2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Towards Simplicial Coverage Repair for Mobile Robot Teams

Jason Derenick, Vijay Kumar and Ali Jadbabaie

Abstract— **In this note, we present initial results towards developing a distributed algorithm for repairing topological holes in the sensor cover of a mobile robot team. Central to our approach is the melding of recent advances in the application of computational homology (a sub-discipline of algebraic topology) to *static* sensor networks with relative metric information (*i.e.* relative pose). More precisely, we consider a greedy, hybrid (discrete–continuous) algorithm whereby a desired *Cĕch complex*, the simplicial complex that captures the underlying topology of the sensing cover, is iteratively generated using local rules (between multi–hop neighbors) and agents are driven towards achieving this topology via a gradient–ascent *simplicial control law*. Convergence of the proposed algorithm is established as a function of the convergence of the underlying *simplicial control law*, and the relationship of the latter to the spectrum of the combinatorial Laplacian is considered. Simulation results for teams operating in $\mathbb{R}^2$ are presented.**

I. INTRODUCTION

The coverage problem is among the most fundamental coordination problems involving multi–agent systems. In recent years, the robotics and wireless sensor network communities have begun exploring the utility of algebraic topology for providing a solution. Algebraic topology is attractive as it provides a metric–free, mathematical toolkit for classifying topological spaces. It has already been applied to wireless sensor networks devoid of traditionally assumed geometric information (*e.g.* GPS, relative localization, *etc.*) yielding impressive algorithms which afford metric–free coverage verification and even hole "localization". Such approaches have only recently emerged, and, as a result, they have almost exclusively been focused upon static network topologies. However, the question remains: "How can algebraic topology be utilized by mobile robot teams for coverage control?"

In this paper, the first steps are taken towards addressing this question by coupling abstract simplicial complexes with relative metric information to facilitate coverage control. Specifically, we consider what may be called the *coverage–repair problem* and formulate a greedy, discrete–continuous algorithm for repairing coverage holes in the network topology. Intuitively, agents postulate in an abstract topological (combinatorial) space to generate a desired simplicial complex (*i.e.* the *Cĕch complex*) that is ultimately used to govern team behavior via decentralized *simplicial control laws*.

J. Derenick and V. Kumar are with the Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania, Philadelphia, PA 19104, USA {jasonder,kumar}@seas.upenn.edu

A. Jadbabaie is with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104, USA jadbabai@seas.upenn.edu

The authors gratefully acknowledge support from: NSF grant no. IIS-0427313, ARO grant no. W911NF-05-1-0219, ONR grants no. N00014-07-1-0829 and N00014-08-1-0696, and ARL grant no. W911NF-08-2-0004. Additionally, they would like to acknowledge helpful discussions with Alireza Tahbaz–Salehi and Robert Ghrist.

II. RELATED WORK

Given the centrality of homological constructs to the forthcoming algorithm, in this section, we only discuss related work where the application of algebraic topology was central to the authors' primary results. In [1] the authors consider metric–free static coverage verification by "sandwiching" the *Cĕch complex* of the sensor network between a pair of bounding *Rips complex*es, which capture the topology of the underlying communication graph. They extend this work in [2] by formulating a set of homological criteria to verify a static deployment of sensors covers a fenced region.

As simplicial homology (see §III) can be interpreted as a higher–dimensional abstraction of connectivity in graph theory it is not surprising that recent efforts have focused upon exploring dynamical flows. In [3], the authors consider such flows over combinatorial Laplaciancs and perform stability analysis for applications to coverage verification. The results presented in [4] exploit this result to localize coverage holes using a decentralized sub–gradient method. Additionally, [5] takes a hybrid systems perspective and establish the asymptotic stability of switched higher–order Laplacians operators for dynamic coverage verification.

Finally, a few have considered dynamic approaches rooted in homology for addressing certain variations of the coverage problem. Among these is [6] who consider a topological variation of the evader–pursuer problem where the objective is to ensure that the evader, initially occupying a coverage hole, cannot go undetected indefinitely within some fenced region. Additionally, in [7], the authors consider a switched dynamical system using higher–order Laplacians and establish that for a team of agents constrained to some domain that each point in that domain will be visited infinitely often.

III. SIMPLICES, COMPLEXES, & HOMOLOGY

Before proceeding, we introduce the homological constructs and terminology that are central to this discussion. The interested reader is referred to [8] for more details.

A. Simplices and Simplicial Complexes

Given a set of points V, a *k-simplex* is an unordered set $\sigma = \{v_0, \ldots, v_k\} \subseteq V$ where $v_i \neq v_j, \forall v_i, v_j \in \sigma$. By definition, each *k-simplex* is closed with respect to its faces where the i^{th} face, $0 \leq i \leq k$, is given by $\{v_0, \ldots, v_{i-1}, v_{i+1}, \ldots, v_k\} \subset \sigma$. A finite collection X of such simplices that maintains closure with respect to faces (*i.e.* $\sigma \in X$ implies that all faces are included in X) is called a *simplicial complex* (see Figure 1) where the dimension of X corresponds to the maximum dimension of any of its simplices with the dimension of a *k–simplex* being given by $\dim \sigma = k$. A subcomplex Y of X is a simplicial complex

978-1-4244-5040-4/10/$26.00 ©2010 IEEE

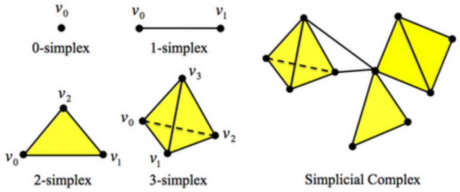


Fig. 1. Simplices are the building blocks of simplicial complexes. In the context of this research, each *0-simplex* naturally corresponds to a single robot with higher–dimensional simplices being given by local properties of the sensor cover.

such that $Y \subseteq X$. Additionally, we can define the *k-skeleton* of a complex as $X^{(k)} = \{\sigma \in X \colon \dim \sigma \leq k\}$. Intuitively, the *k-skeleton* corresponds to the set of *k–simplices* of X.

