February 2002

Safe Cooperative Robot Dynamics on Graphs

Robert W. Ghrist
Georgia Institute of Technology

Daniel E. Koditschek
University of Pennsylvania, kod@seas.upenn.edu

Reprinted from *SIAM Journal on Control and Optimization* , Volume 40, Issue 5, February 6 2002, pages 1556–1575.
Publisher URL: http://dx.doi.org/10.1137/S0363012900368442

NOTE: At the time of publication, the author, Daniel Koditschek, was affiliated with the University of Michigan. Currently, he is a faculty member of the School of Engineering at the University of Pennsylvania.

Safe Cooperative Robot Dynamics on Graphs

Abstract

This paper introduces the use of vector fields to design, optimize, and implement reactive schedules for safe cooperative robot patterns on planar graphs. We consider automated guided vehicles (AGVs) operating upon a predefined network of pathways. In contrast to the case of locally Euclidean configuration spaces, regularization of collisions is no longer a local procedure, and issues concerning the global topology of configuration spaces must be addressed. The focus of the present inquiry is the definition, design, and algorithmic construction of controllers for achievement of safe, efficient, cooperative patterns in the simplest nontrivial example (a pair of robots on a Y-network) by means of a hierarchical event-driven state feedback law.

Keywords

configuration spaces, AGV, graph network, hierarchical control

Comments

Reprinted from *SIAM Journal on Control and Optimization* , Volume 40, Issue 5, February 6 2002, pages 1556–1575.
Publisher URL: http://dx.doi.org/10.1137/S0363012900368442

NOTE: At the time of publication, the author, Daniel Koditschek, was affiliated with the University of Michigan. Currently, he is a faculty member of the School of Engineering at the University of Pennsylvania.

SIAM J. CONTROL OPTIM.
Vol. 40, No. 5, pp. 1556–1575

SAFE COOPERATIVE ROBOT DYNAMICS ON GRAPHS*

ROBERT W. GHRIST† AND DANIEL E. KODITSCHEK‡

Abstract. This paper introduces the use of vector fields to design, optimize, and implement reactive schedules for safe cooperative robot patterns on planar graphs. We consider automated guided vehicles (AGVs) operating upon a predefined network of pathways. In contrast to the case of locally Euclidean configuration spaces, regularization of collisions is no longer a local procedure, and issues concerning the global topology of configuration spaces must be addressed. The focus of the present inquiry is the definition, design, and algorithmic construction of controllers for achievement of safe, efficient, cooperative patterns in the simplest nontrivial example (a pair of robots on a Y-network) by means of a hierarchical event-driven state feedback law.

Key words. configuration spaces, AGV, graph network, hierarchical control

AMS subject classifications. Primary, 93C85, 68T40; Secondary, 93C25, 37N35

PII. S0363012900368442

1. Introduction.

Recent literature suggests the growing awareness of a need for "reactive" scheduling wherein one desires not merely a single deployment of resources but a plan for successive redeployments against a changing environment [19]. However, scheduling problems have been traditionally solved by appeal to a discrete representation of the domain at hand. Thus the need for "tracking" changing goals introduces a conceptual dilemma: there is no obvious topology by which proximity to the target of a given deployment can be measured. In contrast to problems entailing the management of information alone, problems in many robotics and automation settings involve the management of *work*—the exchange of energy in the presence of geometric constraints. In these settings, it may be desirable to postpone the imposition of a discrete representation long enough to gain the benefit of the natural topology that accompanies the original domain.

This paper explores the use of vector fields for reactive scheduling of safe cooperative robot patterns on graphs. The word "safe" means that obstacles—designated illegal portions of the configuration space—are avoided. The word "cooperative" connotes situations wherein physically distributed agents are collectively responsible for executing the schedule. The word "pattern" refers to tasks that cannot be encoded simply in terms of a point goal in the configuration space. The word "reactive" will be interpreted as requiring feedback so that the desired pattern rejects perturbations: conditions close but slightly removed from those desired remain close and, indeed, converge toward the exactly desired pattern.

1.1. Setting: AGVs on a guidepath network of wires.

An automated guided vehicle (AGV) is an unmanned powered cart "capable of following an external guidance signal to deliver a unit load from destination to destination," where, in most

*Received by the editors February 28, 2000; accepted for publication (in revised form) September 24, 2001; published electronically February 6, 2002. This research was supported in part by National Science Foundation grant IRI-9510673 [DK] and by National Science Foundation grant DMS-9971629 [RG]. A sketch of these ideas appeared in [10].
http://www.siam.org/journals/sicon/40-5/36844.html
†School of Mathematics and CDSNS, Georgia Institute of Technology, Atlanta, GA 30017 (ghrist@math.gatech.edu).
‡Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, MI 48109-2110 (kod@eecs.umich.edu).

common applications, the guidepath signal is buried in the floor [6]. Thus the AGV's work space is a network of wires—a graph. The motivation to choose AGV-based materials handling systems over more conventional fixed conveyors rests not simply in their ease of reconfigurability but in the potential they offer for graceful response to perturbations in normal plant operation. In real production facilities, the flow of work in process fluctuates constantly in the face of unanticipated work station downtime, variations in process rate, and, indeed, variations in materials transport and delivery rates [8]. Of course, realizing their potential robustness against these fluctuations in work flow remains an only partially fulfilled goal of contemporary AGV systems.

Choreographing the interacting routes of multiple AGVs in a nonconflicting manner presents a novel, complicated, and necessarily online planning problem. Nominal routes might be designed offline, but they can never truly be traversed with the nominal timing for all the reasons described above. Even under normal operating conditions, no single nominal schedule can suffice to coordinate the work flow as the production volume or product mix changes over time: new vehicles need to be added or deleted, and the routing scheme needs to be adapted. In any case, abnormal conditions—unscheduled process down times, blocked work stations, failed vehicles—continually arise, demanding altered routes.

The traffic control schemes deployed in contemporary AGV systems are designed to simplify the real-time route planning and adaptation process by "blocking zone control" strategies. The work space is partitioned into a small number of cells, and, regardless of the details of their source and destination tasks, no two AGVs are ever allowed into the same cell at the same time [6]. Clearly, this simplification results in significant loss of a network's traffic capacity.

In this paper, we will consider a centralized approach that employs dynamical systems theory to focus on real-time responsiveness and efficiency as opposed to computational complexity or average throughput. Without a doubt, beyond a certain maximum number of vehicles, the necessity to compute in the high dimensional configuration space will limit the applicability of any algorithms that arise. However, this point of view seems not to have been carefully explored in the literature. Indeed, we will sketch some ideas about how an approach that starts from the coupled version of the problem may lend sufficient insight to move back and forth between the individuals' and the group's configuration spaces even in real time. For the sake of concreteness we will work in the so-called pickup and delivery (i.e., where loads are picked up at certain points and dropped off at others, as opposed to "stop and go," where an AGV network stands in for an assembly line [3]) paradigm of assembly or fabrication (where a desired steady state "pattern"—a scheduled series of visits to specific work stations by specific AGVs—is dispatched ahead of time), and we will not be concerned with warehousing-style AGV applications.

1.2. Organization of the paper.

Section 2. We introduce the salient properties of a feedback controlled dynamical system on a graph by addressing the closed loop motion planning problem of a single AGV on its wire network. In this setting, configuration spaces are not required, although the nonmanifold structure of the work space necessitates a mild adaptation of the dynamical systems machinery, specified in Appendix A. We describe a simple hybrid controller built from *edge point fields*—locally defined dynamics that realize single letter patterns—which generalize the scheme Burridge, Rizzi, and Koditschek have proposed in [5].

Section 3. Turning to the central topic, we address the case of multiple AGVs in the simplest possible setting—two AGVs on a Y-graph—as a local exemplar of the general problem. The contributions of this section include

1. an intrinsic coordinate system for the configuration space;
2. a detailed analysis of the topology of the configuration space, affording immediate recourse to previously developed methods of safe controller design [12];
3. the construction of a "circulating flow" on this space that executes a stable safe periodic pattern as a canonical example of dynamically controlled collision-free behavior suitable to more general settings of the problem.

Section 4. Because limit cycles are likely too rigid a means of arbitrary pattern specification in the more generalized settings of the problem, we return to the notion of building a palette of control laws that realize safe "letters" along with a hybrid (logical level) scheme for concatenating them to produce arbitrary patterns in the form of periodic attracting orbits whose limit set is any desired "word," within a "monotone cycle" grammar as formalized in Theorem 3. Section 4 ends with a constructive procedure for incorporating performance guarantees in the construction of these grammars, concluding with a more speculative view of potential extensions of this work.

Appendix A is included to place on a rigorous foundation the use of vector fields on graphs and configuration spaces thereof.

2. Notation and background.

2.1. Graph topology. A *graph*, Γ, consists of a finite collection of 0-dimensional vertices $\mathcal{V}:=\{v_i\}_1^N$, and 1-dimensional edges $\mathcal{E}:=\{e_j\}_1^M$ assembled as follows. Each edge is homeomorphic to the closed interval $[0,1]$ attached to $\mathcal{V}$ along its boundary points $\{0\}$ and $\{1\}$.[1] We place upon Γ the quotient topology given by the endpoint identifications: neighborhoods of a point in the interior of e_j are homeomorphic images of interval neighborhoods of the corresponding point in $[0,1]$, and neighborhoods of a vertex v_i consist of the union of homeomorphic images of half-open neighborhoods of the endpoints for all incident edges.

