



University of Pennsylvania
ScholarlyCommons

Departmental Papers (ESE)

Department of Electrical & Systems Engineering

8-2014

Navigation of Distinct Euclidean Particles via Hierarchical Clustering

Omur Arslan

University of Pennsylvania, omur@seas.upenn.edu


Dan P. Guralnik

University of Pennsylvania

Daniel E. Koditschek

University of Pennsylvania, kod@seas.upenn.edu

Follow this and additional works at: http://repository.upenn.edu/ease_papers

 Part of the [Electrical and Computer Engineering Commons](#), and the [Systems Engineering Commons](#)

Recommended Citation

Omur Arslan, Dan P. Guralnik, and Daniel E. Koditschek, "Navigation of Distinct Euclidean Particles via Hierarchical Clustering", *2014 International Workshop on the Algorithmic Foundations of Robotics* . August 2014.

For more information: [Kod*Lab](#)

This paper is posted at ScholarlyCommons. http://repository.upenn.edu/ease_papers/700

For more information, please contact repository@pobox.upenn.edu.

Navigation of Distinct Euclidean Particles via Hierarchical Clustering

Abstract

We present a centralized online (completely reactive) hybrid navigation algorithm for bringing a swarm of n perfectly sensed and actuated point particles in Euclidean d space (for arbitrary n and d) to an arbitrary goal configuration with the guarantee of no collisions along the way. Our construction entails a discrete abstraction of configurations using cluster hierarchies, and relies upon two prior recent constructions: (i) a family of hierarchy-preserving control policies and (ii) an abstract discrete dynamical system for navigating through the space of cluster hierarchies. Here, we relate the (combinatorial) topology of hierarchical clusters to the (continuous) topology of configurations by constructing “portals” — open sets of configurations supporting two adjacent hierarchies. The resulting online sequential composition of hierarchy-invariant swarming followed by discrete selection of a hierarchy “closer” to that of the destination along with its continuous instantiation via an appropriate portal configuration yields a computationally effective construction for the desired navigation policy.

Disciplines

Electrical and Computer Engineering | Engineering | Systems Engineering

Comments

For more information: [Kod*Lab](#)

Navigation of Distinct Euclidean Particles via Hierarchical Clustering

Omur Arslan, Dan P. Guralnik, and Daniel E. Koditschek

Department of Electrical and Systems Engineering,
University of Pennsylvania, Philadelphia, PA 19104,
{arslan,guralnik,kod}@seas.upenn.edu,

Abstract. We present a centralized online (completely reactive) hybrid navigation algorithm for bringing a swarm of n perfectly sensed and actuated point particles in Euclidean d space (for arbitrary n and d) to an arbitrary goal configuration with the guarantee of no collisions along the way. Our construction entails a discrete abstraction of configurations using cluster hierarchies, and relies upon two prior recent constructions: (i) a family of hierarchy-preserving control policies and (ii) an abstract discrete dynamical system for navigating through the space of cluster hierarchies. Here, we relate the (combinatorial) topology of hierarchical clusters to the (continuous) topology of configurations by constructing “portals” — open sets of configurations supporting two adjacent hierarchies. The resulting online sequential composition of hierarchy-invariant swarming followed by discrete selection of a hierarchy “closer” to that of the destination along with its continuous instantiation via an appropriate portal configuration yields a computationally effective construction for the desired navigation policy.

Keywords: Multi-Agent Coordination, Integrated Planning and Control, Swarm Robotics, Hierarchical Formation

1 Introduction

This paper introduces the use of cluster hierarchies in vector field planners for coordinated swarming. Hierarchical clustering offers an interesting means of ensemble task encoding and control. It provides a formalism for precise yet flexible expression, relaxing local proximity relations while allowing the imposition of more global requirements – and at whatever level of resolution may be appropriate to a given set of goals in a given problem setting. Here, we take a fresh and, as it turns out, completely successful look at what may be considered the simplest instance of a longstanding, familiar, hard problem: coordinated motion planning of a configuration of multiple bodies. Specifically, we address the case of fully actuated, first order point particles constrained only by the requirement to avoid self-intersection in their otherwise free ambient Euclidean space, controlled by a centralized vector field planner that has instantaneous, exact information about the location of each individual. Given a desired, labeled, free configuration of this swarm, along with a labeled target hierarchy that goal configuration

instantiates, we construct a hybrid controller guaranteed to bring almost every initial free configuration to that destination with no collisions along the way via a sequence of continuous controllers. The construction is computationally effective: the number of discrete transitions grows in the worst case with the square of the number of particles; each successive discrete transition can be computed reactively (i.e., as a function of the present configuration) in time that grows linearly with the number of particles; and the formulae that define each successive smooth vector field are rational functions (i.e. defined by quotients of polynomials over the ambient space) entailing terms whose number grows quadratically with the number of particles.

1.1 Background

We do not imagine that the hierarchy abstraction (nor any other) can budge the intrinsic complexity of the coordinated motion planning problem. Beyond this “simplest” (but non-trivial) problem, we suspect that systematic recourse to hierarchy can likely also afford computationally effective solutions to more “realistic” problem settings¹ — so long as they do not step across the line of intractability. For example, whereas motion planning for finite disks in a polygonal environment is strongly NP-hard [32], more relaxed versions entailing (perhaps partially) unlabeled specifications have yielded interesting planners in the recent literature [1, 31, 34], and we suspect that the cluster hierarchy abstraction may be usefully applicable to such partially labeled settings.

Within the domain of reactive or vector field motion planning, it has proven deceptively hard to determine exactly this line of intractability. Since the problem of reactively navigating swarms of disks was first introduced to robotics [35, 36], most research into dynamical coordination planners has embraced the navigation function paradigm [28]. A recent review of this two decade old literature is provided by [33] where a combination of intuitive and analytical results yields centralized planners for achieving goal configurations specified up to rigid transformation. But moving thick bodies in a compact workspace yields hard problems: even determining when and how the configuration space is connected entails an encounter with the ancient sphere packing problem [7]; past reactive solutions have produced controllers with terms growing super-exponentially in the number of disks even when the workspace is not compact [14]; and we suspect that the (hard won) conditions sufficient for guaranteeing the correctness of the traditional navigation function constructions applied to this problem [19] will turn out to imply as hinted in [7] that the resulting free space has the same homotopy type as the “simple” problem we solve here. In sum, we believe there is plenty of useful and challenging work to be done in such tractable settings — with few agents [27]; in low dimensions [10]; and so on — 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.

¹ We will mention in the conclusion a few such extensions presently in progress.

That a hierarchy of proximities might play a key role in the coordinated motion planning had already been hinted at in early work on this problem [22,23]. A cover over the neighborhood of the configuration space boundary by cluster hierarchies (closely related to what we term “strata” here — see [5] and below) plays an important role in the analysis of navigation functions for thickened disks operating with centralized control in a compact workspace [19]. Formulae incorporating “relation verification” functions (again expressing properties of cluster hierarchies closely related to our “strata”) that grow super-exponentially with the number of disks appear directly in the decentralized controllers for the thickened disks in an unbounded workspace proposed by [14]. 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 [6]. Thus, 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 we make between the topology of configuration space [15] and the topology of tree space [16] through the hierarchical clustering relation [18] is entirely new.

