

CLUSTERING-BASED ROBOT NAVIGATION AND CONTROL

Omur Arslan

A DISSERTATION

in

Electrical and Systems Engineering

Presented to the Faculties of the University of Pennsylvania
in Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy
2016

Supervisor of Dissertation

Daniel E. Koditschek, Professor of Electrical and Systems Engineering

Graduate Group Chairperson

Alejandro Ribeiro, Associate Professor of Electrical and Systems Engineering

Dissertation Committee

Alejandro Ribeiro, Associate Professor of Electrical and Systems Engineering

George J. Pappas, Professor of Electrical and Systems Engineering

Vijay Kumar, Professor of Mechanical Engineering and Applied Mechanics

Yuliy Baryshnikov, Professor of Mathematics and Electrical and Computer Engineering,
University of Illinois at Urbana-Champaign

CLUSTERING-BASED ROBOT NAVIGATION AND CONTROL

COPYRIGHT

2016

Omur Arslan

To my family

Acknowledgments

This thesis would not have been possible without the encouragement and support of many people. It is a pleasure to thank those who made this unforgettable journey possible.

First and foremost, I would like to express my sincere appreciation and gratitude to my advisor, Daniel E. Koditschek, for his endless support, patience, and encouragement throughout my graduate studies. It was a great opportunity to work under his guidance and to learn from his research expertise. His constant enthusiasm and dedication to basic research has been a source of inspiration to reach for excellence in my own work.

I would also like to thank Alejandro Ribeiro, George Pappas, Vijay Kumar, and Yuliy Baryshnikov for serving on my dissertation committee, despite their busy schedules, and for their encouragement and insightful comments, but also for their hard questions. I am always deeply impressed with and inspired by the quality of their research. My special thanks go to Yuliy for introducing me to the space of trees and for discussions on clustering methods and the topology of configuration spaces. I would also like to express my deepest gratitude to my former advisor, Uluc Saranli, for encouraging and inspiring me to pursue a doctoral degree and providing guidance to achieve my goals.

I would like to gratefully acknowledge the funding sources that made this research work possible. This work was supported in part by the Air Force Office of Scientific Research under CHASE MURI FA9550-10-1-0567 and in part by the Office of Naval Research under the HUNT MURI N00014-07-0829, and I greatly appreciate the generous support of the University of Pennsylvania through the Alfred Fitler Moore Chair Endowment. The support from AFOSR was not only financial — we also have a great opportunity to collaborate with the researchers at the Air Force Research Laboratory. In particular, I would like to thank Jared Culbertson and Bernard Abayowa for our intellectually stimulating conversations.

I would also like to thank my fellow lab members, past and present, for many intellectual and scientific discussions and for their assistance and support over the years: Dan P. Guralnik, Haldun Komsuoglu, Paul Reverdy, Aaron Johnson, Avik De, Gavin Kenneally, Sonia Roberts, Jeff Duperret, Turner Topping, Feifei Qian, Vasileios Vasilopoulos, Wei-Hsi Chen, and Arunkumar Byravan. Especially, I am very much indebted to Dan Guralnik for many fruitful and enlightening discussions and his critical and very valuable feedback to my work. I am very thankful for his guidance, encouragement and support. I would also like to thank our lab coordinator, Diedra Krieger, for her kind assistance.

Of course, getting through graduate school requires more than academic support, and colleagues and friends made my life at Penn much more enjoyable and valuable. I thank Nikolay Atanasov, Shahin Shahrampour, Shaudi Mahdavi, and Hadi Afrasiabi for their personal support to cope with the ups and downs of graduate school and for the great times

we worked together on projects and assignments. I thank Santiago Segarra and Santiago Paternain for our numerous useful discussions on clustering and robot motion planning. I thank Selman Sakar, Ceyhun Eksin, Tolga Ozaslan, Basak Taraktas, Selman Erol, Fatih Karahan and Fazil Pac, whose company and great conversation over bad coffee helped me a lot to endure and survive the graduate school experience. I cannot fully express my gratitude and appreciation for their friendship.

Last but not the least, I would like to thank my family. I must express my very profound gratitude to my parents, Arife and Ali, and to my brother, Onur, and to my sister, Arzu, for their love and for always being there for me in my life and supporting me throughout my doctoral studies. Most importantly, I thank my loving partner, Alyona, for her understanding, kindness and support during the years of this study. This dissertation would not have been possible without her. Thank you!

ABSTRACT

CLUSTERING-BASED ROBOT NAVIGATION AND CONTROL

Omur Arslan
Daniel E. Koditschek

In robotics, it is essential to model and understand the topologies of configuration spaces in order to design provably correct motion planners. The common practice in motion planning for modelling configuration spaces requires either a global, explicit representation of a configuration space in terms of standard geometric and topological models, or an asymptotically dense collection of sample configurations connected by simple paths, capturing the connectivity of the underlying space. This dissertation introduces the use of clustering for closing the gap between these two complementary approaches. Traditionally an unsupervised learning method, clustering offers automated tools to discover hidden intrinsic structures in generally complex-shaped and high-dimensional configuration spaces of robotic systems. We demonstrate some potential applications of such clustering tools to the problem of feedback motion planning and control.

The first part of the dissertation presents the use of hierarchical clustering for relaxed, deterministic coordination and control of multiple robots. We reinterpret this classical method for unsupervised learning as an abstract 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. Based on this new abstraction and a careful topological characterization of the associated hierarchical structure, a provably correct, computationally efficient hierarchical navigation framework is proposed for collision-free coordinated motion design towards a designated multirobot configuration via a sequence of hierarchy-preserving local controllers.

The second part of the dissertation introduces a new, robot-centric application of Voronoi diagrams to identify a collision-free neighborhood of a robot configuration that captures the local geometric structure of a configuration space around the robot's instantaneous position. Based on robot-centric Voronoi diagrams, a provably correct, collision-free coverage and congestion control algorithm is proposed for distributed mobile sensing applications of heterogeneous disk-shaped robots; and a sensor-based reactive navigation algorithm is proposed for exact navigation of a disk-shaped robot in forest-like cluttered environments.

These results strongly suggest that clustering is, indeed, an effective approach for automatically extracting intrinsic structures in configuration spaces and that it might play a key role in the design of computationally efficient, provably correct motion planners in complex, high-dimensional configuration spaces.

Contents

Acknowledgments	iv
Abstract	vi
Contents	vii
List of Tables	x
List of Figures	xi
1 Introduction	1
1.1 Motivation	1
1.2 What Does Clustering Offer?	2
1.3 Prior Literature on the Use of Clustering in Robot Motion Planning	4
1.4 Contributions and Organization of the Thesis	6
2 Coordinated Robot Navigation via Hierarchical Clustering	9
2.1 Hierarchical Abstraction	13
2.1.1 Configuration Space	13
2.1.2 Cluster Hierarchies	14
2.1.3 Configuration Hierarchies	15
2.1.4 Graphs On Trees	16
2.2 Hierarchical Navigation Framework	17
2.2.1 Generic Components of Hierarchical Navigation	17
2.2.2 Specification and Correctness of the Hierarchical Navigation Control Algorithm	19
2.3 Hierarchical Navigation of Euclidean Spheres via Bisecting K-means Clustering	21
2.3.1 Hierarchical Strata of $HC_{2-means}$	21
2.3.2 Hierarchy-Preserving Navigation	24
2.3.3 Navigation in the Space of Binary Trees	28
2.3.4 Portal Transformations	30
2.4 Numerical Simulations	33
2.5 Summary	37

3	Navigation in Tree Space	38
3.1	Preliminaries	40
3.1.1	Hierarchies	40
3.1.2	Some Operations on Trees	43
3.1.3	Dissimilarities, Metrics and Ultrametrics	45
3.2	Quantifying Incompatibility	47
3.2.1	The Cluster-Cardinality Distance	47
3.2.2	The Crossing Dissimilarity	50
3.3	Navigation in the Space of Trees	54
3.3.1	A Discrete-Time Dynamical System Perspective	54
3.3.2	Special Crossings of Clusters	54
3.3.3	NNI Control Law	57
3.3.4	Resolving Incompatibilities with the Root Split	65
3.4	The NNI Navigation Dissimilarity	66
3.4.1	Basic Properties	72
3.4.2	Relations with Other Tree Measures	74
3.5	Discussion and Statistical Analysis	77
3.5.1	Consensus Models and Median Trees	77
3.5.2	Sample Distribution of Dissimilarities	77
3.6	Summary	78
4	Sensor-Based Reactive Navigation in Unknown Convex Sphere Worlds	80
4.1	Problem Formulation	82
4.2	Encoding Collisions via Separating Hyperplanes	83
4.2.1	Separating Hyperplane Theorem	83
4.2.2	The Safe Neighborhood of a Disk-Shaped Robot	85
4.3	Robot Navigation via Separating Hyperplanes	87
4.3.1	Feedback Robot Motion Planner	87
4.3.2	Qualitative Properties	88
4.3.3	Extensions for Limited Range Sensing Modalities	91
4.3.4	An Extension for Differential Drive Robots	97
4.4	Numerical Simulations	99
4.5	Summary	101
5	Voronoi-Based Coverage Control of Heterogeneous Disk-Shaped Robots	102
5.1	Coverage Control of Point Robots	103
5.1.1	Location Optimization of Homogeneous Robots	103
5.1.2	Location Optimization of Heterogeneous Robots	105
5.2	Occupancy Defects of Power Diagrams	106
5.3	Combining Coverage Control and Collision Avoidance	106
5.3.1	Encoding Collisions via Body Diagrams	107
5.3.2	Coverage Control of Heterogeneous Disk-Shaped Robots	109
5.3.3	Congestion Control of Unassigned Robots	112
5.3.4	Coverage Control of Differential Drive Robots	113
5.4	Numerical Simulations	116
5.5	Summary	118

6	Conclusions and Future Work	119
A	Coordinated Robot Navigation: Proof Details	123
A.1	Properties of the Hierarchy-Invariant Vector Field	123
A.1.1	An Equivalent System Model	123
A.1.2	Online Sequential Composition of Substratum Policies	126
A.1.3	Qualitative Properties of Substratum Policies	129
A.1.4	Qualitative Properties of Stratum Policies	130
A.1.5	On the Cardinality of Substratum Policies	131
A.1.6	A Set of Useful Observations on Substratum Policies	132
A.2	Proofs	134
A.2.1	Proof of Theorem 2.6	134
A.2.2	Proof of Proposition A.1	134
A.2.3	Proof of Proposition A.2	136
A.2.4	Proof of Proposition A.3	137
A.2.5	Proof of Proposition A.4	138
A.2.6	Proof of Proposition A.5	138
A.2.7	Proof of Proposition A.6	139
A.2.8	Proof of Proposition A.8	140
A.2.9	Proof of Proposition A.7	140
A.2.10	Proof of Proposition A.9	141
A.2.11	Proof of Lemma A.2	142
A.2.12	Proof of Lemma A.3	143
A.2.13	Proof of Lemma A.6	145
A.2.14	Proof of Lemma A.7	147
A.2.15	Proof of Lemma A.8	148
A.2.16	Proof of Lemma A.9	148
A.2.17	Proof of Lemma A.10	148
A.2.18	Proof of Lemma A.11	149
B	On the Optimality of Napoleon Triangles	152
B.1	Torricelli and Napoleon Transformations	153
B.2	Optimality of Napoleon Transformations	157
C	Sensor-Based Reactive Navigation: Supplementary Material	160
C.1	Geometric Interpretation of the Curvature Condition	160
C.2	Uniqueness of Maximum Margin Separating Hyperplanes	162
C.3	On the Jacobian of Metric Projection	163
C.4	Convexity in Polar Coordinates	166
C.5	An Extension for a Discrete-Time Robot Model	168
C.6	Motion Pattern Far Away from the Goal	170
	Bibliography	172

List of Tables

2.1	Constituent Problems of Hierarchical Robot Navigation	11
2.2	Principal Symbols Used Throughout Chapter 2	13
2.3	Hierarchical Navigation Control (HNC) Algorithm	20
2.4	Topological Shape of a Hierarchical Stratum	24
2.5	Hierarchy-Preserving Navigation Vector Field	25
2.6	Nearest Neighbor Interchange (NNI) Control Law	29
3.1	Skewness and Kurtosis Values for the Distributions of Tree Measures in $\mathcal{BT}_{[25]}$	78
A.1	Proof Structure of Theorem 2.4 : Logical Dependences	124
A.2	Local Control Policies in a Hierarchical Stratum	126
A.3	Substratum Policy Selection Algorithm	127

List of Figures

1.1	Clustering-based motion planning: A new perspective for closing the gap in modelling configuration spaces between configuration space motion planning and sampling-based motion planning.	2
1.2	A multirobot configuration and its cluster hierarchy	3
1.3	The nearest neighbor interchange (NNI) moves between cluster hierarchies and the NNI graph of hierarchies	3
1.4	An application of robot-centric Voronoi diagrams for identifying a collision-free region around a robot in environments cluttered with obstacles	4
2.1	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).	10
2.2	An illustration of (a) a heterogeneous configuration of unit disks in $\text{Conf}(\mathbb{R}^2, [6], \mathbf{1})$ and (b) its iterative 2-mean clustering 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 - $\text{Pr}(I, \tau)$, children - $\text{Ch}(I, \tau)$, 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 $\mathcal{C}(\tau) = \{\{1\}, \{2\}, \dots, \{6\}, \{1, 6\}, \{3, 5\}, \{2, 4\}, \{1, 3, 5, 6\}, \{1, 2, 3, 4, 5, 6\}\}$. . .	14
2.3	The Quotient Space $\text{Conf}(\mathbb{C}, [3], \mathbf{0})/\sim$, where for any $\mathbf{x}, \mathbf{y} \in \text{Conf}(\mathbb{C}, [3], \mathbf{0})$, $\mathbf{x} \sim \mathbf{y} \iff \frac{x_3 - x_1}{x_2 - x_1} = \frac{y_3 - y_1}{y_2 - y_1}$. Here, point particle configurations are quotiented out by translation, scale, and rotation and, therefore, $\mathbf{x}_1 = 0 + 0i$, $\mathbf{x}_2 = 1 + 0i$ and $\mathbf{x}_3 \in \mathbb{C} \setminus \{\mathbf{x}_1, \mathbf{x}_2\}$. Regions are colored according to the associated cluster hierarchies that results from their iterative 2-mean clustering. For instance, any configuration in the white region supports all hierarchies in $\mathcal{BT}_{[3]}$	16
2.4	(left) An illustration of NNI moves between binary trees: each arrow is labeled by a source tree and an associated cluster defining the move. (right) The NNI Graph: a graphical representation of the space of rooted binary trees, \mathcal{BT}_S , with NNI connectivity, where $S = [4] = \{1, 2, 3, 4\}$	17
2.5	Flowchart of the hybrid vector field planner that is guaranteed to bring almost any initial configuration $\mathbf{x} \in \mathfrak{S}(\sigma)$ associated with an arbitrary hierarchy $\sigma \in \mathcal{BT}_S$ to a specified desired configuration $\mathbf{y} \in \mathfrak{S}(\tau)$ supporting a designated hierarchy $\tau \in \mathcal{BT}_S$	20

2.6	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.	22
2.7	The topological shape of a hierarchical stratum intuitively suggests that global navigation in a hierarchical stratum is accomplished by aligning separating hyperplanes of configurations.	23
2.8	An illustration of “sufficiently aligned” separating hyperplanes of complementary clusters I and $I^{-\tau}$ of τ . Both 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.	26
2.9	(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 $\triangle_{A'B'C'}$ 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., $c(\triangle_{ABC}) = c(\triangle_{A'B'C'}) = c(\triangle_{A''B''C''})$	31
2.10	Illustrative navigation trajectory of the hybrid dynamics generated by the HNC algorithm for four 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) Sequence of trees associated with deployed local controllers during the execution of the hybrid navigation controller. (b) Centroidal trajectory of each disk colored according to 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.	34
2.11	Example trajectories of the hybrid vector field planner for (top) 6, (middle) 8, and (bottom) 16 disks in a planar ambient space. (left) Trajectory and (right) 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.	35
2.12	(left) Average normalized navigation path length versus group size, m , and configuration tightness, k . (right) Mean and standard deviation of the normalized navigation path length for configuration tightness $k = 2$	37
3.1	Hierarchical Relations: ancestors - $\text{Anc}(I, \tau)$, parent - $\text{Pr}(I, \tau)$, children - $\text{Ch}(I, \tau)$, descendants - $\text{Des}(I, \tau)$, and local complement (sibling) - $I^{-\tau}$ of cluster I of a rooted binary phylogenetic tree, $\tau \in \mathcal{BT}_{[13]}$. Filled and unfilled circles represent interior and leaf nodes, respectively. An interior node is referred to by its cluster, the list of leaves below it; for example, $I = \{4, 5, 6, 7\}$. Accordingly, the cluster set of τ is $\mathcal{C}(\tau) = \{\{1\}, \{2\}, \dots, \{13\}, \{1, 2\}, \{1, 2, 3\}, \{4, 5\}, \{6, 7\}, \{4, 5, 6, 7\}, \{1, 2, \dots, 7\}, \{9, 10\}, \{8, 9, 10\}, \{11, 12\}, \{11, 12, 13\}, \{8, 9, \dots, 13\}, \{1, 2, \dots, 13\}\}$	41

3.2	(left) An illustration of NNI moves between binary trees, each move (arrow) is labeled by its source tree and the grandchild defining the move. (right) The NNI Graph: a representation of the space of rooted binary trees, \mathcal{BT}_S , with NNI connectivity, for $S = [4] = \{1, 2, 3, 4\}$	43
3.3	d_{CM} and d_{nav} are not metrics: an example of the triangle inequality failing for both dissimilarities.	51
3.4	A pair of nondegenerate hierarchies realizing $\text{diam}(\mathcal{T}_{[n]}, d_{CM}) = (n-2)^2$ and $\text{diam}(\mathcal{BT}_{[n]}, d_{nav}) = \frac{1}{2}(n-1)(n-2)$	52
3.5	An illustration of $\mathcal{J}(\sigma, \tau; K)$ and $\mathcal{D}(\sigma, \tau; K)$ of $\sigma, \tau \in \mathcal{BT}_{[9]}$, and $K = [9] \in \mathcal{K}(\sigma; \tau)$. The vertices and edges associated with clusters of σ incompatible with $\text{Ch}(K, \tau)$ are thickened. The only deep cluster of σ incompatible with $\text{Ch}(K, \tau)$ is $A = \{1, 2\}$, which is also Type 1. B and $B^{-\sigma}$ are examples of Type 2 clusters incompatible with $\text{Ch}(K, \tau)$	56
3.6	A flowchart of navigating from $\sigma \in \mathcal{BT}_S$ towards $\tau \in \mathcal{BT}_S$ based on the NNI control law \mathbf{u}_τ — a nondeterministic hybrid control policy consisting of local controllers $\{\mathbf{u}_{\text{Ch}(J, \tau)}\}_{J \in \mathcal{C}(\tau), J > 1}$	57
3.7	An illustration of deep clusters incompatible with $\text{Ch}(K, \tau)$: Type 1 (b) and Type 2 (c) incompatibilities with $\text{Ch}(K, \tau)$ (a) of a common cluster $K \in \mathcal{K}(\sigma, \tau)$, and the associated NNI navigation moves until resolving incompatibilities with $\text{Ch}(K, \tau)$. Clusters are colored according to their inclusion relation, and the thickened vertices show a portion of incompatible clusters in $\mathcal{J}(\sigma, \tau; K)$	62
3.8	An NNI navigation path joining σ to τ and associated NNI navigation moves. The NNI move from σ to $\gamma = (\text{NNI} \circ \mathbf{u}_\tau)(\sigma)$ illustrates that d_{CC} and d_{RF} to the desired hierarchy τ might stay the same after an NNI navigation transition.	64
3.9	Empirical distribution of tree dissimilarities in $\mathcal{BT}_{[25]}$: (from left to right) the Robinson-Foulds distance d_{RF} , the matching split distance d_{MS} , the cluster-cardinality distance d_{CC} , the crossing dissimilarity d_{CM} , and the NNI navigation dissimilarity d_{nav} . 100000 sample hierarchies are generated using (a) the uniform and (b) Yule model. The resolutions of histograms of tree measures, from left to right, are 1, 4, 32, 4, 2 unit(s), respectively.	78
4.1	Exact navigation of a disk-shaped robot using separating hyperplanes of the robot body (red at the goal) and convex obstacles (black solid shapes). Separating hyperplanes between the robot and obstacles define an obstacle free convex neighborhood (the yellow region when the robot at the goal) of the robot, and the continuous feedback motion towards the metric projection of a given desired goal (red) onto this convex set asymptotically steers almost all robot configurations (green) to the goal without collisions along the way. The grey regions represent the augmented workspace boundary and obstacles, and the arrows show the direction of the resulting vector field. . .	81

4.2	Local workspace \mathcal{LW} (yellow) and local free space \mathcal{LF} (green) of a disk-shaped robot (blue) for different sensing modalities: (left) Voronoi-adjacent obstacle sensing, (middle) a fixed radius sensory footprint (red), (right) a limited range line-of-sight sensor (red). The boundary of each generalized Voronoi cell is defined by the maximum margin separating hyperplanes of the robot body (blue) and obstacles (black).	85
4.3	Example navigation trajectories of the “move-to-projected-goal” law, starting at a set of initial configurations (green) towards a designated point goal (red), for different sensing and actuation models: (top) a fully actuated robot, (bottom) a differential drive robot, (left) Voronoi-adjacent obstacle sensing, (center) a fixed radius sensory footprint, (right) a limited range LIDAR sensor.	99
4.4	Example navigation trajectories of the “move-to-projected-goal” law starting at a set of initial positions (green) far away from the goal (red).	100
4.5	The Euclidean distance, $\ \Pi_{\mathcal{LF}(x)}(x^*) - x^*\ $, between the projected goal, $\Pi_{\mathcal{LF}(x)}(x^*)$, and the global goal, x^* , for different sensing modalities: (left) Voronoi-adjacent obstacle sensing, (middle) a fixed radius sensory footprint, (right) a limited range line-of-sight sensor.	100
5.1	An illustration of (left) the Voronoi and (middle) power diagrams of an environment based on a noncolliding placement of point robots, where the weights of power cells are shown in parentheses. Although each point robot is always contained in its Voronoi cell, power cells associated with some robots (e.g. the 7th robot) may be empty and/or some robots (e.g. the 1st and 4th robots) may not be contained in their nonempty power cells. (right) A collision free disk configuration does not necessarily have Voronoi cells containing respective robot bodies.	103
5.2	(left) Encoding collision free configurations via body diagrams: A configuration of disks is nonintersecting iff each disk is contained in the interior of its body cell. (right) Free subcells, obtained by eroding each body cell with its associated disk radius.	108
5.3	Avoiding collisions around a concentrated event distribution. (top, left) Spatial event distribution, (bottom, left) Initial configuration of a homogeneous robot network, where the weight of sensory cell are shown in the parenthesis, and the resulting trajectories of (top, center) the standard “move-centroid” law, (bottom, center) the “move-to-constrained-centroid” law, (top, right) the “move-to-constrained-active-centroid” law, (bottom, right) the “move-to-constrained-centroid” law of differential drive robots which are initially aligned with the horizontal axis.	117

5.4	Safe coverage control of heterogeneous disk-shaped robots with a heuristic management of unassigned robots. (top, left) Spatial event distribution, (top, bottom) Initial configuration of a heterogeneous robot network, where the weight of sensory cell are shown in the parenthesis, and the resulting trajectories of (top, middle) the standard “move-centroid” law, (bottom, middle) the “move-to-constrained-centroid” law, (top, right) the “move-to-constrained-active-centroid” law, (bottom, right) the “move-to-constrained-active-centroid” law of differential drive robots which are initially aligned with the horizontal axis.	118
B.1	An illustration of (left) the Torricelli point T , the outer Torricelli configuration with \triangle_{ATB_TCT} and the outer Napoleon triangle $\triangle_{ANB_NC_N}$, and (right) the inner Torricelli configuration with $\triangle_{ATB_TC_T}$ and the inner Napoleon triangle $\triangle_{ANB_NC_N}$ of a triangle \triangle_{ABC} . Note that centroids of the vertices of Torricelli configurations, Napoleon triangles and the original triangle all coincide, i.e., $c(\triangle_{ABC}) = c(\triangle_{ATB_TCT}) = c(\triangle_{ANB_NC_N}) = c(\triangle_{ATB_TC_T}) = c(\triangle_{ANB_NC_N})$	153
B.2	(left) Outer, $\triangle_{ANB_NC_N}$, and double outer, $\triangle_{ADBD_C D}$, Napoleon transformations of a triangle \triangle_{ABC} . (right) The double outer Napoleon triangle $\triangle_{ADBD_C D}$ is a convex combination of the original triangle \triangle_{ABC} and the vertex set of its inner Torricelli configuration $\triangle_{ATB_TC_T}$	157
C.1	Convexity in polar coordinates. A polar curve is convex (concave) with respect to the pole iff its epigraph (hypograph) is a convex set, as illustrated on the right (left, respectively).	166
C.2	Segmentation of a LIDAR scan into convex polar curves using convexity measures Γ and Υ	168
C.3	Example navigation paths of the discrete-time “move-to-projected-goal” law for different sampling times and sensing models: (left) $\Delta t = 1$, (middle) $\Delta t = 0.5$, and (right) $\Delta t = 0.25$; and (top) Voronoi-adjacent obstacle sensing, and (center) a fixed radius sensory footprint, and (bottom) a limited range line-of-sight sensor.	169
C.4	Example navigation trajectories of the “move-to-projected-goal” law starting at a set of initial conditions (green) far away from the goal (red) for different sensing and actuation models: (a,b,c) a fully actuated robot, (d,e,f) a differential drive robot, (a,d) Voronoi-adjacent obstacle sensing, (b,e) a fixed radius sensory footprint, (c,f) a limited range line-of-sight sensor.	171

Chapter 1

Introduction

1.1 Motivation

With the increasing use of robots in our daily lives, from household applications [97] to elder/patient assistance [210] to self-driving vehicles [188], it has become even more crucial for autonomous robotics systems to be able to safely move in their workspaces in order to accomplish given tasks. The design of such provably correct safe motion planners inevitably requires to model and understand the topologies of configuration spaces of robotics systems. Two commonly encountered approaches to tackle the safe robot navigation problem are *configuration space motion planning* and *sampling-based motion planning* [140, 55].

The *configuration space*¹ of a robotic system is a mathematical abstract formalism of all possible allowed robot states that are free of any collisions and satisfy the kinematic and dynamic system constraints [152]. Once an explicit representation of the robot’s configuration space is obtained, a number of configuration space motion planners [140, 55], such as discrete planners, cell decomposition and roadmap methods, and feedback motion planners, can be used to safely steer the robot toward its target configuration, satisfying the given task specifications. However, configuration spaces generally have complex shapes and are difficult, if not impossible, to explicitly describe in terms of standard geometric and topological models. Moreover, the complexity of motion planning is known to grow exponentially as the configuration space grows in dimension [46]. These limitations therefore restrict the applicability of configuration space planners to low dimensional settings.

Alternatively, sampling-based methods, such as probabilistic roadmaps [126], rapidly-exploring random trees [139], and their variants, resolve such limitations by producing (open-loop) navigation paths based on randomly sampled robot configurations and simple connectivity criteria; for instance, a pair of sample configurations are connected if the straight line joining them is free of collisions. Although they require no explicit constructions of configuration spaces, sampling-based methods strongly rely on fast collision detectors, efficient nearest neighbor and graph search algorithms, effective sampling strategies (especially around narrow passages) and informative metric selection [141, 144].

¹Here we use the informal robotics notion of the term *configuration space* from [152] rather than the mathematician’s formal definition used in topology [76].

In brief, these two widely used motion planning methods fundamentally differ from each other in modelling configuration spaces: on one hand, we have methods based on global, explicit representations of configuration spaces; on the other hand, we have methods based on individual, sample configurations connected by simple paths. Hence, the question that naturally arises is whether it is possible to combine the strengths of these methods in modelling configuration spaces. As an affirmative response, in this dissertation we propose the use of *clustering* for closing the gap between these two complementary motion planning methods to take advantages of both constructions, see Figure 1.1. We argue that the intrinsic local structures in configuration spaces that are identified by clustering can be exploited to design computationally efficient, provably correct feedback motion planners.

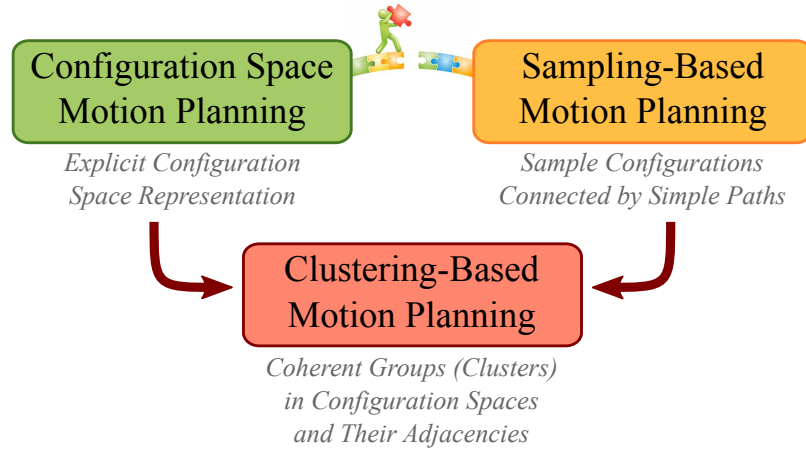


Figure 1.1: Clustering-based motion planning: A new perspective for closing the gap in modelling configuration spaces between configuration space motion planning and sampling-based motion planning.

1.2 What Does Clustering Offer?

Automatic Discovery of Structural Patterns Traditionally an unsupervised learning method, *clustering* offers automated tools to discover coherent groups (clusters) in configuration spaces to model their unknown global organizational structure (e.g., hierarchical clustering, see Figure 1.2), and to determine the local intrinsic structure of configuration spaces around a robot configuration (e.g., partitional clustering, see Figure 1.4) [115]. Since explicit modelling the global geometry and topology of configuration spaces arising in robot motion planning is generally known to be very difficult [140], understanding the local topology of robot configurations sharing the same cluster structure might hold a significant value for simplifying the associated robot navigation problem. In particular, one can use clustering as a magnifying glass to explicitly understand the configuration space topology locally, and, perhaps, exploit the topology of configuration clusters to design provably correct computationally efficient local motion planners to navigate between structurally similar configurations while preserving the shared cluster structure.

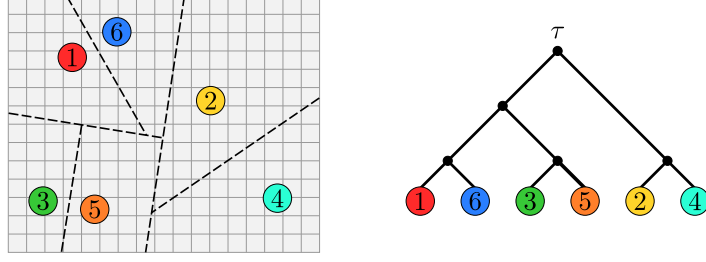


Figure 1.2: A multirobot configuration and its cluster hierarchy

Clustering-Based Symbolic Abstractions A unique strength of configuration space clustering over arbitrary configuration space partitioning, such as cellular decompositions, is that clustering not only yields a cover of a configuration space in terms of configuration clusters, but also relates each covering element to a *clustering model* (e.g., cluster hierarchies and finite set partitions) that deciphers the structural properties of the associated configuration cluster [116, 4]. Hence, on a more conceptual level, clustering can be viewed as a symbolic abstraction [30] relating the continuous space of configurations to the (combinatorial) space of clustering models.

In robot motion planning, discrete abstractions are generally utilized to focus on the high-level aspects of motion design, for instance, how to compose available local behaviours or local motion planners to achieve a given global task, while hiding its continuous low-level details, like collision avoidance, trajectory following and stability that are handled by local planners [30]. Such discrete abstractions mostly represent possible transitions between local planners by means of an *adjacency graph*, instead of an explicit mathematical relation, and a high-level discrete plan is therefore constructed usually by employing standard graph search algorithms, like the A* or Dijkstra’s algorithm [59]. Here, a potential advantage of a clustering-based symbolic abstraction is that the space of the associated clustering models is commonly endowed with some explicit connectivity criteria; for example, restructuring operations of cluster hierarchies [80, Chapter 4], see Figure 1.3. Hence, instead of using a graph search algorithm, one can exploit explicit relations between clustering models to reduce the complexity of high-level motion planning.

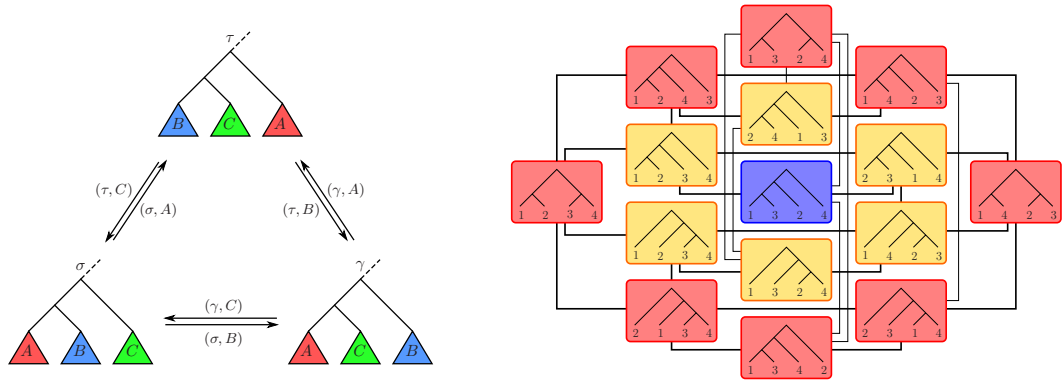


Figure 1.3: The nearest neighbor interchange (NNI) moves between cluster hierarchies and the NNI graph of hierarchies

Locality Identification Another characteristic use of clustering is for locality identification. As opposed to the usual practice of defining a local neighborhood of a robot in terms of simple geometric shapes (e.g., an Euclidean ball around the robot), one can utilize clustering to identify a collision-free neighborhood of a robot that also captures the local geometric structure of the associated configuration space around the robot’s instantaneous position, as illustrated in Figure 1.4. Therefore, one can leverage clustering for designing effective collision avoidance algorithms and provably correct navigation algorithms in cluttered environments.

