# CONFIGURATION SPACES, BRAIDS, AND ROBOTICS

R. GHRIST

ABSTRACT. Braids are intimately related to configuration spaces of points. These configuration spaces give a useful model of autonomous agents (or robots) in an environment. Problems of relevance to autonomous engineering systems (e.g., motion planning, coordination, cooperation, assembly) are directly related to topological and geometric properties of configuration spaces, including their braid groups. These notes detail this correspondence, and explore several novel examples of configuration spaces relevant to applications in robotics. No familiarity with robotics is assumed.

These notes shadow a two-hour set of tutorial talks given at the National University of Singapore, June 2007, for the *IMS Program on Braids*.

CONTENTS

## 1. Configuration spaces and braids

These notes outline an elegant relationship between braids, configuration spaces, and applications across several engineering disciplines associated with robotics and coordination. We begin with a few standard definitions.

**Definition 1.** The **configuration space** of $n$ distinct labeled points on a topological space $X$, denoted[1] $\mathcal{C}^n(X)$, is the space

$$\mathcal{C}^n(X) = \prod_1^n X \ - \Delta, \tag{1.1}$$

where $\Delta$ denotes the **diagonal**

$$\Delta = \{(x_i)_1^n : x_i = x_j \text{ for some } i \neq j\}\,. \tag{1.2}$$

The **unlabeled configuration space**, denoted $\mathcal{UC}^n(X)$, is defined to be the quotient of $\mathcal{C}^n(X)$ by the natural action of the symmetric group $S_n$ which permutes the ordered points in $X$. The $n$-strand **braid group** of a space $X$ is defined to be $B_n(X) := \pi_1(\mathcal{UC}^n(X))$, whereas the $n$-strand **pure braid group** of $X$ is $P_n := \pi_1(\mathcal{C}^n(X))$. (We generally deal with connected configuration spaces and thus ignore basepoints.)

*Example* 2. The simplest and best-known (nontrivial) examples are the configuration space of the plane $\mathbb{R}^2$, whose fundamental groups yield the classical **Artin braid groups** $B_n = \pi_1(\mathcal{UC}^n(\mathbb{R}^2))$ and their pure cousins $P_n = \pi_1(\mathcal{C}^n(\mathbb{R}^2))$.

Planar configuration spaces are not easy to visualize for arbitrary $n$, but the elements of the braid groups are eminently intuitive – almost tactile – objects.

*Exercise* 3. Show that $\mathcal{C}^2(\mathbb{R}^2)$ is homeomorphic to $\mathbb{R}^3 \times S^1$, where $S^1$ denotes the unit circle in $\mathbb{R}^2$. Hint: think about placing tokens on the table one at a time. Does your method of proof give a simple presentation for $\mathcal{C}^3(\mathbb{R}^2)$?

Configuration spaces of the plane have a number of excellent and elegant algebraic properties, including:

[1] Notation varies greatly with author: $F(X, n)$ is a common alternative to $\mathcal{C}^n(X)$.

(1) They are Eilenberg-MacLane spaces of type $K(\pi_1, 1)$, meaning that the fundamental group determines the homotopy type of the space and all higher homotopy groups vanish.
(2) The fundamental groups (the Artin braid groups $B_n$ and $P_n$) are all torsion-free.

The method of *iterated fibrations* — in which one fixes the location of one distinguished point at a time and builds fiber bundles with these restrictions as projection maps — makes it easy to unlock topological properties of configuration spaces of points on the plane. See, *e.g.*, [7, 26] for configuration spaces of manifolds treated in this manner.

## 2. Planning problems in robotics

It is a truth perhaps not universally acknowledged that an outstanding place to find rich topological objects is within the walls of an automated warehouse or factory.

2.1. **Motion planning.** Consider an automated factory equipped with a cadre of **automated guided vehicles** (AGVs), or mobile robots, which transport items from place to place. A common goal is to place several, say $n$, of these robots in motion simultaneously, controlled by an algorithm that either guides the robots from initial positions to goal positions (in a warehousing application), or executes a cyclic pattern (in manufacturing applications). These robots are costly and cannot tolerate collisions (with obstacles or with each other) without a loss of performance.

Anyone who shops at a large supermarket with wide aisles is familiar with this problem and a solution. If two carts are headed toward each other, a slight swerve is sufficient to avoid a collision, assuming the other does not move in the same direction. *The resolution of collisions on $\mathbb{R}^2$ is a local phenomenon.* This does not imply that planning coordinated motions is a simple task: it requires an extraordinary effort to coordinate air traffic at a busy airport, a scenario in which a collision is to be avoided at all costs.

Modeling the factory floor as $\mathbb{R}^2$ and the robots as points, one often wishes to find paths or cycles in $\mathcal{C}^n(\mathbb{R}^2)$ to enact specific behaviors. Obstacles can easily be incorporated into these models — there is a vast literature on this subject [54, 55]. Executing cyclic motions is more complex but can at first be approximated by composed point-to-point motions. Various kinematic issues (*e.g.*, steering geometry) and other physical features of real autonomous systems must be addressed in general, but the configuration space model is an appropriate approximation to reality. Of course, since the robots are not truly points, and since no control algorithm implementation is of infinite precision, we require that the control path reside outside of a neighborhood of the diagonal $\Delta$ in $(\mathbb{R}^2)^n$.

It is possible to construct safe control schemes using configuration spaces. The work of Koditschek and Rimon [51] provides one example of a concrete solution: they write out explicit vector fields on these configuration spaces that can be used to flow from initial to goal positions in the presence of certain types of obstacles. By arranging these vector fields so that they strongly push away from the vestiges of the diagonal $\Delta$ on the boundary of $\mathcal{C}^n(\mathbb{R}^2)$, the control scheme is *provably* safe from collisions (as opposed to being *statistically* safe via computer simulations): no path can ever intersect the diagonal. Furthermore, since a neighborhood of the diagonal is repelling, the control scheme is *stable* with respect to perturbations to the system. This is quite important, as mechanical systems have an annoying tendency to malfunction occasionally. Drawing an appropriate vector field on a configuration space yields an excellent method of self-correction.

This is a clean, direct application of topological and dynamical ideas to a matter of great practical relevance, and it is currently used in various industrial settings.

The robotics community, largely independently of the topology community, has enjoyed great success at identifying and manipulating configuration spaces to their advantage in control problems: see the books [54, 55] and references therein. There are, however, several classes of simple, physically relevant scenarios where there in an underlying configuration space which has remained untapped. We outline a few examples in the following subsections.

2.2. **Motion planning on tracks.** Suppose the robots must move about on a collection of tracks embedded in the floor, or via a path of electrified guidewires from the ceiling; see [14] for examples. Such a restricted network is quite common, mainly because it is more cost-effective than a full two degree-of-freedom steering system for robots. In this setting, the state of the system at any instant of time is a point in the configuration space of the graph $\Gamma$:

$$\mathcal{C}^n(\Gamma) := (\Gamma \times \cdots \times \Gamma) - \Delta.$$

As before, to navigate safely on a graph, one must construct appropriate paths that remain strictly within $\mathcal{C}^n(\Gamma)$ and are repulsed by any boundaries near $\Delta$. But several problems seem to prevent an analogous solution, including the following:

(1) What do these spaces look like?
(2) How does one resolve an impending collision?

A significant difference between this setting and that of motion planning on the full plane is that collisions within a track are no longer locally resolvable. Imagine that the aisles of a grocery store are only as wide as the shopping carts, so that passing another person is impossible. A store full of shoppers (using carts) would pose a difficult coordinated control problem. How can carts avoid a collision in the interior of an aisle? Clearly, at least one of the participants

must make a large-scale change in plans and back up to the end of the aisle. *The resolution of a collision on a graph is a non-local phenomenon*.

2.3. **Shape planning for metamorphic robots.** In recent years, several groups in the robotics community have been modeling and building **reconfigurable** or, more specifically, **metamorphic** robots (*e.g.*, [13, 16, 52, 61, 63, 72, 85, 86]). Such a system consists of multiple identical robotic cells in an underlying lattice structure which can disconnect/reconnect with adjacent neighbors, and slide, pivot, or otherwise locomote to neighboring lattice points following prescribed rules. There are as many models for such robots as there are researchers in the subfield: 2-d and 3-d lattices; hexagonal, square, and dodecahedral cells; pivoting or sliding motion: see, *e.g.*, [16, 52, 58, 61, 62, 10, 64, 85, 86] and the references therein. The common feature of these robots is an aggregate of lattice-based cells having prescribed local transitions from one shape to another.

The primary challenge for such systems is **shape-planning**: how to move from one shape to another via legal moves. One centralized approach [17, 18, 66] is to build a **transition graph** whose vertices are the various shapes and whose edges are elementary legal moves from one shape to the next. It is easily demonstrated that the size of this graph is exponential in the number of cells. For this and other reasons, the transition graph makes a poor model of a configuration space for these systems.

2.4. **Digital microfluidics.** An even better physical instantiation of the previous system arises in recent work on **digital microfluidics** (see, *e.g.*, the work of Fair in [27, 28]). In digital microfluidics, small (appx. 1mm diameter) droplets of fluid can be quickly and accurately manipulated in an inert oil suspension between two plates. The plates are embedded with a grid of wires. Droplet manipulation is performed via **electrowetting** — a process that exploits dynamic surface tension effects to propel a droplet. Applying a current in a particular manner through the grid drives the droplet a discrete distance along the wire grid. The goal of this is to create an efficient "lab on a chip" in which droplets of various chemicals or biological agents can be positioned, mixed, and then directed to the appropriate outputs. Using the grid, one can manipulate many droplets in parallel.

This scenario is reminiscent of manipulating robotics on a factory floor, or, rather, on a 1-dimensional grid within the floor. The chief differences lie in (1) scale; (2) the discrete nature of motion; and (3) the fact that droplets are sometimes allowed to collide. Indeed, digital chemical and biochemical reactions are performed precisely by forcing two droplets to collide, then mix (by a rapid oscillatory motion), then split.

The fundamental problem of microfluidic control — how to locally manipulate an ensemble of droplets to effectuate a global result on the grid — is entirely commensurate with the motion-planning problems in robotics stated above. It

is this passage from local motion rules in the presence of collision constraints which makes this (and many other contemporary problems in systems engineering) a prime candidate for configuration space methods. One feels that there is a sensible topological configuration space lurking beneath the surface of this and many other problems: see also §8.

## 3. Configuration spaces of graphs

Motivated by the applications to robot motion planning, we consider the configuration spaces of points on a graph.

The most obvious difference between this problem and the problem of $\mathcal{C}^n(\mathbb{R}^2)$ is that $\mathcal{C}^n(\Gamma)$ is *not* a manifold: that is, you cannot hope that every point has a neighborhood that is locally homeomorphic to a Euclidean space. Indeed, if we ignore trivial graphs that are homeomorphic to a line segment or a circle, then the graph itself is not locally Euclidean and products of the graph still share this feature.

### 3.1. **Visualization.** It is best to consider some examples that can be visualized.

*Example* 4. [$\mathcal{C}^2(\mathsf{Y})$]

Let $\mathsf{Y}$ denote the graph with three edges obtained by attaching three edges to a central vertex. The space $\mathcal{C}^2(\mathsf{Y})$ is a subset of $\mathsf{Y} \times \mathsf{Y}$, this product consisting of nine squares or glued together. Of these, six correspond to configurations in which the two robots are on distinct edges of $\mathsf{Y}$. Since there are three edges in $\mathsf{Y}$, the remaining configurations, in which both robots are on the same edge, yield three square cells, each of which is divided by the diagonal $\Delta$ into a pair of triangular cells. Thus there are six triangular 2-cells corresponding to the configurations in which both robots are on the same edge, but at distinct locations. By enumerating the behaviors of each of these 2-cells, one can make the identifications to arrive at the space given in Figure 1.

*Exercise* 5. Choose two distinct points on $\mathsf{Y}$. Now, using two fingers, execute a motion that exchanges the two chosen locations without collision. Draw this motion as a path on the configuration space of Figure 1.

*Example* 6. [$\mathcal{C}^2(\mathsf{Q})$]

Let $\mathsf{Q}$ denote the graph with three edges obtained from $\mathsf{Y}$ by gluing two boundary vertices together. One method of constructing $\mathcal{C}^2(\mathsf{Q})$ would be to first remove the configurations in which both robots are on the vertices to be glued. Then identify those portions of the boundary of $\mathcal{C}^2(\mathsf{Y})$ that have a robot at the vertices to be glued in $\mathsf{Q}$, and glue these portions of $\mathcal{C}^2(\mathsf{Y})$ together. The result, although a very simple configuration space, is already somewhat difficult to visualize: we illustrate the space, embedded in $\mathbb{R}^3$, in Figure 2 [left]. Each of the three "punctures" corresponds to a collision of the robots at one of the

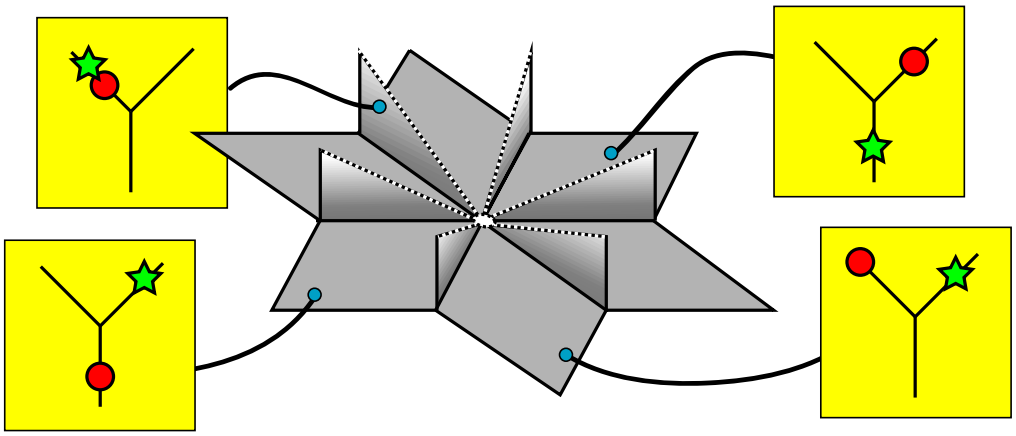

FIGURE 1. The configuration space $\mathcal{C}^2(Y)$ of Example 4 embedded in $\mathbb{R}^3$. Dotted lines refer to edges that lie on the diagonal $\Delta$. Note that the central vertex is deleted.

three vertices. The six dotted edges are the images of the diagonal curves from Figure 1.

