)$.
2: $\mathcal{I}^+ \Leftarrow \#\{$disjoint oriented paths in $\mathcal{L}$ from $\mathcal{N}^1(\mathcal{P}_1)$ to $\mathcal{N}^1(\mathcal{P}_2)$ avoiding $\mathcal{N}^1(\mathcal{P}_3)\}$
3: $\mathcal{I}^- \Leftarrow \#\{$disjoint oriented paths in $\mathcal{L}$ from $\mathcal{N}^1(\mathcal{P}_2)$ to $\mathcal{N}^1(\mathcal{P}_1)$ avoiding $\mathcal{N}^1(\mathcal{P}_3)\}$
4: $\mathcal{I} \Leftarrow |\mathcal{I}^+ - \mathcal{I}^-|$
5: return $\mathcal{I}$

THEOREM 5.1. *Consider a network satisfying Assumptions* **P** *and* **N**. *Let* $\mathcal{L}$ *be a cycle in* Γ *and* x_0 *a node with* $d(x_0, \mathcal{L}) > R > 1$. *Algorithm* IndexCheck *returns* $\mathcal{I}$ *equal to the absolute value of the winding number of* $\overline{\mathcal{L}}$ *about* $x_0 \in \mathbb{R}^2$.

PROOF. The result is transparent: we sketch a proof. Since the 'troika' of projected paths $\{\overline{\mathcal{P}_i}\}_1^3$ are pairwise disjoint, one can draw a ray from x_0 to $\partial\mathcal{D}$ in the component of $\mathcal{D} - \overline{\mathcal{P}_1 \cup \mathcal{P}_2}$ which does not contain $\overline{\mathcal{P}_3}$. The index $\mathcal{I}$ computed by the algorithm equals (up to a sign) the algebraic intersection number of this ray with $\overline{\mathcal{L}}$. □

REMARK 5.2. For a sufficiently dense network of nodes, Algorithm YWinding is complete, since, in such a network, one can find a subgraph Y in a C^0 neighborhood of a troika of straight rays based at x_0. This is an instance of the meta-theorem that any problem which has a solution in the 'smooth' category has a solution in the 'network' category, assuming a sufficiently high density of points. For example, the problem of determining the structure of a manifold by sampling it along a finite set of points has a solution if the sampling is sufficiently dense [**19**].

REMARK 5.3. We note that there are many other ways to compute winding data abandoning cyclic orientation data for some control over density. We claim that using a simplicial completion of the network graph to a *Rips complex* allows for the methods in the spirit of [**5**] to compute winding numbers under a 'homological' density assumption (namely, that the plane is covered by convex hulls of triples of nodes in pairwise communication).

6. Other scenarios

There is an extensive collection of related systems in which varying degrees of sensing capabilities act as a parameter. The minimal-sensing boundary for winding numbers — those critical threshold where there is just enough information from which to derive the answer rigorously — is an important and sometimes elusive object. We list a few interesting generalizations of the questions considered in this paper.

6.1. Global connectivity. Consider a network in which no distance information can be derived from the network. Specifically, assume that the network graph Γ is complete. Initially, this may seem to be an advantage, since, in the case of Assumption **O**, one has a huge amount of exploitable data. However, as illustrated in Fig. 3, cyclic ordering of neighbors in a complete graph does not necessarily fix winding number data. The recent work of Tovar et al. [**22**] shows how to extract winding number data from this setting when one can use a mobile node (i.e., an autonomous robot with navigational capabilities) to explore the network and track how the cyclic ordering of neighbors changes over time. When one does not have this capability, it is much harder to glean information about winding numbers.

6.2. Errors. In our treatment of measurement errors or ambiguities, we have focused exclusively on errors in one sensor reading only — the angular ordering. Even in this setting, we have considered only errors which are known for certain to be wrong/ambiguous. It is by no means the case that these are reasonable assumptions in realistic networks. First, nearly every feature of a wireless network is fraught with errors: the unit disc model for communication is a crude mathematical approximation to a real system. When compounded with signal interference, signal bounce, and node failure, the relatively clean geometric picture in this (and, indeed, the typical mathematical sensor networks) paper is challenged. Second, many errors associated to sensors and sensor readings are not errors of failure-to-read, but rather false (inaccurate) readings which are thought to be true. In this case, one resorts to logical models for consistency checks. It is not inconceivable that these consistency methods are miscible with the geometric techniques of this paper.

6.3. Target isolation. A problem which is dual to the separation problem of this paper is the *isolation* problem. Given a node $x_0 \in V(\Gamma)$, determine if there is a cycle in Γ whose projection to the plane has nonzero winding number with respect to x_0, and, if such exists, find a (short) representative of this cycle. The conference paper [**13**] contains some results in this direction.

7. Concluding remarks

This paper considers the difficult problem of detecting and constructing surrounding cycles in a network whose nodes are neither localized nor endowed with enough information (distances, angles) with which to derive an accurate localization. This is a problem of pressing importance in future applications of ad hoc security networks and networked autonomous agents. In particular, as sensor nodes are miniaturized, the lack of cheap efficient GPS and the expense of localization make it imperative to develop techniques which do not rely on coordinates. In addition, since probabilistic assumptions about random distributions are rarely verifiable or applicable, we focus on methods which assume no control over the node density in the plane.

The challenge of localization in an unknown environment is significant across many areas of robotics and sensor networks, and has generated an impressive array of techniques and perspectives. This paper demonstrates that localization is not a prerequisite to solving problems about the winding of cycles in a planar network.