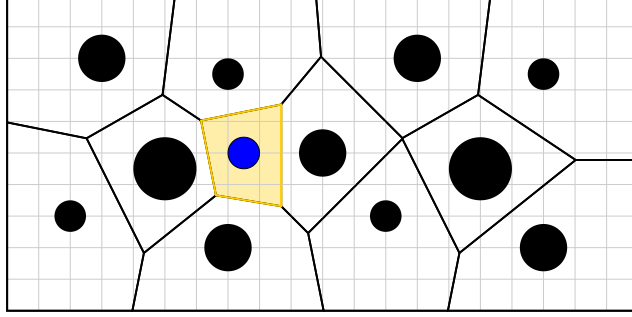


Figure 1.4: An application of robot-centric Voronoi diagrams for identifying a collision-free region (yellow) around the robot body (blue) in environments cluttered with obstacles (black)

1.3 Prior Literature on the Use of Clustering in Robot Motion Planning

A commonly encountered approach in robot motion design that is strongly related to clustering is a *spatial decomposition* of a robot’s workspace for modelling its connectivity and for increasing the computational performance of a motion planner by substantially reducing the associated search space [140, 55]. For example, given its explicit representation, cell decomposition [51, 54] and roadmap [165, 53] methods typically construct a global, one-dimensional graphical representation (skeleton) of an environment based on its trapezoidal [68] or Voronoi [167] decomposition, and seek for a connected path in this skeleton to navigate a robot between any source-destination pair. Hierarchical decomposition methods based on quadrees [84] and octrees [158] are also successfully applied for representing environments at multiple resolutions via adaptive cells, and, in particular, their recursive constructions yield computationally efficient solutions for robots operating in unknown and/or sparse environments, with a substantial reduction in computations and memory requirements compared to a fixed resolution grid representation [120, 225, 41, 227, 79, 107]. In brief, such motion planners make use of spatial decomposition methods for building efficient data structures that approximately model environments, independent of any specific robot configuration and model, whereas our use of clustering aims to explicitly extract immanent local geometric and topological structures in configuration spaces around a given robot configuration. What is more, proximity search in configuration spaces arising in mo-

tion planning is generally complicated due to topological irregularities, and kd-trees [31, 88] are extended to handle various topologies of configuration spaces with increased performance in nearest neighbor search for sampling-based motion planning [226]. To mitigate another bottleneck in the performance of sampling-based motion planning, hierarchical aggregation of workspace obstacles are utilized to effectively sample configurations at various resolutions [91]. Although such studies of individual and/or pairs of configurations are critical for sampling-based motion planning, we believe that automatic identification of structurally similar groups (clusters) in configuration spaces and their geometric and topological properties might be beneficial for a broader family of motion design approaches, including sampling-based methods.

Clustering has also played a key role in the design of scalable algorithms for motion planning and control of large groups of robots, because coordinated motion planning of independent thick bodies in a compact space is known to be computationally hard [203, 111]. Prior work on this problem had already demonstrated that a hierarchy of proximities abstracting the spatial structure of an environment might have a significant value for producing practical coordination strategies that scale well with the number of robots [147, 148, 175]. Hierarchical coordination strategies that divide a large group of robots into small teams and limit coordination across them have been also shown to alleviate the combinatorial growth of complexity by simplifying the systematic enumeration of group interactions [21, 48]. 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 [29], and also find applications for congestion avoidance in swarm navigation [93]. Alternatively, in this dissertation, we show that clustering also achieves a significant reduction in the complexity of coordinated motion design by limiting group interactions in a principled way at multiple levels, based on the spatial distributions of robots, and by using the topological and geometric properties of discovered organizational models.

Pattern formation in biology [92, 173, 86, 61, 69, 7, 194, 23, 110] has recently inspired a great deal of interest in robotics, yielding a growing literature on group coordination behaviors [134, 191, 190, 113] 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. Group coordination via splitting and merging behaviours creates effective strategies for obstacle avoidance [166], congestion control [93], herding [48], area exploration [48, 65], and maintaining persistent and coherent groups while adapting to the environment [113]. In this regard, hierarchical clustering offers an interesting means of ensemble task encoding and control; especially, the ability to specify organizational structure in the precise but flexible terms that hierarchy permits enables us to specify group coordination behaviours at selectively multiple resolutions for safe multi-robot navigation. It also bears mentioning that, motivated by both biological pattern formation [23, 110] and facility localization [167, 72], Voronoi diagrams are widely applied in robotics for coverage control of distributed mobile sensor networks for both solving sensory task assignment and encoding robots' diversity in actuation, sensing, computation, communication and energy resource [60, 137, 178]. In addition to these standard usages in robotics, we tailor Voronoi diagrams to exactly encode robot collisions and to determine a safe neighborhood of a robot that specifies the spatial structure around the robot.

1.4 Contributions and Organization of the Thesis

Clustering-based modelling of configuration spaces, which comprises the central contribution of this dissertation, offers a fresh perspective for computationally efficient provably correct robot motion design. We below present a brief overview of the chapters demonstrating some nontrivial examples of this new perspective to robot motion design.

Chapter 2: Inspired by the use of clustering for modelling global organizational structures, in this chapter we introduce a novel abstraction for ensemble task encoding and control of multirobot systems in terms of hierarchical clustering, yielding precise yet flexible organizational specifications at selectively multiple resolutions. 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 reveal that for a choice of a hierarchical clustering method, the homotopy model of multirobot configurations sharing the same cluster hierarchy is a generalized torus. By exploiting of the underlying topology, for m perfectly sensed and actuated disk-shaped robots in a n -dimensional ambient space (for arbitrary m and n), we propose a provably correct, computationally effective (quadratic in m and algebraic in n) hierarchical navigation framework for collision-free coordinated motion design toward a designated multirobot configuration via an online sequential composition of hierarchy-preserving local controllers, followed by a discrete navigation rule in the space of cluster hierarchies.

In summary, the main contributions of the chapter are:

- a novel abstraction for ensemble task encoding and control in terms of hierarchical clustering,
- an explicit topological characterization of multirobot configurations sharing the same clustering hierarchy,
- 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, this chapter breaks new ground by introducing a topologically nontrivial symbolic abstraction that reduces the complexity of high level planning in the abstract symbol space [30].

Chapter 3: To enable controllable transitions between different cluster hierarchies (and so the associated local controllers of Chapter 2), in this chapter we design a computationally efficient recursive algorithm for navigating through the space of cluster hierarchies of n leaves that is guaranteed to reach a desired hierarchy in $O(n^2)$ steps, each step costing $O(n)$ computations. In Chapter 2, such a high-level discrete plan in the space of cluster hierarchies is used to specify how a multirobot configuration should structurally evolve while navigating toward a desired goal configuration.

Navigation in tree space not only plays a key part in our solution to the multirobot motion planning problem in Chapter 2, but also has a significant value for efficient discriminative comparison of cluster hierarchies — a fundamental classification problem common to

both computational biology and pattern recognition. In this chapter, we also introduce and study three new measures for efficient discriminative comparison of cluster hierarchies. The *NNI navigation dissimilarity* is constructed as the length of a navigation path in the nearest neighbor interchange (NNI) graph of binary hierarchies. It is both an efficient approximation to the (NP-hard) NNI distance and, along with this “edit length” interpretation, its sensitivity to the number of mutually incompatible clusters lends it the character of an edge dissimilarity measure as well. A relaxation of the closed form NNI navigation dissimilarity expression yields a second, simpler measure on all trees — the *crossing dissimilarity* — which explicitly counts pairwise cluster incompatibilities between trees. Both dissimilarities are symmetric and positive definite (vanish only between identical trees) on binary hierarchies but they are not true metrics, because they fail to satisfy the triangle inequality. Nevertheless, both are linearly bounded below by the widely used Robinson-Foulds metric and we achieve a metric upper bound by introducing a true distance, the *cluster-cardinality metric* that is constructed as the pullback of matrix norms along an embedding of trees into the space of matrices. All of these proposed new tree measures can be efficiently computed in time $O(n^2)$ in the number of leaves, n .

In short, this chapter makes the following contributions:

- A recursive algorithm is proposed for efficient navigation in tree space.
- Metric upper and lower bounds on the length of navigation paths are provided.
- Three new tree measures are introduced for efficient discriminative comparison of trees.
- An ordering relation between the proposed tree dissimilarities and related tree metrics in the literature is revealed.
- The discriminative power of the proposed tree measures over existing tree metrics is demonstrated by using a statistical analysis of their empirical distributions.

Chapter 4: Inspired by the use of clustering for locality identification, in this chapter we introduce a novel use of separating hyperplanes for identifying a collision-free convex neighborhood of a robot moving in a cluttered environment, which turns out to be a simple, but effective way of extracting the intrinsic local geometric structure of the configuration space around the robot’s instantaneous position. Accordingly, we construct a sensor-based feedback law that solves the real-time collision-free robot navigation problem in a compact convex Euclidean subset cluttered with sufficiently separated and strongly convex obstacles. We prove that this construction affords a piecewise smooth continuously varied closed-loop vector field whose smooth flow brings almost all configurations in the robot’s free space to a designated goal location, with the guarantee of no collisions along the way. We extend these provable properties to practically motivated limited range sensing models, and the nonholonomically constrained kinematics of the standard differential drive vehicle.

The contributions of this chapter are:

- a novel application of separating hyperplanes for encoding robot collisions precisely and identifying a safe convex neighborhood of a robot,
- a provably correct sensor-based vector field planner for safe robot navigation in cluttered environments,
- a new feedback navigation law that requires no parameter tuning and requires only local knowledge of the environment,

- practical extensions to limited range sensing modalities and the widely used differential drive vehicle model, while retaining the provable properties.

Chapter 5: In distributed mobile sensing applications, networks of agents that are heterogeneous, respecting both actuation as well as body and sensory footprint, are often modelled by recourse to power diagrams — generalized Voronoi diagrams with additive weights. In addition to these standard usages, we adapt power diagrams to encode collisions in a heterogeneous group of disk-shaped robots and determine a safe neighborhood of a multirobot configuration, which is another application of clustering for locality identification in robot motion design. Accordingly, based on standard coverage control of point robots [60, 137, 178], we propose a constrained coverage control law for heterogeneous disk-shaped robots that solves the combined sensory coverage and collision avoidance problem. We also present its practical extensions for a heuristic congestion management of unassigned robots and for a lift of the fully-actuated controller to the more practical differential drive kinematics, while maintaining the convergence and collision avoidance guarantees.

The chapter makes the following contributions:

- A new robot-centric application of Voronoi diagrams to precisely encode multirobot collisions is introduced.
- A constrained coverage control law that solves the combined sensory coverage and collision avoidance problem is proposed.
- A congestion management heuristic for unassigned robots is proposed to hasten the assigned robots progress.
- An extension to the widely used kinematic differential drive robots (retaining the provable properties) is presented.

In Chapter 6, we conclude the dissertation with a summary of our contributions and a brief discussion of future work.



Chapter 2

Coordinated Robot Navigation via Hierarchical Clustering

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 [171]. Despite significant progress in the analysis of how local rules can yield such global spatiotemporal patterns [114, 162, 52], 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 instead focus control effort on the presumably far coarser properties of the collective pattern that matter. In this chapter, we seek a more relaxed means of task specification that is sensitive to spatial distribution at multiple scales (as in influencing the intensity of interactions among individuals and with their environment [168]) and the identities of neighbors (as in determining the capabilities of heterogeneous teams [7]), while affording, nevertheless, a well-formed deterministic characterization of pattern.

Biology offers spectacularly diverse examples of animal spatial organization ranging from self-sorting in cells [205], tissues and organs [213, 92], and groups of individuals [69, 6, 100] to more patterned teams [164, 125, 7, 173], all the way through strategic group formations in vertebrates [28, 207], mammals [90, 194, 204, 163], and primates [218, 63] that are hypothesized to increase efficacy in foraging [164, 125], hunting [218, 28, 90, 194], logistics and construction [7, 173], predator avoidance [211, 102], and even to stabilize whole ecologies [89] — 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 among proximal individuals coupled to specific physical environments [86] in a manner that might be posited as a paradigm for generalized emergent intelligence [61]. Second, these formations appear to resist familiar rigid prescriptions that govern 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 [90, 204]), while, nevertheless, supporting the underlying coarse, deterministic “deep structure” as a dynamical invariant. It is this second remarkable attribute of biological swarms that inspires this chapter.

We are led to the notion of hierarchical clustering. We reinterpret this classical method for unsupervised learning [115] 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 Figure 2.1. There, we display three different configurations of five planar disks. Their 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 two 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 four units of distance, the green disk has merged into the subgroup that includes 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 chapter, we introduce a provably correct and computationally effective machinery for specifying, controlling invariantly to, and passing between such hierarchical clusterings at will.

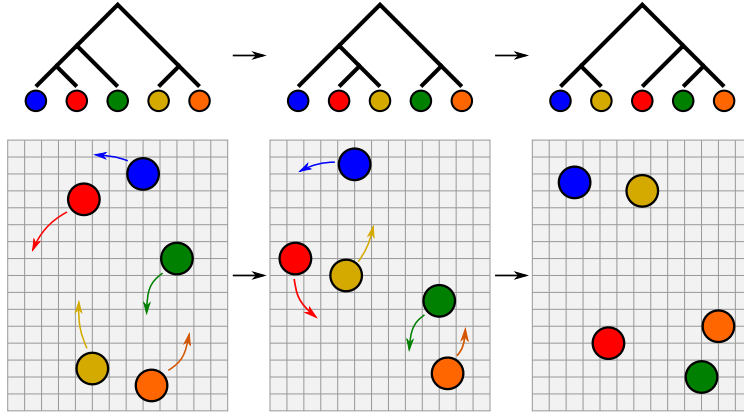


Figure 2.1: Moving from one spatial distribution to another is generally carried through rearrangements of robot groups (clusters) at different resolution corresponding to transitions between different cluster structures (hierarchies).

To illustrate its utility, we use this formalism to solve a specific instance of the reactive motion planning problem that suggests 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 m disk-shaped robots in the n -dimensional Euclidean space \mathbb{R}^n , we presume that a target hierarchy has been specified along with a desired 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 that results in a purely reactive hybrid dynamical system [22] guaranteed to bring the disk robots to both the hierarchical pattern as well as the rigidly specified instance from (almost) arbitrary initial

Table 2.1: Constituent Problems of Hierarchical Robot Navigation

Problem	Solution	Theorem	Description
2.1	Table 2.5	2.4	Hierarchy-invariant vector field planner
2.2	Table 2.6	2.5	Reactive navigation across hierarchies
2.3	Eqn.(2.33)	2.6	Cross-hierarchy geometric realization

conditions, with no collisions of the disks along the way. Stated formally later in Table 2.3, the correctness of this algorithm is guaranteed by Theorem 2.1, whose proof appeals to the resolution of various constituent problems summarized in Table 2.1. 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 formulas 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 this chapter are as follows:

- 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 $n \geq 2$).

On a more conceptual level, this chapter breaks new ground by introducing a topologically nontrivial symbolic abstraction that reduces the complexity of high-level planning in the abstract symbol space [30], 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 [104]. 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.

Complexity of Multirobot Motion Planning Existing work on multirobot motion planning generally suffers from the intrinsic complexity of multibody configurations, which impedes computationally effective generalizations of single-robot motion planners [140, 55]. Coordinated motion planning of thick bodies in a compact space is known to be computationally hard. For example, moving planar rectangular objects within a rectangular box is PSPACE-hard [111], and motion planning for finite planar disks in a polygonal environment is strongly NP-hard [203]. 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 [27]. 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 [172]. Within the domain of reactive or vector field motion planning, which is the main focus of this chapter, 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 [209, 70], as we do in Section 2.3, or simply assuming conditions sufficient to guarantee connectivity between initial and goal configurations [123, 150]. On the other hand, more relaxed versions entailing (perhaps partially) homogeneous¹ (unlabeled) specifications for interchangeable individuals have yielded computationally efficient planners in the recent literature [202, 2, 215, 214], and we suspect that the cluster hierarchy abstraction may be usefully applicable to such partially labeled settings.

Reactive Multirobot Motion Planning Since the problem of reactively navigating groups of disks was first introduced to robotics [221, 222], most research into vector field planners has embraced the navigation function paradigm [183]. A recent review of this two-decade-old literature is provided in [209], 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 [209], the multirobot generalization of a single-agent navigation function is challenged by the violation of certain assumptions inherited from the original formulation [183]. 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 [209]. In some approaches, the departure from isolated interaction has been addressed by encoding all possible collision scenarios, yielding controllers with terms growing superexponentially in the number of robots, even when the workspace is not compact [70]. In contrast, our recourse to the hierarchical representation of configurations affords a computational burden that grows merely quadratically in the number of agents. In [123], 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 (nonintersecting) 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 [123].

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 [150]. In principle, the original results of [183] guarantee that any monotone increasing scheme must eventually resolve the issue of local minima; however, this

¹Following 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 [171, 202, 134]. 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 chapter, we consider fully heterogeneous robot groups, since any method proposed for heterogeneous robots can be easily applied to (partially) homogeneous robots.

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.

This chapter is based on the papers [14, 15, 16, 18, 17]. The journal paper [18] presents a unified view of our hierarchical navigation framework, with some tutorial background, that is developed in preliminary form in the conference paper [16], building on the initial results of the conference paper [15]. The submitted journal paper [17] presents a recursively defined algorithm for navigating in tree space (presented in detail later in Chapter 3) that is used to define a high-level transition rule between (hierarchy-preserving) local navigation controllers whose correct recruitment solves the reactive coordinated motion planning problem. The journal paper [14] studies the optimality of Napoleon triangles, which we utilize to select an optimal multirobot configuration around which instantaneous switching between different cluster structures is feasible.

2.1 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. Table 2.2 summarizes the principal notation that will be used throughout this paper.

Table 2.2: Principal Symbols Used Throughout This Chapter

S, \mathbf{r}	Sets of labels and disk radii [2.1.1]
$\text{Conf}(\mathbb{R}^n, S, \mathbf{r})$	The conf. space of labelled, noncolliding disks (2.1)
\mathcal{BT}_S	The space of binary trees [2.1.2]
HC	Hierarchical clustering [2.1.3]
$\text{HC}_{2\text{-means}}$	Iterative 2-means clustering [2.3]
$\mathfrak{S}(\tau)$	The stratum of a tree, $\tau \in \mathcal{BT}_S$, (2.2)
$\text{Portal}(\sigma, \tau)$	Portal configurations of a pair, (σ, τ) , of trees (2.5)
$\text{Port}_{\sigma, \tau}$	Portal map [2.2.1]
$\mathcal{A}_S = (\mathcal{BT}_S, \mathcal{E}_{\mathcal{A}})$	The adjacency graph of trees [2.1.4]
$\mathcal{N}_S = (\mathcal{BT}_S, \mathcal{E}_{\mathcal{N}_S})$	The nearest neighbor interchange (NNI) graph of trees [2.1.4]

2.1.1 Configuration Space

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

Given an index set $S = [m] := \{1, \dots, m\} \subset \mathbb{N}$, a *heterogeneous multirobot configuration*, $\mathbf{x} = (x_i)_{i \in S}$, is a labeled nonintersecting placement of $|S| = m$ distinct Euclidean spheres³ of nonnegative radii $\mathbf{r} := (r_i)_{i \in S} \in (\mathbb{R}_{\geq 0})^S$, where i th sphere is centered at $x_i \in \mathbb{R}^n$ and has

²It bears mention in passing that partial differential equations (e.g., harmonic potentials [151]) 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.

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

radius $r_i \geq 0$. We find it convenient to identify the *configuration space* [77] with the set of distinct labelings, i.e., the injective mappings of S into \mathbb{R}^n , and we will find it convenient to denote our “thickened” subset of this configuration space as ⁴

$$\text{Conf}(\mathbb{R}^n, S, \mathbf{r}) := \left\{ \mathbf{x} \in (\mathbb{R}^n)^S \mid \|\mathbf{x}_i - \mathbf{x}_j\| > r_i + r_j, \quad \forall i \neq j \in S \right\}, \quad (2.1)$$

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

2.1.2 Cluster Hierarchies

A rooted semilabelled tree τ over a fixed finite index set S , illustrated in Figure 2.2, is a directed acyclic graph $G_\tau = (V_\tau, E_\tau)$, whose leaves, vertices of degree one, are bijectively labeled by S 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* [35]. 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 chapter, \mathcal{BT}_S denotes the set of rooted nondegenerate trees over leaf set S .

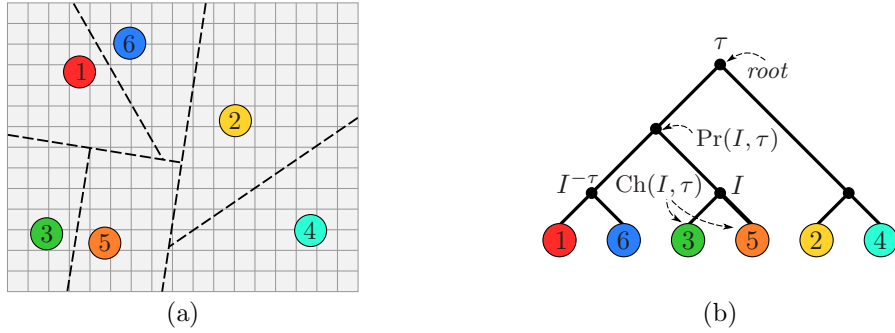


Figure 2.2: An illustration of (a) a heterogeneous configuration of unit disks in $\text{Conf}(\mathbb{R}^2, [6], 1)$ and (b) its iterative 2-mean clustering [193] hierarchy τ in $\mathcal{BT}_{[6]}$, where the dashed lines in (a) depict the separating hyperplanes between clusters, and (b) illustrates hierarchical cluster relations: parent - $\text{Pr}(I, \tau)$, children - $\text{Ch}(I, \tau)$, 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 $\mathcal{C}(\tau) = \{\{1\}, \{2\}, \dots, \{6\}, \{1, 6\}, \{3, 5\}, \{2, 4\}, \{1, 3, 5, 6\}, \{1, 2, 3, 4, 5, 6\}\}$.

A rooted semilabelled tree τ uniquely determines (and henceforth will be interchangeably termed) a *cluster hierarchy* [159]. 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 Figure 2.2). 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) $\text{Pr}(I, \tau)$ and lists of children $\text{Ch}(I, \tau)$, ancestors $\text{Anc}(I, \tau)$ and descendants $\text{Des}(I, \tau)$ of I in τ ,

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

illustrated in Figure 2.2 — see Section 3.1.1 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 $I^{-\tau} := \text{Pr}(I, \tau) \setminus I$.

2.1.3 Configuration Hierarchies

A *hierarchical clustering* ⁵ $\text{HC} \subset \text{Conf}(\mathbb{R}^n, S, \mathbf{r}) \times \mathcal{BT}_S$ is a relation from the configuration space $\text{Conf}(\mathbb{R}^n, S, \mathbf{r})$ to the abstract space of binary trees \mathcal{BT}_S [115], an example depicted in Figure 2.2. In this chapter, we are interested only in clustering methods that can classify all possible configurations (i.e., for which HC assigns some tree to every configuration), and therefore, we need the following:

Property 2.1 *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 [115].

Given such an HC, for any $\mathbf{x} \in \text{Conf}(\mathbb{R}^n, S, \mathbf{r})$ and $\tau \in \mathcal{BT}_S$, we say \mathbf{x} *supports* τ if and only if $(\mathbf{x}, \tau) \in \text{HC}$. The *stratum* associated with a binary hierarchy $\tau \in \mathcal{BT}_S$, denoted by $\mathfrak{S}(\tau) \subset \text{Conf}(\mathbb{R}^n, S, \mathbf{r})$, is the set of all configurations $\mathbf{x} \in \text{Conf}(\mathbb{R}^n, S, \mathbf{r})$ supporting the same tree τ [15],

$$\mathfrak{S}(\tau) := \left\{ \mathbf{x} \in \text{Conf}(\mathbb{R}^n, S, \mathbf{r}) \mid (\mathbf{x}, \tau) \in \text{HC} \right\}, \quad (2.2)$$

and this yields a tree-indexed cover of the configuration space. For purposes of illustration, we depict in Figure 2.3 the strata of $\text{Conf}(\mathbb{C}, [3], \mathbf{0})$ — a space that represents a group of three point particles on the complex plane. ^{6 7}

The restriction to binary trees precludes combinatorial tree degeneracy [35] and we will avoid configuration degeneracy by imposing the following.

Property 2.2 *Each stratum of HC includes an open subset of configurations, i.e., for every $\tau \in \mathcal{BT}_S$, $\mathring{\mathfrak{S}}(\tau) \neq \emptyset$.*⁸

Once again, most standard 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 [223].

Given any two multirobot configurations supporting the same cluster hierarchy, moving between them while maintaining the shared cluster hierarchy (introduced later as Problem 2.1) requires the following:

Property 2.3 *Each stratum of HC is connected.*

⁵Although 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 [115, 223].

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

⁷Note that the quotient space in Figure 2.3 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.

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

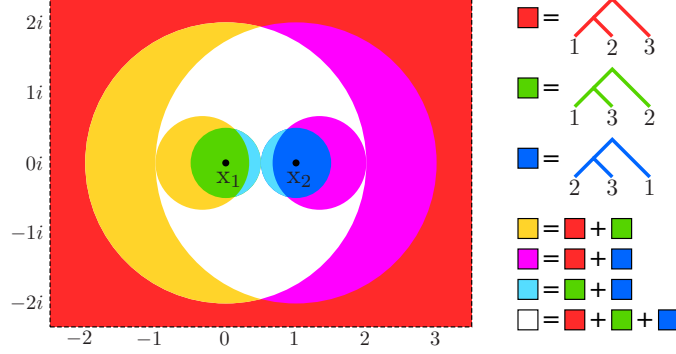


Figure 2.3: The Quotient Space $\text{Conf}(\mathbb{C}, [3], \mathbf{0}) / \sim$, where for any $\mathbf{x}, \mathbf{y} \in \text{Conf}(\mathbb{C}, [3], \mathbf{0})$, $\mathbf{x} \sim \mathbf{y} \iff \frac{x_3 - x_1}{x_2 - x_1} = \frac{y_3 - y_1}{y_2 - y_1}$. Here, point particle configurations are quotiented out by translation, scale, and rotation and, therefore, $\mathbf{x}_1 = 0 + 0i$, $\mathbf{x}_2 = 1 + 0i$ and $\mathbf{x}_3 \in \mathbb{C} \setminus \{\mathbf{x}_1, \mathbf{x}_2\}$. Regions are colored according to the associated cluster hierarchies that results from their iterative 2-mean clustering [193]. For instance, any configuration in the white region supports all hierarchies in $\mathcal{BT}_{[3]}$.

For an arbitrary clustering method, this requirement is generally not trivial to show, but when configuration clusters of HC are linearly separable, one can characterize the topological shape of each stratum to verify this requirement, as we do in Section 2.3.1.

2.1.4 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* $\mathcal{A}_S = (\mathcal{BT}_S, \mathcal{E}_A)$ to be the 1-skeleton of the nerve [104] of the $\text{Conf}(\mathbb{R}^n, S, \mathbf{r})$ -cover induced by HC. That is to say, a pair of hierarchies $\sigma, \tau \in \mathcal{BT}_S$ is connected with an edge in \mathcal{E}_A if and only if their strata intersect, $\mathfrak{S}(\sigma) \cap \mathfrak{S}(\tau) \neq \emptyset$. To enable navigation between structurally different multirobot configurations later (Problem 2.2), we need the following:

Property 2.4 *The adjacency graph is connected.*

Although the adjacency graph is a critical building block of our abstraction, as Figure 2.3 anticipates, HC strata generally have complicated shapes, making it usually hard to compute the complete adjacency graph. Fortunately, the computational biology literature [80] 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}_S , as follows.

The *nearest neighbor interchange (NNI)* move at a cluster $A \in \mathcal{C}(\sigma)$ on a binary tree $\sigma \in \mathcal{BT}_S$, as illustrated in Figure 2.4(left), swaps cluster A with its parent's sibling $C = \text{Pr}(A, \sigma)^{-\sigma}$ to yield another binary tree $\tau \in \mathcal{BT}_S$ [184, 160]. Say that $\sigma, \tau \in \mathcal{BT}_S$ 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 that join them can be determined by identifying their unshared clusters [17]. Moreover, define the *NNI graph* $\mathcal{N}_S = (\mathcal{BT}_S, \mathcal{E}_{\mathcal{N}_S})$ to have vertex set \mathcal{BT}_S , with two trees connected by an

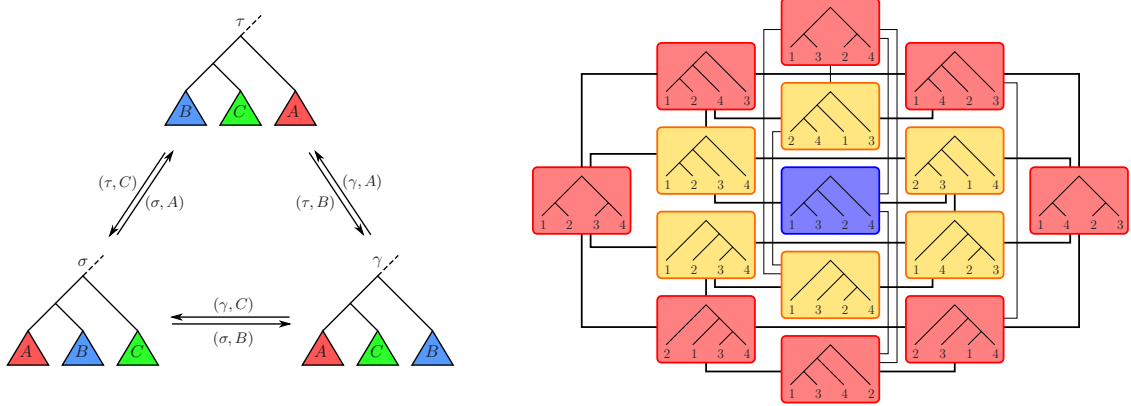


Figure 2.4: (left) An illustration of NNI moves between binary trees: each arrow is labeled by a source tree and an associated cluster defining the move. (right) The NNI Graph: a graphical representation of the space of rooted binary trees, \mathcal{BT}_S , with NNI connectivity, where $S = [4] = \{1, 2, 3, 4\}$.

edge in $\mathcal{E}_{\mathcal{N}_S}$ if and only if they are NNI-adjacent (see Figure 2.4(right)). An important contribution of this chapter will be to show how the NNI graph yields a computationally effective subgraph of the adjacency graph (Theorem 2.6) for our preferred choice of HC.

2.2 Hierarchical Navigation Framework

Hierarchical abstraction introduced in Section 2.1 intrinsically suggests a two-level navigation strategy for coordinated motion design: 1) At the low-level, perform finer adjustments on configurations using hierarchy-preserving vector fields; and 2) At the high-level, resolve structural conflicts between configurations using a discrete transition policy in tree space. The connection between these two levels is 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.

2.2.1 Generic Components of Hierarchical Navigation

Hierarchy-Preserving Navigation

For ease of exposition, we restrict our 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

$$\dot{\mathbf{x}} = f(\mathbf{x}) , \quad (2.3)$$

where $f : \mathbf{Conf}(\mathbb{R}^n, S, \mathbf{r}) \rightarrow (\mathbb{R}^n)^S$ is a vector field over $\mathbf{Conf}(\mathbb{R}^n, S, \mathbf{r})$ (see (2.1)).

⁹A long prior robotics literature motivates the utility of this fully actuated “generalized damper” dynamical model [153] and provides methods for “lifts” to controllers for second-order plants [130, 131] as well.

Denote by φ^t the *flow* [8] on $\mathbf{Conf}(\mathbb{R}^n, S, \mathbf{r})$ induced by the vector field f . For a choice of hierarchical clustering HC, the class of hierarchy-invariant vector fields maintaining the robot group in a specified hierarchical arrangement of clusters $\tau \in \mathcal{BT}_S$ is defined as [15],

$$\mathcal{F}_{\text{HC}}(\tau) := \left\{ f : \mathbf{Conf}(\mathbb{R}^n, S, \mathbf{r}) \rightarrow (\mathbb{R}^n)^S \mid \varphi^t(\mathfrak{S}(\tau)) \subset \mathfrak{S}(\tau), t > 0 \right\}. \quad (2.4)$$

Hierarchy-preserving navigation, the low-level component of our framework, uses the vector fields of $\mathcal{F}_{\text{HC}}(\tau)$ 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 follows.

Problem 2.1 *For any $\tau \in \mathcal{BT}_S$ and $\mathbf{y} \in \mathfrak{S}(\tau)$ associated with HC, construct a control policy $f_{\tau, \mathbf{y}}$ using the hierarchy invariant vector fields of $\mathcal{F}_{\text{HC}}(\tau)$ whose closed loop asymptotically results in a retraction $R_{\tau, \mathbf{y}}$ of $\mathfrak{S}(\tau)$, possibly excluding a set of measure zero¹¹, 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 $\mathbf{Conf}(\mathbb{R}^n, S, \mathbf{r})$ invariant as well; thus, avoiding 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. There are likely to be many alternative approaches to such constructions; and working with the 2-means hierarchical clustering [193], we construct in [15] such a family of hierarchy preserving control policies for particle configurations, and Section 2.3.2 extends that construction to thickened disk configurations.

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 2.3 replaced with the NNI graph — a computationally efficient connected subgraph). Thus, we further require a construction of a discrete feedback policy in \mathcal{BT}_S that recursively generates paths in the adjacency graph toward any specified destination tree from all other trees in \mathcal{BT}_S by reducing a “discrete Lyapunov function” relative to that destination, summarized as follows.

Problem 2.2 *Given any $\tau \in \mathcal{BT}_S$, 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. Once again, there

¹⁰It 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. For such a general case, one can still construct an exact retraction within our framework.

¹¹ Recall from [78] that a continuous motion planner in a configuration space X exists if and only if X is contractible. Hence, if a hierarchical stratum is noncontractible (see Theorem 2.2), the domain of such a vector field planner described in Problem 2.1 must exclude at least a set of measure zero.

are many alternative ways of constructing such a discrete transition rule, for example, using standard graph search algorithms, like the A* or Dijkstra’s algorithm [59]; and we recently develop in [17] such an efficient recursive procedure, summarized in Section 2.3.3 and presented in detail in Chapter 3, to find paths joining any given pair of trees in the NNI graph — a subgraph of the adjacency graph (Theorem 2.6).

Hierarchical Portals

Here, we relate the (combinatorial) topology of hierarchical clusters to the (continuous) topology of configurations by defining “portals” — open sets of configurations that support two adjacent hierarchies.

Definition 2.1 *The portal $\text{Portal}(\sigma, \tau)$ of a pair of hierarchies $\sigma, \tau \in \mathcal{BT}_S$ is the set of all configurations supporting interior strata of both trees,*

$$\text{Portal}(\sigma, \tau) := \mathring{\mathfrak{S}}(\sigma) \cap \mathring{\mathfrak{S}}(\tau) . \quad (2.5)$$

Namely, portals are geometric realizations in the configuration space of the edges of the adjacency graph on trees (see Figure 2.3). 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 2.3 *Given an edge $(\sigma, \tau) \in \mathcal{E}_A$ of the adjacency graph $\mathcal{A}_S = (\mathcal{BT}_S, \mathcal{E}_A)$, construct a geometric realization map $\text{Port}_{(\sigma, \tau)} : \mathfrak{S}(\sigma) \rightarrow \text{Portal}(\sigma, \tau)$ 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” [45] for the sequentially composed local controllers whose correct recruitment solves the reactive coordinated motion planning problem (see Theorem 2.1). As one might expect, there are infinitely many alternative choice of portal configurations as long as the portal of a given pair of trees is nonempty; and we describe in Section 2.3.4 such a portal selection method based on the optimality of Napoleon triangles (see Appendix B).

2.2.2 Specification and Correctness of the Hierarchical Navigation Control Algorithm

Assume the selection of a goal configuration $\mathbf{y} \in \mathfrak{S}(\tau)$ and a hierarchy $\tau \in \mathcal{BT}_S$ that \mathbf{y} supports. Now, given (almost) any initial configuration $\mathbf{x} \in \mathfrak{S}(\sigma)$ for some hierarchy $\sigma \in \mathcal{BT}_S$ that \mathbf{x} supports, Table 2.3 presents the hierarchical navigation control (HNC) algorithm whose flowchart is illustrated in Figure 2.5. 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 follows.