1.2 Organization and Contributions of the Paper

Section 2 introduces some underlying technical concepts and suggests via abstractly stated requirements that there are likely to be many alternative routes to the desired result other than specific instances we recruit from some of our recent previous work (Algorithm 1, constructing a hierarchy-preserving navigation scheme in the configuration space [5]; and Algorithm 2, constructing a computationally effective navigation scheme in the space of abstract clustering trees [4]). Section 3 presents the new results that enable the central contribution of this paper, the HNC Algorithm (Table 1). Namely, we show how to define and compute a “portal map” (17) — a computationally effective geometric realization in the configuration space of the edges of a graph over the space of abstract hierarchies (Theorem 1) — that will serve the role of a dynamically computed “prepares graph” [9] for the sequentially composed particle controllers whose correct recruitment solves the reactive motion planning problem (Theorem 2). Section 4 presents illustrative simulations of this new hybrid dynamical system. We conclude with a brief discussion of future work in Section 5.

2 General Framework

2.1 Background & Notation

Configuration Space Given an index set, $J = [n] := \{1, \dots, n\} \subset \mathbb{N}$, a *configuration*, $\mathbf{x} = (x_i)_{i \in J}$, is a labeled placement of $|J| = n$ distinct Euclidean particles, x_i . We find it convenient to identify the *configuration space* [15] with the set of distinct labelings, i.e., the injective mappings of J into \mathbb{R}^d ,

$$\text{Conf}(\mathbb{R}^d, J) := \left\{ \mathbf{x} \in (\mathbb{R}^d)^J \mid \|x_i - x_j\| \neq 0, \forall i \neq j \in J \right\}. \quad (1)$$

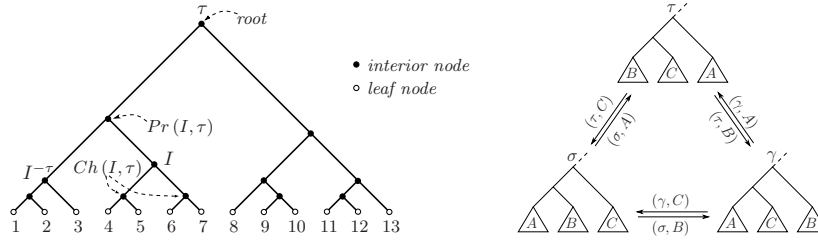


Fig. 1. (left) Hierarchical Relations: parent - $\text{Pr}(I, \tau)$, children - $\text{Ch}(I, \tau)$, and local complement (sibling) - $I^{-\tau}$ of cluster I of a rooted binary tree, $\tau \in \mathcal{BT}_{[13]}$. An interior node is referred by its cluster, the list of leaves below it; for example, $I = \{4, 5, 6, 7\}$. (right) An illustration of NNI moves between binary trees: each arrow is labeled by a source tree and associated cluster defining the move.

Cluster Hierarchies A rooted semi-labelled tree τ over a fixed finite index set J , illustrated in Figure 1, 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* [8]. A rooted tree with all interior vertices of out-degree two is said to be *binary* or, equivalently, *non-degenerate*, 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 rooted semi-labelled tree τ uniquely determines (and henceforth will be interchangeably used with) a *cluster hierarchy* [25]. 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 . 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) $\text{Pr}(I, \tau)$ and lists of children $\text{Ch}(I, \tau)$ of I in τ . Additionally, we find it useful to define the *local complement (sibling)* of cluster $I \in \mathcal{C}(\tau)$ as $I^{-\tau} := \text{Pr}(I, \tau) \setminus I$.

Configuration Hierarchies A *hierarchical clustering*² $\text{HC} \subset \text{Conf}(\mathbb{R}^d, J) \times \mathcal{BT}_J$ is a relation from the configuration space $\text{Conf}(\mathbb{R}^d, J)$ to the abstract space of binary hierarchies \mathcal{BT}_J [18], an example depicted in Figure 2. Here, we will only be interested in clustering methods that can classify all possible configurations (i.e. for which HC assigns some tree to every configuration), and so we impose the condition:

Property 1. HC is a multi-function.

Most standard divisive and agglomerative hierarchical clusterings exhibit this property, but generally fail to be functions because choices may be required between different but equally valid cluster splitting or merging decisions [18].

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

² 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 [18, 37].

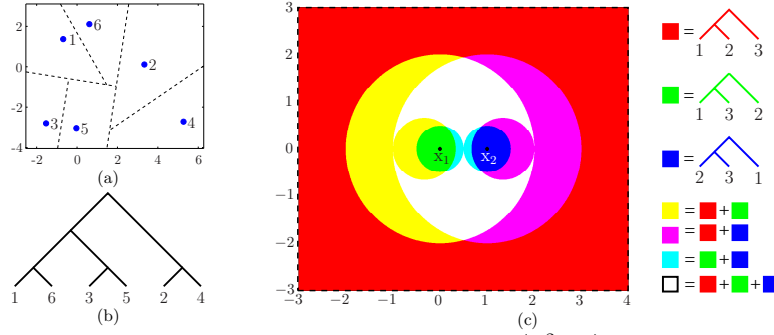


Fig. 2. An illustration of (a) a configuration in $\text{Conf}(\mathbb{R}^2, [6])$ and (b) its iterative 2-mean clustering [30] hierarchy in $\mathcal{BT}_{[6]}$, where the dashed lines in (a) depict the separating hyperplanes between clusters. (c) The Quotient Space $\text{Conf}(\mathbb{C}, [3]) / \sim$, where for any $\mathbf{x}, \mathbf{y} \in \text{Conf}(\mathbb{C}, [3])$, $\mathbf{x} \sim \mathbf{y} \iff \frac{x_3 - x_1}{x_2 - x_1} = \frac{y_3 - y_1}{y_2 - y_1}$. Here, configurations are quotiented out by translation, scale and rotation, and so $\mathbf{x}_1 = 0 + 0i$, $\mathbf{x}_2 = 1 + 0i$ and $\mathbf{x}_3 \in \mathbb{C} \setminus \{\mathbf{x}_1, \mathbf{x}_2\}$. Regions are colored according the associated cluster hierarchies resulting from their iterative 2-mean clustering. For instance, any configuration in the white region supports all hierarchies in $\mathcal{BT}_{[3]}$.

is the set of all configurations $\mathbf{x} \in \text{Conf}(\mathbb{R}^d, J)$ supporting the same tree τ [5],

$$\mathfrak{S}(\tau) := \left\{ \mathbf{x} \in \text{Conf}(\mathbb{R}^d, J) \mid (\mathbf{x}, \tau) \in \text{HC} \right\}, \quad (2)$$

and this yields a tree-indexed cover of the configuration space. For purposes of illustration, we depict in Figure 2(c) the strata of $\text{Conf}(\mathbb{C}, [3])$ — a space that represents a swarm of three particles on the plane.

