

Multi-Robot Mixing Using Braids

Yancy Diaz-Mercado and Magnus Egerstedt

Abstract—This paper presents a method for automatically achieving multi-robot mixing in the sense that the robots follow predefined paths in a somewhat loose sense while ensuring that their actual movements are rich enough. In particular, we focus on the mixing problem, where the robots have to interweave their movements, for example to ensure sufficiently rich pairwise interactions or to cover an area along the path. By formally specifying mixing levels through strings over the Braid Group, the resulting hybrid system can execute a geometric interpretation of these strings, where the level of mixing is dictated by the string length. The feasibility of the proposed approach is illustrated on a particular class of multi-robot systems that cooperatively have to achieve the desired mixing levels.

I. INTRODUCTION

In a number of applications, from robotic search and coverage to convoy protection, the overall objectives can be stated in terms of making a team of robots follow a physical path, such as a road or the movements of a ground convoy, while ensuring that particular search patterns are executed [1], [2], [3]. These patterns should be selected in such a way that certain, secondary geometric objectives are met, including ensuring that an area along the path is covered, that multiple views of the same objects are achieved, that an aerial vehicle is always on top of the convoy, or that sufficiently many vehicle-to-vehicle interactions take place for the purpose of information sharing [4], [5]. In this paper, we collect all of these different secondary objectives under one unified banner, namely *multi-robot mixing*. In particular, we specify a desired level of mixing and then proceed to generating the actual, cooperative movements that realize these mixing levels.

A number of different types of mixing patterns have been previously proposed, corresponding to different types of objectives. When performing search or coverage tasks, line-sweeps and search spirals have been shown to be optimal, given assumptions on the search area and the distribution of objects to be found [1], [2], [6]. If, on the other hand, the objective is to get multiple views of the same objects, e.g., for the purpose of obtaining effective, visual cues, logarithmic spirals have been proposed [7], [8]. When multiple aerial vehicles are to cover ground convoys, the aerial vehicles' motions have been weaved together, while ensuring that at

least one vehicle is sufficiently close to the convoy [9], [10], [11], [4], which both facilitates information sharing and geometric coverage.

In this paper, we specify the mixing patterns through elements in the so-called braid group [12], [13], where each element corresponds to a particular mixing pattern, i.e., we do not focus on a particular pattern *per se*, but rather on the problem of being able to execute a whole class of patterns. The result from such a construction is a hybrid system driven by symbolic inputs [14], i.e., the braids, that must be mapped onto actual paths that both obtain the mixing level specified through the braid, and remain safe in the sense that collisions are avoided.

The outline of the paper is as follows: in Section II, the braid group is introduced as a way of specifying mixing levels and the corresponding symbolic braid objects are given a geometric interpretation in terms of planar robot paths. Controllers are then proposed so that these paths can be executed by a class of robots as described in Section III together with a bound on the highest achievable mixing level.

II. BRAIDS AND ROBOTICS

At the heart of the mixing problem is to be able to produce collision-free trajectories for the individual robots that somehow are rich enough to facilitate enough inter-robot interactions, area coverage, or utilization of potentially heterogeneous capabilities, e.g., sensing capabilities. In this paper, we aim at formalizing this mixing notion using a symbolic approach, where the individual symbols correspond to “mixing strategies” that allow us to describe the mixing task at a high level of abstraction without having to take the actual dynamics of the individual agents into account.

We approach this using ideas from the Braid Group. In particular, we consider only the planar mixing problem where the robots are confined to a plane, e.g., ground robots, marine robots operating at (or near) the surface, or aerial vehicles flying at a constant altitude.

A. Planar Braids

Consider two agents on a square, initially located at the two left vertices of the square as in Fig. 1. The agents' task is to move to the two right vertices of the square. There are two ways in which these target vertices can be assigned. The first is to simply let the robots move along a straight line while the second is to have them cross paths and move to vertices diagonally across from their initial placement. If the robots are to not collide with each other, the crossing paths can be negotiated by letting agent 1 reach the path intersection point first, or by letting agent 2 reach it first. In the braid group,

This work was supported by a grant from the U.S. Air Force Office for Sponsored Research.