Theorem 2.1 *The HNC Algorithm in Table 2.3 defines a hybrid dynamical system whose execution brings almost every initial configuration ¹¹ $\mathbf{x} \in \text{Conf}(\mathbb{R}^n, S, \mathbf{r})$ in finite time to an arbitrarily small neighborhood of the goal configuration $\mathbf{y} \in \mathfrak{S}(\tau)$ with the guarantee of no collisions along the way.*

Table 2.3: Hierarchical Navigation Control (HNC) Algorithm

-
- For (almost) any initial $\mathbf{x} \in \mathfrak{S}(\sigma)$ and $\sigma \in \mathcal{BT}_S$, and desired $\mathbf{y} \in \mathfrak{S}(\tau)$ and $\tau \in \mathcal{BT}_S$,
1. (Hybrid Base Case) if $\mathbf{x} \in \mathfrak{S}(\tau)$ then apply stratum-invariant dynamics, $f_{\tau,\mathbf{y}}$ (Problem 2.1).
 2. (Hybrid Recursive Step) else,
 - (a) invoke the discrete transition rule g_τ (Problem 2.2) to propose an adjacent tree, $\gamma \in \mathcal{BT}_S$, with lowered discrete Lyapunov value.
 - (b) Choose local configuration goal, $\mathbf{z} := \text{Port}_{(\sigma,\gamma)}(\mathbf{x})$ (Problem 2.3).
 - (c) Apply the stratum-invariant continuous controller $f_{\sigma,\mathbf{z}}$ (Problem 2.1).
 - (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).
-

Proof In the base case, i.e., step 1, the conclusion follows directly from the construction of Problem 2.1: The flow $f_{\tau,\mathbf{y}}$ keeps the state in $\mathfrak{S}(\tau)$ and approaches a neighborhood of \mathbf{y} (which is an asymptotically stable equilibrium state for that flow) in finite time.

In the inductive step 2a, the NNI transition rule g_τ guarantees a decrement in the Lyapunov function after a transition from σ to γ (Problem 2.2), 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 step 2b. Next, the flow $f_{\sigma,\mathbf{z}}$ in step 2c 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

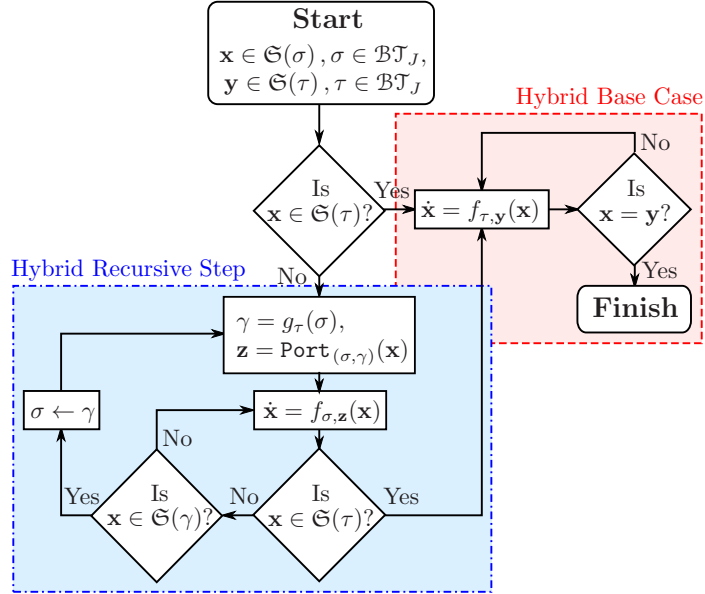


Figure 2.5: Flowchart of the hybrid vector field planner that is guaranteed to bring almost any initial configuration $\mathbf{x} \in \mathfrak{S}(\sigma)$ associated with an arbitrary hierarchy $\sigma \in \mathcal{BT}_S$ to a specified desired configuration $\mathbf{y} \in \mathfrak{S}(\tau)$ supporting a designated hierarchy $\tau \in \mathcal{BT}_S$.

is not triggered in step 2d, 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 step 2a. Because the dynamical transitions g_τ initiated from any hierarchy in \mathcal{BT}_S reaches τ in finite steps (see Problem 2.2), it must eventually trigger the base case. ■

2.3 Hierarchical Navigation of Euclidean Spheres via Bisecting K-means Clustering

We now restrict our attention to 2-means divisive hierarchical clustering [193] $\text{HC}_{2\text{-means}}$ and demonstrate a construction of our hierarchical navigation framework for coordinated navigation of Euclidean spheres via $\text{HC}_{2\text{-means}}$.

2.3.1 Hierarchical Strata of $\text{HC}_{2\text{-means}}$

Iterative 2-means clustering $\text{HC}_{2\text{-means}}$ is a divisive method that recursively constructs a cluster hierarchy of a configuration in a top-down fashion [193]. 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 $\text{HC}_{2\text{-means}}$ are linearly separable by a hyperplane defined by the associated cluster centroids,¹² as illustrated in Figure 2.2, and the stratum of $\text{HC}_{2\text{-means}}$ associated with a binary hierarchy $\tau \in \mathcal{BT}_S$ can be characterized by the intersection of inverse images,

$$\mathfrak{S}(\tau) = \bigcap_{I \in \mathcal{C}(\tau) \setminus \{S\}} \bigcap_{i \in I} \eta_{i,I,\tau}^{-1}[0, \infty) , \quad (2.6)$$

of the scalar valued “separation” function, $\eta_{i,I,\tau} : \text{Conf}(\mathbb{R}^n, S, \mathbf{r}) \rightarrow \mathbb{R}$ [15], which returns the distance of agent i in cluster $I \in \mathcal{C}(\tau) \setminus \{S\}$ to the perpendicular bisector of the centroids of complementary clusters I and $I^{-\tau}$:¹³

$$\eta_{i,I,\tau}(\mathbf{x}) := (\mathbf{x}_i - \mathbf{m}_{I,\tau}(\mathbf{x}))^T \frac{\mathbf{s}_{I,\tau}(\mathbf{x})}{\|\mathbf{s}_{I,\tau}(\mathbf{x})\|} , \quad (2.7)$$

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

$$\mathbf{c}(\mathbf{x}|I) := \frac{1}{|I|} \sum_{i \in I} \mathbf{x}_i , \quad (2.8)$$

$$\mathbf{s}_{I,\tau}(\mathbf{x}) := \mathbf{c}(\mathbf{x}|I) - \mathbf{c}(\mathbf{x}|I^{-\tau}) , \quad (2.9)$$

$$\mathbf{m}_{I,\tau}(\mathbf{x}) := \frac{\mathbf{c}(\mathbf{x}|I) + \mathbf{c}(\mathbf{x}|I^{-\tau})}{2} . \quad (2.10)$$

We now follow [25] in defining terminology and expressions leading to the characteriza-

¹²In the context of self-sorting in heterogeneous swarms [134], two groups of robot swarms are said to be *segregated* if their configurations are linearly separable and, in this regard, configuration hierarchies of $\text{HC}_{2\text{-means}}$ represent spatially cohesive and segregated swarms groups at different resolutions.

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

tion 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 [25], and we omit them for the sake of brevity.

Definition 2.2 A configuration $\mathbf{x} \in \text{Conf}(\mathbb{R}^n, S, \mathbf{r})$ is narrow relative to the split, $\{I, S \setminus I\}$, if

$$\max_{A \in \{I, J \setminus I\}} r(\mathbf{x}|A) < \frac{1}{2} \|c(\mathbf{x}|I) - c(\mathbf{x}|J \setminus I)\|, \quad (2.11)$$

where the radius of a cluster, $A \subset J$, is defined to be¹⁴

$$r(\mathbf{x}|A) := \max_{a \in A} (\|x_a - c(\mathbf{x}|A)\| + r_a). \quad (2.12)$$

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

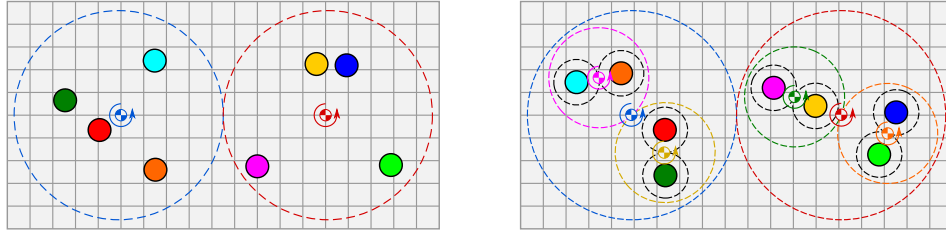


Figure 2.6: An illustration of (left) narrow and (right) standard disk configurations, where arrows and dashed circles indicate clusters that can be rigidly rotated around their centroids while preserving their clustering structures.

Since configuration hierarchies of $\text{HC}_{2\text{-means}}$ 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 the following:

Proposition 2.1 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 $c(\mathbf{x}|I)$, as illustrated in Figure 2.6, preserves the supported hierarchy τ .

Proposition 2.2 For any finite label set $S \subset \mathbb{N}$ and any binary tree $\tau \in \mathcal{BT}_S$, there exists a strong deformation retraction¹⁵

$$R_\tau : \mathfrak{S}(\tau) \times [0, 1] \rightarrow \mathfrak{S}(\tau) \quad (2.13)$$

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

These two observations now yield the key insight reported in [25].

¹⁴Recall from p.13 that r_i denotes the radius of i th sphere for any $i \in S$.

¹⁵In [25], 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 them.

Theorem 2.2 *The set of configurations $\mathbf{x} \in \text{Conf}(\mathbb{R}^n, S, \mathbf{r})$ supporting a binary tree ¹⁶ has the homotopy type of $(\mathbb{S}^{n-1})^{|S|-1}$.*

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. Hence, navigation in a hierarchical stratum is carried out by aligning separating hyperplane normals, ^{17 18 19} illustrated in Figure 2.7. By using this geometric intuition, we construct in [15] a family of hierarchy preserving control policies for point particle configurations, and in the following, we extend that construction to thickened disk configurations.

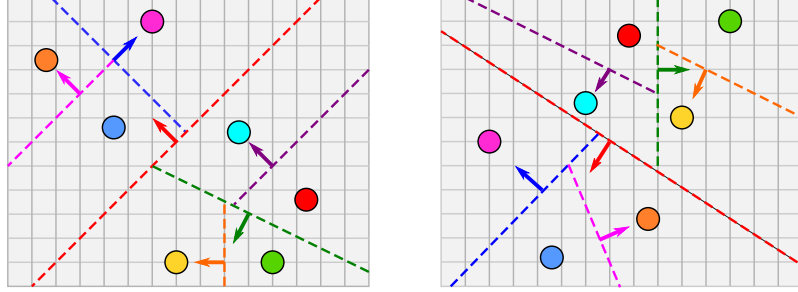


Figure 2.7: The topological shape of a hierarchical stratum intuitively suggests that global navigation in a hierarchical stratum is accomplished by aligning separating hyperplanes of configurations.

In order to appreciate the mathematical importance of Theorem 2.2, it is necessary to bear in mind the fact that configuration spaces have complicated topological shapes and are very difficult, if not impossible, to characterize in terms of simple standard topological models [76, 77]. One has such results only for low-dimensional settings. For example, $\text{Conf}(\mathbb{R}^2, [2], \mathbf{0})$ has the homotopy type of a circle [1], and one can observe from Figure 2.3 that $\text{Conf}(\mathbb{R}^2, [3], \mathbf{0})$ is homotopy equivalent to a circle times a figure-eight (i.e., ∞ -shaped curve). On the other hand, for an arbitrary number of disks and ambient space dimension, a hierarchical stratum of $\text{HC}_{2\text{-means}}$ is topologically well understood and is homotopy equivalent to a generalized torus, see Theorem 2.2 and Table 2.4. Hence, we believe

¹⁶Note that a binary hierarchy over the leaf set S has $|S| - 1$ interior nodes, i.e., nonsingleton clusters [184].






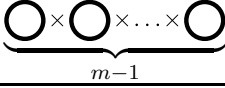
¹⁷In [15], we construct a linear bijective mapping relating the configuration space and the centroidal separations of complementary clusters of any given hierarchy such that a multirobot configuration is uniquely determined by its centroid and the centroidal separations of complementary clusters of the associated hierarchy. Hence, since the Euclidean n -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 2.2.

¹⁸For the stability analysis of hierarchy-invariant local policies of point particle configurations, we use in [15] 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 A.9, we also show that the separating hyperplane normals of complementary clusters are asymptotically aligned with the desired ones.

¹⁹Theorem 2.2 also explains the technical necessity of a rotational vector field in swarm coordination, especially for collision avoidance, which is usually employed in an intuitive, ad hoc manner [93, 191].

that the hierarchical strata might have a significant value for understanding the topological complexity [78] of coordinated motion design in $\text{Conf}(\mathbb{R}^n, S, \mathbf{r})$, because, as opposed to standard convex cell decompositions²⁰ [21], the hierarchical strata comprise a cover of the configuration space by subsets with nontrivial topology.

Table 2.4: Topological Shape of a Hierarchical Stratum

	$m = 2$	$m = 3$	m
$\text{Conf}(\mathbb{R}^2, [m], \mathbf{r})$		—	—
$\text{Conf}(\mathbb{R}^2, [m], \mathbf{0})$			—
$\mathfrak{S}(\tau)$			

Finally, it is useful to highlight some of the desired properties of $\text{HC}_{2\text{-means}}$ (Properties 2.1-2.3):

Theorem 2.3 *Iterative 2-means clustering $\text{HC}_{2\text{-means}}$ is a multi-function, and each of its stratum, $\mathfrak{S}(\tau)$ associated with $\tau \in \mathcal{BT}_S$, is connected and has an open interior.*

Proof It is well known that k -means clustering is a multi-function that generally yields different k -partitions of any given data and, therefore, is $\text{HC}_{2\text{-means}}$ (see Property 2.1) [115, 223]. Further, it follows from Definition 2.2 and Proposition 2.2 that the set of standard configurations in $\mathfrak{S}(\tau)$ is open (see Property 2.2), and Theorem 2.2 guarantees the connectedness of $\mathfrak{S}(\tau)$ (see Property 2.3). ■

2.3.2 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, which supports some desired nondegenerate tree $\tau \in \mathcal{BT}_S$. Our dynamical planner takes the form of a centralized hybrid controller $f_{\tau, \mathbf{y}} : \mathfrak{S}(\tau) \rightarrow (\mathbb{R}^n)^{|S|}$ that defines 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 2.5. Throughout this section, the tree τ and the goal configuration \mathbf{y} are fixed; therefore, we suppress all mention of these terms wherever convenient in order to simplify 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 $\eta_{i,I}(\mathbf{x}) = \eta_{i,I,\tau}(\mathbf{x})$ (2.7), $s_I(\mathbf{x}) = s_{I,\tau}(\mathbf{x})$ (2.9), $\mathbf{m}_I(\mathbf{x}) = \mathbf{m}_{I,\tau}(\mathbf{x})$ (2.10), and so on.

²⁰Note that a convex set has the homotopy type of a point.

Table 2.5: Hierarchy-Preserving Navigation Vector Field

For any initial $\mathbf{x} \in \mathfrak{S}(\tau)$ and desired $\mathbf{y} \in \mathfrak{S}(\tau)$, supporting $\tau \in \mathcal{BT}_S$, the hierarchy-preserving vector field, $f_{\tau, \mathbf{y}} : \mathfrak{S}(\tau) \rightarrow (\mathbb{R}^n)^S$,

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

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

Base Cases	1.	function $\hat{\mathbf{u}} = \hat{f}_{\tau, \mathbf{y}}(\mathbf{x}, \mathbf{u}, I)$	
	2.	if $\mathbf{x} \in \mathcal{D}_A(I)$ (2.15),	
	3.	$\hat{\mathbf{u}} \leftarrow f_A(\mathbf{x}, \mathbf{u}, I)$ (2.14),	% Attracting Field
	4.	else if $\mathbf{x} \notin \mathcal{D}_H(I)$ (2.18),	
	5.	$\hat{\mathbf{u}} \leftarrow f_S(\mathbf{x}, \mathbf{u}, I)$ (2.24),	% Split Separation Field
	6.	else	
Recursion	7.	$\{I_L, I_R\} \leftarrow \text{Ch}(I, \tau)$,	
	8.	$\hat{\mathbf{u}}_L \leftarrow \hat{f}_{\tau, \mathbf{y}}(\mathbf{x}, \mathbf{u}, I_L)$,	% Recursion for Left Child
	9.	$\hat{\mathbf{u}}_R \leftarrow \hat{f}_{\tau, \mathbf{y}}(\mathbf{x}, \hat{\mathbf{u}}_L, I_R)$,	% Recursion for Right Child
	10.	$\hat{\mathbf{u}} \leftarrow f_H(\mathbf{x}, \hat{\mathbf{u}}_R, I)$ (2.19),	% Split Preserving Field
	11.	end	
	12.	return $\hat{\mathbf{u}}$	

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 (2.15) and illustrated in Figure 2.8, and that can be directly moved toward the desired configurations, using a family of attracting fields f_A (2.14), with no collisions along the way. Once the partial configurations associated with sibling clusters I and I^τ 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. Once the separating hyperplanes of all clusters of τ are “sufficiently aligned”, $f_{\tau, \mathbf{y}}$ drives asymptotically each disk directly toward its desired location. We now present and motivate its constituent formulas as follows.

The hierarchy-invariant vector field $f_{\tau, \mathbf{y}}$ in cases 2 and 3 in Table 2.5 recursively detects partial configurations, such as $\mathbf{x}|I$ associated with cluster $I \in \mathcal{C}(\tau)$, that can be safely driven toward the goal formation in $\mathfrak{S}(\tau)$ by using a family of attracting controllers, $f_A : \mathfrak{S}(\tau) \times (\mathbb{R}^n)^S \times \mathcal{C}(\tau) \rightarrow (\mathbb{R}^n)^S$, that are defined in terms of the negated gradient field of $V(\mathbf{x}) := \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|_2^2$ as: for any $j \in S$,

$$f_A(\mathbf{x}, \mathbf{u}, I)_j := \begin{cases} -(\mathbf{x}_j - \mathbf{y}_j), & \text{if } j \in I \\ \mathbf{u}_j, & \text{else} \end{cases} \quad (2.14)$$

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

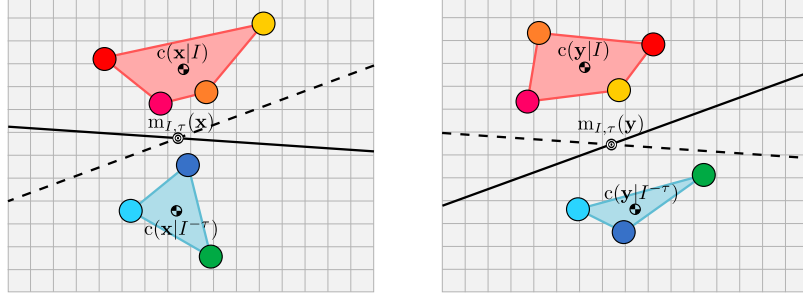


Figure 2.8: An illustration of “sufficiently aligned” separating hyperplanes of complementary clusters I and $I^{-\tau}$ of τ . Both the current (left) and desired (right) partial configurations are linearly separable by each others separating hyperplane, and such an alignment condition needs to be satisfied 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.

To avoid intracluster 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

$$\mathcal{D}_A(I) := \left\{ \mathbf{x} \in \mathfrak{S}(\tau) \mid \mathcal{L}_{\vec{y}} \frac{1}{2} \|\mathbf{x}_i - \mathbf{x}_j\|^2 \geq (r_i + r_j)^2, \quad \forall i \neq j \in I, \right. \\ \left. \mathcal{L}_{\vec{y}}(\mathbf{x}_k - \mathbf{m}_K(\mathbf{x}))^T s_K(\mathbf{x}) \geq 0, \quad \forall k \in K, K \in \text{Des}(I, \tau) \right\}, \quad (2.15)$$

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

$$\mathcal{L}_{\vec{y}} \frac{1}{2} \|\mathbf{x}_i - \mathbf{x}_j\|^2 = (\mathbf{x}_i - \mathbf{x}_j)^T (\mathbf{y}_i - \mathbf{y}_j), \quad (2.16)$$

$$\mathcal{L}_{\vec{y}}(\mathbf{x}_k - \mathbf{m}_K(\mathbf{x}))^T s_K(\mathbf{x}) = (\mathbf{y}_k - \mathbf{m}_K(\mathbf{y}))^T s_K(\mathbf{x}) + (\mathbf{x}_k - \mathbf{m}_K(\mathbf{x}))^T s_K(\mathbf{y}). \quad (2.17)$$

Note that (2.16) quantifies the safety of a resulting trajectory of f_A , and to avoid collision between any pair of disks, i and j , (2.16) should be no less than the square of sum of their radii, $(r_i + r_j)^2$, as required in (2.15); and (2.17) 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 (2.15) 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 intercluster collisions, the failure of the condition in case 4 in Table 2.5 ensures that sibling clusters $\text{Ch}(I, \tau)$ will be separated by a certain distance, specified as

$$\mathcal{D}_H(I) := \left\{ \mathbf{x} \in \mathfrak{S}(\tau) \mid \eta_{k,K}(\mathbf{x}) \geq r_k + \alpha, \quad \forall k \in K, K \in \text{Ch}(I, \tau) \right\}, \quad (2.18)$$

where $\eta_{k,K}(\mathbf{x})$ (2.7) returns the perpendicular distance of k th 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, $\text{Ch}(I, \tau)$, 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. $\text{Ch}(I, \tau) = \emptyset$.

While the disks move in $\mathcal{D}_H(I)$ based on a desired control (velocity) input $\mathbf{u} \in (\mathbb{R}^n)^S$, step 10 in Table 2.5 guarantees the maintenance of the safety margin between children clusters $\text{Ch}(I, \tau)$ by employing an additive repulsive field $f_H : \mathfrak{S}(\tau) \times (\mathbb{R}^n)^S \times \mathcal{C}(\tau) \rightarrow (\mathbb{R}^n)^S$ that rigidly pushes the children clusters apart as follows:

$$f_H(\mathbf{x}, \mathbf{u}, I)_j := u_j + 2\alpha_I(\mathbf{x}, \mathbf{u}) \frac{|K^{-\tau}|}{|I|} \frac{s_K(\mathbf{x})}{\|s_K(\mathbf{x})\|}, \quad (2.19)$$

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

$$\alpha_I(\mathbf{x}, \mathbf{u}) := \max_{\substack{k \in K \\ K \in \text{Ch}(I, \tau)}} \phi_{k,K}(\mathbf{x}) \cdot \psi_{k,K}(\mathbf{x}, \mathbf{u}). \quad (2.20)$$

Here, for each individual k in cluster $K \in \text{Ch}(I, \tau)$, $\phi_{k,K}(\mathbf{x})$ 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 $\beta > \alpha$,

$$\phi_{k,K}(\mathbf{x}) := \max \left(\frac{e^{-(\eta_{k,K}(\mathbf{x}) - r_k - \alpha)} - e^{-(\beta - \alpha)}}{1 - e^{-(\beta - \alpha)}}, 0 \right), \quad (2.21)$$

$$\psi_{k,K}(\mathbf{x}, \mathbf{u}) := \max \left(-(\eta_{k,K}(\mathbf{x}) - r_k - \alpha) - \mathcal{L}_{\vec{\mathbf{u}}} \eta_{k,K}(\mathbf{x}), 0 \right), \quad (2.22)$$

where

$$\mathcal{L}_{\vec{\mathbf{u}}} \eta_{k,K}(\mathbf{x}) = \frac{(\mathbf{u}_k - \mathbf{m}_K(\mathbf{u}))^T s_K(\mathbf{x}) + (\mathbf{x}_k - \mathbf{m}_K(\mathbf{x}))^T s_K(\mathbf{u})}{\|s_K(\mathbf{x})\|} - \eta_{k,K}(\mathbf{x}) \frac{s_K(\mathbf{x})^T s_K(\mathbf{u})}{\|s_K(\mathbf{x})\|^2}. \quad (2.23)$$

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., $f_H(\mathbf{x}, \mathbf{u}, I) = \mathbf{u}$, because $\text{Ch}(I, \tau) = \emptyset$. Also observe that $f_H(\mathbf{x}, \mathbf{u}, I) = \mathbf{u}$ for any $I \in \mathcal{C}(\tau)$ if the complementary clusters $\text{Ch}(I, \tau)$ are well separated, i.e., $\eta_{k,K}(\mathbf{x}) \geq r_k + \beta$ for all $k \in K$ and $K \in \text{Ch}(I, \tau)$. The latter is important to avoid the “finite escape time” phenomenon²¹ (see Proposition A.12).

Finally, case 5 in Table 2.5 guarantees that if a partial configuration is neither in the domain of the attracting field nor are its children clusters $\text{Ch}(I, \tau)$ properly separated, i.e., $\mathbf{x} \notin \mathcal{D}_A(I) \cup \mathcal{D}_H(I)$, then the complementary clusters are driven apart by using another repulsive field, $f_S : \mathfrak{S}(\tau) \times (\mathbb{R}^n)^S \times \mathcal{C}(\tau) \rightarrow (\mathbb{R}^n)^S$, until asymptotically establishing a certain safety margin $\beta > \alpha$:

²¹ A trajectory of a dynamical system is said to have a finite escape time if it escapes to infinity at a finite time [127].

$$f_S(\mathbf{x}, \mathbf{u}, I)_j := -c(\mathbf{x} - \mathbf{y}|I) + 2\beta_I(\mathbf{x}) \frac{|K^{-\tau}|}{|I|} \frac{s_K(\mathbf{x})}{\|s_K(\mathbf{x})\|}, \quad (2.24)$$

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

$$\beta_I(\mathbf{x}) := \max_{\substack{K \in \text{Ch}(I, \tau) \\ k \in K}} \max(-(\eta_{k,K}(\mathbf{x}) - r_k - \beta), 0). \quad (2.25)$$

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 follows.²²

Theorem 2.4 *The recursion of Table 2.5 results in a well-defined function $f_{\tau, \mathbf{y}} : \mathfrak{S}(\tau) \rightarrow (\mathbb{R}^n)^S$ that can be computed in $O(|S|^2)$ time for any $\mathbf{x} \in \mathfrak{S}(\tau)$. For all $\tau \in \mathcal{BT}_S$, 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 of $\mathfrak{S}(\tau)$ with the exception of an empty interior set.*²³

Proof These results are proven in Appendix A according to the following plan. Proposition A.1 establishes that the recursion in Table 2.5 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 [45], as follows. Table A.2 defines a new recursion that is shown in Proposition A.2 to result in a family of continuous and piecewise smooth vector fields. Proposition A.3 asserts that the family of domains associated with these fields (A.3) defines a (finite) open cover of $\mathfrak{S}(\tau)$ relative to which a selection function (see Table A.3) induces a partition of that stratum. Proposition A.4 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 2.5. Finally, Proposition A.12, Proposition A.11 and Proposition A.13 demonstrate, respectively, the flow, positive invariance and stability properties of $f_{\tau, \mathbf{y}}$, which are inherited from the flow, invariance and stability properties (see Proposition A.8, Proposition A.7 and Proposition A.9, respectively) of substratum policies executed over a strictly decreasing finite prepares graph (see Proposition A.5) via their nondegenerately, real-time executed (see Proposition A.10) sequential composition. ■

2.3.3 Navigation in the Space of Binary Trees

In principle, navigation in the adjacency graph of trees (see Problem 2.2) is a trivial matter, because the number of trees over a finite set of leaves is finite. However, in practice, the cardinality of trees grows super exponentially [35]

$$|\mathcal{BT}_S| = (2|S| - 3)!! \stackrel{\text{def.}}{=} (2|S| - 3)(2|S| - 5) \dots 3 \cdot 1, \quad \forall |S| \geq 2. \quad (2.26)$$

²²This construction indeed solves Problem 2.1, since a flow is a retraction of its basin into the attractor [33].

²³It follows from Theorem 2.2 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 .

Hence, standard graph search algorithms, like the A* or Dijkstra's algorithm [59], become rapidly impracticable. In particular, computing the shortest path (geodesic) in the NNI graph, a regular subgraph of the adjacency graph (see Theorem 2.6), is NP-complete [66].

Alternatively, we have recently developed in [17], presented in detail in Chapter 3, an efficient recursive procedure for navigating in the NNI graph $\mathcal{N}_S = (\mathcal{BT}_S, \mathcal{E}_{\mathcal{N}_S})$ toward any given binary tree $\tau \in \mathcal{BT}_S$, taking the form of an abstract discrete dynamical system as follows:

$$\sigma^{k+1} = \text{NNI}(\sigma^k, G^k) , \quad (2.27a)$$

$$G^k = \mathbf{u}_\tau(\sigma^k) , \quad (2.27b)$$

where $\text{NNI}(\sigma^k, G^k)$ denotes the NNI move ²⁴ on σ^k at cluster $G^k \in \mathcal{C}(\tau)$, illustrated in Figure 2.4, and \mathbf{u}_τ is our NNI control policy returning an NNI move as summarized in Table 2.6. Abusing notation, we shall denote the closed-loop dynamical system as

$$\sigma^{k+1} = g_\tau(\sigma^k) := (\text{NNI} \circ \mathbf{u}_\tau)(\sigma^k) . \quad (2.28)$$

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 Chapter 3 for more details).

Table 2.6: Nearest Neighbor Interchange (NNI) Control Law

To navigate from an arbitrary hierarchy $\sigma \in \mathcal{BT}_S$ towards any selected desired hierarchy $\tau \in \mathcal{BT}_S$ in the NNI graph, the NNI control policy \mathbf{u}_τ 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, $G = \emptyset$.
 2. Otherwise,
 - (a) Select a common cluster $K \in \mathcal{C}(\sigma) \cap \mathcal{C}(\tau)$ with $\text{Ch}(K, \sigma) \neq \text{Ch}(K, \tau)$, and let $\{K_L, K_R\} = \text{Ch}(K, \tau)$.
 - (b) Find a nonsingleton cluster $I \in \mathcal{C}(\sigma)$ with children $\{I_L, I_R\} = \text{Ch}(I, \sigma)$ satisfying $I_L \subseteq K_L$ and $I_R \subseteq K_R$.
 - (c) Return a proper NNI navigation move on σ at grandchild $G \in \text{Ch}(I, \sigma)$ 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, $G = I_L$.
-
-

²⁴Here, note that the NNI move at the empty cluster corresponds to the identity map in \mathcal{BT}_S , i.e., $\sigma = \text{NNI}(\sigma, \emptyset)$ for all $\sigma \in \mathcal{BT}_S$. Therefore, the notion of identity map in \mathcal{BT}_S slightly extends the NNI graph by adding self-loops at every vertex, which is necessary for a discrete-time dynamical system in \mathcal{BT}_S 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, $\sigma \in \mathcal{BT}_S$) is tightly bounded by $\frac{1}{2}(|S| - 1)(|S| - 2)$ for $|S| \geq 2$. Given such a goal, we show in Chapter 3 that the cost of computing an appropriate NNI move from any other $\sigma \in \mathcal{BT}_S$ toward an adjacent tree at a lower value of a “discrete Lyapunov function” relative to that destination is $O(|S|)$. We summarize such important properties of our NNI navigation algorithm as follows.

Theorem 2.5 *The NNI control law u_τ (see Table 2.6) recursively defines a closed-loop discrete dynamical system (2.28) in the NNI graph, taking the form of a discrete transition rule, g_τ , with global attractor at τ and longest trajectory of length $O(|S|^2)$ endowed with a discrete Lyapunov function relative to which computing a descent direction from any $\sigma \in \mathcal{BT}_S$ requires a computation of time $O(|S|)$.*

Proof See Chapter 3. ■

2.3.4 Portal Transformations

We now turn our attention to construction of the crucial portal map that affects the geometric realization of the NNI graph as required for Problem 2.3, and herein, we extend our recent construction of the realization function, **Port**, in [16] for point particle configurations to thickened disk configurations.

Throughout this section, the trees $\sigma, \tau \in \mathcal{BT}_S$ are NNI-adjacent (as defined in Section 2.1.4) and fixed; therefore, we 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 3.1 from Chapter 3 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** (2.33), we restrict our attention to the portal configurations with a certain symmetry property, defined as follows.

Definition 2.3 *We call $\mathbf{x} \in (\mathbb{R}^n)^S$ 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 Figure 2.9(left). The set of all symmetric configurations with respect to (σ, τ) is denoted $\text{Sym}(\sigma, \tau)$.*

An important observation about the symmetric configurations is as follows:

Lemma 2.1 ([16]) *Let $\mathbf{x} \in \mathfrak{S}(\sigma)$ be a symmetric configuration in $\text{Sym}(\sigma, \tau)$. 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 ball²⁵ centered at $c(\mathbf{x}|Q)$ with radius*

$$r_Q(\mathbf{x}) := \min_{\substack{\gamma \in (\sigma, \tau) \\ D \in \{Q, \text{Pr}(Q, \gamma)\} \setminus \{P\}}} \left(c(\mathbf{x}|Q) - m_{D, \gamma}(\mathbf{x}) \right)^T \frac{s_{D, \gamma}(\mathbf{x})}{\|s_{D, \gamma}(\mathbf{x})\|}, \quad (2.29)$$

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

²⁵In a metric space (X, d) , the open ball $B(\mathbf{x}, r)$ centered at $\mathbf{x} \in X$ with radius $r \in \mathbb{R}_{\geq 0}$ is the set of points in X whose distance to \mathbf{x} is less than r , i.e., $B(\mathbf{x}, r) = \{y \in X \mid d(\mathbf{x}, y) < r\}$.

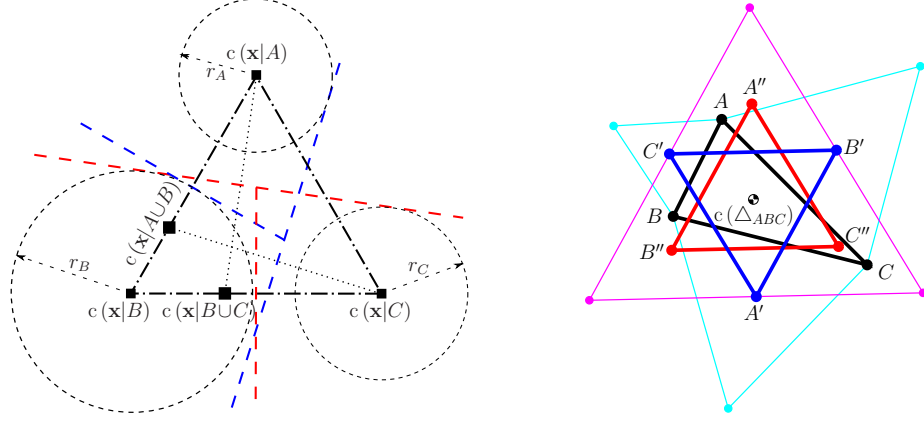


Figure 2.9: (left) An illustration of a symmetric configuration $\mathbf{x} \in \text{Sym}(\sigma, \tau)$, where the consensus ball $B_Q(\mathbf{x})$ of partial configuration of cluster $Q \in \{A, B, C\}$ has a positive radius. (right) Outer Napoleon Triangles $\triangle_{A'B'C'}$ and $\triangle_{A''B''C''}$ 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., $c(\triangle_{ABC}) = c(\triangle_{A'B'C'}) = c(\triangle_{A''B''C''})$.