The restriction to binary trees precludes combinatorial tree degeneracy [8] and we will avoid configuration degeneracy by imposing:

Property 2. Each stratum of HC includes an open subset of configurations, i.e. for every $\tau \in \mathcal{BT}_J$, $\mathring{\mathfrak{S}}(\tau) \neq \emptyset$.³

Once again, most standard hierarchical clusterings respect this requirement: they generally all agree (i.e. return the same result) and are robust to small perturbations of a configuration whenever all its clusters are well separated [37].

Graphs on Trees Define the *adjacency graph* $\mathcal{A}_J = (\mathcal{BT}_J, \mathcal{E}_{\mathcal{A}})$ to be the 1-skeleton of the nerve [17] of the $\text{Conf}(\mathbb{R}^d, J)$ -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, $\mathfrak{S}(\sigma) \cap \mathfrak{S}(\tau) \neq \emptyset$. The adjacency graph is a central object of interest in this paper; however, as Figure 2(c) anticipates, HC strata generally have complicated shapes, making it usually hard to compute the complete adjacency graph.

Fortunately, the computational biology literature [16] offers an alternative notion of adjacency that turns out to be both feasible and nicely compatible with our needs, yielding a computationally effective, fully connected subgraph of the adjacency graph, \mathcal{A}_J , as follows.

³ Here, \mathring{A} denotes the interior of set A .

The *Nearest Neighbor Interchange (NNI)* move at a cluster $A \in \mathcal{C}(\sigma)$ on a binary hierarchy $\sigma \in \mathcal{BT}_J$, as illustrated in Figure 1, swaps cluster A with its parent’s sibling $C = \text{Pr}(A, \sigma)^{-\sigma}$ to yield another binary hierarchy $\tau \in \mathcal{BT}_J$ [26, 29]. 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. 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. A central result of this paper will be to show how the NNI-graph yields a computationally effective sub-graph of the adjacency graph (Theorem 1).

2.2 Closely Related Prior Work

Hierarchy-Invariant Control Policies For ease of exposition we restrict attention to first order (completely actuated single integrator) particle dynamics, and we will be interested in smooth closed loop feedback laws (or hybrid controllers composed from them) that result in complete flows,

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

where $f : \text{Conf}(\mathbb{R}^d, J) \rightarrow (\mathbb{R}^d)^J$ is a vector field over $\text{Conf}(\mathbb{R}^d, J)$.⁴

Denote by φ^t the *flow* [2] on $\text{Conf}(\mathbb{R}^d, J)$ induced by the vector field, f . In [5] we introduce the class of hierarchy-invariant vector fields,

$$\mathcal{F}_{\text{HC}}(\tau) := \left\{ f : \text{Conf}(\mathbb{R}^d, J) \rightarrow (\mathbb{R}^d)^J \mid \varphi^t(\mathfrak{S}(\tau)) \subset \mathring{\mathfrak{S}}(\tau), t > 0 \right\}, \quad (4)$$

and use them to construct a hybrid controller that invariantly retracts almost all of a stratum onto any designated interior goal configuration. Namely, working with the 2-means divisive hierarchical clustering method [30], $\text{HC}_{2\text{-means}}$, given a hierarchy $\tau \in \mathcal{BT}_J$ and an interior goal, $\mathbf{y} \in \mathring{\mathfrak{S}}(\tau)$ we construct a pair of vector fields, $f_{\mathbf{y}}, f_{s(\mathbf{y})} \in \mathcal{F}_{\text{HC}}(\tau)$ with the following properties. The goal field, $f_{\mathbf{y}}$, has \mathbf{y} as a point attractor and includes in its basin a neighborhood of a suitably well separated and compactly clustered “standard” exemplar, $s(\mathbf{y}) \in \mathfrak{S}(\tau)$. The global field, $f_{s(\mathbf{y})}$ has $s(\mathbf{y})$ as a point attractor and includes in its basin a set $\mathfrak{S}_z(\tau) \subset \mathfrak{S}(\tau)$ that excludes at most a zero measure subset of $\mathfrak{S}(\tau)$. The formulae defining $f_{s(\mathbf{y})}$ and $f_{\mathbf{y}}$ are both rational functions (i.e. defined by quotients of polynomials over the ambient space) entailing terms whose numbers, respectively, grow quadratically and linearly with the number of particles. Using the standard “prepares” construction [9], wherein initial application of control $f_{s(\mathbf{y})}$ is switched to $f_{\mathbf{y}}$ upon reaching a suitably small neighborhood of $s(\mathbf{y})$, there results a deformation retraction [17], $R_{\tau, \mathbf{y}}$, of (almost all of) $\mathfrak{S}_z(\tau)$ onto $\{\mathbf{y}\}$.

Key for purposes of the present application is the observation that any hierarchy-invariant field $f \in \mathcal{F}_{\text{HC}}(\tau)$ must leave $\text{Conf}(\mathbb{R}^d, J)$ invariant as well, and thus avoids any self-collisions of the particles along the way. There are likely to be many alternative approaches to such results, but for purposes of this paper we will simply assume the availability of exactly such a prior construction that we summarize as follows.

⁴ A long prior robotics literature motivates the utility of this fully actuated “generalized damper” dynamical model [24], and provides methods for “lifts” to controllers for second order plants [20, 21] as well.

Algorithm 1 ([5]) *For any $\tau \in \mathcal{BT}_J$ and $\mathbf{y} \in \mathfrak{S}(\tau)$ associated with HC construct a (possibly hybrid) quadratic, $O(|J|^2)$, time computable control policy, $f_{\tau, \mathbf{y}}$, using the hierarchy invariant vector fields of $\mathcal{F}_{\text{HC}}(\tau)$ whose closed loop results in a retraction, $R_{\tau, \mathbf{y}}$, of $\mathfrak{S}_z(\tau)$ onto $\{\mathbf{y}\}$, where $\mathfrak{S}(\tau) \setminus \mathfrak{S}_z(\tau)$ has zero measure.*

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 NNI-graph. In principle, this is a trivial matter since the number of trees over a finite set of leaves is finite. In practice, the cardinality grows super exponentially [8],

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

for $|J| \geq 2$. Hence standard graph search algorithms, like the A* or Dijkstra’s algorithm [11], become rapidly impracticable. In particular, computing the shortest path (geodesic) in the NNI-graph is NP-complete [13].

Given a $\tau \in \mathcal{BT}_J$, we have recently developed in [4] an efficient recursive procedure for endowing 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 $\frac{1}{2}(|J| - 1)(|J| - 2)$ for $|J| \geq 2$. We interpret that directed NNI-graph as defining a deterministic discrete dynamical system in \mathcal{BT}_J that recursively generates paths toward the specified destination tree $\tau \in \mathcal{BT}_J$ from all other trees in \mathcal{BT}_J by reducing a “discrete Lyapunov function” relative to that destination. Given such a goal we show in [4] 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 the Lyapunov function is $O(|J|)$.

In this paper, such a provably correct, computationally efficient and recursively generated choice of next NNI moves will play the role of a discrete feedback policy used to define the reset map of our hybrid dynamical system. Thus, we further require the availability of such a construction, summarized as:

Algorithm 2 ([4]) *Given any $\tau \in \mathcal{BT}_J$ construct recursively a closed loop discrete dynamical system in the NNI-graph, taking the form of a deterministic discrete transition rule, g_τ , with global attractor at τ and longest trajectory of length $O(|J|^2)$ 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 $O(|J|)$.*

3 Hierarchical Navigation

The central technical result of this paper endows the strata of $\text{HC}_{2\text{-means}}$ [30] with a complete prepares graph [9] via a computationally effective geometric realization of the NNI-graph on trees.