Y. Diaz-Mercado is with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, 777 Atlantic Drive NW, Atlanta, GA 30332-0250, USA yancy.diaz@gatech.edu

M. Egerstedt is with Faculty of Electrical and Computer Engineering, Georgia Institute of Technology, 777 Atlantic Drive NW, Atlanta, GA 30332-0250, USA magnus@gatech.edu

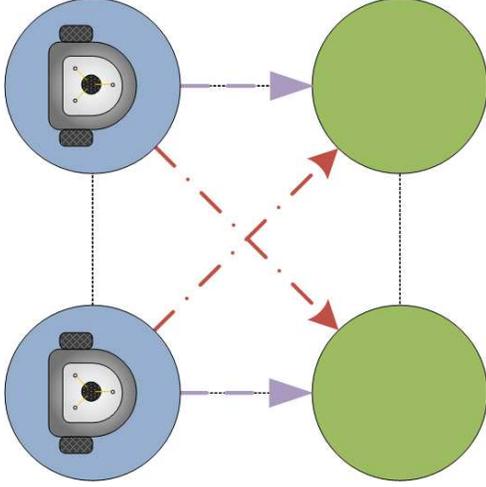


Fig. 1. Two robots, originally at one side of a square, must map to the other.

these two options correspond to different “braids”, and we have thus identified three planar braids for two agents, as shown in Fig. 2. Let us denote these three braids, $\sigma_1, \sigma_2, \sigma_3$.

Now, given these three braids, we can concatenate them together to form other braids, as seen in Fig. 3. The left braid is given by $\sigma_3 \cdot \sigma_3$ and the right braid is $\sigma_2 \cdot \sigma_3$. In the braid group, what really matters is not the geometric layout of the paths, but how the paths wrap-around each other. As can be seen, if we were to “pull” the right corners in $\sigma_3 \cdot \sigma_3$ we get a “tangle” in the middle, while a “stretched-out” $\sigma_2 \cdot \sigma_3$ is simply σ_1 . If we let σ_1 be the identity braid, σ_2 and σ_3 are each others’ inverses in the sense that

$$\sigma_2 \cdot \sigma_3 = \sigma_3 \cdot \sigma_2 = \sigma_1.$$

In fact, every braid has an inverse and, as such, the set of braids (together with the concatenation operation) is indeed a group. And, as $\sigma_2^{-1} = \sigma_3$ (and $\sigma_3^{-1} = \sigma_2$), σ_1 and σ_2 (or σ_1 and σ_3 for that matter) are the so-called generator braids for this group in that all planar braids can be written as concatenations of these two braids and their inverses.

This notion of generator braids can be extended to the case when there are N rather than two agents, with the only difference being that we now have more than two generators. If we let Σ_N be the set of all planar generator braids over N agents, this set will serve as the alphabet over which braid strings (themselves braids) are produced from, and we let Σ_N^M denote the set of all braids of length M over N agents.

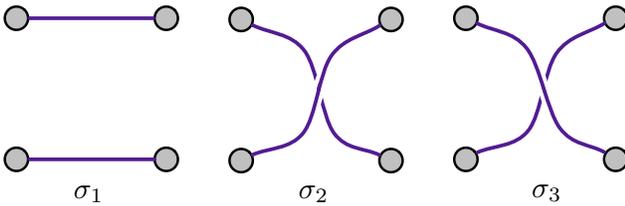


Fig. 2. Three braids for two planar agents.

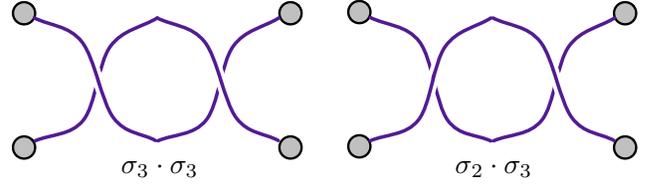


Fig. 3. Two concatenated braids $\sigma_3 \cdot \sigma_3$ and $\sigma_2 \cdot \sigma_3$. The latter of these two braids is the same as the identity braid σ_1 .