Note that for any symmetric configuration $\mathbf{x} \in \text{Sym}(\sigma, \tau)$, the consensus ball of each partial configuration of cluster $Q \in \{A, B, C\}$ always has a nonempty interior, i.e., $r_Q(\mathbf{x}) > 0$ [16] — see Figure 2.9(left).

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.

Napoleon Triangles

We recall a theorem of geometry that describes 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; therefore, the centroids of the constructed equilateral triangles form another equilateral triangle in the same plane, known as the “*Napoleon triangle*” [62] (see Figure 2.9(right)). 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 $\text{NT} : \mathbb{R}^{3n} \rightarrow \mathbb{R}^{3n}$ denote the double outer Napoleon transformation (see Appendix B for an explicit form of NT). It is also worth mentioning 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 (Appendix B).

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

$$\triangle_{A,B,C}(\mathbf{x}) := [c(\mathbf{x}|A), c(\mathbf{x}|B), c(\mathbf{x}|C)]^T. \quad (2.30)$$

The double outer Napoleon transformation of $\triangle_{A,B,C}(\mathbf{x})$ 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

$\text{Noff}_{A,B,C} : \text{Conf}(\mathbb{R}^n, S, \mathbf{r}) \rightarrow \mathbb{R}^n$, is given by the formula

$$\text{Noff}_{A,B,C}(\mathbf{x}) := \mathbf{c}(\mathbf{x}|P) - \Gamma \cdot \mathbf{NT} \circ \Delta_{A,B,C}(\mathbf{x}) , \quad (2.31)$$

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

$$[c_A, c_B, c_C]^T := \mathbf{NT} \circ \Delta_{A,B,C}(\mathbf{x}) + \mathbf{1}_3 \otimes \text{Noff}_{A,B,C}(\mathbf{x}) . \quad (2.32)$$

Here, \mathbf{I}_n is the $n \times n$ 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.

Portal Maps

We now define a portal map, $\text{Port} : \mathfrak{S}(\sigma) \rightarrow \text{Portal}(\sigma, \tau)$, to be

$$\text{Port}(\mathbf{x}) := \begin{cases} \mathbf{x}, & \text{if } \mathbf{x} \in \text{Portal}(\sigma, \tau) , \\ (\text{Mrg} \circ \text{Sc1} \circ \text{Ctr})(\mathbf{x}), & \text{otherwise} , \end{cases} \quad (2.33)$$

where $\text{Ctr} : \mathfrak{S}(\sigma) \rightarrow \text{Sym}(\sigma, \tau)$ 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 (2.32), respectively, yielding a symmetric configuration

$$\text{Ctr}(\mathbf{x}) := \begin{cases} \mathbf{x}_i, & \text{if } i \notin P , \\ \mathbf{x}_i - \mathbf{c}(\mathbf{x}|Q) + c_Q, & \text{if } i \in Q, Q \in \{A, B, C\} . \end{cases} \quad (2.34)$$

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

After obtaining a symmetric configuration in $\text{Sym}(\sigma, \tau)$, $\text{Sc1} : \text{Sym}(\sigma, \tau) \rightarrow \text{Sym}(\sigma, \tau)$ 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 ,

$$\text{Sc1}(\mathbf{x}) := \begin{cases} \mathbf{x}_i, & \text{if } i \notin P \\ \mathbf{x}_i + \zeta \cdot (\mathbf{c}(\mathbf{x}|Q) - \mathbf{c}(\mathbf{x}|P)), & \text{if } i \in Q, Q \in \{A, B, C\} , \end{cases} \quad (2.35)$$

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

$$\zeta := \max_{Q \in \{A, B, C\}} \max \left(\frac{r(\mathbf{x}|Q) + \alpha}{r_Q(\mathbf{x})}, 1 \right) - 1 . \quad (2.36)$$

Here, $\alpha > 0$ is a safety margin as used in (2.20), and $r(\mathbf{x}|Q)$ (2.12) denotes the centroidal radius of partial configuration $\mathbf{x}|Q$ and $r_Q(\mathbf{x})$ (2.29) is the radius of its consensus ball. Note that Sc1 preserves the configuration symmetry, i.e., centroids $\mathbf{c}(\mathbf{x}|A)$, $\mathbf{c}(\mathbf{x}|B)$, and $\mathbf{c}(\mathbf{x}|C)$ still form an equilateral triangle after the mapping and leaves the barycenter of $\mathbf{x}|P$ unchanged.

Finally, $\text{Mrg} : \text{Sym}(\sigma, \tau) \rightarrow \text{Sym}(\sigma, \tau)$ 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 ,

$$\text{Mrg}(\mathbf{x}) := \text{Mrg}_P(\mathbf{x}), \quad (2.37)$$

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

$$\text{Mrg}_I(\mathbf{x}) := \begin{cases} \mathbf{x}, & \text{if } I = S, \\ (\text{Mrg}_{\text{Pr}(I, \tau)} \circ \text{Sep}_I)(\mathbf{x}), & \text{otherwise.} \end{cases} \quad (2.38)$$

Here, 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 $\hat{\mathbf{x}} = \text{Sep}_I(\mathbf{x})$ for some $\mathbf{x} \in (\mathbb{R}^n)^S$ with $s_{I, \sigma}(\mathbf{x}) \neq 0$, then $\eta_{k, K, \sigma}(\hat{\mathbf{x}}) \geq r_k + \alpha$ for any $k \in K$, $K \in \{I, I^{-\sigma}\}$: for any $j \in S$

$$\text{Sep}_I(\mathbf{x})_j := \begin{cases} \mathbf{x}_j, & \text{if } j \notin \text{Pr}(I, \sigma), \\ \mathbf{x}_j + 2\lambda \frac{|K^{-\sigma}|}{|\text{Pr}(K, \sigma)|} \frac{s_{K, \sigma}(\mathbf{x})}{\|s_{K, \sigma}(\mathbf{x})\|}, & \text{if } j \in K, K \in \{I, I^{-\sigma}\}, \end{cases} \quad (2.39)$$

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

$$\lambda := \max_{K \in \{I, I^{-\sigma}\}} \max_{k \in K} (-(\eta_{k, K, \sigma}(\mathbf{x}) - r_k - \alpha), 0). \quad (2.40)$$

Note that since $c(\mathbf{x}|P) = c(\hat{\mathbf{x}}|P)$ for any $\mathbf{x} \in \mathfrak{S}(\sigma)$ and $\hat{\mathbf{x}} = (\text{Sc1} \circ \text{Ctr})(\mathbf{x})$, we always have $s_{I, \sigma}(\hat{\mathbf{x}}) \neq 0$ for any $I \in \{P\} \cup \text{Anc}(P, \sigma)$, which is required for Sep_I to be well defined. Further, using (2.39), one can verify that $c(\mathbf{x}|\text{Pr}(I, \sigma)) = c(\hat{\mathbf{x}}|\text{Pr}(I, \sigma)) = c(\tilde{\mathbf{x}}|\text{Pr}(I, \sigma))$ for $\tilde{\mathbf{x}} = \text{Sep}_I(\hat{\mathbf{x}})$; therefore, $s_{A, \sigma}(\tilde{\mathbf{x}}) \neq 0$ for any $A \in \text{Anc}(I, \sigma)$, which guarantees that recursive calls of Sep_I in the computation of Port are always well defined.

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

Theorem 2.6 *The NNI graph $\mathcal{N}_S = (\mathcal{BT}_S, \mathcal{E}_{\mathcal{N}_S})$ is a subgraph of the $\text{HC}_{2\text{-means}}$ adjacency graph $\mathcal{A}_S = (\mathcal{BT}_S, \mathcal{E}_{\mathcal{A}})$, i.e., for any pair (σ, τ) of NNI-adjacent trees in \mathcal{BT}_S , $\text{Portal}(\sigma, \tau) \neq \emptyset$. Further, given an edge $(\sigma, \tau) \in \mathcal{E}_{\mathcal{N}_S} \subset \mathcal{E}_{\mathcal{A}}$, a geometric realization via the map $\text{Port}_{(\sigma, \tau)} : \mathfrak{S}(\sigma) \rightarrow \text{Portal}(\sigma, \tau)$ (2.33) can be computed in quadratic $\mathcal{O}(|S|^2)$ time with the number of leaves $|S|$.*

Proof See Appendix A.2.1. ■

2.4 Numerical Simulations

For the sake of clarity, we first illustrate the behavior of the hybrid system defined in Section 2.3 for the case of four disks moving in a 2-D ambient space.²⁶

²⁶For all simulations, we consider unit disks moving in an ambient plane, i.e., $r_i = 1$ for all $i \in S$, 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 (see Table 2.3) using the `ode45` function of MATLAB.

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 Figure 2.10).

The resultant trajectory of each disk following the hybrid navigation planner in Section 2.3, the relative distance between each pair of disks, and the sequence of trees associated with visited hierarchical strata are shown in Figure 2.10. 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 Figure 2.10).

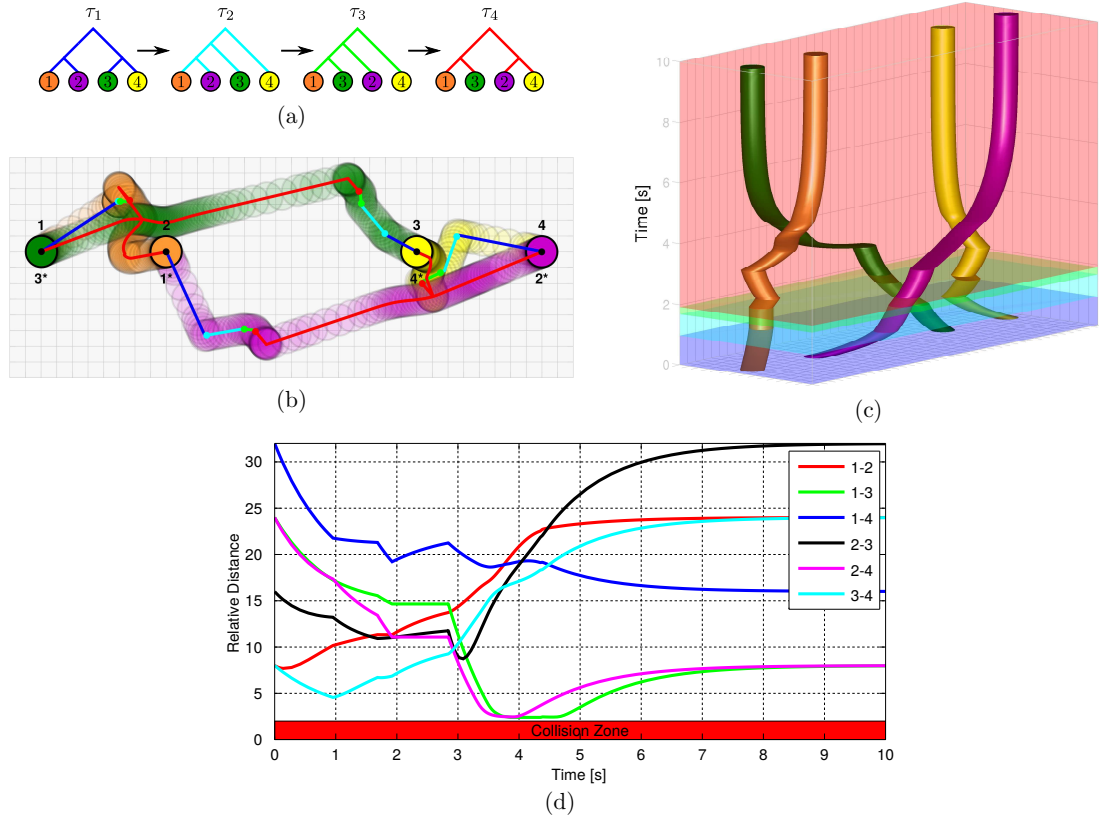


Figure 2.10: Illustrative navigation trajectory of the hybrid dynamics generated by the HNC algorithm for four disks in a planar ambient space. Disks are placed on the horizontal axis for both the initial and desired configurations in different orders, from left to right $(1, 2, 3, 4)$ and $(3^*, 1^*, 4^*, 2^*)$ at the start and goal, respectively. (a) Sequence of trees associated with deployed local controllers during the execution of the hybrid navigation controller. (b) Centroidal trajectory of each disk colored according to 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.

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_{τ_4, \mathbf{x}^*} (see Table 2.5) at $\mathbf{x}_r \in \mathfrak{S}(\tau_3) \cap \mathfrak{S}(\tau_4)$ (shown by red dots in Figure 2.10), and f_{τ_4, \mathbf{x}^*} asymptotically steers the 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 four 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 Figure 2.11(top), which is also used in [209]

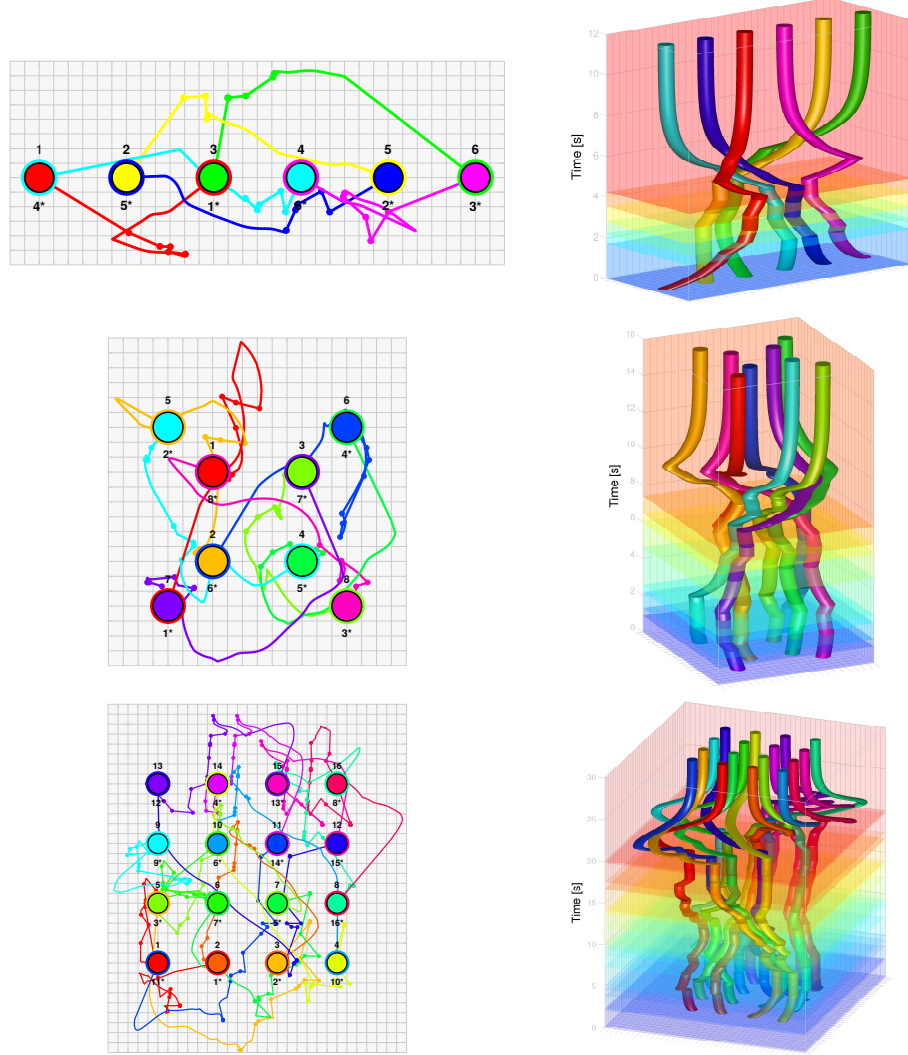


Figure 2.11: Example trajectories of the hybrid vector field planner for (top) 6, (middle) 8, and (bottom) 16 disks in a planar ambient space. (left) Trajectory and (right) state-time curve of each disk. Each colored time interval demonstrates the execution duration of an activated local controller. Dots correspond to the portal configurations where transitions between local controllers occur.

as an example of complicated multi-agent arrangements. Here, the order of disk labels from left to right are, respectively, $(1, 2, 3, 4, 5, 6)$ and $(4^*, 5^*, 1^*, 6^*, 2^*, 3^*)$ for the initial and goal configurations. While steering the disks toward the goal, the hybrid navigation planner automatically deploys only six local controllers out of the family of 945 local controllers. The time evolution of the disk is illustrated in Figure 2.11(top).

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 Figure 2.11. The eight disks are initially located at the corner of two squares whose centroids coincide, and the perimeter of one is twice that of the perimeter of the other. At the destination, disks switch their locations as illustrated in Figure 2.11(middle). 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 Figure 2.11(bottom). Although there are a large number of local controllers for the case of groups of 8 and 16 disks ($|\mathcal{BT}_{[8]}| > 10^5$ and $|\mathcal{BT}_{[16]}| > 6 \times 10^{15}$), our hybrid navigation planner only deploys 9 and 19 local controllers, respectively.

The number of potentially available local controllers for a group of m disks (2.26) grows super exponentially with m . On the other hand, if agents have perfect sensing and actuation modelled as in this chapter, the hybrid navigation planner automatically deploys at most $\frac{1}{2}(m-1)(m-2)$ local controllers (see Chapter 3), illustrating the computational efficiency of our construction.

Finally, although the HNC algorithm in Section 2.3 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 [123, 222], in order to quantify navigation paths, we consider the *normalized navigation path length*, Γ , which is defined as the ratio of the total navigation distance traveled by all robots to the straight-line Euclidean distance between any initial and goal configurations [222]

$$\Gamma := \frac{\sum_{i=1}^n \int_0^\infty \|\dot{\mathbf{x}}_i(t)\| dt}{\sum_{i=1}^n \|\mathbf{x}_i(0) - \mathbf{x}_i^*\|}, \quad (2.41)$$

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 $\mathbf{x}^* = \lim_{t \rightarrow \infty} \mathbf{x}(t)$. Further, to ensure an unbiased selection of initial and goal configurations, we consider unit disk configurations (i.e., $r_i = 1$ for all $i = 1 \dots n$) 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 Figure 2.12, we present the effect of group size m , and configuration tightness k , on the normalized navigation path length Γ .²⁷ 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 it is for them to navigate to their destination. We also observe that the average normalized

²⁷Each data point in Figure 2.12 is obtained using 500 pairs of uniformly sampled random initial and goal configurations.

path length of the HNC algorithm has the same order of magnitude as those of other available navigation function based algorithms [222, 123] whose convergence and path properties significantly depend on parameter tuning.

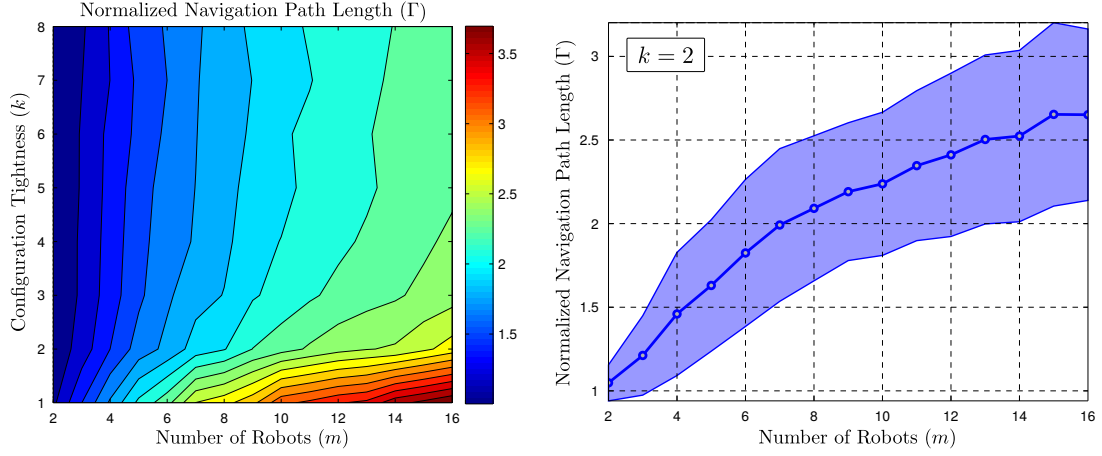


Figure 2.12: (left) Average normalized navigation path length versus group size, m , and configuration tightness, k . (right) Mean and standard deviation of the normalized navigation path length for configuration tightness $k = 2$.

2.5 Summary

In this chapter, we have introduced a novel application of clustering to address the problem of coordinated robot navigation. The notion of 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 toward any given destination via a sequence of hierarchy-preserving local controllers. For 2-means divisive hierarchical clustering [193], 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.

Specifically, matching the component problem statements of Section 2.2 to their subsequent resolution: we address Problem 2.1 in Theorem 2.4 (guaranteeing that the construction of Table 2.5 results in a hierarchy invariant vector field planner); we address Problem 2.2 in Theorem 2.5 (guaranteeing that the construction of Table 2.6 results in a reactive strategy that finds, given any non-goal tree, an edge in the graph of all hierarchies leading to a new tree that is closer to the desired goal hierarchy); and we address Problem 2.3 in Theorem 2.6 (providing a geometric realization in the configuration space of the combinatorial edge toward the physical goal). The efficacy of this overarching strategy is guaranteed by Theorem 2.1 (proving the correctness of these problems steps and their resolutions as presented in Table 2.3).

Chapter 3

Navigation in Tree Space

A fundamental classification problem common to both computational biology and engineering is the efficient and informative comparison of hierarchical structures. In bioinformatics settings, these typically take the form of phylogenetic trees representing evolutionary relationships within a set of taxa. In pattern recognition and data mining settings, hierarchical trees are often used to encode nested sequences of groupings of a set of observations. Dissimilarity between combinatorial trees has been measured in the past literature largely by recourse to one of two separate approaches: comparing edges and counting edit distances. Representing the former approach, a widely used tree metric is the Robinson-Foulds (RF) distance, d_{RF} , [185] whose count of the disparate edges between trees requires linear time, $O(n)$, in the number of leaves, n , to compute [67]. Empirically, d_{RF} offers only a very coarse measure of disparity, and among its many proposed refinements, the recent matching split distance d_{MS} , [36, 143] offers a more discriminative metric albeit with considerably higher computational cost, $O(n^{2.5} \log n)$. Alternatively, various edit distances have been proposed [184, 160, 5, 80] but the most natural variant, the nearest neighbor interchange (NNI) distance d_{NNI} , entails an NP-complete computation for both labelled and unlabelled trees [66].

The main contribution of this chapter is the introduction of a dissimilarity measure on the space \mathcal{BT}_S of labelled binary trees over a fixed finite index set S , which bridges the above approaches by what is, effectively, a solution to the NNI navigation problem in \mathcal{BT}_S :

NNI Navigation Problem *Given a target tree $\tau \in \mathcal{BT}_S$, provide a computationally efficient discrete transition rule \mathbf{u}_τ which, for any $\sigma \in \mathcal{BT}_S$, computes a nearest neighbor interchange (NNI) operation to be performed on σ while guaranteeing that successive application of \mathbf{u}_τ terminates in τ .*

This problem is motivated by our applications in coordinated robot navigation, presented in Chapter 2, where a group of robots is required to reconfigure reactively in real time their (structural) adjacencies while navigating towards a desired goal configuration. Thus, our particular formulation of the problem is inspired by the notion of reactive planning [45], but may likely hold value for researchers interested in tree consensus and averaging as well.

Of course, since computation of the NNI distance d_{NNI} is NP-hard, one cannot hope for repeated applications of \mathbf{u}_τ to produce NNI geodesics without incurring prohibitive complexity in each iteration. However, as we will show, constructing an efficient navigation scheme is possible if we allow the algorithm to produce less restricted paths: for $|S| = n$,

our navigation algorithms require $O(n)$ time for each iteration and produce paths of length $O(n^2)$ (as compared to the $O(n \log n)$ diameter of d_{NNI} [201] — see (3.26)).

Additional insight into the geometry of the space $(\mathcal{BT}_S, d_{NNI})$ is gained by recognizing a significant degree of freedom with which our navigation algorithm may select the required tree restructuring operation at each stage. As it turns out, for any given target τ , the repeated application of \mathbf{u}_τ to a tree σ until reaching τ will yield paths of equal lengths regardless of any choices made along the way. This length, by definition, is the *NNI navigation dissimilarity* $d_{nav}(\sigma, \tau)$ (and is obtained, in the manner described, in $O(n^3)$ time, though more efficient implementations will guarantee $O(n^2)$). At the same time, a closed form formula we derive for d_{nav} allows us to avoid computing a navigation path when only the value of d_{nav} is needed, and computes it in $O(n^2)$ time. This closed form expression comprise a weighted count of certain incompatible pairs of clusters formed by two trees and is derived as a kind of discrete “path integral” that counts the steps along trajectories of a discrete dynamical system (3.57) defined over the NNI-graph of tree space, which seeks to reduce the number of incompatible cluster pairs, level by level, with each chosen NNI operation. In this sense, d_{nav} seems distinguished in the large and still rapidly growing tree distance literature by offering a compromise between the two traditional approaches. It is both an efficient approximation to the (NP-hard) NNI distance and, along with this “edit length” interpretation, its sensitivity to the number of mutually incompatible clusters lends it the character of an edge dissimilarity measure as well.

Surprisingly, despite the asymmetric character of its construction, d_{nav} is a symmetric (and positive definite) dissimilarity on \mathcal{BT}_S , though it fails to be a true metric. Although d_{nav} does not satisfy the triangle inequality, it is related to the well accepted Robinson-Foulds distance by the following tight bounds:

$$d_{RF} \leq d_{nav} \leq \frac{1}{2}d_{RF}^2 + \frac{1}{2}d_{RF} . \quad (3.1)$$

We find it useful to introduce a “relaxation” of d_{nav} , the *crossing dissimilarity* d_{CM} . This dissimilarity simply counts all the pairwise cluster incompatibilities between two trees, hence it is symmetric, positive-definite, and computable in $O(n^2)$ time. In fact, the two dissimilarities are commensurable, leading to similar bounds in terms of d_{RF} :

$$d_{nav} \leq \frac{3}{2}d_{CM} , \quad \text{and} \quad d_{RF} \leq d_{CM} \leq d_{RF}^2 . \quad (3.2)$$

But unlike d_{nav} , d_{CM} is simple enough to work with that it can be linearly bounded from above by a true metric whose spatial resolution and computational complexity is comparable to those our new dissimilarities. Namely, exploiting a well known relation between trees and ultrametrics [47], we also introduce *the cluster-cardinality distance* d_{CC} — the pullback of a matrix norm along an embedding of hierarchies into the space of matrices and computable in $O(n^2)$ time — which is a true metric bounding d_{CM} from above (and hence also d_{nav} , up to a constant factor). Thus, cumulatively we obtain:

$$\frac{2}{3}d_{RF} \leq \frac{2}{3}d_{nav} \leq d_{CM} \leq d_{CC} . \quad (3.3)$$

To summarize, in a manner of speaking, the dissimilarities d_{nav} and d_{CM} are not that far from being metrics as one might have worried.

We have surveyed some of the new features of our tree proximity measures that might hold interest for pattern classification and phylogeny analysis relative to the diverse alternatives that have appeared in the literature. Closest among these many alternatives [142, 64, 42], d_{nav} has some resemblance to an early NNI graph navigation algorithm, d_{ra} [42] which used a divide-and-conquer approach with a balancing strategy to achieve an $O(n \log n)$ computation of tree dissimilarity. Notwithstanding its lower computational cost, in contrast to d_{nav} , the recursive definition of d_{ra} , as with many NNI distance approximations [142, 64, 42], does not admit a closed form expression.

It is often of interest to compare more than pairs of hierarchies at a time, and the notion of a “consensus” tree has accordingly claimed a good deal of attention in the literature [43]. For instance, the majority rule tree [154] of a set of trees is a median tree respecting the RF distance and provides statistics on the central tendency of trees [24]. When d_{nav} and d_{CM} are extended to degenerate trees, they fail to be positive definite, and thus their behavior over (typically degenerate) consensus trees departs still further from the properties of a true metric. However, it turns out that both notions of a consensus tree (strict [117], and loose/semi-strict [39]) behave as median trees with respect to both our dissimilarities. In fact, the loose consensus tree is the maximal (finest) median tree with respect to inclusion for both d_{nav} and d_{CM} .

This chapter is based on the submitted journal paper [17] that introduce and study these proposed tree measures for efficient discriminative comparison of phylogenetic trees, and to demonstrate their discriminative power, the chapter concludes with a brief numerical exploration of the distribution over tree space of these dissimilarities in comparison with the Robinson-Foulds metric and the more recently introduced matching-split distance.

3.1 Preliminaries

We now introduce our basic notation used throughout the chapter and recall several standard notions of hierarchies, such as cluster compatibility, hierarchical relations of clusters and tree operations, from a set theoretical perspective.

3.1.1 Hierarchies

By a *hierarchy* τ over a fixed non-empty finite index set S , say $S = [n] := \{1, 2, \dots, n\}$, we shall mean a rooted tree with labeled leaves (see Figure 3.1). Formally, τ is a finite connected acyclic graph with leaves (vertices of degree one) bijectively labelled by S , and edges oriented in such a way that (i) all interior vertices have out-degree at least two, and (ii) there is a vertex, referred to as the *root of* τ , such that every edge is oriented away from the root. Under these assumptions all the vertices V_τ of τ are reachable from the root through a directed path in τ [35].

The *cluster* $\mathcal{C}(v)$ of a vertex $v \in V_\tau$ is defined to be the set of leaves reachable from v by a directed path in τ . Singleton clusters and the root cluster S are common to all trees, and we refer to them as the trivial clusters. We denote by $\mathcal{C}(\tau)$ (respectively $\mathcal{C}_{int}(\tau)$) the set of all clusters (resp. nontrivial clusters) of τ :

$$\mathcal{C}(\tau) := \{\mathcal{C}(v) \mid v \in V_\tau\} \subseteq \mathcal{P}(S) \ , \quad \mathcal{C}_{int}(\tau) := \left\{ I \in \mathcal{C}(\tau) \setminus \{S\} \mid |I| \geq 2 \right\} \ , \quad (3.4)$$

where $\mathcal{P}(S)$ denotes the power set of S .

Compatibility

Definition 3.1 ([195, 80]) Subsets $A, B \subset S$ are said to be compatible, $A \bowtie B$, if

$$A \cap B = \emptyset \vee A \subseteq B \vee B \subseteq A. \quad (3.5)$$

If $A \not\bowtie B$, then we say that A and B cross. We further extend the compatibility relation (\bowtie) as follows:

- For $\mathcal{A}, \mathcal{B} \subseteq \mathcal{P}(S)$, write $\mathcal{A} \bowtie \mathcal{B}$ if $A \bowtie B$ for all $A \in \mathcal{A}$ and $B \in \mathcal{B}$;
- For a cluster $I \subseteq S$ and a tree τ over the leaf set S , write $I \bowtie \tau$ if $\{I\} \bowtie \mathcal{C}(\tau)$;
- For two trees σ and τ over the leaf set S , write $\sigma \bowtie \tau$ if $\mathcal{C}(\sigma) \bowtie \mathcal{C}(\tau)$.

By construction, any two elements of $\mathcal{C}(\tau)$ are compatible for any tree τ . This motivates the following definition:

Definition 3.2 ([195]) A subset \mathcal{A} of $\mathcal{P}(S)$ is said to be nested — also referred to in the literature as a “laminar family” — if any two elements of \mathcal{A} are compatible. $\mathcal{C}(\tau)$ is known as the laminar family associated with τ .

Hierarchical Relations

The cluster set $\mathcal{C}(\tau)$ of a hierarchy τ completely determines its representation as a rooted tree with labeled leaves: $\mathcal{C}(\tau)$ stands in bijective correspondence with the vertex set V_τ of τ , and (v, v') is an edge in τ if and only if $\mathcal{C}(v) \supset \mathcal{C}(v')$ and there is no vertex $\tilde{v} \in V_\tau$ such that $\mathcal{C}(v) \supset \mathcal{C}(\tilde{v}) \supset \mathcal{C}(v')$. Consequently, the standard notions of ancestor, descendant, parent and child of a vertex in common use for rooted trees carry over to the cluster representation as follows:

$$\text{Anc}(I, \tau) = \{J \in \mathcal{C}(\tau) \mid I \subsetneq J\}, \quad \text{Des}(I, \tau) = \{J \in \mathcal{C}(\tau) \mid J \subsetneq I\}, \quad (3.6a)$$

$$\text{Pr}(I, \tau) = \min(\text{Anc}(I, \tau)), \quad \text{Ch}(I, \tau) = \{J \in \mathcal{C}(\tau) \mid \text{Pr}(J, \tau) = I\}, \quad (3.6b)$$

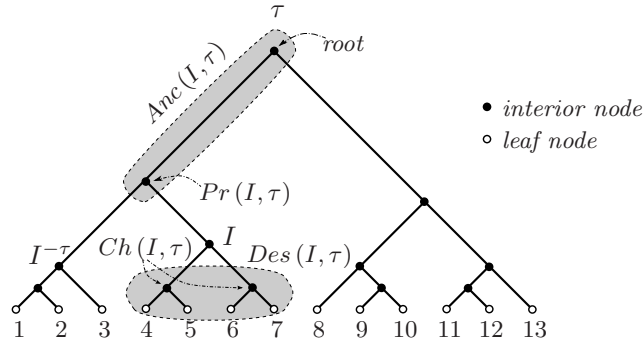


Figure 3.1: Hierarchical Relations: ancestors - $\text{Anc}(I, \tau)$, parent - $\text{Pr}(I, \tau)$, children - $\text{Ch}(I, \tau)$, descendants - $\text{Des}(I, \tau)$, and local complement (sibling) - $I^{-\tau}$ of cluster I of a rooted binary phylogenetic tree, $\tau \in \mathcal{BT}_{[13]}$. Filled and unfilled circles represent interior and leaf nodes, respectively. An interior node is referred to by its cluster, the list of leaves below it; for example, $I = \{4, 5, 6, 7\}$. Accordingly, the cluster set of τ is $\mathcal{C}(\tau) = \{\{1\}, \{2\}, \dots, \{13\}, \{1, 2\}, \{1, 2, 3\}, \{4, 5\}, \{6, 7\}, \{4, 5, 6, 7\}, \{1, 2, \dots, 7\}, \{9, 10\}, \{8, 9, 10\}, \{11, 12\}, \{11, 12, 13\}, \{8, 9, \dots, 13\}, \{1, 2, \dots, 13\}\}$.

where $\min(\text{Anc}(I, \tau))$ is computed with respect to the inclusion order. Note that for the trivial clusters we have $\text{Pr}(S, \tau) = \emptyset$ and $\text{Ch}((s), \tau) = \emptyset$ for $s \in S$.

Since the set of children partitions each parent, we find it useful to define the *local complement (sibling)* $I^{-\tau}$ of $I \in \mathcal{C}(\tau)$ as

$$I^{-\tau} := \text{Pr}(I, \tau) \setminus I, \quad (3.7)$$

not to be confused with the standard (global) complement, $I^C = S \setminus I$. Further, a grandchild in τ is a cluster $G \in \mathcal{C}(\tau)$ having a grandparent $\text{Pr}^2(G, \tau) := \text{Pr}(\text{Pr}(G, \tau), \tau)$ in τ . We denote the set of all grandchildren in τ by $\mathcal{G}(\tau)$,

$$\mathcal{G}(\tau) := \left\{ G \in \mathcal{C}(\tau) \mid \text{Pr}^2(G, \tau) \neq \emptyset \right\}. \quad (3.8)$$

If A, B are either elements of S or clusters of τ , it is convenient to have $(A \wedge B)_\tau$ denote the smallest (in terms of cardinality) common ancestor of A and B in τ . Finally, the depth (level) $\ell_\tau(I)$ of a cluster in a hierarchy τ is defined to equal the number of distinct ancestors of I in τ ,

$$\ell_\tau(I) := |\text{Anc}(I, \tau)|, \quad \forall I \in \mathcal{C}(\tau), \quad (3.9)$$

where $|\cdot|$ denotes the set cardinality operator.