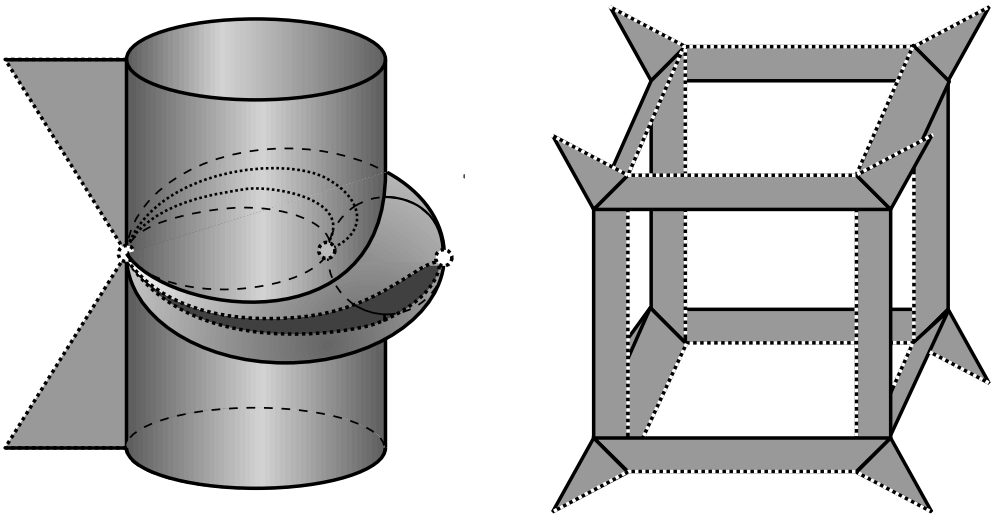

FIGURE 2. The configuration spaces $\mathcal{C}^2(Q)$ [left] and $\mathcal{C}^2(X)$ [right]. Dotted lines refer to vestiges of the diagonal $\Delta$.

*Example 7.* [$\mathcal{C}^2(X)$]

Increasing the incidence number of the central vertex complicates the configuration space. Consider X, a radial tree of four edges emanating from a central vertex. The visualization of $\mathcal{C}^2(X)$ is a bit more involved and requires some work to obtain. For the purpose of stimulating curiosity, we include this configuration space as Figure 2 [right].

3.2. **Simplification: deformation.** For simple graphs, though the configuration spaces illustrated earlier are visualizable, the full representation is not entirely elegant. Removal of the diagonal $\Delta$ yields a space which is not compact, and which furthermore has some "dangling" features. In particular, many of the examples of configuration spaces we have considered are not parsimonious with respect to dimension: the configuration space can be deformed to a subset of strictly lower dimension. Otherwise said, the number of degrees of freedom of the system (the configuration space dimension) is not always fully needed to capture the essential features of the space.

*Example* 8. Note that the spaces illustrated in Figs. 1-2 can all be deformation retracted to a 1-dimensional subspace: the "essential dimension" of these spaces is one.

The computation of this essential dimension is encapsulated in the following result:

**Theorem 9** ([37])**.** *Given a graph $\Gamma$ having $V$ vertices of valence greater than two, the space $\mathcal{C}^n(\Gamma)$ deformation retracts to a subcomplex of dimension at most $V$.*

The proof of this result in [37] is by inductive (and inelegant) manipulation: a recent simplification appears in [33]. We illustrate the result by continuing the some of the examples of the previous subsection. Consider for each $k > 2$ the radial $k$-prong tree $\mathsf{T}_k$ with vertices $\{v_i\}_0^k$ and edges $\{e_i\}_1^k$ attaching the central vertex $v_0$ to the outer vertices $\{v_i\}_1^k$. For example, $\mathsf{T}_3 = \mathsf{Y}$, and $\mathsf{T}_4 = \mathsf{X}$.

Theorem 9 ensures that $\mathcal{C}^n(\mathsf{T}_k)$ deformation retracts to a 1-dimensional subcomplex — that is, a graph. Since the essential topological features of a graph are determined by its Euler characteristic, one need merely compute the number of vertices and edges to classify these spaces.

**Proposition 10.** *The braid group $P_n\mathsf{T}_k = \pi_1(\mathcal{C}^n(\mathsf{T}_k))$ is isomorphic to a free group on*

$$1 + (nk - 2n - k + 1)\,\frac{(n+k-2)!}{(k-1)!} \tag{3.1}$$

*generators.*

*Proof.* One derives a recursion relation for the Euler characteristic of $\mathcal{C}^n(\mathsf{T}_k)$ by holding fixed the point on the $k^{th}$ edge of $\mathsf{T}_k$ which is farthest from the central node. With some work, one obtains:

$$\chi(\mathcal{C}^n(\mathsf{T}_k)) = \chi(\mathcal{C}^n(\mathsf{T}_{k-1})) + n\chi(\mathcal{C}^{n-1}(\mathsf{T}_k)) - n\prod_{i=1}^{n-1}(k+i-2). \tag{3.2}$$

The first term on the right hand side comes from the case where there is no point in the interior of the $k^{th}$ edge. The second term comes from fixing one of the $n$ points in the interior of this dedicated edge. The last term comes from fixing

this point at the central vertex of $\mathsf{T}_k$: this subspace of the configuration space has the homotopy type of a discrete set since no other points can pass through the central vertex. Each such component contributes one edge in the deformation retracted space and hence contributes a $-1$ to the value of $\chi(\mathcal{C}^n(\mathsf{T}_k))$.

The seed for this recursion relation is the fact that $\mathcal{C}^1(\mathsf{T}_k) \simeq \mathsf{T}_k \simeq \{pt\}$. Solving (3.2) yields

$$\chi(\mathcal{C}^n(\mathsf{T}_k)) = -\left(nk - 2n - k + 1\right) \frac{(n+k-2)!}{(k-1)!},$$

which, in turn, implies that the configuration space is homotopic to a wedge of $1 - \chi$ circles. □

The factorial growth of the rank in $n$ is due to the fact that we label the $n$ robots on $\mathsf{T}_k$. If one considers the unlabeled configuration spaces, then $\chi$ is reduced by a factor of $n!$.

It is worth emphasizing that while the control problem of robots on a graph is rather intuitive for two robots, it quickly builds in complexity. Since the dimension alone makes most configuration spaces nearly impossible to visualize, Theorem 9 is quite helpful — the "essential" dimension of the configuration space is independent of the number of robots on the graph. For the graph $\mathsf{T}_k$, Theorem 9 implies that there is a one-dimensional *roadmap* that gives a perfect representation of the configuration space: no topological data are lost. Since the proof of Theorem 9 is constructive, one can use standard algorithms for determining shortest paths on a graph to develop efficient path planning for multiple robots on $\mathsf{T}_k$ via the roadmap.

3.3. **Simplification: simplicial.** The simplification expressed in Theorem 9 is not so easy to visualize. In this section, we outline a different simplification process to a topologically equivalent subset of the configuration space which has a few distinctions. First, it carries a natural simplicial structure. Second, the simplification process is easier to define explicitly and track. This simplicial model makes for easy visualization of some examples, especially radial trees.

It is easiest to explain and visualize examples in the context of unlabeled configuration spaces $\mathcal{UC}^n$. For the remainder of this subsection, we work exclusively in the unlabeled setting.

Consider again the radial $k$-prong tree $\mathsf{T}_k$ with vertices $\{v_i\}_0^k$ and edges $\{e_i\}_1^k$ attaching the central vertex $v_0$ to the outer vertices $\{v_i\}_1^k$. Place a Euclidean metric on each $e_i$ of total length one.

To establish convenient coordinates on $\mathcal{UC}^n(\mathsf{T}_k)$, consider any bi-indexed sequence $\mathbf{x} := \{x_j^i\}$ where $x_j^i \in [0,1]$ is the distance from the vertex $v_j$ to the $i^{th}$ token on $\overline{e_j}$ closest to $v_0$. If there are no tokens on $\overline{e_j}$, any reference to $x_j^i$ returns zero. It is clear that as long as (1) $x_j^i \neq x_j^{i'}$ for all $i \neq i'$; and (2) if $x_j = 1$ for some

$j$ then $x_j^1 = 1$ for all $j$; then, the sequence $\mathbf{x}$ determines a well defined point of $\mathcal{UC}^n(\mathsf{T}_k)$ and that all points of this configuration space can be so specified.

One can use an explicit vector field to specify a projection of $\mathcal{UC}^n(\mathsf{T}_k)$ to a simplicial complex. Consider the vector field:

$$\frac{d}{dt}x_j^i = \begin{cases} x_j^i(x_j^i - 1) & i \neq 1 \\ \Phi \cdot x_j^i(x_j^i - 1) & i = 1 \end{cases} \quad \text{where} \quad \Phi := \frac{\sum_{\ell=1}^k x_\ell^1 - 1}{1 + \left(\sum_{\ell=1}^k x_\ell^1 - 1\right)^2} \,. \tag{3.3}$$

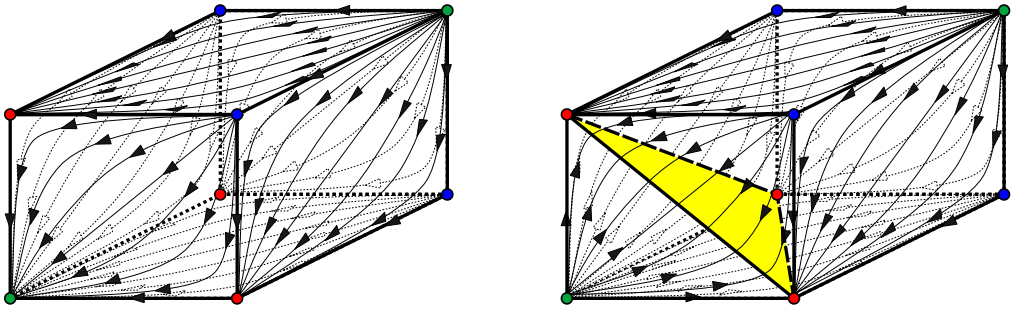

FIGURE 3. [left] The vector field $x_j^1(x_j^1 - 1)$ acts on a cube of "inmost" points; [right] the rescaled field yields a deformation retraction of the cube (minus those portions of the boundary where $x_j^1 = 1$ but $x_{j'}^1 \neq 1$ for some $j, j'$) to the unit simplex.

*Exercise* 11. Show that this vector field determines a projection map from $\mathcal{UC}^n(\mathsf{T}_k)$ to a simplicial complex $\mathcal{B}$ whose fibers are contractible.

*Example* 12. For $n = 2$, the simplicial model $\mathcal{B}$ is always a complex of dimension one. In fact, $\mathcal{B}$ is precisely the 1-skeleton of the standard $(k-1)$-simplex. The vertices correspond to configurations with both tokens on the same edge of $\mathsf{T}_k$; the edges correspond to configurations with two tokens on different edges of $\mathsf{T}_k$.

*Example* 13. For $k = 3$, the simplicial model $\mathcal{B}$ is a complex of dimension at most two. These complexes are pictured in Figure 4.

The general case is not difficult to discern: one inducts on $n$ and $k$ as in the example of Fig. 5. Note that the simplicial model $\mathcal{B}$ will have principal $p$-simplices for all $0 \leq p \leq \min\{n-1, k-1\}$. Thus, this simplification reduces dimension to a subset of dimension potentially far below that of $n$. This is a foreshadowing of Theorem 9.

More general trees can be handled by induction and a *graph of spaces* construction.

*Exercise* 14. Give a simplicial model for the configuration space of points on a tree which is the wedge product of two copies of $\mathsf{T}_k$ along a common boundary vertex.