The configuration spaces we consider in section 3 and throughout are subsets of self-products of graphs. The topology of $\Gamma \times \Gamma$ is easily understood in terms of the topology of Γ as follows [17]. Let $(x, y) \in \Gamma \times \Gamma$ denote an ordered pair in the product. Then any small neighborhood of (x, y) within $\Gamma \times \Gamma$ is the union of neighborhoods of the form $\mathcal{N}(u) \times \mathcal{N}(v)$, where $\mathcal{N}(\cdot)$ denotes the neighborhood within Γ. In other words, the products of neighborhoods form a *basis* of neighborhoods in the product space.

Given a graph, Γ, outfitted with a finite number N of noncolliding AGVs constrained to move on Γ, the (labeled) configuration space of safe motions is defined as

$$\mathcal{C}:= (\Gamma \times \cdots \times \Gamma) - \mathcal{N}(\Delta), \tag{2.1}$$

where $\Delta:=\{(x_i) \in \Gamma \times \cdots \times \Gamma : x_j = x_k \text{ for some } j \neq k\}$ denotes the pairwise diagonal and $\mathcal{N}(\cdot)$ denotes the (small) neighborhood.

[1] In our model, we will disallow "homoclinic" edges whose boundary points are attached to the same vertex. With respect to the application setting, this is very natural since vertices correspond to work stations along a path. It is hard to imagine networks designed with loops that do not service any work stations. In the worst case, there is established precedent in the AGV technology literature for introducing additional "transfer point" technology to a factory setting solely for purposes of traffic control [3].

For general graphs, the topological features of $\mathcal{C}$ can be extremely complicated, as measured by, say, the rank of the fundamental group (see [17] for definitions). Even in the case where the work space, Γ, is contractible (and thus, the product of its n copies is contractible), removal of this collision diagonal often creates spaces with a large fundamental group. For example, given a graph Γ_K with K edges all connected at a single point (forming a K-pronged "star"), it follows from the more general results in [9] that the fundamental group of the configuration space $\Gamma_K \times \Gamma_K - \mathcal{N}(\Delta)$ is a free group on $K^2 - 3K + 1$ generators; i.e., the number of "independent" closed paths in this space (with respect to continuous deformation) grows quadratically with K.

Mathematically, it is usually most interesting to pass to the quotient of $\mathcal{C}$ by the action of the permutation group on N elements, thus forgetting the identities of the AGV elements; however, as such spaces are almost completely divorced from any applications involving coordinated transport, we work on the "full" configuration space $\mathcal{C}$. We do not treat the general aspects of this problem comprehensively in this paper; rather, we restrict our attention to the simplest nontrivial example, which illustrates nicely the relevant features present in the more general situation.

In order to proceed, it is necessary to clarify what we mean by a vector field on a simplicial complex that fails to be a manifold. This is a nontrivial issue: for example, in the case of a graph, the tangent space to a vertex with incidence number greater than two is not well defined. We defer a more detailed discussion of these statements to Appendix A. The essential difference is that we construct *semiflows*—flows which possess unique forward orbits.

2.2. Edge point fields for single AGV control. In the context of describing and executing *patterns* or periodic motions on a graph, one desires a set of building blocks for moving from one goal to the next. We introduce the terminology and philosophy for constructing patterns by way of the simplest possible examples: a single AGV on a graph. This avoids the additional topological complications present in the context of cooperative motion.

To this end, we introduce the class of *edge point fields* as a dynamical toolbox for a hybrid controller. Given a specified goal point $g \in e_j$ within an edge of Γ, an *edge point field* is a locally defined vector field X_g on Γ with the following properties:

Locally defined. X_g is defined on a neighborhood $\mathcal{N}(e_j)$ of the goal edge e_j within the graph topology, and forward orbits under X_g are uniquely defined.

Point attractor. Every forward orbit of X_g asymptotically approaches the unique fixed point $g \in e_j$.[2]

Navigation-like. X_g admits a C^0 Lyapunov function, $\Phi_g : \Gamma \to R$.

The following existence lemma (whose trivial proof we omit) holds.

LEMMA 1. *Given any edge $e_j \subset \Gamma$ which is contractible within Γ, there exists an edge point field X_g for any desired goal $g \in e_j$.*

As a remark, we note that, as is usual in the traditional dynamical systems settings, the orbits of an edge point field may take an infinite amount of time to reach their destination. We can always rectify this situation by modifying the flow in a neighborhood of the goal via a sublinear term, e.g., $\dot{x} = -x^{1/3}$. This comment applies to vector fields used throughout the remainder of this work.

2.3. Discrete regulation of patterns. By an excursion on a graph, we mean a (possibly infinite) sequence of edges from the graph, $E = e_{i_1} \dots e_{i_N} \dots \in \mathcal{E}^Z$, having

[2] When it is not clear from the context, we shall denote the goal point achieved by an edge point flow as $\mathsf{g}(X_g) = \{g\}$.

the property that each pair of contiguous edges e_{i_j} and $e_{i_{j+1}}$ share a vertex in common. The set of excursions forms a language, $\mathcal{L}$, the so-called *subshift* on the alphabet defined by the named edges (we assume each name is unique) [13]. Given a legal block, $B = e_{i_1} \ldots e_{i_M} \in \mathcal{L}$, we say that an excursion realizes that pattern if its periodic extension eventually reaches the "goal" $BBBBB\ldots$ under the iterates of the block shift. In other words, after some transient behavior, the excursion consists of repetitions of the block B (terminating possibly with the empty edge).

In a previous paper [5], Burridge, Rizzi, and Koditschek introduced a very simple but effective discrete event controller for regulating patterns on abstract graphs representing a "prepares" relation imposed on families of controllers over general smooth manifolds. We introduce this prepares relation below and prune it as in [5]. The resulting ordering on the controllers yields a controller transition logic that enlarges the basin of any one member of the family to include the union of all "higher" controllers. This simple idea has a much longer history. It was in introduced in robotics as "preimage backchaining" [14], pursued in [15] as a method for building verifiable hardened automation via the metaphor of a family of funnels, and pursued in [7] as a means of prescribing sensor specifications from goals and action sets. In the discrete event systems literature, an optimal version of this procedure has been introduced in [4], and a generalization has recently been proposed in [18].

Let $\mathcal{E}^0{:=}B \subset \mathcal{E}$ denote the edges of Γ that appear in the block of letters specifying the desired pattern. Denote by

$$\mathcal{E}^{n+1} \subset \mathcal{E} - \bigcup_{k \leq n} \mathcal{E}^k$$

those edges that share a vertex with an edge in $\mathcal{E}^n$ but are not in any of the previously defined subsets. This yields a finite partition of $\mathcal{E}$ into "levels," $\{\mathcal{E}^p\}_{p=0}^P$, such that for each edge, $e_i^p \in \mathcal{E}^p$, there can be found a legal successor edge, $e_j^{p-1} \in \mathcal{E}^{p-1}$, such that $e_i^p e_j^{p-1} \in \mathcal{L}$ is a legal block in the language. Note that we have implicitly assumed that $\mathcal{E}^0$ is reachable from the entire graph—otherwise, there will be some "leftover" component of $\mathcal{E}$ forming the last cell in the partition starting within which it is not possible to achieve the pattern. Note as well that we impose some ordering of each cell $\mathcal{E}^p = \{e_i^p\}_{i=1}^{M_p}$: the edges of $\mathcal{E}^0 = B$ are ordered by their appearance in the block; the ordering of edges in higher level cells is arbitrary.

We may now define a "graph control" law $G{:}\mathcal{E} \to \mathcal{E}$ as follows. From the nature of the partition $\{\mathcal{E}^p\}$ above, it is clear that the least legal successor function,

$$L(p,i){:=}\begin{cases} i+1 \bmod M & : p=0, \\ \min\{j \leq M_p : e_i^p e_j^{p-1} \in \mathcal{L}\} & : p>0 \end{cases} \tag{2.2}$$

is well defined. From this, we construct the graph controller:

$$G(e_i^p){:=}e_{L(p,i)}^{p-1}. \tag{2.3}$$

It follows almost directly from the definition of this function that its successive application to any edge leads eventually to a repetition of the desired pattern.

PROPOSITION 2. *The iterates of G on $\mathcal{E}$ achieve the pattern B.*

2.4. Hybrid edge point fields. A semiflow, $(X)^t$, on the graph induces excursions in $\mathcal{L}$ parametrized by an initial condition as follows. The first letter corresponds to the edge in which the initial condition is located. (Initial conditions at vertices are assigned to the incident edge along which the semiflow points.) The next letter is added to the sequence by motion through a vertex from one edge to the next.

We will say of two edge point fields X_1, X_2 on a graph, Γ, that X_1 *prepares* X_2, denoted $X_1 \succ X_2$, if the goal of the first is in the domain of attraction of the second, $\mathsf{g}(X_1) \subset \mathcal{N}(X_2)$. Given any finite collection of edge point fields on Γ, we will choose some $0 < \alpha < 1$ and assume that their associated Lyapunov functions have been scaled in such a fashion that $X_1 \succ X_2$, implies $(\Phi_1)^{-1}[0, \alpha] \subset \mathcal{N}(X_2)$. In other words, an α crossing of the trajectory $\Phi_1 \circ (X_1)^t$ signals arrival in $\mathcal{N}(X_2)$.