B. A Geometric Interpretation

Although the braid group is not concerned with the actual geometry of the braid strands, we will connect geometric paths with the different braids. First of all, we assume that the braid is geometrically located in a rectangular area of height h and length ℓ no matter how long the braid string is. Using the particular 2-agent braids discussed in the previous paragraphs, we assume that the two agents are initially located at the points $(0, 0)$ and $(0, h)$, while the final locations are at $(\ell, 0)$ and (ℓ, h) . If the total braid results from the use of one single generator braid σ_1 , or σ_2 then no additional points are needed. However, if the braid has length 2, then we also need to introduce intermediary “half-way” points $(\ell/2, 0)$ and $(\ell/2, h)$. As such, we let $\mathcal{S}_2^p = \{(p\ell, 0), (p\ell, h)\}$, where the subscript 2 denotes the two-agent case, and $p \in [0, 1]$.

Using this notation, we can refer back to Fig. 1 and say that each of the two generator braids correspond to an assignment, i.e., a bijective map, between \mathcal{S}_2^0 and \mathcal{S}_2^1 , and we use the following notation to denote this fact

$$\sigma_i : \mathcal{S}_2^0 \rightarrow_b \mathcal{S}_2^1, \quad i = 1, 2,$$

where \rightarrow_b denotes “bijection”. Note that this is a slight abuse of notation in that σ_i now denotes both an element in the braid group as well as a map – this distinction, however, should be entirely clear from the context.

If we generalize this to $N \geq 2$ agents and let σ denote a string of generators of length $M \geq 2$, i.e. $\sigma \in \Sigma_N^M$, we will use the notation

$$\sigma(k) : \mathcal{S}_N^{(k-1)/M} \rightarrow_b \mathcal{S}_N^{k/M}, \quad k = 1, \dots, M,$$

where $\sigma(k)$ is the k^{th} generator in the string σ and

$$\mathcal{S}_N^p = \{(p\ell, 0), (p\ell, h/(N-1)), (p\ell, 2h/(N-1)), \dots, (p\ell, h)\},$$

as shown for the three-agent case in Fig. 4.

We moreover will use the notation $\xi(i, j)$ to denote what point agent j should go to at step i , $i = 0, \dots, M$, where we use the convention that $\xi(0, j) = (0, (j-1)h/(N-1))$, $j = 1, 2, \dots, N$ denotes agent j ’s initial position. In other words,

$$\begin{aligned} \xi(1, j) &= \sigma(1)[\xi(0, j)], \\ \xi(2, j) &= \sigma(2)[\xi(1, j)] = \sigma(2) \circ \sigma(1)[\xi(0, j)], \end{aligned}$$

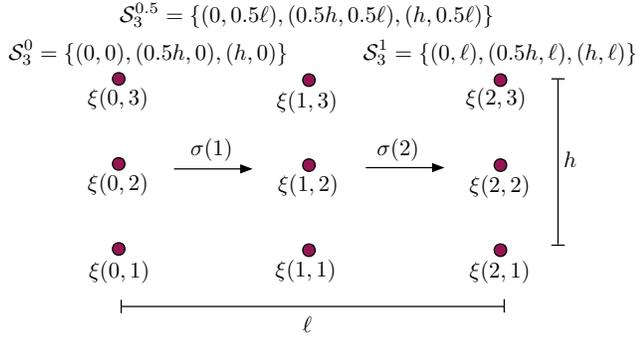


Fig. 4. The geometric interpretation of a braid string for the three-agent case.

or more generally,

$$\begin{aligned}\xi(i, j) &= \sigma(i)[\xi(i-1, j)] \\ &= \sigma(i) \circ \sigma(i-1) \circ \dots \circ \sigma(1)[\xi(0, j)],\end{aligned}$$

where we use the $[\cdot]$ notation to denote the argument to $\sigma(i)$ and \circ to denote composition. This construction is also illustrated in Fig. 4 for the three-agent case.

The geometric interpretation we will make of the planar braids is that the mobile agents that are to execute them must traverse through these points. They must moreover do so in an orderly and safe manner, which will be the topic of the next section.

