3-21-2016

Coordinated Robot Navigation via Hierarchical Clustering

Omur Arslan
University of Pennsylvania, omur@seas.upenn.edu

Dan P. Guralnik
University of Pennsylvania, guralnik@seas.upenn.edu

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

Recommended Citation

Omur Arslan, Dan P. Guralnik, and Daniel E. Koditschek, "Coordinated Robot Navigation via Hierarchical Clustering", *IEEE Transactions on Robotics* 32(2), 352-371. March 2016. http://dx.doi.org/10.1109/TRO.2016.2524018

Coordinated Robot Navigation via Hierarchical Clustering

Abstract

We introduce the use of hierarchical clustering for relaxed, deterministic coordination and control of multiple robots. Traditionally an unsupervised learning method, hierarchical clustering offers a formalism for identifying and representing spatially cohesive and segregated robot groups at different resolutions by relating the continuous space of configurations to the combinatorial space of trees. We formalize and exploit this relation, developing computationally effective reactive algorithms for navigating through the combinatorial space in concert with geometric realizations for a particular choice of hierarchical clustering method. These constructions yield computationally effective vector field planners for both hierarchically invariant as well as transitional navigation in the configuration space. We apply these methods to the centralized coordination and control of n perfectly sensed and actuated Euclidean spheres in a d-dimensional ambient space (for arbitrary n and d). Given a desired configuration supporting a desired hierarchy, we construct a hybrid controller which is quadratic in n and algebraic in d and prove that its execution brings all but a measure zero set of initial configurations to the desired goal with the guarantee of no collisions along the way.

Keywords

Multirobot systems, navigation functions, formation control, swarm robots, configuration space, coordinated motion planning, hierarchical clustering, cohesion, segregation

Disciplines

Controls and Control Theory | Electrical and Computer Engineering | Engineering | Robotics | Systems Engineering

Coordinated Robot Navigation via Hierarchical Clustering

Omur Arslan, *Student Member, IEEE,* Dan P. Guralnik and Daniel E. Koditschek, *Fellow, IEEE*

***Abstract*—We introduce the use of hierarchical clustering for relaxed, deterministic coordination and control of multiple robots. Traditionally an unsupervised learning method, hierarchical clustering offers a formalism for identifying and representing spatially cohesive and segregated robot groups at different resolutions by relating the continuous space of configurations to the combinatorial space of trees. We formalize and exploit this relation, developing computationally effective reactive algorithms for navigating through the combinatorial space in concert with geometric realizations for a particular choice of hierarchical clustering method. These constructions yield computationally effective vector field planners for both hierarchically invariant as well as transitional navigation in the configuration space. We apply these methods to the centralized coordination and control of n perfectly sensed and actuated Euclidean spheres in a d-dimensional ambient space (for arbitrary n and d). Given a desired configuration supporting a desired hierarchy, we construct a hybrid controller which is quadratic in n and algebraic in d and prove that its execution brings all but a measure zero set of initial configurations to the desired goal with the guarantee of no collisions along the way.**

***Index Terms*—multirobot systems, navigation functions, formation control, swarm robots, configuration space, coordinated motion planning, hierarchical clustering, cohesion, segregation.**

I. Introduction

Cooperative, coordinated action and sensing can promote efficiency, robustness, and flexibility in achieving complex tasks such as search and rescue, area exploration, surveillance and reconnaissance, and warehouse management [2]. Despite significant progress in the analysis of how local rules can yield such global spatiotemporal patterns [3]–[5], there has been strikingly less work on their specification. With few exceptions, the engineering literature on multirobot systems relies on task representations expressed in terms of rigidly imposed configurations — either by absolutely targeted positions, or relative distances — missing the intuitively substantial benefit of ignoring fine details of individual positioning, to focus control effort instead on the presumably far coarser properties of the collective pattern that matter. We seek a more relaxed means of specification that is sensitive to spatial distribution at multiple scales (as influencing the intensity of interactions among individuals and with their environment [6]) and the identities of neighbors (as determining the capabilities of heterogeneous teams [7]) while affording, nevertheless, a well-formed deterministic characterization of pattern.

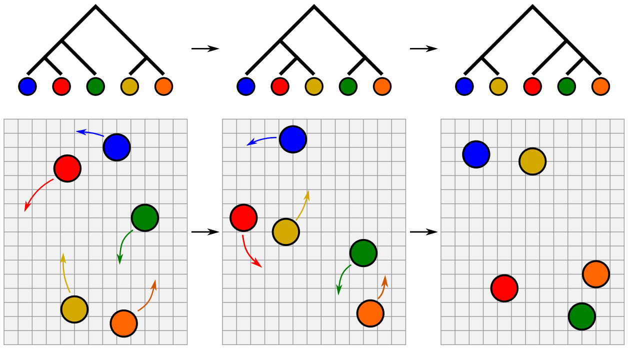


Fig. 1. Moving from one spatial distribution to another is generally carried through rearrangements of robot groups (clusters) at different resolution corresponding to transitions between different cluster structures (hierarchies).

We are led to the notion of hierarchical clustering. We reinterpret this classical method for unsupervised learning [8] as a formalism for the specification and reactive implementation of collective mobility tasks expressed with respect to successively refined partitions of the agent set in a manner depicted in Fig. 1. There, we display three different configurations of five planar disks whose relative positions are specified by three distinct trees that represent differently nested clusters of relative proximity. The first configuration exhibits three distinct clusters at a resolution in the neighborhood of 2 units of distance: the red and the blue disks; the yellow and the orange disks; and the solitary green disk. At a coarser resolution, in the neighborhood of 4 units of distance, the green disk has merged into the subgroup including the red and the blue disks to comprise one of only two clusters discernible at this scale, the other formed by the orange and the yellow disks. It is intuitively clear that this hierarchical arrangement of subgroupings will persist under significant variations in the position of each individual disk. It is similarly clear that the second and third configurations (and significant variations in the positions of the individual disks of both) support the very differently nested clusters represented by the second and third trees, respectively. In this paper, we introduce a provably correct and computationally effective machinery for specifying, controlling invariantly to, and passing between such hierarchical clusterings at will.

As an illustration of its utility, we use this formalism to solve a specific instance of the reactive motion planning

The authors are with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104. E-mail: {omur, guralnik, kod}@seas.upenn.edu. This work was supported in part by AFOSR under the CHASE MURI FA9550-10-1-0567 and in part by ONR under the HUNT MURI N00014070829.

A preliminary version of this paper is presented in the conference paper [1] for point particles and a certain choice of hierarchical clustering. In this paper, we propose a general hierarchical navigation framework for a broad class of clustering methods and disk-shaped robots.

TABLE I
CONSTITUENT PROBLEMS OF HIERARCHICAL ROBOT NAVIGATION

Problem	Solution	Theorem	Description
1	Table IV	4	Hierarchy invariant vector field planner
2	Table V	5	Reactive navigation across hierarchies
3	Eqn.(33)	6	Cross-hierarchy geometric realization

problem suggesting how the new "relaxed" hierarchy-sensitive layer of control can be merged with a task entailing a traditional rigidly specified goal pattern. Namely, for a collection of n disk robots in $\mathbb{R}^d$ we presume that a target hierarchy has been specified along with a goal configuration that supports it, and that the robot group is controlled by a centralized source of perfect, instantaneous information about each agent's position that can command exact instantaneous velocities for each disk. We present an algorithm resulting in a purely reactive hybrid dynamical system [9] guaranteed to bring the disk robots to both the hierarchical pattern as well as the rigidly specified instance from (almost) arbitrary initial conditions with no collisions of the disks along the way. Stated formally in Table III, the correctness of this algorithm is guaranteed by Theorem 1 whose proof appeals to the resolution of various constituent problems summarized in Table I. The construction is computationally effective: the number of discrete transitions grows in the worst case with the square of the number of robots, n; each successive discrete transition can be computed reactively (i.e., as a function of the present configuration) in time that grows linearly with the group size; and the formulae that define each successive vector field and guard condition are rational functions (defined by quotients of polynomials over the ambient space of degree less than 3) entailing terms whose number grows quadratically with the number of robots. In summary, the main contributions of the present paper are:

- a novel abstraction for ensemble task encoding and control in terms of hierarchical clustering, yielding precise yet flexible organizational specifications at selectively multiple resolutions,
- a provably correct generic hierarchical navigation framework for collision free feedback motion planning for multirobot systems,
- a computationally efficient instantiation of the hierarchical navigation framework for coordinated control of an arbitrary number of disk-shaped robots operating in an ambient space (of dimension $d \geq 2$).

On a more conceptual level, we believe this paper breaks new ground by introducing a topologically nontrivial symbolic abstraction that reduces the complexity of high level planning in the abstract symbol space [10] while nevertheless keeping the associated physical problems within the scope of reactive (real time) planning methods. In particular, our hierarchical decomposition is not cellular — i.e., it is not the case that a stratum of clusterings is contractible [11]. Rather, each component has a known homotopy type. That information enables the construction of a vector field to handle continuous motions whose flow is designed to respect it, as must be the case if its basin (the physical initial conditions it can handle correctly) is to fill out the entire component.

This paper is organized as follows. We review in the next section the relevant background literature: first on reactive multirobot motion planning to relate the difficulty and importance of our sample problem to the state of the art in this field; next on the role of hierarchy in configuration spaces as explored both in biology and engineering. Because the notion of hierarchical clustering is a new abstraction for motion planning we devote Section III to a presentation of the key background technical ideas: first we review the relevant topological properties of configuration spaces; next the relevant topological properties of tree spaces; and, finally, prior work establishing properties of certain functions and relations between them. Because we feel that the specific motion planning problem we pose and solve represents a mere illustration of the larger value of this abstraction for multirobot systems we devote Section IV to a presentation of some of the more generic tools from which our particular construction is built: first we introduce the notion of hierarchy invariant navigation; next we discuss the combinatorial problem of hierarchy rearrangement as a graph navigation problem; and finally we interpret a subgraph of that combinatorial space as a "prepares" graph [12] for the hierarchy-invariant cover of configuration space. In Section V we pose and solve the specific motion planning problem using the concepts introduced in Section III and the tools introduced in Section IV. Section VI offers some numerical studies of the resulting algorithm. We conclude in Section VII with a summary of the major technical results that yield the specific contribution followed by some speculative remarks bearing on the likelihood that recent extensions of these ideas presently in progress [13] might afford a distributed reformulation, thus addressing the first (and better explored) remarkable biological inspiration for multirobot systems.

II. RELATED LITERATURE

A. Multirobot Motion Planning

1) Complexity: The intrinsic complexity of multibody configurations impedes computationally effective generalizations of single-robot motion planners [14], [15]. Coordinated motion planning of thick bodies in a compact space is computationally hard. For example, moving planar rectangular objects within a rectangular box is PSPACE-hard [16] and motion planning for finite planar disks in a polygonal environment is strongly NP-hard [17]. Even determining when and how the configuration space of noncolliding spheres in a unit box is connected entails an encounter with the ancient sphere packing problem [18]. As a result, although they ensure certain optimality properties and handle complex environments, most available multirobot path planning algorithms suffer from having at least exponential computation time with the number of robots limiting their applicability to problems entailing a small number of robots in real-time settings [19]. Within the domain of reactive or vector field motion planning, which is the main focus of this paper, it has proven deceptively hard to determine exactly this line of intractability. Consequently, this intrinsic complexity for coordinated vector field planners is generally mitigated by either assuming objects move in an unbounded (or sufficiently large) space [20], [21], as we do in Section V, or simply assuming conditions sufficient to guarantee connectivity between initial and goal configurations [22], [23]. On the other hand, more relaxed versions entailing (perhaps partially) homogeneous[1] (unlabeled) specifications for interchangeable individuals have yielded computationally efficient planners in the recent literature [24]–[27], and we suspect that the cluster hierarchy abstraction may be usefully applicable to such partially labeled settings.

2) Reactive Multirobot Motion Planning: Since the problem of reactively navigating groups of disks was first introduced to robotics [29], [30], most research into vector field planners has embraced the navigation function paradigm [31]. A recent review of this two decade old literature is provided in [20], where a combination of intuitive and analytical results yields a nonsmooth centralized planner for achieving goal configurations specified up to rigid transformation. As noted in [20], the multirobot generalization of a single-agent navigation function is challenged by the violation of certain assumptions inherited from the original formulation [31]. One such assumption is that obstacles are "isolated" (i.e. nonintersecting). In the multirobot case, every robot encounters others as mobile obstacles, and any collision between more than two robots breaks down the isolated obstacle assumption [20]. In some approaches, the departure from isolated interaction has been addressed by encoding all possible collision scenarios, yielding controllers with terms growing super-exponentially in the number of robots, even when the workspace is not compact [21]. In contrast, our recourse to the hierarchical representation of configurations affords a computational burden growing merely quadratically in the number of agents. In [22], the problem is circumvented by allowing critical points on the boundary (with no damage to the obstacle avoidance and convergence guarantees), but, as mentioned above, very conservative assumptions about the degree of separation between agents at the goal state are required. In contrast, our recourse to hierarchy allows us to handle arbitrary (non-intersecting) goal configurations, albeit our reliance upon the homotopy type of the underlying space presently precludes the consideration of a compact configuration space as formally allowed in [22].

Another limitation of navigation function approaches is the requirement of proper parameter tuning to eliminate local minima. Some effort has been given to automatic adaptation of this parameter [23], and, in principle, the original results of [31] guarantee that any monotone increasing scheme must eventually resolve the issue of local minima, however, this is numerically unfavorable (the Hessian of the resulting field becomes stiffer) and incurs substantial performance costs (transients must slow as the tuning parameter increases).[2] In contrast, our recourse to hierarchy removes the need for any comparable tuning parameter.

Many of the concepts and some of the technical constructions we develop here were presented in preliminary form in the conference paper [1], building on the initial results of the conference paper [33]. This presentation gives a unified view of the detailed results (with some tutorial background) and contributes a major new extension by generalizing the construction of [1] from point particles to thickened disks of non-zero radius (necessitating a more involved version of the hierarchy invariant fields in Section V-B).

B. The Use of Hierarchies as Organizational Models

1) Hierarchy in Configuration Space: That a hierarchy of proximities might play a key role in computationally efficient coordinated motion planning had already been hinted at in early work on this problem [34]–[36]. Partial hierarchies that limit the combinatorial growth of complexity have been explicitly applied algorithmically to organize and simplify the systematic enumeration of cluster adjacencies in the configuration space [37]. Moreover, hierarchical discrete abstraction methods are successfully applied for scalable steering of a large number of robots as a group all together by controlling the group shape [38], and also find applications for congestion avoidance in swarm navigation [39]. While the utility of hierarchies and expressions for manipulating them are by no means new to this problem domain, we believe that the explicit formal connection [40] we exploit between the topology of configuration space [41] and the topology of tree space [42] through the hierarchical clustering relation [8] is entirely new.

2) Hierarchy in Biology and Engineering: Biology offers spectacularly diverse examples of animal spatial organization ranging from self-sorting in cells [43], tissues and organs [44], [45], and groups of individuals [46]–[48] to more patterned teams [7], [49]–[51], all the way through strategic group formations in vertebrates [52], [53], mammals [54]–[57], and primates [58], [59] hypothesized to increase efficacy in foraging [49], [50], hunting [52], [54], [55], [58], logistics and construction [7], [51], predator avoidance [60], [61], and even to stabilize whole ecologies [62] — all consequent upon the collective ability to target, track, and transform geometrically structured patterns of mutual location in response to environmental stimulus. These formations are remarkable for at least two reasons. First, their global structure seems to arise from local signaling and response amongst proximal individuals coupled to specific physical environments [63], in a manner that might be posited as a paradigm for generalized emergent intelligence [64]. Second, these formations appear to resist familiar rigid prescriptions governing absolute or relative location, instead giving wide latitude for individual autonomy and detailed positioning (intuitively, a necessity for negotiating fraught, highly dynamic interactions such as arise in, say, hunting [54], [56]), while, nevertheless, supporting the underlying coarse, deterministic "deep structure" as a dynamical invariant. It is this second remarkable attribute of biological swarms that inspires the present paper.

This profusion of pattern formation in biology has inspired a commensurate interest in robotics, yielding a growing literature on group coordination behaviors [28], [65]–[67] motivated

[1] Following the literature we use the term "heterogeneity" to connote the robots' diversity in actuation, sensing, computation, communication and energy resources, which generally determines constraints on task assignment [2], [24], [28]. For example, each robot in a fully heterogeneous (uniquely labeled) group has a specific task (or target) whereas robots in a homogeneous (unlabeled) group are interchangeable. In this paper we consider fully heterogeneous robot groups since any method proposed for heterogeneous robots can be easily applied to (partially) homogeneous robots.

[2] It bears mention in passing that partial differential equations (e.g., harmonic potentials [32]) yield self-tuning navigation functions but these are intrinsically numerical constructions that forfeit the reactive nature of the closed form vector field planners under discussion here.

by the intuition that the heterogeneous action and sensing abilities of a group of robots might enable a comparably diverse range of complex tasks beyond the capabilities of a single individual. For example, group coordination via splitting and merging behaviours creates effective strategies for obstacle avoidance [68], congestion control [39], shepherding [69], area exploration [69], [70], and maintaining persistent and coherent groups while adapting to the environment [67]. In almost all of the robotics work in this area, formation tasks are given based upon rigid specifications taking either the form of explicit formation or relative distance graphs, with few exceptions including the "shape" abstraction of [38] or applications in unknown environments such as area coverage and exploration [71]. Alternatively, hierarchical clustering offers an interesting means of ensemble task encoding and control; and it seems likely that the ability to specify organizational structure in the precise but flexible terms that hierarchy permits will add a useful tool to the robot motion planner's toolkit.

TABLE II
PRINCIPAL SYMBOLS USED THROUGHOUT THIS PAPER

J, $\mathbf{r}$	Sets of labels and disk radii [III-A]
$\mathtt{Conf}\left(\mathbb{R}^d, J, \mathbf{r}\right)$	The conf. space of labelled, noncolliding disks (1)
$\mathcal{BT}_J$	The space of binary trees [III-B]
$\mathtt{HC}$	Hierarchical clustering [III-C]
$\mathtt{HC}_{\text{2-means}}$	Iterative 2-means clustering [V]
$\mathfrak{S}(\tau)$	The stratum of a tree, $\tau \in \mathcal{BT}_J$, (2)
$\mathtt{Portal}\,(\sigma, \tau)$	Portal configurations of a pair, (σ, τ), of trees (5)
$\mathtt{Port}_{\sigma,\tau}$	Portal map [IV-A3]
$\mathcal{A}_J = (\mathcal{BT}_J, \mathcal{E}_{\mathcal{A}})$	The adjacency graph of trees [III-D]
$\mathcal{N}_J = (\mathcal{BT}_J, \mathcal{E}_{\mathcal{N}})$	The NNI-graph of trees [III-D]

III. Hierarchical Abstraction

This section describes how we relate multirobot configurations to abstract cluster trees via hierarchical clustering methods and how we define connectivity in tree space.

A. Configuration Space

For convenience, we restrict our attention to Euclidean disks moving in a d-dimensional ambient space, but many concepts introduced here can be generalized to any metric space.