Suppose now that for every edge in some pattern block, $e_i^0 \in \mathcal{E}^0$, there has been designated a goal point g_i^0 along with an edge point field X_i^0 taking that goal: $\mathsf{g}(X_i^0) = g_i^0$. Assume as well that the edge point field associated with each previous edge in the pattern prepares the flow associated with the next edge; in other words, using the successor function (2.2), we have

$$\mathsf{g}\left(X_j^0\right) \subset \mathcal{N}\left(X^0_{L(p,j)}\right).$$

Now construct edge point fields on all the edges of Γ such that the tree representation of their $\succ$ relations is exactly the tree pruned from the original graph above:

$$\mathsf{g}\left(X_j^p\right) \subset \mathcal{N}\left(X^{p-1}_{L(p,j)}\right).$$

We are finally in a position to construct a hybrid semiflow on Γ. This feedback controller will run the piecewise smooth vector field, $\dot{x} = X$, as follows:

$$X := \begin{cases} X_j^p & : x \in e_j^p \text{ and } \Phi_j^p > \alpha, \\ X^{p-1}_{L(p,j)} & : x \in e^{p-1}_{L(p,j)} \text{ or } \Phi_j^p \leq \alpha. \end{cases} \tag{2.4}$$

It is clear from the construction that progress from edge to edge of the state of this flow echoes the graph transition rule G constructed above.

Proposition 3. *The edge transitions induced by the hybrid controller* (2.4) *are precisely the iterates of the graph map G* (2.3) *in the language $\mathcal{L}$.*

3. The Y-graph. We now turn our attention to the safe control of multiple AGVs on a graph work space via vector fields. Whereas the case of a single AGV on a graph could be controlled by vector fields on the graph itself, the safe coordination of multiple agents necessitates vector field controls on the appropriate configuration space—a space whose topological features are by no means obvious.

For the remainder of this work, we consider the simplest example of a nontrivial configuration space: that associated with the *Y-graph*, Υ, having four vertices $\{v_i\}_0^3$ and three edges $\{e_i\}_1^3$. Each edge e_i attaches a vertex v_i to the central vertex v_0. Although this is a simple scenario compared to what one finds in a typical setting, there are several reasons why this example is in many respects canonical.

1. *Simplicity.* Any graph may be constructed by gluing index-K radial graphs together for various K. The $K = 3$ model we consider is the simplest nontrivial case and is instructive for understanding the richness and challenges of local cooperative dynamics on graphs.
2. *Genericity with respect to graphs.* Graphs which consist of copies of Υ glued together, the *trivalent graphs*, are generic: any nontrivial graph may be perturbed in a neighborhood of the vertex set so as to be trivalent. For example, the 4-valent graph resembling the letter "X" may be perturbed slightly to

resemble the letter "H"—a trivalent graph. An induction argument shows that this is true for all graphs. Hence the dynamics on an arbitrary graph are approximated by patching together dynamics on copies of Υ.

3. *Genericity with respect to local dynamics.* Finally, pairwise local AGV interactions on an arbitrary graph restrict themselves precisely to the dynamics of two agents on Υ as follows. Given a vertex v of a graph Γ, assume that two AGVs x and y are on different edges e_1 and e_2 incident to v and moving toward v with the goal of switching positions. A collision is imminent unless one AGV "moves out of the way" onto some other edge e_3 incident to v. The local interactions thus restrict themselves to dynamics of a pair of AGVs on the subgraph defined by $\{v; e_1, e_2, e_3\}$. Hence the case we treat in this paper is the generic scenario for the local resolution of collision singularities in cooperative dynamics on graphs and forms a basis for decentralized control of large numbers of independent agents.

3.1. Intrinsic coordinates. The configuration space $\mathcal{C}$ of two points on Υ is a subset of the cartesian product $\Upsilon \times \Upsilon$. Since Υ (and indeed any graph which is physically relevant to the setting of this paper) is embedded in a factory floor or ceiling and thus planar, the configuration space $\mathcal{C}$ embeds naturally in R^4. We wish to modify this embedding to facilitate both analysis on and visualization of the configuration space. We will present alternate embeddings in both higher and lower dimensional Euclidean spaces for these purposes.

We begin by representing the configuration space within a higher dimensional Euclidean space via *intrinsic* coordinates—coordinates independent of the graphs embedded in space. We illustrate this coordinate system with the Y-graph Υ, noting that a few simple modifications yield coordinate schemes for general graphs.

Let $\{e_i\}_1^3$ denote the three edges in Υ, parametrized so that the closure of each edge e_i is identified with $[0, 1]$ oriented so that $[0, 1]$ is mapped to $[v_0, v_i]$. Any point in Υ is thus given by a vector x in the $\{e_i\}$ basis whose magnitude $|x| \in [0, 1]$ determines the position of the point in the e_i direction. For $|x| > 0$, denote by $\iota(x)$ the value of i so that $x = |x|e_{\iota(x)}$. This parameterization embeds Υ as the positive unit axis frame in R^3. Likewise, a point in $\mathcal{C}$ is given as a pair of distinct vectors (x, y), i.e., as the positive unit axis frame in R^3 cross itself sitting inside of $R^3 \times R^3 \cong R^6$. We have thus embedded the configuration space of two distinct points on Υ in the positive orthant of R^6. It is clear that one can embed the more general configuration space of N points on Υ in R^{3N} in this manner.

This coordinate system is particularly well suited to describing vector fields on $\mathcal{C}$ and implementing numerical simulations of dynamics, as the coordinates explicitly track the physical position of each point on the graph.

3.2. A topological analysis. Visualizing $\mathcal{C}$ as a subset of R^4 or R^6 is unenlightening. More useful for visualization purposes is the following construction which embeds $\mathcal{C}$ within R^3.

THEOREM 1. *The configuration space $\mathcal{C}$ associated with a pair of AGVs restricted to the Y-graph Υ is homeomorphic to a punctured disc with six 2-simplices attached as per Figure 3.1.*

Proof. Recall that $\mathcal{C}$ consists of pairs of distinct vectors (x, y) in intrinsic coordinates. Restrict attention to the subspace $\mathcal{D} \subset \mathcal{C}$ defined by

$$\mathcal{D} := \{(x, y) \in \mathcal{C} : \iota(x) \neq \iota(y)\}, \tag{3.1}$$

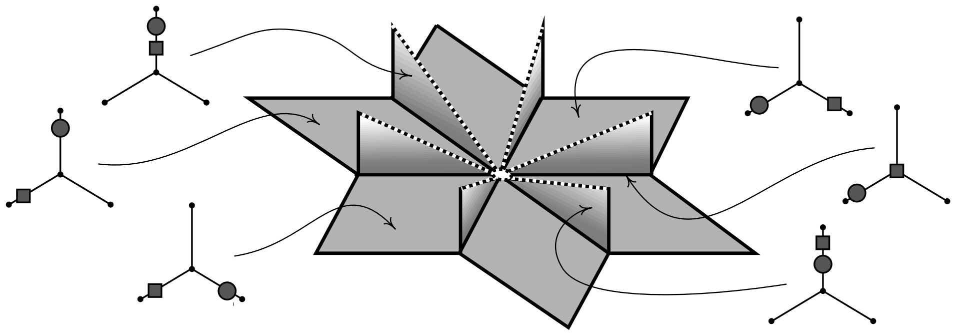

FIG. 3.1. *The configuration space $\mathcal{C}$ embedded in R^3. Dashed lines refer to open boundaries; sample configurations for representative* 2*-cells are illustrated to the sides.*

where an undefined index is considered to be not equal to one which is defined. Thus $\mathcal{D}$ consists of configurations for which both AGVs do not occupy the same edge interior.

The set $\mathcal{D}$ has a cellular decomposition as follows. There are 2 AGVs and 3 edges in Υ; hence there are $3 \cdot 2 = 6$ cells $\mathcal{D}_{i,j} \subset \mathcal{D}$, where $i := \iota(x) \neq \iota(y) =: j$. Since (the closure of) each edge in Υ is homeomorphic to $[0,1]$ (determined by $|\cdot|$), the cell $\mathcal{D}_{i,j}$ is homeomorphic to $([0,1] \times [0,1]) - \{(0,0)\}$, where, of course, the origin $(0,0)$ is removed as it belongs to the diagonal Δ. A path in $\mathcal{D}$ can move from cell to cell only along the subsets where the index of one AGV changes, e.g., $|x| = 0$ or $|y| = 0$. Thus the edges $\{0\} \times (0,1]$ and $(0,1] \times \{0\}$ of the punctured square $\mathcal{D}_{i,j}$ are attached, respectively, to $\mathcal{D}_{k,j}$ and $\mathcal{D}_{i,k}$, where k is the unique index not equal to i or j.

Furthermore, each 2-cell $\mathcal{D}_{i,j}$ has a product structure as follows: decompose $\mathcal{D}_{i,j}$ along the lines of constant $\theta := \tan^{-1}(\frac{|y|}{|x|})$. It is clear that θ is the angle in the unit's first quadrant in which $\mathcal{D}_{i,j}$ sits. Hence each $\mathcal{D}_{i,j}$ is decomposed into a product of a closed interval $S_{i,j} := \theta \in [0, \pi/2]$ (an "angular" coordinate) with the half-open interval $(0,1]$ (a "radial" coordinate). As this product decomposition is respected along the gluing edges, we have a decomposition of all of $\mathcal{D}$ into the product of $(0,1] \times S$, where S is a cellular complex given by gluing the six segments $S_{i,j}$ end-to-end cyclically along their endpoints. The set S is a 1-manifold without boundary since each $S_{i,j}$ is a closed interval, each of whose endpoints is glued to precisely one other $S_{i,j}$. Hence, by the classification of 1-manifolds, S is homeomorphic to a circle. We have thus decomposed $\mathcal{D}$ as the cross product of a circle with $(0,1]$—a punctured unit disc.