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

$$\text{Portal}(\sigma, \tau) := \mathring{\mathfrak{S}}(\sigma) \cap \mathring{\mathfrak{S}}(\tau). \quad (6)$$

Theorem 1. *The NNI-graph $\mathcal{N}_J = (\mathcal{BT}_J, \mathcal{E}_\mathcal{N})$ is a sub-graph of the $\text{HC}_2\text{-means}$ adjacency graph $\mathcal{A}_J = (\mathcal{BT}_J, \mathcal{E}_\mathcal{A})$, and given an edge, $(\sigma, \tau) \in \mathcal{E}_\mathcal{N} \subset \mathcal{E}_\mathcal{A}$, a geometric realization via the map $\text{Port}_{(\sigma, \tau)} : \mathfrak{S}(\sigma) \rightarrow \text{Portal}(\sigma, \tau)$ (17) can be computed in linear, $O(|J|)$, time with the number of leaves, $|J|$.*

Proof. The relation between the tree graphs directly follows from Proposition 1. Further, $\text{Port}_{(\sigma, \tau)}$ is shown in Proposition 2 to be a retraction of $\mathfrak{S}(\sigma)$ into the set of standard portal configurations in $\text{Portal}(\sigma, \tau)$. Observe that by construction $\text{Port}_{(\sigma, \tau)}$ (17) only requires centroids of clusters of σ , computable in linear time by post-order traversal of σ , and some associated cluster radii in (11) - (13), also computable in linear time given cluster centroids. Thus, the result follows. \square

Before proceeding to the details of this construction, we summarize how it, together with the constructions reviewed in Section 2, solve the centralized hierarchical navigation problem.

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

Assume the selection of a goal configuration $\mathbf{y} \in \mathring{\mathfrak{S}}(\tau)$ 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 1 presents the HNC algorithm.

Table 1. The HNC Algorithm

<p>For (almost) any initial $\mathbf{x} \in \mathfrak{S}(\sigma)$ and $\sigma \in \mathcal{BT}_J$, and desired $\mathbf{y} \in \mathring{\mathfrak{S}}(\tau)$ and $\tau \in \mathcal{BT}_J$,</p> <ol style="list-style-type: none"> 1. (Hybrid Base Case) if $\mathbf{x} \in \mathfrak{S}(\tau)$ then apply stratum-invariant dynamics, $f_{\tau, \mathbf{y}}$ (Algorithm 1). 2. (Hybrid Recursive Step) else, <ol style="list-style-type: none"> (a) invoke the NNI transition rule g_τ (Algorithm 2) to propose an adjacent tree, $\gamma \in \mathcal{BT}_J$, with lowered discrete Lyapunov value. (b) Choose local configuration goal, $\mathbf{z} := \text{Port}_{(\sigma, \gamma)}(\mathbf{x})$ (17). (c) Apply the stratum-invariant continuous controller $f_{\sigma, \mathbf{z}}$ (Algorithm 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 2. *The HNC Algorithm in Table 1 defines a hybrid dynamical system whose execution brings almost every initial configuration, $\mathbf{x} \in \text{Conf}(\mathbb{R}^d, J)$, in finite time to an arbitrarily small neighborhood of $\mathbf{y} \in \mathring{\mathfrak{S}}(\tau)$ with the guarantee of no collisions along the way and with a computational cost no greater than $O(|J|)$ at each discrete transition.*

Proof. In the base case, 1) the conclusion follows from the construction of Algorithm 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 (Algorithm 2), and a new local policy $f_{\sigma, \mathbf{z}}$ is automatically deployed with a local goal configuration

$\mathbf{z} \in \text{Portal}(\sigma, \gamma)$ found in b). Recall from Algorithm 2 and Theorem 1 that the transition from σ to γ and the portal location \mathbf{z} can be both computed in linear $O(|J|)$ time. Next, the flow $f_{\sigma, \mathbf{z}}$ in c) is guaranteed to keep the state in $\mathfrak{S}(\sigma)$ and approach $\mathbf{z} \in \text{Portal}(\sigma, \gamma)$ 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 $\text{Portal}(\sigma, \gamma) \subset \mathfrak{S}(\gamma)$ 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 (Algorithm 2), it must eventually trigger the base case. \square

3.2 Hierarchical Portals

We now turn attention to construction of the crucial portal map (17) that effects the geometric realization of the NNI-graph as required for Theorem 1, above. Throughout the sequel, we confine our attention to 2-means divisive hierarchical clustering [30], $\text{HC}_{2\text{-means}}$. We first detail our construction of the realization function, Port (17), that takes an NNI-edge and returns a target configuration, and then verify that this image does indeed lie in the interior of $\text{Portal}(\sigma, \tau)$.

Hierarchical Strata of $\text{HC}_{2\text{-means}}$ The open and closed strata of $\text{HC}_{2\text{-means}}$ can be characterized respectively, by the intersection inverse images,⁵ [5]

$$\mathfrak{S}_o(\tau) = \bigcap_{I \in \mathcal{C}(\tau) \setminus \{J\}} \bigcap_{i \in I} \eta_{i, I, \tau}^{-1}(-\infty, 0), \quad \mathfrak{S}(\tau) = \bigcap_{I \in \mathcal{C}(\tau) \setminus \{J\}} \bigcap_{i \in I} \eta_{i, I, \tau}^{-1}(-\infty, 0], \quad (7)$$

of the scalar valued “separation” function, $\eta_{i, I, \tau} : \text{Conf}(\mathbb{R}^d, J) \rightarrow \mathbb{R}$. This function returns the distance of agent i in cluster $I \in \mathcal{C}(\tau) \setminus \{J\}$ to the separating hyperplane that is perpendicular to the separation vector, $s_{I, \tau}(\mathbf{x})$, between centroids of complementary clusters I and $I^{-\tau}$ and passes through the midpoint, $m_{I, \tau}(\mathbf{x})$, of their centroids,⁶

$$\eta_{i, I, \tau}(\mathbf{x}) := (\mathbf{x}_i - m_{I, \tau}(\mathbf{x}))^T s_{I, \tau}(\mathbf{x}), \quad (8)$$

where

$$c(\mathbf{x}|I) := \frac{1}{|I|} \sum_{i \in I} \mathbf{x}_i, \quad s_{I, \tau}(\mathbf{x}) := c(\mathbf{x}|I^{-\tau}) - c(\mathbf{x}|I), \quad m_{I, \tau}(\mathbf{x}) := \frac{c(\mathbf{x}|I) + c(\mathbf{x}|I^{-\tau})}{2}. \quad (9)$$

Definition 2. Let $\mathbf{x} \in \text{Conf}(\mathbb{R}^d, J)$ and $\tau \in \mathcal{BT}_J$. Then cluster I of τ is said to be admissible (valid) for \mathbf{x} if $\eta_{i, I, \tau}(\mathbf{x}) \leq 0$ for all $i \in I$.