For many systems, the unit disc graph possesses sufficient information to find separating cycles about a node. We also demonstrate that, in addition to the unit disc graph, an angular ordering of neighbors suffices to solve winding number problems for all possible networks. Absolute angles are not needed, and large uncertainty in (and, sometimes, abandonment of) the angular ordering data may be tolerated.

It may seem surprising that one could compute winding numbers rigorously with a very primitive vision system that cannot determine cyclic orders unless neighbors are sufficiently separated. Even more remarkable is the fact that 'most' graphs will admit a Y-tree as in Algorithm YWinding, allowing for winding number computation without any angular information at all, even in relatively sparse graphs. It is a characteristic feature of a topological approach that the solution is robust to such uncertainties: an "outrageous slop" is permissible. As with many problems in manipulation, localization, mapping, etc., the amount and quality of sensory information needed to solve the problem is sometimes far below what one would expect.

Acknowledgements

This work supported by DARPA HR0011-07-1-0002 and by NSF PECASE DMS-0337713. A preliminary version of some of the results in this work appeared in the proceedings of the 2006 *Workshop in the Algorithmic Foundations of Robotics* [**13**]. The examples of Fig. 3 and Fig. 4 come from conversations with David Lipsky and Sameera Poduri.

References

[1] A. Abrams and R. Ghrist, "State complexes for metamorphic robots," *Intl. J. Robotics Research*, **23**(7,8), 809–824, 2004.

[2] S. Alexander, R. Bishop, and R. Ghrist, "Pursuit and evasion in non-convex domains of arbitrary dimensions," in Proc. of Robotics: Systems & Science, 2006.

[3] H. Breu and D. G. Kirkpatrick. "Unit Disk Graph Recognition is NP-hard," *Comput. Geom. Theory Appl.*, 9(1-2):3–24, 1998.

[4] J. Bruck, J. Gao, and A. Jiang, "Localization and routing in sensor networks by local angle information," in Proc. of the Sixth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'05), 181-192, 2005.

[5] V. de Silva and R. Ghrist, "Coordinate-free coverage in sensor networks with controlled boundaries via homology," to appear, *Intl. J. Robotics Research*.

[6] V. de Silva and R. Ghrist, "Coverage in sensor networks via persistent homology," to appear, *Alg. & Geom. Topology*.

[7] V. de Silva, R. Ghrist, and A. Muhammad, "Blind swarms for coverage in 2-d," in Proc. Robotics: Systems & Science, 2005.

[8] J. Edmonds, "A combinatorial representation for polyhedral surfaces," *Notices of Amer. Math. Soc.*, 7, 646, 1960.

[9] Q. Fang and J. Gao and L. Guibas, "Locating and Bypassing Routing Holes in Sensor Networks," in Proc. 23rd Conference of the IEEE Communications Society (InfoCom), 2004.

[10] Q. Fang, J. Gao, L. J. Guibas, Vin de Silva, Li Zhang, "GLIDER: Gradient landmark-based distributed routing for sensor networks," in IEEE INFOCOM'05, March, 2005.

[11] M. Farber, "Topological Complexity of Motion Planning," *Discrete and Computational Geometry* 29, 211221, 2003.

[12] S. Fekete, A. Kröller, D. Pfisterer, and S. Fischer, "Deterministic boundary recongnition and topology extraction for large sensor networks," in *Algorithmic Aspects of Large and Complex Networks*, 2006.

[13] R. Ghrist, D. Lipsky, S. Poduri, and G. Sukhatme, "Surrounding nodes in coordinate-free networks," in *Proc. Workshop in Algorithmic Foundations of Robotics*, 2006.

[14] V. Guillemin and A. Pollack. *Differential Topology.* Prentice Hall, 1974.

[15] A. Jadbabaie, “On geographic routing without location information,” in Proc. IEEE Conf. on Decision and Control, 2004.

[16] E. Klavins, R. Ghrist, and D. Lipsky, “The graph grammatical approach to self-organizing robotic systems,” *IEEE Trans. Automatic Controls*, 51(6), 949–962, 2006.

[17] F. Kuhn, T. Moscibroda, and R. Wattenhofer. “Unit Disk Graph Approximation,” In Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL-M), 2004.

[18] A. Muhammad and M. Egerstedt, “Connectivity Graphs as Models of Local Interactions,” *J. Applied Mathematics and Computation* 168(1), 243–269, 2005.

[19] P. Niyogi, S. Smale, and S. Weinberger, “Finding the Homology of Submanifolds with High Confidence from Random Samples,” to appear, Discrete and Computational Geometry, 2006.

[20] S. Poduri, S. Pattem, B. Krishnamachari, and G. Sukhatme, “A unifying framework for tunable topology control in sensor networks,” USC CRES Technical Report *CRES-05-004*, 2005.

[21] A. Rao, C. Papadimitriou, S. Shenker, and I. Stoica, “Geographic Routing without Location Information,” in Proceedings of 9th Annual International Conference on Mobile Computing and Networking (Mobicom’03), September 2003.

[22] B. Tovar and S. LaValle, “Using a robot to learn geometric information from permutations of landmarks,” preprint, 2006.

Department of Mathematics and Coordinated Sciences Laboratory, University of Illinois, Urbana, IL 61801, USA

E-mail address: ghrist@math.uiuc.edu