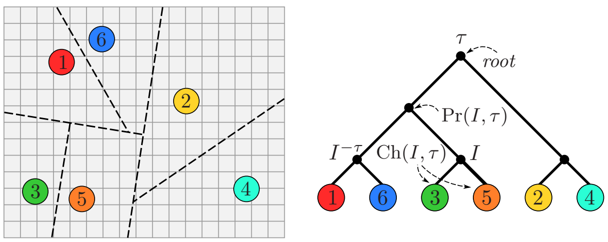
Given an index set, $J = [n] := \{1, \ldots, n\} \subset \mathbb{N}$, a *heterogeneous multirobot configuration*, $\mathbf{x} = (\mathrm{x}_j)_{j \in J}$, is a labeled nonintersecting placement of $|J| = n$ distinct Euclidean spheres,[3] where ith sphere is centered at $\mathrm{x}_i \in \mathbb{R}^d$ and has radius $r_i \geq 0$. We find it convenient to identify the *configuration space* [41] with the set of distinct labelings, i.e., the injective mappings of J into $\mathbb{R}^d$, and, given a vector of nonnegative radii, $\mathbf{r} := (r_j)_{j \in J} \in \left(\mathbb{R}_{\geq 0}\right)^J$, we will find it convenient to denote our "thickened" subset of this configuration space as[4]

$$\mathtt{Conf}\left(\mathbb{R}^d, J, \mathbf{r}\right) := \left\{ \mathbf{x} \in \left(\mathbb{R}^d\right)^J \middle| \left\| \mathrm{x}_i - \mathrm{x}_j \right\| > r_i + r_j, \forall i \neq j \in J \right\}, \quad (1)$$

where $\|.\|$ denotes the standard Euclidean norm on $\mathbb{R}^d$.

B. Cluster Hierarchies

A rooted semi-labelled tree τ over a fixed finite index set J is a directed acyclic graph $G_\tau = (V_\tau, E_\tau)$, whose leaves, vertices of degree one, are bijectively labeled by J and interior vertices all have out-degree at least two; and all of whose edges in E_τ are directed away from a vertex designated to be the *root* [73]. A rooted tree with all interior vertices of out-degree two is said to be *binary* or, equivalently, *nondegenerate*, and all other trees are said to be *degenerate*. In this paper $\mathcal{BT}_J$ denotes the set of rooted nondegenerate trees over leaf set J.



(a) (b)

Fig. 2. An illustration of (a) a heteregeneous configuration of unit disks in $\mathtt{Conf}\left(\mathbb{R}^2, [6], \mathbf{1}\right)$ and (b) its iterative 2-mean clustering [72] hierarchy τ in $\mathcal{BT}_{[6]}$, where the dashed lines in (a) depict the separating hyperplanes between clusters, and (b) illustrates hierarchical cluster relations: parent - $\mathrm{Pr}\,(I, \tau)$, children - $\mathrm{Ch}\,(I, \tau)$, and local complement (sibling) - $I^{-\tau}$ of cluster I of the rooted binary tree, $\tau \in \mathcal{BT}_{[6]}$. An interior node is referred by its cluster, the list of leaves below it; for example, $I = \{3, 5\}$. Accordingly the cluster set of τ is $\mathcal{C}(\tau) = \{\{1\}, \{2\}, \ldots, \{6\}, \{1, 6\}, \{3, 5\}, \{2, 4\}, \{1, 3, 5, 6\}, \{1, 2, 3, 4, 5, 6\}\}$.

A rooted semi-labelled tree τ uniquely determines (and henceforth will be interchangeably termed) a *cluster hierarchy* [74]. By definition, all vertices of τ can be reached from the root through a directed path in τ. The *cluster* of a vertex $v \in V_\tau$ is defined to be the set of leaves reachable from v by a directed path in τ, see Fig. 2. Let $\mathcal{C}\,(\tau)$ denote the set of all vertex clusters of τ.

For every cluster $I \in \mathcal{C}\,(\tau)$ we recall the standard notion of parent (cluster) $\mathrm{Pr}\,(I, \tau)$ and lists of children $\mathrm{Ch}\,(I, \tau)$, ancestors $\mathrm{Anc}\,(I, \tau)$ and descendants $\mathrm{Des}\,(I, \tau)$ of I in τ, illustrated in Fig. 2 — see [33] for explicit definitions of cluster relations. Additionally, we find it useful to define the *local complement (sibling)* of cluster $I \in \mathcal{C}\,(\tau)$ as $I^{-\tau} := \mathrm{Pr}\,(I, \tau) \setminus I$.

C. Configuration Hierarchies

A *hierarchical clustering*[5] $\mathtt{HC} \subset \mathtt{Conf}\left(\mathbb{R}^d, J, \mathbf{r}\right) \times \mathcal{BT}_J$ is a relation from the configuration space $\mathtt{Conf}\left(\mathbb{R}^d, J, \mathbf{r}\right)$ to the space of binary trees $\mathcal{BT}_J$ [8], an example depicted in Fig. 2. In this paper we will only be interested in clustering methods that can classify all possible configurations (i.e. for which $\mathtt{HC}$ assigns some tree to every configuration), and so we need:

Property 1 *$\mathtt{HC}$ is a multi-function.*

Most standard divisive and agglomerative hierarchical clustering methods exhibit this property, but generally fail to be functions because choices may be required between different but equally valid cluster splitting or merging decisions [8].

Given such an $\mathtt{HC}$, for any $\mathbf{x} \in \mathtt{Conf}\left(\mathbb{R}^d, J, \mathbf{r}\right)$ and $\tau \in \mathcal{BT}_J$, we say $\mathbf{x}$ *supports* τ if and only if $(\mathbf{x}, \tau) \in \mathtt{HC}$. The *stratum* associated with a binary hierarchy $\tau \in \mathcal{BT}_J$, denoted

[3] Here, $|A|$ denotes the cardinality of set A.

[4] Here, $\mathbb{R}$ and $\mathbb{R}_{\geq 0}$ denote the set of real and nonnegative real numbers, respectively; and $\mathbb{R}^d$ is the d-dimensional Euclidean space.

[5] Although clustering algorithms generating degenerate hierarchies are available, many standard hierarchical clustering methods return binary clustering trees as a default, thereby avoiding commitment to some "optimal" number of clusters [8], [75].

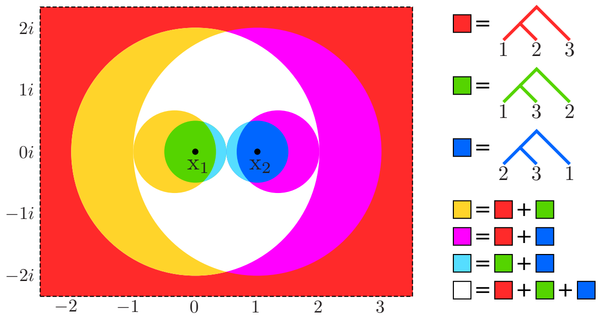


Fig. 3. The Quotient Space $\texttt{Conf}(\mathbb{C}, [3], \mathbf{0}) / \sim$, where for any $\mathbf{x}, \mathbf{y} \in \texttt{Conf}(\mathbb{C}, [3], \mathbf{0})$, $\mathbf{x} \sim \mathbf{y} \iff \frac{\mathrm{x}_3 - \mathrm{x}_1}{\mathrm{x}_2 - \mathrm{x}_1} = \frac{\mathrm{y}_3 - \mathrm{y}_1}{\mathrm{y}_2 - \mathrm{y}_1}$. Here, point particle configurations are quotiented out by translation, scale and rotation, and so $\mathrm{x}_1 = 0 + 0i$, $\mathbf{x}_2 = 1 + 0i$ and $\mathrm{x}_3 \in \mathbb{C} \setminus \{\mathrm{x}_1, \mathrm{x}_2\}$. Regions are colored according the associated cluster hierarchies results from their iterative 2-mean clustering [72]. For instance, any configuration in the white region supports all hierarchies in $\mathcal{BT}_{[3]}$.

by $\mathfrak{S}(\tau) \subset \texttt{Conf}\left(\mathbb{R}^d, J, \mathbf{r}\right)$, is the set of all configurations $\mathbf{x} \in \texttt{Conf}\left(\mathbb{R}^d, J, \mathbf{r}\right)$ supporting the same tree τ [33],

$$\mathfrak{S}(\tau) := \left\{ \mathbf{x} \in \texttt{Conf}\left(\mathbb{R}^d, J, \mathbf{r}\right) \,\middle|\, (\mathbf{x}, \tau) \in \texttt{HC} \right\}, \qquad (2)$$

and this yields a tree-indexed cover of the configuration space. For purposes of illustration, we depict in Fig. 3 the strata of $\texttt{Conf}(\mathbb{C}, [3], \mathbf{0})$ — a space that represents a group of three point particles on the complex plane. [6] [7]

The restriction to binary trees precludes tree degeneracy [73] and we will avoid configuration degeneracy by imposing:

Property 2 *Each stratum of* HC *includes an open subset of configurations, i.e. for every $\tau \in \mathcal{BT}_J$, $\mathring{\mathfrak{S}}(\tau) \neq \varnothing$.*[8]

Once again, most hierarchical clustering methods respect this assumption: they generally all agree (i.e. return the same result) and are robust to small perturbations of a configuration whenever all its clusters are compact and well separated [75].

Given any two configurations supporting the same cluster hierarchy, moving between them while maintaining the shared cluster hierarchy (introduced later as Problem 1) requires:

Property 3 *Each stratum of* HC *is connected.*

For an arbitrary clustering method this requirement is generally not trivial to show, but when clusters of HC are linearly separable, one can characterize the topological shape of each stratum to verify this requirement, as we do in Section V-A.

D. Graphs On Trees

After establishing the relation between multirobot configurations and cluster hierarchies, the final step of our proposed abstraction is to determine the connectivity of tree space.

Define the *adjacency graph* $\mathcal{A}_J = (\mathcal{BT}_J, \mathcal{E}_\mathcal{A})$ to be the 1-skeleton of the nerve [11] of the $\texttt{Conf}\left(\mathbb{R}^d, J, \mathbf{r}\right)$-cover induced by HC. That is to say, a pair of hierarchies, $\sigma, \tau \in \mathcal{BT}_J$, is connected with an edge in $\mathcal{E}_\mathcal{A}$ if and only if their strata intersect, $\mathfrak{S}(\sigma) \cap \mathfrak{S}(\tau) \neq \varnothing$. To enable navigation between structurally different configurations later(Problem 2), we need:

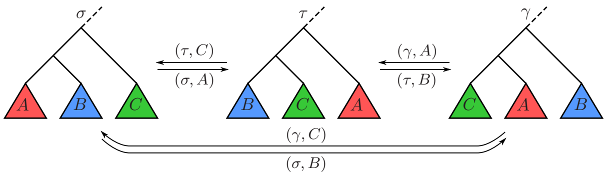


Fig. 4. An illustration of NNI moves between binary trees: each arrow is labeled by a source tree and an associated cluster defining the move.

Property 4 *The adjacency graph is connected.*

Although the adjacency graph is a critical building block of our abstraction, as Fig. 3 anticipates, HC strata generally have complicated shapes, making it usually hard to compute the complete adjacency graph. Fortunately, the computational biology literature [42] offers an alternative notion of adjacency that turns out to be both feasible and nicely compatible with our needs, yielding a computationally effective, connected subgraph of the adjacency graph, $\mathcal{A}_J$, as follows.

The *Nearest Neighbor Interchange (NNI)* move at a cluster $A \in \mathcal{C}(\sigma)$ on a binary tree $\sigma \in \mathcal{BT}_J$, as illustrated in Fig. 4, swaps cluster A with its parent's sibling $C = \text{Pr}(A, \sigma)^{-\sigma}$ to yield another binary tree $\tau \in \mathcal{BT}_J$ [76], [77]. Say that $\sigma, \tau \in \mathcal{BT}_J$ are *NNI-adjacent* if and only if one can be obtained from the other by a single NNI move. Note that a pair of NNI-adjacent trees differs only by one cluster, and the associated NNI moves joining them can be determined by identifying their unshared clusters [78]. Moreover, define the *NNI-graph* $\mathcal{N}_J = (\mathcal{BT}_J, \mathcal{E}_\mathcal{N})$ to have vertex set $\mathcal{BT}_J$, with two trees connected by an edge in $\mathcal{E}_\mathcal{N}$ if and only if they are NNI-adjacent, see Fig. 5. An important contribution of this paper will be to show how the NNI-graph yields a computationally effective subgraph of the adjacency graph (Theorem 6) for our preferred choice of HC.

IV. Hierarchical Navigation Framework

Hierarchical abstraction introduced in Section III intrinsically suggests a two-level navigation strategy for coordinated

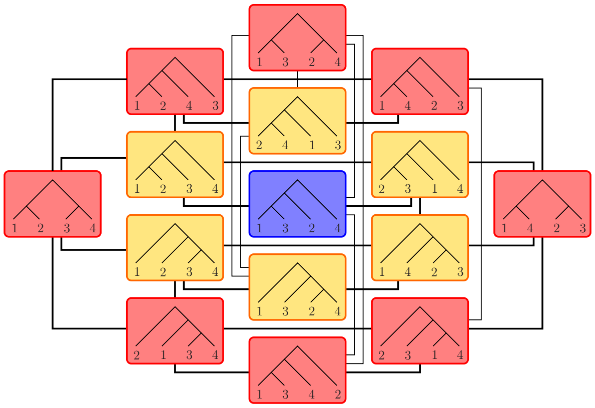


Fig. 5. The NNI Graph: a graphical representation of the space of rooted binary trees, $\mathcal{BT}_J$, with NNI connectivity, where $J = [4] = \{1, 2, 3, 4\}$.

[6] Here, $\mathbf{0}$ and $\mathbf{1}$ are, respectively, vectors of all zeros and ones with the appropriate sizes.

[7] Note that the quotient space in Fig. 3 is not fully symmetric for all three cluster hierarchies because of the nonlinearity of the quotient map. One can visualize the full symmetry of these hierarchical strata by taking the inverse stereographic projection of the planar quotient space onto a sphere.

[8] Here, $\mathring{A}$ denotes the interior of set A.

motion design: (i) at the low-level perform finer adjustments on configurations using hierarchy preserving vector fields, (ii) and at the high-level resolve structural conflicts between configurations using a discrete transition policy in tree space; and the connection between these two levels are established through "portals" — open sets of configurations supporting two adjacent hierarchies. In this section we abstractly describe the generic components of our navigation framework and we show how they are put together.

A. Generic Components of Hierarchical Navigation

1) Hierarchy Preserving Navigation: For ease of exposition we restrict attention to first order (completely actuated single integrator) robot dynamics, and we will be interested in smooth closed loop feedback laws (or hybrid controllers composed from them) that result in complete flows,[9]

$$\dot{\mathbf{x}} = f(\mathbf{x}), \tag{3}$$

where $f : \mathtt{Conf}\big(\mathbb{R}^d, J, \mathbf{r}\big) \rightarrow \big(\mathbb{R}^d\big)^J$ is a vector field over $\mathtt{Conf}\big(\mathbb{R}^d, J, \mathbf{r}\big)$ (1).

Denote by φ^t the *flow* [82] on $\mathtt{Conf}\big(\mathbb{R}^d, J, \mathbf{r}\big)$ induced by the vector field f. For a choice of hierarchical clustering $\mathtt{HC}$, the class of hierarchy-invariant vector fields maintaining the robot group in a specified hierarchical arrangement of clusters, $\tau \in \mathcal{BT}_J$, is defined as [33],

$$\mathcal{F}_{\mathtt{HC}}(\tau) := \left\{ f : \mathtt{Conf}\left(\mathbb{R}^d, J, \mathbf{r}\right) \rightarrow \left(\mathbb{R}^d\right)^J \middle| \varphi^t\left(\mathfrak{S}(\tau)\right) \subset \mathfrak{S}(\tau), t > 0 \right\}. \tag{4}$$

Hierarchy preserving navigation, the low-level component of our framework, uses the vector fields of $\mathcal{F}_{\mathtt{HC}}(\tau)$ to invariantly retract almost all of a stratum onto any designated goal configuration.[10] Thus, we require the availability of such a construction, summarized as:

Problem 1 *For any $\tau \in \mathcal{BT}_J$ and $\mathbf{y} \in \mathfrak{S}(\tau)$ associated with $\mathtt{HC}$ construct a control policy, $f_{\tau,\mathbf{y}}$, using the hierarchy invariant vector fields of $\mathcal{F}_{\mathtt{HC}}(\tau)$ whose closed loop asymptotically results in a retraction, $R_{\tau,\mathbf{y}}$, of $\mathfrak{S}(\tau)$, possibly excluding a set of measure zero*[11]*, onto $\{\mathbf{y}\}$.*

Key for purposes of the present application is the observation that any hierarchy-invariant field $f \in \mathcal{F}_{\mathtt{HC}}(\tau)$ must leave $\mathtt{Conf}\big(\mathbb{R}^d, J, \mathbf{r}\big)$ invariant as well, and thus avoids any self-collisions of the agents along the way. Moreover, the availability of such a family of hierarchy preserving local controllers will enable us to focus on the structural aspects of the multirobot navigation problem while hiding its continuous details such as collision avoidance and stability.

2) Navigation in the Space of Binary Trees: Whereas the controlled deformation retraction, $R_{\tau,\mathbf{y}}$, above generates paths "through" the strata, we will also want to navigate "across" them along the adjacency graph (which will be later in Section V replaced with the NNI-graph — a computationally efficient, connected subgraph). Thus, we further require a construction of a discrete feedback policy in $\mathcal{BT}_J$ that recursively generates paths in the adjacency graph toward any specified destination tree from all other trees in $\mathcal{BT}_J$ by reducing a "discrete Lyapunov function" relative to that destination, summarized as:

Problem 2 *Given any $\tau \in \mathcal{BT}_J$ construct recursively a closed loop discrete dynamical system in the adjacency graph, taking the form of a deterministic discrete transition rule, g_τ, with global attractor at τ endowed with a discrete Lyapunov function relative to the attractor τ.*

Such a recursively generated choice of next hierarchy will play the role of a discrete feedback policy used to define the reset map of our hybrid dynamical system.

3) Hierarchical Portals: Here, we relate the (combinatorial) topology of hierarchical clusters to the (continuous) topology of configurations by defining "portals" — open sets of configurations supporting two adjacent hierarchies.

Definition 1 The *portal*, $\mathtt{Portal}(\sigma,\tau)$, of a pair of hierarchies, $\sigma, \tau \in \mathcal{BT}_J$, is the set of all configurations supporting interior strata of both trees,

$$\mathtt{Portal}(\sigma,\tau) := \mathring{\mathfrak{S}}(\sigma) \cap \mathring{\mathfrak{S}}(\tau). \tag{5}$$

Namely, portals are geometric realizations in the configuration space of the edges of the adjacency graph on trees, see Fig. 3. To realize discrete transitions in tree space via hierarchy preserving navigation in the configuration space, we need a portal map that takes an edge of the adjacency graph, and returns a target configuration in the associated portal:

Problem 3 *Given an edge $(\sigma,\tau) \in \mathcal{E}_{\mathcal{A}}$ of the adjacency graph $\mathcal{A}_J = (\mathcal{BT}_J, \mathcal{E}_{\mathcal{A}})$, construct a geometric realization map $\mathtt{Port}_{(\sigma,\tau)} : \mathfrak{S}(\sigma) \rightarrow \mathtt{Portal}(\sigma,\tau)$ that takes a configuration supporting σ, and returns a target configuration supporting both trees σ and τ.*

A portal map will serve the role of a dynamically computed "prepares graph" [12] for the sequentially composed local controllers whose correct recruitment solves the reactive coordinated motion planning problem (Theorem 1).

B. Specification and Correctness of the Hierarchical Navigation Control (HNC) Algorithm

Assume the selection of a goal configuration $\mathbf{y} \in \mathfrak{S}(\tau)$ and a hierarchy $\tau \in \mathcal{BT}_J$ that $\mathbf{y}$ supports. Now, given (almost) any initial configuration $\mathbf{x} \in \mathfrak{S}(\sigma)$ for some hierarchy $\sigma \in \mathcal{BT}_J$ that $\mathbf{x}$ supports, Table III presents the HNC algorithm whose flowchart is illustrated in Fig. 6. In short, the HNC algorithm solves the collision-free multirobot navigation problem by reactively concatenating low-level continuous hierarchy preserving vector field planners based on a high-level discrete navigation planner in tree space and a selection of a "portal" configuration supporting two adjacent hierarchies. We summarize the important properties of the HNC algorithm as:

[9] A long prior robotics literature motivates the utility of this fully actuated "generalized damper" dynamical model [79], and provides methods for "lifts" to controllers for second order plants [80], [81] as well.