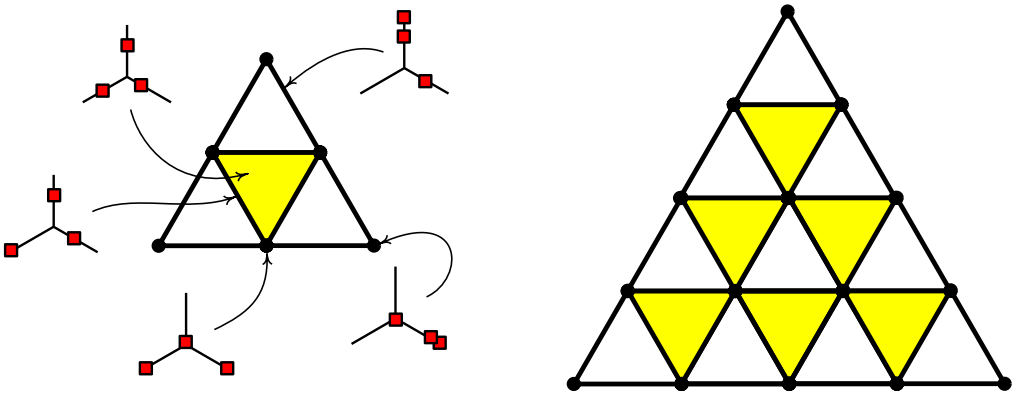

FIGURE 4. Examples of $\mathcal{B}$ for $k = 3$ and $n = 3$ [left], along with samples of representative configurations in the fiber; [right] $k = 3$ and $n = 5$.

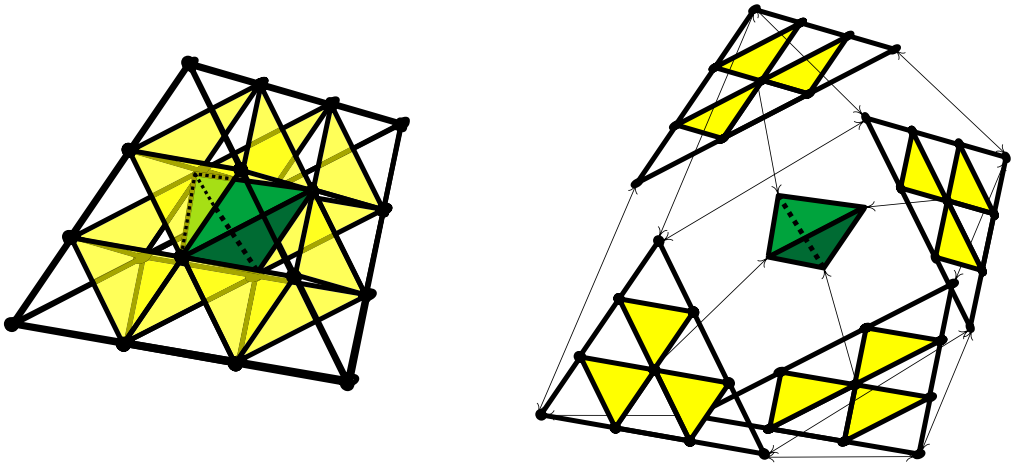

FIGURE 5. An example of $\mathcal{B}$ for $k = 4$ and $n = 4$ [left]. There is precisely one solid 3-simplex in the center of the complex, surrounded by four identified copies of the simplicial model for $k = 3$ and $n = 4$ [right].

*Exercise* 15. Can you draw the simplicial model associated $\mathcal{C}^2(\mathsf{T}_k)$ based on that of $\mathcal{UC}^2(\mathsf{T}_k)$?

3.4. **Simplification: discretization.** The last example of simplification for configuration spaces is a reduction not to the smallest possible dimension subspace (as in §3.2), nor to a simplicial complex via a projection (as in §3.3), but rather to a subset that approximates the space with cubes. This simplification method has a fundamental impact in construction configuration spaces for a number of related problems in robotics: we thus relegate it to its own section.

## 4. DISCRETIZATION

In this section, we approximate configuration spaces of graphs with **cubical complexes** — cell complexes which are built from finite-dimensional Euclidean cubes and whose gluing maps, like those of a simplicial complex, are well-behaved with respect to gluing along faces, see, *e.g.*, [65, 8]. This **discretization** method will be seen to lead us to several interesting mathematical detours, as well as a deeper understanding of what configuration spaces are and how they arise in the applied settings of §2.

4.1. **Cubes and collisions.** Any graph $\Gamma$ comes equipped with a cellular structure: 0-cells (vertices) and 1-cells (edges). The $n$-fold cross product of $\Gamma$ with itself inherits a cell structure, each cell being a product of $n$ (not necessarily distinct) cells in $\Gamma$. However, the configuration space does not quite have a natural cell structure, since the diagonal $\Delta$ slices through all product cells with repeated factors. Notice, however, that in several of the previous examples, these partial cells dangle "inessentially" and could be collapsed onto a more "essential" skeleton of the configuration space.

This skeleton can be specified as follows [1]. Consider the *discretized configuration space* of $\Gamma$, denoted $\mathcal{D}^n(\Gamma)$, defined as $(\Gamma \times \cdots \times \Gamma) - \tilde{\Delta}$, where $\tilde{\Delta}$ denotes the set of all product cells in $\Gamma \times \cdots \times \Gamma$ whose closures intersect the diagonal $\Delta$. Equivalently, we can describe $\mathcal{D}^n(\Gamma)$ as the set of configurations for which, given any two robots on $\Gamma$ and any path in $\Gamma$ connecting them, the path contains at least one entire edge. Thus, instead of restricting robots to be at least some intrinsic distance $\epsilon$ apart (i.e., removing an $\epsilon$ neighborhood of $\Delta$), one now restricts robots on $\Gamma$ to be "at least one full edge apart." Note that $\mathcal{D}^n(\Gamma)$ is a subcomplex of the cubical complex $\Gamma^n$ and a subset of $\mathcal{C}^n(\Gamma)$ (it does not contain "partial cells" that arise when cutting along the diagonal), and is, in fact, the largest subcomplex of $\Gamma^n$ that does not intersect $\Delta$.

With this natural cell structure, one can think of the vertices (0-cells) of $\mathcal{D}^n(\Gamma)$ as "discretized" configurations — arrangements of labeled tokens at the vertices of the graph. The edges of $\mathcal{D}^n(\Gamma)$, or 1-cells, tell us which discrete configurations can be connected by moving one token along an edge of $\Gamma$. Each 2-cell in $\mathcal{D}^n(\Gamma)$ represents two physically independent edges: one can move a pair of tokens independently along disjoint edges. A $k$-cell in $\mathcal{D}^n(\Gamma)$ likewise represents the ability to move $k$ tokens along $k$ disjoint edges in $\Gamma$ in an independent manner.

Returning to Figure 1, discretizing $\mathcal{C}^2(\mathsf{Y})$ removes much of the space. For example, the triangular two-dimensional cells represent configurations in which both robots are on the interior of the same edge. Since they are not "one full edge apart", these cells are deleted. The same is true of all the other two-dimensional cells that represent robots in the interior of separate edges. Which configurations of two robots on $\mathsf{Y}$ are separated by a full edge?

*Exercise* 16. Show that discretizing the configuration spaces of Examples 4 through 7 yields the configuration spaces of Figure 6. How well do these spaces "approximate" the configuration spaces?

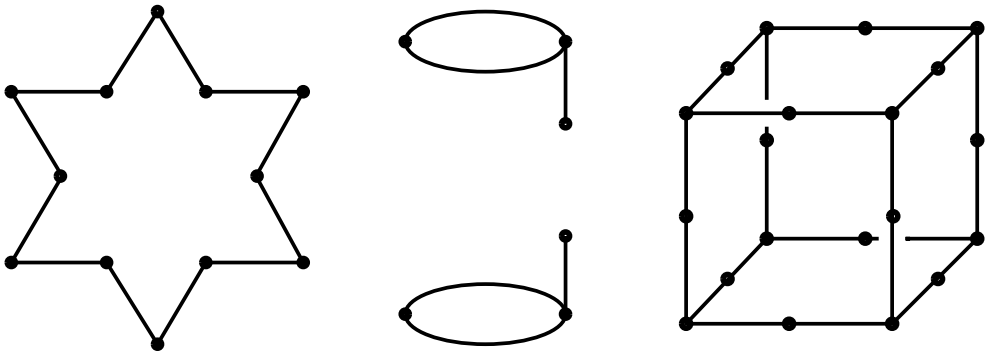

FIGURE 6. The discretizations of the configuration spaces in Examples 1-3 (left to right).

One could compute the discretization of Example 4 in a less direct manner that generalizes to some lovely examples to follow. For this example, simple counting reveals that the space $\mathcal{D}^2(\mathsf{Y})$ possesses twelve 0-cells (both robots are at distinct vertices of Y), twelve 1-cells (one robot is at a vertex and the other is on an edge whose closure does not contain said vertex), and zero 2-cells (since any two edges intersect at the central vertex). With a little thought, one can see that $\mathcal{D}^2(\mathsf{Y})$ is a connected manifold: each zero-cell connects to exactly two 1-cells, and all of the 1-cells are joined end-to-end cyclically. It follows that $\mathcal{D}^2(\mathsf{Y})$ is a topological circle, obtained by deleting all the near-diagonal cells from $\mathcal{C}^2(\mathsf{Y})$ in Figure 1. The discretization operation yields a homotopy-equivalent subcomplex of $\mathcal{C}^2(\mathsf{Y})$. However, this is certainly not the case for the discretization of $\mathcal{C}^2(\mathsf{Q})$, which becomes disconnected.

4.2. **Manifold examples.** Combinatorial arguments like those above can sometimes determine the discretized configuration space, even when the full configuration space is hidden from view. The following are some surprising examples of interesting spaces that arise as the discretized configuration space of non-planar graphs [1].

*Example* 17. Consider the complete graph $K_5$ pictured in Figure 7 [left]. The discretized configuration space of two robots on this graph is a two-dimensional complex. A simple counting argument reveals the cell-structure.

Each 0-cell corresponds to a configuration in which the two robots are at distinct vertices. Since $K_5$ has five vertices, there are exactly $(5)(5-1) = 20$ such 0-cells. (There is no vertex where two edges cross in the picture; there are vertices only at the corners of the pentagon.) Each 1-cell corresponds to a configuration

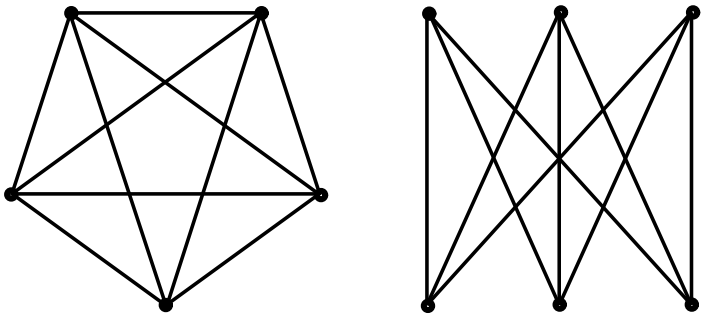

FIGURE 7. The non-planar graphs $K_5$ [left] and $K_{3,3}$ [right].

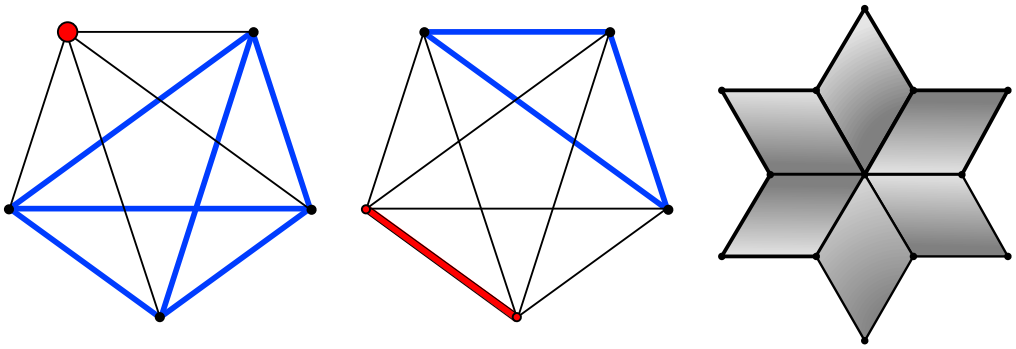

FIGURE 8. [left] For every vertex in the space $K_5$ there are six disjoint edges. Likewise [middle] for each edge there are three totally disjoint edges. In $\mathcal{D}^2(K_5)$, these cells fit together to form a locally Euclidean two-dimensional complex [right].

in which one robot is at a vertex and the other is on an edge whose endpoints do not include the vertex already occupied. From the diagram of $K_5$ one counts that there are $(2)(5)(6) = 60$ such 1-cells, as in Figure 8 [left]. The factor of two is present because we label the two robots on $K_5$. Each 2-cell corresponds to a configuration in which the two robots occupy edges whose closures are disjoint. Again, from the diagram (and Figure 8 [middle]) one counts that there are $(10)(3) = 30$ such 2-cells in the complex.

One then demonstrates that each edge borders a pair of 2-cells preserving an orientation and that each vertex is incident to six edges, as in Figure 8 [right]. Also, the space $\mathcal{D}^2(K_5)$ is connected: one can move from any configuration to any other. Thus $\mathcal{D}^2(K_5)$ is a connected orientable surface, and the classification theorem for surfaces implies that the space is determined uniquely up to homeomorphism by the Euler characteristic:

$$\chi(\mathcal{D}^2(K_5)) := \#\text{faces} - \#\text{edges} + \#\text{vertices} = 30 - 60 + 20 = -10. \tag{4.1}$$

Thus, $\mathcal{D}^2(K_5)$ is a closed orientable surface of genus $g = 1 - \frac{1}{2}\chi = 6$.

It is not at all obvious that the motion of two robots on this graph should produce a genus six surface. Obtaining a manifold is surprising enough, but a manifold with genus larger than one is at odds with the notion that all of the interesting topology in these spaces is "localized" in configurations about a vertex.