Nondegeneracy

A rooted tree where every interior vertex has exactly two children is said to be *binary* or *nondegenerate*. All other trees are said to be *degenerate*. We will denote the sets of hierarchies over a fixed finite leaf set S by \mathcal{T}_S . The subset of nondegenerate hierarchies will be denoted by \mathcal{BT}_S .

Note that the laminar family $\mathcal{C}(\tau)$ of a degenerate tree τ may always be augmented with additional clusters while remaining nested (Definition 3.2). This leads to the well known result:

Remark 3.1 ([216, 195]). Let $\tau \in \mathcal{T}_S$. Then τ has at most $2|S| - 1$ vertices, with equality if and only if τ is nondegenerate, if and only if $\mathcal{C}(\tau)$ is a maximal laminar family in $\mathcal{P}(S)$ with respect to inclusion.¹

Consensus

Definition 3.3 ([117, 39]) For any set of trees T in \mathcal{T}_S , the *strict consensus tree* T_* of T is defined to be the tree consisting of all common clusters of trees in T , i.e.,

$$\mathcal{C}(T_*) = \bigcap_{\tau \in T} \mathcal{C}(\tau), \quad (3.10)$$

and the *loose consensus tree* T^* of T is the tree each of whose clusters is a cluster of at least one tree in T and is compatible with all trees in T , i.e.,

$$\mathcal{C}(T^*) = \left\{ I \in \bigcup_{\tau \in T} \mathcal{C}(\tau) \mid \forall \sigma \in T \quad I \bowtie \sigma \right\}. \quad (3.11)$$

¹In this chapter, we adopt the convention that a laminar family does not contain the empty set (as an element).

Note that the loose consensus tree T^* of T refines the strict consensus tree T_* , i.e., $\mathcal{C}(T^*) \supseteq \mathcal{C}(T_*)$.

3.1.2 Some Operations on Trees

NNI Moves

A convenient restatement of the standard definition of NNI walks of unrooted binary trees [184, 160] for rooted binary trees, illustrated in Figure 3.2, is:

Definition 3.4 Let $\sigma \in \mathcal{BT}_S$. We say that $\tau \in \mathcal{BT}_S$ is the result of performing a nearest neighbor interchange (NNI) move on σ at a grandchild $G \in \mathcal{G}(\sigma)$ (3.8) if

$$\mathcal{C}(\tau) = \left(\mathcal{C}(\sigma) \setminus \{\text{Pr}(G, \sigma)\} \right) \cup \{\text{Pr}^2(G, \sigma) \setminus G\}. \quad (3.12)$$

We often indicate this by writing $\tau = \text{NNI}(\sigma, G)$.

Note that the NNI move at cluster G on σ swaps cluster G with its parent's sibling $\text{Pr}(G, \sigma)^{-\sigma}$ to yield τ , depicted in Figure 3.2(left); and after an NNI move at cluster G of σ , grandchild G of grandparent $P = \text{Pr}^2(G, \sigma)$ with respect to σ becomes child G of parent $P = \text{Pr}(G, \tau)$ with respect to τ .

It is standard to say that $\sigma, \tau \in \mathcal{BT}_S$ are NNI-adjacent if and only if one can be obtained from the other by a single NNI move. Figure 3.2(left) illustrates the NNI moves on \mathcal{BT}_S and their inverses.

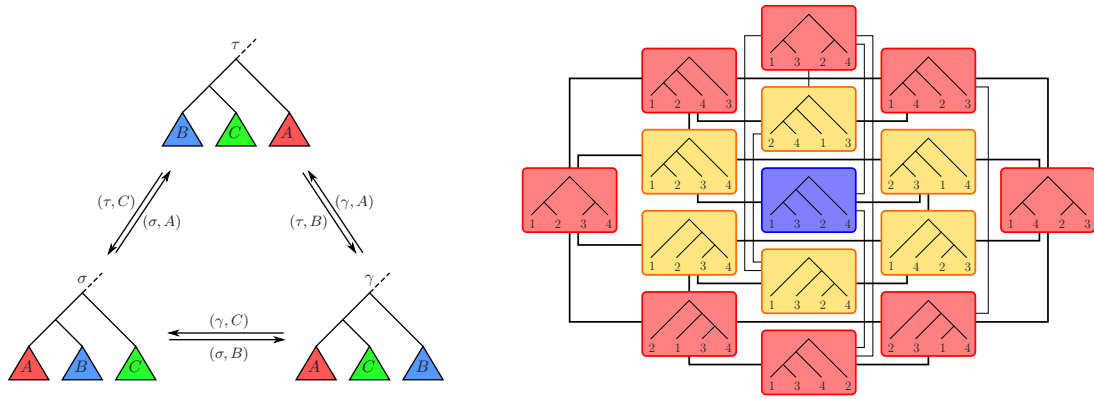


Figure 3.2: (left) An illustration of NNI moves between binary trees, each move (arrow) is labeled by its source tree and the grandchild defining the move. (right) The NNI Graph: a representation of the space of rooted binary trees, \mathcal{BT}_S , with NNI connectivity, for $S = [4] = \{1, 2, 3, 4\}$.

The NNI Graph

The NNI graph is formed over the vertex set \mathcal{BT}_S by declaring two trees to be connected by an edge if and only if they are NNI-adjacent, see e.g. Figure 3.2(right). We will work with a directed version of this graph:

Definition 3.5 The directed NNI graph $\mathcal{N}_S = (\mathcal{BT}_S, \mathcal{E}_{\mathcal{N}_S})$ is the directed graph on \mathcal{BT}_S with $(\sigma, \tau) \in \mathcal{E}_{\mathcal{N}_S}$ if and only if τ results from applying an NNI move to σ .

We will henceforth identify the notation for an NNI move (σ, G) , $G \in \mathcal{G}(\sigma)$ with the directed edge $(\sigma, \text{NNI}(\sigma, G)) \in \mathcal{E}_{\mathcal{N}_S}$ wherever there is no danger of confusion.

The (directed) NNI graph on n leaves is a regular graph of out-degree $2(n-2)$ [184]. Our description clarifies this by parametrizing the set of neighbors of $\tau \in \mathcal{BT}_S$ with its grandchildren, $|\mathcal{G}(\tau)| = 2(|S| - 2)$. The vertex set of the NNI graph is known to grow superexponentially with the number of leaves [35],

$$|\mathcal{BT}_{[n]}| = (2n-3)!! \stackrel{\text{def.}}{=} (2n-3)(2n-5)\dots 3 \cdot 1, \quad n \geq 2. \quad (3.13)$$

As a result, exploration of the NNI-graph (for example, searching for the shortest path between hierarchies or an optimal phylogenetic tree model) rapidly becomes impractical and costly as the number of leaves increases.

A useful observation for NNI-adjacent trees is:

Lemma 3.1 An ordered pair of hierarchies (σ, τ) is an edge in the NNI graph \mathcal{N}_S if and only if there exists an ordered triple (A, B, C) of common clusters of σ and τ such that $\{A \cup B\} = \mathcal{C}(\sigma) \setminus \mathcal{C}(\tau)$ and $\{B \cup C\} = \mathcal{C}(\tau) \setminus \mathcal{C}(\sigma)$. The triple (A, B, C) is uniquely determined by the pair (σ, τ) and will be referred to as the NNI-triplet associated with (σ, τ) .

Proof The proof amounts to a formal restatement of the observations made in Figure 3.2.

Sufficiency is directly evident from Definition 3.4, because the cluster sets of a pair of nondegenerate hierarchies differ exactly by one cluster if and only if they are NNI-adjacent.

To verify necessity, let the NNI move (σ, P) on σ at $P \in \mathcal{G}(\sigma)$ joins σ to τ , and $R = P^{-\sigma}$ and $Q = \text{Pr}^2(P, \sigma) \setminus \text{Pr}(P, \sigma)$. By Definition 3.4, $\{\text{Pr}(P, \sigma)\} = \{P \cup R\} = \mathcal{C}(\sigma) \setminus \mathcal{C}(\tau)$ and $\{\text{Pr}^2(P, \sigma) \setminus P\} = \{R \cup Q\} = \mathcal{C}(\tau) \setminus \mathcal{C}(\sigma)$. Further, (P, R, Q) is the only ordered triple of common clusters of σ and τ with the property that $\{P \cup R\} = \mathcal{C}(\sigma) \setminus \mathcal{C}(\tau)$ and $\{R \cup Q\} = \mathcal{C}(\tau) \setminus \mathcal{C}(\sigma)$, because the cluster sets of any two NNI-adjacent hierarchies differ exactly by one element. ■

Observe that the triplet in reverse order (C, B, A) is the NNI-triplet associated with the edge (τ, σ) . Also note that the NNI moves on σ at A and on τ at C yield τ and σ , respectively.

Tree Restriction

Definition 3.6 Let S be a fixed finite set and $K \subseteq S$. The restriction map $\text{res}_K : \mathcal{P}(S) \rightarrow \mathcal{P}(K)$ is defined to be

$$\text{res}_K(\mathcal{A}) := \{A \cap K \mid A \in \mathcal{A}, A \cap K \neq \emptyset\} \quad (3.14)$$

for any $\mathcal{A} \subseteq \mathcal{P}(S)$. It is convenient to have $\mathcal{A}|_K$ denote $\text{res}_K(\mathcal{A})$. For $\sigma \in \mathcal{T}_K$ and $\tau \in \mathcal{T}_S$ we will write:

$$\sigma = \text{res}_K(\tau) = \tau|_K \iff \mathcal{C}(\sigma) = \mathcal{C}(\tau)|_K. \quad (3.15)$$

Remark 3.2. Let $\tau \in \mathcal{BT}_S$ and $\{L, R\} = \text{Ch}(S, \tau)$. Then one has $\mathcal{C}(\tau) = \mathcal{C}(\tau|_L) \cup \{S\} \cup \mathcal{C}(\tau|_R)$.

Lemma 3.2 *For any finite set S and $K \subseteq S$ with $|K| \geq 2$, $\text{res}_K(\mathcal{BT}_S) = \mathcal{BT}_K$.*

Proof To observe that $\text{res}_K(\mathcal{BT}_S) \supseteq \mathcal{BT}_K$, consider any two nondegenerate trees $\sigma \in \mathcal{BT}_K$ and $\gamma \in \mathcal{BT}_{S \setminus K}$, and let $\tau \in \mathcal{BT}_S$ be the nondegenerate tree with cluster set $\mathcal{C}(\tau) = \mathcal{C}(\sigma) \cup \{S\} \cup \mathcal{C}(\gamma)$. Note that $\text{Ch}(S, \tau) = \{K, S \setminus K\}$. Hence, we have from Remark 3.2 that $\sigma = \text{res}_K(\tau)$.

To prove that $\text{res}_K(\mathcal{BT}_S) \subseteq \mathcal{BT}_K$, let $\tau \in \mathcal{BT}_S$ and $I \in \mathcal{C}(\tau)$ with the property that $|I \cap K| \geq 2$. Note that $I \cap K$ is an interior cluster of $\tau|_K$. We shall show that the cluster $I \cap K \in \mathcal{C}(\tau|_K)$ always admits a bipartition in $\tau|_K$. That is to say, there exist a cluster $A \in \mathcal{C}(\tau)$ with children $\{A_L, A_R\} = \text{Ch}(A, \tau)$ such that $A \cap K = I \cap K$ and $A_L \cap K \neq \emptyset$ and $A_R \cap K \neq \emptyset$. Hence, $\text{Ch}(I \cap K, \tau|_K) = \{A_L \cap K, A_R \cap K\}$.

Now observe that either $I_L \cap K \neq \emptyset$ and $I_R \cap K \neq \emptyset$ for $\{I_L, I_R\} = \text{Ch}(I, \tau)$, or there exists one and only one descendant $D \in \text{Des}(I, \tau)$ with $\{D_L, D_R\} = \text{Ch}(D, \tau)$ such that $I \cap K = D \cap K$ and $D_L \cap K \neq \emptyset$ and $D_R \cap K \neq \emptyset$.

Thus, all the interior clusters of $\tau|_K$ have exactly two children, which completes the proof. \blacksquare

3.1.3 Dissimilarities, Metrics and Ultrametrics

A *dissimilarity measure* on X , or simply a *dissimilarity*, is a real-valued nonnegative symmetric function $d : X \times X \rightarrow \mathbb{R}_{\geq 0}$ on $X \times X$ satisfying $d(x, x) = 0$ for all $x \in X$.² Recall that a dissimilarity d on X is *positive definite* if $d(x, y) = 0$ implies $x = y$ for all $x, y \in X$. For instance, many approximations of the (NP-hard) NNI metric are positive definite dissimilarities [142, 64, 42]. A dissimilarity d is a metric if it satisfies the triangle inequality,

$$d(x, y) \leq d(x, z) + d(z, y), \quad \forall x, y, z \in X. \quad (3.16)$$

For example, recall the definition of the commonly-used Robinson-Foulds metric on $X = \mathcal{T}_S$:

Definition 3.7 ([185]) *The Robinson-Foulds distance d_{RF} on \mathcal{T}_S is defined by:*³

$$d_{RF}(\sigma, \tau) = \frac{1}{2} |\mathcal{C}(\sigma) \ominus \mathcal{C}(\tau)|, \quad \sigma, \tau \in \mathcal{T}_S. \quad (3.17)$$

Recently a more discriminative metric was introduced:

Definition 3.8 ([36, 143]) *Let $\sigma, \tau \in \mathcal{BT}_S$ and $G_S(\sigma, \tau)$ denote the complete bipartite graph with sides $\mathcal{C}_{int}(\sigma)$ and $\mathcal{C}_{int}(\tau)$ with each edge $(I, J) \in \mathcal{C}_{int}(\sigma) \times \mathcal{C}_{int}(\tau)$ carrying the weight*⁴ $A_S(I, J) = \min(|I \ominus J|, |I \ominus J^C|)$.

The matching split distance $d_{MS} : \mathcal{BT}_S \times \mathcal{BT}_S \rightarrow \mathbb{R}_{\geq 0}$ between a pair of hierarchies σ and τ is defined to be the value of a minimum-weighted perfect matching in $G_S(\sigma, \tau)$.

It is known that $d_{RF} \leq d_{MS} \leq \frac{|S|+1}{2} d_{RF}$ [36], which explains the improvement of d_{MS} over d_{RF} in discriminative power. At the same time, the cost of computing a minimum weighted perfect matching in any $G_S(\sigma, \tau)$ is $O(|S|^{2.5} \log |S|)$, which motivates the search

²We use \mathbb{R} and $\mathbb{R}_{\geq 0}$ to denote the real line and the set of nonnegative reals, respectively.

³Here, \ominus denotes the symmetric set difference, i.e. $A \ominus B = (A \setminus B) \cup (B \setminus A)$ for any sets A and B .

⁴This corresponds to the Hamming distance of clusters.

for dissimilarities producing similar improvement in discriminative power (bounding d_{RF} from above) yet having a lower computational cost than that of d_{MS} .

Recall that an ultrametric d on X is a metric on X satisfying the strengthened triangle inequality:

$$d(x, y) \leq \max(d(x, z), d(z, y)), \quad \forall x, y, z \in X. \quad (3.18)$$

A restatement of a well-known fact (see, e.g., [47, 115, 180]) revealing the relation between hierarchies and ultrametrics is:

Lemma 3.3 *Let $\tau \in \mathcal{T}_S$ and $h_\tau : \mathcal{C}(\tau) \rightarrow \mathbb{R}_{\geq 0}$. For any $i, j \in S$, let $(i \wedge j)_\tau$ denote the smallest cluster in $\mathcal{C}(\tau)$ containing the pair $\{i, j\}$. Then the dissimilarity on S given by*

$$d_\tau(i, j) := h_\tau((i \wedge j)_\tau), \quad i, j \in S, \quad (3.19)$$

is an ultrametric if and only if the following conditions are satisfied for any $I, J \in \mathcal{C}(\tau)$:

- (i) *if $I \subseteq J$, then $h_\tau(I) \leq h_\tau(J)$,*
- (ii) *$h_\tau(I) = 0$ if and only if $|I| = 1$.*

Proof The proof of the sufficiency for being an ultrametric is as follows. Positive definiteness and symmetry of d_τ are evident from (3.19) and Lemma 3.3.(i)-(ii). To show the strong triangle inequality, let $i \neq j \neq k \in S$ and $I = (i \wedge j)_\tau$, and so $d_\tau(i, j) = h_\tau(I)$. Accordingly, let $\{I_i, I_j\} \subseteq \text{Ch}(I, \tau)$ with the property that $i \in I_i$ and $j \in I_j$.

If $k \in I$, without loss of generality, let $k \in I_i$, and so $k \notin I_j$. Then, using (3.19) and Lemma 3.3.(i), one can verify that $d_\tau(i, k) \leq h_\tau(I_i) \leq h_\tau(I)$ and $d_\tau(j, k) = h_\tau(I)$ because $(i \wedge k)_\tau \subseteq I_i$ and $(j \wedge k)_\tau = I$. Also note that if neither $k \in I_i$ nor $k \in I_j$ (but still $k \in I$), then $d_\tau(i, k) = d_\tau(j, k) = h_\tau(I)$ since $(i \wedge k)_\tau = (j \wedge k)_\tau = I$. Similarly, if $k \notin I$, then $d_\tau(i, k) \geq h_\tau(I)$ and $d_\tau(j, k) \geq h_\tau(I)$ because only some ancestors of I in τ might contain all i, j, k . Therefore, overall, one always has $d_\tau(i, j) \leq \max(d_\tau(i, k), d_\tau(k, j))$, which completes the proof of the sufficiency.

Let us continue with the necessity for being an ultrametric. Note that Lemma 3.3.(ii) directly follows from positive definiteness of d_τ . Let $I \in \mathcal{C}(\tau) \setminus \{S\}$ be any nonsingleton cluster of τ and $i \neq j \in I$ with the property that $(i \wedge j)_\tau = I$. For any $k \in I^{-\tau}$, we always have $(i \wedge k)_\tau = (j \wedge k)_\tau = \text{Pr}(I, \tau)$. Now, using the strong triangle inequality of d_τ , one can deduce Lemma 3.3.(i) from

$$h_\tau(I) = d_\tau(i, j) \leq \max(d_\tau(i, k), d_\tau(j, k)) = h_\tau(\text{Pr}(I, \tau)), \quad (3.20)$$

which completes the proof. ■

Recall that a set X may always inherit a metric from a metric space (Y, d_Y) by pullback: any injective map $f : X \rightarrow Y$ of X into Y yields a metric d_X on X defined by $d_X(x_1, x_2) := d_Y(f(x_1), f(x_2))$, where $x_1, x_2 \in X$, and known as the pullback of d_Y along f . For example, the RF metric is a pullback: it is common knowledge that the set $F(X)$ of all finite subsets of a set X forms a metric space under the metric $d(A, B) = |A \ominus B|$, which is one of the ways of defining Hamming distance; thus, the RF distance is (one half times) the pullback of this metric on $F(\mathcal{P}(S))$ under the map $\tau \mapsto \mathcal{C}(\tau)$.

3.2 Quantifying Incompatibility

3.2.1 The Cluster-Cardinality Distance

We now introduce an embedding of hierarchies into the space of matrices based on the relation between hierarchies and ultrametrics, summarized in Lemma 3.3:

Definition 3.9 *The ultrametric representation is the map $\mathbf{U} : \mathcal{T}_S \rightarrow \mathbb{R}^{|S| \times |S|}$ defined by $\mathbf{U}(\tau)_{ij} := h((i \wedge j)_\tau)$, where $h : \mathcal{P}(S) \rightarrow \mathbb{N}$ is set to be $h(I) := |I| - 1$ for $I \subseteq S$.*

Lemma 3.4 *The map \mathbf{U} is injective.*

Proof To see the injectivity of \mathbf{U} (Definition 3.9), we shall show that $\mathbf{U}(\sigma) \neq \mathbf{U}(\tau)$ for any $\sigma \neq \tau \in \mathcal{T}_S$.

Two trees $\sigma, \tau \in \mathcal{T}_S$ are distinct if and only if they have at least one unshared cluster. Accordingly, for any $\sigma \neq \tau \in \mathcal{T}_S$ consider a common cluster $I \in \mathcal{C}(\sigma) \cap \mathcal{C}(\tau)$ with distinct parents $\text{Pr}(I, \sigma) \neq \text{Pr}(I, \tau)$. Depending on the cardinality of parent clusters:

- If $|\text{Pr}(I, \sigma)| = |\text{Pr}(I, \tau)|$, then observe that there exists some $j \in \text{Pr}(I, \sigma)$ such that $j \notin \text{Pr}(I, \tau)$ because $\text{Pr}(I, \sigma) \neq \text{Pr}(I, \tau)$. In fact, notice that $j \in I^{-\sigma}$ and $j \notin I^{-\tau}$ (recall (3.7)). Hence, for any $i \in I$ we have $(i \wedge j)_\sigma = \text{Pr}(I, \sigma)$ and $\text{Pr}(I, \tau) \subsetneq (i \wedge j)_\tau$. Thus, it follows from Definition 3.9 that for any $i \in I$

$$\mathbf{U}(\sigma)_{ij} = |\text{Pr}(I, \sigma)| - 1 < \mathbf{U}(\tau)_{ij} = |(i \wedge j)_\tau| - 1. \quad (3.21)$$

- Otherwise, without loss of generality, let $|\text{Pr}(I, \sigma)| < |\text{Pr}(I, \tau)|$. Then, observe that for any $i \in I$ and $j \in I^{-\sigma}$,

$$\mathbf{U}(\sigma)_{ij} = |\text{Pr}(I, \sigma)| - 1 < \mathbf{U}(\tau)_{ij} = |(i \wedge j)_\tau| - 1, \quad (3.22)$$

since $(i \wedge j)_\tau \supseteq \text{Pr}(I, \tau)$.

Therefore, for any $\sigma \neq \tau \in \mathcal{B}\mathcal{T}_S$ one has $\mathbf{U}(\sigma) \neq \mathbf{U}(\tau)$, and the result follows. \blacksquare

Using the embedding \mathbf{U} of \mathcal{T}_S into $\mathbb{R}^{|S| \times |S|}$, we can construct tree metrics as pullback metrics induced from matrix norms, such as the one below:

Definition 3.10 The cluster-cardinality metric, $d_{CC} : \mathcal{T}_S \times \mathcal{T}_S \rightarrow \mathbb{R}_{\geq 0}$, on \mathcal{T}_S is defined to be ⁵

$$d_{CC}(\sigma, \tau) := \frac{1}{2} \|\mathbf{U}(\sigma) - \mathbf{U}(\tau)\|_1, \quad \forall \sigma, \tau \in \mathcal{T}_S. \quad (3.23)$$

Proposition 3.1 *The cluster-cardinality distance d_{CC} on \mathcal{T}_S is computable in $O(|S|^2)$ time.*

Proof The 1-norm of the difference of a pair of $|S| \times |S|$ matrices obviously requires $O(|S|^2)$ time to compute, giving a lower bound on the computation cost of d_{CC} . It remains to show that the embedding \mathbf{U} (Definition 3.9) may be obtained at this cost.

We proceed by induction based on a post-order traversal of the trees involved, $\tau \in \mathcal{T}_S$.

⁵Here $\|\cdot\|_1$ denotes the 1-norm of a matrix, i.e., $\|\mathbf{U}\|_1 := \sum_{i=1}^n \sum_{j=1}^n |\mathbf{U}_{ij}|$ for $\mathbf{U} \in \mathbb{R}^{n \times n}$. Our choice of the 1-norm was guided by the resulting relationships between d_{CC} and the dissimilarity measures d_{CM} and d_{nav} introduced below. Other choices of norm on $\mathbb{R}^{S \times S}$ may prove useful.

For the base case, consider the two-leaf tree $\tau \in \mathcal{BT}_{[2]}$, i.e. $|S| = 2$: then we simply assign $\mathbf{U}(\tau) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$.

For the induction step, assume $|S| \geq 3$ and denote $\text{Ch}(S, \tau) = \{S_k\}_{1 \leq k \leq K}$, where $K \geq 2$ is the number of children of the root S in τ . We observe:

- For every singleton child $\{i\}$ of S in τ (if any), then set $\mathbf{U}(\tau)_{ii} = 0$, which takes up $O(1)$ time.
- Note that all clusters of τ and their sizes can be obtained in $O(|S|^2)$ time by a single post-order traversal, as each individual cluster (as well as its cardinality) takes at most linear time to compute from those of its children.
- Suppose that for any $1 \leq k \leq K$ and $|S_k| \geq 2$ the elements of $\mathbf{U}(\tau)$ associated with the subtree rooted at S_k can be computed in $O(|S_k|^2)$ time. Then, the total number of updates associated with the root S is $\sum_{k=1}^K \sum_{l=1}^K |S_k| |S_l|$ and corresponds to setting $\mathbf{U}(\tau)_{ij} = \mathbf{U}(\tau)_{ji} = |S| - 1$ for all $i \in S_k, j \in S_l$ and $1 \leq k, l \leq K$.

In total, the cost of obtaining $\mathbf{U}(\tau)$ is $\sum_{k=1}^K O(|S_k|^2) + \sum_{k=1}^K \sum_{l=1}^K |S_k| |S_l| + O(|S|^2) = O(|S|^2)$, as required. \blacksquare

The diameter, $\text{diam}(X, d) := \max \{d(x, y) \mid x, y \in X\}$, of a finite metric space (X, d) is always of interest in algorithmic applications. Some known diameters for hierarchies [36, 143, 201] are:

$$\text{diam}(\mathcal{T}_S, d_{RF}) = |S| - 2, \quad (3.24)$$

$$\text{diam}(\mathcal{BT}_S, d_{MS}) = O(|S|^2), \quad (3.25)$$

$$\text{diam}(\mathcal{BT}_S, d_{NNI}) = O(|S| \log |S|), \quad (3.26)$$

For the cluster-cardinality distance, we have:

Proposition 3.2 $\text{diam}(\mathcal{T}_S, d_{CC}) = O(|S|^3)$.

Proof From Definition 3.9, the minimum and maximum ultrametric distances between two distinct elements of S are, respectively, 1 and $|S| - 1$, implying the bound

$$\max_{i, j \in S} (\mathbf{U}(\sigma)_{ij} - \mathbf{U}(\tau)_{ij}) \leq |S| - 2 \quad \forall \sigma, \tau \in \mathcal{T}_S. \quad (3.27)$$

Hence, it follows from (3.23) that the diameter of \mathcal{T}_S with respect to d_{CC} is bounded above as follows:

$$\text{diam}(\mathcal{T}_S, d_{CC}) \leq \frac{1}{2} |S| (|S| - 1) (|S| - 2). \quad (3.28)$$

Now consider two NNI-adjacent binary trees $\sigma, \tau \in \mathcal{BT}_S$ such that the NNI triple (A, B, C) associated with (σ, τ) (see Lemma 3.1) satisfies $|A| = |B| = |C| = \left\lfloor \frac{|S|}{3} \right\rfloor$. It is straightforward to observe that for $|S| \geq 3$ there always exists such a pair of NNI-adjacent trees, because A, B , and C are disjoint and $|A| + |B| + |C| = 3 \left\lfloor \frac{|S|}{3} \right\rfloor \leq |S|$. Hence, we have from Proposition 3.3 that $d_{CC}(\sigma, \tau) = 2|A||B||C| = 2 \left\lfloor \frac{|S|}{3} \right\rfloor^3$, which yields the following

lower bound on the diameter of \mathcal{T}_S with respect to d_{CC} ,

$$2 \left\lfloor \frac{|S|}{3} \right\rfloor^3 \leq \text{diam}(\mathcal{T}_S, d_{CC}). \quad (3.29)$$

Note that these bounds on $\text{diam}(\mathcal{T}_S, d_{CC})$ in (3.28) and (3.29) hold for all $|S| \geq 2$. Thus, the result follows. \blacksquare

A common question regarding any distance being proposed for the space of trees is how it behaves with respect to certain tree rearrangements. For instance, any pair of NNI-adjacent trees, $\sigma, \tau \in \mathcal{BT}_S$, are known to satisfy [36]⁶

$$d_{NNI}(\sigma, \tau) = 1 \iff d_{RF}(\sigma, \tau) = 1, \quad (3.30)$$

$$d_{NNI}(\sigma, \tau) = 1 \implies 2 \leq d_{MS}(\sigma, \tau) \leq \left\lfloor \frac{|S|}{2} \right\rfloor. \quad (3.31)$$

Similarly for d_{CC} we have:

Proposition 3.3 *Let (σ, τ) be an edge of the NNI-graph $\mathcal{N}_S = (\mathcal{BT}_S, \mathcal{E}_{\mathcal{N}_S})$ and (A, B, C) be the associated NNI triplet (Lemma 3.1). Then*

$$2 \leq d_{CC}(\sigma, \tau) = 2|A||B||C| \leq \left\lfloor \frac{2}{27} |S|^3 \right\rfloor, \quad (3.32)$$

and both bounds are tight.

Proof Let $P = A \cup B \cup C$ and recall from Lemma 3.1 that $A \cup B \in \mathcal{C}(\sigma)$ and $B \cup C \in \mathcal{C}(\tau)$. Note that $P \in \mathcal{C}(\sigma) \cap \mathcal{C}(\tau)$ is a common (grand)parent cluster, and A, B and C are pairwise disjoint.

Since the NNI moves between σ and τ only change the relative relations of clusters A, B and C , the distance between σ and τ can be rewritten as

$$d_{CC}(\sigma, \tau) = \frac{1}{2} \|\mathbf{U}(\sigma) - \mathbf{U}(\tau)\|_1, \quad (3.33)$$

$$= \sum_{\substack{i \in A \\ j \in B}} |\mathbf{U}(\sigma)_{ij} - \mathbf{U}(\tau)_{ij}| + \sum_{\substack{i \in A \\ j \in C}} |\mathbf{U}(\sigma)_{ij} - \mathbf{U}(\tau)_{ij}| + \sum_{\substack{i \in B \\ j \in C}} |\mathbf{U}(\sigma)_{ij} - \mathbf{U}(\tau)_{ij}|, \quad (3.34)$$

$$= \sum_{\substack{i \in A \\ j \in B}} \underbrace{|h(A \cup B) - h(P)|}_{=|C|} + \sum_{\substack{i \in A \\ j \in C}} \underbrace{|h(P) - h(P)|}_{=0} + \sum_{\substack{i \in B \\ j \in C}} \underbrace{|h(P) - h(B \cup C)|}_{=|A|}, \quad (3.35)$$

$$= 2|A||B||C|. \quad (3.36)$$

Clearly, the lower bound in (3.32) is realized when $|A| = |B| = |C| = 1$. Since the maximum product of three numbers with a prescribed sum occurs when all the numbers are equal — in our case, $|A| + |B| + |C| \leq |S|$ — we must have $|A||B||C| \leq \left\lfloor \frac{|S|^3}{27} \right\rfloor$, as $\lfloor \cdot \rfloor$ is integer-valued. The result follows. \blacksquare

⁶ $\lfloor \cdot \rfloor$ denotes the floor operator returning the largest integer not greater than its operand.

Inequalities of the above form allow one to take advantage of the combinatorial nature of d_{NNI} through repeated application of the triangle inequality:

Corollary 3.1 *Over \mathcal{BT}_S one has $d_{RF} \leq d_{NNI}$.*

Proof Let $\sigma, \tau \in \mathcal{BT}_S$ and let $\Gamma = (\gamma_k)_{0 \leq k \leq K}$ be shortest path in the NNI graph $\mathcal{N}_S = (\mathcal{BT}_S, \mathcal{E}_{\mathcal{N}_S})$ from $\sigma = \gamma_0$ to $\tau = \gamma_K$. This means that $(\gamma_{k-1}, \gamma_k) \in \mathcal{E}_{\mathcal{N}_S}$ – or, equivalently, $d_{NNI}(\gamma_{k-1}, \gamma_k) = 1$ – for all $1 \leq k \leq K$, and that $K = d_{NNI}(\sigma, \tau)$. Repeatedly applying the triangle inequality for d_{RF} and then equation (3.30), we obtain:

$$d_{RF}(\sigma, \tau) \leq \sum_{k=1}^K d_{RF}(\gamma_{k-1}, \gamma_k) = \sum_{k=1}^K d_{NNI}(\gamma_{k-1}, \gamma_k) = K = d_{NNI}(\sigma, \tau) , \quad (3.37)$$

which completes the proof. ■

Indeed, the length of a path in \mathcal{N}_S produces a bound on the RF distance between its endpoints by repeatedly applying the triangle inequality to (3.30). A similar argument yields:

Corollary 3.2 *Let d be a dissimilarity on \mathcal{BT}_S with the property that $d(\sigma, \tau) \leq 1$ for any pair of NNI-adjacent hierarchies $\sigma, \tau \in \mathcal{BT}_S$. If $d(\sigma, \tau) > d_{NNI}(\sigma, \tau)$ for some $\sigma, \tau \in \mathcal{BT}_S$, then d is not a metric.*

Proof Assume, on the contrary, that d is a metric. Then the argument of the proof of Corollary 3.1 may be repeated, replacing d_{RF} with d and reaching the conclusion that $d(\sigma, \tau) \leq d_{NNI}(\sigma, \tau)$ for all $\sigma, \tau \in \mathcal{BT}_S$ – contradiction. ■

3.2.2 The Crossing Dissimilarity

Definition 3.11 *Let $\sigma, \tau \in \mathcal{T}_S$. We define their compatibility matrix $\mathbf{C}(\sigma, \tau)$ and their crossing matrix $\mathbf{X}(\sigma, \tau)$ to be ⁷*

$$\mathbf{C}(\sigma, \tau)_{I,J} := \mathbb{1}(I \bowtie J) , \quad \text{and} \quad \mathbf{X}(\sigma, \tau)_{I,J} := 1 - \mathbf{C}(\sigma, \tau)_{I,J} , \quad (3.38)$$

where $I \in \mathcal{C}(\sigma)$, $J \in \mathcal{C}(\tau)$, and $\mathbb{1}(\cdot)$ denotes the indicator function returning unity if its argument holds true and zero otherwise. The crossing dissimilarity d_{CM} is defined by ⁸

$$d_{CM}(\sigma, \tau) := \|\mathbf{X}(\sigma, \tau)\|_1 , \quad (3.39)$$

counting the pairs of incompatible clusters in $\mathcal{C}(\sigma) \cup \mathcal{C}(\tau)$.

We list some useful properties of d_{CM} :

⁷ $\mathbf{C}(\sigma, \tau)$ and $\mathbf{X}(\sigma, \tau)$ can be defined only in terms of nontrivial clusters of σ and τ since any trivial cluster of σ and τ is compatible with any cluster $K \subseteq S$. As a result, we are required to separately consider the special case in which one of the trees has only trivial clusters whenever \mathbf{C} or \mathbf{X} are used to reason about degenerate trees.

⁸We find that choosing to use the 1-norm of the crossing matrix easily reveals combinatorial relations between d_{CM} (3.39) and d_{CC} (3.23); of course, one could use other matrix norms to construct alternative dissimilarities.