A generalization of graphs, simplicial complexes also embed some notion of adjacency. Specifically, a pair of *k–simplices* are *upper-adjacent* (denoted $\sigma_i \frown \sigma_j$) if they are faces of a *(k+1)–simplex*. Similarly, a pair of *k–simplices* are lower–adjacent (denoted $\sigma_i \smile \sigma_j$) if they share a face.

B. Homology Groups and Combinatorial Laplacians

Central to simplicial homology is the notion of the boundary homomorphism between *k-simplices* and their lower–dimensional faces. To generate this mapping on a complex X, we induce an ordering (similar to graphs) where an ordered *k–simplex* in X is denoted $\sigma = [v_0, \ldots, v_k]$. Given this ordering, the vector space $C_k(X; F)$ can be defined as the space whose basis corresponds to the set of all *k-simplices* in X with coefficients in some field (for convenience, we assume $F = \mathbb{R}$ and write $C_k(X)$) and whose members correspond to chains of oriented *k-simplices*. Given these definitions, a *boundary homomorphism* can be defined which provides the linear mapping $\partial \colon C_k(X) \rightarrow C_{k-1}(X)$ by operating on the basis elements of $C_k(X)$. Denoted B_k, in matrix form, ∂ intuitively maps chains of *k–simplices* (*i.e. k–chains*) to a linear combination of their faces.

Given the boundary operator, the k^{th} *homology group* of X can be defined as the quotient group whose generators correspond to equivalence classes of non–reducible cycles (*i.e. k–dimensional* cycles bounding a topological hole)

$$H_k(X) = \ker B_k / imB_{k+1} \tag{1}$$

Among the key properties of $H_k(X)$ is that its dimensionality (*i.e.* number of generators) corresponds to the number of *k–dimensional* holes in the corresponding complex.

Perhaps more important, however, is the kinship between $H_k(X)$ and a linear operator called the k^{th} *combinatorial Laplacian*, which is defined as the following linear combination of boundary operators mapped with their adjoints

$$\mathcal{L}_k = B_k^T B_k + B_{k+1} B_{k+1}^T \tag{2}$$

A classical result from algebraic topology establishes the isomorphic relationship $\ker \mathcal{L}_k \cong H_k(X)$. This is powerful as it tells us that $\ker \mathcal{L}_k$ captures all of the information regarding the underlying topology. As an example, $nullity(\mathcal{L}_k) = \dim H_k(X)$. More precisely, it holds that

$$\mathcal{L}_k \succ 0 \iff H_k(X) = \emptyset \tag{3}$$

IV. PROBLEM STATEMENT

Given these homological constructs, we now present a formal statement of the *coverage–repair problem*. Let $\mathcal{R} = \{r_1, \ldots, r_n\}$ denote a finite set of fully–actuated mobile robots operating on the plane with dynamics

$$\dot{q}_i = u_i \tag{4}$$

Accordingly, let $Q(t) = (q_1(t), \ldots, q_n(t))^T \in \mathbb{R}^{2n}$ denote the system's state at time t, and assume the following $\forall r_i$

- **A1** r_i has a radially symmetric coverage domain with radius $s_r \in \mathbb{R}_+$
- **A2** r_i has radially symmetric low and high–power communication broadcast ranges with respective radii $b_r^l, b_r^h \in \mathbb{R}^+$ that satisfy $2s_r \leq b_r^l < 4s_r \leq b_r^h$ (see Figure 2).
- **A3** r_i is able to measure the relative pose of neighbors within broadcast radii b_r^l or b_r^h
- **A4** r_i only broadcasts at b_r^h when necessary to ensure local network interconnectivity
- **A5** r_i has a unique identifier that it includes in broadcasts

Intuitively, **A4** serves as a mechanism for energy conservation to extend the mission–life of the team. Additionally, for notational convenience, we define $Q \triangleq Q(t)$ and $q_i \triangleq q_i(t)$.

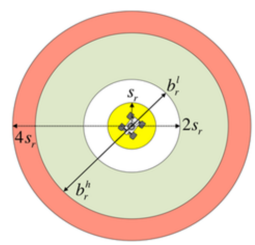


Fig. 2. Illustrating **A2**: s_r denotes the sensing (coverage) radius for r_i and b_r^l and b_r^h respectively denote its low and high–power broadcasting ranges. It is assumed r_i is capable of sensing the relative pose to proximal neighbors within b_r^l or b_r^h units – depending upon the required broadcast strength to maintain a desired level of local network connectivity.