The complement of $\mathcal{D}$ in $\mathcal{C}$ consists of those regions where $\iota(x) = \iota(y)$. For each $i = 1 \ldots 3$, the subset of $\mathcal{C}$ where $\iota(x) = \iota(y) = i$ is homeomorphic to $((0,1] \times (0,1]) - \{|x| = |y|\}$: this consists of two disjoint triangular "fins." A total of six such fins are thus attached to $\mathcal{D}$ along the six edges where $|x|$ or $|y| = 0$. In the coordinates of the product decomposition for $\mathcal{D}$, these fins emanate along the radial lines where θ equals zero or $\pi/2$, yielding the topological space illustrated in Figure 3.1. ∎

COROLLARY 4. *Given any point goal $g \in \mathcal{D} \subset \mathcal{C}$, there exists an explicit navigation function (of class piecewise real-analytic) generating a semiflow which sends all but a measure-zero set of initial conditions to g under the gradient semiflow.*

Proof. The subset $\mathcal{D} \subset \mathcal{C}$ is homeomorphic to a punctured disc $S \times (0,1]$ and may easily be compactified to an annulus with boundary $S \times [\epsilon, 1]$ by removing an open neighborhood of the diagonal. Then the conditions for the theorems of Koditschek and Rimon [12] are met since an annulus is a *sphereworld*. Hence not only does a navigation function Φ on this subspace exist, but an explicit procedure for determining Φ is given [12]. One may then extend Φ to the remainder of $\mathcal{C}$ as follows: choose a point (x, y) on the fin, and define

$$\Phi(x,y) := \begin{cases} \dfrac{1}{1-|x|}\Phi(0,y), & |x| < |y|, \\ \dfrac{1}{1-|y|}\Phi(x,0), & |y| < |x|, \end{cases} \tag{3.2}$$

so that Φ increases sharply along the fins.[3] This directs the gradient flow to monotonically "descend" away from the diagonal and onto $\mathcal{D}$. Note that $\mathcal{D}$ is forward-invariant under the dynamics and that, upon prescribing the vector field on the fins to point into $\mathcal{D}$, we have defined a semiflow and hence a well-defined navigational procedure. □

This result is very satisfying in the sense that it guarantees a navigation function by applying existing theory to a situation which, from the definition alone, would not appear to be remotely related to a sphereworld. However, a deeper analysis of configuration spaces of graphs [9] reveals that, for more than two AGVs, the configuration space of a graph is never a sphereworld.[4]

We thus consider alternate methods for realizing compatible goals by means of a vector field on the configuration space, focusing, in particular, on the use of attracting periodic orbits as a controller component in the "toolbox" for building up the sort of hybrid feedback laws to be considered carefully in section 4.

3.3. Example: A circulating flow. We begin with a simple example of a vector field on $\mathcal{C}$ which possesses an attracting limit cycle as a goal. This "circulating field," which cycles a pair of AGVs through states on the boundary of $\mathcal{D} \subset \mathcal{C}$, is a canonical example of (1) a meaningful semiflow with limit cycle and (2) a practical field for implementing collision avoidance in a hybrid controller (cf. item 3 in the preface to section 3). Figure 3.2 (right) illustrates the flow restricted to $\mathcal{D}$.

THEOREM 2. *There exists a piecewise-smooth vector field X on $\mathcal{C}$ which has the following properties:*

1. *X defines a nonsingular semiflow on $\mathcal{C}$.*
2. *The diagonal Δ is repelling with respect to X.*
3. *Every orbit of X approaches a unique attracting limit cycle on $\mathcal{C}$ which cycles through all possible ordered pairs of distinct edge states.*

Proof. Recall that $\mathcal{D}$ denotes that portion of the configuration space corresponding to a placement of the AGVs on distinct edges of the graph; from the proof of Theorem 1, $\mathcal{D}$ is homeomorphic to a punctured disc. The intrinsic coordinates on the configuration space $\mathcal{C}$ are illustrated in Figure 3.2 (left), where only $\mathcal{D}$ is shown for simplicity. The reader should think of this as a collection of six square coordinate planes, attached together pairwise along axes with the origin removed.[5] The six triangular fins are then attached as per Figure 3.1.

Recall that any point in the graph is represented as a vector $x = |x|e_i$ for some i. Denote by $\hat{e}_i$ the unit tangent vector in each tangent space $T_x e_i$ pointing in the

[3] This construction does not satisfy the formal requirements for a "navigation function" since it is not bounded on the closure of the configuration space. There is a straightforward procedure detailed in [12] that can be used to complete the construction.

[4] Any configuration space of any graph is *aspherical*; there are no essential closed spheres of dimension larger than one, in contrast to a sphereworld. Thus, although a navigation is guaranteed to result in any case, the explicit constructions [12] are inapplicable.

[5] In the natural product metric on $\mathcal{C}$, these six 2-cells are flat Euclidean squares.

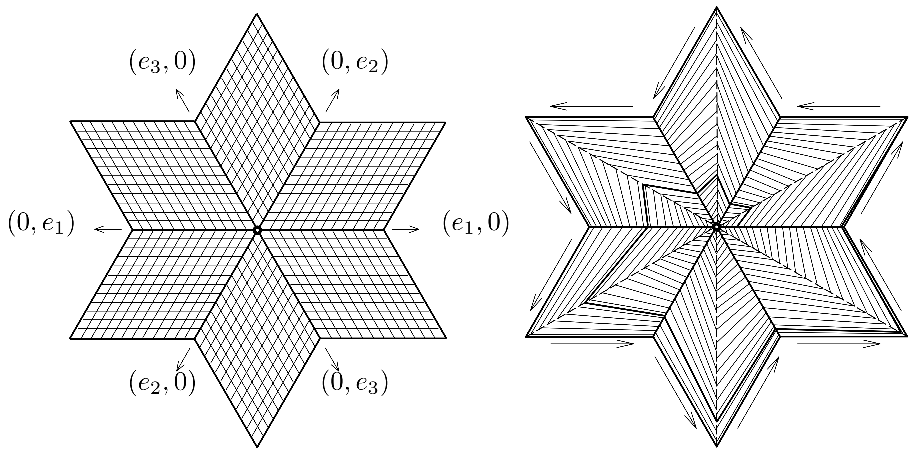

FIG. 3.2. *Left: The coordinate system on the unfinned region $\mathcal{D}$ of $\mathcal{C}$. Right: The circulating flow with a typical orbit.*

positive (outward) direction toward the endpoint v_i. The vector field we propose is the following. Given $(x, y) \in \mathcal{C}$,

1. if $\iota(x) = \iota(y)$, then

$$\left.\begin{cases} \dot{x} = -|y|\hat{e}_{\iota(x)}, \\ \dot{y} = |y|(1-|y|)\hat{e}_{\iota(y)}, \end{cases}\right\} \quad 0 < |x| < |y|, \qquad \left.\begin{cases} \dot{x} = |x|(1-|x|)\hat{e}_{\iota(x)}, \\ \dot{y} = -|x|\hat{e}_{\iota(y)}, \end{cases}\right\} \quad 0 < |y| < |x|; \tag{3.3}$$

2. if $\iota(x) = \iota(y) + 1$ or $|x| = 0$, then

$$\left.\begin{cases} \dot{x} = |y|\hat{e}_{(\iota(y)+1)}, \\ \dot{y} = |y|(1-|y|)\hat{e}_{\iota(y)}, \end{cases}\right\} \quad 0 \leq |x| < |y|, \qquad \left.\begin{cases} \dot{x} = |x|(1-|x|)\hat{e}_{\iota(x)}, \\ \dot{y} = -|x|\hat{e}_{\iota(y)}, \end{cases}\right\} \quad 0 < |y| \leq |x|; \tag{3.4}$$

3. if $\iota(y) = \iota(x) + 1$ or $|y| = 0$, then

$$\left.\begin{cases} \dot{x} = -|y|\hat{e}_{\iota(x)}, \\ \dot{y} = |y|(1-|y|)\hat{e}_{\iota(y)}, \end{cases}\right\} \quad 0 < |x| \leq |y|, \qquad \left.\begin{cases} \dot{x} = |x|(1-|x|)\hat{e}_{\iota(x)}, \\ \dot{y} = |x|\hat{e}_{(\iota(x)+1)}, \end{cases}\right\} \quad 0 \leq |y| < |x|. \tag{3.5}$$

Note that all addition operations on $\iota(x)$ and $\iota(y)$ are performed mod three.

The vector field is nonsingular as follows: if $|x||y| \neq 0$, then the vector field is by inspection nonsingular. If $|x| = 0$, then $|y| > 0$ since the points are distinct. It then follows from (3.4) that the vector field on this region has $d|x|/dt = |y| \neq 0$. A similar argument holds for the case where $|y| = 0$.