[10] It is important to remark that, instead of a single goal configuration, a more general family of problems can be parametrized by a set of goal configurations sharing a certain homotopy model comprising a set of appropriately nested spheres; and for such a general case one can still construct an exact retraction within our framework.

[11] Recall from [83] that a continuous motion planner in a configuration space X exists if and only if X is contractible. Hence, if a hierarchical stratum is non-contractible (Theorem 2), the domain of such a vector field planner described in Problem 1 must exclude at least a set of measure zero.

TABLE III
THE HNC ALGORITHM

For (almost) any initial $\mathbf{x} \in \mathfrak{S}(\sigma)$ and $\sigma \in \mathcal{BT}_J$, and desired $\mathbf{y} \in \mathfrak{S}(\tau)$ and $\tau \in \mathcal{BT}_J$,

1) (Hybrid Base Case) if $\mathbf{x} \in \mathfrak{S}(\tau)$ then apply stratum-invariant dynamics, $f_{\tau,\mathbf{y}}$ (Problem 1).
2) (Hybrid Recursive Step) else,
    a) invoke the discrete transition rule g_τ (Problem 2) to propose an adjacent tree, $\gamma \in \mathcal{BT}_J$, with lowered discrete Lyapunov value.
    b) Choose local configuration goal, $\mathbf{z} := \mathtt{Port}_{(\sigma,\gamma)}(\mathbf{x})$ (Problem 3).
    c) Apply the stratum-invariant continuous controller $f_{\sigma,\mathbf{z}}$ (Problem 1).
    d) If the trajectory enters $\mathfrak{S}(\tau)$ then go to step 1; else, the trajectory must enter $\mathfrak{S}(\gamma)$ in finite time in which case terminate $f_{\sigma,\mathbf{z}}$, reassign $\sigma \leftarrow \gamma$, and go to step 2a).

Theorem 1 *The HNC Algorithm in Table III defines a hybrid dynamical system whose execution brings almost every initial configuration*[11] *,* $\mathbf{x} \in \mathtt{Conf}\left(\mathbb{R}^d, J\right)$*, in finite time to an arbitrarily small neighborhood of* $\mathbf{y} \in \mathfrak{S}(\tau)$ *with the guarantee of no collisions along the way.*

Proof In the base case, 1) the conclusion follows from the construction of Problem 1: the flow $f_{\tau,\mathbf{y}}$ keeps the state in $\mathfrak{S}(\tau)$, approaches a neighborhood of $\mathbf{y}$ (which is an asymptotically stable equilibrium state for that flow) in finite time.

In the inductive step, a) The NNI transition rule g_τ guarantees a decrement in the Lyapunov function after a transition from σ to γ (Problem 2), and a new local policy $f_{\sigma,\mathbf{z}}$ is automatically deployed with a local goal configuration $\mathbf{z} \in \mathtt{Portal}(\sigma,\gamma)$ found in b). Next, the flow $f_{\sigma,\mathbf{z}}$ in c) is guaranteed to keep the state in $\mathfrak{S}(\sigma)$ and approach $\mathbf{z} \in \mathtt{Portal}(\sigma,\gamma)$ asymptotically from almost all initial configurations. If the base case is not triggered in d), then the state enters arbitrarily small neighborhoods of $\mathbf{z}$ and, hence, must eventually reach $\mathtt{Portal}(\sigma,\gamma) \subset \mathfrak{S}(\gamma)$ in finite time, triggering a return to 2a). Because the dynamical transitions g_τ initiated from any hierarchy in $\mathcal{BT}_J$ reaches τ in finite steps (Problem 2), it must eventually trigger the base case. ■

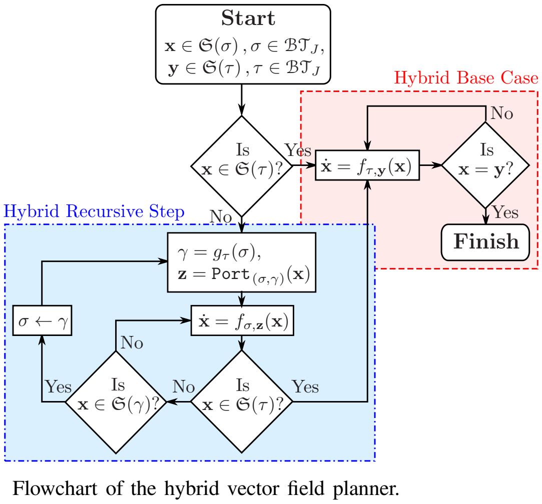

Start
$\mathbf{x} \in \mathfrak{S}(\sigma), \sigma \in \mathcal{BT}_J,$
$\mathbf{y} \in \mathfrak{S}(\tau), \tau \in \mathcal{BT}_J$
Hybrid Base Case
Is $\mathbf{x} \in \mathfrak{S}(\tau)$?
Yes
$\dot{\mathbf{x}} = f_{\tau,\mathbf{y}}(\mathbf{x})$
Is $\mathbf{x} = \mathbf{y}$?
No
Yes
Finish
Hybrid Recursive Step
No
$\gamma = g_\tau(\sigma),$
$\mathbf{z} = \mathtt{Port}_{(\sigma,\gamma)}(\mathbf{x})$
$\sigma \leftarrow \gamma$
$\dot{\mathbf{x}} = f_{\sigma,\mathbf{z}}(\mathbf{x})$
No
Yes
Is $\mathbf{x} \in \mathfrak{S}(\gamma)$?
No
Is $\mathbf{x} \in \mathfrak{S}(\tau)$?
Yes

Fig. 6. Flowchart of the hybrid vector field planner.

V. Hierarchical Navigation of Euclidean Spheres via Bisecting K-means Clustering

We now confine our attention to 2-means divisive hierarchical clustering [72], $\mathtt{HC}_{\text{2-means}}$, and demonstrate a construction of our hierarchical navigation framework for coordinated navigation of Euclidean spheres via $\mathtt{HC}_{\text{2-means}}$.

A. Hierarchical Strata of $\mathtt{HC}_{\text{2-means}}$

Iterative 2-means clustering, $\mathtt{HC}_{\text{2-means}}$, is a divisive method that recursively constructs a cluster hierarchy of a configuration in a top-down fashion [72]. Briefly, this method splits each successive (partial) configuration by applying 2-means clustering, and successively continues with each subsplit until reaching singletons. By construction, complementary configuration clusters of $\mathtt{HC}_{\text{2-means}}$ are linearly separable by a hyperplane defined by the associated cluster centroids[12], as illustrated in Fig. 2; and the stratum of $\mathtt{HC}_{\text{2-means}}$ associated with a binary hierarchy $\tau \in \mathcal{BT}_J$ can be characterized by the intersection inverse images,

$$\mathfrak{S}(\tau) = \bigcap_{I \in \mathcal{C}(\tau)\setminus\{J\}} \bigcap_{i \in I} \eta_{i,I,\tau}^{-1}[0,\infty), \tag{6}$$

of the scalar valued "separation" function, $\eta_{i,I,\tau} : \mathtt{Conf}\left(\mathbb{R}^d, J, \mathbf{r}\right) \to \mathbb{R}$ [33] returning the distance of agent i in cluster $I \in \mathcal{C}(\tau) \setminus \{J\}$ to the perpendicular bisector of the centroids of complementary clusters I and $I^{-\tau}$: [13]

$$\eta_{i,I,\tau}(\mathbf{x}) := \left(\mathrm{x}_i - \mathrm{m}_{I,\tau}(\mathbf{x})\right)^{\mathrm{T}} \frac{\mathrm{s}_{I,\tau}(\mathbf{x})}{\left\|\mathrm{s}_{I,\tau}(\mathbf{x})\right\|}, \tag{7}$$

where the associated "cluster functions" of a partial configuration, $\mathbf{x}|I = (\mathrm{x}_i)_{i\in I}$, are defined as

$$\mathrm{c}(\mathbf{x}|I) := \frac{1}{|I|}\sum_{i\in I}\mathrm{x}_i, \tag{8}$$

$$\mathrm{s}_{I,\tau}(\mathbf{x}) := \mathrm{c}(\mathbf{x}|I) - \mathrm{c}\left(\mathbf{x}|I^{-\tau}\right), \tag{9}$$

$$\mathrm{m}_{I,\tau}(\mathbf{x}) := \frac{\mathrm{c}(\mathbf{x}|I) + \mathrm{c}\left(\mathbf{x}|I^{-\tau}\right)}{2}. \tag{10}$$

We now follow [40] in defining terminology and expresssions leading to the characterization of the homotopy type of the stratum, $\mathfrak{S}(\tau)$, associated with a nondegenerate hierarchy. The proofs of our formal statements all follow the same pattern as established in [40], and we omit them to save space here.

Definition 2 A configuration $\mathbf{x} \in \mathtt{Conf}\left(\mathbb{R}^d, J, \mathbf{r}\right)$ is *narrow* relative to the split, $\{I, J \setminus I\}$, if

$$\max_{A\in\{I,J\setminus I\}} r(\mathbf{x}|A) \;<\; \frac{1}{2}\left\|\mathrm{c}(\mathbf{x}|I) - \mathrm{c}(\mathbf{x}|J\setminus I)\right\|, \tag{11}$$

where the radius of a cluster, $A \subset J$, is defined to be[14]

$$r(\mathbf{x}|A) := \max_{a\in A}\left(\left\|\mathrm{x}_a - \mathrm{c}(\mathbf{x}|A)\right\| + r_a\right). \tag{12}$$

[12] In the context of self-sorting in heterogeneous swarms [28], two groups of robot swarms are said to be *segregated* if their configurations are linearly separable; and in this regard configuration hierarchies of $\mathtt{HC}_{\text{2-means}}$ represent spatially cohesive and segregated swarms groups at different resolutions.

[13] Here, $\mathbf{A}^{\mathrm{T}}$ denotes the transpose of a matrix $\mathbf{A}$.

[14] Recall from p.4 that r_i denotes the radius of ith sphere for any $i \in J$.

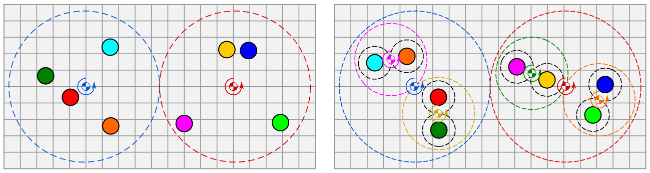


Fig. 7. An illustration of (left) narrow and (right) standard disk configurations, where arrows and dashed circles indicate clusters that can be rigidly rotated around their centroids while preserving their clustering structures.

Say that $\mathbf{x} \in \mathfrak{S}(\tau)$ is a *standard* configuration relative to the nondegenerate hierarchy, $\tau \in \mathcal{BT}_J$, if it is narrow relative to each local split, $\mathrm{Ch}\,(I, \tau)$, of every cluster, $I \in \mathcal{C}\,(\tau)$.

Since configuration hierarchies of $\mathtt{HC}_{\text{2-means}}$ are invariant under rigid transformations, and the separating hyperplanes of complementary clusters are preserved whenever the associated cluster centroids are kept unchanged, one can observe that:

Proposition 1 *If $\mathbf{x} \in \mathfrak{S}(\tau)$ is a standard configuration then for each cluster, $I \in \mathcal{C}\,(\tau)$, any rigid rotation of the partial configuration, $\mathbf{x}|I$, around its centroid, $\mathrm{c}\,(\mathbf{x}|I)$, as illustrated in Fig. 7, preserves the supported hierarchy τ.*

Proposition 2 *For any finite label set $J \subset \mathbb{N}$ and binary tree $\tau \in \mathcal{BT}_J$, there exists a strong deformation retraction*[15]

$$R_\tau : \mathfrak{S}(\tau) \times [0, 1] \to \mathfrak{S}(\tau) \qquad (13)$$

of $\mathfrak{S}(\tau)$ onto the subset of standard configurations of $\mathfrak{S}(\tau)$.

These two observations yield the key insight reported in [40].

Theorem 2 *The set of configurations $\mathbf{x} \in \mathtt{Conf}\big(\mathbb{R}^d, J, \mathbf{r}\big)$ supporting a binary tree has the homotopy type of $\big(\mathbb{S}^{d-1}\big)^{|J|-1}$.*

To gain an intuitive appreciation, one can restate this result as follows: two configurations in $\mathfrak{S}(\tau)$ are topologically equivalent if and only if the corresponding separating hyperplane normals of configuration clusters are the same.[16] Hence, navigation in a hierarchical stratum is carried out by aligning separating hyperplane normals[17] [18], illustrated in Fig. 8; and using this geometric intuition, we construct in [33] a family of hierarchy preserving control policies for point particle configurations, and in the following we extend that construction to thickened disk configurations.

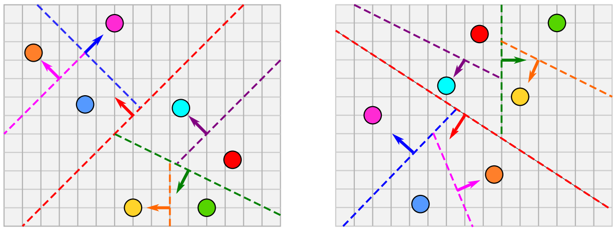


Fig. 8. The topological shape of a hierarchical stratum intuitively suggests that global navigation in a hierarchical stratum is accomplished by aligning separating hyperplanes of configurations.

Theorem 3 *Iterative 2-means clustering $\mathtt{HC}_{\text{2-means}}$ is a multi-function, and each of its stratum, $\mathfrak{S}(\tau)$ associated with $\tau \in \mathcal{BT}_J$, is connected and has an open interior.*

Proof It is well known that k-means clustering is a multi-function generally yielding different k-partitions of any given data, and so is $\mathtt{HC}_{\text{2-means}}$ (Property 1) [8], [75]. Further, it follows from Definition 2 and Proposition 2 that standard configurations in $\mathfrak{S}(\tau)$ is open (Property 2), and Theorem 2 guarantees the connectedness of $\mathfrak{S}(\tau)$ (Property 3). ■

B. Hierarchy Preserving Navigation

We now introduce a recursively defined vector field for navigation in a hierarchical stratum and list its invariance and stability properties.

Suppose that some desired configuration, $\mathbf{y} \in \mathfrak{S}(\tau)$ has been selected, supporting some desired nondegenerate tree, $\tau \in \mathcal{BT}_J$. Our dynamical planner takes the form of a centralized hybrid controller, $f_{\tau,\mathbf{y}} : \mathfrak{S}(\tau) \to \big(\mathbb{R}^d\big)^{|J|}$, defining a hierarchy-invariant vector field whose flow in $\mathfrak{S}(\tau)$ yields the desired goal configuration, $\mathbf{y}$, recursively defined according to logic presented in Table IV. Throughout this section, the tree τ and the goal configuration $\mathbf{y}$ are fixed, and we therefore suppress all mention of these terms wherever convenient, in order to compress the notation. For example, for any $\mathbf{x} \in \mathfrak{S}(\tau)$, $I \in \mathcal{C}\,(\tau)$ and $i \in I$ we use the shorthand $\eta_{i,I}\,(\mathbf{x}) = \eta_{i,I,\tau}\,(\mathbf{x})$ (7), $\mathrm{s}_I\,(\mathbf{x}) = \mathrm{s}_{I,\tau}\,(\mathbf{x})$ (9), $\mathrm{m}_I\,(\mathbf{x}) = \mathrm{m}_{I,\tau}\,(\mathbf{x})$ (10) and so on.

TABLE IV
THE HIERARCHY-PRESERVING NAVIGATION VECTOR FIELD

For any initial $\mathbf{x} \in \mathfrak{S}(\tau)$ and desired $\mathbf{y} \in \mathfrak{S}(\tau)$, supporting $\tau \in \mathcal{BT}_J$, the hierarchy preserving vector field, $f_{\tau,\mathbf{y}} : \mathfrak{S}(\tau) \to \big(\mathbb{R}^d\big)^J$,

$$f_{\tau,\mathbf{y}}\,(\mathbf{x}) := \hat{f}_{\tau,\mathbf{y}}\,(\mathbf{x}, \mathbf{0}, J)\,,$$

is recursively computed starting at the root cluster J with the zero control input $\mathbf{0} \in \big(\mathbb{R}^d\big)^J$ as follows: for any $\mathbf{u} \in \big(\mathbb{R}^d\big)^J$ and $I \in \mathcal{C}\,(\tau)$,

```
function $\hat{\mathbf{u}} = \hat{f}_{\tau,\mathbf{y}}(\mathbf{x}, \mathbf{u}, I)$
Base Cases
  if $\mathbf{x} \in \mathcal{D}_A(I)$ (15),
    $\hat{\mathbf{u}} \leftarrow f_A(\mathbf{x}, \mathbf{u}, I)$ (14),          % Attracting Field
  else if $\mathbf{x} \notin \mathcal{D}_H(I)$ (18),
    $\hat{\mathbf{u}} \leftarrow f_S(\mathbf{x}, \mathbf{u}, I)$ (24),          % Split Separation Field
Recursion
  else
    $\{I_L, I_R\} \leftarrow \mathrm{Ch}(I, \tau)$,
    $\hat{\mathbf{u}}_L \leftarrow \hat{f}_{\tau,\mathbf{y}}(\mathbf{x}, \mathbf{u}, I_L)$,        % Recursion for Left Child
    $\hat{\mathbf{u}}_R \leftarrow \hat{f}_{\tau,\mathbf{y}}(\mathbf{x}, \hat{\mathbf{u}}_L, I_R)$,       % Recursion for Right Child
    $\hat{\mathbf{u}} \leftarrow f_H(\mathbf{x}, \hat{\mathbf{u}}_R, I)$ (19),      % Split Preserving Field
  end
return $\hat{\mathbf{u}}$
```

[15] In [40] authors study point particle configurations, and they construct a strong deformation retraction onto standard configurations by shrinking clusters around their centroids; and one can obtain this result for thickened spheres by properly expanding cluster configurations instead of shrinking.

[16] Note that a binary hierarchy over the leaf set J has $|J| - 1$ interior nodes, i.e. nonsingleton clusters [76].

[17] In [33] we construct a linear bijective mapping relating the configuration space and the centroidal separations of complementary clusters of any given hierarchy such that a multirobot configuration is uniquely determined by its centroid and the centroidal separations of complementary clusters of the associated hierarchy. Hence, since the Euclidean d-space and a connected subset of the real line are both contractible, one can establish the intuitive connection between the separating hyperplane normals and the homotopy type of a hierarchical stratum in Theorem 2.

[18] For the stability analysis of hierarchy invariant local policies of point particle configurations we use in [33] a Lyapunov function that quantifies how well the separating hyperplanes of the current and the desired multirobot configurations are aligned. Similarly, in the proof of Proposition 11 we also show that the separating hyperplane normals of complementary clusters are asymptotically aligned with the desired ones.

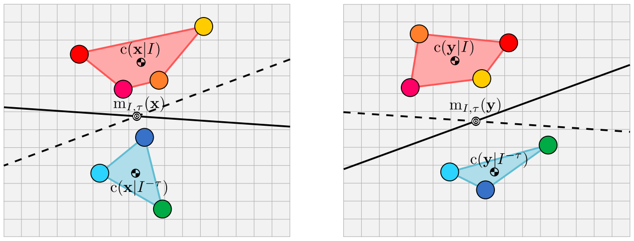

$\mathrm{c}(\mathbf{x}|I)$
$\mathrm{m}_{I,\tau}(\mathbf{x})$
$\mathrm{c}(\mathbf{x}|I^{-\tau})$
$\mathrm{c}(\mathbf{y}|I)$
$\mathrm{m}_{I,\tau}(\mathbf{y})$
$\mathrm{c}(\mathbf{y}|I^{-\tau})$

Fig. 9. An illustration of "sufficiently aligned" separating hyperplanes of complementary clusters I and $I^{-\tau}$ of τ. Both of the current (left) and desired (right) partial configurations are linearly separable by each others separating hyperplane, and such an alignment condition needs to be satisfied at each level of the subtrees rooted at I and $I^{-\tau}$ so that the partial configurations $\mathbf{x}|I$ and $\mathbf{x}|I^{-\tau}$ are steered by the associated attracting fields.