Using this terminology, we observe from (7) that $\mathfrak{S}(\tau)$ comprises the set of all configurations in $\text{Conf}(\mathbb{R}^d, J)$ for which every cluster of τ is admissible [5].

Portal Configurations A critical observation for the strata of $\text{HC}_{2\text{-means}}$ is:

Proposition 1. The NNI-graph is a sub-graph of the adjacency graph, i.e. for any pair (σ, τ) of NNI-adjacent trees in \mathcal{BT}_J , $\text{Portal}(\sigma, \tau) \neq \emptyset$.

Proof. The result directly follows from Corollary 1. \square

⁵ Note that for all $\tau \in \mathcal{BT}_J$, $\mathfrak{S}_o(\tau) \subseteq \mathring{\mathfrak{S}}(\tau)$.

⁶ Here, \mathbf{A}^T denotes the transpose of \mathbf{A} .

Throughout this section, the trees $\sigma, \tau \in \mathcal{BT}_J$ are NNI-adjacent 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 [4] 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 $P := A \cup B \cup C$.

We now introduce a set of useful notation and lemmas for characterizing a particular subset of $\text{Portal}(\sigma, \tau)$. A relaxation on Definition 2 is:

Definition 3. Let $\mathbf{x} \in (\mathbb{R}^d)^J$, $\tau \in \mathcal{BT}_J$ and $K \subseteq J$. Then cluster I of τ is said to be partially admissible for $\mathbf{x}|K$ if $\eta_{i,I,\tau}(\mathbf{x}) \leq 0$ for all $i \in I \cap K$.⁷

For a partition $\{I_\alpha\}$ of cluster $I \in \mathcal{C}(\tau)$, observe that cluster I of τ is admissible for \mathbf{x} if and only if I is partially admissible for all $\mathbf{x}|I_\alpha$'s.

Definition 4. Let $\mathbf{x} \in (\mathbb{R}^d)^J$, $Q \in \{A, B, C\}$, and for any $H \subseteq \mathbb{R}^d$ define

$$\mathcal{Y}_Q(\mathbf{x}, H) := \left\{ \mathbf{y} \in (\mathbb{R}^d)^J \mid \forall R \in \{A, B, C\} \ c(\mathbf{y}|R) = c(\mathbf{x}|R), \forall i \in Q \ y_i \in H \right\}. \quad (10)$$

The consensus ball $B_Q(\mathbf{x})$ of partial configuration $\mathbf{x}|Q$ is defined to be the largest open ball⁸ centered at $c(\mathbf{x}|Q)$ so that for any $\mathbf{y} \in \mathcal{Y}_Q(\mathbf{x}, B_Q(\mathbf{x}))$ and $\gamma \in \{\sigma, \tau\}$ every cluster $D \in \{Q, \text{Pr}(Q, \gamma)\} \setminus \{P\}$ of γ are partially admissible for $\mathbf{y}|Q$.

An explicit form of the radius $r_Q(\mathbf{x})$ of $B_Q(\mathbf{x})$ can be obtained as [3]⁹

$$r_Q(\mathbf{x}) := \min \left\{ -\left(c(\mathbf{x}|Q) - m_{D,\gamma}(\mathbf{x}) \right)^T \left(\frac{s_{D,\gamma}(\mathbf{x})}{\|s_{D,\gamma}(\mathbf{x})\|_2} \right) \mid \gamma \in \{\sigma, \tau\}, D \in \{Q, \text{Pr}(Q, \gamma)\} \setminus \{P\} \right\}. \quad (11)$$

Here, $r_Q(\mathbf{x}) < 0$ means $B_Q(\mathbf{x})$ is empty. We will abuse the notion of the consensus ball for a single tree, σ , and its cluster, $I \in \mathcal{C}(\sigma) \setminus \{J\}$, as the open ball centered at $c(\mathbf{x}|I)$ with radius

$$r_{I,\sigma}(\mathbf{x}) := \min \left\{ -\left(c(\mathbf{x}|I) - m_{D,\sigma}(\mathbf{x}) \right)^T \left(\frac{s_{D,\sigma}(\mathbf{x})}{\|s_{D,\sigma}(\mathbf{x})\|_2} \right) \mid D \in \{K \in \mathcal{C}(\sigma) \mid I \subseteq K \subsetneq J\} \right\}. \quad (12)$$

It is also convenient to have $r(\mathbf{x})$ denote the centroidal radius of $\mathbf{x} \in (\mathbb{R}^d)^J$,

$$r(\mathbf{x}) := \max_{i \in J} \|x_i - c(\mathbf{x})\|_2. \quad (13)$$

Looking ahead toward Lemma 1, the sufficiency condition for the existence of nontrivial consensus balls motivates:

⁷ Here, we use $\eta_{i,I,\tau} : (\mathbb{R}^d)^J \rightarrow \mathbb{R}$ (8).

⁸ In a metric space (X, d) , the open ball $B(x, r)$ centered at x with radius $r \in \mathbb{R}_{\geq 0}$ is the set of points in X whose distance to x is less than r , i.e. $B(x, r) = \{y \in X \mid d(x, y) < r\}$.

⁹ Here, we set $\frac{x}{\|x\|_2} = 0$ for $x = 0$.

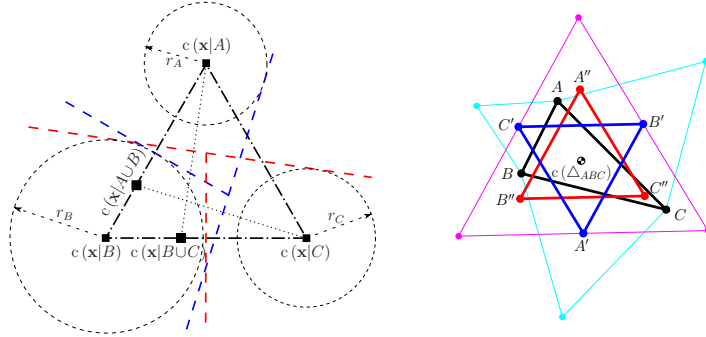


Fig. 3. (left) An illustration of a symmetric configuration $\mathbf{x} \in \text{Sym}(\sigma, \tau)$, where the consensus ball $B_Q(\mathbf{x})$ of partial configuration of cluster $Q \in \{A, B, C\}$ has a positive radius. (right) Outer Napoleon Triangles $\Delta_{A'B'C'}$ and $\Delta_{A''B''C''}$ of Δ_{ABC} and $\Delta_{A'B'C'}$, respectively, and $\Delta_{A''B''C''}$ is referred to as the double outer triangle of Δ_{ABC} . Note that centroids of all triangles coincides, i.e. $c(\Delta_{ABC}) = c(\Delta_{A'B'C'}) = c(\Delta_{A''B''C''})$.

Definition 5. We call $\mathbf{x} \in (\mathbb{R}^d)^J$ 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. The set of all symmetric configurations with respect to (σ, τ) is denoted $\text{Sym}(\sigma, \tau)$.

Lemma 1 ([3]). For any symmetric configuration $\mathbf{x} \in \text{Sym}(\sigma, \tau)$, the consensus ball $B_Q(\mathbf{x})$ of each partial configuration of cluster $Q \in \{A, B, C\}$ always has a nonempty interior, i.e. $r_Q(\mathbf{x}) > 0$ — see Figure 3.