III. MULTI-ROBOT MIXING

A. Motivational Application

Consider the convoy protection surveillance coverage problem. The objective is to have a group of planar aerial vehicles, i.e. aerial vehicles flying at a constant altitude, surveil the area around a path that the ground convoy will be traversing. The agents have a limited footprint in which they can scan for threats. One approach at increasing the probability of detecting threats in the area is to have the agents perform braids, covering as much of the area as possible by swapping positions as they move loosely along the path. This scenario is illustrated in Fig. 5.

There are multiple considerations that naturally arise, such as the existence of a pattern or patterns of movement that ensure maximum coverage. One also expects these patterns to be safe enough for multiple agents to occupy the space. In this section we make several definitions in order to address some of these considerations and conclude by providing a hybrid control scheme that achieves these secondary objectives.

B. Executing Braids

Given a collection of N agents with dynamics

$$\dot{x}_i = f(x_i, u_i),$$

and planar output

$$y_i = h(x_i) \in \mathbb{R}^2, \quad i = 1, \dots, N,$$

then it is of interest to have these agents execute a braid $\sigma \in \Sigma_N^M$. We now define what it means for this braid to be executed.

Given an input braid string σ , what each individual agent should do is “hit” the intermediary braid points $\xi(i, j)$ at specified time instances. We let T denote the time it should take for the entire string to be executed. As such, the first condition for a multi-agent motion to be feasible with respect to the braid is the following:

Definition 3.1 (Braid-Point Feasibility):

A multi-robot trajectory is *braid-point feasible* if

$$y_j(iT/M) = \xi(i, j), \quad i = 0, \dots, M, \quad j = 1, \dots, N.$$

◇

On top of braid-point feasibility, we also insist on the robots not colliding during the maneuvers. To a certain degree, this condition is what restricts the level of mixing that is possible, i.e., since the braid is constrained in a rectangle of fixed height and width, what length strings the multi-robot system can execute while maintaining a desired level of safety separation.

Definition 3.2 (Collision-Free):

A multi-robot trajectory is *collision-free* if

$$\|y_i(t) - y_j(t)\| \geq \Delta, \quad \forall i \neq j, \quad t \in [0, T],$$

where $\Delta > 0$ is the desired level of safety separation.

◇

We are missing a notion to describe what the multi-robot mixing problem is, that is what constitutes a braid controller.

Definition 3.3 (Braid Controller):

A multi-robot controller is a *braid controller* if the resulting trajectories are both braid-point feasible and collision-free, for all collision-free initial conditions such that

$$y_i(0) = \xi(0, i), \quad i = 1, \dots, N.$$

◇

As a final notion, we are interested in how much mixing a particular system can support.

Definition 3.4 (Mixing Limit):

The *mixing limit* M^* is the largest integer M such that there

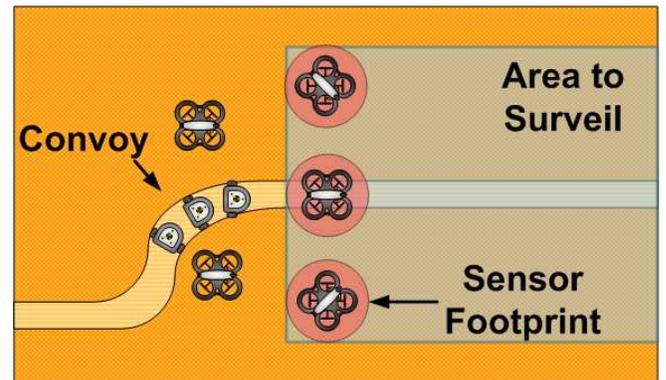


Fig. 5. A subset of the agents protecting the convoy are tasked with surveilling an area for threats.

exists a braid controller for every string in Σ_N^M .

◇

The mixing limit is in general quite hard to compute, it needs to consider every permutation of strings of varying lengths up to some number, the geometry assigned to each string and is dependent on the kinematical response of the multi-robot system. In the next section, we give a lower bound on M^* for a certain class of systems. In particular, we define a hybrid braid controller and then compute how large M can be, for this particular choice of controller.