Remark 3.3. The crossing dissimilarity d_{CM} on \mathcal{BT}_S is positive definite and symmetric, but it is not a metric (apply Corollary 3.2 to the observations of Figure 3.3).

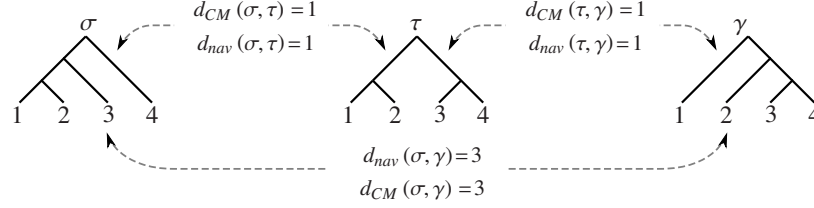


Figure 3.3: d_{CM} and d_{nav} are not metrics: an example of the triangle inequality failing for both dissimilarities.

Proposition 3.4 *The crossing dissimilarity d_{CM} over \mathcal{T}_S can be computed in $O(|S|^2)$ time.*

Proof The crossing matrix $\mathbf{X}(\sigma, \tau)$ (3.38) of a pair of hierarchies $\sigma, \tau \in \mathcal{T}_S$ has at most $2|S| - 1$ rows and columns. Hence, the 1-norm of $\mathbf{X}(\sigma, \tau)$ requires $O(|S|^2)$ time to compute, bounding the cost of d_{CM} from below. To obtain the upper bound, we show that $\mathbf{X}(\sigma, \tau)$ can be obtained in $O(|S|^2)$ time by post-order traversal.

Observe that for any cluster $J \in \mathcal{C}(\tau)$ (and symmetrically, for any cluster of $\mathcal{C}(\sigma)$) one can check whether J is disjoint with or a superset of each cluster I of σ by a post-order traversal of σ in $O(|S|)$ time using the following recursion:

- If either I or J is a singleton then the cluster inclusions $I \subseteq J$, $J \subseteq I$ and their disjointness can be determined in constant time using a hash map.
- Otherwise ($|I| \geq 2$ and $|J| \geq 2$), we have

$$I \subseteq J \iff \forall D \in \text{Ch}(I, \sigma) \quad D \subseteq J, \quad (3.40)$$

$$I \cap J = \emptyset \iff \forall D \in \text{Ch}(I, \sigma) \quad D \cap J = \emptyset. \quad (3.41)$$

Thus, it follows from Definition 3.1 that a complete list of compatibilities between σ and τ can be produced in $O(|S|^2)$ time, and so $\mathbf{X}(\sigma, \tau)$ can be obtained at the same cost, $O(|S|^2)$. \blacksquare

Proposition 3.5 $\text{diam}(\mathcal{T}_S, d_{CM}) = (|S| - 2)^2$.

Proof Two clusters of a pair of trees can only be incompatible if they are both nontrivial. Recall from Remark 3.1 that the number of nontrivial clusters of a tree in \mathcal{T}_S is at most $|S| - 2$. Hence, by definition (3.39), an upper bound on $\text{diam}(\mathcal{T}_S, d_{CM})$ is $(|S| - 2)^2$. To observe that this upper bound is realized, see Figure 3.4. \blacksquare

Proposition 3.6 *Two nondegenerate trees $\sigma, \tau \in \mathcal{BT}_S$ are NNI-adjacent if and only if $d_{CM}(\sigma, \tau) = 1$.*

Proof The result is evident from Remark 3.1 and Definition 3.4. \blacksquare

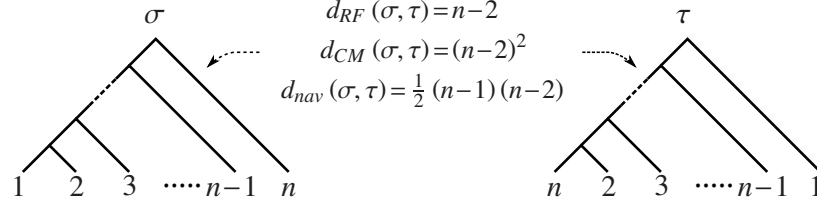


Figure 3.4: A pair of nondegenerate hierarchies realizing $\text{diam}(\mathcal{T}_{[n]}, d_{CM}) = (n-2)^2$ and $\text{diam}(\mathcal{BT}_{[n]}, d_{nav}) = \frac{1}{2}(n-1)(n-2)$.

Despite the result of the last proposition, d_{CM} does not provide a linear lower bound on d_{NNI} since $\text{diam}(\mathcal{BT}_S, d_{NNI}) = O(|S| \log |S|) < \text{diam}(\mathcal{BT}_S, d_{CM}) = O(|S|^2)$ (Proposition 3.5). This inequality provides us with an additional, more conceptual, argument that d_{CM} is not a metric, by applying Corollary 3.2.

Proposition 3.7 *Over \mathcal{T}_S one has $d_{RF} \leq d_{CM} \leq d_{RF}^2$. These bounds are tight.*

Proof The lower bound directly follows from Remark 3.1. Because a pair of distinct binary hierarchies always have uncommon clusters whose count is equal to d_{RF} , and an unshared cluster of one tree crosses at least one unshared cluster of the other tree. This bound is tight since for any $\sigma, \tau \in \mathcal{BT}_S$,

$$d_{RF}(\sigma, \tau) = 1 \iff d_{NNI}(\sigma, \tau) = 1 \iff d_{CM}(\sigma, \tau) = 1. \quad (3.42)$$

For any $\sigma, \tau \in \mathcal{BT}_S$, the columns and rows of $\mathbf{X}(\sigma, \tau)$ (3.38) associated with common clusters of σ, τ are necessarily null. Hence, each nonzero element of $\mathbf{X}(\sigma, \tau)$ is associated with a pair of unshared clusters of σ and τ , i.e., $\mathbf{X}(\sigma, \tau)_{I,J} \neq 0$ for some $I \in \mathcal{C}(\sigma)$, $J \in \mathcal{C}(\tau)$ implies $I \notin \mathcal{C}(\tau)$ and $J \notin \mathcal{C}(\sigma)$. By the definition of d_{RF} , there are no more than $d_{RF}(\sigma, \tau)^2$ such pairs — hence the claimed upper bound. To observe that this bound is also tight, see Figure 3.4. ■

Proposition 3.8 *Over \mathcal{T}_S one has $d_{CM} \leq d_{CC}$.*

Proof Given any $\sigma, \tau \in \mathcal{T}_S$, we claim that there is a function $q : \mathcal{C}(\sigma) \times \mathcal{C}(\tau) \rightarrow S \times S$ with the following properties:

- (i) for any $I \in \mathcal{C}(\sigma)$ and $J \in \mathcal{C}(\tau)$, $I \bowtie J$ if and only if $(i, j) = q(I, J)$ with $i = j$,
- (ii) for any $i \neq j \in S$, $|q^{-1}(i, j)| \leq |\mathbf{U}(\sigma)_{ij} - \mathbf{U}(\tau)_{ij}|$.

Observe that, if such a function does exist, then (i) implies:

$$\bigcup_{i \neq j \in S} q_{\sigma, \tau}^{-1}(i, j) = \left\{ (I, J) \in \mathcal{C}(\sigma) \times \mathcal{C}(\tau) \mid I \not\bowtie J \right\}. \quad (3.43)$$

It is then evident from (3.43) and (ii) that

$$d_{CM}(\sigma, \tau) \leq \sum_{i \neq j \in S} |q_{\sigma, \tau}^{-1}(i, j)| \leq d_{CC}(\sigma, \tau), \quad (3.44)$$

proving our proposition.

We proceed to construct the function q . If $I \not\bowtie J$, then there exist $i \in I \cap J$ and $j \in I \setminus J$ with the property that $(i \wedge j)_\sigma = I$. Accordingly, define

$$Q(I, J) := \left\{ (i, j) \in S \times S \mid i \in I \cap J, j \in I \setminus J, (i \wedge j)_\sigma = I \right\}, \quad (3.45)$$

$$R(I, J) := \left\{ (i, j) \in S \times S \mid i \in I \cap J, j \in J \setminus I, (i \wedge j)_\tau = J \right\}. \quad (3.46)$$

Note that if $(i, j) \in Q(I, J) \cup R(I, J)$, then $i \neq j$.

Have S totally ordered (say, by enumerating its elements) and have $S \times S$ ordered lexicographically according to the order of S . Then, define $q : \mathcal{C}(\sigma) \times \mathcal{C}(\tau) \rightarrow S \times S$ to be

$$q(I, J) := \begin{cases} (\min(I \cup J), \min(I \cup J)) & , \text{ if } I \bowtie J, \\ \min Q(I, J) & , \text{ if } I \not\bowtie J, |I| \leq |J|, \\ \min R(I, J) & , \text{ if } I \not\bowtie J, |I| > |J|. \end{cases} \quad (3.47)$$

Recall that $Q(I, J)$ and $R(I, J)$ both contain pairs of distinct elements of S . Hence, q satisfies the property (i) above.

By construction, for any $i \neq j$, we have:

$$q^{-1}(i, j) \subseteq A(i, j) \cup B(i, j), \quad (3.48)$$

where

$$A(i, j) := \left\{ (I, J) \in \mathcal{C}(\sigma) \times \mathcal{C}(\tau) \mid I \not\bowtie J, |I| \leq |J|, (i, j) \in Q(I, J) \right\}, \quad (3.49)$$

$$B(i, j) := \left\{ (I, J) \in \mathcal{C}(\sigma) \times \mathcal{C}(\tau) \mid I \not\bowtie J, |I| \geq |J|, (i, j) \in R(I, J) \right\}. \quad (3.50)$$

Remark from (3.45) that if $(I, J) \in A(i, j)$ then $(i \wedge j)_\sigma = I$ and $(i \wedge j)_\tau \supsetneq J$. Hence, if $|(i \wedge j)_\sigma| \geq |(i \wedge j)_\tau|$, then $A(i, j) = \emptyset$. Similarly, $(i \wedge j)_\sigma \supsetneq I$ and $(i \wedge j)_\tau = J$ whenever $(I, J) \in B(i, j)$; and $B(i, j) = \emptyset$ if $|(i \wedge j)_\sigma| \leq |(i \wedge j)_\tau|$. Thus, one can observe that for any $i, j \in S$,

$$A(i, j) \neq \emptyset \implies B(i, j) = \emptyset. \quad (3.51)$$

Recall that for any $i, j \in S$ and $(I, J) \in A(i, j)$, we have:

$$I = (i \wedge j)_\sigma, J \subsetneq (i \wedge j)_\tau, |I| \leq |J| \text{ and } J \in \text{Anc}(\{i\}, \tau). \quad (3.52)$$

Hence, one can conclude that

$$|A(i, j)| \leq \left| |(i \wedge j)_\tau| - |(i \wedge j)_\sigma| \right| = \left| \mathbf{U}(\tau)_{ij} - \mathbf{U}(\sigma)_{ij} \right|. \quad (3.53)$$

Similarly, for any $i, j \in S$,

$$|B(i, j)| \leq \left| |(i \wedge j)_\sigma| - |(i \wedge j)_\tau| \right| = \left| \mathbf{U}(\sigma)_{ij} - \mathbf{U}(\tau)_{ij} \right|. \quad (3.54)$$

Thus, overall, using (3.48) and (3.51), one can obtain the second property of q as follows: for any $i \neq j \in S$,

$$|q_{\sigma, \tau}^{-1}(i, j)| \leq |A(i, j)| + |B(i, j)| \leq \left| \mathbf{U}(\tau)_{ij} - \mathbf{U}(\sigma)_{ij} \right|, \quad (3.55)$$

which completes the proof. ■

3.3 Navigation in the Space of Trees

We now introduce an abstract discrete dynamical system in the NNI graph $\mathcal{N}_S = (\mathcal{BT}_S, \mathcal{E}_{\mathcal{N}_S})$ of binary hierarchies over a fixed finite leaf set S . First, we will introduce a discrete transition policy (the *NNI control law*) for navigating toward a specified target hierarchy $\tau \in \mathcal{BT}_S$ from any arbitrary hierarchy $\sigma \in \mathcal{BT}_S$ with provable termination guarantees. Next, we will present an alternative interpretation of the NNI control law from a hybrid system perspective.

3.3.1 A Discrete-Time Dynamical System Perspective

Recall that in order to define a control policy for navigating in the NNI graph \mathcal{N}_S , we need to construct an input bundle capturing the possible transitions (edges) in \mathcal{N}_S . Let $\widehat{\mathcal{E}}$ denote the set of *directed* edges of the NNI graph with self-loops,

$$\widehat{\mathcal{E}} := \bigcup_{\sigma \in \mathcal{BT}_S} \{\sigma\} \times \widehat{\mathcal{E}}_\sigma, \quad \widehat{\mathcal{E}}_\sigma := \mathcal{G}(\sigma) \cup \{\emptyset\}. \quad (3.56)$$

Thus, every directed edge in $\widehat{\mathcal{E}}$ is referenced by a source tree and an associated grandchild in that tree, defining a map $\text{NNI} : \widehat{\mathcal{E}} \rightarrow \mathcal{BT}_S$ where $\text{NNI}(\sigma, G)$ is the result of applying the NNI move at the grandchild cluster $G \in \widehat{\mathcal{E}}_\sigma$ to the nondegenerate hierarchy $\sigma \in \mathcal{BT}_S$. Note that the NNI move at the empty cluster coincides with the identity mapping of \mathcal{BT}_S (that is, $\sigma = \text{NNI}(\sigma, \emptyset)$ for all $\sigma \in \mathcal{BT}_S$). Thus, the transition structure captured by the map NNI may be thought of as having been obtained from \mathcal{N}_S by simply adding a loop labelled with \emptyset at each vertex.

Accordingly, we will consider discrete-time dynamical systems in \mathcal{BT}_S of the form

$$\sigma^{k+1} = \text{NNI}(\sigma^k, G^k), \quad (3.57a)$$

$$G^k = \mathbf{u}(\sigma^k), \quad (3.57b)$$

where \mathbf{u} is a control policy of $\sigma^k \in \mathcal{BT}_S$ returning a grandchild $G^k \in \widehat{\mathcal{E}}_{\sigma^k}$. Abusing notation, we shall denote the closed-loop dynamical system as

$$\sigma^{k+1} = (\text{NNI} \circ \mathbf{u})(\sigma^k). \quad (3.58)$$

3.3.2 Special Crossings of Clusters

Throughout this section, we shall consider the compatibility of clusters with cluster splits. Hence, it is useful to make the following simple observation:

Lemma 3.5 *Let $\{K_L, K_R\}$ be a bipartition of set K and $I \subsetneq K$. Then, the following equivalence holds*

$$I \bowtie \{K_L, K_R\} \iff (I \subseteq K_L) \vee (I \subseteq K_R). \quad (3.59)$$

Proof It follows from Definition 3.1 that

$$I \bowtie K_L \iff (I \subseteq K_L) \vee (K_L \subseteq I) \vee \underbrace{(I \cap K_L = \emptyset)}_{\iff I \subseteq K_R}, \quad (3.60)$$

$$\iff (I \subseteq K_L) \vee (K_L \subseteq I) \vee (I \subseteq K_R). \quad (3.61)$$

Therefore, the lemma can be verified as follows:

$$I \bowtie \{K_L, K_R\} = (I \bowtie K_L) \wedge (I \bowtie K_R), \quad (3.62)$$

$$= (I \subseteq K_L) \vee (I \subseteq K_R) \vee \underbrace{((K_L \subseteq I) \wedge (K_R \subseteq I))}_{\text{false since } I \subsetneq K = K_L \cup K_R}, \quad (3.63)$$

$$= (I \subseteq K_L) \vee (I \subseteq K_R). \quad (3.64)$$

■

Any pair of binary trees in \mathcal{BT}_S has a common cluster (the cluster S , for example), and one might hope to quantify the discrepancy between a pair of trees by counting common clusters which split differently in the two trees (perhaps, somehow accounting for the depth of these clusters). This motivates:

Definition 3.12 For any $\sigma, \tau \in \mathcal{BT}_S$, let $\mathcal{K}(\sigma, \tau)$ denote the set of common clusters of σ and τ with crossing splits,

$$\mathcal{K}(\sigma, \tau) := \left\{ K \in \mathcal{C}(\sigma) \cap \mathcal{C}(\tau) \mid \text{Ch}(K, \sigma) \neq \text{Ch}(K, \tau) \right\}. \quad (3.65)$$

Remark 3.4. In \mathcal{BT}_S , $\sigma = \tau$ if and only if $\mathcal{K}(\sigma, \tau) = \emptyset$.

Corollary 3.3 For all $\sigma, \tau \in \mathcal{BT}_S$, $\mathcal{K}(\sigma, \tau) = \left\{ K \in \mathcal{C}(\sigma) \cap \mathcal{C}(\tau) \mid \text{Ch}(K, \sigma) \not\bowtie \text{Ch}(K, \tau) \right\}$.

Proof Follows directly from Lemma 3.5 and the definitions. ■

Let $\sigma \neq \tau \in \mathcal{BT}_S$ be two distinct trees which splits $K \in \mathcal{K}(\sigma, \tau)$ into two different pairs of children. Observe now that any cluster $I \in \mathcal{C}(\sigma)$ which is not a σ -descendant of K is automatically compatible with $\text{Ch}(K, \tau)$. Thus, incompatibilities of σ with $\text{Ch}(K, \tau)$ could only occur among σ -descendants of K . This motivates the following definition:

Definition 3.13 (Deep Incompatibility) For $\sigma \neq \tau \in \mathcal{BT}_S$ and $K \in \mathcal{K}(\sigma, \tau)$, let denote the set of clusters of σ incompatible with $\text{Ch}(K, \tau)$ as

$$\mathcal{J}(\sigma, \tau; K) := \left\{ I \in \text{Des}(K, \sigma) \mid I \not\bowtie \text{Ch}(K, \tau) \right\}, \quad (3.66)$$

and denote the subset of deep clusters incompatible with $\text{Ch}(K, \tau)$ as (see Figure 3.5)

$$\mathcal{D}(\sigma, \tau; K) := \left\{ I \in \mathcal{J}(\sigma, \tau; K) \mid \text{Ch}(I, \sigma) \bowtie \text{Ch}(K, \tau), \text{Ch}(I^{-\sigma}, \sigma) \bowtie \text{Ch}(K, \tau) \right\}. \quad (3.67)$$

Note that $\mathcal{J}(\sigma, \tau; K)$ and $\mathcal{D}(\sigma, \tau; K)$ are nonempty since at least an element of $\text{Ch}(K, \sigma)$ is incompatible with $\text{Ch}(K, \tau)$, and vice versa, which is evident from Lemma 3.5.

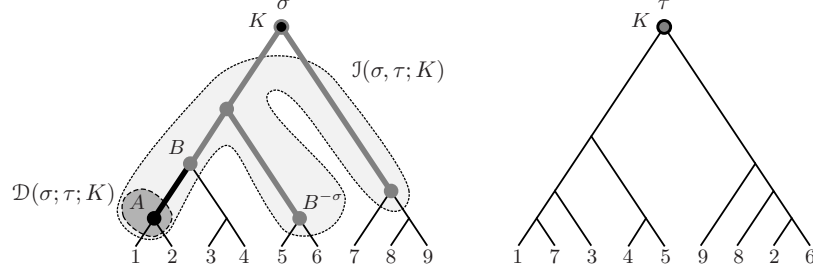


Figure 3.5: An illustration of $\mathcal{J}(\sigma, \tau; K)$ (3.66) and $\mathcal{D}(\sigma, \tau; K)$ (3.67) of $\sigma, \tau \in \mathcal{BT}_{[9]}$, and $K = [9] \in \mathcal{K}(\sigma, \tau)$ (3.65). The vertices and edges associated with clusters of σ incompatible with $\text{Ch}(K, \tau)$ are thickened. The only deep cluster of σ incompatible with $\text{Ch}(K, \tau)$ is $A = \{1, 2\}$, which is also Type 1. B and $B^{-\sigma}$ are examples of Type 2 clusters incompatible with $\text{Ch}(K, \tau)$.

Corollary 3.4 *For any $\sigma \neq \tau \in \mathcal{BT}_S$ and $K \in \mathcal{K}(\sigma, \tau)$, if $I \in \mathcal{J}(\sigma, \tau; K)$, then $\text{Anc}(I, \sigma) \cap \text{Des}(K, \sigma) \subseteq \mathcal{J}(\sigma, \tau; K)$.*

Definition 3.14 (Incompatibility Types) *For any $\sigma \neq \tau \in \mathcal{BT}_S$ and $K \in \mathcal{K}(\sigma, \tau)$, a cluster $I \in \mathcal{J}(\sigma, \tau; K)$ is said to be Type 1 if $I^{-\sigma} \bowtie \text{Ch}(K, \tau)$, and otherwise it is said to be Type 2 (see Figure 3.5).*

Lemma 3.6 *Let $\sigma \neq \tau \in \mathcal{BT}_S$ and $K \in \mathcal{K}(\sigma, \tau)$. Siblings $I, I^{-\sigma} \in \mathcal{J}(\sigma, \tau; K)$ are both Type 2 if and only if they are both incompatible with each child D of K in τ . That is to say, for any $I \in \text{Des}(K, \sigma)$ and $D \in \text{Ch}(K, \tau)$,*

$$I \not\bowtie \text{Ch}(K, \tau), I^{-\sigma} \not\bowtie \text{Ch}(K, \tau) \iff I \not\bowtie D, I^{-\sigma} \not\bowtie D. \quad (3.68)$$

Proof The sufficiency for being Type 2 directly follows from Definition 3.1,

$$I \not\bowtie D, I^{-\sigma} \not\bowtie D \implies I \not\bowtie \text{Ch}(K, \tau), I^{-\sigma} \not\bowtie \text{Ch}(K, \tau). \quad (3.69)$$

To see the necessity for being Type 2, recall from Lemma 3.5 that

$$\begin{aligned} I \bowtie \text{Ch}(K, \tau) &\iff (I \subseteq D) \vee (I \subseteq (K \setminus D)) \iff (I \subseteq D) \vee (I \cap D = \emptyset), \\ I^{-\sigma} \bowtie \text{Ch}(K, \tau) &\iff (I^{-\sigma} \subseteq D) \vee (I^{-\sigma} \cap D = \emptyset). \end{aligned} \quad (3.70)$$

Further, using Lemma 3.5, observe that

$$\begin{aligned} D \subseteq I &\implies I^{-\sigma} \subseteq (K \setminus D) \implies I^{-\sigma} \bowtie \text{Ch}(K, \tau), \\ D \subseteq I^{-\sigma} &\implies I \bowtie \text{Ch}(K, \tau). \end{aligned} \quad (3.71)$$

As a result, using (3.70) and (3.71), one can verify the necessity as

$$I \not\bowtie \text{Ch}(K, \tau), I^{-\sigma} \not\bowtie \text{Ch}(K, \tau) \implies I \not\bowtie D, I^{-\sigma} \not\bowtie D, \quad (3.72)$$

which completes the proof. ■

3.3.3 NNI Control Law

To navigate from an arbitrary tree $\sigma \in \mathcal{BT}_S$ towards any desired tree $\tau \in \mathcal{BT}_S$ in the NNI-graph, we propose an NNI control policy \mathbf{u}_τ that returns an NNI move on σ at a grandchild $G \in \mathcal{G}(\sigma) \cup \{\emptyset\}$, see Figure 3.6, as follows:

1. If $\sigma = \tau$, then just return the identity move, $G = \emptyset$.
2. Otherwise,
 - (a) Select a common cluster $K \in \mathcal{K}(\sigma, \tau)$ (3.65).
 - (b) Find a deep cluster $I \in \mathcal{D}(\sigma, \tau; K)$ (3.67) incompatible with $\text{Ch}(K, \tau)$.
 - (c) Return a proper NNI navigation move on σ at grandchild $G \in \text{Ch}(I, \sigma)$ selected as follows:
 - i. If $I^{-\sigma} \bowtie \text{Ch}(K, \tau)$ (Type 1), then return $G \in \text{Ch}(I, \sigma)$ with the property that $G^{-\sigma}, I^{-\sigma} \subset J$ for some $J \in \text{Ch}(K, \tau)$.
 - ii. Otherwise (Type 2), return an arbitrary NNI move at a child of I in σ .

This NNI control law preserves common clusters of trees. As a result, the navigation problem of trees can be divided into subproblems of disjoint⁹ trees which then may be solved in parallel. This is known as the decomposability property [217].

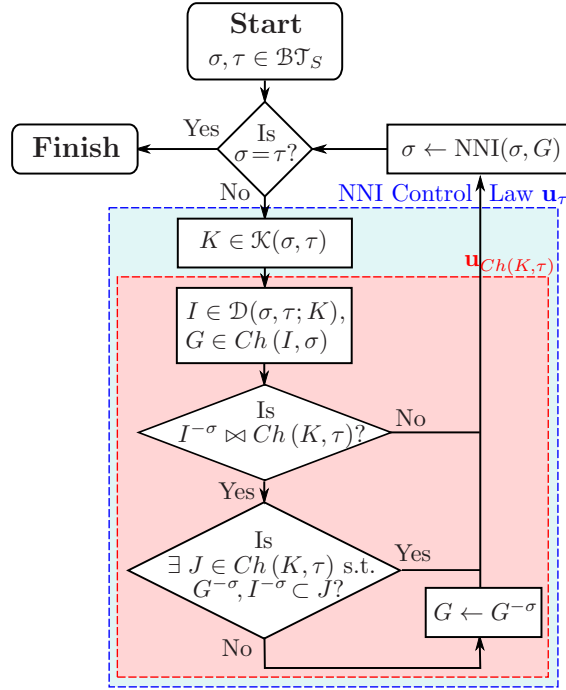


Figure 3.6: A flowchart of navigating from $\sigma \in \mathcal{BT}_S$ towards $\tau \in \mathcal{BT}_S$ based on the NNI control law \mathbf{u}_τ — a nondeterministic hybrid control policy consisting of local controllers $\{\mathbf{u}_{\text{Ch}(J, \tau)}\}_{J \in \mathcal{C}(\tau), |J| > 1}$.

In brief, our NNI control scheme resolves incompatibilities between clusters of σ and τ level by level, depending on the selected common cluster K and one of its deep clusters

⁹Two trees $\sigma, \tau \in \mathcal{T}_S$ are said to be *disjoint* if they have no nontrivial clusters in common [169].

I in Step 2. More precisely, for a fixed $K \in \mathcal{K}(\sigma, \tau)$, the clusters of σ incompatible with $\text{Ch}(K, \tau)$ are replaced by compatible ones using NNI moves associated with deep clusters in $\mathcal{D}(\sigma, \tau; K)$ in a bottom to top fashion. If desired, one can choose the highest common cluster, $K = \arg \min_{J \in \mathcal{K}(\sigma, \tau)} \ell_\sigma(J) + \ell_\tau(J)$ – a top-down strategy, to obtain common splits at higher levels first, yielding higher priority resolution of incompatibilities for clusters closer to the root.

By construction, the NNI control law \mathbf{u}_τ is nondeterministic, and therefore generates multiple choices of paths from any given source σ to the target τ . All such paths will be referred to as *NNI navigation paths*. We leave the making of systematic selections of K and I in Step 2 to a future discussion of specific implementations of the control policy [43, 81, 49]. Here we only mean to focus on properties of this control scheme pertinent to its application as a means for constructing NNI navigation paths and computing their length: since any two NNI navigation paths joining a given pair of vertices turn out (Theorem 3.2) to have equal lengths, the NNI control scheme gives rise to a new dissimilarity, d_{nav} (3.118).

Proposition 3.9 *An output of the NNI control law \mathbf{u}_τ to navigate in the NNI graph $\mathcal{N}_S = (\mathcal{BT}_S, \mathcal{E}_{\mathcal{N}_S})$ towards a desired hierarchy $\tau \in \mathcal{BT}_S$ can be obtained in $O(|S|)$ time.*

Proof Let $\sigma \in \mathcal{BT}_S$. Using the algorithm in [67], the common clusters and equality of σ and τ can be determined in linear time, $O(|S|)$.

If $\sigma = \tau$, then \mathbf{u}_τ returns the identity move. Otherwise, $\mathcal{K}(\sigma, \tau) \neq \emptyset$ and, given the common clusters of σ and τ , a common cluster $K \in \mathcal{K}(\sigma, \tau)$ with crossing splits can be found in $O(|S|)$ time by a traversal of σ . Given $K \in \mathcal{K}(\sigma, \tau)$, as discussed in the proof of Proposition 3.4, the clusters of σ incompatible with $\text{Ch}(K, \tau)$, i.e., $\mathcal{J}(\sigma, \tau; K)$, can be determined in $O(|K|)$ time using Lemma 3.5 and by the post-order traversal of subtree of σ rooted at K . Given $\mathcal{J}(\sigma, \tau; K)$, a deep cluster $I \in \mathcal{D}(\sigma, \tau; K)$ and a proper NNI move on σ at $G \in \text{Ch}(I, \sigma)$ can be found in $O(|K|)$ time by a traversal of the subtree of σ rooted at K .

Thus, the overall cost of computing the NNI control law \mathbf{u}_τ is $O(|S|)$, which completes the proof. \blacksquare

In order to find a path from any given vertex $\sigma \in \mathcal{BT}_S$ to any desired hierarchy $\tau \in \mathcal{BT}_S$ one simply obeys the controller \mathbf{u}_τ . The rest of this section is dedicated to discussing the termination time complexity of this algorithm.

Stability Properties

For a desired nondegenerate hierarchy $\tau \in \mathcal{BT}_S$, a discrete Lyapunov candidate function [127] $V_\tau : \mathcal{BT}_S \rightarrow \mathbb{R}_{\geq 0}$ can be defined using the crossing matrix \mathbf{X} (3.38) as

$$V_\tau(\sigma) := \mathbf{p}_\sigma^T \mathbf{X}(\sigma, \tau) \mathbf{p}_\tau, \quad (3.73)$$

where $\mathbf{p}_\tau := \left(\frac{1}{\rho^{\ell_\tau(I)}} \right)_{I \in \mathcal{C}(\tau)} \in \mathbb{R}_{\geq 0}^{(2|S|-1) \times 1}$ is the *hierarchical attenuation vector* associated with τ . Here, $\rho \geq 1$ is a *hierarchical attenuation constant* and ℓ_τ (3.9) returns the level (depth) of a cluster of τ . Note that since each nondegenerate hierarchy corresponds to a unique set of compatible clusters of maximum cardinality (Remark 3.1), it is clear that $V_\tau(\tau) = 0$ and $V_\tau(\sigma) > 0$ for all $\sigma \in \mathcal{BT}_S \setminus \{\tau\}$. Also, observe that $V_\tau(\sigma)$ is a weighted

version (a continuous one-parameter deformation) of $d_{CM}(\sigma, \tau)$ (3.39), where equality holds for $\rho = 1$.

Lemma 3.7 *For any cluster $K \in \mathcal{C}(\tau)$ of a binary hierarchy $\tau \in \mathcal{BT}_S$, the hierarchical attenuation constant ρ satisfies*

$$\sum_{I \in \text{Des}(K, \tau)} \frac{1}{\rho^{\ell_\tau(I)}} < \frac{2}{\rho - 2} \frac{1}{\rho^{\ell_\tau(K)}} , \quad \forall \rho > 2 . \quad (3.74)$$

Proof Proof by induction.

- (Base Case) If $|K| = 1$, then the result trivially holds since $\text{Des}(K, \tau) = \emptyset$.
- (Induction) Else, suppose the lemma holds for any child cluster in $\{K_L, K_R\} = \text{Ch}(K, \tau)$. Then, using the hierarchical relation between descendants of cluster K and its children, we can factor the left hand side and verify the upper bound as

$$\sum_{I \in \text{Des}(K, \tau)} \frac{1}{\rho^{\ell_\tau(I)}} = \underbrace{\sum_{I \in \text{Ch}(K, \tau)} \frac{1}{\rho^{\ell_\tau(I)}}}_{= \frac{2}{\rho^{\ell_\tau(K)+1}}} + \underbrace{\sum_{I \in \text{Des}(K_L, \tau)} \frac{1}{\rho^{\ell_\tau(I)}}}_{< \frac{2}{\rho-2} \frac{1}{\rho^{\ell_\tau(K_L)}}} + \underbrace{\sum_{I \in \text{Des}(K_R, \tau)} \frac{1}{\rho^{\ell_\tau(I)}}}_{< \frac{2}{\rho-2} \frac{1}{\rho^{\ell_\tau(K_R)}}} , \quad (3.75)$$

$$< \frac{1}{\rho^{\ell_\tau(K)}} \left(\frac{2}{\rho} + \frac{4}{\rho(\rho-2)} \right) = \frac{2}{\rho-2} \frac{1}{\rho^{\ell_\tau(K)}} , \quad (3.76)$$

where the cluster depths satisfy $\ell_\tau(K_L) = \ell_\tau(K_R) = \ell_\tau(K) + 1$. ■

Lemma 3.8 *For any desired $\tau \in \mathcal{BT}_S$ and $\rho > 2$, the change in the value of Lyapunov function V_τ (3.73) after the NNI move on $\sigma \in \mathcal{BT}_S$ at a grandchild $G \in \mathcal{G}(\sigma)$ joining σ to $\gamma \in \mathcal{BT}_S$ is bounded from above as*

$$V_\tau(\gamma) - V_\tau(\sigma) < \frac{1}{\rho^{\ell_\sigma(P) + \ell_\tau(Q) + 2}} \left(\frac{14\rho - 24}{(\rho - 2)^2} + \sum_{\substack{I \in \text{Ch}(P, \gamma) \\ J \in \text{Ch}(Q, \tau)}} \mathbb{1}(I \nparallel J) - \sum_{\substack{I \in \text{Ch}(P, \sigma) \\ J \in \text{Ch}(Q, \tau)}} \mathbb{1}(I \nparallel J) \right) , \quad (3.77)$$

where $P = \text{Pr}^2(G, \sigma) = \text{Pr}(G, \gamma)$ and $Q \in \mathcal{C}(\tau)$ satisfying $P \subseteq Q$.¹⁰

Proof Note that the NNI move on σ at $G \in \mathcal{G}(\sigma)$ joining σ to γ changes the hierarchical organization of common clusters descending the (grand) parent cluster $P = \text{Pr}^2(G, \sigma) = \text{Pr}(G, \gamma)$ of (grand) child G and keeps the remaining clusters unchanged such that $\ell_\sigma(I) = \ell_\gamma(I)$ for all $I \in \mathcal{C}(\sigma) \setminus \text{Des}(P, \sigma) = \mathcal{C}(\gamma) \setminus \text{Des}(P, \gamma)$. Hence, the change in the value of Lyapunov function V_τ after the NNI move from σ to γ can be written as

$$V_\tau(\gamma) - V_\tau(\sigma) = \sum_{\substack{I \in \text{Des}(P, \gamma) \\ J \in \mathcal{C}(\tau)}} \frac{1}{\rho^{\ell_\gamma(I) + \ell_\tau(J)}} \mathbb{1}(I \nparallel J) - \sum_{\substack{I \in \text{Des}(P, \sigma) \\ J \in \mathcal{C}(\tau)}} \frac{1}{\rho^{\ell_\sigma(I) + \ell_\tau(J)}} \mathbb{1}(I \nparallel J) . \quad (3.78)$$

¹⁰Such a cluster $Q \in \mathcal{C}(\tau)$ always exists since $P \subseteq S$ and $S \in \mathcal{C}(\tau)$.