In general, the geometric shape of $\text{Portal}(\sigma, \tau)$ is very hard to characterize, as suggested by Figure 2. Fortunately, Lemma 1 lets us point out an easily identifiable open subset:

Definition 6. The standard portal $\text{StdPortal}(\sigma, \tau)$ of the pair (σ, τ) is the set of all configurations $\mathbf{x} \in \mathfrak{S}_o(\sigma) \cap \text{Sym}(\sigma, \tau)$ with the property that $\mathbf{x}|Q$ is contained in the consensus ball $B_Q(\mathbf{x})$ for all $Q \in \{A, B, C\}$.

Accordingly, using Lemma 1, one can conclude that:

Corollary 1. $\text{StdPortal}(\sigma, \tau) \neq \emptyset$, and $\text{StdPortal}(\sigma, \tau) \subset \text{Portal}(\sigma, \tau)$.

Portal Transformations

Napoleon Triangles [12] 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*” [12] — see Figure 3. 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 $\text{NT} : \mathbb{R}^{3d} \rightarrow \mathbb{R}^{3d}$ denote the double outer Napoleon transformation, see [3] for an explicit form of NT.

The NNI-triplet $\{A, B, C\}$ defines an associated triangle with distinct vertices for each configuration, $\Delta_{A,B,C} : \mathfrak{S}(\tau) \rightarrow \mathbf{Conf}(\mathbb{R}^d, [3])$,

$$\Delta_{A,B,C}(\mathbf{x}) := [c(\mathbf{x}|A), c(\mathbf{x}|B), c(\mathbf{x}|C)]^T. \quad (14)$$

The double outer Napoleon transformation of $\Delta_{A,B,C}(\mathbf{x})$ returns symmetric target locations for $c(\mathbf{x}|A)$, $c(\mathbf{x}|B)$ and $c(\mathbf{x}|C)$, and the corresponding displacement of $c(\mathbf{x}|P)$, denoted $\mathbf{Noff}_{A,B,C} : \mathbf{Conf}(\mathbb{R}^d, J) \rightarrow \mathbb{R}^d$, is given by the formula¹⁰

$$\mathbf{Noff}_{A,B,C}(\mathbf{x}) := c(\mathbf{x}|P) - \Gamma \cdot \mathbf{NT} \circ \Delta_{A,B,C}(\mathbf{x}), \quad (15)$$

where $\Gamma := \frac{1}{|P|} [|A|, |B|, |C|] \otimes \mathbf{I}_d \in \mathbb{R}^{d \times 3d}$, and the vertices of the associated equilateral triangle with compensated offset of $c(\mathbf{x}|P)$ are¹⁰

$$[c_A, c_B, c_C]^T := \mathbf{NT} \circ \Delta_{A,B,C}(\mathbf{x}) + \mathbf{1}_3 \otimes \mathbf{Noff}_{A,B,C}(\mathbf{x}). \quad (16)$$

Portal Maps Define a continuous map,

$$\mathbf{Port} : \mathfrak{S}(\sigma) \rightarrow \mathbf{Sym}(\sigma, \tau) : \mathbf{x} \rightarrow \begin{cases} \mathbf{x} & , \text{ if } \mathbf{x} \in \mathbf{StdPortal}(\sigma, \tau), \\ (\mathbf{Mrg} \circ \mathbf{Sc1} \circ \mathbf{Ctr})(\mathbf{x}), & \text{ otherwise,} \end{cases} \quad (17)$$

where

$$\mathbf{Ctr} : \mathfrak{S}(\sigma) \rightarrow \mathbf{Sym}(\sigma, \tau) : \mathbf{x} \rightarrow \begin{cases} x_i & , \text{ if } i \notin P, \\ x_i - c(\mathbf{x}|Q) + c_Q, & \text{ if } i \in Q, Q \in \{A, B, C\}, \end{cases} \quad (18)$$

and c_A , c_B and c_C are the new centroids of the corresponding partial configurations (16). It is important to observe that \mathbf{Ctr} keeps the barycenter of $\mathbf{x}|P$ fixed, and so the rest of clusters ascending and disjoint with P are kept unchanged.

After obtaining a symmetric configuration in $\mathbf{Sym}(\sigma, \tau)$, based on Lemma 1, $\mathbf{Sc1} : \mathbf{Sym}(\sigma, \tau) \rightarrow \mathbf{Sym}(\sigma, \tau)$ scales each partial configuration, $\mathbf{x}|A$, $\mathbf{x}|B$ and $\mathbf{x}|C$, to fit into the corresponding consensus ball, and then $\mathbf{Mrg} : \mathbf{Sym}(\sigma, \tau) \rightarrow \mathbf{Sym}(\sigma, \tau)$ scales $\mathbf{x}|P$ to merge with the rest of (unchanged) particles, $\mathbf{x}|J - P$, to simultaneously support both hierarchies σ and τ ,

$$\mathbf{Sc1}(\mathbf{x})_i = \zeta \frac{r_Q(\mathbf{x})}{r(\mathbf{x}|Q)} (x_i - c(\mathbf{x}|Q)) + c(\mathbf{x}|Q), \quad \mathbf{Mrg}(\mathbf{x})_i = \zeta \frac{r_{P,\sigma}(\mathbf{x})}{r(\mathbf{x}|P)} (x_i - c(\mathbf{x}|P)) + c(\mathbf{x}|P), \quad (19)$$

for all $i \in Q$ and $Q \in (A, B, C)$; otherwise ($i \notin P$), $\mathbf{Sc1}(\mathbf{x})_i = \mathbf{Mrg}(\mathbf{x})_i = x_i$, where $\zeta \in (0, 1)$ is a parameter describing the scale of each configuration with respect to the consensus ball.

Proposition 2 ([3]). $\mathbf{Port} : \mathfrak{S}(\sigma) \rightarrow \mathbf{StdPortal}(\sigma, \tau)$ is a retraction.

4 Numerical Simulations

For the sake of clarity, we first illustrate the behavior of the hybrid system defined in Section 3.1 for the case of four particles moving in a two dimensional ambient space.

¹⁰ 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.

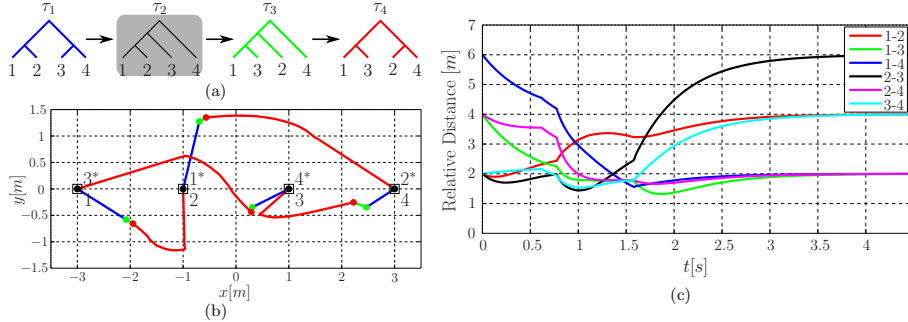


Fig. 4. An illustrative navigation trajectory of the hybrid dynamics generated by the HNC algorithm for 4 particles. (a) The sequence of trees associated with deployed local controllers during the execution of the hybrid navigation controller. Here, the hybrid planner instantaneously switches from the second controller to the next controller. (b) Trajectory of each particle colored according to the active local controller, where $\mathbf{x}_g \in \mathfrak{S}(\tau_1) \cap \mathfrak{S}(\tau_2) \cap \mathfrak{S}(\tau_3)$ and $\mathbf{x}_r \in \mathfrak{S}(\tau_3) \cap \mathfrak{S}(\tau_4)$ shown by green and red dots, respectively, are portal configurations. (3) Pairwise distances between particles.