The vector field defines a semiflow as follows: on those regions where $0 \neq |x| \neq |y| \neq 0$, the vector field is smooth and hence defines a true flow. Along the lines where $|x| = |y|$, the vector field is not smooth but nevertheless is constructed so as to define unique solution curves; hence the region $\mathcal{D}$, where $\iota(x) \neq \iota(y)$, is invariant under the flow. Finally, along the branch line curves where $|x| = 0$ or $|y| = 0$, the vector field points into the branch lines from the fins, implying that the dynamics is a semiflow (see the remarks in Appendix A).

This vector field admits a C^0 Lyapunov function $\Phi : \mathcal{C} \to [0, 1)$ of the form

$$\Phi(x, y) := \begin{cases} 1 - |(|x| - |y|)| & : \iota(x) = \iota(y), \\ 1 - \max\{|x|, |y|\} & : \iota(x) \neq \iota(y). \end{cases} \tag{3.6}$$

From (3.3), one computes that on the fins (where $\iota(x) = \iota(y)$),

$$\frac{d\Phi}{dt} = -\left|\left(\frac{d|x|}{dt} - \frac{d|y|}{dt}\right)\right| < 0 \tag{3.7}$$

since here $|x| \neq |y|$. Furthermore, on the disc $\mathcal{D}$ ($\iota(x) \neq \iota(y)$), Φ changes as $\frac{d\Phi}{dt} = \Phi(\Phi - 1)$. Hence Φ strictly decreases off of the boundary of the disc

$$\partial\mathcal{D} := \{(x, y) : |x| = 1 \text{ or } |y| = 1\} = \Phi^{-1}(0). \tag{3.8}$$

It follows from the computation of $d\Phi/dt$ that the diagonal set Δ of $\Upsilon \times \Upsilon$ is repelling and that the boundary cycle $\partial\mathcal{D}$ is an attracting limit cycle. ∎

This example illustrates how one can use a relatively simple vector field on the configuration space to construct a pattern which is free from collisions. Indeed, as part of a hybrid control scheme, one could use this circulating flow to resolve potential collisions between AGVs in a general setting by localizing the dynamics near a pairwise collision to those on a trivalent subgraph. In practice, the fact that the outer vertices of the Y-graph are never quite reached by an interior orbit is irrelevant: a near-approach suffices for any practical application.

4. Patterns and vector fields for monotone cycles. In this section, we consider the problem of constructing vector fields which are tuned to trace out specific collision-free patterns—scheduled series of visits to specific work stations by the pair of AGVs whose regularity we wish to achieve at steady state, and return back to from any temporary perturbation or disruption. We begin with a specification of a suitable language for describing patterns.

4.1. A grammar for patterns. The setting we envisage is as follows: the three ends of the graph Υ are stations at which an AGV can perform some function. The AGV pair is required to execute an ordered sequence of functions, requiring an interleaved sequence of visitations. In order to proceed with vector field controls for cooperative patterns, it is helpful to construct the appropriate symbolic language, as introduced in section 2 for single AGV systems. Denote the pair of AGV states as x and y, respectively. Also, denote the three docking stations as vertices v_1 through v_3 as in Figure 3.1. The grammar $\mathcal{G}$ we use is defined as follows:

- (**xi**): These represent configurations for which the AGV x is docked at the vertex v_i, $i = 1 \ldots 3$. The AGV y is at an unspecified undocked position.
- (**yi**): These represent configurations for which the AGV y is docked at the vertex v_i, $i = 1 \ldots 3$. The AGV x is at an unspecified undocked position.
- (**xiyj**): These represent configurations for which the AGV x is docked at vertex v_i, while the AGV y is simultaneously docked at the vertex v_j, $j \neq i$.

For example, the word (**x1**)(**y2**)(**x3y2**) executes a sequence in which the first AGV docks at Station v_1 and then undocks while the second AGV docks at Station v_2. Finally, the AGVs simultaneously dock at Stations v_3 and v_2, respectively.

As we have assumed from the beginning, the one-dimensional nature of the graph-constraints precludes the presence of multiple agents at a single docking station; hence there are exactly twelve symbols in the grammar $\mathcal{G}$. From this assumption, it follows that particular attention is to be paid to those trajectories which do not make excursions onto the "fins" of the configuration space. It is obvious from the physical nature of the problem that planning paths which involve traveling on the fins is not a locally optimal trajectory with respect to minimizing distance or elapsed time. It suffices to say that we restrict our attention for the moment to trajectories and limit cycles for patterns, in particular, which are constrained to the region $\mathcal{D} \subset \mathcal{C}$.

We identify each symbol with a region of the boundary of the unbranched portion of $\mathcal{C}$; namely, $\partial\mathcal{D}$ is partitioned into twelve *docking zones* as in Figure 4.1. Note further that there is a cyclic ordering, $\prec$, on $\mathcal{G}$ induced by the orientation on the boundary of the disc along which the zones lie. By a cyclic ordering, we mean a way of determining whether a point q lies between any ordered pair of points (p_1, p_2).

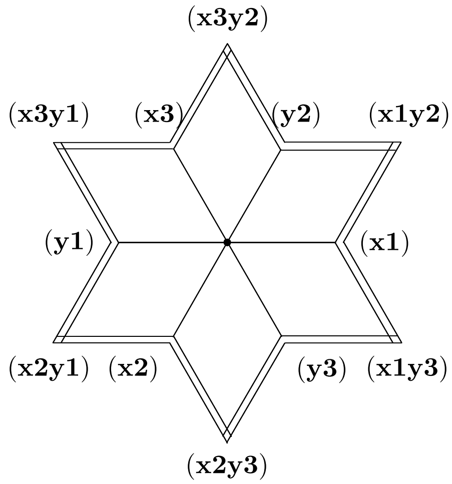

FIG. 4.1. *Labels for the cyclically ordered grammar* $\mathcal{G}$.

We proceed with the analysis of limit cycles on $\mathcal{C}$. Consider the class of *pattern vector fields*, $\mathcal{X}_P$, on $\mathcal{C}$ defined as follows. For every $X \in \mathcal{X}_P$,

1. X defines a semiflow on $\mathcal{C}$ and a true flow off the nonmanifold set of $\mathcal{C}$;
2. there is a unique limit cycle γ which is attracting and which traces out a nonempty word in the grammar $\mathcal{G}$;
3. the diagonal set Δ is a repeller with respect to X;
4. there are no fixed invariant sets of X which attract a subset of positive measure save γ.

Denote by $\mathcal{X}_M$ the subset of $\mathcal{X}_P$ for which the limit cycle, γ, lies in $\mathcal{D}$. The question of which words in the grammar $\mathcal{G}$ are admissible for the class $\mathcal{X}_P$ has a simple answer in terms of the cyclic ordering $\prec$. A word $\mathbf{w}$ composed of elements $\mathbf{w} = w_1 w_2 \dots w_n$ in the grammar $\mathcal{G}$ said to be *monotone* with respect to the cyclic ordering $\prec$ if $w_{i-1} \prec w_i \prec w_{i+1}$ for every i (index operations all mod n).

THEOREM 3. *Within the class of vector fields $\mathcal{X}_M$, the limit cycles trace out monotone words in the cyclically ordered grammar $(\mathcal{G}, \prec)$.*

Proof. The idea of the proof is simple and follows from the observation that any limit cycle of the flow must be embedded (the curve does not intersect itself). After a small perturbation, one may assume that the boundary zone $\partial\mathcal{D}$ is visited by γ in a finite number of points, $Q := \gamma \cap \partial\mathcal{D}$. Consider two points $p, q \in Q$, which are consecutive in the limit cycle: that is, there is an embedded subarc $\alpha \subset \gamma$ which connects p to q within the interior of $\mathcal{D}$. The arc α separates $\mathcal{D}$ into two topological discs (this is the Jordan curve theorem [17]); hence γ must lie entirely within the closure of one of these discs. This implies that the limit cycle cannot visit any point $x \in \partial\mathcal{D}$ satisfying $p \prec x \prec q$. Repeating this argument for all pairs of consecutive points yields the monotonicity property. ∎

Although the only admissible words in the grammar $\mathcal{G}$ are those which are monotone, it is possible to realize many if not all of the nonmonotone cycles as limit cycles for a semiflow on the *full* configuration space $\mathcal{C}$; one must design the semiflow so as to utilize the fins for "jumping" over regions of $\mathcal{D}$ cut off by the limit cycle. Such vector fields quickly become very convoluted, even for relatively simple nonmonotone limit cycles, and a more explicit constructive procedure would need stronger motivation from the application domain than we are presently aware of.

4.2. Isotopy classes of limit cycles. Given a limit cycle γ which traces out a pattern by visiting the boundary zone $\partial\mathcal{D}$ in the ordered set $Q \subset \partial\mathcal{D}$, one wants to know which other limit cycles minimize a given performance functional while still visiting Q in the proper sequence. The mathematical framework for dealing with this problem is the notion of *isotopy classes* of curves.

Two subsets A_0 and A_1 of a set B are said to be (ambiently) *isotopic rel* C (where $C \subset B$) if there exists a continuous 1-parameter family of homeomorphisms $f_t : B \to B$ such that

1. f_0 is the identity map on B,
2. $f_1(A_0) = A_1$, and
3. $f_t|_C$ is the identity map on C for all t.

As t increases, f_t deforms B, pushing A_0 to A_1 without cutting or tearing the spaces and without disturbing C.

There are two ways in which optimization questions relate to isotopy classes of limit cycles: (1) Given an element of the grammar $\mathcal{G}$, in which isotopy class (rel the docking zones) of curves does an optimal limit cycle reside? (2) Within a given isotopy class of cycles rel Q, which particular cycle is optimal?