C. A Mixing Bound

As the level of mixing one can achieve in a multi-robot team depends on the dynamics of the robots as well as the actual paths that are being taken, we here discuss a particular such choice. In order to be able to actually formulate mixing bounds, we focus initially on a particularly simple class of systems, namely planar integrators moving along piecewise straight lines. In other words, we let

$$\dot{x}_i = u_i, \quad y_i = x_i.$$

Assume that we are to execute a braid string of length M in time T . To design a braid controller, it is sufficient to focus on a single braid (let us say the j^{th} braid) in the string, which thus has to be executed in time T/M . The direction that robot i should be traveling in, in order to ensure braid-point feasibility is given by

$$\theta_{i,j} = \arctan \left(\frac{\xi(j,i)_2 - \xi(j-1,i)_2}{\xi(j,i)_1 - \xi(j-1,i)_1} \right),$$

where the subscripts on the ξ -variables note the first and second components, respectively. For the sake of notational convenience, we let $\delta_{i,j} = (\cos(\theta_{i,j}), \sin(\theta_{i,j}))^T$ denote the unit vector pointing in the $\theta_{i,j}$ direction. We moreover assume that the robots are not allowed to drive backwards and, at the same time, there is an upper bound on the velocity of the robots, i.e., $\|u_i\| \in [0, v_{\max}]$, $i = 1, \dots, N$.

The control strategy we will employ is a STOP-GO-STOP strategy, where the robots stay still, then move at some velocity (bounded by the maximum velocity), and then remain still again. This is just one of a number of possible such choices, but this choice will simplify our search for a braid controller. The key idea is to organize the agents according to how far they have to travel during the j^{th} braid, and we let $s_j : \{1, \dots, N\} \rightarrow \{1, \dots, M\}$ denote this ordering, i.e.,

$$s_j(i) < s_j(k) \Rightarrow \|\xi(j,i) - \xi(j-1,i)\| \geq \|\xi(j,k) - \xi(j-1,k)\|,$$

where we break ties arbitrarily such that s_j is a bijection. The idea is that the further you have to go, the shorter you should be in the initial STOP mode, with agent $s_j^{-1}(1)$ spending no time in the initial stop mode.

Now, as all we can really hope for is a lower bound for the mixing limit, we will conduct the analysis along the horizontal direction, and the idea is to let agent $s_j^{-1}(2)$

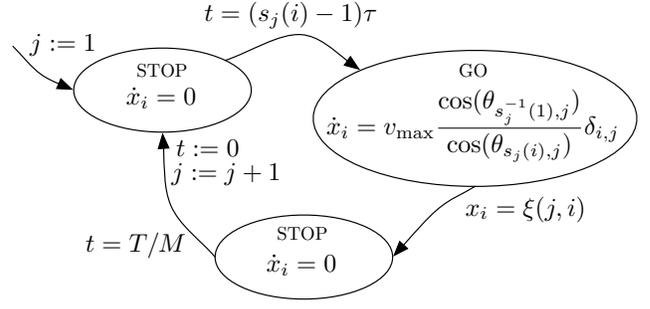


Fig. 6. A STOP-GO-STOP braid controller.

enter the GO mode once agent $s_j^{-1}(1)$ has traveled Δ along the horizontal direction, at maximum velocity. This happens when $\Delta / (\cos(\theta_{s_j^{-1}(1),j}) v_{\max})$ time has elapsed. But, to not make this timing dependent on the direction of travel, we note that

$$\cos(\theta_{i,j}) \leq \frac{\ell/M}{\sqrt{\ell^2/M^2 + h^2}} = \cos(\theta^*), \quad \forall i, j,$$

and as such, we let amount of time that the second agent has to wait before entering the GO mode be given by

$$\tau_2 = \frac{\Delta}{v_{\max} \cos(\theta^*)},$$

and we drop the subscript from τ_2 to refer to this waiting time simply as τ . But now, to ensure that this new agent does not overtake the first agent, its speed in the horizontal direction has to be the same as the first agent's, i.e., the speed needs to be