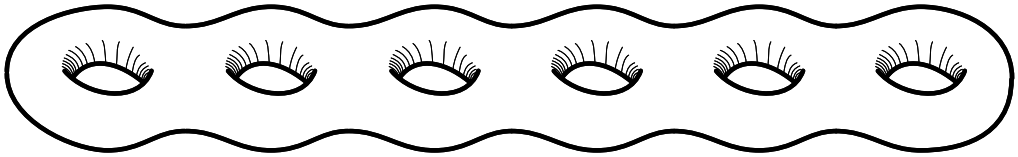

FIGURE 9. The space $\mathcal{D}^2(K_5)$ is homeomorphic to a closed orientable surface of genus six.

*Exercise* 18. For $K_5$ as above and $K_{3,3}$ the complete bipartite graph of Figure 7 [right], show the following using combinatorics and the Euler characteristic:

(1) $\mathcal{D}^2(K_{3,3})$ is a closed orientable surface of genus four.
(2) $\mathcal{D}^3(K_5)$ is a closed orientable surface of genus 16.
(3) $\mathcal{D}^4(K_{3,3})$ is a closed orientable surface of genus 37.

For the latter two, try tracking the holes instead of the robots, and be careful about labels!

4.3. **Everything that rises must converge.**

*Question* 19. When does the discretized configuration space of a graph represent an accurate approximation to the topological configuration space? That is, when is $\mathcal{D}^n(\Gamma)$ a deformation retract of $\mathcal{C}^n(\Gamma)$?

This question was answered definitively by Abrams in his 2000 thesis.

**Theorem 20** ([1])**.** *For any $n > 1$ and any graph $\Gamma$ with at least $n$ vertices, the inclusion $\mathcal{D}^n(\Gamma) \hookrightarrow \mathcal{C}^n(\Gamma)$ is a homotopy equivalence if and only if*

(1) *Each path between distinct vertices of valence not equal to two passes through at least $n-1$ edges; and*
(2) *Each path from a vertex to itself that cannot be shrunk to a point in $\Gamma$ passes through at least $n+1$ edges.*

Thus, the braid group of two strands on $K_5$ is the fundamental group of the closed oriented surface of genus 6: a hyperbolic group. This is surprising, and differs starkly from the intuition of Artin braid groups.

4.4. **Curvature.** Notice that all of the interesting examples of discretized configuration spaces of graphs have the homotopy type of a graph or of a surface of nonzero genus.

*Question* 21. Are (discretized) configuration spaces of graphs ever spheres? Are they ever manifolds of dimension greater than two?

The answers to these questions require a detour from topology to geometry. The scenic route is worth the time.

The spaces $\mathcal{D}^n(\Gamma)$ can be given a natural piecewise Euclidean geometry inherited from (1) the path metric on $\Gamma$, combined with (2) the flat product metric on $\Gamma^n$. Every $k$-cube of $\mathcal{D}^n(\Gamma)$ thus has the geometry of a Euclidean cube.[2]

The critical concept is that of a metric space which is **nonpositively curved**. This notion of curvature for metric spaces is both old and rich, dating back to work of Alexandrov, Rauch, Hadamard, Toponogov, Cartan, and others. We modify and specialize the definitions to the class of Euclidean cubical complexes: see [8] for a more robust treatment.

Let $X$ denote a piecewise Euclidean cubical complex. Though it may seem ironic to talk about the curvature of a space built from decidedly flat pieces, there is, nevertheless, a great deal of positive or negative curvature that can hide in the corners of the cubes.

**Definition 22.** A piecewise Euclidean cube complex $X$ is **nonpositively curved** or **NPC** if, for any sufficiently small geodesic triangle $\triangle PQR$ in $X$, the sum of the angles $\angle RPQ + \angle PQR + \angle QRP$ is no greater than $\pi$.

A geodesic triangle is a set of three points $\{P, Q, R\}$ together with geodesics between them. These edges are piecewise linear (since $X$ is piecewise Euclidean), and, consequently, the angles at the three vertices are well-defined. The intuition behind this (non-standard!) definition of NPC is that on a flat Euclidean plane, the angles of a triangle sum to $\pi$, whereas on a positively curved sphere, the angle sum is greater than $\pi$, and on a negatively curved hyperbolic space, the angle sum is less than $\pi$. The *"sufficiently small"* portion of Definition 22 is meant to ensure that the triangle is a contractible loop in $X$.

We note that the definition of NPC extends beyond piecewise Euclidean spaces to arbitrary metric spaces for which local geodesic paths exist. NPC spaces are of fundamental importance in geometric group theory: see [8] for an extensive introduction. They are local versions of **CAT(0) spaces**, another foundational class of geometric spaces.

There is a well-known combinatorial approach due to Gromov for determining when a cubical complex is nonpositively curved in terms of vertex links.

**Definition 23.** Let $X$ denote a piecewise Euclidean cubical complex and let $v$ denote a vertex of $X$. The **link** of $v$, $\text{LINK}[v]$, is defined to be the abstract simplicial complex whose $k$-dimensional simplices are the $(k+1)$-dimensional cubes incident to $v$ with the natural boundary relationships.

[2]It may be a rectangular prism, if the edge lengths differ. For this article, we will assume uniform lengths on all edges.

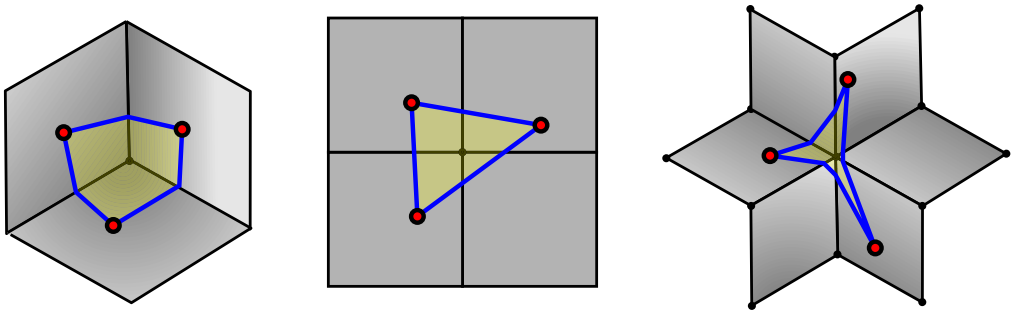

FIGURE 10. Geodesic triangles in piecewise Euclidean cube complexes can detect vertices of positive (left), zero (center), or negative (right) curvature based on angle sums.

Certain global topological features of a metric cubical complex are completely determined by the local structure of the vertex links: a theorem of Gromov [42] asserts that a finite dimensional Euclidean cubical complex is NPC if and only if the link of every vertex is a flag complex without digons. Recall: a **digon** is a pair of vertices connected by two edges, and a **flag complex** is a simplicial complex which is maximal among all simplicial complexes with the same 1-dimensional skeleton. Gromov's theorem permits an elementary proof of the following general result.

**Theorem 24** ([1, 77])**.** *$\mathcal{D}^n(\Gamma)$ is NPC for any $n$ and any $\Gamma$.*

*Exercise* 25. Prove it using Gromov's criterion. See Theorem 40 for a complete proof of a more general setting.

Thanks to the body of knowledge about NPC spaces, Theorem 24 combined with Theorem 20 yield a number of important corollaries:

**Corollary 26.** *Configuration spaces of graphs are aspherical (all higher homotopy groups vanish) and their fundamental groups are torsion-free.*

## 5. RECONFIGURABLE SYSTEMS

The discretization in §4 has a far-reaching generalization which provides models for the many discretely actuated systems in robotics and beyond surveyed in §2.

5.1. **Generators and relations.** A reconfigurable system is a collection of states on a graph, where each state is thought of as a vertex labeling function. Any state can be modified by local rearrangements, these local changes being rigidly specified. We distinguish between the amount of information needed to determine the legality of an elementary move (the "support" of the move) and the precise subset on which the reconfiguration physically occurs (the "trace" of the move).

**Definition 27.** Fix $\mathcal{A}$ to be a set of labels. Fix $\mathcal{G}$ to be a graph. A **generator** $\phi$ for a local reconfigurable system is a collection of three objects:

(1) the **support**, $\mathrm{SUP}(\phi) \subset \mathcal{G}$, a subgraph of $\mathcal{G}$;
(2) the **trace**, $\mathrm{TR}(\phi) \subset \mathrm{SUP}(\phi)$, a subgraph of $\mathrm{SUP}(\phi)$;
(3) an *unordered* pair of **local states**

$$\mathbf{u}_0^{loc}, \mathbf{u}_1^{loc} : V(\mathrm{SUP}(\phi)) \to \mathcal{A},$$

which are labelings of the vertex set of $\mathrm{SUP}(\phi)$ by elements of $\mathcal{A}$. These local states must agree on $\mathrm{SUP}(\phi) - \mathrm{TR}(\phi)$: i.e.,

$$\mathbf{u}_0^{loc}\Big|_{\mathrm{SUP}(\phi)-\mathrm{TR}(\phi)} = \mathbf{u}_1^{loc}\Big|_{\mathrm{SUP}(\phi)-\mathrm{TR}(\phi)}. \tag{5.1}$$

All generators are assumed to be **nontrivial** in the sense that $\mathbf{u}_0^{loc} \neq \mathbf{u}_1^{loc}$.

**Definition 28.** A **state** is a labeling of the vertices of $\mathcal{G}$ by $\mathcal{A}$. A generator $\phi$ is said to be **admissible** at a state $\mathbf{u}$ if $\mathbf{u}|_{\mathrm{SUP}(\phi)} = \mathbf{u}_0^{loc}$. For such a pair $(\mathbf{u}, \phi)$, we say that the **action** of $\phi$ on $\mathbf{u}$ is the new state given by

$$\phi[\mathbf{u}] := \begin{cases} \mathbf{u} & : \text{ on } \mathcal{G} - \mathrm{SUP}(\phi) \\ \mathbf{u}_1^{loc} & : \text{ on } \mathrm{SUP}(\phi) \end{cases}, \tag{5.2}$$

*Remark* 29. Since the local states of each generator are unordered, it follows that any generator $\phi$ which is admissible at a state $\mathbf{u}$ is also admissible at the state $\phi[\mathbf{u}]$, and that $\phi[\phi[\mathbf{u}]] = \mathbf{u}$.

**Definition 30.** A **reconfigurable system** on $\mathcal{G}$ consists of a collection of generators and a collection of states closed under all possible admissible actions.

Given a pair of local moves, one can understand intuitively what it means for them to act "independently." One way to codify this in a formal reconfigurable system is as follows:

**Definition 31.** In a reconfigurable system, a collection of generators $\{\phi_{\alpha_i}\}$ is said to **commute** if

$$\mathrm{TR}(\phi_{\alpha_i}) \cap \mathrm{SUP}(\phi_{\alpha_j}) = \emptyset \quad \forall i \neq j. \tag{5.3}$$

This definition is the reason for the distinction between a local move's support and its trace. The trace is where the move 'happens' and when execution can disrupt other moves trying to execute simultaneously.

Commutativity connotes physical independence.

5.2. **State complexes.** The reconfigurable systems of the previous section possess a configuration space which naturally generalizes the example of the discretized configuration space of a graph. Recall, $\mathcal{D}^n(\Gamma)$ is a cubical complex where edges denote moves (or, in our language, 'generators') and cubes correspond to a commutative collection of physically independent moves.

We define the state complex of a reconfigurable system to be the cube complex with an abstract $k$-cube for each collection of $k$ admissible commuting generators:

**Definition 32.** The **state complex** $\mathcal{S}$ of a local reconfigurable system is the following abstract cubical complex. Each abstract $k$-cube $e^{(k)}$ of $\mathcal{S}$ is an equivalence class $[\mathbf{u}; (\phi_{\alpha_i})_{i=1}^k]$ where

(1) $(\phi_{\alpha_i})_{i=1}^k$ is a $k$-tuple of commuting generators;
(2) $\mathbf{u}$ is some state for which all the generators $(\phi_{\alpha_i})_{i=1}^k$ are admissible; and
(3) $[\mathbf{u}_0; (\phi_{\alpha_i})_{i=1}^k] = [\mathbf{u}_1; (\phi_{\beta_i})_{i=1}^k]$ if and only if the list $(\beta_i)$ is a permutation of $(\alpha_i)$ and $\mathbf{u}_0 = \mathbf{u}_1$ on the set $\mathcal{G} - \bigcup_i \mathrm{SUP}(\phi_{\alpha_i})$ .

The boundary of each abstract $k$-cube is the collection of $2k$ faces obtained by deleting the $i^{th}$ generator from the list and using $\mathbf{u}$ and $\phi_{\alpha_i}[\mathbf{u}]$ as the ambient states, for $i = 1 \dots k$. Specifically,

$$\partial[\mathbf{u}; (\phi_{\alpha_i})_{i=1}^k] = \bigcup_{i=1}^{k} \left( [\mathbf{u}; (\phi_{\alpha_j})_{j\neq i}] \cup [\phi_{\alpha_i}[\mathbf{u}]; (\phi_{\alpha_j})_{j\neq i}] \right) \tag{5.4}$$

The weak topology is used for reconfigurable systems which are not locally finite. In the locally finite case, the state complex is a locally compact cubical complex.

It follows from repeated application of Remark 29 that the $k$-cells are well-defined with respect to admissibility of actions.