Given assumptions **A1** – **A5**, associate with agent r_i its convex sensor support $\mathcal{U}_i = \left\{x \in \mathbb{R}^2 \colon \parallel x - q_i \parallel_2 \leq s_r\right\} \subset \mathbb{R}^2$ corresponding to the compact disk of radius s_r centered at q_i. It is known (see [1]) that the topology of the *sensor cover*, given by the union of *convex sensor supports*

$$\mathcal{U}_{\mathcal{R}} = \bigcup_{\forall r_i \in \mathcal{R}} \mathcal{U}_i, \tag{5}$$

is fully captured by a simplicial complex known as the *Cěch complex* (see Figure 3(a)). It is defined as follows

Definition 1 (The Cěch Complex): Given a finite collection of convex sets, the corresponding Cěch complex is the simplicial complex where each k–simplex corresponds to the non–empty intersection of $k+1$ sets in the collection.

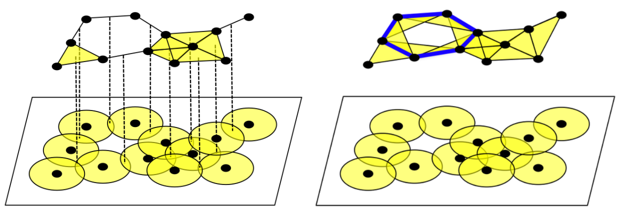
Given this definition and stated assumptions, our problem can be articulated using the terminology of §III as follows

Problem 1 (The Coverage–repair Problem): Given **A1**–**A5** and an initial *Cěch complex* X_0 for $\mathcal{U}_{\mathcal{R}}$ such that $H_1(X_0) \neq \emptyset$ (*i.e.* $\mathcal{L}_k \not\succ 0$) transition $\mathcal{R}$ to new *Cěch complex* X_n such that $H_1(X_n) = \emptyset$ (*i.e.* $\mathcal{L}_k \succ 0$).

In this research, we are interested in developing a distributed solution to this problem.

V. AN ALGORITHMIC APPROACH

The intuition behind the proposed algorithm is to identify cycles bounding coverage holes in X_0 and supplant each such cycle of length k with a *k–simplex* in the final topology. Towards this end, we consider a coupled approach whereby the original *Cěch complex* (*i.e.* X_0) is iteratively augmented with *weighted 2–simplices* among 2–hop neighbors lying along the bounding cycle. By driving the robots to maximize the weights of their respective *2–simplices* (byway of our *simplicial control law*) additional weighted simplices will be introduced as robots become proximal. This process continues until each bounding cycle is retracted and the nullspace of the combinatorial Laplacian becomes trivial.



(a) (b)

Fig. 3. (a) An example of the *Cěch complex* capturing the sensor cover (*i.e.* $\mathcal{U}_\mathcal{R}$) of the underlying sensor network. Observe that a hole in $\mathcal{U}_\mathcal{R}$ corresponds to a hole in the *Cěch complex*. (b) Given a bounding cycle (bold), Algorithm 2 introduces desired simplices among *2–hop* neighbors in $X_d(k)$ lying along the cycle. Additionally, to ensure each robot is involved in a *2–simplex*, it introduces the desired *2–simplex* that lies to the far right.

Given this intuition, the key to formulating the proposed algorithm is the realization that a *sufficient* criteria for solving Problem 1 is to guarantee the convergence of the network topology to a hole–free subcomplex of some *Cěch complex* capturing $\mathcal{U}_\mathcal{R}$. Towards this end, we let $X_d(k)$ denote the desired subcomplex at step k and let $\Delta_{ijk} \in X_d^{(2)}(k) \subseteq X_d(k)$ denote the *2–simplex* governed by the intersection of supports $\mathcal{U}_i, \mathcal{U}_j, \mathcal{U}_k \subset \mathcal{U}_\mathcal{R}$ corresponding to agents $r_i, r_j, r_k \in \mathcal{R}$. Accordingly, define the smooth functional

$$f(\Delta_{ijk})\colon X_d^{(2)}(k) \to [0,1] \tag{6}$$

mapping each Δ_{ijk} to its corresponding state–dependent weight value. Intuitively, the weight $f(\Delta_{ijk})$ loosely serves as an indicator of $\mathcal{U}_i \cap \mathcal{U}_j \cap \mathcal{U}_k \neq \emptyset$ (or, more naturally, as a measure of adjacency for its 1–dimensional faces) as it should achieve maximal value (*i.e.* 1) when the corresponding *2–simplex* is present. For the moment we defer further discussion of f until §V-C with the exception of noting that it will lead r_i, r_j, r_k associated with Δ_{ijk} towards a configuration where their supports have non–empty intersection.

Given this functional, consider the *weighted 2–skeleton* of $X_d(k)$, which we define as the 2–tuple

$$X_w^{(2)}(t) \triangleq \left(X_d^{(2)}(k), W(t)\right) \tag{7}$$

where $W(t)$ is defined

$$W(t) = \frac{1}{\#_2(X_d(k))} \sum_{\forall \Delta_{ijk}} f(\Delta_{ijk}) \tag{8}$$

with $\#_k(X_d)$ denoting the number of *k–simplices* in X. Observe that when $W(t) = 1$ all simplices in $X_d(k)$ are defined in the actual *Cěch complex*. For notational convenience, we write $X_w^{(2)}(t) = X_d^{(2)}(k)$ (or equivalently $X_w(t) = X_d(k)$).