$$v_{\max} \frac{\cos(\theta_{s_j^{-1}(1),j})}{\cos(\theta_{s_j^{-1}(2),j})}.$$

The third agent to leave the STOP mode should thus wait twice as long as the second agent, i.e., $\tau_3 = 2\tau$. Following this pattern, the k^{th} agent to leave the STOP mode should wait $\tau_k = (k-1)\tau$, if the agent moves in the GO mode with the speed $v_{\max} \cos(\theta_{s_j^{-1}(1),j}) / \cos(\theta_{k,j})$. As such, we are ensuring that all agents are always at least Δ separated if agent i waits a total of $(s_j(i) - 1)\tau$ in the initial STOP mode before entering the GO mode, at which point it is moving at speed $v_{\max} \cos(\theta_{s_j^{-1}(1),j}) / \cos(\theta_{s_j(i),j})$, until the next braid point is reached. The hybrid automaton describing this strategy is given in Fig. 6, together with a simulation of five agents executing this strategy over a braid of length 4 in Fig. 7.

Before we can state the mixing bound condition, we first need to assume that the braid points themselves are sufficiently separate from each other, i.e., that

$$\|\xi(j,i) - \xi(j,k)\| \geq \Delta, \quad \forall i, j, k.$$

Theorem 3.1: The STOP-GO-STOP controller in Fig. 6 is a braid controller if the braid points themselves are sufficiently separated and

$$\cos(\theta^*) v_{\max} (T/M - (N-1)\tau) \geq \sqrt{\ell^2/M^2 + h^2}.$$

Proof: We have already established that the STOP-GO-STOP controller ensures that the agents are never within Δ of each other by virtue of the fact that they have to wait until they are indeed at least that far apart. As such, the trajectories are collision-free. What remains to show is that they are also braid-point feasible.

We will consider the agent that has to wait the longest before it can move, i.e., this agent has to wait a total of $(N - 1)\tau$, and it thus has $T/M - (N - 1)\tau$ left to reach the next braid point. In other words, we need that the distance traveled in that amount of time at the speed $v_{\max} \cos(\theta_{s_j^{-1}(1),j}) / \cos(\theta_{s_j(i),j})$ is greater than the distance required. But, we note that

$$v_{\max} \cos(\theta_{s_j^{-1}(1),j}) / \cos(\theta_{s_j(i),j}) \geq v_{\max} \cos(\theta^*)$$

and, as we are only looking for a bound, we assume that we use this lower speed and that the distance required to travel is the largest distance possible (which it really is not). In other words, we need

$$\cos(\theta^*) v_{\max} (T/M - (N - 1)\tau) \geq \sqrt{\ell^2/M^2 + h^2},$$

and the proof follows. ■

Note that the theorem also (implicitly) provides a mixing bound for how many agents you can use given a mixing level of M (and vice versa), as shown in Fig. 8. In that figure, the following parameters were used: $v_{\max} = 5$, $T = 20$, $\ell = 5$, $h = 10$, $\Delta = 0.2$.

IV. CONCLUSION

This paper introduces the notion of multi-robot mixing as a collective term for the types of mobility patterns found in search and coverage applications, convoy protection, and multi-robot interactions for the purpose of information sharing. The idea is to have a secondary objective be coupled to the main task of the robot team, given on terms of how much they need to “mix”. We represent mixing as formal elements in a braid group, and the level of mixing corresponds to the length of the string used to produce the braid from the generator braids. A hybrid formalism is used to connect the

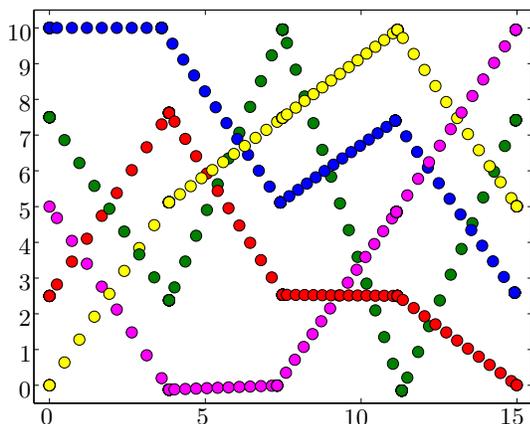


Fig. 7. Five robots are executing a string of length four using the strategy in Fig. 6.