*Exercise* 33. Show that the 1-skeleton of a state complex $\mathcal{S}$ is precisely the transition graph associated to the reconfigurable system.

5.3. **Examples.** The following examples come from [3, 41].

*Example* 34 (hex-lattice metamorphic robots). This reconfigurable system is based on the first metamorphic robot system pioneered by Chirikjian [16]. It consists of a finite aggregate of planar hexagonal units locked in a hex lattice, with the ability to pivot sufficiently unobstructed units on the boundary of the aggregate.

More specifically, $\mathcal{G}$ is a graph whose vertices correspond to hex lattice points and whose edges correspond to neighboring lattice points. The alphabet is $\mathcal{A} = \{0, 1\}$ with $0$ connoting an unoccupied site and $1$ connoting an occupied site. There is one type of generator, represented in Fig. 11[left], which generates a homogeneous system: this local rule can be applied to any translated or rotated position in the lattice. This generator allows for local changes in the topology of the aggregate (disconnections are possible). For physical systems in which this is undesirable — say, for power transmission purposes — one can choose a generator with larger support.

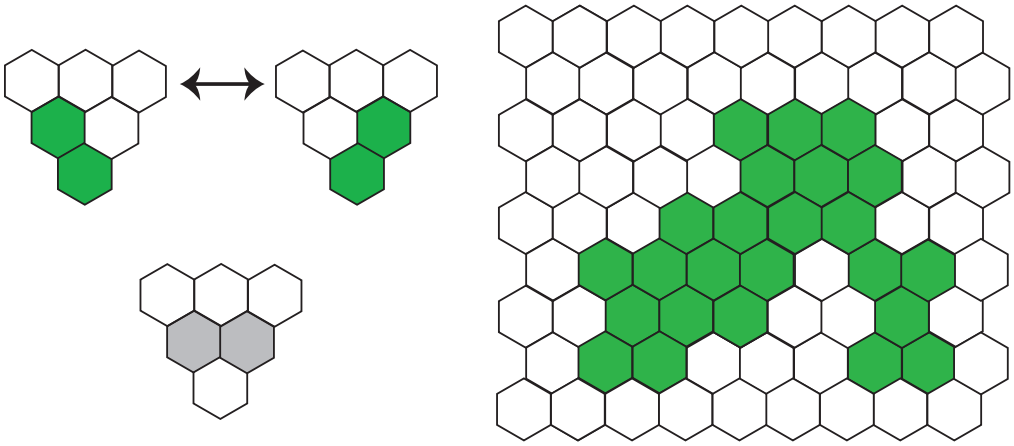

FIGURE 11. The generator for a 2-d hexagonal lattice system with pivoting locomotion. The domain is the graph dual to the hex lattice shown. Shaded cells are occupied, white are unoccupied. [left, top] The local states $\mathbf{u}_0^{loc}$ and $\mathbf{u}_1^{loc}$ are shown. [left, bottom] The support of the generator, with trace shaded. [right] A typical state in this reconfigurable system.

As an example of a state complex for this system, consider a workspace $\mathcal{G}$ consisting of three rows of lattice points with a line of occupied cells as in Fig. 12. This line of cells can "climb" on itself from the left and migrate to the right, one by one. The entire state complex is illustrated in Fig. 12[center]. Although the transition graph appears complicated, this state complex is contractible and remains so for any length channel.

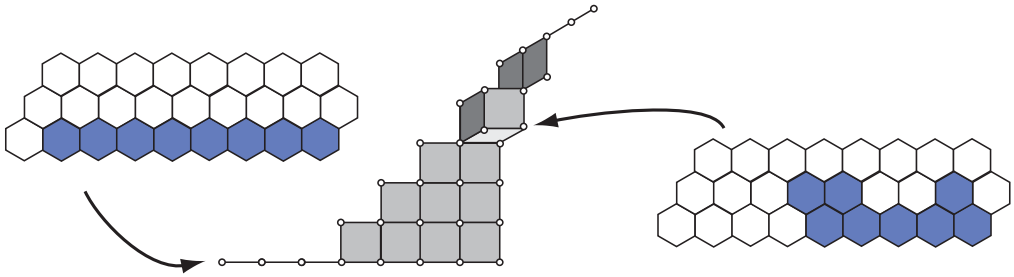

FIGURE 12. For a line of hexagons filing out of a constrained tunnel, the state complex is contractible.

*Example* 35 (2-d articulated planar robot arm). Consider as a domain $\mathcal{G}$ the lattice of edges in the first quadrant of the plane. This system consists of two types of generators, pictured in Fig. 13. The support of each generator is the union of eight edges as shown. The trace of each generator is as described in the

figure caption. Beginning with a state having $n$ vertical edges end-to-end, the reconfigurable system models the position of an articulated robotic arm which is fixed at the origin and which can (1) rotate at the end and (2) flip corners as per the diagram. This arm is **positive** in the sense that it may extend up and to the right only.

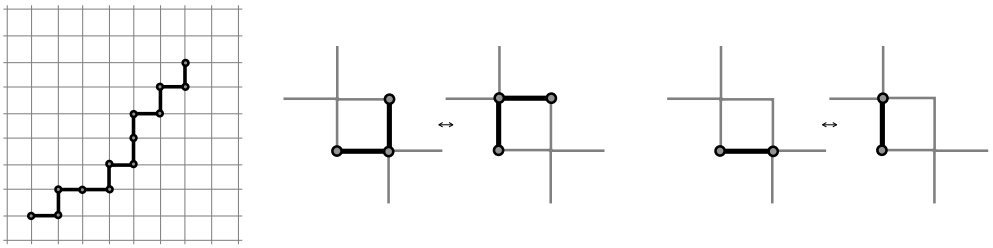

FIGURE 13. A positive articulated robot arm example [left] with fixed endpoint. One generator [center] flips corners and has as its trace the central four edges. The other generator [right] rotates the end of the arm, and has trace equal to the two activated edges.

The state complex in the case $n = 5$ is illustrated in Fig. 14. Note that there can be at most three independent motions (when the arm is in a "staircase" configuration); hence the state complex has top dimension three. In this case also, although the transition graph for this system is complicated, the state complex itself is contractible: this is the case for all lengths $n$.

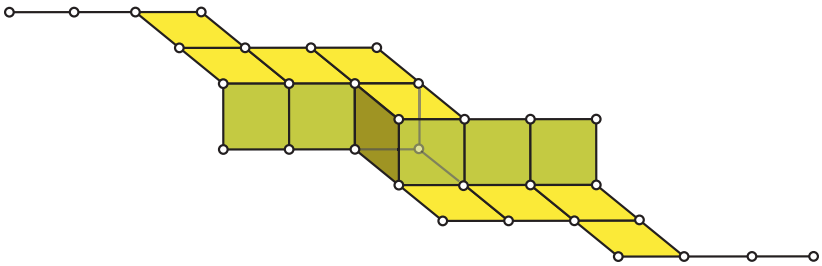

FIGURE 14. The state complex of a 5-link positive arm has one cell of dimension three, along with several cells of lower dimension.

*Example* 36 (2-d expansion-compression square system). Consider a planar square lattice workspace. This system will use an alphabet of labels $\{0, 1, =, \|\}$ whose interpretation is as follows: "0" means that a cell is unoccupied; "1" means that the cell is occupied by one module; "=" and "$\|$" imply that the cell is occupied by two modules compressed together in a horizontal or vertical orientation (resp.). The catalogue consists of six generators, illustrated in Fig. 15

(the lower two generators are only represented up to flips). The trace is equal to the support for the top two generators illustrated; for the bottom two generators, the trace is equal to the support minus the single square which remains unoccupied (label "0").

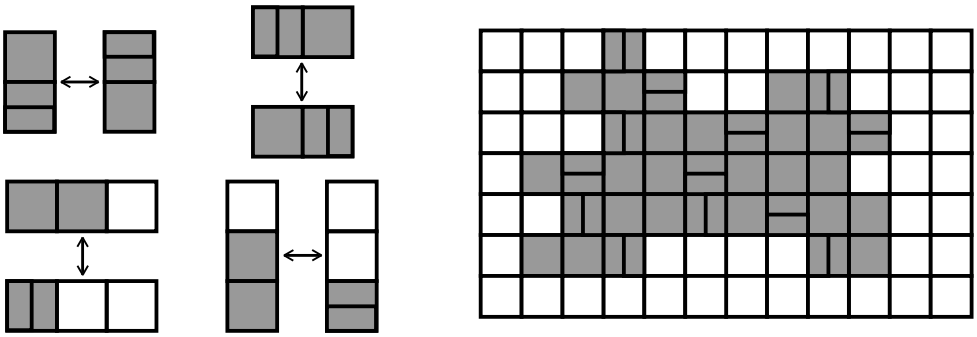

FIGURE 15. Generators for a simple compression-expansion system on the square planar lattice [left]; an example of a typical state [right].

This example is based on the Crystalline robots of Rus et al. [72]. Extensions to 3-d cubical systems and more elaborate motions can be accommodated with minor modifications.

*Example* 37 (configuration space of points on a graph). Consider a graph $\mathcal{G}$ and alphabet $\mathcal{A} = \{0, \ldots, n\}$ used to specify empty/occupied vertices. There are $n$ types of generators $\{\phi_i\}_1^n$ in this homogeneous system, one for each nonzero element of $\mathcal{A}$. The support and trace of each $\phi_i$ is precisely the closure of an (arbitrary) edge. The local states of this $\phi_i$ evaluate to $0$ on one of the endpoints and $i$ on the other. The homogeneous reconfigurable system generated from a state $\mathbf{u}$ on $\mathcal{G}$ having exactly one vertex labeled $i$ for each $i = 1, \ldots, n$ mimics an ensemble of $n$ distinct non-colliding points on the graph $\mathcal{G}$. If we reduce the alphabet to $\{0, 1\}$, then the system represents $n$ identical agents.

*Example* 38 (digital microfluidics). An even better physical instantiation of the previous system arises in digital microfluidics (recall §2.4). Representing system states as marked vertices ('droplets') on a graph is appropriate given the discrete nature of the motion by electrophoresis on a graph of wires. This adds a few new ingredients to the setting of the previous example. For $n$ different chemical agents, an alphabet of $\{0, \ldots, n\}$ is appropriate (the '0' connoting absence); however, a typical state may have many vertices with the same nonzero label (corresponding to the number of droplets of substance $i$ in use at a given time). Furthermore, it is possible to mix droplets by merging them together, rapidly oscillating along an edge, then splitting the mixed product. This leads to a new type of generator of the form $(i \text{ — } j) \Longleftrightarrow (k \text{ — } k)$.

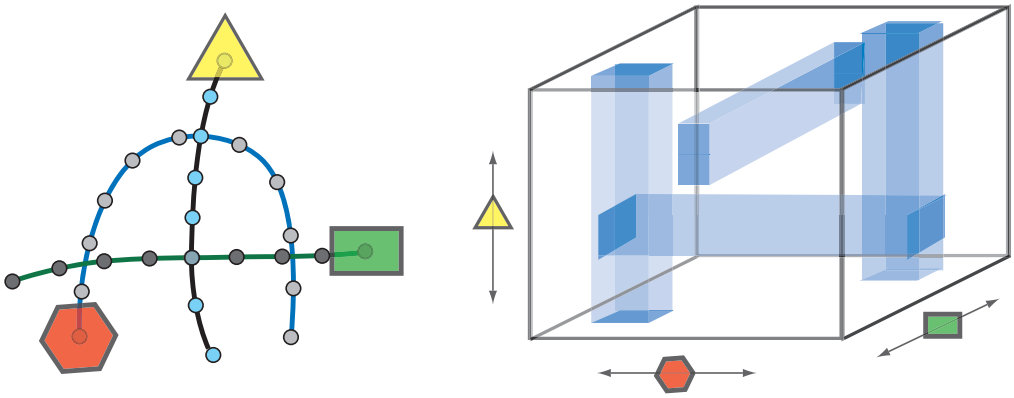

FIGURE 16. [left] A coordination problem with three robots translating on (discretized, intersecting) intervals. [right] The state complex approximation to the coordination space, with collision set shaded.

*Example* 39 (robot coordination). There is a broad generalization of configuration spaces of graphs developed in [40, 39] which has an interpretation as a state complex. We outline a simple example. Consider a collection of $n$ planar graphs $(\Gamma_i)_1^n$, each embedded in the plane of a common workspace (with intersections between different graphs permitted). On each $\Gamma_i$, a robot $R_i$ with some particular fixed size/shape is free to translate along $\Gamma_i$: one thinks of the graph as being a physical groove in the floor, or perhaps an electrified overhead guidewire. The **coordination space** of this system is defined to be the space of all configurations in $\prod_i \Gamma_i$ for which there are no collisions — the robots $R_i$ have no intersections.

We can approximate these coordination spaces by the following reconfigurable systems. Assume that each graph $\Gamma_i$ has been refined by adding multiple (trivial) vertices along the interiors of edges. We will approximate the robot motion by performing discrete jumps to neighboring vertices, much as in Example 37.