In brief, the hierarchy invariant vector field $f_{\tau,\mathbf{y}}$ recursively detects partial configurations whose separating hyperplanes are "sufficiently aligned" with the desired ones, as specified in (15) and illustrated in Fig. 9, and that can be directly moved towards the desired configurations, using a family of attracting fields f_A (14), with no collisions along the way. Once the partial configurations associated with sibling clusters I and $I^{-\tau}$ of τ are in the domains of their associated attracting fields, $f_{\tau,\mathbf{y}}$ rotates these partial configurations while preserving the hierarchy so that their separating hyperplane is also asymptotically aligned. Hence, $f_{\tau,\mathbf{y}}$ asymptotically aligns the separating hyperplanes of clusters of τ in a bottom-up fashion; and once the separating hyperplanes of all clusters of τ are "sufficiently aligned", $f_{\tau,\mathbf{y}}$ drives asymptotically each disk directly towards its desired location. We now present and motivate its constituent formulae as follows.

The hierarchy-invariant vector field, $f_{\tau,\mathbf{y}}$, in Table IV.2) & IV.3) recursively detects partial configurations, $\mathbf{x}|I$ associated with cluster $I \in \mathcal{C}(\tau)$, that can be safely driven toward the goal formation in $\mathfrak{S}(\tau)$ using a family of attracting controllers, $f_A : \mathfrak{S}(\tau) \times \left(\mathbb{R}^d\right)^J \times \mathcal{C}(\tau) \rightarrow \left(\mathbb{R}^d\right)^J$, defined in terms of the negated gradient field of $V(\mathbf{x}) := \frac{1}{2}\left\|\mathbf{x}-\mathbf{y}\right\|_2^2$: for any $j \in J$,

$$f_A(\mathbf{x},\mathbf{u},I)_j := \begin{cases} -(\mathrm{x}_j - \mathrm{y}_j), & \text{if } j \in I, \\ \mathrm{u}_j & \text{, else,} \end{cases} \tag{14}$$

where $\mathbf{u} \in \left(\mathbb{R}^d\right)^J$ is a desired (velocity) control input specifying the motion of complementary cluster $J \setminus I$.

To avoid intra-cluster collisions along the way and preserve (local) clustering hierarchy, for any $I \in \mathcal{C}(\tau)$ the set of configurations in the domain of the attracting field, f_A, is restricted to

$$\begin{aligned}\mathcal{D}_A(I) := \Big\{\mathbf{x} \in \mathfrak{S}(\tau) \Big| \, & \mathcal{L}_{\overrightarrow{\mathbf{y}}} \tfrac{1}{2}\left\|\mathrm{x}_i - \mathrm{x}_j\right\|^2 \geq (r_i + r_j)^2, \forall i \neq j \in I, \\ & \mathcal{L}_{\overrightarrow{\mathbf{y}}}(\mathrm{x}_k - \mathrm{m}_K(\mathbf{x}))^{\mathrm{T}} \mathrm{s}_K(\mathbf{x}) \geq 0, \forall k \in K, K \in \mathrm{Des}(I,\tau)\Big\},\end{aligned} \tag{15}$$

where $\mathrm{Des}(I,\tau)$ is the set of descendants of I in τ. Here, $\mathcal{L}_{\overrightarrow{\mathbf{y}}} f$ denotes the Lie derivative of a scalar-valued function f along a constant vector field $\overrightarrow{\mathbf{y}}$ which assigns the same vector $\mathbf{y}$ to every point in its domain, and one can simply verify that

$$\mathcal{L}_{\overrightarrow{\mathbf{y}}} \tfrac{1}{2}\left\|\mathrm{x}_i - \mathrm{x}_j\right\|^2 = (\mathrm{x}_i - \mathrm{x}_j)^{\mathrm{T}} (\mathrm{y}_i - \mathrm{y}_j), \tag{16}$$

$$\begin{aligned}\mathcal{L}_{\overrightarrow{\mathbf{y}}}(\mathrm{x}_k - \mathrm{m}_K(\mathbf{x}))^{\mathrm{T}} \mathrm{s}_K(\mathbf{x}) = {} & (\mathrm{y}_k - \mathrm{m}_K(\mathbf{y}))^{\mathrm{T}} \mathrm{s}_K(\mathbf{x}) \\ & + (\mathrm{x}_k - \mathrm{m}_K(\mathbf{x}))^{\mathrm{T}} \mathrm{s}_K(\mathbf{y}).\end{aligned} \tag{17}$$

Note that (16) quantifies the safety of a resulting trajectory of f_A, and to avoid collision between any pair of disks, i and j, (16) should be no less than the square of sum of their radii, $(r_i + r_j)^2$, as required in (15); and (17) quantifies the preservation of (local) clustering hierarchy and should be nonnegative for hierarchy invariance. Also observe that since a singleton cluster contains no pair of distinct indices, and has an empty set of descendants, the predicate in (15) is always true for these "leaf" node cases and we have $\mathcal{D}_A(I) = \mathfrak{S}(\tau)$ for any singleton cluster $I \in \mathcal{C}(\tau)$. Further, one can simply verify that $\mathbf{y} \in \mathcal{D}_A(I)$ for any $I \in \mathcal{C}(\tau)$.

If a partial configuration, $\mathbf{x}|I$, is not contained in the domain of the associated attracting field, i.e. $\mathbf{x} \notin \mathcal{D}_A(I)$, to avoid inter-cluster collisions the failure of the condition in Table IV.4) ensures sibling clusters, $\mathrm{Ch}(I,\tau)$, will be separated by a certain distance, specified as:

$$\mathcal{D}_H(I) := \Big\{\mathbf{x} \in \mathfrak{S}(\tau) \Big| \eta_{k,K}(\mathbf{x}) \geq r_k + \alpha, \forall k \in K, K \in \mathrm{Ch}(I,\tau)\Big\}, \tag{18}$$

where $\eta_{k,K}(\mathbf{x})$ (7) returns the perpendicular distance of kth agent to the separating hyperplane of cluster $K \in \mathcal{C}(\tau)$, and $\alpha > 0$ is a safety margin guaranteeing that the clearance between any pair of disks in complementary clusters, $\mathrm{Ch}(I,\tau)$, is at least 2α units. Observe that $\mathcal{D}_H(I) = \mathfrak{S}(\tau)$ for any singleton cluster $I \in \mathcal{C}(\tau)$ because such leaf clusters of a binary tree have no children, i.e. $\mathrm{Ch}(I,\tau) = \varnothing$.

While the disks move in $\mathcal{D}_H(I)$ based on a desired control (velocity) input $\mathbf{u} \in \left(\mathbb{R}^d\right)^J$, Table IV.10) guarantees the maintenance of the safety margin between children clusters $\mathrm{Ch}(I,\tau)$ by employing an additive repulsive field, $f_H : \mathfrak{S}(\tau) \times \left(\mathbb{R}^d\right)^J \times \mathcal{C}(\tau) \rightarrow \left(\mathbb{R}^d\right)^J$, that rigidly pushes the children clusters apart as follows:

$$f_H(\mathbf{x},\mathbf{u},I)_j := \mathrm{u}_j + 2\alpha_I(\mathbf{x},\mathbf{u}) \frac{\left|K^{-\tau}\right|}{|I|} \frac{\mathrm{s}_K(\mathbf{x})}{\left\|\mathrm{s}_K(\mathbf{x})\right\|}, \tag{19}$$

for all $j \in K$ and $K \in \mathrm{Ch}(I,\tau)$; otherwise, $f_H(\mathbf{x},\mathbf{u},I)_j := \mathrm{u}_j$, where $\alpha_I(\mathbf{x},\mathbf{u})$ is a scalar valued function describing the strength of the repulsive field,

$$\alpha_I(\mathbf{x},\mathbf{u}) := \max_{\substack{k \in K \\ K \in \mathrm{Ch}(I,\tau)}} \phi_{k,K}(\mathbf{x}) \cdot \psi_{k,K}(\mathbf{x},\mathbf{u}). \tag{20}$$

Here, for each individual k in cluster $K \in \mathrm{Ch}(I,\tau)$, $\phi_{k,K}(\mathbf{x})$ is exponential damping on the repulsion strength $\psi_{k,K}(\mathbf{x},\mathbf{y})$, in which the amplitude envelop exponentially decays to zero after a certain safety margin $\beta > \alpha$,

$$\phi_{k,K}(\mathbf{x}) := \max\left(\frac{e^{-\left(\eta_{k,K}(\mathbf{x}) - r_k - \alpha\right)} - e^{-(\beta-\alpha)}}{1 - e^{-(\beta-\alpha)}}, 0\right), \tag{21}$$

$$\psi_{k,K}(\mathbf{x},\mathbf{u}) := \max\Big(-\left(\eta_{k,K}(\mathbf{x}) - r_k - \alpha\right) - \mathcal{L}_{\overrightarrow{\mathbf{u}}}\eta_{k,K}(\mathbf{x}), 0\Big), \tag{22}$$

where

$$\begin{aligned}\mathcal{L}_{\overrightarrow{\mathbf{u}}}\eta_{k,K}(\mathbf{x}) = {} & \frac{(\mathrm{u}_k - \mathrm{m}_K(\mathbf{u}))^{\mathrm{T}} \mathrm{s}_K(\mathbf{x}) + (\mathrm{x}_k - \mathrm{m}_K(\mathbf{x}))^{\mathrm{T}} \mathrm{s}_K(\mathbf{u})}{\left\|\mathrm{s}_K(\mathbf{x})\right\|} \\ & - \eta_{k,K}(\mathbf{x}) \frac{\mathrm{s}_K(\mathbf{x})^{\mathrm{T}} \mathrm{s}_K(\mathbf{u})}{\left\|\mathrm{s}_K(\mathbf{x})\right\|^2}.\end{aligned} \tag{23}$$

Note that $f_H(\mathbf{x},\mathbf{u},I)$ is well defined for any singleton cluster $I \in \mathcal{C}(\tau)$ and is equal to the identity map, i.e. $f_H(\mathbf{x},\mathbf{u},I) = \mathbf{u}$, since $\mathrm{Ch}(I,\tau) = \varnothing$; and also observe that $f_H(\mathbf{x},\mathbf{u},I) = \mathbf{u}$ for any $I \in \mathcal{C}(\tau)$ if the complementary clusters $\mathrm{Ch}(I,\tau)$ are well-separated, i.e. $\eta_{k,K}(\mathbf{x}) \geq r_k + \beta$ for all $k \in K$ and $K \in \mathrm{Ch}(I,\tau)$. The latter is important to avoid the "finite escape time" phenomenon[19] (Proposition 14).

Finally, Table IV.5) guarantees that if a partial configuration is neither in the domain of the attracting field nor are its children clusters, $\mathrm{Ch}(I,\tau)$, properly separated, i.e. $\mathbf{x} \notin \mathcal{D}_A(I) \cup \mathcal{D}_H(I)$, then the complementary clusters are driven apart using another repulsive field, $f_S : \mathfrak{S}(\tau) \times \left(\mathbb{R}^d\right)^J \times \mathcal{C}(\tau) \to \left(\mathbb{R}^d\right)^J$, until asymptotically establishing a certain safety margin $\beta > \alpha$:

$$f_S(\mathbf{x},\mathbf{u},I)_j := -\,\mathrm{c}\,(\mathbf{x}-\mathbf{y}|I) + 2\beta_I(\mathbf{x})\frac{|K^{-\tau}|}{|I|}\frac{\mathrm{s}_K(\mathbf{x})}{\|\mathrm{s}_K(\mathbf{x})\|}, \quad (24)$$

for all $j \in K$ and $K \in \mathrm{Ch}(I,\tau)$; otherwise, $f_S(\mathbf{x},\mathbf{u},I)_j := \mathrm{u}_j$, where the magnitude, $\beta_I(\mathbf{x})$, of repulsion between complementary clusters $\mathrm{Ch}(I,\tau)$ is given by

$$\beta_I(\mathbf{x}) := \max_{\substack{k \in K \\ K \in \mathrm{Ch}(I,\tau)}} \max\left(-\left(\eta_{k,K}(\mathbf{x}) - r_k - \beta\right), 0\right). \quad (25)$$

For completeness, we set $f_S(\mathbf{x},\mathbf{u},I) = f_A(\mathbf{x},\mathbf{u},I)$ for any singleton cluster $I \in \mathcal{C}(\tau)$.

We summarize the properties of this construction as:[20]

Theorem 4 *The recursion of Table IV results in a well-defined function $f_{\tau,\mathbf{y}} : \mathfrak{S}(\tau) \to \left(\mathbb{R}^d\right)^J$ that can be computed in $\mathrm{O}\left(|J|^2\right)$ time for any $\mathbf{x} \in \mathfrak{S}(\tau)$. For all $\tau \in \mathcal{BT}_J$, the stratum $\mathfrak{S}(\tau)$ is positive invariant and any $\mathbf{y} \in \mathfrak{S}(\tau)$ is an asymptotically stable equilibrium point of a continuous piecewise smooth flow arising from $f_{\tau,\mathbf{y}}$ whose basin of attraction includes all, except a set of measure zero[21], of $\mathfrak{S}(\tau)$.*

Proof These results are proven in Appendix I according to the following plan. Proposition 3 establishes that the recursion in Table IV indeed results in a function computable in quadratic time. The invariance, stability, and continuous flow generating properties of $f_{\tau,\mathbf{y}}$ are shown using an equivalent system model within the sequential composition framework [12], as follows. Table VI defines a new recursion shown in Proposition 4 to result in a family of continuous and piecewise smooth vector fields. Proposition 5 asserts that the family of domains associated with these fields (44) defines a (finite) open cover of $\mathfrak{S}(\tau)$ relative to which a selection function (Table VII) induces a partition of that stratum. Proposition 6 demonstrates that the composition of the covering vector field family with the output of this partitioning function yields a new function that coincides exactly with the original control field defined in Table IV. Finally, Proposition 14, Proposition 13 and Proposition 15 demonstrate, respectively, the flow, positive invariance and stability properties of $f_{\tau,\mathbf{y}}$, which are inherited from the flow, invariance and stability properties (Proposition 10, Proposition 9 and Proposition 11, respectively) of substratum policies executed over a strictly decreasing finite prepares graph (Proposition 7) via their nondegenerately, real-time executed (Proposition 12) sequential composition. ■

C. Navigation in the Space of Binary Trees

In principle, navigation in the adjacency graph of trees (Problem 2) is a trivial matter since the number of trees over a finite set of leaves is finite. However, in practice, the cardinality of trees grows super exponentially [73],

$$|\mathcal{BT}_J| = (2|J|-3)!! = (2|J|-3)(2|J|-5)\ldots 3, \quad (26)$$

for $|J| \geq 2$. Hence standard graph search algorithms, like the A* or Dijkstra's algorithm [86], become rapidly impracticable. In particular, computing the shortest path (geodesic) in the NNI-graph, a regular subgraph of the adjacency graph (Theorem 6), is NP-complete [87].

Alternatively, we have recently developed in [78] an efficient recursive procedure for navigating in the NNI graph $\mathcal{N}_J = (\mathcal{BT}_J, \mathcal{E}_\mathcal{N})$ towards any given binary tree $\tau \in \mathcal{BT}_J$, taking the form of a discrete dynamical system as follows:

$$\sigma^{k+1} = \mathrm{NNI}\left(\sigma^k, G^k\right), \quad (27a)$$

$$G^k = \mathbf{u}_\tau(\sigma^k), \quad (27b)$$

where $\mathrm{NNI}\left(\sigma^k, G^k\right)$ denotes the NNI move[22] on σ^k at cluster $G^k \in \mathcal{C}(\tau)$, illustrated in Fig. 4, and $\mathbf{u}_\tau$ is our NNI control policy returning an NNI move as summarised in Table V. Abusing notation, we shall denote the closed-loop system as

$$\sigma^{k+1} = g_\tau\left(\sigma^k\right) := (\mathrm{NNI} \circ \mathbf{u})\left(\sigma^k\right). \quad (28)$$

In short, since a binary cluster hierarchy is a maximal collection of "compatible" clusters and two distinct binary hierarchy always have some incompatible clusters, the NNI control law recursively identifies and fixes cluster incompatibilities of any given hierarchy with the desired target hierarchy, refer to [78] for more details.

TABLE V
THE NNI CONTROL LAW

To navigate from an arbitrary hierarchy $\sigma \in \mathcal{BT}_J$ towards any selected desired hierarchy $\tau \in \mathcal{BT}_J$ in the NNI-graph, the NNI control policy $\mathbf{u}_\tau$ returns an NNI move on σ at a cluster $G \in \mathcal{C}(\sigma)$, as follows:

1) If $\sigma = \tau$, then just return the identity move, $G = \varnothing$.
2) Otherwise,
   a) Select a common cluster $K \in \mathcal{C}(\sigma) \cap \mathcal{C}(\tau)$ with $\mathrm{Ch}(K,\sigma) \neq \mathrm{Ch}(K,\tau)$, and let $\{K_L, K_R\} = \mathrm{Ch}(K,\tau)$.
   b) Find a nonsingleton cluster $I \in \mathcal{C}(\sigma)$ with children $\{I_L, I_R\} = \mathrm{Ch}(I,\sigma)$ satisfying $I_L \subseteq K_L$ and $I_R \subseteq K_R$.
   c) Return a proper NNI navigation move on σ at grandchild $G \in \mathrm{Ch}(I,\sigma)$ selected as follows:
      i) If $I^{-\sigma} \subset K_L$, then return $G = I_R$.
      ii) Else if $I^{-\sigma} \subset K_R$, then return $G = I_L$.
      iii) Otherwise , return an arbitrary NNI move at a child of I in σ; for example, $G = I_L$.

[19] A trajectory of a dynamical system is said to have a finite escape time if it escapes to infinity at a finite time [84].

[20] This construction indeed solves Problem 1 since a flow is a retraction of its basin into the attractor [85].

[21] It follows from Theorem 2 that the measure zero set excluded from the basin of $\mathbf{y}$ under the flow generated by $f_{\tau,\mathbf{y}}$ is the set of configurations in $\mathfrak{S}(\tau)$ whose separating hyperplane normals are in the opposite direction from the associated separating hyperplane normal of $\mathbf{y}$ for at least one pair of complementary clusters of τ.

[22] Here, note that the NNI move at the empty cluster corresponds to the identity map in $\mathcal{BT}_J$, i.e. $\sigma = \mathrm{NNI}(\sigma, \varnothing)$ for all $\sigma \in \mathcal{BT}_J$. Therefore, the notion of identity map in $\mathcal{BT}_J$ slightly extends the NNI graph by adding self-loops at every vertex, which is necessary for a discrete-time dynamical system in $\mathcal{BT}_J$ to have fixed points.

The NNI control law endows the NNI-graph with a directed edge structure whose paths all lead to τ, and whose longest path (from the furthest possible initial hierarchy, $\sigma \in \mathcal{BT}_J$) is tightly bounded by $\frac{1}{2}\left(|J|-1\right)\left(|J|-2\right)$ for $|J| \geq 2$. Given such a goal we show in [78] that the cost of computing an appropriate NNI move from any other $\sigma \in \mathcal{BT}_J$ toward an adjacent tree at a lower value of a "discrete Lyapunov function" relative to that destination is $\mathrm{O}\left(|J|\right)$. We summarize such important properties of our NNI navigation algorithm as:

Theorem 5 ([78]) *The NNI control law* u_τ *(Table V) recursively defines a closed loop discrete dynamical system (28) in the NNI-graph, taking the form of a discrete transition rule,* g_τ*, with global attractor at* τ *and longest trajectory of length* $\mathrm{O}\left(|J|^2\right)$ *endowed with a discrete Lyapunov function relative to which computing a descent direction from any* $\sigma \in \mathcal{BT}_J$ *requires a computation of time* $\mathrm{O}\left(|J|\right)$.