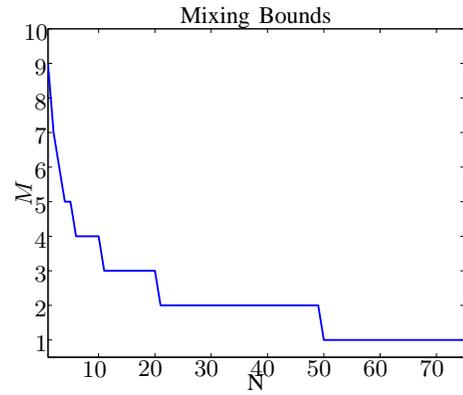


Fig. 8. A lower bound on the mixing limit.

braid strings to the actual movements of the robots and under a hybrid STOP-GO-STOP strategy, a lower-bound is given on the maximum amount of mixing supported by a team of robots.

REFERENCES

- [1] R. Hall, “Efficient spiral search in bounded spaces,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 4, no. 2, pp. 208–215, March 1982.
- [2] W. Huang, “Optimal line-sweep-based decompositions for coverage algorithms,” in *IEEE International Conference on Robotics and Automation*, 2001.
- [3] R. Hashemi, L. Jin, G. Anderson, E. Wilson, and M. Clark, “A comparison of search patterns for cooperative robots operating in remote environment,” in *International Conference on Information Technology: Coding and Computing*, 2001.
- [4] J. Kim, F. Zhang, and M. Egerstedt, “Simultaneous cooperative exploration and networking based on Voronoi diagrams,” in *IFAC Workshop on Networked Systems*, October 2009.
- [5] B. Tovar, L. Freda, and S. M. LaValle, “Using a robot to learn geometric information from permutations of landmarks,” in *Contemporary Mathematics*. American Mathematical Society, 2007, vol. 438, pp. 33–45.
- [6] Y. Mei, Y.-H. Lu, Y. Hu, and C. Lee, “Energy-efficient motion planning for mobile robots,” in *Proceedings of the 2004 IEEE International Conference on Robotics and Automation. ICRA '04.*, vol. 5, May 2004, pp. 4344 – 4349 Vol.5.
- [7] S. Bhattacharya, R. Murrieta-Cid, and S. Hutchinson, “Optimal paths for landmark-based navigation by differential drive vehicles with field-of-view constraints,” *IEEE Trans. on Robotics*, vol. 23, no. 1, pp. 47–59, feb 2007.
- [8] E. Tunstel, J. Dolan, T. Fong, and D. Schreckenghost, “Mobile robotic surveying performance for planetary surface site characterization,” in *ACM Workshop on Performance Metrics for Intelligent Systems*, 2008.
- [9] X. Ding, A. Rahmani, and M. Egerstedt, “Multi-UAV convoy protection: An optimal approach to path planning and coordination,” *IEEE Trans. on Robotics*, vol. 26, no. 2, pp. 256–268, 2010.
- [10] D. J. Klein, C. Matlack, and K. A. Morgansen, “Cooperative target tracking using oscillator models in 3D,” in *American Control Conference*, July 2007.
- [11] J. Lee, R. Huang, A. Vaughn, X. Xiao, J. K. Hedrick, M. Zennaro, and R. Sengupta, “Strategies of path-planning for a UAV to track a ground vehicle,” in *The Second Annual Symposium on Autonomous Intelligent Networks and Systems*, 2003.
- [12] E. Artin, “Theory of braids,” *Annals of Mathematics*, vol. 48, no. 1, pp. 101–126, 1947.
- [13] J. Birman, *Braids, links, and mapping class groups*. Princeton University Press, 1974.
- [14] C. Belta, A. Bicchi, M. Egerstedt, E. Frazzoli, E. Klavins, and G. J. Pappas, “Symbolic planning and control of robot motion,” *IEEE Robotics and Automation Magazine*, vol. 14, no. 1, pp. 61–70, March 2007.