Let the underlying graph be $\mathcal{G} := \coprod \Gamma_i$, the disjoint union of the individual graphs. The generators for this system are as follows. For each edge $\alpha \in E(\Gamma_i)$, there is exactly one generator $\phi_\alpha$. The trace is the edge itself, $\text{TR}(\phi_\alpha) = \alpha$, and the generator corresponds to sliding the robot $R_i$ from one end of the edge to the other. The support, $\text{SUP}(\phi_\alpha)$ consists of the edge $\alpha \in E(\Gamma_i)$ along with any other edges $\beta$ in $\Gamma_j$ ($j \neq i$) for which the robot $R_j$ sliding along the edge $\beta$ can collide with $R_i$ as it slides along $\alpha$. The alphabet is $\mathcal{A} = \{0, 1\}$ and the local states for $\phi_\alpha$ have zeros at all vertices of all edges in the support, except for a single $1$ at the boundary vertices of $\alpha$ (these two boundary vertices yield the two local states, as in Example 37).

Any state for this reconfigurable system is one for which all vertices of each $\Gamma_i$ are labeled with zeros except for one vertex with a label $1$. The resulting state

complex is a cubical complex which approximates the cylindrical coordination space, as pictured in Fig. 16 [40, 39]. Of course, in the case where $\Gamma_i = \Gamma$ for all $i$ and the robots $R_i$ are sufficiently small, this reconfigurable system is exactly that of Example 37.

Other concrete examples of reconfigurable systems fall outside the realm of robotics applications. These include the discrete models of protein folding considered in [74, 75], spaces of phylogenetic trees [6], and parallel computation with shared resource constraints [67]. More abstract and algebraic examples abound.

## 6. CURVATURE AGAIN

In §**??**, a discrete curvature approach to discretized configuration spaces of graphs was seen to be extremely powerful. This theme remains true in the context of reconfigurable systems.

### 6.1. **State complexes are NPC.**

**Theorem 40.** *The state complex of any (locally finite) reconfigurable system is NPC.*

*Proof.* Let $\mathbf{u}$ denote a vertex of $\mathcal{S}$. Consider the link $\text{LINK}[\mathbf{u}]$. The 0-simplices of the $\text{LINK}[\mathbf{u}]$ correspond to all edges in $\mathcal{S}^{(1)}$ incident to $\mathbf{u}$; that is, actions of generators based at $\mathbf{u}$. A $k$-simplex of $\text{LINK}[\mathbf{u}]$ is thus a commuting set of $k+1$ of these generators based at $\mathbf{u}$.

We argue first that there are no digons in $\text{LINK}[\mathbf{u}]$ for any $\mathbf{u} \in \mathcal{S}$. Assume that $\phi_1$ and $\phi_2$ are admissible generators for the state $\mathbf{u}$, and that these two generators correspond to the vertices of a digon in $\text{LINK}[\mathbf{u}]$. Each edge of the digon in $\text{LINK}[\mathbf{u}]$ corresponds to a distinct 2-cell in $\mathcal{S}$ having a corner at $\mathbf{u}$ and edges at $\mathbf{u}$ corresponding to $\phi_1$ and $\phi_2$. By Definition 32, each such 2-cell is the equivalence class $[\mathbf{u}; (\phi_1, \phi_2)]$: the two 2-cells are therefore equivalent and not distinct.

To complete the proof, we must show that the link is a flag complex. The interpretation of the flag condition for a state complex is as follows: if at $\mathbf{u} \in \mathcal{S}$, one has a set of $k$ generators $\phi_{\alpha_i}$, of which each pair of generators commutes, then the full set of $k$ generators must commute. The proof follows directly from the definitions, especially from two observations from Definition 31: (1) commutativity of a set of actions is independent of the states implicated; and (2) any collection of pairwise commutative actions is totally commutative. □

The result of this is, first, informative restrictions on the topology of state complexes, *e.g.*, the impossibility of spheres as state complexes for any system of reconfiguration. More important to applications, however, is the implication to path-planning and efficient coordination.

6.2. **Efficient state planning.** Theorem 40 has an important corollary for optimal planning.

**Corollary 41.** *Each homotopy class of paths connecting two given points of a state complex contains a unique shortest path.*

This is well-known for NPC spaces [8]. Fig. 17[left] gives a simple example of a 2-d cubical complex with positive curvature for which the above corollary fails.

This corollary is a key ingredient in the applications of NPC geometry to path-planning on a configuration space, since one expects geodesics on $\mathcal{S}$ to coincide with optimal solutions to the state planning problem. However, in the context of robotics applications, the goal of solving the state-planning problem is *not* necessarily coincident with the geodesic problem on the state complex. Fig. 18[left] illustrates the matter concisely. Consider a portion of a state complex $\mathcal{S}$ which is planar and two-dimensional. To get from point $p$ to point $q$ in $\mathcal{S}$, any edge-path which is weakly monotone increasing in the horizontal and vertical directions is of minimal length in the transition graph. The true geodesic is, of course, the straight line, which is not well-positioned with respect to the discrete cubical structure.

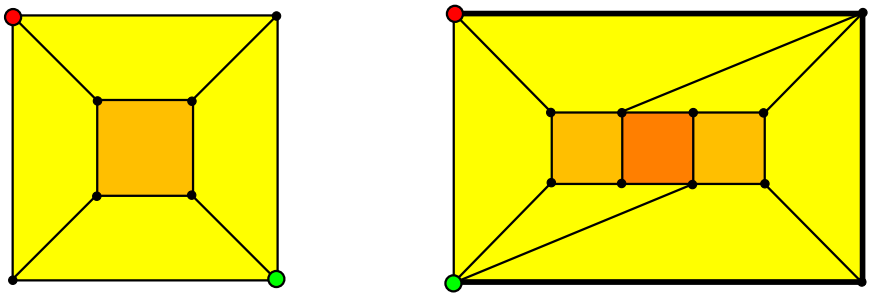

FIGURE 17. [left] A 2-d cube-complex (with positive curvature) possessing two distinct homotopic shortest paths between a pair of marked points; [right] a 2-d cube-complex (with positive curvature) possessing an edge path (the thick line on the boundary) which is a locally (but not globally) shortest path. Any cube path near this path is strictly longer. Note: for both of these complexes, all 2-cells are unit-length Euclidean cubes — the figures are deformed for purposes of illustration.

Given the assumption that *each elementary move can be executed at a uniform maximum rate*, it is clear that the $\ell^2$ geodesic on $\mathcal{S}$ is also **time-minimal** in the sense that the elapsed time is minimal among all paths from $p$ to $q$. However, there is an envelope of non-geodesic paths which are yet time-minimizing. Indeed, the $\ell^2$ geodesic in Fig. 18[left] "slows down" some of the moves unnecessarily in order to maintain the constant slope.

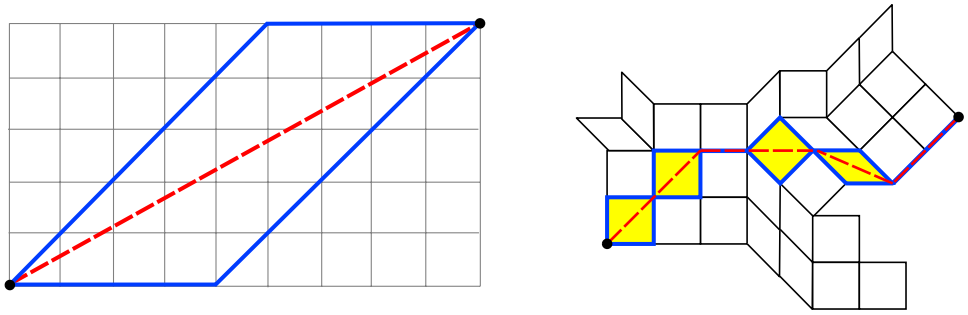

FIGURE 18. [left] The true geodesic lies within an envelope of time-minimizing paths. No minimal edge-paths are time-minimizing; [right] a normal path in an NPC complex (after [65]) follows along the diagonals of cubes in a greedy manner.

This leads us to define a second metric on $\mathcal{S}$, one which measures elapsed time. Namely, instead of the Euclidean metric on the cells of $\mathcal{S}$, consider the space $\mathcal{S}$ with the $\ell^\infty$ norm on each cell. The geodesics in this geometry represent reconfiguration paths which are *time minimizing*.

The paper [3] gives an algorithm for taking any path between vertices on $\mathcal{S}$ and reducing it to a **normal form** which is an $\ell^i nfty$ geodesic. One can show using techniques from **cube paths** of NPC geometry [65] that this normal form is unique — each homotopy class of paths in $\mathcal{S}$ from $p$ to $q$ possesses a unique normal path.

This is very significant. In practice, state-planning via constructing all of $\mathcal{S}$ and determining geodesics is computationally infeasible: the total size of the state complex is often huge. If however we assume that the state complex is unknown *but* that some path between states has been pre-computed (in the case of metamorphic robots, this done often by a decentralized planner [9, 10, 81, 82, 83, 87]), then we may optimize this trajectory. One can employ a gradient-descent curve shortening on that portion of the state complex 'explored' in real-time by the path. The above results imply that any algorithm which monotonically reduces path length must converge to the shortest path (in that homotopy class) and cannot be hung up on a locally minimal path. Thus nonpositive curvature of $\mathcal{S}$ allows for a time-optimization which does not require explicit construction of $\mathcal{S}$.

The presence of local minima is a persistent problem in optimization schemes across many disciplines: nonpositive curvature is a handy antidote.

6.3. **Back to braids.** The fundamental group of a configuration space is an $n$-stranded braid group. By analogy, the fundamental groups of state complexes should be, to some degree, *braidish*. This begs the question:

*Question* 42. Which groups are realizable as the fundamental group of a state complex?

Nonpositive curvature provides some elementary restrictions: *e.g.*, the fundamental groups are torsion-free, like the Artin braid groups. The careful reader will note that not all braid groups are torsion-free — the braid groups of $S^2$ are a classical example. We note, however, that $S^2$ is a font of positive curvature, and we are not surprised at the algebraic reflection of this geometric fact.

Returning to state complexes, the following result yields some additional restrictions.

**Theorem 43** ([41])**.** *The fundamental group of any finite state complex $\mathcal{S}$ embeds into the group with presentation*

$$\mathcal{A}_{\mathcal{S}} = \langle \phi_\alpha \, : \, [\phi_\alpha, \phi_\beta] \rangle \, , \tag{6.1}$$

*whose generators $\phi_\alpha$ are the generators of the reconfigurable system, and whose relations are commutators of pairs of generators in the reconfigurable system which commute.*

Groups for which all relations are commutators of generators are called **Artin right-angled groups**. These play an important role in geometric group theory, along side Coxeter groups and similar objects. Theorem 43 validates the terminology of *generators* and *commuting* from Definitions 27 and 31.

The proof follows almost directly from the proof of Theorem 2 of [20], which states that the fundamental groups of $\mathcal{D}^n(\Gamma)$ embed in right-angled Artin groups.

**Corollary 44.** *Fundamental groups of state complexes are linear: they embed in $GL_n(\mathbb{R})$*

Such embedding properties come for free from properties of Artin right-angled groups [23].

## 7. Last strands

The topics touched upon in these notes are, of necessity, selective in both nature and scope. To give a glimpse of the available breadth, we list a few related results, future directions, and extensions.

7.1. **Configuration spaces.** The topology of configuration spaces goes well beyond the examples and perspectives sampled in these notes. The following remarks offer pointers to the literature for the interested reader.

7.1.1. *Linkages.* Absent from these notes is an explication of the long and detailed history of configuration spaces for mechanical linkages. Readers are perhaps familiar with the classical **Peaucellier linkage** [19] for converting circular to linear motion. This is a question about the algebraic geometry of linkage

configuration spaces. Numerous papers have considered the topology and geometry of configuration spaces of closed-chain linkages [80, 46, 48, 79, 59, 44]. A universality result [47, 49] states that every orientable manifold arises as (a connected component of) the configuration space of some (planar) linkage.

7.1.2. *Algorithmic complexity.* Upon translating the problem of motion-planning in robotics to that of path-planning on a configuration space, the topologist breathes a sigh of contentment: "problem solved." The roboticist, on the other hand, finds that the problem is just beginning. A real physical environment is cluttered with obstacles to be avoided. Real robots have physical shape and appendages which must not collide. Real motion planning involves programming work, with all the difficulties of kinematics, friction, and the like.

But for the sake of maintaining the topologists' equanimity, let us assume all these problems away and return to the setting of point robots in a domain $X$. Even here, actually computing $\mathcal{C}^n(X)$ and performing path-planning is computationally challenging. There is an innumerable literature on the computational complexity of path planning in robotics. The books [54, 55] give an overview with references. Suffice to say that the standard complete algorithm of Canny [11] using cylindrical algebraic decompositions is doubly exponential in the geometric complexity of the workspace. Worse still, *optimal* path-planning — that is, finding a shortest path between two points in a space — was shown to be NP complete for spaces of dimension three and above [12]. (Interestingly enough, the hardness proof requires using quite a bit of positive curvature in the space to force many geodesics between points.)

7.1.3. *Topological complexity.* A more recent reformulation of the topology of path planning on configuration spaces has been given by Farber [29]. In his work, the goal of path-planning is not to generate a single path between initial and terminal points on a configuration space, but rather to construct a **path planner**: a function from $\mathcal{C}^n(X) \times \mathcal{C}^n(X)$ to the path space $\mathcal{P}(\mathcal{C}^n(X))$. For purposes of stability, one wants this mapping to be continuous. For example, on a web-based mapping system that gives directions (*e.g.*, `Google Earth`), it would be suboptimal if perturbing the start or end points on the map gave a wholly different set of directions.