D. Portal Transformations

We now turn attention to construction of the crucial portal map that effects the geometric realization of the NNI-graph as required for Problem 3; and herein we extend our recent construction of the realization function, `Port`, in [1] for point particle configurations to thickened disk configurations.

Throughout this section, the trees $\sigma, \tau \in \mathcal{BT}_J$ are NNI-adjacent (as defined in Section III-D) and fixed, and we therefore take the liberty of suppressing all mention of these trees wherever convenient, for the sake of simplifying the presentation of our equations. Since the trees σ, τ are NNI-adjacent, we may apply Lemma 1 from [78] to find common disjoint clusters A, B, C such that $\{A \cup B\} = \mathcal{C}(\sigma) \setminus \mathcal{C}(\tau)$ and $\{B \cup C\} = \mathcal{C}(\tau) \setminus \mathcal{C}(\sigma)$. Note that the triplet $\{A, B, C\}$ of the pair (σ, τ) is unique. We call $\{A, B, C\}$ the *NNI-triplet* of the pair (σ, τ). Since σ and τ are fixed throughout this section, so will be A, B, C and $P := A \cup B \cup C$.

In the construction of the portal map, `Port` (33), we restrict our attention to the portal configurations with a certain symmetry property, defined as:

Definition 3 ([1]) We call $\mathbf{x} \in \left(\mathbb{R}^d\right)^J$ a *symmetric* configuration associated with (σ, τ) if centroids of partial configurations $\mathbf{x}|A$, $\mathbf{x}|B$ and $\mathbf{x}|C$ form an equilateral triangle, as illustrated in Fig. 10. The set of all symmetric configurations with respect to (σ, τ) is denoted $\mathtt{Sym}(\sigma, \tau)$.

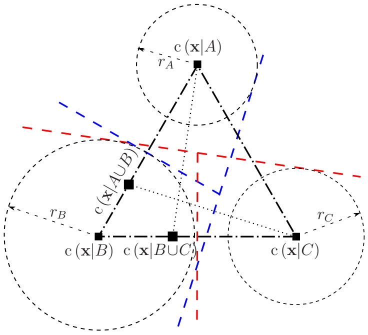

$\mathrm{c}(\mathbf{x}|A)$
r_A
$\mathrm{c}(\mathbf{x}|A \cup B)$
r_B
r_C
$\mathrm{c}(\mathbf{x}|B)$
$\mathrm{c}(\mathbf{x}|B \cup C)$
$\mathrm{c}(\mathbf{x}|C)$

Fig. 10. An illustration of a symmetric configuration $\mathbf{x} \in \mathtt{Sym}(\sigma, \tau)$, where the consensus ball $B_Q(\mathbf{x})$ of partial configuration of cluster $Q \in \{A, B, C\}$ has a positive radius.

An important property of the symmetric configurations is:

Lemma 1 ([1]) *Let* $\mathbf{x} \in \mathfrak{S}(\sigma)$ *be a symmetric configuration in* $\mathtt{Sym}(\sigma, \tau)$*. If each partial configuration* $\mathbf{x}|Q$ *of cluster* $Q \in \{A, B, C\}$ *is contained in the associated "consensus" ball* $B_Q(\mathbf{x})$ *— an open ball*[23] *centered at* $\mathrm{c}(\mathbf{x}|Q)$ *with radius*

$$r_Q(\mathbf{x}) := \min_{\substack{\gamma \in (\sigma,\tau) \\ D \in \{Q, \mathrm{Pr}(Q,\gamma)\} \setminus \{P\}}} -\left(\mathrm{c}(\mathbf{x}|Q) - \mathrm{m}_{D,\gamma}(\mathbf{x})\right)^{\mathrm{T}} \frac{\mathrm{s}_{D,\gamma}(\mathbf{x})}{\left\|\mathrm{s}_{D,\gamma}(\mathbf{x})\right\|}, \tag{29}$$

then $\mathbf{x}$ *also supports* τ*, i.e.* $\mathbf{x} \in \mathfrak{S}(\tau)$*, and so* $\mathbf{x}$ *is a portal configuration,* $\mathbf{x} \in \mathtt{Portal}(\sigma, \tau)$*.*

Note that for any configuration $\mathbf{x} \in \mathtt{Sym}(\sigma, \tau)$ the consensus ball of each partial configuration of cluster $Q \in \{A, B, C\}$ has a nonempty interior, i.e. $r_Q(\mathbf{x}) > 0$ [1] — see Fig. 10.

In the following, we first describe how we relate any given triangle to an equilateral triangle using Napoleon transformations, and then define our portal map.

1) Napoleon Triangles: We recall a theorem of geometry describing how to create an equilateral triangle from an arbitrary triangle: construct, either all outer or all inner, equilateral triangles at the sides of a triangle in the plane containing the triangle, and so centroids of the constructed equilateral triangles form another equilateral triangle in the same plane, known as the "*Napoleon triangle*" [88] — see Fig. 11. We will refer to this construction as the Napoleon transformation, and we find it convenient to define the *double outer Napoleon triangle* as the equilateral triangle resulting from two concatenated outer Napoleon transformations of a triangle. Let $\mathtt{NT} : \mathbb{R}^{3d} \to \mathbb{R}^{3d}$ denote the double outer Napoleon transformation, see [89] for an explicit form of `NT`. It is also useful to remark that the double outer Napoleon transformation yields an equilateral triangle optimally aligned with an arbitrary given triangle by virtue of minimizing sum of square distances between the paired vertices [90].

The NNI-triplet $\{A, B, C\}$ defines an associated triangle with distinct vertices for each configuration, $\triangle_{A,B,C} : \mathfrak{S}(\sigma) \to \mathtt{Conf}\left(\mathbb{R}^d, [3], \mathbf{0}\right)$,

$$\triangle_{A,B,C}(\mathbf{x}) := \left[\mathrm{c}(\mathbf{x}|A), \mathrm{c}(\mathbf{x}|B), \mathrm{c}(\mathbf{x}|C)\right]^{\mathrm{T}}. \tag{30}$$

The double outer Napolean tranformation of $\triangle_{A,B,C}(\mathbf{x})$ returns symmetric target locations for $\mathrm{c}(\mathbf{x}|A)$, $\mathrm{c}(\mathbf{x}|B)$ and

[23] In a metric space (X, d), the open ball $B(\mathrm{x}, r)$ centered at $\mathrm{x} \in X$ with radius $r \in \mathbb{R}_{\geq 0}$ is $B(\mathrm{x}, r) = \{\mathrm{y} \in X \mid d(\mathrm{x}, \mathrm{y}) < r\}$.

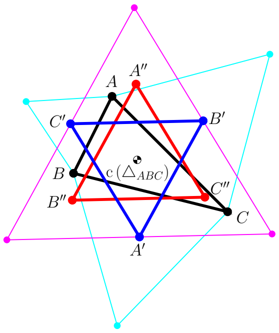


Fig. 11. Outer Napoleon Triangles $\triangle_{A'B'C'}$ and $\triangle_{A''B''C''}$ of $\triangle_{ABC}$ and $\triangle_{A'B'C'}$, respectively, and $\triangle_{A''B''C''}$ is referred to as the double outer triangle of $\triangle_{ABC}$. Note that centroids of all triangles coincides, i.e. $\mathrm{c}(\triangle_{ABC}) = \mathrm{c}(\triangle_{A'B'C'}) = \mathrm{c}(\triangle_{A''B''C''})$.

$\mathrm{c}\left(\mathbf{x}|C\right)$, and the corresponding displacement of $\mathrm{c}\left(\mathbf{x}|P\right)$, denoted $\mathtt{Noff}_{A,B,C} : \mathtt{Conf}\left(\mathbb{R}^d, J, \mathbf{r}\right) \to \mathbb{R}^d$, is given by the formula[24]

$$\mathtt{Noff}_{A,B,C}\left(\mathbf{x}\right) := \mathrm{c}\left(\mathbf{x}|P\right) - \Gamma \cdot \mathtt{NT} \circ \triangle_{A,B,C}\left(\mathbf{x}\right), \quad (31)$$

where $\Gamma := \frac{1}{|P|}\left[|A|, |B|, |C|\right] \otimes \mathbf{I}_d \in \mathbb{R}^{d \times 3d}$, and the vertices of the associated equilateral triangle with compensated offset of $\mathrm{c}\left(\mathbf{x}|P\right)$ are[24]

$$\left[c_A, c_B, c_C\right]^{\mathrm{T}} := \mathtt{NT} \circ \triangle_{A,B,C}\left(\mathbf{x}\right) + \mathbf{1}_3 \otimes \mathtt{Noff}_{A,B,C}\left(\mathbf{x}\right). \quad (32)$$

2) Portal Maps: We now define a portal map, $\mathtt{Port} : \mathfrak{S}(\sigma) \to \mathtt{Portal}\left(\sigma, \tau\right)$, to be

$$\mathtt{Port}\left(\mathbf{x}\right) := \begin{cases} \mathbf{x} & \text{, if } \mathbf{x} \in \mathtt{Portal}\left(\sigma, \tau\right), \\ \left(\mathtt{Mrg} \circ \mathtt{Scl} \circ \mathtt{Ctr}\right)\left(\mathbf{x}\right), & \text{otherwise,} \end{cases} \quad (33)$$

where $\mathtt{Ctr} : \mathfrak{S}(\sigma) \to \mathtt{Sym}\left(\sigma, \tau\right)$ rigidly translates the partial configurations, $\mathbf{x}|A$, $\mathbf{x}|B$ and $\mathbf{x}|C$, to the new centroid locations, c_A, c_B and c_C (32), respectively, yielding a symmetric configuration,

$$\mathtt{Ctr}\left(\mathbf{x}\right) := \begin{cases} \mathrm{x}_i & \text{, if } i \notin P, \\ \mathrm{x}_i - \mathrm{c}\left(\mathbf{x}|Q\right) + \mathrm{c}_Q, & \text{if } i \in Q, Q \in \{A, B, C\}, \end{cases} \quad (34)$$

It is important to observe that $\mathtt{Ctr}$ keeps the barycenter of $\mathbf{x}|P$ fixed, and so separating hyperplanes of the rest of clusters ascending and disjoint with P are kept unchanged.

After obtaining a symmetric configuration in $\mathtt{Sym}\left(\sigma, \tau\right)$, $\mathtt{Scl} : \mathtt{Sym}\left(\sigma, \tau\right) \to \mathtt{Sym}\left(\sigma, \tau\right)$ rigidly translates each partial configuration, $\mathbf{x}|A$, $\mathbf{x}|B$ and $\mathbf{x}|C$, to scale and fit into the corresponding consensus ball so that the new configuration simultaneously support both subtrees of σ and τ rooted at P,

$$\mathtt{Scl}\left(\mathbf{x}\right) := \begin{cases} \mathrm{x}_i, & \text{, if } i \notin P \\ \mathrm{x}_i + \zeta \cdot \left(\mathrm{c}\left(\mathbf{x}|Q\right) - \mathrm{c}\left(\mathbf{x}|P\right)\right), & \text{if } \begin{array}{c} i \in Q, \\ Q \in \{A, B, C\}, \end{array} \end{cases} \quad (35)$$

where $\zeta \in [0, \infty)$ is a scale parameter defined as

$$\zeta := \max_{Q \in \{A,B,C\}} \max\left(\frac{r\left(\mathbf{x}|Q\right) + \alpha}{r_Q\left(\mathbf{x}\right)}, 1\right) - 1. \quad (36)$$

Here, $\alpha > 0$ is a safety margin as used in (20), and $r\left(\mathbf{x}|Q\right)$ (12) denotes the centroidal radius of partial configuration $\mathbf{x}|Q$ and $r_Q\left(\mathbf{x}\right)$ (29) is the radius of its consensus ball. Note that $\mathtt{Scl}$ preserves the configuration symmetry, i.e. centroids $\mathrm{c}\left(\mathbf{x}|A\right)$, $\mathrm{c}\left(\mathbf{x}|B\right)$ and $\mathrm{c}\left(\mathbf{x}|C\right)$ still form an equilateral triangle after the mapping, and lefts the barycenter of $\mathbf{x}|P$ unchanged.

Finally, $\mathtt{Mrg} : \mathtt{Sym}\left(\sigma, \tau\right) \to \mathtt{Sym}\left(\sigma, \tau\right)$ iteratively translates and merges partial configurations of common complementary clusters of σ and τ, in a bottom-up fashion starting at P, to simultaneously support both hierarchies σ and τ,

$$\mathtt{Mrg}\left(\mathbf{x}\right) := \mathtt{Mrg}_P\left(\mathbf{x}\right), \quad (37)$$

where for any $I \in \{P\} \cup \mathrm{Anc}\left(P, \sigma\right)$

$$\mathtt{Mrg}_I\left(\mathbf{x}\right) := \begin{cases} \mathbf{x} & \text{, if } I = J, \\ \left(\mathtt{Mrg}_{\mathrm{Pr}(I,\tau)} \circ \mathtt{Sep}_I\right)\left(\mathbf{x}\right), & \text{otherwise.} \end{cases} \quad (38)$$

Here, $\mathtt{Sep}_I$ separates sibling clusters I and $I^{-\sigma}$ such that the clearance between every agent in $I \cup I^{-\sigma}$ and the associated separating hyperplane is at least α units (i.e. if $\hat{\mathbf{x}} = \mathtt{Sep}_I\left(\mathbf{x}\right)$ for some $\mathbf{x} \in \left(\mathbb{R}^d\right)^J$ with $\mathrm{s}_{I,\sigma}\left(\mathbf{x}\right) \neq 0$, then $\eta_{k,K,\sigma}\left(\hat{\mathbf{x}}\right) \geq r_k + \alpha$ for any $k \in K$, $K \in \{I, I^{-\sigma}\}$): for any $j \in J$

$$\mathtt{Sep}_I\left(\mathbf{x}\right)_j := \begin{cases} \mathbf{x}_j & \text{, if } j \notin \mathrm{Pr}\left(I, \sigma\right), \\ \mathbf{x}_j + 2\lambda \frac{|K^{-\sigma}|}{|\mathrm{Pr}(K,\sigma)|} \frac{\mathrm{s}_{K,\sigma}(\mathbf{x})}{\|\mathrm{s}_{K,\sigma}(\mathbf{x})\|}, & \text{if } \begin{array}{c} j \in K, \\ K \in \{I, I^{-\sigma}\}, \end{array} \end{cases} \quad (39)$$

where the required amount of centroidal separation, $\lambda \in [0, \infty)$, is given by

$$\lambda := \max_{\substack{k \in K \\ K \in \{I, I^{-\sigma}\}}} \max\left(-\left(\eta_{k,K,\sigma}\left(\mathbf{x}\right) - r_k - \alpha\right), 0\right). \quad (40)$$

Note that since $\mathrm{c}\left(\mathbf{x}|P\right) = \mathrm{c}\left(\hat{\mathbf{x}}|P\right)$ for any $\mathbf{x} \in \mathfrak{S}(\sigma)$ and $\hat{\mathbf{x}} = \left(\mathtt{Scl} \circ \mathtt{Ctr}\right)\left(\mathbf{x}\right)$, we always have $\mathrm{s}_{I,\sigma}\left(\hat{\mathbf{x}}\right) \neq 0$ for any $I \in \{P\} \cup \mathrm{Anc}\left(P, \sigma\right)$, which is required for $\mathtt{Sep}_I$ to be well defined. Further, using (39), one can verify that $\mathrm{c}\left(\mathbf{x}|\mathrm{Pr}\left(I, \sigma\right)\right) = \mathrm{c}\left(\hat{\mathbf{x}}|\mathrm{Pr}\left(I, \sigma\right)\right) = \mathrm{c}\left(\tilde{\mathbf{x}}|\mathrm{Pr}\left(I, \sigma\right)\right)$ for $\tilde{\mathbf{x}} = \mathtt{Sep}_I\left(\hat{\mathbf{x}}\right)$, and so $\mathrm{s}_{A,\sigma}\left(\tilde{\mathbf{x}}\right) \neq 0$ for any $A \in \mathrm{Anc}\left(I, \sigma\right)$, which guarantees that recursive calls of $\mathtt{Sep}_I$ in the computation of $\mathtt{Port}$ are always well-defined.

We find it useful to summarize some critical properties of the portal map for the strata of $\mathtt{HC}_{\text{2-means}}$ as:

Theorem 6 ([91]) *The NNI-graph $\mathcal{N}_J = \left(\mathcal{BT}_J, \mathcal{E}_{\mathcal{N}}\right)$ is a subgraph of the $\mathtt{HC}_{\text{2-means}}$ adjacency graph $\mathcal{A}_J = \left(\mathcal{BT}_J, \mathcal{E}_{\mathcal{A}}\right)$, i.e. for any pair $\left(\sigma, \tau\right)$ of NNI-adjacent trees in $\mathcal{BT}_J$, $\mathtt{Portal}\left(\sigma, \tau\right) \neq \varnothing$. Further, given an edge, $\left(\sigma, \tau\right) \in \mathcal{E}_{\mathcal{N}} \subset \mathcal{E}_{\mathcal{A}}$, a geometric realization via the map $\mathtt{Port}_{(\sigma,\tau)} : \mathfrak{S}(\sigma) \to \mathtt{Portal}\left(\sigma, \tau\right)$ (33) can be computed in quadratic, $\mathrm{O}\left(|J|^2\right)$, time with the number of leaves, $|J|$.*

VI. Numerical Simulations

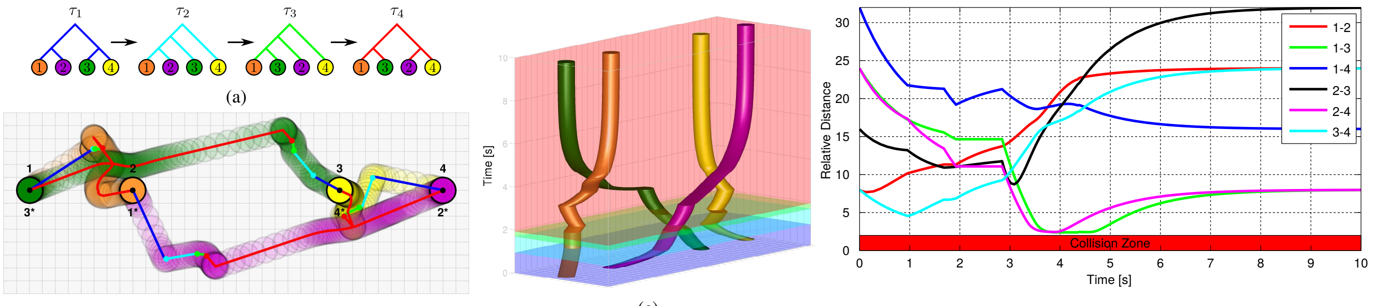
For the sake of clarity, we first illustrate the behavior of the hybrid system defined in Section V for the case of four disks moving in a two-dimensional ambient space.[25]

In order to visualize in this simple setting the most complicated instance of collision-free navigation and observe maximal number of transitions between local controllers, we pick the initial, $\mathbf{x} \in \mathfrak{S}(\tau_1)$, and desired configurations, $\mathbf{x}^* \in \mathfrak{S}(\tau_4)$, where disks are placed almost on the horizontal axis and left-to-right ordering of their labels are $(1, 2, 3, 4)$ and $(3^*, 1^*, 4^*, 2^*)$, respectively, and their corresponding clustering trees are $\tau_1 \in \mathcal{BT}_{[4]}$ and $\tau_4 \in \mathcal{BT}_{[4]}$, see Fig. 12.