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, $\mathbf{x}_o \in \mathfrak{S}(\tau_1)$, and desired configurations, $\mathbf{x}^* \in \mathfrak{S}(\tau_4)$, where particles are evenly placed 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 $\tau_1 \in \mathcal{BJ}_{[4]}$ and $\tau_4 \in \mathcal{BJ}_{[4]}$, see Figure 4.

The resultant trajectory of each particle following the hybrid navigation planner in Section 3.1, the relative distance between each pair of particles and the sequence of trees associated with visited hierarchical strata are shown in Figure 4. Here, notice that when the swarm enters the domain of local controller associated with τ_2 at $\mathbf{x}_g \in \mathfrak{S}(\tau_1) \cap \mathfrak{S}(\tau_2)$ — shown by green dots in Figure 4, it already finds itself in the domain of the following controller associated with τ_3 , i.e. $\mathbf{x}_g \in \mathfrak{S}(\tau_3)$, but not still in $\mathfrak{S}(\tau_4)$. After a finite time navigating in $\mathfrak{S}(\tau_3)$, the swarm enters the domain of the goal controller f_{τ_4, \mathbf{x}^*} (Algorithm 1) at $\mathbf{x}_r \in \mathfrak{S}(\tau_3) \cap \mathfrak{S}(\tau_4)$ — shown by red dots in Figure 4, and f_{τ_4, \mathbf{x}^*} asymptotically steers particles to the desired configuration $\mathbf{x}^* \in \mathfrak{S}(\tau_4)$. Finally, note that the total number of binary trees over four leaves is 15; however, our hybrid navigation planner reactively deploys only 4 of them.

We now consider a similar, but slightly more complicated setting: a swarm of six particles in a plane where agents are initially placed evenly on the horizontal axes and switch their positions at the destination as shown in Figure 5(a), which is also used in [33] as an example of complicated multi-agent arrangements. While steering the swarm towards the goal, the hybrid navigation planner automatically deploys only 8 local controllers out of the family of 945 local controllers. The time evolution of the swarm is illustrated in Figure 5(a).

Finally, to demonstrate the efficiency of the deployment policy of our hybrid planner, we separately consider swarms of 8 and 16 particles in an ambient plane, illustrated in Figure 5. The eight particles are initially located at the

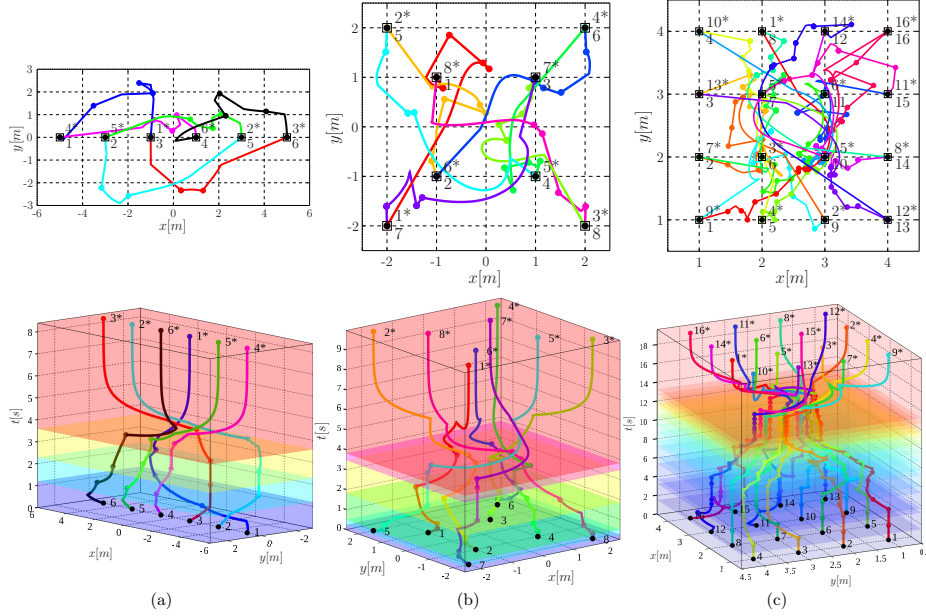


Fig. 5. Example trajectories of the hybrid vector field planner for (a) 6, (b) 8 and (c) 16 particles in a planar ambient space. (top) Trajectory and (bottom) state-time curve of each agent. Each colored time interval demonstrates the execution duration of an excited local controller. Dots correspond to the portal configurations where transitions between local controllers occur at.

corner of two squares whose centroids coincide and the perimeter of one is twice of the perimeter of the other. At the destination, agents switch their locations as illustrated in Figure 5(b). For sixteen particle case, agents are initially placed at the vertices of a 4 by 4 grid, and their task is to switch their location as illustrated in Figure 5(c). Although there are a large number of local controllers for the case of swarms of 8 and 16 particles ($|\mathcal{BT}_{[8]}| > 10^5$ and $|\mathcal{BT}_{[16]}| > 6 \times 10^{15}$), our hybrid navigation planner only deploys 16 and 34 local controllers, respectively.

The number of potentially available local controllers for a swarm of n particles (5) grows super exponentially with n . On the other hand, if agents have perfect sensing and actuation modelled as in the present paper, the hybrid navigation planner automatically deploys at most $\frac{1}{2}(n-1)(n-2)$ local controllers [4], illustrating the computational efficiency of our construction.

5 Conclusion

In this paper, we introduce an online centralized hybrid vector field planner for navigation in the configuration space of n distinct points in \mathbb{R}^d , using the hierarchy invariant controllers of [5], the combinatorial tree navigation algorithm of [4], and its “pullback” into the configuration space, **Port** (17). This last step comprises the central contribution of the paper, revealing the relation between the combinatorial NNI neighborhood of hierarchy trees and the intersection of their associated configuration space strata. The new result, the HNC Algorithm,

now affords provably correct online reactive planing and execution of arbitrary reconfiguration in the space of multiple, distinct, completely actuated first order particles in \mathbb{R}^d .

Work now in progress targets more practical settings in the field of robotics including navigating around obstacles and handling thickened disk agents in compact spaces. Another focus of ongoing work addresses the realization of tree space topology via online, “cluster-local” computation that might afford a distributed version of the current centralized framework.

ACKNOWLEDGMENT

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.