However, such is often unavoidable for topological reasons. Farber observes that a continuous path planner is possible if and only if $C^n(X)$ is a contractible space. He then transforms the question into one of complexity: what is the smallest number of continuous path-planners on subsets of $\mathcal{C}^n(X) \times \mathcal{C}^n(X)$ which cover all possible initial and final locations? Farber relates this **topological complexity**, $TC(\mathcal{C}^n(X))$, to Schwarz genus and the algebraic topology of $\mathcal{C}^n(X)$. This topological measure of the difficulty of path planning is the subject of much investigation. For example, [31] shows that the topological complexity of $\mathcal{C}^n(\mathbb{R}^2)$ and $\mathcal{C}^n(\mathbb{R}^3)$ is linear in $n$. In contrast, the topological complexity

of motion planning on graphs has an upper bound independent of $n$ [30], *cf.* Theorem 9.

7.2. **Graph braid groups.** There has been a flurry of recent work in understanding configuration spaces of graphs and their braid groups. The following are examples:

7.2.1. *Embeddings.* Crisp and Wiest showed [20] that braid groups of graphs embed in Artin right-angled groups. The embedding of Theorem 43 agrees with their embedding; indeed, the proof of [20] is the basis of the more general result in [41]. A dual result of that cited above has appeared recently in the thesis of Sabalka [73]. Namely, any right-angled Artin group embeds into the braid group of some number of points on a graph. (In combination with Theorem 43, we see that any state complex group embeds into a graph braid group.)

7.2.2. *Right-angled groups.* Determining which groups arise as graph braid groups has received a great deal of recent attention. It was conjectured in [37] that all graph braid groups were right-angled Artin groups. This conjecture has been repeatedly disproved: see [1, 77]. A cohomological argument of Farley and Sabalka shows that $\mathcal{UC}^n(\mathsf{T})$ is Artin right-angled if and only if $n < 4$ or there exists an embedded arc in $\mathsf{T}$ passing through all vertices of $\mathsf{T}$ [34].

7.2.3. *$\pi_1$ and $H^*$.* Farley and Sabalka have recently [33] given a presentation for $\mathcal{UC}^n(\mathsf{T})$ for any tree $\mathsf{T}$. Their methods stem from examining the discretized configuration space and using Forman's discrete Morse theory to determine critical cells. These methods have led also to an understanding of the cohomology ring [34], which has surprising connections to face algebras and Stanley-Reisner rings.

7.2.4. *Rigidity.* Sabalka has a preprint giving a rigidity result for tree braid groups. A simple version of the result is the following: if one is given the isomorphism type of $\pi_1(\mathcal{UC}^4(\mathsf{T}))$ — the 4-strand braid group, and no other information — then one can determine the combinatorial type of $\mathsf{T}$. This is an incredibly surprising and elegant result.

7.3. **State complexes.** There are a number of interesting questions concerning state complexes, many of which are open.

7.3.1. *Manifolds.* Which manifolds can arise as state complexes of a reconfigurable system? Thanks to Theorem 40, only aspherical manifolds are within the realm of possibility. There is a simple proof in [41] that given any simplicial complex $L$ which is flag, there exists a reconfigurable system whose state complex has link $L$ at each state. Thus, many different manifolds best described as

*hyperbolic* may arise. Combining this result with Theorem 43, we see that there are hyperbolic 3-manifold groups which embed in Artin right-angled groups.

7.3.2. *Realization.* It is currently unknown which NPC cubical complexes can arise as state complexes for some reconfigurable system. The thesis of Peterson (in progress) gives some surprising examples of NPC cubical complexes which cannot be realized as state complexes: see Fig. 19.

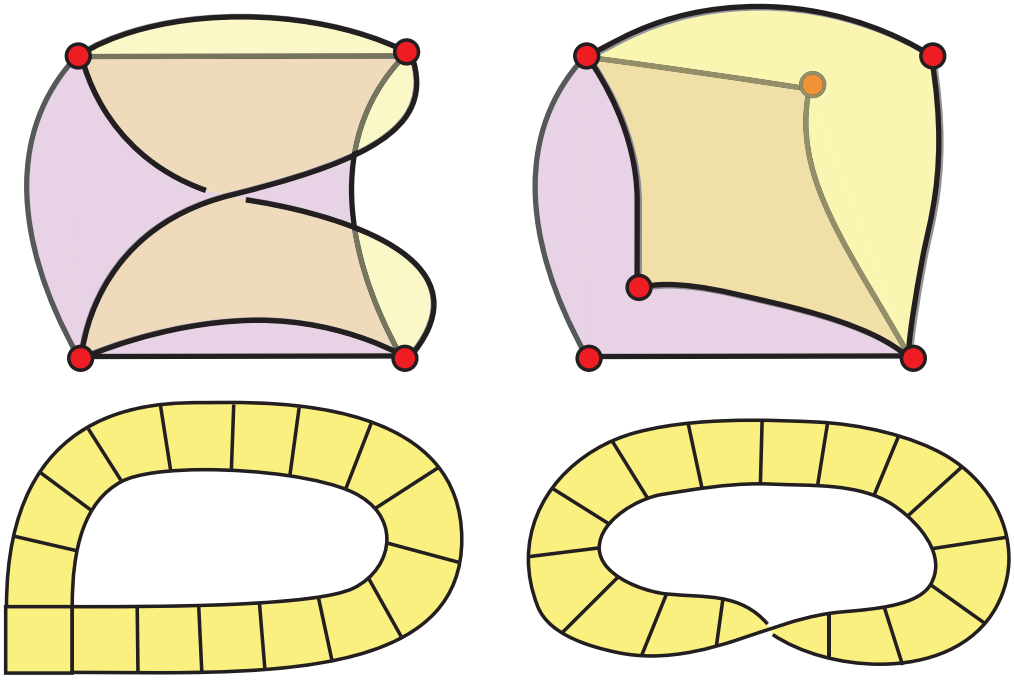

FIGURE 19. None of these NPC cube complexes can be realized as a state complex of a reconfigurable system.

7.3.3. *Limits.* To some extent, state complexes are generalizations of the discretized configuration spaces $\mathcal{D}^n(\Gamma)$ and thus "approximate" a topological configuration space. It is tempting to conject that all state complexes are "discretizations" of some topological configuration space. There are certain examples of reconfigurable systems for which it makes sense to refine the underlying graph and obtain a sequence of reconfigurable systems. In such examples, one can ask whether the sequence of state complexes enjoys any sort of convergence properties. A canonical example of such refinement occurs in the statement of Theorem 20. The refinement of the underlying edges of $\Gamma$ which inserts additional vertices with the zero label has the effect of enlarging the state complex. Let $\Gamma^{(i)}$ denote the $i^{th}$ refinement of $\Gamma$ (each edge of the original $\Gamma$ now has $2^i$ subedges). Under the canonical inclusions of $\mathcal{D}^n(\Gamma)$ into $\mathcal{C}^n(\Gamma)$, the sequence $\mathcal{D}^n(\Gamma^{(i)})$ can be discussed in terms of Hausdorff and Gromov-Hausdorff limits.

It follows from Theorem 20 that the state complexes $\mathcal{D}^n(\Gamma^{(i)})$ stabilizes in homotopy type as $i \to \infty$ and that this stable homotopy type is precisely that of $\mathcal{C}^n(\Gamma)$.

It is intriguing to consider similar refinement and limit questions for general state complexes. One can certainly construct examples for which this type of refinement does not lead to state complexes which stabilize in homotopy type. However, it may be that there is a notion of refinement for which convergence in the Gromov-Hausdorff sense works.

## 8. HOPE AT THE BOTTOM OF THE BOX

There are numerous settings across many scientific disciplines in which a configuration space seems to be lurking beneath the surface. These notes have detailed a few examples, mostly inspired by robotics applications and mostly amenable to techniques from the topology and geometry of cube complexes.

Some important examples not covered here include the following.

8.1. **Protein folding:** Perhaps the largest scientific challenge of our day is the need for fast, accurate models of protein folding that can be used to assist drug design. The sheer enormity of degrees of freedom (read, "dimension of the configuration space") frustrates analysts. Putting this problem on the "Moore's Law Credit Card"[3] is futile.

It is perhaps not unreasonable to hope that understanding better the configuration space associated to a protein will carry explanatory power in a biological setting. One famous enigma concerning the folding of proteins, is the *Levinthal paradox* [56]: although the number of possible conformations of a polypeptide chain is too large to be sampled exhaustively, protein sequences do fold into unique native states in seconds. The paradox, posed by Cyrus Levinthal in 1969, is still not completely resolved. Since in order to decipher and control cellular processes one needs to master the mechanisms regulating the configurational changes of nucleic acids, the demand for algorithms that accurately predict the paths of these changes is high.

A speculative geometric explanation for the Levinthal paradox is that the typical protein configuration space has a residue of hyperbolic geometry. Recall that in hyperbolic space, the volume of a ball grows exponentially as a function of the radius: hence a configuration space with lots of negative (sectional) curvature would encompass a large volume (and thus, a large number of discretized states) within a set of small diameter (as measured by time-to-goal under local folding rules).

[3]That is, waiting for processor speed to catch up to the complexity of the problem...

8.2. **Self-assembly:** The collective behavior of mechanical devices at micro- and nano- scales increasingly resembles that of chemical or biological reactions, in which individual particles endowed with intrinsic geometric data (the shape of the particle) and algorithmic data (reactivity, bonding proclivities, repulsions) yield assemblies of complex devices through purely local means. The challenge of being able to control these processes for purposes of design in the mechanical realm is two-fold. First is the *physical* challenge: to be able to understand and exploit the behavior of mechanical devices at micro- (or smaller) scales. Second is the *algorithmic* challenge: how does one 'program' a collection of parts under physical constraints to assemble into a desired device. At small scales, one exchanges the ability to fabricate very intricate components for the ability to produce huge numbers of simple components. A significant algorithmic question is therefore which complex devices can be built out of simple pieces with simple assembly rules?

Not surprisingly, experimentalists have been able to build objects at small scales which have complete regularity with respect to the local geometry: planar hex and square lattices [84], cylindrical sheets [45], and spheres [5]. These have the property that the component 'tiles' are regularly shaped and the assembly 'rules' are determined by passive physical means, e.g., via surface tension effects on a liquid surface. Such geometrically regular assemblies can and are having significant impact on manufacturing at small scales [53, 60].

With respect to 'programmable' self-assembly of mechanical devices at small scales, a fairly recent NASA-IAC report on *Kinematic Cellular Automata* [78] contains extensive information on small-scale programmable mechanical assembly, mostly in the context of self-replication. We note in particular the claim presented there that self-assembly requires "a new mathematics" to be effective in programming devices.

One possible avenue of exploration is that of modeling the appropriate variables as topological spaces, with a *parts space*, a *design space*, an *assembly products space* and other constructs. All of these would be examples of systems with very many degrees-of-freedom and potentially intricate features. The paper [50] gives a grammatical approach to some of these problems.

8.3. **Genetics:** One area in which a great deal of attention has been recently focused is on configuration spaces associated to **phylogenetic trees** — data structures for organizing and comparing genetic sequencing of organisms. Here, we refer the interested reader to the seminal paper [6], which build configuration spaces for rooted labeled trees and uses the NPC geometry of these spaces to, *e.g.*, give a well-defined notion of averages for phylogenetic data. A more recent survey article [76] gives a broader treatment of applications of geometric and topological combinatorics to a broad class of problems in phylogenetics. In all contexts, the challenge of finding the correct space for representing phylogenetic data is central. It is not a coincidence that, as in the case of configuration

spaces for robot coordination, the geometry as well as the topology of these spaces are central.

8.4. **"...in cui la mia speranza vige".** In all the above cases, whatever configuration spaces regulate the phenomena, they are not of the 'traditional' form of (1.1). Some of these examples are being addressed by mathematicians in creative and compelling ways.

A mathematician should be optimistic in the face of what seems to scientists insurmountable challenges. After all, classifying spaces, sheaf theory, K-theory, Floer theory, and the entire Grothendieck programme are but a few examples of the tools topologists have invented to manage the complexities of topological spaces, and most of these tools are not merely comfortable with but wholly dependent upon the use of infinite-dimensional spaces. The fact that there has been so little historical communication between the topologists who have invented the tools and the scientists who may need the tools is a reason to rejoice, for it may be that the solutions to these significant scientific challenges are not so far away.

## ACKNOWLEDGEMENTS

The writing of these notes is supported in part by DARPA # HR0011-07-1-0002 and by NSF-DMS 0337713.