The resultant trajectory of each disk following the hybrid navigation planner in Section V, the relative distance between each pair of disks and the sequence of trees associated with visited hierarchical strata are shown in Fig. 12. Here, the disks start following the local controller associated with τ_1 until they enter in finite time the domain of the following local controller associated with τ_2 at $\mathbf{x}_c \in \mathfrak{S}(\tau_1) \cap \mathfrak{S}(\tau_2)$ — shown by cyan dots in Fig. 12. After a finite time navigating in $\mathfrak{S}(\tau_2)$ and $\mathfrak{S}(\tau_3)$, respectively, the group enters the domain of the goal controller $f_{\tau_4,\mathbf{x}^*}$ (Table IV) at $\mathbf{x}_r \in \mathfrak{S}(\tau_3) \cap \mathfrak{S}(\tau_4)$ — shown by red dots in Fig. 12, and $f_{\tau_4,\mathbf{x}^*}$ asymptotically steers the disks

[24] Here, $\mathbf{I}_d$ is the $d \times d$ identity matrix, and $\mathbf{1}_k$ is the $\mathbb{R}^k$ column vector of all ones. Also, $\otimes$ and $\cdot$ denote the Kronecker product and the standard array product, respectively.

[25] For all simulations we consider unit disks moving in an ambient plane, i.e. $r_j = 1$ for all $j \in J$, and we set $\alpha = 0.2$ and $\beta = 1$; and all simulations are obtained through numerical integration of the hybrid dynamics generated by the HNC algorithm (Table III) using the `ode45` function of MATLAB.



(a)

(b)

(c)

(d)

Fig. 12. An illustrative navigation trajectory of the hybrid dynamics generated by the HNC algorithm for 4 disks in a planar ambient space. Disks are placed on the horizontal axis for both the initial and desired configurations in different orders, from left to right $(1, 2, 3, 4)$ and $(3^*, 1^*, 4^*, 2^*)$ at the start and goal, respectively. (a) The sequence of trees associated with deployed local controllers during the execution of the hybrid navigation controller. (b) Centroidal trajectory of each disk colored according the active local controller, where $\mathbf{x}_c \in \mathfrak{S}(\tau_1) \cap \mathfrak{S}(\tau_2)$, $\mathbf{x}_g \in \mathfrak{S}(\tau_2) \cap \mathfrak{S}(\tau_3)$ and $\mathbf{x}_r \in \mathfrak{S}(\tau_3) \cap \mathfrak{S}(\tau_4)$ shown by cyan, green and red dots, respectively, are portal configurations. (c) Space-time curve of disks (d) Pairwise distances between disks.

to the goal configuration $\mathbf{x}^* \in \mathfrak{S}(\tau_4)$. Finally, note that the total number of binary trees over four leaves is 15; however, our navigation planner reactively deploys only 4 of them.

We now consider a similar, but slightly more complicated setting: a group of six disks in a plane where agents are initially placed evenly almost on the horizontal axes and switch their positions at the destination as shown in Fig. 13(a), which is also used in [20] as an example of complicated multi-agent arrangements. While steering the disks towards the goal, the hybrid navigation planner automatically deploys only 6 local controllers out of the family of 945 local controllers. The time evolution of the disk is illustrated in Fig. 13(a).

Moreover, to demonstrate the efficiency of the deployment policy of our hybrid planner, we separately consider groups of 8 and 16 disks in an ambient plane, illustrated in Fig. 13. The eight disks are initially located at the corner of two squares whose centroids coincide and the perimeter of one is twice of the perimeter of the other. At the destination, disks switch their locations as illustrated in Fig. 13(b). For sixteen disk case, disks are initially placed at the vertices of a 4 by 4 grid, and their task is to switch their location as illustrated in Fig. 13(c). Although there are a large number of local controllers for the case of groups of 8 and 16 disks ($\left|\mathcal{BT}_{[8]}\right| > 10^5$ and $\left|\mathcal{BT}_{[16]}\right| > 6 \times 10^{15}$), our hybrid navigation planner only deploys 9 and 19 local controllers, respectively.

The number of potentially available local controllers for a group of n disks (26) grows super exponentially with n. On the other hand, if agents have perfect sensing and actuation modelled as in this paper, our hybrid navigation planner automatically deploys at most $\frac{1}{2}(n-1)(n-2)$ local controllers [78], illustrating its computational efficiency.

Finally, although the HNC algorithm in Section V is primarily constructed based on the topological characterization of the associated hierarchical strata and does not ensure the optimality of its resulting navigation paths, we still find it useful to include a brief statistical analysis of the metric properties of its navigation paths. Since the geodesic distance (i.e. the shortest path length) between any pair of multirobot configurations is very difficult to compute in practice, as done in [22], [30], in order to quantify navigation paths we consider the *normalized navigation path length*, Γ, defined as the ratio of the total navigation distance travelled by all robots to the straight-line Euclidean distance between any initial and goal configurations [30],

$$\Gamma := \frac{\sum_{i=1}^{n} \int_0^\infty \left\| \dot{\mathrm{x}}_i(t) \right\| \mathrm{d}t}{\sum_{i=1}^{n} \left\| \mathrm{x}_i(0) - \mathrm{x}_i^* \right\|}, \qquad (41)$$

where $\mathbf{x}(t)$ is the time trajectory of the navigation path of the HNC algorithm asymptotically joining the initial configuration $\mathbf{x}(0)$ to the goal configuration $\mathbf{x}^* = \lim_{t\to\infty} \mathbf{x}(t)$. Further, to ensure an unbiased selection of initial and goal configurations, we consider unit disk configurations (i.e., $r_i = 1$ for all $i = 1 \ldots n$) uniformly distributed in a square region of edge length $2k \sum_{i=1}^{n} r_i = 2kn$, where the parameter $k > 0$ models how tight disks are packed. In Fig. 14 we present the effect of group size, n, and configuration tightness, k, on the normalized navigation path length, Γ. [26] As expected, the normalized navigation path length increases with increasing configuration tightness and group size in average, since the closer the disks are packed and the greater they are in number, the more difficult they navigate to their destination. We also observe that the average normalized path length of the HNC algorithm has the same order of magnitude as those of other available navigation function based algorithms [22], [30] whose convergence and path properties significantly depend on parameter tuning.

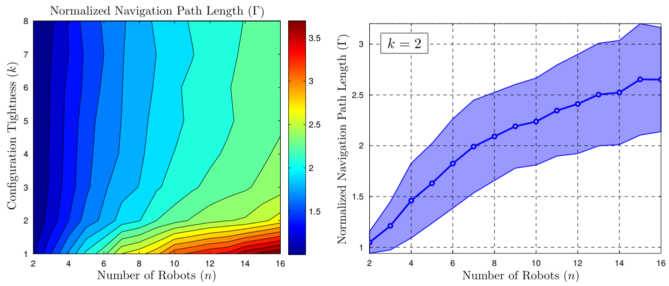


Fig. 14. (left) Average normalized navigation path length versus group size, n, and configuration tightness, k. (right) Mean and standard deviation of the normalized navigation path length for configuration tightness $k = 2$.

VII. Conclusion

In this paper, we introduce a novel application of clustering to the problem of coordinated robot navigation. The notion of

[26] Each data point in Fig. 14 is obtained using 500 pairs of uniformly sampled random initial and goal configurations.

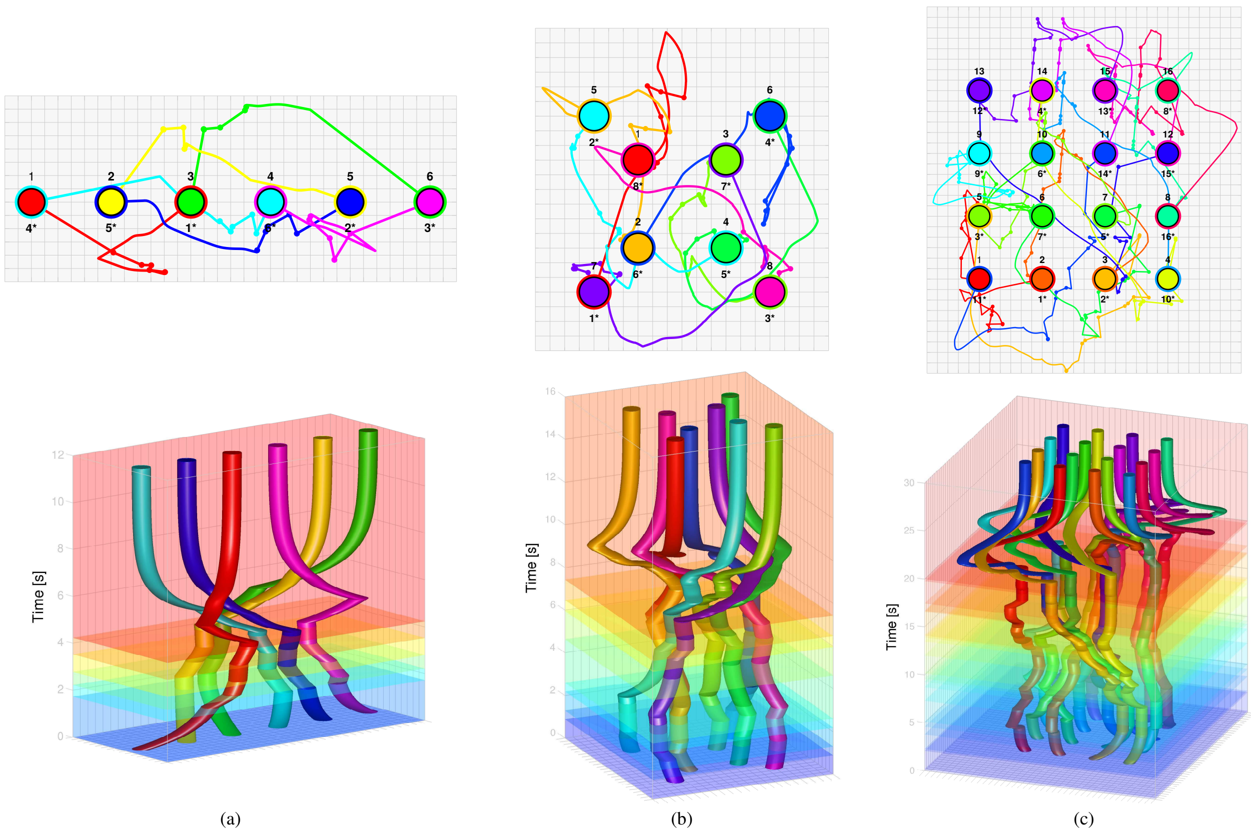


Fig. 13. Example trajectories of the hybrid vector field planner for (a) 6, (b) 8 and (c) 16 disks in a planar ambient space. (top) Trajectory and (bottom) state-time curve of each disk. Each colored time interval demonstrates the execution duration of an activated local controller. Dots correspond to the portal configurations where transitions between local controllers occur at.

hierarchical clustering offers a natural abstraction for ensemble task encoding and control in terms of precise yet flexible organizational specifications at different resolutions. Based on this new abstraction, we propose a provably correct generic hierarchical navigation framework for collision-free motion design towards any given destination via a sequence of hierarchy preserving controllers. For 2-means divisive hierarchical clustering [72], based on a topological characterization of the underlying space, we present a centralized online (completely reactive) and computationally efficient instance of our hierarchical navigation framework for disk-shaped robots, which generalizes to an arbitrary number of disks and ambient space dimension.

Work now in progress targets more practical settings in the field of robotics including navigating around obstacles in compact spaces and a distributed implementation of our navigation framework. We are also exploring a number of application settings for hierarchical formation specification and control including problems of perception, perceptual servoing, anomaly detection and automated exploration and various problems of multi-agent coordination.

In the longer term, especially when the scalability and efficiency of hierarchical protocols in sensor networks for information routing and aggregation is of concern [92], these methods suggest a promising unifying framework to simultaneously handle control, communication and information aggregation (fusion) in multi-agent systems.

Appendix I
Properties of The Hierarchy Invariant Vector Field

Although the recursive definition of the hierarchy preserving navigation policy, $f_{\tau,\mathbf{y}}$, in Table IV expresses an efficient encoding of intra-cluster and inter-cluster interactions and dependencies of individuals, which we suspect will prove to have value for distributed settings, it yields a discontinuous vector field complicating the qualitative (existence, uniqueness, invariance and stability) analysis. We find it convenient to proceed instead by developing an alternative, equivalent representation of this vector field. Namely, we introduce a family of continuous and piecewise smooth covering vector fields whose application over a partition (derived from their covering domains) of the stratum yields a continuous piecewise smooth flow (identical to that generated by the original construction) which is considerably easier to analyze because it admits an interpretation as a sequential composition [12] over the covering family. Space limitations force a choice between a complete listing of the detailed proofs vs. a more intuitive explanation of how the construction works. We have chosen to err on the latter side, merely stating the main results and omitting all proofs (for which the reader is referred to the extended technical report version [91]) in favor of an informal interpretation of the their meaning.

We find it useful to first observe that the original construction yields a well defined and effectively computable function.

Proposition 3 ([91]) *The recursion in Table IV results in a well defined function, $f_{\tau,\mathbf{y}} : \mathfrak{S}(\tau) \to \left(\mathbb{R}^d\right)^J$, that can be computed for each configuration $\mathbf{x} \in \mathfrak{S}(\tau)$ in $\mathrm{O}\left(|J|^2\right)$ time.*

A. An Equivalent System Model

Key for understanding the hierarchy preserving navigation policy, $f_{\tau,\mathbf{y}}$, in Table IV is the observation that for any $\mathbf{x} \in \mathfrak{S}(\tau)$ the list of visited clusters of τ satisfying base conditions during the recursive computation of $f_{\tau,\mathbf{y}}$ defines a partition $\mathcal{J}$ of J compatible with τ, i.e. $\mathcal{J} \subset \mathcal{C}(\tau)$.[27]

Now observe, depending on which base condition holds (Table IV.2) or Table IV.4)), every block I of partition $\mathcal{J}$, associated with any given $\mathbf{x}\in\mathcal{D}_A(I)\cup\left(\mathfrak{S}(\tau)\setminus\mathcal{D}_H(I)\right)$, can be associated with a binary scalar $\hat{\mathrm{b}}_I(\mathbf{x})\in\{-1,+1\}$ such that[28]

$$\hat{\mathrm{b}}_I(\mathbf{x}) = \begin{cases} +1 & \text{, if } \mathbf{x} \in \mathcal{D}_A(I), \\ -1 & \text{, if } \mathbf{x} \notin \mathcal{D}_A(I) \cup \mathcal{D}_H(I), \end{cases} \tag{42}$$

where $\mathcal{D}_A(I)$ and $\mathcal{D}_H(I)$ are defined as in (15) and (18), respectively. We will use this configuration space labeling scheme to recast the hierarchy preserving control policy $f_{\tau,\mathbf{y}}$ as an online sequential composition of a family of continuous and piecewise smooth local controllers indexed by partitions of J compatible with τ and associated binary vectors as follows.

A partition $\mathcal{J}$ of J is said to be compatible with $\tau \in \mathcal{BT}_J$ if and only if $\mathcal{J} \subset \mathcal{C}(\tau)$, and denote by $\mathcal{P}_J(\tau)$ the set of partitions of J compatible with τ. Accordingly, define $\mathcal{SP}_J(\tau)$ to be the set of substratum policy indices,

$$\mathcal{SP}_J(\tau) := \left\{(\mathcal{J},\mathbf{b}) \,\middle|\, \mathcal{J} \in \mathcal{P}_J(\tau), \mathbf{b} \in \{-1,+1\}^{\mathcal{J}}\right\}. \tag{43}$$

For any partition $\mathcal{J} \in \mathcal{P}_J(\tau)$ of J and $\mathbf{b} := (\mathrm{b}_I)_{I\in\mathcal{J}} \in \{-1,+1\}^{\mathcal{J}}$, the domain $\mathcal{D}(\mathcal{J},\mathbf{b})$ of a local control policy $h_{\mathcal{J},\mathbf{b}}$, presented in Table VI, is defined to be

$$\mathcal{D}(\mathcal{J},\mathbf{b}) := \bigcap_{I\in\mathcal{J}} \Big(\mathcal{D}_B(I,\mathrm{b}_I) \cap \bigcap_{K\in\mathrm{Anc}\,(I,\tau)} \mathcal{D}_H(K)\Big), \tag{44}$$

where the set of configurations satisfying the base condition associated with cluster I of $\mathcal{J}$ and binary scalar b_I is given by

$$\mathcal{D}_B(I,\mathrm{b}_I) := \begin{cases} \mathcal{D}_A(I), & \text{if } \mathrm{b}_I = +1, \\ \mathfrak{S}(\tau)\ , & \text{if } \mathrm{b}_I = -1, \end{cases} \tag{45}$$

and all ancestors $K \in \mathrm{Anc}(I,\tau)$ of I in τ satisfy the recursion condition of having properly separated children clusters described by $\mathcal{D}_H(K)$ (18). Accordingly, let $\mathcal{V}_\tau(\mathcal{J})$ denote the set of clusters of τ visited during the recursive computation of $h_{\mathcal{J},\mathbf{b}}$ in Table VI,

$$\mathcal{V}_\tau(\mathcal{J}) := \left\{K \in \mathcal{C}(\tau) \,\middle|\, K \supseteq I, I \in \mathcal{J}\right\}. \tag{46}$$

Note that $J \in \mathcal{V}_\tau(\mathcal{J})$ since $\mathcal{J}$ is a partition of the root cluster J and any block $I \in \mathcal{J}$ satisfies $I \subseteq J$.

TABLE VI
LOCAL CONTROL POLICIES IN A HIERARCHICAL STRATUM

Let $\mathcal{J}$ be a partition of J with $\mathcal{J} \subset \mathcal{C}(\tau)$, and $\mathbf{b} = (\mathrm{b}_I)_{I\in\mathcal{J}} \in \{-1,+1\}^{\mathcal{J}}$. For any desired $\mathbf{y} \in \mathfrak{S}(\tau)$, supporting $\tau \in \mathcal{BT}_J$, and initial $\mathbf{x} \in \mathcal{D}(\mathcal{J},\mathbf{b})$(44), the local control policy, $h_{\mathcal{J},\mathbf{b}} : \mathcal{D}(\mathcal{J},\mathbf{b}) \to \left(\mathbb{R}^d\right)^J$,

$$h_{\mathcal{J},\mathbf{b}}(\mathbf{x}) := \hat{h}_{\mathcal{J},\mathbf{b}}(\mathbf{x},\mathbf{0},J),$$

is recursively computed starting at the root cluster J with the zero control input $\mathbf{0} \in \left(\mathbb{R}^d\right)^J$ as follows: for any $\mathbf{u} \in \left(\mathbb{R}^d\right)^J$ and $I \in \mathcal{V}_\tau(\mathcal{J})$ (46),

```
function $\hat{\mathbf{u}} = \hat{h}_{\mathcal{J},\mathbf{b}}(\mathbf{x},\mathbf{u},I)$
Base Cases
    if $I \in \mathcal{J}$,
        if $\mathrm{b}_I = +1$
            $\hat{\mathbf{u}} \leftarrow f_A(\mathbf{x},\mathbf{u},I)$ (14),            % Attracting Field
        else
            $\hat{\mathbf{u}} \leftarrow f_S(\mathbf{x},\mathbf{u},I)$ (24),            % Split Separation Field
        end
Recursion
    else
        $\{I_L, I_R\} \leftarrow \mathrm{Ch}(I,\tau)$,
        $\hat{\mathbf{u}}_L \leftarrow \hat{h}_{\mathcal{J},\mathbf{b}}(\mathbf{x},\mathbf{u},I_L)$,          % Recursion for Left Child
        $\hat{\mathbf{u}}_R \leftarrow \hat{h}_{\mathcal{J},\mathbf{b}}(\mathbf{x},\hat{\mathbf{u}}_L,I_R)$,         % Recursion for Right Child
        $\hat{\mathbf{u}} \leftarrow f_H(\mathbf{x},\hat{\mathbf{u}}_R,I)$ (19),          % Split Preserving Field
    end
return $\hat{\mathbf{u}}$
```

Observe that each local control policy $h_{\mathcal{J},\mathbf{b}}$ is a recursive composition of continuous functions of $\mathbf{x}$, so it is continuous:

Proposition 4 ([91]) *The recursion in Table VI defines a continuous and piecewise smooth function,*[29] $h_{\mathcal{J},\mathbf{b}} : \mathfrak{S}(\tau) \to \left(\mathbb{R}^d\right)^J$.