Our objective then is to design a *hybrid algorithm* that generates a finite sequence of discrete topological transitions

$$\underbrace{X_w^{(2)}(t) \to \left(X_d^{(2)}(0), \mathbf{1}\right)}_{k=0}, \ldots, \underbrace{X_w^{(2)}(t) \to \left(X_d^{(2)}(n), \mathbf{1}\right)}_{k=n} \tag{9}$$

where $X_0 \subset X_d(0)$, $H_1(X_d(0)) \neq \emptyset$, and $H_1(X_d(n)) = \emptyset$. Intuitively, the transition at step k should occur when the team achieves $X_w(t) = X_d(k)$. At which point, a new desired subcomplex $X_d(k+1)$ will be generated. This process continues until a desired subcomplex is achieved that is hole–free. As it is natural to consider the generation of $X_d(k+1)$ by simply augmenting $X_w(t) = X_d(k)$ with additional weighted simplices, it is assumed $X_d(k) \subset X_d(k+1)$.

Considering (9), it is clear that two convergent (dependent) sequences must be established. First, it must be shown how to effectively generate $X_d(k+1)$, given that $X_d(k) = X_w(t)$, to ensure convergence to a hole–free topology. Second, it must be shown that $X_w(t)$ can be driven to achieve $X_d(k)$ via a *simplicial controller*. To facilitate understanding, we adopt a top–down approach and consider the former before concluding with a formulation of *simplicial control*, and how it can be utilized to solve Problem 1.

A. Statement of Algorithm

In generating the proposed algorithm, we leverage the recent results of [4] and [7]. In [4] the authors employ a decentralized subgradient method for metric–free hole localization by solving the following LP

$$\min_{z \in \mathbb{R}^{\#_k(X)}} \| x + B_{k+1} z \|_1 \tag{10}$$

where $x \in \ker \mathcal{L}_k$. The solution corresponds to an approximation of the sparsest generator of $H_k(X)$. Loosely speaking, this generator corresponds to the cycle (or linear combination of such cycles) that has/have the fewest number of hops and bounds some *k–dimensional* hole. In the context of this research, we shall exploit this result to identify *1–cycles* bounding topological holes in the *Cěch complex*.

Additionally, we leverage the results of [7]. In this work, the authors consider the combinatorial Laplacian flow

$$\dot{x}(t) = -\mathcal{L}_k x(t), x(0) \in \mathbb{R}^{\#_k(X)} \tag{11}$$

and show that of $H_k(X) = \emptyset$ implies that (11) is *asymptotically stable* and that $H_k(X) \neq \{\emptyset\}$ implies (11) is *semi–stable* with solution $x \in \ker \mathcal{L}_k$. These results are key as they provide a mechanism for distributed coverage verification and computation of $x \in \ker \mathcal{L}_k$, which is necessary for (10).

Algorithm 1 presents our high–level algorithm, and it can be thought of as behaving as a ternary state machine. During the initial state, (11) is solved to verify coverage. If the solution $x \notin \ker \mathcal{L}_k$, it can be used to seed the second state of the system, which localizes topological holes by

Algorithm 1 repairCoverage(X_0, ϵ)

Require: $0 < \epsilon \ll 1$ and $X_0^{(1)}$ is connected.

$X \leftarrow X_0$
while $x \neq 0$ **do**
  Solve: $\min_{z \in \mathbb{R}^{\#_k(X)}} \| x + B_2 z \|_2$
  $X \leftarrow retractCycles(X, \epsilon)$
  Solve: $\dot{x}(t) = -\mathcal{L}_1(X)x(t), x(0) \in \mathbb{R}^{\#_1(X)}$
end while
return X

way of (10). In the final state, localized cycles are retracted via a distributed algorithm (§V-B). Naturally, the machine is repeated until coverage is repaired.

B. Retracting Localized Holes

Recall that our approach for repairing coverage is to close a hole by retracting a bounding cycle of length k to a corresponding *k–simplex* in X. To this end, momentarily assume the asymptotic convergence of $X_w(t) \to (X_d(k), \mathbf{1})$. Given this assumption, we now show how to generate a sequence of desired *Cěch subcomplex*es that ultimately converges to new topology where a bounding cycle of length k has been retracted to a corresponding *k–simplex*. Begin by considering Algorithm 2 which generates $X_d(k+1) \supset X_d(k) = X_w(t)$.

Algorithm 2 getDesiredCechSubcomplex($X_d(k)$)

Require: $X_w(t) = X_d(k)$

1: $X_d(k+1) \leftarrow X_d(k)$
2: **for all** $r_i \in \mathcal{R}$ **do**
3:   $\phi_1 \leftarrow \mathcal{R}_{\mathcal{C}_i} \cap \mathcal{H}^1_{r_i}(X_d(k+1))$
4:   $\phi_2 \leftarrow getAllCombinations(\phi_1, 2)$
5:   **for all** $(r_u, r_v) \in \phi_2$ **do**
6:     $X_d^{(2)}(k+1) \leftarrow X_d^{(2)}(k+1) \cup \Delta_{iuv}$
7:   **end for**
8:   **for all** $r_u \in \mathcal{H}^1_{r_i}(X_d(k+1))$ **do**
9:     **if** $\nexists r_v, u \neq v, r_v \in \mathcal{H}^1_{r_i}(X_d(k+1))$, $r_v \in \mathcal{H}^1_{r_u}(X_d(k+1))$ **then**
10:       $X_d^{(2)}(k+1) \leftarrow X_d^{(2)}(k+1) \cup \Delta_{iuv}$
11:     **end if**
12:   **end for**
13: **end for**
14: **return** $X_d(k+1)$

Algorithm 2 performs two separate steps. During the first (lines 2–9), *2-simplices* are introduced between r_i and each pair of 1–hop neighbors along its bounding cycle(s). The second step is given by lines 10–21, and it updates the desired subcomplex $X_d(k+1)$ to ensure each agent is involved in at least a single *2–simplex*. See Figure 3(b) for example output.