References

1. Adler, A., de Berg, M., Halperin, D., Solovey, K.: Efficient multi-robot motion planning for unlabeled discs in simple polygons. In: 30th European Workshop on Computational Geometry (EuroCG 2014) (2013) [2](#)
2. Arnold, V.I.: Ordinary Differential Equations. MIT Press (1973) [6](#)
3. Arslan, O., Guralnik, D.P., Koditschek, D.E.: Navigation of distinct euclidean particles via hierarchical clustering (extended version). Tech. rep., University Of Pennsylvania (2013), <http://kodlab.seas.upenn.edu/Omur/TechReport2013> [10](#), [11](#), [12](#)
4. Arslan, O., Guralnik, D., Koditschek, D.E.: Discriminative measures for comparison of phylogenetic trees. Tech. rep., University Of Pennsylvania (2013), <http://arxiv.org/abs/1310.5202> [3](#), [7](#), [10](#), [14](#)
5. Arslan, O., Guralnik, D.P., Koditschek, D.E.: Hierarchically clustered navigation of distinct euclidean particles. In: 50th Annual Allerton Conference on Communication, Control, and Computing (October 2012), <http://kodlab.seas.upenn.edu/Main/Allerton2012> [3](#), [5](#), [6](#), [7](#), [9](#), [14](#)
6. Ayanian, N., Kumar, V., Koditschek, D.: Synthesis of controllers to create, maintain, and reconfigure robot formations with communication constraints. In: Robotics Research, Springer Tracts in Advanced Robotics, vol. 70, pp. 625–642 (2011) [3](#)
7. Baryshnikov, Y., Bubenik, P., Kahle, M.: Min-type morse theory for configuration spaces of hard spheres. International Mathematics Research Notices (2013) [2](#)
8. Billera, L.J., Holmes, S.P., Vogtmann, K.: Geometry of the space of phylogenetic trees. Advances in Applied Mathematics 27(4), 733–767 (2001) [4](#), [5](#), [7](#)
9. Burridge, R.R., Rizzi, A.A., Koditschek, D.E.: Sequential composition of dynamically dexterous robot behaviors. The International Journal of Robotics Research 18(6), 534–555 (1999) [3](#), [6](#), [7](#)
10. Choi, Y.C., Ahn, H.S.: Formation control of quad-rotors in three dimension based on euclidean distance dynamics matrix. In: Control, Automation and Systems (IC-CAS), 2011 11th International Conference on. pp. 1168–1173 (Oct 2011) [2](#)
11. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, Third Edition. The MIT Press, 3rd edn. (2009) [7](#)
12. Coxeter, H.S.M., Greitzer, S.L.: Geometry revisited, vol. 19. Mathematical Association of America (1996) [11](#)
13. DasGupta, B., He, X., Jiang, T., Li, M., Tromp, J., Zhang, L.: On distances between phylogenetic trees. In: Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 427–436. Society for Industrial and Applied Mathematics (1997) [7](#)
14. Dimarogonas, D.V., Loizou, S.G., Kyriakopoulos, K.J., Zavlanos, M.M.: A feedback stabilization and collision avoidance scheme for multiple independent non-point agents. Automatica 42(2), 229–243 (2006) [2](#), [3](#)

15. Fadell, E.R., Husseini, S.Y.: *Geometry and Topology of Configuration Spaces*. Springer (2001) 3
16. Felsenstein, J.: *Inferring Phylogenies*. Sinauer Associates, Inc. (2004) 3, 5
17. Hatcher, A.: *Algebraic topology*. Cambridge Univ. Press (2002) 5, 6
18. Jain, A.K., Dubes, R.C.: *Algorithms for clustering data*. Prentice-Hall, Inc. (1988) 3, 4
19. Karagoz, C.S., Bozma, H.I., Koditschek, D.E.: Coordinated motion of disk-shaped independent robots in 2d workspaces. Univ. Michigan, Ann Arbor, Tech. Rep. CSE-TR-486-04, Feb (2004) 2, 3
20. Koditschek, D.E.: Adaptive techniques for mechanical systems. In: Proc. 5th. Yale Workshop on Adaptive Systems. pp. 259–265 (May 1987) 6
21. Koditschek, D.E.: Some applications of natural motion control. *Journal of Dynamic Systems, Measurement, and Control* 113, 552–557 (1991) 6
22. Liu, Y.H., Kuroda, S., Naniwa, T., Noborio, H., Arimoto, S.: A practical algorithm for planning collision-free coordinated motion of multiple mobile robots. In: *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*. pp. 1427–1432 (1989) 3
23. Liu, Y.H., Arimoto, S., Noborio, H.: New solid model hsm and its application to interference detection between moving objects. *Journal of Robotic Systems* 8(1), 39–54 (1991) 3
24. Lozano-Perez, T., Mason, M.T., Taylor, R.H.: Automatic synthesis of fine-motion strategies for robots. *The International Journal of Robotics Research* 3(1), 3–24 (1984) 6
25. Mirkin, B.: *Mathematical Classification and Clustering*. Kluwer Academic Publishers (1996) 4
26. Moore, G., Goodman, M., Barnabas, J.: An iterative approach from the standpoint of the additive hypothesis to the dendrogram problem posed by molecular data sets. *Journal of Theoretical Biology* 38(3), 423–457 (1973) 6
27. Oh, K.K., Ahn, H.S.: Distance-based formation control using euclidean distance dynamics matrix: Three-agent case. In: *American Control Conference (ACC), 2011*. pp. 4810–4815 (July 2011) 2
28. Rimón, E., Koditschek, D.: Exact robot navigation using artificial potential functions. *Robotics and Automation, IEEE Transactions on* 8(5), 501–518 (1992) 2
29. Robinson, D.: Comparison of labeled trees with valency three. *Journal of Combinatorial Theory, Series B* 11(2), 105–119 (1971) 6
30. Savaresi, S.M., Boley, D.L.: On the performance of bisecting k-means and pddp. In: *Proceedings of the First SIAM International Conference on Data Mining (ICDM 2001)*. pp. 1–14 (2001) 5, 6, 7, 9
31. Solovey, K., Halperin, D.: k-color multi-robot motion planning. *The International Journal of Robotics Research* 33(1), 82–97 (2014) 2
32. Spirakis, P., Yap, C.K.: Strong np-hardness of moving many discs. *Information Processing Letters* 19(1), 55–59 (1984) 2
33. Tanner, H., Boddu, A.: Multiagent navigation functions revisited. *Robotics, IEEE Transactions on* 28(6), 1346–1359 (Dec 2012) 2, 13
34. Turpin, M., Michael, N., Kumar, V.: Concurrent assignment and planning of trajectories for large teams of interchangeable robots. In: *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. pp. 842–848. IEEE (2013) 2
35. Whitcomb, L.L., Koditschek, D.E., Cabrera, J.B.D.: Toward the automatic control of robot assembly tasks via potentialfunctions: the case of 2-d sphere assemblies. *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on* pp. 2186–2191 (1992) 2
36. Whitcomb, L.L., Koditschek, D.E.: Automatic assembly planning and control via potential functions. In: *Intelligent Robots and Systems 91. Intelligence for Mechanical Systems, Proceedings IROS91. IEEE/RSJ International Workshop on*. pp. 17–23 (1991) 2
37. Witten, I.H., Frank, E.: *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann (2005) 4, 5