To conclude our introduction of the family of covering fields in Table VI, we now observe that the vector field $f_{\tau,\mathbf{y}}$ in Table IV is an online concatenation of continuous local controllers, $h_{\mathcal{J},\mathbf{b}}$, of Table VI using a policy selection method described in Table VII, summarized as:

TABLE VII
POLICY SELECTION ALGORITHM

For any initial $\mathbf{x} \in \mathfrak{S}(\tau)$ and desired $\mathbf{y} \in \mathfrak{S}(\tau)$, supporting $\tau \in \mathcal{BT}_J$, the policy selection algorithm, $p : \mathfrak{S}(\tau) \to \mathcal{SP}_J(\tau)$,

$$p(\mathbf{x}) := \hat{p}(\mathbf{x},J),$$

recursively generates a local policy index in $\mathcal{SP}_J(\tau)$ (43) starting at the root cluster J as follows: for any $I \in \mathcal{C}(\tau)$,

```
function $(\hat{\mathcal{J}},\hat{\mathbf{b}}) = \hat{p}(\mathbf{x},I)$
Base Cases
    if $\mathbf{x} \in \mathcal{D}_A(I)$ (15),
        $\hat{\mathcal{J}} \leftarrow \{I\}$,
        $\hat{\mathbf{b}} \leftarrow +1$,
    else if $\mathbf{x} \notin \mathcal{D}_H(I)$ (18),
        $\hat{\mathcal{J}} \leftarrow \{I\}$,
        $\hat{\mathbf{b}} \leftarrow -1$,
Recursion
    else
        $\{I_L, I_R\} \leftarrow \mathrm{Ch}(I,\tau)$,
        $(\hat{\mathcal{J}}_L,\hat{\mathbf{b}}_L) \leftarrow \hat{p}(\mathbf{x},I_L)$,
        $(\hat{\mathcal{J}}_R,\hat{\mathbf{b}}_R) \leftarrow \hat{p}(\mathbf{x},I_R)$,
        $\hat{\mathcal{J}} \leftarrow \hat{\mathcal{J}}_L \cup \hat{\mathcal{J}}_R$,
        $\hat{\mathbf{b}} \leftarrow \hat{\mathbf{b}}_L \| \hat{\mathbf{b}}_R$, [30]
    end
return $(\hat{\mathcal{J}},\hat{\mathbf{b}})$
```

[27] Note that the recursions in Table IV and Table VII have the same base and recursion conditions, and the recursion in Table VII returns the list of clusters satisfying base conditions, which defines a partition of J (Proposition 4). Hence, using the relation between these recursions in Proposition 6, one can conclude this observation.

[28] Observe from Table IV that any $\mathbf{x} \in \mathfrak{S}(\tau)$ satisfies a base condition (Table IV.2) or Table IV.4)) at cluster $I \in \mathcal{C}(\tau)$ if $\mathbf{x} \in \mathcal{D}_A(I) \cup \left(\mathfrak{S}(\tau)\setminus\mathcal{D}_H(I)\right)$. Note that $\mathcal{D}_A(I) \cup \left(\mathfrak{S}(\tau)\setminus\mathcal{D}_H(I)\right) = \mathcal{D}_A(I) \cup \left(\mathfrak{S}(\tau)\setminus\left(\mathcal{D}_A(I)\cup\mathcal{D}_H(I)\right)\right)$, and $\mathcal{D}_A(I)$ and $\mathfrak{S}(\tau)\setminus\left(\mathcal{D}_A(I)\cup\mathcal{D}_H(I)\right)$ are disjoint.

[29] Note that if $f : U \to \mathbb{R}^m$ is continuous and piecewise smooth on an open set $U \subset \mathbb{R}^n$, then it is locally Lipschitz on U [93].

[30] Here, $\mathbf{p}\|\mathbf{q}$ denotes the concatenation of vectors $\mathbf{p}$ and $\mathbf{q}$. That is to say, let X, Y be two sets and A, B be two finite sets of coordinate indices, then for any $\mathbf{p} \in X^A$ and $\mathbf{q} \in Y^B$ we say $\mathbf{r} \in X^A \times Y^B$ is the concatenation of $\mathbf{p}$ and $\mathbf{q}$, denoted by $\mathbf{r} = \mathbf{p}\|\mathbf{q}$, if and only if $\mathrm{r}_a = \mathrm{p}_a$ and $\mathrm{r}_b = \mathrm{q}_b$ for all $a \in A$ and $b \in B$.

Proposition 5 ([91]) *For any given configuration $\mathbf{x} \in \mathfrak{S}(\tau)$ the policy selection algorithm p in Table VII always returns a valid policy index, $(\mathcal{J}, \mathbf{b}) = p(\mathbf{x})$, in $\mathcal{SP}_J(\tau)$ (43) such that the domain $\mathcal{D}(\mathcal{J}, \mathbf{b})$ (44) of the associated local control policy $h_{\mathcal{J},\mathbf{b}}$ (Table VI) contains $\mathbf{x}$, i.e. $\mathbf{x} \in (\mathcal{D} \circ p)(\mathbf{x})$.*

Proposition 6 ([91]) *For any given $\mathbf{x} \in \mathfrak{S}(\tau)$, the vector field $f_{\tau,\mathbf{x}}$ (Table IV) and the local control policy $h_{\mathcal{J},\mathbf{b}}(\mathbf{x})$ (Table VI) selected as $(\mathcal{J}, \mathbf{b}) = p(\mathbf{x})$ (Table VII) generate the same control (velocity) inputs, i.e. $f_{\tau,\mathbf{y}}(\mathbf{x}) = h_{p(\mathbf{x})}(\mathbf{x})$.*

Since the vector field $f_{\tau,\mathbf{y}}$ is defined for entire $\mathfrak{S}(\tau)$, it is useful to remark that the domains, $\mathcal{D}(\mathcal{J}, \mathbf{b})$ (44), of substratum policies, $h_{\mathcal{J},\mathbf{b}}$, define a cover of $\mathfrak{S}(\tau)$ indexed by partitions of J compatible with τ and associated binary vectors.

B. Online Sequential Composition of Substratum Policies

We now briefly describe the logic behind online sequential composition [12] of substratum policies.

To characterize our policy selection strategy, we first define a priority measure[31] for each local controller $h_{\mathcal{J},\mathbf{b}}$ associated with a partition $\mathcal{J} \in \mathcal{P}_J(\tau)$ of J and $\mathbf{b} \in \{-1, +1\}^{\mathcal{J}}$ to be

$$\texttt{priority}(\mathcal{J}, \mathbf{b}) := \sum_{I \in \mathcal{J}} \mathrm{b}_I \, |I|^2 . \tag{47}$$

Note that the maximum and minimum of the priority measure is attained at the coarsest partition $\{J\}$ of J, and $\mathrm{b}_J = +1$ and $\mathrm{b}_J = -1$, respectively, i.e. $\texttt{priority}(\{J\}, +1) = |J|^2$ and $\texttt{priority}(\{J\}, -1) = -|J|^2$. Accordingly, we shall refer to the local control policy with index $(\{J\}, +1)$ as the goal policy since it has the highest priority and asymptotically steers all configurations in its domain $\mathcal{D}(\{J\}, +1)$(44) to $\mathbf{y}$ following the negated gradient of $V(\mathbf{x}) = \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|$, i.e. $h_{\{J\},+1}(\mathbf{x}) = -(\mathbf{x} - \mathbf{y})$ for any $\mathbf{x} \in \mathcal{D}(\{J\}, +1)$. Note that since the root cluster J has no ancestor, i.e. $\mathrm{Anc}(J, \tau) = \varnothing$, by definition (44), $\mathcal{D}(\{J\}, +1) = \mathcal{D}_A(J)$, and $\mathcal{D}_A(J)$ (15) contains the goal configuration $\mathbf{y}$.

We now introduce an abstract connection between local policies for high-level planning:

Definition 4 Let $(\mathcal{J}, \mathbf{b}), (\mathcal{J}', \mathbf{b}') \in \mathcal{SP}_J(\tau)$ be two distinct substratum policy indices. Then $h_{\mathcal{J},\mathbf{b}}$ is said to *prepare* $h_{\mathcal{J}',\mathbf{b}'}$ if and only if all trajectories of $h_{\mathcal{J},\mathbf{b}}$ starting in its domain $\mathcal{D}(\mathcal{J}, \mathbf{b})$, possibly excluding a set of measure zero, reach $\mathcal{D}(\mathcal{J}', \mathbf{b}')$ in finite time.[32]

Accordingly, define the *prepares graph* $\mathcal{PG} = (\mathcal{SP}_J(\tau), \mathcal{E}_{\mathcal{PG}})$ to have vertex set $\mathcal{SP}_J(\tau)$(43) with a policy index $(\mathcal{J}, \mathbf{b}) \in \mathcal{SP}_J(\tau)$ connected to another policy index $(\mathcal{J}', \mathbf{b}')$ by a directed edge in $\mathcal{E}_{\mathcal{PG}}$ if and only if $h_{\mathcal{J},\mathbf{b}}$ prepares $h_{\mathcal{J}',\mathbf{b}'}$.

Although, the prepares graph $\mathcal{PG}$ is the most critical component of the sequential composition framework [12] defining a discrete abstraction of continuous control policies, the exponentially growing cardinality of substratum policies [91] and the lack of an explicit characterization of globally asymptotically stable configurations of substratum policies make it usually difficult to compute the complete prepares graph.

Alternatively, we introduce a computationally efficient and recursively constructed graph of substratum policies that is nicely compatible with our needs, yielding a subgraph of the prepares graph, where every policy index is connected to the goal policy index $(\{J\}, +1)$ through a directed path:

Definition 5 Let $\widehat{\mathcal{PG}} = (\mathcal{SP}_J(\tau), \widehat{\mathcal{E}}_{\mathcal{PG}})$ be a graph with vertex list $\mathcal{SP}_J(\tau)$, and a policy index $(\mathcal{J}, \mathbf{b}) \in \mathcal{SP}_J(\tau)$ that is connected to another policy index $(\mathcal{J}', \mathbf{b}') \in \mathcal{SP}_J(\tau)$ by a directed edge in $\widehat{\mathcal{E}}_{\mathcal{PG}}$ if and only if at least one of the following properties holds:[33]

(i) (Complement) There exists a singleton cluster $I \in \mathcal{J}$ such that $\mathrm{b}_I = -1$, and $\mathcal{J}' = \mathcal{J}$ and $\mathbf{b}' \in \{-1, +1\}^{\mathcal{J}'}$ with $\mathrm{b}'_I = +1$ and $\mathrm{b}'_D = \mathrm{b}_D$ for all $D \in \mathcal{J} \setminus \{I\}$.

(ii) (Split) There exists a nonsingleton cluster $I \in \mathcal{J}$ such that $\mathrm{b}_I = -1$, and $\mathcal{J}' = \mathcal{J} \setminus \{I\} \cup \mathrm{Ch}(I, \tau)$ and $\mathbf{b}' \in \{-1, +1\}^{\mathcal{J}'}$ with $\mathrm{b}'_K = -1$ for all $K \in \mathrm{Ch}(I, \tau)$ and $\mathrm{b}'_D = \mathrm{b}_D$ for all $D \in \mathcal{J} \setminus \mathrm{Ch}(I, \tau)$.

(iii) (Merge) There exists a nonsingleton cluster $I \in \mathcal{C}(\tau)$ such that $\mathrm{Ch}(I, \tau) \subset \mathcal{J}$ and $\mathrm{b}_K = +1$ for all $K \in \mathrm{Ch}(I, \tau)$, and $\mathcal{J}' = \mathcal{J} \setminus \mathrm{Ch}(I, \tau) \cup \{I\}$ and $\mathbf{b}' \in \{-1, +1\}^{\mathcal{J}'}$ with $\mathrm{b}'_I = +1$ and $\mathrm{b}'_D = \mathrm{b}_D$ for all $D \in \mathcal{J} \setminus \mathrm{Ch}(I, \tau)$.

We summarize some important properties of $\widehat{\mathcal{PG}}$ as follows:

Proposition 7 ([91]) *The graph $\widehat{\mathcal{PG}} = (\mathcal{SP}_J(\tau), \widehat{\mathcal{E}}_{\mathcal{PG}})$, as defined in Definition 5, is an acyclic subgraph of the prepares graph $\mathcal{PG} = (\mathcal{SP}_J(\tau), \mathcal{E}_{\mathcal{PG}})$ (Definition 4) such that all policy indices in $\mathcal{SP}_J(\tau)$ are connected to the goal policy index $(\{J\}, +1)$ through directed paths in $\widehat{\mathcal{E}}_{\mathcal{PG}}$, of length at most $\mathrm{O}\left(|J|^2\right)$ hops, along which* `priority` *(47) is strictly increasing, i.e. for any $\left((\mathcal{J}, \mathbf{b}), (\mathcal{J}', \mathbf{b}')\right) \in \widehat{\mathcal{E}}_{\mathcal{PG}}$*

$$\texttt{priority}(\mathcal{J}', \mathbf{b}') > \texttt{priority}(\mathcal{J}, \mathbf{b}) . \tag{48}$$

Although a given local policy can prepare more than one potential successor (i.e. higher priority), our policy selection method chooses the one with the strictly highest priority:

Proposition 8 ([91]) *For any given $\mathbf{x} \in \mathfrak{S}(\tau)$ the policy selection method, p, in Table VII always returns the index of a local controller with the maximum priority among all local controllers whose domain contains $\mathbf{x}$,*

$$p(\mathbf{x}) = \underset{\substack{(\mathcal{J}', \mathbf{b}') \in \mathcal{SP}_J(\tau) \\ \mathbf{x} \in \mathcal{D}(\mathcal{J}', \mathbf{b}')}}{\arg\max} \; \texttt{priority}(\mathcal{J}', \mathbf{b}'). \tag{49}$$

and all the other available local controllers have strictly lower priorities.

C. Qualitative Properties of Substratum Policies

We now list important qualitative properties of the substratum control policies of Table VI. Let $\mathcal{J}$ be a partition of J compatible with τ, i.e. $\mathcal{J} \subset \mathcal{C}(\tau)$, and $\mathbf{b} \in \{-1, 1\}^{\mathcal{J}}$.

[31] In the past literature, such a priority assignment of local controllers is done using backchaining of the prepares graph in an offline manner [12].

[32] Here, we slightly relax the original definition of the prepares relation in [12] by not requiring the knowledge of goal sets, globally asymptotically stable states, of local control policies in advance.

[33] One may think of these conditions as restructuring operations of policy indices by merging/splitting of partition blocks and/or alternating binary index values, like NNI moves of trees in Section III-D.

Proposition 9 ([91]) *The domain, $\mathcal{D}(\mathcal{J}, \mathbf{b})$ (44), of a substratum policy, $h_{\mathcal{J},\mathbf{b}}$ (Table VI), is positive invariant.*

Proposition 10 ([91]) *(Substratum Existence and Uniqueness) The vector field $h_{\mathcal{J},\mathbf{b}}$ (Table VI) is locally Lipschitz in $\mathfrak{S}(\tau)$; and for any initial $\mathbf{x} \in \mathcal{D}(\mathcal{J}, \mathbf{b}) \subset \mathfrak{S}(\tau)$ there always exists a compact (bounded and closed) subset W of $\mathcal{D}(\mathcal{J}, \mathbf{b})$ (44) such that all trajectories of $h_{\mathcal{J},\mathbf{b}}$ starting at $\mathbf{x}$ remain in W for all future time. Thus, there is a unique continuous and piecewise smooth flow of $h_{\mathcal{J},\mathbf{b}}$ in $\mathcal{D}(\mathcal{J}, \mathbf{b})$ that is defined for all future time.*

Proposition 11 ([91]) *(Finite Time Prepares Relation) Each local control policy, $h_{\mathcal{J},\mathbf{b}}$, with the exception of the goal controller $h_{\{J\},+1}$, steers (almost) all configurations in its domain, $\mathcal{D}(\mathcal{J}, \mathbf{b})$, to the domain, $\mathcal{D}(\mathcal{J}', \mathbf{b}')$, of another local controller, $h_{\mathcal{J}',\mathbf{b}'}$, at a higher* `priority` *(47) in finite time.*

Proposition 12 ([91]) *(Nonzero Execution Time) Let $\mathbf{x}^t$ be a trajectory of the local control policy $h_{\mathcal{J},\mathbf{b}}$ starting at $\mathbf{x}^0 \in \mathcal{D}(\mathcal{J}, \mathbf{b})$ such that $p\left(\mathbf{x}^0\right) = (\mathcal{J}, \mathbf{b})$. Then the local controller is guaranteed to steers the group for a nonzero time until reaching the domain of a local controller at a higher* `priority` *(47), i.e.*

$$\inf_{t} \left\{ t \geq 0 \middle| p\left(\mathbf{x}^t\right) \neq (\mathcal{J}, \mathbf{b}) \right\} > 0. \tag{50}$$

D. Qualitative Properties of Stratum Policies

A list of important qualitative properties of the hierarchy preserving navigation policy of Table IV are:

Proposition 13 ([91]) *The stratum $\mathfrak{S}(\tau)$ is positive invariant under the hierarchy-invariant control policy, $f_{\tau,\mathbf{y}}$ (Table IV).*

Proof Recall that the domains, $\mathcal{D}$ (44), of local control policies in Table VI define a cover of $\mathfrak{S}(\tau)$ (Proposition 5) each of whose elements is positively invariant under the flow of the associated local policy (Proposition 9). Thus, the result follows since the hierarchy preserving vector field $f_{\tau,\mathbf{y}}$ is equivalent to online sequential composition of local control policies of Table VI based on the policy selection algorithm in Table VII (Proposition 6). ■

Proposition 14 ([91]) *(Stratum Existence and Uniqueness) The hierarchy invariance control policy, $f_{\tau,\mathbf{y}}$ (Table IV), has a unique, continuous and piecewise smooth flow, φ^t, in $\mathfrak{S}(\tau)$, defined for all $t \geq 0$.*

Proof Recall from Proposition 6 that $f_{\tau,\mathbf{y}}$ is equivalent to online sequential composition of a family of substratum policies which have unique, continuous and piecewise smooth flows, defined for all $t \geq 0$, in their positive invariant domains (Proposition 10). Since their domains define a finite closed cover of $\mathfrak{S}(\tau)$ (Proposition 5), the unique, continuous and piecewise flow of $f_{\tau,\mathbf{y}}$ is constructed by piecing together trajectories of these substratum policies. ■

Proposition 15 ([91]) *Any $\mathbf{y} \in \mathfrak{S}(\tau)$ is an asymptotically stable equilibrium point of the hierarchy-invariant control policy, $f_{\tau,\mathbf{y}}$ (Table IV), whose basin of attraction includes $\mathfrak{S}(\tau)$, except a set of measure zero.*

Proof Using the equivalence (Proposition 6) of the hierarchy preserving field $f_{\tau,\mathbf{y}}$ and the sequential composition of substratum control policies of Table VI based on the policy selection method in Table VII, the result can be obtained as follows.

Since `priority` (47) is an integer-valued function with bounded range $\left[-|J|^2, |J|^2\right]$, using Proposition 8 and Proposition 11, one can conclude that the disks starting at almost any configuration in $\mathfrak{S}(\tau)$ reach the domain $\mathcal{D}(\{J\}, +1)$ of the goal policy $h_{\{J,+1\}}$ in finite time after visiting at most $\mathrm{O}\left(|J|^2\right)$ of other local control policies. Note that $\mathbf{y} \in \mathcal{D}(\{J\}, +1)$. Then, the goal policy $h_{\{J\},+1}$,

$$h_{\{J\},+1}(\mathbf{x}) = -\nabla \tfrac{1}{2} \left\|\mathbf{x} - \mathbf{y}\right\|_2^2 = -(\mathbf{x} - \mathbf{y}), \tag{51}$$

asymptotically steers all configuration in $\mathcal{D}(\{J\}, +1)$ to $\mathbf{y}$ while keeping its domain of attraction $\mathcal{D}_A(J)$ positively invariant (Proposition 9), which completes the proof ■

Acknowledgment

The authors would like to thank Yuliy Baryshnikov and Fred Cohen for discussions on the topology of configuration spaces.