For a monotone limit cycle on $\mathcal{D}$, question (1) focuses on the location of the cycle with respect to the central point $(0, 0)$, which is deleted from the disc $\mathcal{D}$. It is a standard fact from planar topology that every curve in the punctured disc has a well-defined *winding number*, which measures how many times the cycle goes about the origin, and, furthermore, that this number is -1, 0, or 1 if the cycle is an embedded curve. This winding number determines the isotopy class of the curve in $\mathcal{D}$. Hence the problem presents itself as follows: given an element of the grammar $\mathcal{G}$, which isotopy class rel the docking zones is optimal (with respect to any or all of the functionals defined)? Is the winding number zero or nonzero?[6]

To address this question, we define the *gap angles* associated to a limit cycle. For the remainder of this section, we will place standard polar coordinates on the region $\mathcal{D}$ (given as a subset of the plane as per Figures 3.2 and 4.1) with the central puncture corresponding to the origin. Given a set of "docked states"—or points $Q = \{q_1, q_2, \ldots, q_J\}$ ordered with respect to time—we define the gap angles to be the successive differences in the angular coordinates of the q_j: thus $\angle_j := P(q_{j+1}) - P(q_j)$, where P denotes projection of points in $\mathcal{D}$ onto their angular coordinates, and

[6]The difference between +1 and −1 is the orientation of time.

subtraction is performed with respect to the orientation on $\partial\mathcal{D}$.

For simplicity, we consider the optimization-isotopy problem in the case of a discrete cost functional $\mathbf{W}_d$, defined to be the intersection number of the path with the branch locus of $\mathcal{C}$, i.e., the number of times an AGV occupies the central vertex (the shared resource in the problem). Similar arguments are possible for other natural performance metrics.

PROPOSITION 5. *Given a cyclically ordered set of points $Q = \{q_j\}_1^J$ on the boundary of $\mathcal{D}$, consider the class of embedded monotone cycles on $\mathcal{D}$ which trace out the points of Q.*

1. *There is a $\mathbf{W}_d$-minimizing embedded monotone cycle on $\mathcal{D}$ having winding number zero with respect to the origin if there is a gap angle greater than π.*
2. *Conversely, if there are no gap angles greater than π, then there is a $\mathbf{W}_d$-minimizing embedded cycle of index ± 1.*

Proof. Define the gap angles $\{\angle_j\}_1^J$ to be the differences of the angles between the points q_j and q_{j+1} (in standard planar polar coordinates with all indices mod J). Since $\sum_j \angle_j = 2\pi$, there can be at most one gap angle greater than π. To simplify the problem, use a 1-parameter family P_t of maps from the identity P_0 to the projection $P = P_1$, which deforms $\mathcal{D}$ to the boundary circle $S := \partial\mathcal{D}$ by projecting along radial lines. The index of a curve on $\mathcal{D}$ is invariant under this deformation as is the functional $\mathbf{W}_d$.

Denote by γ_j the subarc of γ between points q_j and q_{j+1} (all indices mod J). Denote by α_j the subarc of the boundary S between points q_j and q_{j+1}, where the arc is chosen to subtend the gap angle $\angle_j$. Since the boundary curve $S = \cup_j \alpha_j$ is a curve of index ± 1, the arcs γ_j and α_j are isotopic in $\mathcal{D}$ rel their endpoints for all j if and only if γ is a curve of index ± 1.

Assume first that there is a gap angle $\angle_j > \pi$ with γ an index ± 1 curve on S which intersects the branch angles $\Theta = \{n\pi/3 : n \in Z\}$ in a minimal number of points among all other closed curves on S which visit the points Q in the specified order. It follows that the arc $P(\gamma_j)$ subtends an angle greater than π and thus increments $\mathbf{W}_d$ by at least three. One may replace γ_j by a curve γ_j', which substitutes for the arc γ_j, one which wraps around "the other way" monotonically. This changes the index of γ from nonzero to zero since the arc γ_j' is no longer isotopic to α_j. Also, it is clear that this either decreases the number of intersections with Θ or leaves this number unchanged.

We must show that the replacement arc γ_j' can be chosen in such a way that it does not intersect the remainder of γ. However, since γ is a curve of index ± 1, we may isotope each arc γ_i to the boundary curve α_i without changing the value of $\mathbf{W}_d$. Thus we may remove γ_j and replace it with the curve which is, say, a geodesic (in the natural metric geometry) from q_j to q_{j+1}. As this curve does not approach the boundary S apart from its ends, the new curve γ' is an embedded curve of index zero without an increase in $\mathbf{W}_d$.

Now assume, on the contrary, that γ is a $\mathbf{W}_d$ minimizer of index zero which has all gap angles strictly less than π. Then each arc from γ_i must intersect the branch set Θ in at most three components since, otherwise, the subtended arc would be in excess of $4\pi/3$. In the case where there exists an arc with exactly three intersections with the branch set, this arc may be replaced by an arc which goes around the singularity in the other direction without changing the number of intersections with the branch set (since there are a total of six branch lines); however, the index of the curve is toggled between zero and nonzero.

The final case is that in which each arc intersects the branch set in at most two places. However, since γ is a curve of index zero, some arc γ_j must not be isotopic to α_j. Hence the projection deformations P_t must push γ_j to a curve in the boundary S whose subtended gap angle is $2\pi - \angle_j > \pi$. Thus γ_j intersects the branch set in at least three places, yielding a contradiction. Replacing γ_j by the appropriate arc which is isotopic to α_j yields a $\mathbf{W}_d$-minimal cycle of nonzero index. □

4.3. Tuning cycles. Designing a customized "pattern" of two AGVs on the Y-graph is as simple as drawing a vector field on $\mathcal{C}$ with a stable limit cycle tracing out the desired motion. The problem then is how to specify such a vector field in coordinates. Since we focus on those limit cycles which are contained within $\mathcal{D}$, we can exploit the fact that $\mathcal{D}$ is topologically a punctured disc. We thus give an explicit coordinate-change between the natural polar coordinates on a disc and the intrinsic coordinates of section 3. Once we possess an explicit coordinate change (and its inverse), we can design a vector field in polar coordinates (an easy task to do in these coordinates) and then take the push-forward of the vector field under the coordinate change.

It will be convenient to keep track of which "wedge" of the annular region a point (r, θ) is. To do so, we introduce a parity function

$$P(\theta) := (-1)^{\{\lfloor 3\theta/\pi \rfloor + \lfloor 6\theta/\pi \rfloor\}}, \tag{4.1}$$

where $\lfloor t \rfloor$ is the integer-valued floor function. Recall the notation for the intrinsic coordinates for a point x on the graph Υ: $x = |x|\hat{e}_{\iota(x)}$, where $|x| \in [0,1]$ is the distance from x to the central vertex, and $\hat{e}_{\iota(x)}$ is the unit tangent vector pointing along the direction of the $\iota(x)$-edge. Here the index $\iota(x)$ is an integer (defined modulo 3) and will be undefined in the case when $|x| = 0$, i.e., x is at the central vertex.

LEMMA 6. *The following is a piecewise-linear homeomorphism from the punctured unit disc in R^2 to the subset $\mathcal{D}$. Define $F(r, \theta) = (x, y)$, where*

$$\begin{array}{llll} \iota(x) = \left\lfloor -\dfrac{3}{2\pi}(\theta - \pi) \right\rfloor & , & |x| = \left\{ \begin{array}{cc} r & \mathcal{P}(\theta) = +1, \\ r\left|\cot \frac{3}{2}\theta\right| & \mathcal{P}(\theta) = -1, \end{array} \right. \\ \iota(y) = \left\lfloor -\dfrac{3}{2\pi}\theta \right\rfloor & , & |y| = \left\{ \begin{array}{cc} r\left|\tan \frac{3}{2}\theta\right| & \mathcal{P}(\theta) = +1, \\ r & \mathcal{P}(\theta) = -1. \end{array} \right. \end{array} \tag{4.2}$$

The inverse of this homeomorphism is given by $F^{-1}(x, y) = (r, \theta)$, where

$$\begin{array}{l} \theta = \left\{ \begin{array}{ll} \frac{2}{3}\tan^{-1}\frac{|y|}{|x|} - \frac{2\pi}{3}(\iota(y) + 1), & \begin{array}{c}\iota(y) = \iota(x) + 1, \\ \textit{or } |x| = 0,\end{array} \\ -\frac{2}{3}\tan^{-1}\frac{|y|}{|x|} - \frac{2\pi}{3}(\iota(x) - 1), & \begin{array}{c}\iota(x) = \iota(y) + 1, \\ \textit{or } |y| = 0,\end{array} \end{array} \right. \\ r = \left\{ \begin{array}{ll} |x| & \mathcal{P}(\theta) = +1, \\ |y| & \mathcal{P}(\theta) = -1. \end{array} \right. \end{array} \tag{4.3}$$

Note that all θ values are defined modulo 2π, and all index values are integers defined modulo 3.