Let $Q \in \mathcal{C}(\tau)$ with $P \subseteq Q$. Then every cluster J in $\mathcal{C}(\tau) \setminus \text{Des}(Q, \tau)$ either contains or is disjoint with any cluster $I \in \text{Des}(P, \sigma) \cup \text{Des}(P, \gamma)$, and so $I \bowtie J$. Therefore, (3.78) can be further simplified as

$$V_\tau(\gamma) - V_\tau(\sigma) = \sum_{\substack{I \in \text{Des}(P, \gamma) \\ J \in \text{Des}(Q, \tau)}} \frac{1}{\rho^{\ell_\gamma(I) + \ell_\tau(J)}} \mathbb{1}(I \not\bowtie J) - \sum_{\substack{I \in \text{Des}(P, \sigma) \\ J \in \text{Des}(Q, \tau)}} \frac{1}{\rho^{\ell_\sigma(I) + \ell_\tau(J)}} \mathbb{1}(I \not\bowtie J) . \quad (3.79)$$

Now let (A, B, C) be the NNI-triplet associated with the pair (σ, γ) (Lemma 3.1). Note that $\{A \cup B\} = \mathcal{C}(\sigma) \setminus \mathcal{C}(\gamma)$ and $\{B \cup C\} = \mathcal{C}(\gamma) \setminus \mathcal{C}(\sigma)$ and $P = A \cup B \cup C$. Using the hierarchical relations between the NNI triplet (A, B, C) and P in trees σ and γ and Lemma 3.7, (3.79) can be factored out and bounded from above for $\rho > 2$ as

$$\begin{aligned} V_\tau(\gamma) - V_\tau(\sigma) &= \underbrace{\sum_{\substack{I \in \text{Des}(A, \gamma) \\ J \in \text{Des}(Q, \tau)}} \frac{1}{\rho^{\ell_\gamma(I) + \ell_\tau(J)}} \mathbb{1}(I \not\bowtie J) - \sum_{\substack{I \in \text{Des}(A, \sigma) \\ J \in \text{Des}(Q, \tau)}} \frac{1}{\rho^{\ell_\sigma(I) + \ell_\tau(J)}} \mathbb{1}(I \not\bowtie J)}_{= \sum_{\substack{I \in \text{Des}(A, \sigma) \\ J \in \text{Des}(Q, \tau)}} \frac{\rho-1}{\rho^{\ell_\sigma(I) + \ell_\tau(J)}} \mathbb{1}(I \not\bowtie J) , \quad \text{since } \begin{matrix} \text{Des}(A, \gamma) = \text{Des}(A, \sigma), \\ \ell_\gamma(I) = \ell_\sigma(I) - 1 \quad \forall I \in \text{Des}(A, \sigma) \end{matrix}} \\ &\quad + \underbrace{\sum_{\substack{I \in \text{Des}(B, \gamma) \\ J \in \text{Des}(Q, \tau)}} \frac{1}{\rho^{\ell_\gamma(I) + \ell_\tau(J)}} \mathbb{1}(I \not\bowtie J) - \sum_{\substack{I \in \text{Des}(B, \sigma) \\ J \in \text{Des}(Q, \tau)}} \frac{1}{\rho^{\ell_\sigma(I) + \ell_\tau(J)}} \mathbb{1}(I \not\bowtie J)}_{= 0 , \quad \text{since } \begin{matrix} \text{Des}(B, \gamma) = \text{Des}(B, \sigma), \\ \ell_\gamma(I) = \ell_\sigma(I) \quad \forall I \in \text{Des}(B, \sigma) \end{matrix}} \\ &\quad + \underbrace{\sum_{\substack{I \in \text{Des}(C, \gamma) \\ J \in \text{Des}(Q, \tau)}} \frac{1}{\rho^{\ell_\gamma(I) + \ell_\tau(J)}} \mathbb{1}(I \not\bowtie J) - \sum_{\substack{I \in \text{Des}(C, \sigma) \\ J \in \text{Des}(Q, \tau)}} \frac{1}{\rho^{\ell_\sigma(I) + \ell_\tau(J)}} \mathbb{1}(I \not\bowtie J)}_{= \sum_{\substack{I \in \text{Des}(C, \sigma) \\ J \in \text{Des}(Q, \tau)}} \frac{1/\rho-1}{\rho^{\ell_\sigma(I) + \ell_\tau(J)}} \mathbb{1}(I \not\bowtie J) \leq 0 , \quad \text{since } \begin{matrix} \text{Des}(C, \gamma) = \text{Des}(C, \sigma), \\ \ell_\gamma(I) = \ell_\sigma(I) + 1 \quad \forall I \in \text{Des}(C, \sigma), \\ \rho \geq 2 \end{matrix}} \\ &\quad + \underbrace{\sum_{\substack{I \in \text{Ch}(B \cup C, \gamma) \\ J \in \text{Des}(Q, \tau)}} \frac{1}{\rho^{\ell_\gamma(I) + \ell_\tau(J)}} \mathbb{1}(I \not\bowtie J) - \sum_{\substack{I \in \text{Ch}(A \cup B, \sigma) \\ J \in \text{Des}(Q, \tau)}} \frac{1}{\rho^{\ell_\sigma(I) + \ell_\tau(J)}} \mathbb{1}(I \not\bowtie J)}_{\leq \sum_{J \in \text{Des}(Q, \tau)} \frac{1}{\rho^{\ell_\sigma(P) + \ell_\tau(J) + 2}} \quad \text{since } \begin{matrix} \ell_\gamma(C) = \ell_\sigma(P) + 2, \\ \ell_\gamma(B) = \ell_\sigma(B) \end{matrix}} \\ &\quad + \sum_{\substack{I \in \text{Ch}(P, \gamma) \\ J \in \text{Des}(Q, \tau)}} \frac{1}{\rho^{\ell_\gamma(I) + \ell_\tau(J)}} \mathbb{1}(I \not\bowtie J) - \sum_{\substack{I \in \text{Ch}(P, \sigma) \\ J \in \text{Des}(Q, \tau)}} \frac{1}{\rho^{\ell_\sigma(I) + \ell_\tau(J)}} \mathbb{1}(I \not\bowtie J) , \quad (3.80) \\ V_\tau(\gamma) - V_\tau(\sigma) &\leq \underbrace{\sum_{\substack{I \in \text{Des}(A, \sigma) \\ J \in \text{Des}(Q, \tau)}} \frac{\rho-1}{\rho^{\ell_\sigma(I) + \ell_\tau(J)}} \mathbb{1}(I \not\bowtie J)}_{< \frac{4(\rho-1)}{(\rho-2)^2} \frac{1}{\rho^{\ell_\sigma(P) + \ell_\tau(Q) + 2}} \quad \text{by Lemma 3.7}} + \underbrace{\sum_{J \in \text{Des}(Q, \tau)} \frac{1}{\rho^{\ell_\sigma(P) + \ell_\tau(J) + 2}}}_{< \frac{2}{\rho-2} \frac{1}{\rho^{\ell_\sigma(P) + \ell_\tau(Q) + 2}} \quad \text{by Lemma 3.7}} \\ &\quad + \sum_{\substack{I \in \text{Ch}(P, \gamma) \\ J \in \text{Des}(Q, \tau)}} \frac{1}{\rho^{\ell_\gamma(I) + \ell_\tau(J)}} \mathbb{1}(I \not\bowtie J) - \sum_{\substack{I \in \text{Ch}(P, \sigma) \\ J \in \text{Des}(Q, \tau)}} \frac{1}{\rho^{\ell_\sigma(I) + \ell_\tau(J)}} \mathbb{1}(I \not\bowtie J) , \quad (3.81) \end{aligned}$$

$$\begin{aligned}
V_\tau(\gamma) - V_\tau(\sigma) &< \frac{6\rho - 8}{(\rho - 2)^2} \frac{1}{\rho^{\ell_\sigma(P) + \ell_\tau(Q) + 2}} + \sum_{\substack{I \in \text{Ch}(P, \gamma) \\ J \in \text{Des}(Q, \tau)}} \frac{1}{\rho^{\ell_\gamma(I) + \ell_\tau(J)}} \mathbb{1}(I \not\bowtie J) \\
&\quad - \sum_{\substack{I \in \text{Ch}(P, \sigma) \\ J \in \text{Des}(Q, \tau)}} \frac{1}{\rho^{\ell_\sigma(I) + \ell_\tau(J)}} \mathbb{1}(I \not\bowtie J). \tag{3.82}
\end{aligned}$$

Finally, using the hierarchical relation between τ -descendants of cluster Q and its children $\{Q_L, Q_R\} = \text{Ch}(Q, \tau)$, we obtain the result for $\rho > 2$ as follows:

$$\begin{aligned}
V_\tau(\gamma) - V_\tau(\sigma) &< \frac{6\rho - 8}{(\rho - 2)^2} \frac{1}{\rho^{\ell_\sigma(P) + \ell_\tau(Q) + 2}} \\
&\quad + \sum_{\substack{I \in \text{Ch}(P, \gamma) \\ J \in \text{Ch}(Q, \tau)}} \frac{1}{\rho^{\ell_\gamma(I) + \ell_\tau(J)}} \mathbb{1}(I \not\bowtie J) - \sum_{\substack{I \in \text{Ch}(P, \sigma) \\ J \in \text{Ch}(Q, \tau)}} \frac{1}{\rho^{\ell_\sigma(I) + \ell_\tau(J)}} \mathbb{1}(I \not\bowtie J) \\
&\quad + \underbrace{\sum_{\substack{I \in \text{Ch}(P, \gamma) \\ J \in \text{Des}(Q_L, \tau)}} \frac{1}{\rho^{\ell_\gamma(I) + \ell_\tau(J)}} \mathbb{1}(I \not\bowtie J) - \sum_{\substack{I \in \text{Ch}(P, \sigma) \\ J \in \text{Des}(Q_L, \tau)}} \frac{1}{\rho^{\ell_\sigma(I) + \ell_\tau(J)}} \mathbb{1}(I \not\bowtie J)}_{< \frac{4}{\rho - 2} \frac{1}{\rho^{\ell_\sigma(P) + \ell_\tau(Q) + 2}} \text{ by Lemma 3.7}} \\
&\quad + \underbrace{\sum_{\substack{I \in \text{Ch}(P, \gamma) \\ J \in \text{Des}(Q_R, \tau)}} \frac{1}{\rho^{\ell_\gamma(I) + \ell_\tau(J)}} \mathbb{1}(I \not\bowtie J) - \sum_{\substack{I \in \text{Ch}(P, \sigma) \\ J \in \text{Des}(Q_R, \tau)}} \frac{1}{\rho^{\ell_\sigma(I) + \ell_\tau(J)}} \mathbb{1}(I \not\bowtie J)}_{< \frac{4}{\rho - 2} \frac{1}{\rho^{\ell_\sigma(P) + \ell_\tau(Q) + 2}} \text{ by Lemma 3.7}}, \tag{3.83}
\end{aligned}$$

$$V_\tau(\gamma) - V_\tau(\sigma) < \frac{1}{\rho^{\ell_\sigma(P) + \ell_\tau(Q) + 2}} \left(\frac{14\rho - 24}{(\rho - 2)^2} + \sum_{\substack{I \in \text{Ch}(P, \gamma) \\ J \in \text{Ch}(Q, \tau)}} \mathbb{1}(I \not\bowtie J) - \sum_{\substack{I \in \text{Ch}(P, \sigma) \\ J \in \text{Ch}(Q, \tau)}} \mathbb{1}(I \not\bowtie J) \right), \tag{3.84}$$

where we have $\ell_\tau(Q_L) = \ell_\tau(Q_R) = \ell_\tau(Q) + 1$. ■

Remark 3.5. Let $\sigma, \tau \in \mathcal{BT}_S$ be two distinct trees, and $K \in \mathcal{K}(\sigma, \tau)$ (3.65) and $I \in \mathcal{D}(\sigma, \tau; K)$ (3.67) be the cluster selected by the NNI control law \mathbf{u}_τ . After a single NNI move,

- If I is Type 1, \mathbf{u}_τ replaces I by a cluster compatible with $\text{Ch}(K, \tau)$ (see Figure 3.7(a)).
- If I is Type 2, \mathbf{u}_τ replaces I by a Type 1 cluster incompatible with $\text{Ch}(K, \tau)$ and its sibling $I^{-\sigma}$, a Type 2 cluster in $\mathcal{D}(\sigma, \tau; K)$, becomes Type 1 incompatible with $\text{Ch}(K, \tau)$ in the next hierarchy (see Figure 3.7(b)).

Meanwhile, the incompatibility types of the rest of clusters in $\mathcal{C}(\sigma) \setminus \{I, I^{-\sigma}\}$ with $\text{Ch}(K, \tau)$ are kept unchanged in the next hierarchy.

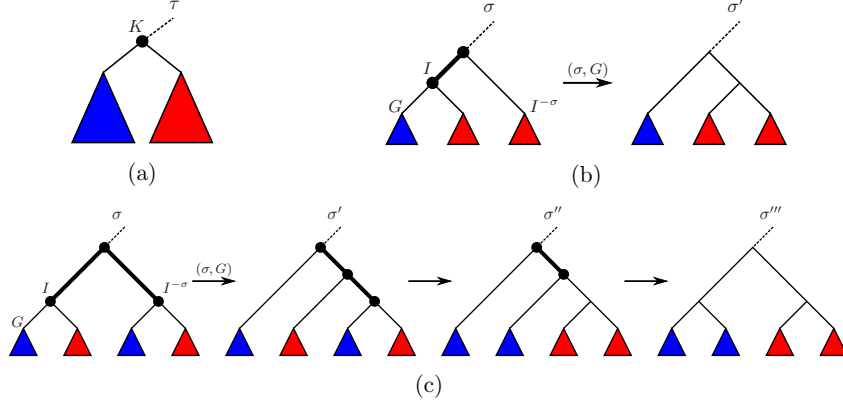


Figure 3.7: An illustration of deep clusters incompatible with $\text{Ch}(K, \tau)$: Type 1 (b) and Type 2 (c) incompatibilities with $\text{Ch}(K, \tau)$ (a) of a common cluster $K \in \mathcal{K}(\sigma, \tau)$, and the associated NNI navigation moves until resolving incompatibilities with $\text{Ch}(K, \tau)$. Clusters are colored according to their inclusion relation, and the thickened vertices show a portion of incompatible clusters in $\mathcal{J}(\sigma, \tau; K)$.

Theorem 3.1 *The NNI control law \mathbf{u}_τ defines an abstract closed-loop discrete dynamical system (3.58) in the NNI graph $\mathcal{N}_S = (\mathcal{BT}_S, \mathcal{E}_{\mathcal{N}_S})$. There exists $\epsilon > 0$ such that for every $\rho \geq 9 + \sqrt{53}$ and for any $\sigma \in \mathcal{BT}_S \setminus \{\tau\}$, one has*

$$V_\tau((\text{NNI} \circ \mathbf{u}_\tau)(\sigma)) - V_\tau(\sigma) < -\epsilon < 0. \quad (3.85)$$

Hence all paths generated by the control policy \mathbf{u}_τ terminate at the goal, τ .

Proof Let $K \in \mathcal{K}(\sigma, \tau)$ (3.65) and $I \in \mathcal{D}(\sigma, \tau; K)$ (3.67) be the clusters selected by the NNI control policy \mathbf{u}_τ while determining the NNI move on σ at $G \in \text{Ch}(I, \sigma)$ towards $\gamma = (\text{NNI} \circ \mathbf{u}_\tau)(\sigma)$, and let $P = \text{Pr}(I, \sigma) = \text{Pr}(G, \gamma) \subseteq K$.

It follows from Lemma 3.8 that for any $\rho \geq 9 + \sqrt{53}$, the change of Lyapunov function V_τ after the aforementioned NNI move is bounded from above as

$$V_\tau(\gamma) - V_\tau(\sigma) < \frac{1}{\rho^{\ell_\sigma(P) + \ell_\tau(K) + 2}} \left(1 + \sum_{\substack{E \in \text{Ch}(P, \gamma) \\ F \in \text{Ch}(K, \tau)}} \mathbb{1}(E \not\bowtie F) - \sum_{\substack{E \in \text{Ch}(P, \sigma) \\ F \in \text{Ch}(K, \tau)}} \mathbb{1}(E \not\bowtie F) \right). \quad (3.86)$$

To complete the proof of the theorem, we shall show that

$$\sum_{\substack{E \in \text{Ch}(P, \gamma) \\ F \in \text{Ch}(K, \tau)}} \mathbb{1}(E \not\bowtie F) - \sum_{\substack{E \in \text{Ch}(P, \sigma) \\ F \in \text{Ch}(K, \tau)}} \mathbb{1}(E \not\bowtie F) \leq -1. \quad (3.87)$$

Depending on the incompatibility of I with $\text{Ch}(K, \tau)$, one has:

- If $I \in \mathcal{D}(\sigma, \tau; K)$ is Type 1 (Definition 3.14), then $I \not\bowtie \text{Ch}(K, \tau)$ and $I^{-\sigma} \bowtie \text{Ch}(K, \tau)$. Recall that $\{I, I^{-\sigma}\} = \text{Ch}(P, \sigma)$. Hence,

$$\sum_{F \in \text{Ch}(K, \tau)} \underbrace{\sum_{E \in \text{Ch}(P, \sigma)} \mathbb{1}(E \not\bowtie F)}_{=\mathbb{1}(I \not\bowtie F), \text{ since } I^{-\sigma} \bowtie \text{Ch}(K, \tau)} = \sum_{F \in \text{Ch}(K, \tau)} \underbrace{\mathbb{1}(I \not\bowtie F)}_{\geq 1, \text{ since } I \not\bowtie \text{Ch}(K, \tau)} \geq 1. \quad (3.88)$$

Further, the NNI control rule \mathbf{u}_τ replaces I by $J = \text{Pr}^2(G, \sigma) \setminus G = P \setminus G$ whose local complement $J^{-\gamma}$ in γ is G , and so $\{J, J^{-\gamma}\} = \text{Ch}(P, \gamma)$. Recall from Remark 3.5 that J and $J^{-\gamma}$ are both compatible with $\text{Ch}(K, \tau)$. Therefore,

$$\sum_{\substack{E \in \text{Ch}(P, \gamma) \\ F \in \text{Ch}(K, \tau)}} \underbrace{\mathbb{1}(E \not\bowtie F)}_{=0, \text{ since } E \bowtie \text{Ch}(K, \tau)} = 0. \quad (3.89)$$

As a result, (3.87) always holds for any Type 1 cluster.

- Otherwise (I is Type 2), by Definition 3.14, siblings $I, I^{-\sigma} \in \mathcal{D}(\sigma, \tau; K)$ are both incompatible with $\text{Ch}(K, \tau)$. In fact, by Lemma 3.6, for all $E \in \{I, I^{-\sigma}\} = \text{Ch}(P, \sigma)$ and $F \in \text{Ch}(K, \tau)$, we have $E \not\bowtie F$. Thus,

$$\sum_{\substack{E \in \text{Ch}(P, \sigma) \\ F \in \text{Ch}(K, \tau)}} \underbrace{\mathbb{1}(E \not\bowtie F)}_{=1, \text{ by Lemma 3.6}} = 4. \quad (3.90)$$

On the other hand, any arbitrary NNI move $G \in \text{Ch}(I, \sigma)$ replaces cluster I by $J = \text{Pr}^2(G, \sigma) \setminus G$ incompatible with $\text{Ch}(K, \tau)$. Note that its sibling $J^{-\gamma}$ in γ is G and compatible with split $\text{Ch}(K, \tau)$. Hence, we have $J \not\bowtie \text{Ch}(K, \tau)$ and $J^{-\gamma} \bowtie \text{Ch}(K, \tau)$ for children clusters $\{J, J^{-\gamma}\} = \text{Ch}(P, \gamma)$, which yields

$$\sum_{F \in \text{Ch}(K, \tau)} \underbrace{\sum_{E \in \text{Ch}(P, \gamma)} \mathbb{1}(E \not\bowtie F)}_{=\mathbb{1}(J \not\bowtie F), \text{ since } J^{-\gamma} \bowtie \text{Ch}(K, \tau)} = \sum_{F \in \text{Ch}(K, \tau)} \mathbb{1}(J \not\bowtie F) \leq 2. \quad (3.91)$$

Therefore, for any Type 2 cluster, we always have

$$\sum_{\substack{E \in \text{Ch}(P, \gamma) \\ F \in \text{Ch}(K, \tau)}} \mathbb{1}(E \not\bowtie F) - \sum_{\substack{E \in \text{Ch}(P, \sigma) \\ F \in \text{Ch}(K, \tau)}} \mathbb{1}(E \not\bowtie F) \leq -2. \quad (3.92)$$

To sum up, the NNI control policy \mathbf{u}_τ always guarantees that (3.87) holds after each evolution of the dynamical system (3.57) at every σ away from τ , which completes the proof. \blacksquare

Tree Metrics and the NNI Control Law

The NNI control law is compatible with d_{RF} (3.17) and d_{CC} (3.23) in the sense that:

Proposition 3.10 *The Robinson-Foulds, d_{RF} (3.17), and the cluster cardinality, d_{CC} (3.23), distances to any desired hierarchy $\tau \in \mathcal{BT}_S$ are nonincreasing at each evolution of the closed loop discrete dynamical system (3.58) obeying the NNI control law \mathbf{u}_τ , i.e., for any $d \in \{d_{RF}, d_{CC}\}$ and $\sigma \in \mathcal{BT}_S$*

$$d(\text{NNI} \circ \mathbf{u}_\tau(\sigma), \tau) - d(\sigma, \tau) \leq 0. \quad (3.93)$$

Proof The result for d_{RF} is evident from that the NNI control law \mathbf{u}_τ preserves the common clusters of the current and goal hierarchies.

For d_{CC} , the statement holds trivially whenever $\sigma = \tau$. If $\sigma \neq \tau$, then let $K \in \mathcal{K}(\sigma, \tau)$ with $\{K_L, K_R\} = \text{Ch}(K, \tau)$ and $I \in \mathcal{D}(\sigma, \tau; K)$ be the selected clusters by the NNI control law \mathbf{u}_τ while determining the NNI move on σ at $G \in \text{Ch}(I, \sigma)$ yielding $\gamma = (\text{NNI} \circ \mathbf{u}_\tau)(\sigma)$. Note that this restructuring of σ only changes relative relations of G , $G^{-\sigma}$ and $I^{-\sigma}$ below $P = \text{Pr}^2(G, \sigma) \subseteq K$ such that $\{G, G^{-\sigma}\} = \text{Ch}(I, \sigma)$, $\{I, I^{-\sigma}\} = \text{Ch}(P, \sigma)$ and $\{G^{-\sigma} \cup I^{-\sigma}, G\} = \text{Ch}(P, \gamma)$. Further, by Definition 3.14, $G \subseteq K_A$ and $G^{-\sigma} \subseteq K_B$ for some $A \neq B \in \{L, R\}$, and so $(i \wedge j)_\tau = K$ for any $i \in G$ and $j \in G^{-\sigma}$. Accordingly, the change in d_{CC} with respect to τ after the transition from σ to γ can be written as ¹¹

$$d_{CC}(\gamma, \tau) - d_{CC}(\sigma, \tau) = \frac{1}{2} \|\mathbf{U}(\gamma) - \mathbf{U}(\tau)\|_1 - \frac{1}{2} \|\mathbf{U}(\sigma) - \mathbf{U}(\tau)\|_1, \quad (3.94)$$

$$\begin{aligned} &= \sum_{\substack{i \in G \\ j \in G^{-\sigma}}} |\mathbf{U}(\gamma)_{ij} - \mathbf{U}(\tau)_{ij}| - |\mathbf{U}(\sigma)_{ij} - \mathbf{U}(\tau)_{ij}| \\ &\quad + \sum_{\substack{i \in G \\ j \in I^{-\sigma}}} |\mathbf{U}(\gamma)_{ij} - \mathbf{U}(\tau)_{ij}| - |\mathbf{U}(\sigma)_{ij} - \mathbf{U}(\tau)_{ij}| \\ &\quad + \sum_{\substack{i \in G^{-\sigma} \\ j \in I^{-\sigma}}} |\mathbf{U}(\gamma)_{ij} - \mathbf{U}(\tau)_{ij}| - |\mathbf{U}(\sigma)_{ij} - \mathbf{U}(\tau)_{ij}|, \quad (3.95) \end{aligned}$$

$$\begin{aligned} &= \sum_{\substack{i \in G \\ j \in G^{-\sigma}}} \underbrace{|h(P) - h(K)| - |h(I) - h(K)|}_{=-h(P)+h(I)=-|I^{-\sigma}|} \\ &\quad + \sum_{\substack{i \in G \\ j \in I^{-\sigma}}} |h(P) - \mathbf{U}(\tau)_{ij}| - |h(P) - \mathbf{U}(\tau)_{ij}| \\ &\quad + \sum_{\substack{i \in G^{-\sigma} \\ j \in I^{-\sigma}}} \underbrace{|h(G^{-\sigma} \cup I^{-\sigma}) - \mathbf{U}(\tau)_{ij}| - |h(P) - \mathbf{U}(\tau)_{ij}|}_{\in [h(G^{-\sigma} \cup I^{-\sigma}) - h(P), h(P) - h(G^{-\sigma} \cup I^{-\sigma})] = [-|G|, |G|]}, \quad (3.96) \end{aligned}$$

$$\leq -|G| |G^{-\sigma}| |I^{-\sigma}| + |G| |G^{-\sigma}| |I^{-\sigma}| = 0. \quad (3.97)$$

Note that the equality in (3.93) can hold for both d_{RF} and d_{CC} as depicted in Figure 3.8. ■

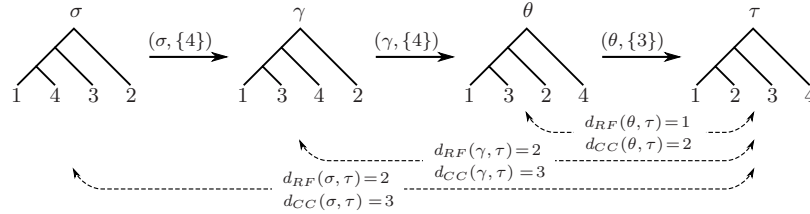


Figure 3.8: An NNI navigation path joining σ to τ and associated NNI navigation moves. The NNI move from σ to $\gamma = (\text{NNI} \circ \mathbf{u}_\tau)(\sigma)$ illustrates that d_{CC} and d_{RF} to the desired hierarchy τ might stay the same after an NNI navigation transition.

¹¹Here, one can easily verify that $d_{CC}(\gamma, \tau) - d_{CC}(\sigma, \tau) < 0$ if I is Type 2 incompatible with split $\text{Ch}(K, \tau)$.

3.3.4 Resolving Incompatibilities with the Root Split

For any bipartition $\{S_L, S_R\}$ of the leaf set S , define $\mathcal{BT}_{\{S_L, S_R\}}$ to be the subset of \mathcal{BT}_S containing nondegenerate hierarchies with the root split of $\{S_L, S_R\}$,

$$\mathcal{BT}_{\{S_L, S_R\}} := \left\{ \tau \in \mathcal{BT}_S \mid \text{Ch}(S, \tau) = \{S_L, S_R\} \right\} . \quad (3.98)$$

Now, instead of the original problem of navigating from any $\sigma \in \mathcal{BT}_S$ towards a single desired hierarchy $\tau \in \mathcal{BT}_S$ in the NNI-graph, consider a related and simpler problem of navigating hierarchies towards the set $\mathcal{BT}_{\{S_L, S_R\}}$ (3.98) containing binary trees with the root split of $\{S_L, S_R\}$. Observe that the NNI control law in Section 3.3.3 can be used to solve this new problem by selecting any desired hierarchy $\tau \in \mathcal{BT}_{\{S_L, S_R\}}$ and fixing the common cluster $K \in \mathcal{K}(\sigma, \tau)$, in Step 2a of the NNI control policy, as $K = S$. We denote this version of the NNI control policy by $\mathbf{u}_{\{S_L, S_R\}}$.

In general, for any bipartition $\{M_L, M_R\}$ of subset $M \subseteq S$, let $\mathbf{u}_{\{M_L, M_R\}}$ be the local controller, whose domain is

$$\mathcal{BT}_S(M) := \left\{ \sigma \in \mathcal{BT}_S \mid M \in \mathcal{C}(\sigma) \right\} , \quad (3.99)$$

that coincides with the NNI control law \mathbf{u}_τ for any $\tau \in \mathcal{BT}_S(M)$ if for every $\sigma \in \mathcal{BT}_S(M)$ the common cluster $K \in \mathcal{K}(\sigma, \tau)$, selected by the NNI control policy \mathbf{u}_τ in Step 2a, is fixed as $K = M$. One can use $\mathbf{u}_{\{M_L, M_R\}}$ to replace children of M in $\sigma \in \mathcal{BT}_S(M)$ with the desired split $\{M_L, M_R\}$ by resolving cluster incompatibilities of σ with $\{M_L, M_R\}$. Here, observe that the closed-loop dynamical system (3.57) obeying the control law $\mathbf{u}_{\{M_L, M_R\}}$ guarantees the positive invariance of $\mathcal{BT}_S(M)$ and terminates at

$$\mathcal{BT}_{\{M_L, M_R\}} := \left\{ \sigma \in \mathcal{BT}_S(M) \mid \text{Ch}(M, \sigma) = \{M_L, M_R\} \right\} . \quad (3.100)$$

In fact, from a hybrid systems perspective, for any desired hierarchy $\tau \in \mathcal{BT}_S$, the NNI control law \mathbf{u}_τ consists of local controllers, $\mathbf{u}_{\text{Ch}(K, \tau)}$'s, associated with nonsingleton clusters, $K \in \mathcal{C}(\tau)$. For any hierarchy $\sigma \in \mathcal{BT}_S \setminus \{\tau\}$, by selecting $K \in \mathcal{K}(\sigma, \tau)$, the NNI control law \mathbf{u}_τ arbitrarily excites one, $\mathbf{u}_{\text{Ch}(K, \tau)}$, of the local controllers, $\left\{ \mathbf{u}_{\text{Ch}(J, \tau)} \right\}_{\substack{J \in \mathcal{C}(\tau) \\ |J| > 1}}$, whose domain contains σ , illustrated in Figure 3.6.

For the sake of clarity, we restrict the discussion of resolution of incompatibilities to the special case when the desired split is the root split $\{S_L, S_R\}$, but the results generalize to any aforementioned local controller as well.

A critical observation regarding the location of endpoints of NNI navigation paths consistent with the control law $\mathbf{u}_{\{S_L, S_R\}}$ is:

Lemma 3.9 *Let $\sigma \in \mathcal{BT}_S$ and $\{S_L, S_R\}$ be a bipartition of S . The closed loop dynamical system (3.58) obeying the NNI control rule $\mathbf{u}_{\{S_L, S_R\}}$ terminates at the nondegenerate hierarchy $\gamma \in \mathcal{BT}_{\{S_L, S_R\}}$ with cluster set*

$$\mathcal{C}(\gamma) = \mathcal{C}(\sigma|_{S_L}) \cup \{S\} \cup \mathcal{C}(\sigma|_{S_R}) , \quad (3.101)$$

where $\sigma|_I$ (3.14) denotes the subtree of σ rooted at cluster $I \in S$.

Proof If $\text{Ch}(S, \sigma) = \{S_L, S_R\}$, then the results directly holds with $\gamma = \sigma$. Otherwise, let $\{\sigma_0 = \sigma, \sigma_1, \sigma_2, \dots, \sigma_k\}$ be an NNI navigation path, consistent with the NNI control law $\mathbf{u}_{\{S_L, S_R\}}$, starting from σ and ending at $\sigma_k \in \mathcal{BT}_{\{S_L, S_R\}}$ such that $\sigma_i \notin \mathcal{BT}_{\{S_L, S_R\}}$ for all $0 \leq i \leq k-1$. We will show that for all $0 \leq i \leq k-1$ and $J \in \{S_L, S_R\}$

$$\mathcal{C}(\sigma_i|_J) = \mathcal{C}(\sigma_{i+1}|_J). \quad (3.102)$$

Hence, since $\sigma_k \in \mathcal{BT}_{\{S_L, S_R\}}$, by Remark 3.2 and (3.102), we have

$$\mathcal{C}(\sigma_k) = \mathcal{C}(\sigma_k|_{S_L}) \cup \{S\} \cup \mathcal{C}(\sigma_k|_{S_R}) = \mathcal{C}(\sigma|_{S_L}) \cup \{S\} \cup \mathcal{C}(\sigma|_{S_R}). \quad (3.103)$$

Moreover, termination directly follows from Theorem 3.1, and so the result follows.

Now, to prove (3.102), for any $0 \leq i \leq k-1$, let $I_i \in \mathcal{C}(\sigma_i)$ and $G_i \in \text{Ch}(I_i, \sigma_i)$ be, respectively, the deep cluster incompatible with $\{S_L, S_R\}$ and the grandchildren, selected by $\mathbf{u}_{\{S_L, S_R\}}$, determining the NNI move from σ_i to $\sigma_{i+1} = \text{NNI}(\sigma_i, G_i)$. Note that, by Definition 3.4, $I_i \in \mathcal{C}(\sigma_i)$ is replaced by $G_i^{-\sigma_i} \cup I_i^{-\sigma_i} \in \sigma_{i+1}$, and so $\{I_i\} = \mathcal{C}(\sigma_i) \setminus \mathcal{C}(\sigma_{i+1})$ and $\{G_i^{-\sigma_i} \cup I_i^{-\sigma_i}\} = \mathcal{C}(\sigma_{i+1}) \setminus \mathcal{C}(\sigma_i)$. Without loss of generality, let $G_i \subset S_L$, and observe that $I_i \cap S_L = G_i$ and $I_i \cap S_R = G_i^{-\sigma_i}$ are common clusters of σ_i and σ_{i+1} . Depending on the type of incompatibility of I_i in σ_i with $\{S_L, S_R\}$:

- If I_i is Type 1 incompatible with $\{S_L, S_R\}$ (Definition 3.14), then observe from Figure 3.7(a) that $(G_i^{-\sigma_i} \cup I_i^{-\sigma_i}) \cap S_L = \emptyset$ and $(G_i^{-\sigma_i} \cup I_i^{-\sigma_i}) \cap S_R = G_i^{-\sigma_i} \cup I_i^{-\sigma_i} = \text{Pr}(I_i, \sigma_i) \cap S_R$ from which (3.102) follows.
- Otherwise (I_i is Type 2 incompatible with $\{S_L, S_R\}$), observe from Figure 3.7(b) that $(G_i^{-\sigma_i} \cup I_i^{-\sigma_i}) \cap S_L = I_i^{-\sigma_i} \cap S_L$ and $(G_i^{-\sigma_i} \cup I_i^{-\sigma_i}) \cap S_R = \text{Pr}(I_i, \sigma_i) \cap S_R$. Since $I_i^{-\sigma_i}$ and $\text{Pr}(I_i, \sigma_i)$ are common clusters of σ_i and σ_{i+1} , (3.102) holds.

This completes the proof. ■

3.4 The NNI Navigation Dissimilarity

In Section 3.3, to navigate from any initial hierarchy in \mathcal{BT}_S towards a selected desired hierarchy $\tau \in \mathcal{BT}_S$ in the NNI graph $\mathcal{N}_S = (\mathcal{BT}_S, \mathcal{E}_{\mathcal{N}_S})$, we design an abstract closed-loop discrete dynamical system obeying a nondeterministic control rule, named the *NNI control law* \mathbf{u}_τ . In this section, we will introduce a new dissimilarity between a pair of trees counting the steps along a trajectory of this nondeterministic dynamical system joining these trees.