Given this result, we can now establish a general algorithm for retracting topological holes. Let $\mathcal{C}_{\mathcal{R}} = \{c_1, \ldots, c_\ell\}$ denote a set of cycles (or more generally, linear combinations of such cycles) known to bound holes in X_0. Let $\mathcal{C}_{r_i} = \{c_j \colon c_j \in \mathcal{C}_{\mathcal{R}}, r_i \in c_j\}$ denote those cycles involving r_i. Additionally, let $\mathcal{R}_{\mathcal{C}}$ denote the set of all robots involved in such cycles (*i.e.* $\mathcal{R}_{\mathcal{C}} = \{r_i \colon \exists c_j \in \mathcal{C}_{\mathcal{R}}, r_i \in c_j\}$) and define $\mathcal{R}_{\mathcal{C}_i}$ as the set of all robots $r_j \in \mathcal{R}, i \neq j$ such that $\exists c_j \in \mathcal{C}_{r_i}, r_j \in c_j$. Define $\mathcal{H}^1_{r_i}(X_d(k))$ as the mapping of r_i to its 1–hop neighbors in $X_d(k)$. Given these definitions, consider Algorithm 3 for generating a nested–sequence, $X_d(k) \subset X_d(k+1)$ that converges to a subcomplex where each k–hop cycle has been retracted to a *k–simplex*. Accordingly, we now formalize its convergence in Theorem 5.1.

Algorithm 3 retractCycles(X_0, ϵ)

Require: $0 < \epsilon \ll 1$ and $X_0^{(1)}$ is connected.

$X_d \leftarrow X_0$
repeat
  $X_d \leftarrow getDesiredCechSubcomplex(X_d)$
  while $\exists \Delta_{ijk} \in X_d^{(2)}, f(\Delta_{ijk}) < 1 - \epsilon$ **do**
    $\Delta_{ijk} = \Delta_{ijk} + \dot{\Delta}_{ijk}\delta t$
  end while
until $\forall r_i \in \mathcal{R}_{\mathcal{C}}, \forall r_j \in \mathcal{R}_{\mathcal{C}_i}, r_j \in \mathcal{H}^1_{r_i}(X_d(k))$
$X \leftarrow getCechComplex(X_d(k))$
return X

Theorem 5.1: Assume $X_w(t) \to (X_d(k), \mathbf{1})$ and let the length of $c_j \in \mathcal{C}$ be given by $k_j \in \mathbb{Z}_{\geq 3}$. Algorithm 3 converges to a subcomplex $X_d(m)$ such that each c_j is retracted to a k_j*–simplex*, $\sigma_j \in X_d(m)$.

Proof: By contradiction. Assume Algorithm 3 terminates and $\exists c_j \in \mathcal{C}, \sigma_j \notin X_d(m)$. This implies $\exists \sigma_u, \sigma_v \in c_j, u \neq v$ such that σ_u and σ_v are *0–simplices* and $\sigma_u \not\frown \sigma_v$. Noting that $X_d^{(1)}(0)$ must be connected since $X_0^{(1)}$ is connected (since $X_w(t) \to (X_d(k), \mathbf{1})$ at step k), $\exists$ a sequence in $X_d(m)$ of *1–simplices* $c_{uv} \subset c_j$ that joins σ_u and σ_v. Since $\sigma_u \not\frown \sigma_v$, $\exists \sigma_s, \sigma_t \in c_{uv}, s \neq t, s \neq u, t \neq u$ such that σ_s and σ_t are *0–simplices*, $\sigma_t \frown \sigma_s$, $\sigma_s \frown \sigma_u$, and $\sigma_u \not\frown \sigma_t$. However, by Algorithm 2, $\sigma_t \frown \sigma_s$, $\sigma_s \frown \sigma_u \Rightarrow \sigma_u \frown \sigma_t$ in X. This yields the necessary contradiction. ■

It should be noted that applying Algorithm 3 may inadvertently introduce holes in the desired topology $X_d(k+1)$. It is for this reason that (11) is solved after each iteration of Algorithm 1 to verify coverage and ensure the algorithm only terminates when a hole–free topology has been achieved.