## REFERENCES

[1] A. Abrams. *Configuration spaces and braid groups of graphs.* Ph.D. thesis, UC Berkeley, 2000.
[2] A. Abrams and R. Ghrist. Finding topology in a factory: configuration spaces. *Amer. Math. Monthly*, 109 (2002) 140–150.
[3] A. Abrams and R. Ghrist. State complexes for metamorphic robot systems. *Intl. J. Robotics Research*, 23 (2004) 809–824.
[4] S. Alexander, R. Bishop, and R. Ghrist. Pursuit and evasion on non-convex domains of arbitrary dimensions. In Proc. Robotics: Systems and Science, 2006.
[5] B. Berger, P. Shor, L. Tucker-Kellogg, and J. King. Local rule-based theory of virus shell assembly. *Proceedings of the National Academy of Science, USA*, 91(6):7732–7736, August 1994.
[6] L. Billera, S. Holmes, and K. Vogtmann. Geometry of the space of phylogenetic trees. *Adv. Applied Math.* 27 (2001) 733-767.
[7] J. Birman, *Braids, Links, and Mapping Class Groups*, Princeton University Press, Princeton, N.J., 1974.
[8] M. Bridson and A. Haefliger. *Metric Spaces of Nonpositive Curvature*, Springer-Verlag, Berlin, 1999.
[9] Z. Butler, S. Byrnes, and D. Rus. Distributed motion planning for modular robots with unit-compressible modules. In Proc. IROS, 2001.
[10] Z. Butler, K. Kotay, D. Rus, and K. Tomita. Cellular automata for decentralized control of self-reconfigurable robots. In Proc. IEEE ICRA Workshop on Modular Robots, 2001.
[11] J. Canny, *The Complexity of Robot Motion Planning*, MIT Press, Cambridge, MA, 1988.
[12] J. Canny and J. Reif. Lower bounds for shortest path and related problems. In *Proc. 28th Ann. IEEE Symp. Found. Comp. Sci.*, 49–60, 1987.

[13] A. Castano, W.M. Shen, and P. Will, CONRO: Towards deployable robots with inter-robots metamorphic capabilities, *Autonomous Robots* 8(3), 2000, 309-324.
[14] G. Castleberry, *The AGV Handbook*, Braun-Brumfield, Ann Arbor, MI, 1991.
[15] R. Charney. The Tits conjecture for locally reducible Artin groups. *Internat. J. Algebra Comput.* 10 (2000) 783-797.
[16] G. Chirikjian. Kinematics of a metamorphic robotic system. In Proc. IEEE ICRA, 1994.
[17] G. Chirikjian and A. Pamecha, *Bounds for self-reconfiguration of metamorphic robots*, in Proc. IEEE ICRA, 1996.
[18] G. Chirikjian, A. Pamecha, and I. Ebert-Uphoff, *Evaluating efficiency of self-reconfiguration in a class of modular robots*, J. Robotics Systems **13**(5): 317-338, 1996.
[19] R. Courant and H. Robbins, *What is Mathematics*, Oxford University Press, 1941.
[20] J. Crisp and B. Wiest. Embeddings of graph braid groups and surface groups in right-angled Artin groups and braids groups. *Alg. & Geom. Top.* 4 (2004) 439–472.
[21] P. Csorba and F. Lutz. Graph coloring manifolds. To appear, *Contemp. Math.* Preprint, 2005.
[22] M. Davis. Groups generated by reflections and aspherical manifolds not covered by Euclidean space. *Ann. Math.* (2) 117 (1983) 293–324.
[23] M. Davis and T. Januszkiewicz. Right angled Artin groups are commensurable with right angled Coxeter groups. *J. Pure and Appl. Algebra* 153 (2000) 229–235.
[24] D. Epstein et al. *Word Processing in Groups*. Jones & Bartlett Publishers, Boston MA, 1992.
[25] M. Erdmann and T. Lozano-Perez. On multiple moving objects. In Proc. IEEE ICRA, 1986.
[26] E. Fadell and S. Husseini. *Geometry and Topology of Configuration Spaces*, Springer, Berlin, 2001.
[27] R. Fair, V. Srinivasan, H. Ren, P. Paik, V. Pamula, and M. Pollack. Electrowetting based on chip sample processing for integrated microfluidics. *IEEE Inter. Electron Devices Meeting (IEDM)* 2003.
[28] R. Fair, A. Khlystov, V. Srinivasan, V. Pamula, and K. Weaver. Integrated chemical/biochemical sample collection, pre-concentration, and analysis on a digital microfluidic lab-on-a-chip platform. In *Lab-on-a-Chip: Platforms, Devices, and Applications, Conf. 5591, SPIE Optics East*, Philadelphia, Oct. 25–28, 2004.
[29] M. Farber, Topological Complexity of Motion Planning. *Discrete and Computational Geometry* 29, 2003, 211221.
[30] M. Farber, Collision free motion planning on graphs, in *Algorithmic Foundations of Robotics IV*, M. Erdmann, D. Hsu, M. Overmars, A. F. van der Stappen eds., Springer, 2005, 123–138.
[31] M. Farber, S. Yuzvinsky, Topological Robotics: Subspace Arrangements and Collision Free Motion Planning. *Transl. of AMS* 212, 2004, 145-156.
[32] D. Farley. Finiteness and CAT(0) properties of diagram groups. *Topology* 42 (2003) 1065–1082.
[33] D. Farley and L. Sabalka, Discrete Morse theory and graph braid groups, *Algebraic & Geometric Topology* 5, 2005, 1075-1109.
[34] D. Farley and L. Sabalka, On the cohomology rings of tree braid groups, `arXiv:math.GR/0602444.`
[35] P. Gaucher. About the globular homology of higher dimensional automata. *Cahiers de Top. et Geom. Diff. Categoriques.* 43(2) (2002) 107–156.
[36] R. Ghrist. Shape complexes for metamorphic robot systems. In *Algorithmic Foundations of Robotics V, STAR 7* (2004) 185–201.
[37] R. Ghrist. Configuration spaces and braid groups on graphs in robotics. In *AMS/IP Studies in Mathematics* 24 (2001) 29–40.
[38] R. Ghrist and D. Koditschek. Safe, cooperative robot dynamics on graphs. *SIAM J. Control Optim.* 40 (2002) 1556–1575.
[39] R. Ghrist and S. LaValle. Nonpositive curvature and Pareto optimal motion planning. To appear, *SIAM J. Control Optim.*, 2006.
[40] R. Ghrist, J. O'Kane, and S. M. LaValle. Computing Pareto optimal coordinations on roadmaps. *Intl. J. Robotics Research*, 12 (2006) 997–1012.

[41] R. Ghrist and V. Peterson. The geometry and topology of reconfiguration, *Adv. Appl. Math.*, 38, 2007, 302323.
[42] M. Gromov. Hyperbolic groups. In *Essays in Group Theory*, MSRI Publ. 8, Springer-Verlag, 1987.
[43] F. Haglund and D. Wise. Special cube complexes. Preprint, March 2005.
[44] J.-C. Hausmann and A. Knutson, Polygon spaces and Grassmannians. *Enseign. Math.* (2) 43(1-2), 1997, 173–198.
[45] H. Jacobs, A. Tao, A. Scwartz, D. Gracias, and G. Whitesides. Fabrication of a cylindrical display by patterned assembly. *Science*, 296:4763–4768, April 2002.
[46] M. Kapovich and J. Millson, Moduli spaces of linkages and arrangements. Advances in geometry, 237–270, Progr. Math., 172, Birkhuser Boston, Boston, MA, 1999.
[47] M. Kapovich and J. Millson, Universality theorems for configuration spaces of planar linkages. *Topology* 41(6), 2002, 1051–1107.
[48] M. Kapovich and J. Millson, On the moduli space of polygons in the Euclidean plane, *J. Diff. Geo.* 42, 1995, 133-164.
[49] H. King, Planar linkages and algebraic sets. *Turkish J. Math.*, 23(1), 1999, 33–56.
[50] E. Klavins, R. Ghrist, and D. Lipsky, The graph grammatical approach to self-organizing robotic systems, *IEEE Trans. Automatic Controls*, 51(6), 2006, 949–962.
[51] D. Koditschek and E. Rimon, Robot navigation functions on manifolds with boundary, *Adv. in Appl. Math.* 11 (1990) 412–442.
[52] K. Kotay and D. Rus, The self-reconfiguring robotic molecule: design and control algorithms, in *Proc. Workshop Alg. Found. Robotics*, 1998.
[53] D. Lammers. Motorola speeds the move to nanocrystal flash. *EE Times*, December 8, 2003.
[54] J.-C. Latombe, *Robot Motion Planning*, Kluwer Academic Press, Boston, MA, 1991.
[55] S. LaValle, *Planning Algorithms*, Cambridge University Press, 2006.
[56] C. Levinthal, in Mössbauer Spectroscopy in Biological Systems, Proceedings of a meeting held at Allerton House, Monticello, Illinois , edited by J. T. P. DeBrunner and E. Munck, pp. 22-24, University of Illinois Press, Illinois, 1969.
[57] L. Lovász. Kneser's conjecture, chromatic number, and homotopy. *J. Combin. Theory Ser. A*, 25 (1978) 319–324.
[58] C. McGray and D. Rus, *Self-reconfigurable molecule robots as 3-d metamorphic robots*, in Proc. Intl. Conf. Intelligent Robots & Design, 2000.
[59] R. Milgram and J. Trinkle. The geometry of configuration spaces of closed chains in two and three dimensions, *Homology, Homotopy, and Applications*, 6(1), 2004, 237-267.
[60] N. Mokhoff. Crystals line up in IBM flash chip. *EE Times*, December 8, 2003.
[61] S. Murata, H. Kurokawa, and S. Kokaji, *Self-assembling machine*, in Proc. IEEE ICRA, 1994.
[62] S. Murata, H. Kurokawa, E. Yoshida, K. Tomita, and S. Kokaji, *A 3-d self-reconfigurable structure*, in Proc. IEEE ICRA, 1998.
[63] S. Murata, E. Yoshida, A. Kamikura, H. Kurokawa, K. Tomita, and S. Kokaji, *M-TRAN: Self-reconfigurable modular robotic system*, IEEE-ASME Trans. on Mechatronics 7(4), 431-441, 2002.
[64] A. Nguyen, L. Guibas, and M. Yim, *Controlled module density helps reconfiguration planning*, in Proc. WAFR, 2000.
[65] G. Niblo and L. Reeves. The geometry of cube complexes and the complexity of their fundamental groups. *Topology*, 37 (1998) 621–633.
[66] A. Pamecha, I. Ebert-Uphoff, and G. Chirikjian, *Useful metric for modular robot motion planning*, IEEE Trans. Robotics & Automation, **13**(4), 531-545, 1997.
[67] V. Pratt. Modelling concurrency with geometry. In Proc. 18th Symp. on Principles of Programming Languages, 1991.
[68] M. Raussen. State spaces and dipaths up to dihomotopy. *Homotopy, Homology, & Appl.*, 5, 2003, 257–280.
[69] M. Raussen. On the classification of dipaths in geometric models of concurrency. *Math. Structures Comp. Sci.* 10, 2000, 427–457.

[70] L. Reeves. Rational subgroups of cubed 3-manifold groups. Michigan Math. J. 42, 1995, 109–126.
[71] C. Reutenauer. *The Mathematics of Petri Nets*. Prentice-Hall, 1990.
[72] D. Rus and M. Vona, *Crystalline robots: Self-reconfiguration with unit-compressible modules*, Autonomous Robots, **10**(1), 107-124, 2001.
[73] L. Sabalka, Embedding right-angled Artin groups into graph braid groups, `arXiv:math.GR/0506253`.
[74] A. Sali, E. Shakhnovich, and M. Karplus. How does a protien fold? *Nature* 369, 1994, 248–251.
[75] A. Sali, E. Shakhnovich, and M. Karplus. Kinetics of protien folding. *J. Mol. Biol.* 235, 1994, 1614–1636.
[76] B. Sturmfels and L. Pachter, The mathematics of phylogenomics, (with L. Pachter), *SIAM Review* 49, 2007, 3-31.
[77] J. Swiatkowski. Estimates for homological dimension of configuration spaces of graphs. *Colloq. Math.* 89 (2001) 69–79.
[78] T. Toth-Fejel. Modeling Kinematic Cellular Automata. NASA Institute for Advanced Concepts Phase I: CP-02-02. General Dynamics Advanced Information Systems Contract # P03-0984. April 30, 2004.
[79] J. Trinkle and R. Milgram, Complete path planning for closed kinematic chains with spherical joints, *Intl. J. Robotics Research*, 21(9), 2002, 773-789.
[80] K. Walker, *Configuration Spaces of Linkages*, Undergraduate thesis, Princeton University, 1985.
[81] J. Walter, J. Welch, and N. Amato. Distributed reconfiguration of metamorphic robot chains. In Proc. ACM Symp. on Distributed Computing, 2000.
[82] J. Walter, E. Tsai, and N. Amato. Choosing good paths for fast distributed reconfiguration of hexagonal metamorphic robots. In Proc. IEEE ICRA, 2002.
[83] J. Walter, J. Welch, and N. Amato. Concurrent metamorphosis of hexagonal robot chains into simple connected configurations. *IEEE Trans. Robotics & Automation* 15 (1999) 1035–1045.
[84] G. Whitesides and B. Grzybowski. Self assembly at all scales. *Science*, 295:2418–2421, 2002.
[85] M. Yim, *A reconfigurable robot with many modes of locomotion*, in Proc. Intl. Conf. Adv. Mechatronics, 1993.
[86] M. Yim, J. Lamping, E. Mao, and J. Chase, *Rhombic dodecahedron shape for self-assembling robots*, Xerox PARC Tech. Rept. P9710777, 1997.
[87] M. Yim, Y. Zhang, J. Lamping, and E. Mao. Distributed control for 3-d metamorphosis. *Autonomous Robots J.* 10 (2001) 41–56.
[88] E. Yoshida, S. Murata, K. Tomita, H. Kurokawa, and S. Kokaji. Distributed formation control of a modular mechanical system. In Proc. Intl. Conf. Intelligent Robots & Sys., 1997.

DEPARTMENT OF MATHEMATICS AND COORDINATED SCIENCE LABORATORY, UNIVERSITY OF ILLINOIS, URBANA IL, USA

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