Proof. Begin by working on the region $\mathcal{D}_{1,2} \subset \mathcal{D}$, where $\iota(x) = 1$ and $\iota(y) = 2$. As noted earlier, this subspace is isometric to the positive unit square in R^2 with the origin removed. We need to map this to the subset $\{(r, \theta) : r \in (0, 1], \theta \in [0, \pi/3]\}$. The simplest such homeomorphism is to first shrink along radial lines, leaving the angle invariant; hence

$$r = \begin{cases} |x| & : |x| \leq |y|, \\ |y| & : |y| \leq |x|. \end{cases} \tag{4.4}$$

Next, we squeeze the quarter-circle into a sixth of a circle by multiplying the angle by 2/3, leaving the radial coordinate invariant:

$$\theta = \frac{2}{3} \tan^{-1} \frac{|y|}{|x|}. \tag{4.5}$$

This gives the basic form of F^{-1} as per (4.3). To extend this to the remainder of $\mathcal{D}$, it is necessary to carefully keep track of $\iota(x)$ and $\iota(y)$ and subtract the appropriate angle from the computation of θ. Also, the condition of $|x| \leq |y|$, etc., in (4.4) is incorrect on other domains of $\mathcal{D}$ since the inequalities flip as one traverses from square to square: the parity function $\mathcal{P}(\theta)$ keeps track of which "wedge" one is working on.

To determine F from F^{-1} is a tedious but unenlightening calculation, made more unpleasant by the various indices to be kept track of. Briefly, given r and θ on the first sixth of the unit disc, one knows from (4.4) that either $|x| = r$ or $|y| = r$, depending on whether θ is above or below $\pi/4$. To solve for the other magnitude, one inverts (4.5) to obtain $|y| = r\left|\tan \frac{3}{2}\theta\right|$ or $|x| = r\left|\cot \frac{3}{2}\theta\right|$, respectively. To generalize this to the other $\mathcal{D}_{i,j}$ domains of $\mathcal{D}$, it is necessary to take absolute values and to use the parity function $\mathcal{P}(\theta)$ as before. Finally, the computation of the index is obtainable from the combinatorics of the coordinate system as illustrated in Figure 3.2. ∎

For the design of limit cycles, it is easier to work on the polar disc and write out an explicit vector field $X = (\dot{r}, \dot{\theta})$ with a limit cycle. To transform this into intrinsic coordinates, one takes the push-forward of X with respect to F, obtaining the piecewise-smooth vector field

$$\begin{cases} \begin{pmatrix} \dot{|x|} = \dot{r}, \\ \dot{|y|} = \dot{r}\left|\tan(\frac{3}{2}\theta)\right| + \frac{3}{2}r\dot{\theta}\sec^2(\frac{3}{2}\theta) \end{pmatrix} & \mathcal{P}(\theta) = +1, \\ \begin{pmatrix} \dot{|x|} = \dot{r}\left|\cot(\frac{3}{2}\theta)\right| + \frac{3}{2}r\dot{\theta}\csc^2(\frac{3}{2}\theta), \\ \dot{|y|} = \dot{r} \end{pmatrix} & \mathcal{P}(\theta) = -1, \end{cases} \tag{4.6}$$

which simplifies to

$$\begin{cases} \begin{pmatrix} \dot{|x|} = \dot{r}, \\ \dot{|y|} = \dot{r}\dfrac{|y|}{|x|} + \dfrac{3}{2}\dot{\theta}\dfrac{|x|}{1+\left(\frac{|y|}{|x|}\right)^2} \end{pmatrix} & \mathcal{P}(\theta) = +1, \\ \begin{pmatrix} \dot{|x|} = \dot{r}\dfrac{|x|}{|y|} + \dfrac{3}{2}\dot{\theta}\dfrac{|y|}{1+\left(\frac{|x|}{|y|}\right)^2}, \\ \dot{|y|} = \dot{r} \end{pmatrix} & \mathcal{P}(\theta) = -1. \end{cases} \tag{4.7}$$

We present a more explicit example. Given a simple closed curve γ in R^2 which has nonzero winding number with respect to the origin, γ may be parametrized as $\{(r, \theta) : r = f(\theta)\}$ for some periodic positive function f. To construct a vector field on R^2 whose limit sets consist of the origin as a source and γ as an attracting limit cycle, it suffices to take the push-forward of the vector field $\dot{r} = r(1-r)$, $\dot{\theta} = \omega$ under the planar homeomorphism $\phi : (r, \theta) \mapsto (f(\theta)r, \theta)$, which rescales linearly in the angular component. The calculations follow:

$$\begin{aligned} \phi_* \begin{pmatrix} \dot{r} \\ \dot{\theta} \end{pmatrix} &= D\phi \begin{pmatrix} \dot{r} \\ \dot{\theta} \end{pmatrix}\Bigg|_{r \mapsto \frac{r}{f}} = \begin{bmatrix} f & rf' \\ 0 & 1 \end{bmatrix} \begin{pmatrix} r(1-r) \\ \omega \end{pmatrix}\Bigg|_{r \mapsto \frac{r}{f}} \\ &= \begin{pmatrix} r\left(1 - \dfrac{r - f'\omega}{f}\right) \\ \omega \end{pmatrix}. \end{aligned} \tag{4.8}$$

Hence, given $f(\theta)$, we may tune a vector field to trace out the desired limit cycle and then use (4.2) and (4.3) to map it into intrinsic coordinates.

4.4. Optimal chords within a hybrid controller. To design optimal cycles with winding number zero, then we turn to constructing customized portions of limit cycles, or *chords* which can be pieced together via a state-actuated hybrid controller, much as in section 2. In other words, instead of building a simple fixed vector field with a limit cycle, we will use a set of vector fields which vary discretely in time and which may be pieced together so as to tune a limit cycle to the desired specifications. There is nothing in this construction which relies on the index-zero property, and thus these chords can be used to generate all monotone limit cycles on $\mathcal{C}$.

Let G denote a word representing a desired monotone limit cycle on the configurations space $\mathcal{C}$. Choose points $\{q_i\}$ on the boundary of $\mathcal{D}$ which correspond to the docking zones for the cycle given by G. Choose arcs α_i on $\mathcal{D}$ which connect q_i to q_{i+1} (using cyclic index notation). The arcs α_i are assumed given in the intrinsic coordinates on $\mathcal{D}$, as would be the case if one were determining a length-minimizing curve.

In the case where the limit cycle $\alpha := \cup_i \alpha_i$ is an embedded curve of nonzero index, the procedure of the previous subsection determines a vector field X_α on $\mathcal{C}$ which realizes α as an attracting limit cycle with the appropriate dynamics on the complementary region. Recall that one translates α to a curve on the disc model via the homeomorphism of (4.3). Then, representing the limit cycle α as a function $f_\alpha(\theta)$, one takes the vector field of (4.8) and, if desired, takes the image of this vector field under (4.7).

If, however, this is not the case, consider the arc α_j for a fixed j, and construct an index ± 1 cycle $\beta^j = \cup_i \beta^j_i$ which has docking zones $\{q_i\}$ such that $\beta^j_j = \alpha_j$. Then the vector field X^j as constructed above has β as an attracting limit cycle. Denote by Φ^j the Lyapunov function which measures proximity to β: $\Phi^j(p) := \left\|p - \beta^j\right\|$ (with distance measured in say the product metric on $\mathcal{C}$). Then consider the modified Lyapunov function $\Psi^j(p) := \Phi^j(p) + \|p - q_{j+1}\|$, which measures the distance to the endpoint of the arc β^j_j in addition to the proximity to β^j.

Repeat this procedure for each j, yielding the vector fields $\{X^j\}$ which attract, respectively, to limit cycles β^j. It follows that X^j prepares X^{j+1} since the goal point of X^j, q_{j+1} lies on the attracting set of X^{j+1}. The Lyapunov functions $\{\Psi^j\}$ serve as a set of funnels which channel the orbit into the sequence of arcs α_j, forming α. One scales the Ψ^j so that a $\Psi^j < \epsilon$ event triggers the switching in the hybrid controller from X^j to X^{j+1}:

$$X := \begin{cases} X^1 & : \quad \Phi^j > \epsilon \quad \forall j, \\ X^j & : \quad \Phi^j < \epsilon \text{ and } \Psi^j > \epsilon. \end{cases} \tag{4.9}$$

By construction, the hybrid controller (4.9) realizes a limit cycle within ϵ of α as the attracting set.

5. Future directions. A point of primary concern is the adaptability of the global topological approach to systems which increase in complexity, either through more intricate graphs or through increased numbers of AGVs. The latter is of greater difficulty than the former since the dimension of the resulting configuration space is equal to the number of AGVs. Hence, no matter how simple the underlying graph is, a system with ten independent AGVs will require a dynamical controller on a (topologically complicated) ten-dimensional space—a formidable problem both from the topological, dynamical, and computational viewpoints.

However, there are some approaches which may facilitate working with such spaces. Consider the model space $\mathcal{C}$ with which this paper is concerned: although a two-dimensional space $\mathcal{C}$ is homeomorphic to the product of a graph (a circle with six radial edges attached) with the interval $(0, 1]$. In fact, if we consider the circulating flow of (3.3)–(3.5), one can view this as a product field of a semiflow on the graph (which "circulates") with a vector field on the factor $(0, 1]$ (which "pushes out" to the boundary).

A similar approach is feasible for arbitrary graphs [9].