C. Defining Simplicial Control Laws

Given Algorithm 1, it is evident that a hole–free topology can be successfully generated provided that $X_w(t) \to (X_d(k), 1), \forall k$. In this section, we now consider this convergence in terms of *simplicial control laws*. Ideally, a *simplicial control law* will allow us to abstract the team in terms of *k–simplices* (in this research, $k = 2$) and drive the underlying state implicitly as a functional of simplex weights that capture the *combinatorial relationship* between robots.

a) Controller Synthesis: Begin by recalling the general mapping (6) which serves to indicate $\mathcal{U}_i \cap \mathcal{U}_j \cap \mathcal{U}_k \neq \emptyset$. Given our application to coverage–repair, it is natural to consider a *simplicial control law* that is a functional of the max–distance separating r_i, r_j, r_k who comprise Δ_{ijk}. Accordingly, define $d_{ijk} \triangleq \max\{d_{ij}, d_{ik}, d_{jk}\} \leq 4s_r, \forall \Delta_{ijk} \in X_d^{(2)}(k)$ where

$d_{ij} \in \mathbb{R}^+$ denotes the ℓ_2–norm between r_i and r_j. Abusing notation slightly, define (6) as the real–valued mapping $f(d_{ijk})\colon \mathbb{R}^3 \to [0,1]$ given by the *sigmoid*

$$f(\Delta_{ijk}) \triangleq f(d_{ijk}) = \left(1 + \epsilon^{\gamma(d_\mu - d_{ijk})}\right)^{-1} \tag{12}$$

where $0 < \epsilon \ll 1$, $\gamma = \frac{1}{d_\mu - d_{ijk}^{min}}$, $d_{ijk}^{min} = 2s_r cos(\frac{\pi}{6})$ is the conservative maximal distance allowed between r_i, r_j, and r_k while still ensuring $\mathcal{U}_i \cap \mathcal{U}_j \cap \mathcal{U}_k \neq \emptyset$ (see Figure 4(a)), and $d_\mu = \frac{1}{2}(d_{ijk}^{max} + d_{ijk}^{min})$ with $d_{ijk}^{max} = 4s_r$.

By definition, observe that $f(d_{ijk}^{min}) = (1+\epsilon)^{-1} \approx 1$ and $f(d_{ijk}^{max}) = \epsilon(1+\epsilon)^{-1} \approx 0$. The latter value indicates that the sensor supports are "far" from intersecting and thus the face associated with $\Delta_{ijk} \in X_w^{(2)}(t)$ is "weak". Conversely, the former condition tells us that their supports meet in a non–empty intersection. An additional benefit of choosing (12) is that as $f \to 1$ it holds that $\triangledown f \to 0$. This property serves as an embedded mechanism to reduce the collision–likelihood between involved agents. Figure 4(b) illustrates this function.

At the simplex–level, driving $X_w(t) = (X_d(k), \mathbf{1})$ (*i.e.* solving (8) w.r.t. $X_d(k)$) corresponds to solving a standard constrained convex optimization problem since $f(d_{ijk})$ is a quasi–convex function in the entries of the combinatorial Laplacian $\mathcal{L}_1(X_d(k))$. Unfortunately, developing such a controller directly as a functional of *1–simplices* is hardly straightforward. As such, for these initial results, we consider *simplicial control* with respect to q_i and sacrifice problem convexity as $f(d_{ijk})$ is non–convex in Q.

To this end, consider the analytic approximation of the $\max$ function (see [9]) given by the *log–sum–exponential*

$$d_{ijk} \approx \frac{1}{\alpha}\log\left(e^{\alpha d_{ij}^2} + e^{\alpha d_{ik}^2} + e^{\alpha d_{jk}^2}\right) \tag{13}$$

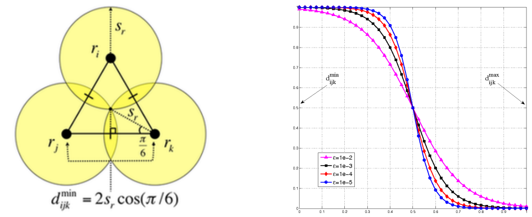
where $\alpha \in \mathbb{R}^+, \alpha \gg 1$ and $d_{ij} \in \mathbb{R}^+$ is as previously defined with respect to r_i and r_j. Differentiating (8) with respect to q_i yields the following *simplicial control law* for $r_i \in \mathcal{R}$

$$\dot{q}_i = \sum_{\forall \Delta_{ijk}} \frac{\alpha\gamma\log(\epsilon)\epsilon^{\gamma(d_\mu - d_{ijk})} f(d_{ijk})^2}{e^{\alpha d_{ij}^2} + e^{\alpha d_{ik}^2} + e^{\alpha d_{jk}^2}} (e^{\alpha d_{ij}^2} q_{ij} + e^{\alpha d_{ik}^2} q_{ik}) \tag{14}$$

where $q_{ij} = (q_i - q_j)$ is the relative pose of q_j with respect to q_i with α and γ being as previously defined.

Note that (14) lends itself to a decentralized control policy. This results as $\dot{q}_i$ is only computed over simplices in which r_i is involved and is, thus, determined locally (over *2–hop* neighbors in the *Čech complex*). Furthermore, notice that (14) requires r_i to only estimate the *relative pose* of robots comprising a *2–simplex* with r_i and does not require metric–localization. Its convergence to a local equilibrium is ensured since $\dot{Q}$ is an ascent–direction for (8), which is bounded.

In order for (14) to be properly implemented, it is required that the constraint $d_{ijk} \leq 4s_r \forall \Delta_{ijk} \in X_d(k)$ be satisfied at all times. Enforcing this constraint is straightforward via standard gradient–projection algorithms [10]. Furthermore, it is also required that agents maintain network connectivity across each $\Delta_{ijk} \in X_d(k)$. Although $X_w(t) = X_d(k)$ ensures b_r^l will provide this level of connectivity, b_r^l may not



(a) (b)

Fig. 4. (a) $f(\Delta_{ijk})$ for different values of ϵ. $f(\Delta_{ijk})$ is a quasi–convex function of d_{ijk} that is used to drive agents towards a configuration where $\mathcal{U}_i \cap \mathcal{U}_j \cap \mathcal{U}_k \neq \emptyset$. (b) Geometric derivation of $d_{ijk}^{min} = 2s_r cos(\frac{\pi}{6})$, where s_r corresponds to the uniform radius of sensor coverage.

be sufficient in general as $r_i, r_j, r_k \in \Delta_{ijk}$ can be separated by d_{ijk}^{max} units. Accordingly, r_i must utilize b_r^h to maintain connectivity with r_j and r_k in Δ_{ijk} when necessary.