References

[1] O. Arslan, D. Guralnik, and D. E. Koditschek, "Navigation of distinct euclidean particles via hierarchical clustering," in *Algorithmic Foundations of Robotics XI*, ser. Springer Tracts in Advanced Robotics, 2015, vol. 107, pp. 19–36.

[2] L. E. Parker, "Multiple mobile robot systems," in *Springer Handbook of Robotics*, 2008, pp. 921–941.

[3] A. Jadbabaie, J. Lin, and A. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *Automatic Control, IEEE Transactions on*, vol. 48, no. 6, pp. 988 – 1001, 2003.

[4] B. Nabet, N. E. Leonard, I. D. Couzin, and S. A. Levin, "Dynamics of decision making in animal group motion," *Journal of Nonlinear Science*, vol. 19, no. 4, pp. 399–435, 2009.

[5] B. Chazelle, "The convergence of bird flocking," *Journal of the ACM (JACM)*, vol. 61, no. 4, p. 21, 2014.

[6] A. Okubo and S. A. Levin, *Diffusion and ecological problems: modern perspectives*. Springer Science & Business Media, 2001.

[7] C. Anderson and N. R. Franks, "Teams in animal societies," *Behavioral Ecology*, vol. 12, no. 5, pp. 534–540, 2001.

[8] A. K. Jain and R. C. Dubes, *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[9] A. Back, J. Guckenheimer, and M. Myers, "A dynamical simulation facility for hybrid systems," in *Hybrid Systems*, ser. Lecture Notes in Computer Science. Springer, 1993, vol. 736, pp. 255–267.

[10] C. Belta, V. Isler, and G. Pappas, "Discrete abstractions for robot motion planning and control in polygonal environments," *Robotics, IEEE Transactions on*, vol. 21, no. 5, pp. 864–874, 2005.

[11] A. Hatcher, *Algebraic topology*. Cambridge Univ. Press, 2002.

[12] R. R. Burridge, A. A. Rizzi, and D. E. Koditschek, "Sequential composition of dynamically dexterous robot behaviors," *The International Journal of Robotics Research*, vol. 18, no. 6, pp. 534–555, 1999.

[13] O. Arslan and D. E. Koditschek, "Anytime hierarchical clustering," *arXiv:1404.3439 [cs, stat]*, 2014. [Online]. Available: arxiv.org/abs/1404.3439

[14] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[15] H. Choset, K. M. Lynch, S. Hutchinson, G. A. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, 2005.

[16] J. E. Hopcroft, J. T. Schwartz, and M. Sharir, "On the complexity of motion planning for multiple independent objects; pspace-hardness of the "warehouseman's problem"," *The International Journal of Robotics Research*, vol. 3, no. 4, pp. 76–88, 1984.

[17] P. Spirakis and C. K. Yap, "Strong np-hardness of moving many discs," *Information Processing Letters*, vol. 19, no. 1, pp. 55–59, 1984.

[18] Y. Baryshnikov, P. Bubenik, and M. Kahle, “Min-type Morse theory for configuration spaces of hard spheres,” *International Mathematics Research Notices*, vol. 2014, no. 9, pp. 2577–2592, 2014.

[19] L. E. Parker, “Path planning and motion coordination in multiple mobile robot teams,” in *Encyclopedia of Complexity and Systems Science*. Springer New York, 2009, pp. 5783–5800.

[20] H. Tanner and A. Boddu, “Multiagent navigation functions revisited,” *Robotics, IEEE Transactions on*, vol. 28, no. 6, pp. 1346–1359, 2012.

[21] D. V. Dimarogonas, S. G. Loizou, K. J. Kyriakopoulos, and M. M. Zavlanos, “A feedback stabilization and collision avoidance scheme for multiple independent non-point agents,” *Automatica*, vol. 42, no. 2, pp. 229–243, 2006.

[22] C. S. Karagoz, H. I. Bozma, and D. E. Koditschek, “Coordinated navigation of multiple independent disk-shaped robots,” *Robotics, IEEE Transactions on*, vol. 30, no. 6, pp. 1289–1304, 2014.

[23] S. Loizou, “The multi-agent navigation transformation: Tuning-free multi-robot navigation,” in *Proceedings of Robotics: Science and Systems (RSS)*, 2014.

[24] K. Solovey and D. Halperin, “k-color multi-robot motion planning,” *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 82–97, 2014.

[25] A. Adler, M. de Berg, D. Halperin, and K. Solovey, “Efficient multi-robot motion planning for unlabeled discs in simple polygons,” *Automation Science and Engineering, IEEE Transactions on*, vol. 12, no. 4, pp. 1309–1317, 2015.

[26] M. Turpin, N. Michael, and V. Kumar, “Concurrent assignment and planning of trajectories for large teams of interchangeable robots,” in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, 2013, pp. 842–848.

[27] ——, “Trajectory planning and assignment in multirobot systems,” in *Algorithmic Foundations of Robotics X*, ser. Springer Tracts in Advanced Robotics, 2013, vol. 86, pp. 175–190.

[28] M. Kumar, D. Garg, and V. Kumar, “Self-sorting in a swarm of heterogeneous agents,” in *American Control Conference*, 2008, pp. 117–122.

[29] L. L. Whitcomb and D. E. Koditschek, “Automatic assembly planning and control via potential functions,” in *Intelligent Robots and Systems (IROS), Intelligence for Mechanical Systems, 1991 IEEE/RSJ International Workshop on*, 1991, pp. 17–23.

[30] L. L. Whitcomb, D. E. Koditschek, and J. B. D. Cabrera, “Toward the automatic control of robot assembly tasks via potentialfunctions: the case of 2-d sphere assemblies,” in *Robotics and Automation (ICRA), 1992 IEEE International Conference on*, 1992, pp. 2186–2191.

[31] E. Rimon and D. Koditschek, “Exact robot navigation using artificial potential functions,” *Robotics and Automation, IEEE Transactions on*, vol. 8, no. 5, pp. 501–518, 1992.

[32] S. Loizou, “Closed form navigation functions based on harmonic potentials,” in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, 2011, pp. 6361–6366.

[33] O. Arslan, D. P. Guralnik, and D. E. Koditschek, “Hierarchically clustered navigation of distinct euclidean particles,” in *Communication, Control, and Computing, 2012 Allerton Conference on*, 2012, pp. 946–953.

[34] Y.-H. Liu, S. Kuroda, T. Naniwa, H. Noborio, and S. Arimoto, “A practical algorithm for planning collision-free coordinated motion of multiple mobile robots,” in *Robotics and Automation (ICRA), 1989 IEEE International Conference on*, 1989, pp. 1427–1432.

[35] Y.-H. Liu, S. Arimoto, and H. Noborio, “New solid model hsm and its application to interference detection between moving objects,” *Journal of Robotic Systems*, vol. 8, no. 1, pp. 39–54, 1991.

[36] M. Peasgood, C. Clark, and J. McPhee, “A complete and scalable strategy for coordinating multiple robots within roadmaps,” *Robotics, IEEE Transactions on*, vol. 24, no. 2, pp. 283–292, April 2008.

[37] N. Ayanian, V. Kumar, and D. Koditschek, “Synthesis of controllers to create, maintain, and reconfigure robot formations with communication constraints,” in *Robotics Research*, ser. Springer Tracts in Advanced Robotics, 2011, vol. 70, pp. 625–642.

[38] C. Belta and V. Kumar, “Abstraction and control for groups of robots,” *Robotics, IEEE Transactions on*, vol. 20, no. 5, pp. 865–875, 2004.

[39] V. Graciano Santos and L. Chaimowicz, “Hierarchical congestion control for robotic swarms,” in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, 2011, pp. 4372–4377.

[40] Y. Baryshnikov and D. P. Guralnik, “Hierarchical clustering and configuration spaces,” *(in preparation)*.

[41] E. R. Fadell and S. Y. Husseini, *Geometry and Topology of Configuration Spaces*. Springer, 2001.

[42] J. Felsenstein, *Inferring Phylogenies*. Sinauer Associates, Inc., 2004.

[43] M. S. Steinberg, “Reconstruction of tissues by dissociated cells,” *Science*, vol. 141, no. 3579, pp. pp. 401–408, 1963.

[44] A. M. Turing, “The chemical basis of morphogenesis,” *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, vol. 237, no. 641, pp. 37–72, 1952.

[45] A. Gierer and H. Meinhardt, “A theory of biological pattern formation,” *Kybernetik*, vol. 12, no. 1, pp. 30–39, 1972.

[46] J.-L. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain, and L. Chrétien, “The dynamics of collective sorting robot-like ants and ant-like robots,” in *From Animals to Animats: International Conference on Simulation of Adaptive Behavior*. MIT Press, 1991, pp. 356–363.

[47] J.-M. Ame, C. Rivault, and J.-L. Deneubourg, “Cockroach aggregation based on strain odour recognition,” *Animal Behaviour*, vol. 68, no. 4, pp. 793 – 801, 2004.

[48] M. A. Halverson, D. K. Skelly, and A. Caccone, “Kin distribution of amphibian larvae in the wild,” *Molecular Ecology*, vol. 15, no. 4, pp. 1139–1145, 2006.

[49] D. O’Brien, “Analysis of the internal arrangement of individuals within crustacean aggregations (euphausiacea, mysidacea),” *Journal of Experimental Marine Biology and Ecology*, vol. 128, no. 1, pp. 1 – 30, 1989.

[50] P. Kareiva and G. Odell, “Swarms of predators exhibit ”preytaxis” if individual predators use area-restricted search,” *The American Naturalist*, vol. 130, no. 2, pp. 233–270, 1987.

[51] J. K. Parrish and L. Edelstein-Keshet, “Complexity, pattern, and evolutionary trade-offs in animal aggregation,” *Science*, vol. 284, no. 5411, pp. 99–101, 1999.

[52] J. C. Bednarz, “Cooperative hunting in harris’ hawks (parabuteo unicinctus),” *Science*, vol. 239, no. 4847, pp. 1525–1527, 1988.

[53] C. Strbin, M. Steinegger, and R. Bshary, “On group living and collaborative hunting in the yellow saddle goatfish (parupeneus cyclostomus)1,” *Ethology*, vol. 117, no. 11, pp. 961–969, 2011.

[54] S. K. Gazda, R. C. Connor, R. K. Edgar, and F. Cox, “A division of labour with role specialization in grouphunting bottlenose dolphins (tursiops truncatus) off cedar key, florida,” *Proceedings of the Royal Society B: Biological Sciences*, vol. 272, no. 1559, pp. 135–140, 2005.

[55] D. Scheel and C. Packer, “Group hunting behaviour of lions: a search for cooperation,” *Animal Behaviour*, vol. 41, no. 4, pp. 697 – 709, 1991.

[56] P. E. Stander, “Cooperative hunting in lions: the role of the individual,” *Behavioral Ecology and Sociobiology*, vol. 29, no. 6, pp. 445–454, 1992.

[57] L. A. Nituch, J. A. Schaefer, and C. D. Maxwell, “Fine-scale spatial organization reflects genetic structure in sheep,” *Ethology*, vol. 114, no. 7, pp. 711–717, 2008.

[58] D. Watts and J. Mitani, “Hunting behavior of chimpanzees at ngogo, kibale national park, uganda,” *International Journal of Primatology*, vol. 23, no. 1, pp. 1–28, 2002.

[59] M. C. Crofoot and I. C. Gilby, “Cheating monkeys undermine group strength in enemy territory,” *Proceedings of the National Academy of Sciences*, vol. 109, no. 2, pp. 501–505, Jan 2012.

[60] J. H. Tien, S. A. Levin, and D. I. Rubenstein, “Dynamics of fish shoals: identifying key decision rules,” *Evolutionary Ecology Research*, vol. 6, no. 4, pp. 555–565, 2004.

[61] W. M. Hamner and P. P. Hamner, “Behavior of antarctic krill (euphausia superba): schooling, foraging, and antipredatory behavior,” *Canadian Journal of Fisheries and Aquatic Sciences*, vol. 57, no. S3, pp. 192–202, 2000.

[62] J. M. Fryxell, A. Mosser, A. R. E. Sinclair, and C. Packer, “Group formation stabilizes predatorprey dynamics,” *Nature*, vol. 449, no. 7165, pp. 1041–1043, 2007.

[63] G. Flierl, D. Grnbaum, S. Levin, and D. Olson, “From individuals to aggregations: the interplay between behavior and physics,” *Journal of Theoretical Biology*, vol. 196, no. 4, pp. 397–454, 1999.

[64] I. Couzin, “Collective minds,” *Nature*, vol. 445, no. 7129, p. 715, 2007.

[65] V. Santos, L. Pimenta, and L. Chaimowicz, “Segregation of multiple heterogeneous units in a robotic swarm,” in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, 2014, pp. 1112–1117.

[66] V. G. Santos and L. Chaimowicz, “Cohesion and segregation in swarm navigation,” *Robotica*, vol. 32, no. 02, pp. 209–223, 2014.

[67] T. Huang, M. Kapadia, N. Badler, and M. Kallmann, “Path planning for coherent and persistent groups,” in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, 2014, pp. 1652–1659.

[68] P. Ogren, “Split and join of vehicle formations doing obstacle avoidance,” in *Robotics and Automation (ICRA), 2004 IEEE International Conference on*, 2004, pp. 1951–1955.

[69] L. Chaimowicz and V. Kumar, “Aerial shepherds: Coordination among uavs and swarms of robots,” in *Distributed Autonomous Robotic Systems 6*. Springer Japan, 2007, pp. 243–252.

[70] P. Dames, M. Schwager, V. Kumar, and D. Rus, “A decentralized control policy for adaptive information gathering in hazardous environments,” in *Decision and Control, 2012 IEEE Conference on*, 2012, pp. 2807–2813.

[71] M. Schwager, D. Rus, and J.-J. Slotine, "Decentralized, adaptive coverage control for networked robots," *The International Journal of Robotics Research*, vol. 28, no. 3, pp. 357–375, 2009.

[72] S. M. Savaresi and D. L. Boley, "On the performance of bisecting k-means and pddp," in *Proceedings of the First SIAM International Conference on Data Mining (ICDM 2001)*, 2001, pp. 1–14.

[73] L. J. Billera, S. P. Holmes, and K. Vogtmann, "Geometry of the space of phylogenetic trees," *Advances in Applied Mathematics*, vol. 27, no. 4, pp. 733–767, 2001.

[74] B. Mirkin, *Mathematical Classification and Clustering*. Kluwer Academic Publishers, 1996.

[75] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers, 2011.

[76] D. Robinson, "Comparison of labeled trees with valency three," *Journal of Combinatorial Theory, Series B*, vol. 11, no. 2, pp. 105–119, 1971.

[77] G. Moore, M. Goodman, and J. Barnabas, "An iterative approach from the standpoint of the additive hypothesis to the dendrogram problem posed by molecular data sets," *Journal of Theoretical Biology*, vol. 38, no. 3, pp. 423–457, 1973.

[78] O. Arslan, D. Guralnik, and D. E. Koditschek, "Discriminative measures for comparison of phylogenetic trees," *arXiv:1310.5202 [q-bio.PE]*, 2013. [Online]. Available: arxiv.org/abs/1310.5202

[79] T. Lozano-Perez, M. T. Mason, and R. H. Taylor, "Automatic synthesis of fine-motion strategies for robots," *The International Journal of Robotics Research*, vol. 3, no. 1, pp. 3–24, 1984.

[80] D. E. Koditschek, "Adaptive techniques for mechanical systems," in *Proc. 5th. Yale Workshop on Adaptive Systems*, 1987, pp. 259–265.

[81] ——, "Some applications of natural motion control," *Journal of Dynamic Systems, Measurement, and Control*, vol. 113, pp. 552–557, 1991.

[82] V. I. Arnold, *Ordinary Differential Equations*. MIT Press, 1973.

[83] M. Farber, "Topological complexity of motion planning," *Discrete and Computational Geometry*, vol. 29, no. 2, p. 211221, 2003.

[84] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Prentice Hall, 2001.

[85] N. P. Bhatia and G. P. Szegö, *Dynamical Systems: Stability Theory and Applications*. Springer-Verlag, 1967.

[86] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, Third Edition*, 3rd ed. The MIT Press, 2009.

[87] B. DasGupta, X. He, T. Jiang, M. Li, J. Tromp, and L. Zhang, "On distances between phylogenetic trees," in *Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms*, 1997, pp. 427–436.

[88] H. S. M. Coxeter and S. L. Greitzer, *Geometry revisited*. Mathematical Association of America, 1996, vol. 19.

[89] O. Arslan, D. P. Guralnik, and D. E. Koditschek, "Navigation of distinct euclidean particles via hierarchical clustering (extended version)," University Of Pennsylvania, Tech. Rep., 2013, http://kodlab.seas.upenn.edu/Omur/TechReport2013 .

[90] O. Arslan and D. E. Koditschek, "On the optimality of Napoleon triangles," *Journal of Optimization Theory and Applciations*, (under review). [Online]. Available: http://arxiv.org/abs/1509.07218

[91] O. Arslan, D. P. Guralnik, and D. E. Koditschek, "Coordinated robot navigation via hierarchical clustering (extended version)," *arXiv:1507.01637 [cs.RO]*, 2015. [Online]. Available: http://arxiv.org/abs/1507.01637

[92] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 3, no. 3, pp. 325 – 349, 2005.

[93] L. Kuntz and S. Scholtes, "Structural analysis of nonsmooth mappings, inverse functions, and metric projections," *Journal of Mathematical Analysis and Applications*, vol. 188, no. 2, pp. 346 – 386, 1994.

Omur Arslan (S'09) received the B.Sc. and M.Sc. degrees in electrical and electronics engineering from the Middle East Technical University, Ankara, Turkey, in 2007 and from Bilkent University, Ankara, in 2009, respectively. He is currently working toward the Ph.D. degree with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia. He is also affiliated with Kod*Lab, which is a subsidiary of the General Robotics, Automation, Sensing, and Perception Laboratory. His current research interests include topological understanding of clustering methods and its application to multiagent control, machine perception, machine learning and data mining.

Dan P. Guralnik has completed his B.A., M.Sc. and Ph.D. (2005) in Pure Mathematics at the Technion IIT in Haifa, Israel, in the area of Geometric Group Theory. After two post-doctoral appointments at Vanderbilt University and the University of Oklahoma Mathematics departments he is now a post-doctoral researcher at the University of Pennsylvania department of Electrical and Systems Engineering. Along with his continuing interest in the asymptotic geometry of groups, he is currently interested in the formal connections between Geometry and Information Theory and their possible applications to the design of intelligent systems.

Daniel E. Koditschek (S'80–M'83–SM'93–F'04) received the Ph.D. degree in electrical engineering from Yale University, New Haven, CT, USA, in 1983.

Koditschek is the Alfred Fitler Moore Professor of Electrical and Systems Engineering at the University of Pennsylvania, where he served as Chair from 2005 - 2012. He holds secondary appointments in the Departments of Computer and Information Science and Mechanical Engineering and Applied Mechanics. Prior to joining Penn, Koditschek held faculty positions in the Electrical Engineering and Computer Science Department at the University of Michigan, Ann Arbor (1993-2004) and the Electrical Engineering Department at Yale University (1984 - 1992).

Koditschek's current research interests include robotics, the application of dynamical systems theory to intelligent machines, and nonlinear control.

Dr. Koditschek is a member of the AMS, ACM, MAA, SIAM, SICB, and Sigma Xi. He is a fellow of the AAAS. He received the Presidential Young Investigators Award in 1986.