THEOREM 4. *Given any graph* Γ *(except the graph homeomorphic to a circle), the configuration space of* N *distinct points on* Γ *can be deformation retracted to a subcomplex whose dimension is bounded above by the number of vertices of* Γ *of valency greater than two.*[7]

This theorem implies the existence of low dimensional *spines* which carry all of the topology of the configuration space. For example, the configuration space of N points on the Y-graph can be continuously deformed to a one-dimensional graph, regardless of the size of N. Since the full space can be deformation retracted onto the spine, a vector field defined on the spine can be pulled back continuously to the full configuration space, thus opening up the possibility of reducing the control problem to that on a much "smaller" space. Additional results about the topology of configuration spaces on graphs may yield computationally tractable means of dealing with complex path planning: for example, having a presentation for the fundamental group of a configuration space of a graph in terms of a suitably simple set of cycles would be extremely well-suited to a hybrid control algorithm based on "localized" vector fields supported on small portions of the full configuration space.

Results connected with computational issues for configuration spaces of graphs are also being developed. Abrams has developed a "discretization" algorithm for converting the configuration space of a graph into a cubical complex [1]. This is then perfectly suited to the recent algorithms in computational homology [11] which prefer cube complex structures and can quickly determine geodesic paths.

The optimization problem is another avenue for inquiry. The fact that a dynamical approach allows for increased density of AGVs on a graph (as compared with blocking-zone strategies) would indicate an increased efficiency with respect to, say, elapsed time of flight. However, a more careful investigation of the tuning of optimal cycles is warranted. A careful treatment of the geometry of configuration spaces of graphs is essential to the optimization problem: it follows from the recent thesis of Abrams [1] that these spaces always possess a remarkable geometric property (*NPC* or *nonpositively curved*) which implies, among other things, that geodesics are unique

[7] Added in proof: A similar result has been shown in [16].

within their homotopy class. Such properties, though rare in the world of topological spaces, appear to be not at all uncommon among real-world robotic systems [2].

We believe that the benefits associated with using the full configuration space to tune optimal dynamical cycles justifies a careful exploration of these challenging spaces.

Appendix A. The topology and dynamics of graphs.

In this appendix, we provide a careful basis for the use of vector fields on configuration spaces of graphs. In the setting of manifolds, all of the constructions used in this paper are entirely natural and well defined. However, on spaces like $\mathcal{C}$, the most fundamental of notions (like the existence and uniqueness theorems for ODEs) are not in general valid.

We begin by defining vector fields on graphs. For present purposes, it is convenient to work with an intrinsic formulation (i.e., directly in the graph rather than via an embedding) of these objects. To this end, denote by v a vertex with K incident edges $\{e_i\}_1^K$ and by $\{X_i\}_1^K$ a collection of nonsingular vector fields locally defined on a neighborhood of the endpoint of each e_i (homeomorphic to $[0,1)$).

LEMMA 7. *A set of nonsingular vector fields* $\{X_i\}$ *on the local edge set of a graph* Γ *generates a well-defined semiflow on* Γ *if the following hold:*

1. *Each edge field* X_i *generates a well-defined local semiflow on* $(0,1)$.
2. *The magnitude of the endpoint vectors* $\|X_i(0)\|$ *(taken with respect to the attaching homeomorphisms) are all identical.*
3. *Among the* signs *of the endpoint vectors* $X_i(0)$ *(either positive if pointing into* $[0,\epsilon)$ *or negative if pointing out) there is a single positive sign.*

Proof. Since the vector field is well defined away from the vertex, it is only necessary to have the magnitudes $\|X_i(0)\|$ agree in order to have a well-defined function $\|X\|$ on Γ. In order to make this a well-defined field of directions, we must also consider in which direction the vector is pointing. Again, this is determined off of the vertex by (1). Condition (3) means that at the vertex there is a unique direction along which the vector field is pointing out: all other edges point in. Hence the direction field, as well as the magnitude field, is well defined.

The semiflow property follows naturally from this. Assume that the Nth edge of Γ has the positive sign. Then, given an initial point $x \in \Gamma$, if $x \in e_N$, then the orbit of x under the local field X_N remains in e_N and is well defined. If $x \in e_j$ for some $j \neq N$, then the union of the edges $e_j \cup e_N$ is a manifold homeomorphic to R on which the vector fields X_j and X_N combine to yield a well-defined vector field, since the directions are "opposite." As we are now on a manifold, the standard existence theorem implies that x has a forward orbit (which passes through the vertex and continues into e_N). Thus every point on Γ has a well-defined forward orbit. □

In the case where the vector fields have singularities, it is a simpler matter. If the singularities are not at the vertex, then there is no difference. If there is a singularity at the vertex, then condition (3) in Lemma 7 is void—all such vector fields are well defined.

In order to extend these results to the configuration space of this paper, consider the space $\mathcal{C} = \Upsilon \times \Upsilon - \Delta$, and let $(x,y) \in \mathcal{C}$ denote a point on the branch set of $\mathcal{C}$. Because of the structure of Υ and the fact that the diagonal points are deleted, it follows that at most one AGV may occupy a nonmanifold point of Υ. Hence a neighborhood of (x,y) in $\mathcal{C}$ has a natural product structure $N \cong \Upsilon \times R$. Let $P : N \to \Upsilon$ denote projection onto the first factor.

LEMMA 8. *A nonsingular vector field* X *on the individual cells of* $\mathcal{C}$ *generates* *a well-defined semiflow if* (1) *the projection of the local vector fields onto the graph factor,* $P_*(X|_{\{x\}\times\Upsilon})$*, satisfies Lemma* 7 *for each point* x *in the branch set of* $\mathcal{C}$ *and* (2) *the projections of the vector fields on the branch set to the* R*-factor are equal up to the attaching maps.*

Proof. Off of the branch set, the space is a manifold, and hence the vector field gives a well defined flow. If p is a point on the branch line, condition (2) implies that the vector field is well defined with respect to the attaching maps and the net effect in the R-factor is a drift in this direction. In the graph factor, condition (1) and the proof of Lemma 7 imply that there is a unique forward orbit through p. □

Intuitively, this condition means that, as in the case of a graph, the vector field must point "in" on all but one sheet of the configuration space in order to have well-defined orbits. We may thus lift the criteria of Lemma 7 to the product configuration space. All of the vector fields in this paper are so constructed.

REFERENCES

[1] A. Abrams, *Configuration Spaces of Graphs and Brownian Motion*, Ph.D. thesis, University of California at Berkeley, Berkeley, CA, 2000.

[2] A. Abrams and R. Ghrist, *Shape Complexes for Reconfigurable Robotic Systems*, in preparation, 2001.

[3] Y. A. Bozer and M. M. Srinivasan, *Tandem configurations for automated guided vehicle systems and the analysis of single vehicle loops*, IIE Transactions, 23 (1991), pp. 72–82.

[4] Y. Brave and M. Heymann, *On optimal attraction of discrete-event processes*, Inform. Sci., 67 (1993), pp. 245–276.

[5] R. R. Burridge, A. A. Rizzi, and D. E. Koditschek, *Sequential composition of dynamically dexterous robot behaviors*, Int. J. Rob. Res., 18 (1999), pp. 534–555.

[6] G. A. Castleberry, *The AGV Handbook*, Braun-Brumfield, Ann Arbor, MI, 1991.

[7] M. Erdmann, *Understanding action and sensing by designing actions-based sensors*, Int. J. Rob. Res., 14 (1995), pp. 483–509.

[8] S. B. Gershwin, *Manufacturing Systems Engineering*, Prentice-Hall, Englewood Cliffs, NJ, 1994.

[9] R. Ghrist, *Configuration spaces and braid groups on graphs in robotics*, in Braids, Links, and Mapping Class Groups: The Proceedings of Joan Birman's 70th Birthday, AMS/IP Stud. Adv. Math. 24, AMS, Providence, RI, 2001, pp. 29–39. ArXiv preprint math.GT/9905023.

[10] R. Ghrist and D. E. Koditschek, *Safe cooperative robot dynamics on graphs*, in Proceedings of the 8th International Symposium on Robotic Research, Y. Nakayama, ed., Springer-Verlag, New York, 1998, pp. 81–92.

[11] W. Kalies, K. Mischaikow, and G. Watson, *Cubical approximation and computation of homology*, in Conley Index Theory, Banach Center Publ. 47, Polish Acad. Sci., Warsaw, Poland, 1999, pp. 115–131.

[12] D. E. Koditschek and E. Rimon, *Robot navigation functions on manifolds with boundary*, Adv. Appl. Math., 11 (1990), pp. 412–442.

[13] D. Lind and B. Marcus, *An Introduction to Symbolic Dynamics and Coding*, Cambridge University Press, Cambridge, UK, 1995.

[14] T. Lozano-Perez, M. T. Mason, and R. H. Taylor, *Automatic synthesis of fine-motion strategies for robots*, Int. J. Rob. Res., 3 (1984), pp. 3–23.

[15] M. T. Mason, *The mechanics of manipulation*, in Proceedings of the IEEE International Conference on Robotics and Automation, IEEE Robotics and Automation Society, Piscataway, NJ, 1985, pp. 544–548.

[16] R. J. Milgram and S. Kaufman, *Topological Characterization of Safe Coordinated Vehicle Motion*, in preparation.

[17] J. R. Munkres, *Topology: A First Course*, Prentice-Hall, Englewood Cliffs, NJ, 1975.

[18] R. Sengupta and S. Lafortune, *An optimal control theory for discrete event systems*, SIAM J. Control Optim., 36 (1998), pp. 488–541.

[19] S. Smith, *Reactive scheduling systems*, in Intelligent Scheduling Systems, D. Brown and W. Schering, eds., Kluwer Academic Publishers, Boston, 1995, pp. 155–192.