Before proceeding, it is important to note that (14) does *not* guarantee the convergence of $X_w(t)$ to $(X_d(k), 1)$ However, it should be noted that given this initial formulation, our simulation results indicate that the convergence of $X_w(t)$ to $(X_d(k), 1)$ is a functional of the underlying slope of the chosen sigmoid. An exploration of this point is the focus of ongoing research. Nevertheless, in our simulation results, employing (14) has worked quite well.

b) Relating the Combinatorial Laplacian: Our objective is to now establish a relationship between (8) and the spectrum of $\mathcal{L}_k$. Begin by defining a weighted variation of the combinatorial operator as follows

$$\mathcal{L}_k^w = W_k^T B_k^T B_k W_k + B_{k+1} W_{k+1} B_{k+1}^T \tag{15}$$

where $W_k = diag\left(w_{k,1}, \ldots, w_{k,\#_k(X)}\right) \in \mathbb{R}^{\#_k(X) \times \#_k(X)}$, and $w_{k,i} \in [0,1]$. It is straightforward to establish $\ker \mathcal{L}_k^w \cong \ker \mathcal{L}_k$ since all simplex weights are positive. Given this discussion, we now present the following natural result

Theorem 5.2: Let $\mathcal{R}$ denote a team with kinematics (4), and let $X_d(k)$ denote its desired *Čech subcomplex*. Maximizing (8) is equivalent to solving

$$\begin{aligned} &\max \sum_{\forall \Delta_{ijk}} \tfrac{1}{3}\lambda_{(1,ijk)} \\ &\text{s.t. } \mathcal{L}_{(1,ijk)}^w - \lambda_{(1,ijk)} I_3 \succeq 0 \\ &\quad\;\; \mathcal{L}_{(1,ijk)}^w = W_1^T B_1^T B_1 W_1 + B_2 W_2 B_2^T \end{aligned} \tag{16}$$

where I_3 is the 3×3 identity, $W_1 = I_3$, $W_2 = f(d_{ijk})$, and $\lambda_{(1,ijk)}$ denotes the smallest eigenvalue corresponding to the combinatorial Laplacian associated with $\Delta_{ijk} \in X_d^{(2)}(k)$.

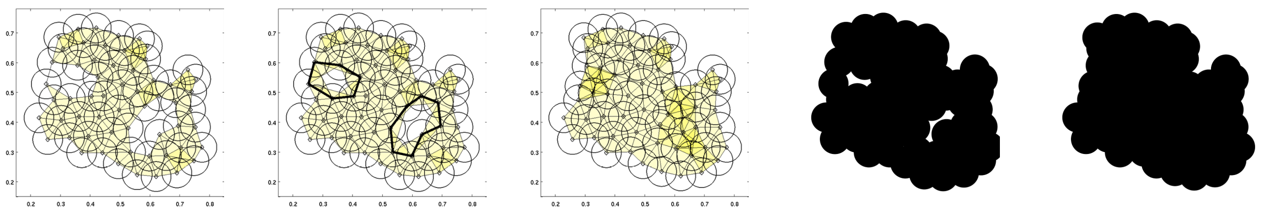
Proof: Observe that for Δ_{ijk}, it holds that $\mathcal{L}_{(1,ijk)}^w = W_1^T B_1^T B_1 W_1 + B_2 W_2 B_2^T = B_1^T B_1 + B_2 f(\Delta_{ijk}) B_2^T =$

$$\begin{pmatrix} 2 + f(\Delta_{ijk}) & 1 - f(\Delta_{ijk}) & f(\Delta_{ijk}) - 1 \\ 1 - f(\Delta_{ijk}) & 2 + f(\Delta_{ijk}) & 1 - f(\Delta_{ijk}) \\ f(\Delta_{ijk}) - 1 & 1 - f(\Delta_{ijk}) & 2 + f(\Delta_{ijk}) \end{pmatrix} \tag{17}$$

which has eigenvalues $[3f(\Delta_{ijk}), 3, 3]^T$. Given the range of f, it follows $\lambda_{(1,ijk)} = 3f(\Delta_{ijk})$. Normalizing this sum over all $\lambda_{(1,ijk)}$ with respect to $\#_2(X_d(k))$ yields (8). ■

D. A Computationally Distributed Implementation

Accordingly, we now discuss a computationally distributed implementation of the proposed algorithm. Begin by observing that both Algorithms 2 and 3 lend themselves to



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

Fig. 5. Applying Algorithm 1 with *simplicial control law* (14): (a) Initial *Čech complex* (*i.e.* X_0) for a network of 58 robots in $\mathbb{R}^2$. (b) An embedding of $X_d(0)$ shown with localized holes (bold). (c) The final *Čech complex* having trivial generator (*i.e.* X is hole–free) shown with the retracted cycles (bold). Here, the algorithm converges to a local, hole–free equilibrium. (d) Initial $\mathcal{U}_{\mathcal{R}}$ showing coverage holes in (a). (e) Hole–free $\mathcal{U}_{\mathcal{R}}$ corresponding to (c).

a distributed implementation. Specifically, Algorithm 3 behaves as a state machine whose states and transitions can be managed by some subset of robots in the network. Coupling this with the fact that that the construction of a desired subcomplex is inherently local as it only depends upon localized distance measures fosters a distributed algorithm. In fact, decentralized network protocols already exist for computing simplicial complexes [11], which can be exploited.

Melding these observations with the results of [4] and [7] fosters a computationally distributed implementation of Algorithm 1. Supporting this point, recall that solving (10) can be readily done via a decentralized subgradient method [4]. Additionally, a robot can readily detect if it lies on a cycle by evaluating the coefficients associated with its *1–simplices* determined by (10). Finally, as the *Čech complex* can be constructed locally, (11) is also readily distributed.

VI. Simulation Results

Algorithm 1 was implemented in Matlab. In this implementation, a slightly more intelligent variation of Algorithm 3 was chosen[1] that checks the underlying topology during each iteration of lines 2–7 to determine whether the retraction of each bounding *k–cycle* to a *k–simplex* is still necessary for repair. Figure 5 shows the results obtained for a team of 58 robots in $\mathbb{R}^2$. The initial topology, X_0, and cover (see Figures 5(a), 5(d)) reveal a pair of holes. Given X_0, Algorithm 1 localizes the coverage holes by finding bounding cycles for each. These cycles are then utilized to generate $X_d(0)$ as seen in Figure 5(b). Figures 5(c) and 5(e) show the final *Čech complex* and cover, $\mathcal{U}_{\mathcal{R}}$. Each r_i had a uniform sensing range $s_r = 0.05$ with $\epsilon = 1e - 8$ and $\alpha = 5000$.

As a final note, our communication radii were respectively chosen such that $r_c^w = 2s_r$ and $b_r^h = 4s_r = d_{ijk}^{max}$. Given our choice of b_r^h, each hop in our desired *Čech subcomplex* corresponded to a single hop in the weak communication graph. Furthermore, given the choice of both b_r^l and b_r^h, our algorithm can be loosely interpreted as sandwiching the desired *Čech complex* between strong and weak communication topologies. Such an approach is reminiscent of [1], who consider metric–free, static coverage verification.

VII. Conclusions

In this paper, we presented initial results in developing a distributed, greedy algorithm to solve the *coverage–repair problem* for planar networks. Central to these results is the coupling of an abstract *Čech complex* with relative metric–information. An algorithm was presented that generates a nested sequence of desired *Čech complexes* heading towards a hole–free topology. The notion of a *simplicial controller* was proposed to drive agents towards achieving each desired complex, and we considered its interpretation with respect to the spectrum of the combinatorial Laplacian. When the underlying *simplicial control law* drives $X_w(t)$ asymptotically to $(X_d(k), 1)$, it was shown that the proposed algorithm will converge to a hole–free topology. Finally, we presented initial results in developing a position–based *simplicial control law*.

A few final points should be made. First, given the greedy nature of the proposed algorithm, it is best suited for networks where topological holes are relatively small with respect to the size of the overall network. Second, it is theoretically possible that $\mathcal{U}_{\mathcal{R}}$ can be retracted to a point; however, this has not been seen in any of our simulations.

As continued research on this topic, we aim to characterize the conditions under which pathological retraction can occur. Additionally, we are exploring the effects of stationary nodes on convergence and extensions to perimeter surveillance.

[1] this variation still preserves our original convergence results and analysis

References

[1] R. Ghrist and A. Muhammad, "Coverage and hole-detection in sensor networks via homology," in *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*. Piscataway, NJ, USA: IEEE Press, 2005, p. 34.

[2] R. Ghrist and V. De Silva, "Coverage in sensor networks via persistent homology," *Algebraic and Geometric Topology*, p. pp. 339358, 2007.

[3] A. Muhammad and M. Egerstedt, "Control using higher order laplacians in network topologies," in *Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems*, Kyoto, Japan, 2006, pp. 1024–1038.

[4] A. Tahbaz-Salehi and A. Jadbabaie, "Distributed coverage verification algorithms in sensor networks without location information," *IEEE Transactions on Automatic Control*, Feb 2009.

[5] A. Muhammad and A. Jadbabaie, "Decentralized computation of homology groups in networks by gossip," in *Proceedings of American Control Conference*, New York, NY, Apr 2007, pp. 3438–3443.

[6] R. Ghrist, V. de Silva, and A. Muhammad, "Blind swarms for coverage in 2-d," in *Robotics: Science and Systems II*, Jun 2005.

[7] A. Muhammad and A. Jadbabaie, "Dynamic coverage verification in mobile sensor networks via switched higher order laplacians," in *Robotics: Science and Systems III*. The MIT Press, 2007.

[8] A. Hatcher, *Algebraic Topology*. Cambridge University Press, 2002.

[9] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge Unviersity Press, 2004.

[10] D. Luenberger, *Linear and Nonlinear Programming*. Springer, 2003.

[11] A. Muhammad and A. Jadbabaie, "Decentralized computation of homology groups in networks by gossip," in *Proceedings of American Control Conference*, New York, NY, 2007.