Definition 3.15 *An NNI navigation path from $\sigma \in \mathcal{BT}_S$ to $\tau \in \mathcal{BT}_S$ is a path in the NNI graph \mathcal{N}_S that is consistent with the (nondeterministic) closed-loop dynamical system (3.58) subject to the NNI control law \mathbf{u}_τ of Section 3.3.3.*

The NNI navigation dissimilarity $d_{nav}(\sigma, \tau)$ between the trees σ, τ is the length of an NNI navigation path joining them.

Here we prove (Theorem 3.2) that all NNI navigation paths joining a pair of trees have the same length, which guarantees that d_{nav} is a well defined dissimilarity on \mathcal{BT}_S . Furthermore, we will provide an explicit expression for the NNI navigation dissimilarity, d_{nav} , and show that it can be computed in $O(|S|^2)$ time (Proposition 3.13).

We now introduce a version of the crossing matrix \mathbf{X} (3.38) encoding weighted special cluster crossings between trees, which will be shown to have a significant connection with d_{nav} later in Theorem 3.2.

Definition 3.16 *The special crossing matrix $\mathbf{S}(\sigma, \tau)$ of a pair of hierarchies $\sigma, \tau \in \mathcal{T}_S$ is defined to be*

$$\mathbf{S}(\sigma, \tau)_{I,J} := (\eta \circ \kappa) (\text{Ch}(I, \sigma)|_J, \text{Ch}(J, \tau)|_I) , \quad (3.104)$$

where $I \in \mathcal{C}(\sigma)$, $J \in \mathcal{C}(\tau)$, and $(.)|_J$ denotes restriction to J (see Definition 3.6), and

$$\kappa(\mathcal{A}, \mathcal{B}) := \sum_{A \in \mathcal{A}} \mathbb{1}(A \not\bowtie \mathcal{B}) , \quad \eta(x) := \frac{1}{2}(x^2 + x) . \quad (3.105)$$

Remark 3.6. While $\kappa(\mathcal{A}, \mathcal{B})$ returns the number of elements of $\mathcal{A} \subseteq \mathcal{P}(S)$ incompatible with $\mathcal{B} \subseteq \mathcal{P}(S)$,¹² the quantity $\eta(x)$ heuristically encodes the required number of NNI moves to resolve a certain number of cluster crossings.

Note that $\mathbf{X}(\sigma, \tau)_{I,J} = (\eta \circ \kappa)(\{I\}, \{J\})$ for all $I \in \mathcal{C}(\sigma)$ and $J \in \mathcal{C}(\tau)$ (3.38). Recall that the crossing matrix $\mathbf{X}(\sigma, \tau)$ (3.38) encodes the pairwise cluster incompatibilities of trees. Similarly, $\mathbf{S}(\sigma, \tau)_{I,J}$ measures how consistent the grouping of common elements of $I \in \mathcal{C}(\sigma)$ and $J \in \mathcal{C}(\tau)$ in σ and τ is. Additionally, note that for any bipartitions \mathcal{A} and \mathcal{B} of S , $\kappa(\mathcal{A}, \mathcal{B}) = \kappa(\mathcal{B}, \mathcal{A})$ (by Lemma 3.10 below), and so for binary hierarchies $\sigma, \tau \in \mathcal{BT}_S$ we have $\mathbf{S}(\sigma, \tau) = \mathbf{S}(\tau, \sigma)^T$.¹³

Lemma 3.10 *Let $\{S_L, S_R\}$ and $\{S_L^*, S_R^*\}$ be two bipartitions of a fixed finite set S . The sum of compatible elements of one bipartition with the other bipartition is symmetric,*

$$\sum_{I \in \{S_L, S_R\}} \mathbb{1}(I \bowtie \{S_L^*, S_R^*\}) = \sum_{I^* \in \{S_L^*, S_R^*\}} \mathbb{1}(I^* \bowtie \{S_L, S_R\}) , \quad (3.106)$$

and is only two when the bipartitions are the same.

Proof If the bipartitions are the same, both sides of (3.106) simply sum to two.

Otherwise, since $\{S_L, S_R\}$ and $\{S_L^*, S_R^*\}$ are distinct binary partitions of S , at most an element of $\{S_L, S_R\}$ is a proper subset of an element of $\{S_L^*, S_R^*\}$ and vice versa. One way to observe this is a proof by contradiction. Let each element of $\{S_L, S_R\}$ is a proper subset of an element of $\{S_L^*, S_R^*\}$, then $S_L \subsetneq S_L^* \cup S_R^*$ and $S_R \subsetneq S_L^* \cup S_R^*$. Hence, we have $S_L \cup S_R \subsetneq S_L^* \cup S_R^* = S$, which is a contradiction.

Now, if an element of one bipartition is a proper subset of an element of the other bipartition, i.e., without loss of generality, $S_L \subsetneq S_L^*$ (and so $S_R^* \subsetneq S_R$, $S_R \not\subseteq S_L^*$ and $S_L^* \not\subseteq S_R$), then using Lemma 3.5 one can verify that both sides of (3.106) sum to one. Otherwise, the summations on both sides of (3.106) are equal to zero, because every pair of elements of the bipartitions are not subset of each other and so are incompatible. Thus, the result follows. \blacksquare

¹²Here, for any $\mathcal{A} \subseteq \mathcal{P}(S)$ we set $\kappa(\mathcal{A}, \emptyset) = \kappa(\emptyset, \mathcal{A}) = 0$.

¹³ \mathbf{A}^T denotes the transpose of a matrix \mathbf{A} .

Corollary 3.5 Let $\sigma, \tau \in \mathcal{BT}_S$. Then for any $I \in \mathcal{C}(\sigma)$ and $J \in \mathcal{C}(\tau)$, the following symmetry holds,

$$\kappa(\text{Ch}(I, \sigma)|_J, \text{Ch}(J, \tau)|_I) = \kappa(\text{Ch}(J, \tau)|_I, \text{Ch}(I, \sigma)|_J) , \quad (3.107)$$

where κ is defined as in (3.105).

Proof Consider a special case where $\text{Ch}(I, \sigma)|_J = \{I \cap J, \emptyset\}$ or $\text{Ch}(J, \tau)|_I = \{J \cap I, \emptyset\}$. Then, since the empty cluster and $I \cap J$ are always compatible with any subset of $I \cap J$, both sides of (3.107) are zero.

Otherwise, observe that $\text{Ch}(I, \sigma)|_J$ and $\text{Ch}(J, \tau)|_I$ are bipartitions of $I \cap J$, and so (3.107) takes the specific form in Lemma 3.10, which completes the proof. ■

Lemma 3.11 Let $\{S_L, S_R\}$ be a bipartition of set S , and I and J be sets with the property that $I \subseteq S$ and $I \subseteq J$. Then the following equivalence holds

$$I \bowtie \{S_L, S_R\} = I \bowtie \{S_L, S_R\}|_J , \quad (3.108)$$

where $\{S_L, S_R\}|_J$ is the restriction of $\{S_L, S_R\}$ to J (Definition 3.6).

Proof If $I = S$, then $S \subseteq J$ and $\{S_L, S_R\}|_J = \{S_L, S_R\}$, and so the result follows.

Otherwise ($I \subsetneq S$), observe that for any sets $X \subseteq Z$ and Y , one always has $X \subseteq Y \iff X \subseteq Y \cap Z$. Accordingly, one can verify the result using Lemma 3.5 as follows:

$$I \bowtie \{S_L, S_R\} \iff \underbrace{(I \subseteq S_L)}_{\iff I \subseteq S_L \cap J} \vee \underbrace{(I \subseteq S_R)}_{\iff I \subseteq S_R \cap J} \iff I \bowtie \{S_L \cap J, S_R \cap J\} , \quad (3.109)$$

$$\iff I \bowtie \underbrace{(\{S_L \cap J, S_R \cap J\} \setminus \{\emptyset\})}_{=\{S_L, S_R\}|_J} \iff I \bowtie \{S_L, S_R\}|_J . \quad (3.110)$$

■

An NNI navigation path joining $\sigma, \tau \in \mathcal{BT}_S$ might not be unique since the NNI control law in Section 3.3.3 is nondeterministic. However:

Lemma 3.12 For any bipartition $\{S_L, S_R\}$ of S , the lengths of all NNI navigation paths from $\sigma \in \mathcal{BT}_S$ to $\mathcal{BT}_{\{S_L, S_R\}}$ (3.98) are the same, and the NNI navigation dissimilarity $d_{nav}(\sigma, \mathcal{BT}_{\{S_L, S_R\}})$ is given by

$$d_{nav}(\sigma, \mathcal{BT}_{\{S_L, S_R\}}) = \sum_{I \in \mathcal{C}(\sigma)} (\eta \circ \kappa) \left(\text{Ch}(I, \sigma), \{S_L, S_R\} \right) , \quad (3.111)$$

where κ and η are, respectively, defined as in (3.105).

Proof Proof by induction.

(Base Case) If $\sigma \in \mathcal{BT}_{\{S_L, S_R\}}$, then the result directly follows.

(Induction) Otherwise, let $\{\sigma_0 = \sigma, \sigma_1, \sigma_2, \dots, \sigma_k\}$ be an NNI navigation path, consistent with the NNI control law $\mathbf{u}_{\{S_L, S_R\}}$, starting from σ and ending at $\sigma_k \in \mathcal{BT}_{\{S_L, S_R\}}$ such that for all $0 \leq i < k$ $\sigma_i \notin \mathcal{BT}_{\{S_L, S_R\}}$.

Observe that $(\sigma_1, \sigma_2, \dots, \sigma_k)$ is also an NNI navigation path, consistent with the NNI control law $\mathbf{u}_{\{S_L, S_R\}}$, starting from σ_1 and ending in $\mathcal{BT}_{\{S_L, S_R\}}$.

For any $\gamma \in \mathcal{BT}_S$ let $\alpha(\gamma)$ and $\beta(\gamma)$, respectively, denote the numbers of Type 1 and Type 2 clusters of γ incompatible with $\{S_L, S_R\}$ (Definition 3.14).

Now suppose (3.111) holds for the path $(\sigma_1, \sigma_2, \dots, \sigma_k)$, and so one can rewrite $d_{nav}(\sigma_1, \mathcal{BT}_{\{S_L, S_R\}})$ as

$$d_{nav}(\sigma_1, \mathcal{BT}_{\{S_L, S_R\}}) = \alpha(\sigma_1) + \frac{3}{2}\beta(\sigma_1) . \quad (3.112)$$

As a result,

$$d_{nav}(\sigma, \mathcal{BT}_{\{S_L, S_R\}}) = \alpha(\sigma_1) + \frac{3}{2}\beta(\sigma_1) + 1 . \quad (3.113)$$

Let $I \in \mathcal{C}(\sigma)$ be the selected cluster by $\mathbf{u}_{\{S_L, S_R\}}$ so that the NNI move on σ at a child $G \in \text{Ch}(I, \sigma)$ yields σ_1 .

- If I is Type 1 incompatible with $\{S_L, S_R\}$, then using Remark 3.5, one can verify that

$$\alpha(\sigma) = \alpha(\sigma_1) + 1 , \quad \text{and} \quad \beta(\sigma) = \beta(\sigma_1) . \quad (3.114)$$

Hence, (3.113) yields

$$d_{nav}(\sigma, \mathcal{BT}_{\{S_L, S_R\}}) = \alpha(\sigma) + \frac{3}{2}\beta(\sigma) . \quad (3.115)$$

- Otherwise (I is Type 2 incompatible with $\{S_L, S_R\}$), using Remark 3.5, one can similarly observe that

$$\alpha(\sigma) = \alpha(\sigma_1) - 2 , \quad \text{and} \quad \beta(\sigma) = \beta(\sigma_1) + 2 , \quad (3.116)$$

and so (3.113) becomes

$$d_{nav}(\sigma, \mathcal{BT}_{\{S_L, S_R\}}) = \alpha(\sigma) + \frac{3}{2}\beta(\sigma) . \quad (3.117)$$

Therefore, the result holds. ■

The observation made in Lemma 3.12 is a good example of how the dual representation of d_{nav} — both in terms of paths in the NNI graph, and in terms of a closed-form formula quantifying inter-cluster incompatibility — offers a practical compromise between the heretofore separate traditional approaches to constructing dissimilarities on \mathcal{BT}_S , those of edge comparison and of estimation of edit distances. A particular application of this dual nature is that d_{nav} coincides with the standard 1-norm of the special crossing matrix and possesses a nice decomposability property (as defined in [217]):

Theorem 3.2 All NNI navigation paths joining $\sigma, \tau \in \mathcal{BT}_S$ are of equal length, given by:¹⁴

$$d_{nav}(\sigma, \tau) = \|\mathbf{S}(\sigma, \tau)\|_1, \quad (3.118a)$$

$$= d_{nav}(\sigma, \mathcal{BT}_{\text{Ch}(S, \tau)}) + d_{nav}(\sigma|_{S_L}, \tau|_{S_L}) + d_{nav}(\sigma|_{S_R}, \tau|_{S_R}), \quad (3.118b)$$

where $\mathbf{S}(\sigma, \tau)$ (3.104) is the special crossing matrix of σ and τ , and $\{S_L, S_R\} = \text{Ch}(S, \tau)$, and $d_{nav}(\sigma, \mathcal{BT}_{\text{Ch}(S, \tau)})$ (3.111) is the NNI navigation dissimilarity of σ to $\mathcal{BT}_{\text{Ch}(S, \tau)}$ (3.98).

Proof If $\text{Ch}(S, \sigma) = \text{Ch}(S, \tau)$, then the result directly follows from the decomposability property of the NNI control law in Section 3.3.3.

Otherwise, let $\Gamma = \{\sigma^0 = \sigma, \sigma^1, \sigma^2, \dots, \sigma^M = \tau\}$ be an NNI navigation path, consistent with the NNI control \mathbf{u}_τ , from σ to τ , where $M \in \mathbb{N}$ is the length of Γ . Let $\mathcal{G} = (G^0, G^1, \dots, G^{M-1})$ and $\mathcal{K} = (K^0, K^1, \dots, K^{M-1})$ be the ordered set of grandchild and common clusters with crossing splits selected by the NNI control law \mathbf{u}_τ while constructing Γ . That is to say, for all $0 \leq m \leq M-1$, we have $G^m = \mathbf{u}_\tau(\sigma^m)$ and $\sigma^{m+1} = \text{NNI}(\sigma^m, G^m)$.

Now consider a reordering of \mathcal{G} and \mathcal{K} into $\mathcal{G}_f = (G^{f(0)}, G^{f(1)}, \dots, G^{f(M-1)})$ and $\mathcal{K}_f = (K^{f(0)}, K^{f(1)}, \dots, K^{f(M-1)})$, respectively, using an a bijective function $f : [0, M-1] \rightarrow [0, M-1]$ with the property that for any $i, j \in [0, M-1]$

- the order relation of any pair of common clusters is consistent with inclusion: that is, if $i \leq j$, then $K^{f(i)} \supseteq K^{f(j)}$ or $K^{f(i)} \cap K^{f(j)} = \emptyset$.
- the order relation of any pair of grandchildren in \mathcal{G} associated with the same common cluster $K \in \mathcal{K}$ is preserved in \mathcal{G}_f , i.e., if $K^i = K^j$, then $f(i) \leq f(j) \iff i \leq j$.

For example, consider the following order relation \leq_f on $[0, M-1]$: for any $i, j \in [0, M-1]$,

$$i \leq_f j \iff \begin{cases} i \leq j, & \text{if } K^i = K^j, \\ \min(K^i) \leq \min(K^j), & \text{if } K^i \cap K^j = \emptyset, \\ K^i \supseteq K^j, & \text{otherwise.} \end{cases} \quad (3.119)$$

The total ordering of $\{0, M-1\}$ according to \leq_f defines f , i.e. for any $i, j \in [0, M-1]$ $f(i) \leq_f f(j)$.

Observe that, by decomposability property of the NNI control law, the NNI moves associated with \mathcal{G}_f define an NNI navigation path Γ_f joining σ to τ . Note that Γ and Γ_f might be different, but they have the same length.

Hence, all NNI navigation paths from σ to τ can be rearranged in an appropriate way so that they first solve the incompatibilities of σ with the root split $\text{Ch}(S, \tau)$ of τ and continue with joining subtrees of the root. To put it another way, using the decomposability property of the NNI control law, the length of any NNI navigation path Γ joining σ to τ can be recursively obtained as

$$M = d_{nav}(\sigma, \mathcal{BT}_{\text{Ch}(S, \tau)}) + d_{nav}(\sigma|_{S_L}, \tau|_{S_L}) + d_{nav}(\sigma|_{S_R}, \tau|_{S_R}), \quad (3.120)$$

where $\{S_L, S_R\} = \text{Ch}(S, \tau)$.

¹⁴Note that $d_{nav}(\sigma, \tau)$ is always zero for $|J| = 2$, which is the base case of the recursion in (3.118). For any $I \subseteq S$ with $|I| = 1$ we must set $d_{nav}(\sigma|_I, \tau|_I) = 0$ in order for d_{nav} to be a dissimilarity.

Moreover, using Lemma 3.9, one can rewrite (3.120) as

$$M = \sum_{\substack{J \in \mathcal{C}(\tau) \\ I \in \mathcal{C}(\sigma|_J)}} (\eta \circ \kappa) (\text{Ch}(I, \sigma|_J), \text{Ch}(J, \tau)), \quad (3.121)$$

where κ and η are, respectively, defined as in (3.105).

Let $J \in \mathcal{C}(\tau)$ with $|J| \geq 2$. As discussed in the proof of Lemma 3.2, an interior cluster K of $\sigma|_J$ is associated with a unique cluster I of σ such that $K = I \cap J$ and $\text{Ch}(I, \sigma)|_J \neq \{I \cap J, \emptyset\}$. Hence, observe that $\mathcal{C}(\sigma|_J)$ can be written as

$$\mathcal{C}(\sigma|_J) = \left\{ I \cap J \mid I \in \mathcal{C}(\sigma), I \cap J \neq \emptyset \right\}, \quad (3.122)$$

$$= \left\{ I \cap J \mid I \in \mathcal{C}(\sigma), \text{Ch}(I, \sigma)|_J \neq \{I \cap J, \emptyset\} \right\} \cup \bigcup_{i \in J} \{i\}. \quad (3.123)$$

Accordingly, let $\mathcal{C}_J(\sigma)$ denote the set of all clusters of σ defining the interior clusters of $\sigma|_J$,

$$\mathcal{C}_J(\sigma) = \left\{ I \in \mathcal{C}(\sigma) \mid \text{Ch}(I, \sigma)|_J \neq \{I \cap J, \emptyset\} \right\}. \quad (3.124)$$

Now, we manipulate (3.121) into the desired form of the result:

$$M = \sum_{\substack{J \in \mathcal{C}(\tau) \\ I \in \mathcal{C}(\sigma|_J)}} \eta \left(\sum_{A \in \text{Ch}(I, \sigma|_J)} \underbrace{\mathbb{1}(A \not\bowtie \text{Ch}(J, \tau))}_{\substack{= A \not\bowtie \text{Ch}(J, \tau)|_I \\ \text{by Lemma 3.11}}} \right) = \sum_{\substack{J \in \mathcal{C}(\tau) \\ I \in \mathcal{C}(\sigma|_J)}} \eta \left(\sum_{A \in \text{Ch}(I, \sigma|_J)} \underbrace{\mathbb{1}(A \not\bowtie \text{Ch}(J, \tau)|_I)}_{\substack{=0 \text{ if } |I|=1, \\ \text{otherwise there is exactly one } \tilde{I} \in \mathcal{C}_J(\sigma) \text{ s.t.} \\ \text{Ch}(I, \sigma|_J) = \text{Ch}(\tilde{I}, \sigma)|_J}} \right), \quad (3.125)$$

$$= \sum_{\substack{J \in \mathcal{C}(\tau) \\ I \in \mathcal{C}_J(\sigma)}} \eta \left(\underbrace{\sum_{A \in \text{Ch}(I, \sigma)|_J} \mathbb{1}(A \not\bowtie \text{Ch}(J, \tau)|_I)}_{\substack{=0 \text{ for all } I \in \mathcal{C}(\sigma) \setminus \mathcal{C}_J(\sigma) \\ \text{since } \text{Ch}(I, \sigma)|_J = \{I \cap J, \emptyset\}}} \right) = \sum_{\substack{J \in \mathcal{C}(\tau) \\ I \in \mathcal{C}(\sigma)}} \eta \left(\underbrace{\sum_{A \in \text{Ch}(I, \sigma)|_J} \mathbb{1}(A \not\bowtie \text{Ch}(J, \tau)|_I)}_{= \kappa(\text{Ch}(I, \sigma)|_J, \text{Ch}(J, \tau)|_I)} \right), \quad (3.126)$$

$$= \sum_{\substack{J \in \mathcal{C}(\tau) \\ I \in \mathcal{C}(\sigma)}} (\eta \circ \kappa) (\text{Ch}(I, \sigma)|_J, \text{Ch}(J, \tau)|_I) = \|\mathbf{S}(\sigma, \tau)\|_1. \quad (3.127)$$

Thus, in addition to (3.120), the length of any NNI navigation path from σ to τ is equivalently given by $\|\mathbf{S}(\sigma, \tau)\|_1$. Due to symmetry, the length of any NNI navigation path from τ to σ is equal to $\|\mathbf{S}(\tau, \sigma)\|_1$. Recall from Corollary 3.5 that $\|\mathbf{S}(\sigma, \tau)\|_1 = \|\mathbf{S}(\tau, \sigma)\|_1$ for any binary tree $\sigma, \tau \in \mathcal{BT}_S$. Therefore, the length of any NNI navigation path joining σ and τ is equal to $\|\mathbf{S}(\sigma, \tau)\|_1 = \|\mathbf{S}(\tau, \sigma)\|_1$, and the result follows. \blacksquare

3.4.1 Basic Properties

We now continue with a list of significant properties of d_{nav} and its relation with other tree measures:

Proposition 3.11 *The NNI navigation dissimilarity d_{nav} is positive definite and symmetric, but it does not define a metric.*

Proof That d_{nav} is positive definite follows directly from its definition. Lemma 3.10 proves it is symmetric and Corollary 3.2 and Figure 3.3 show where the triangle inequality fails. ■

Theorem 3.2 yields the following (very crude) bounds on the performance of the corresponding NNI navigation algorithm:

Proposition 3.12 *The length of a navigation path through \mathcal{N}_S does not exceed $O(|S|^2)$.*

Proof Let $n = |S|$. For all $\sigma, \tau \in \mathcal{BT}_S$, we have $|\mathcal{C}(\tau)| = O(n)$, implying $\mathbf{S}(\sigma, \tau)$ has $O(n^2)$ entries. The value of $d_{nav}(\sigma, \tau)$ never exceeds three times the number of entries in $\mathbf{S}(\sigma, \tau)$. ■

The decomposability of the NNI navigation dissimilarity may be used for its efficient computation:

Lemma 3.13 *For any bipartition $\{S_L, S_R\}$ of S and $\sigma \in \mathcal{BT}_S$, the NNI navigation dissimilarity $d_{nav}(\sigma, \mathcal{BT}_{\{S_L, S_R\}})$ (3.111) can be computed in linear time, $O(|S|)$.*

Proof As discussed in the proof of Proposition 3.4, all cluster compatibilities of σ with $\{S_L, S_R\}$ can be determined in $O(|S|)$ time using Lemma 3.5 and by the post order traversal of σ . Therefore, the NNI navigation dissimilarity $d_{nav}(\sigma, \mathcal{BT}_{\{S_L, S_R\}})$ in (3.111) can be computed in $O(|S|)$ by a complete traversal of σ . ■

Proposition 3.13 *The NNI navigation dissimilarity d_{nav} on \mathcal{BT}_S is computable in $O(|S|^2)$ time.*

Proof Let $\sigma, \tau \in \mathcal{BT}_S$ and $\{S_L, S_R\} = \text{Ch}(S, \tau)$. The recursion in (3.118) requires $d_{nav}(\sigma, \mathcal{BT}_{\text{Ch}(S, \tau)})$, which can be obtained in $O(|S|)$ time (Lemma 3.13), and the restrictions of σ to S_L and S_R , which can be computed using the post-order traversal of σ in linear time, $O(|S|)$. Hence, d_{nav} requires a complete (depth-first) traversal of τ each of whose iteration costs linear time with the number of leaves. Thus, the recursive computation of d_{nav} can be preformed in $O(|S|^2)$ time. Also note that $\mathbf{S}(\sigma, \tau)$ (3.104) can be obtained with the same cost, $O(|S|^2)$, similar to our computation of $\mathbf{X}(\sigma, \tau)$ (3.38) in the proof of Proposition 3.4. ■

Lemma 3.14 *Let $\{S_L, S_R\}$ be a bipartition of S . An NNI navigation path, consistent with the control law $\mathbf{u}_{\{S_L, S_R\}}$, starting from $\sigma \in \mathcal{BT}_S$ ending in $\mathcal{BT}_{\{S_L, S_R\}}$ can be computed in $O(|S|)$ time.*

Proof Let $\tau \in \mathcal{BT}_{\{S_L, S_R\}}$. All cluster of σ incompatible with $\{S_L, S_R\}$, i.e. $\mathcal{J}(\sigma, \tau; S)$, can be determined in $O(|S|)$ time as discussed in the proof of Lemma 3.13. Hence, if $\sigma \notin \mathcal{BT}_{\{S_L, S_R\}}$, then a deep cluster $I \in \mathcal{D}(\sigma, \tau; S)$ incompatible with $\{S_L, S_R\}$ can be found in $O(|S|)$ by the post-order traversal of σ .

Let $G \in \text{Ch}(I, \sigma)$ be the grandchild chosen by $\mathbf{u}_{\{S_L, S_R\}}$ yielding the next hierarchy $\gamma = \text{NNI}(\sigma, G)$. Recall from Remark 3.5 that if I is Type 1, then I is replaced by a compatible cluster in γ , and so $\mathcal{J}(\gamma, \tau; S) = \mathcal{J}(\sigma, \tau; S) \setminus \{I\}$. Hence, all the descendant of $\text{Pr}(I, \sigma)$ in γ are compatible with $\{S_L, S_R\}$. If I is Type 2, then I is replaced by $G^{-\sigma} \cup I^{-\sigma}$ incompatible with $\{S_L, S_R\}$, and so $\mathcal{J}(\gamma, \tau; S) = (\mathcal{J}(\sigma, \tau; S) \cup \{G^{-\sigma} \cup I^{-\sigma}\}) \setminus \{I\}$. Note that $I^{-\sigma}$ and $G^{-\sigma} \cup I^{-\sigma}$ are the only descendant of $\text{Pr}(I, \sigma)$ in γ incompatible with $\{S_L, S_R\}$. Since both $I^{-\sigma}$ and $G^{-\sigma} \cup I^{-\sigma}$ in σ are Type 1, it requires two more NNI moves to resolve all incompatibilities descending $\text{Pr}(I, \sigma)$ in γ with $\{S_L, S_R\}$.

Thus, instead of searching for a deep cluster in γ starting from the root S , using Corollary 3.4, one can continue the post-order search for a deep cluster in γ at $\text{Pr}(I, \sigma)$.

In fact, observe that $\mathcal{J}(\sigma, \tau; S) \cup \{S\}$ defines a tree-like data structure (see Figure 3.5). Therefore, one can conclude that the overall construction of a NNI navigation path only requires a complete post-order traversal of σ for $\mathcal{J}(\sigma, \tau; S)$ in $O(|S|)$ time. ■

Proposition 3.14 *An NNI navigation path joining $\sigma \in \mathcal{BT}_S$ to $\tau \in \mathcal{BT}_S$ consistent with the NNI control \mathbf{u}_τ can be computed in $O(|S|^2)$ time.*

Proof Similar to the recursive expression of d_{nav} (3.118b), an NNI navigation path joining σ to τ can be found using the decomposability property within a divide-and-conquer approach as follows: first obtain an NNI navigation path from σ to $\mathcal{BT}_{\{S_L, S_R\}}$ in $O(|S|)$ (Lemma 3.14) and then find NNI navigation paths between subtrees. Hence, this requires the pre-order traversal of τ each of whose step costs $O(|S|)$. Thus, an NNI navigation path joining σ to τ can be recursively computed in $O(|S|^2)$ time, which completes the proof. ■

Lemma 3.15 *Let $\{S_L, S_R\}$ be a bipartition of S and $\sigma \in \mathcal{BT}_S$. Then we have the tight bound:*

$$d_{nav}(\sigma, \mathcal{BT}_{\{S_L, S_R\}}) \leq |S| + \min(|S_L|, |S_R|) - 3. \quad (3.128)$$

Proof For any $\sigma \in \mathcal{BT}_S$ and $I \in \mathcal{C}(\sigma)$ observe that

- (i) if I is a singleton or $|I| = 2$, we have $(\eta \circ \kappa)(\text{Ch}(I, \sigma), \{S_L, S_R\}) = 0$,
- (ii) otherwise for larger clusters $(\eta \circ \kappa)(\text{Ch}(I, \sigma), \{S_L, S_R\})$ equals 3 or 1 only if, respectively, both clusters or only one cluster of $\text{Ch}(I, \sigma)$ are incompatible with $\{S_L, S_R\}$.

Here, κ and η are, respectively, defined as in (3.105). Since there are at least $|S| + 1$ clusters of the first kind, there are at most $|S| - 2$ clusters of the second kind. Thus, applying Lemma 3.12 we have

$$d_{nav}(\sigma, \mathcal{BT}_{\{S_L, S_R\}}) \leq (|S| - 2) + |\mathcal{X}|, \quad (3.129)$$

where \mathcal{X} is the set of all $I \in \mathcal{C}(\sigma)$ both of whose children are incompatible with $\{S_L, S_R\}$,

$$\mathcal{X} := \left\{ I \in \mathcal{C}(\sigma) \mid \forall D \in \text{Ch}(I, \sigma), D \not\bowtie \{S_L, S_R\} \right\}. \quad (3.130)$$

For each $I \in \mathcal{X}$, both $I \cap S_L$ and $I \cap S_R$ are nonsingleton clusters of $\sigma|_{S_L}$ and $\sigma|_{S_R}$, respectively (each child of I intersects each of S_L, S_R). Suppose now that $I, J \in \mathcal{X}$ are distinct. There are two cases, without loss of generality:

- If $I \cap J = \emptyset$, then $I \cap S_L \neq J \cap S_L$ (and similarly for S_R);
- If $I \subsetneq J$, then J has a child I' disjoint from I , and this child must intersect S_L . Hence, $I \cap S_L \subsetneq J \cap S_L$.

We conclude that the map $I \mapsto I \cap S_L$ (respectively $I \cap S_R$) of \mathcal{X} to $\mathcal{C}(\sigma|_{S_L})$ (respectively to $\mathcal{C}(\sigma|_{S_R})$) is injective, and has no singleton clusters in its image. Thus, $|\mathcal{X}| \leq \min(|S_L| - 1, |S_R| - 1)$, proving the desired inequality.

The example $\sigma, \tau \in \mathcal{BT}_{[n]}$ in Figure 3.4 with $\{S_L, S_R\} = \text{Ch}([n], \tau) = \{\{1\}, \{2, 3, \dots, n\}\}$ shows that the upper bound in (3.128) is tight (where $d_{\text{nav}}(\sigma, \mathcal{BT}_{\{S_L, S_R\}}) = n - 2$). ■

Proposition 3.15 $\text{diam}(\mathcal{BT}_S, d_{\text{nav}}) = \frac{1}{2}(|S| - 1)(|S| - 2)$.

Proof We proceed by induction over $|S|$, with the base case $|S| = 2$ satisfying $|\mathcal{BT}_S| = 1$. The formula then holds trivially, as $d_{\text{nav}} = 0$.

For the induction step, assume $|S| \geq 3$ and that $\sigma, \tau \in \mathcal{BT}_S$ satisfy $d_{\text{nav}}(\sigma|_K, \tau|_K) \leq \frac{1}{2}(|K| - 1)(|K| - 2)$ for every $K \in \text{Ch}(S, \tau) = \{S_L, S_R\}$. Then, using the recursive formula of d_{nav} in (3.118) in Theorem 3.2, we obtain:

$$d_{\text{nav}}(\sigma, \tau) = \underbrace{d_{\text{nav}}(\sigma, \mathcal{BT}_{\text{Ch}(S, \tau)})}_{\substack{\text{by Lemma 3.15} \\ \leq |S| + \min(|S_L|, |S_R|) - 3}} + \underbrace{d_{\text{nav}}(\sigma|_{S_L}, \tau|_{S_L})}_{\substack{\text{by induction} \\ \leq \frac{1}{2}(|S_L| - 1)(|S_L| - 2)}} + \underbrace{d_{\text{nav}}(\sigma|_{S_R}, \tau|_{S_R})}_{\substack{\text{by induction} \\ \leq \frac{1}{2}(|S_R| - 1)(|S_R| - 2)}}, \quad (3.131)$$

$$\leq \frac{1}{2}(|S_L|^2 + |S_R|^2) - \frac{3}{2} \underbrace{(|S_L| + |S_R|)}_{|S|} + |S| + \min(|S_L|, |S_R|) - 1, \quad (3.132)$$

$$= \frac{1}{2} \left(|S|^2 - 2 \underbrace{|S_L||S_R|}_{=\min(|S_L|, |S_R|)(|S| - \min(|S_L|, |S_R|))} \right) - \frac{1}{2}|S| + \min(|S_L|, |S_R|) - 1, \quad (3.133)$$

$$= \frac{1}{2}(|S| - 1)(|S| - 2) + \underbrace{(1 - \min(|S_L|, |S_R|))(|S| - \min(|S_L|, |S_R|) - 2)}_{\leq 0, \forall |S| \geq 3}, \quad (3.134)$$

$$\leq \frac{1}{2}(|S| - 1)(|S| - 2). \quad (3.135)$$

Here, recall that $d_{\text{nav}}(\sigma|_I, \tau|_I) = 0$ for any $I \subset S$ with $|I| = 1$. Finally, the trees in Figure 3.4 realize this bound on the diameter. ■

3.4.2 Relations with Other Tree Measures

Although d_{nav} is not a true metric, like d_{CM} (Proposition 3.7), it can be tightly bounded from below and above in terms of d_{RF} as follows:

Proposition 3.16 Over \mathcal{BT}_S one has $d_{RF} \leq d_{\text{nav}} \leq \frac{1}{2}d_{RF}^2 + \frac{1}{2}d_{RF}$ and both bounds are tight.

Proof Since d_{nav} is realized by paths in the NNI graph we have $d_{NNI} \leq d_{nav}$. The lower bound then follows from $d_{RF} \leq d_{NNI}$ (Corollary 3.1). The bound is tight because for any $\sigma, \tau \in \mathcal{BT}_S$,

$$d_{RF}(\sigma, \tau) = 1 \iff d_{NNI}(\sigma, \tau) = 1 \iff d_{nav}(\sigma, \tau) = 1. \quad (3.136)$$

Let us deduce the upper bound now. In the case when σ and τ have no nontrivial clusters in common, $d_{RF}(\sigma, \tau) = |S| - 2$ and the result follows from Proposition 3.15. We argue by induction over $|S|$ to prove the upper bound in the generic case, keeping in mind that for $|S| = 2$ the result holds trivially, because $|\mathcal{BT}_{[2]}| = 1$. For the induction step ($|S| \geq 3$), let $I \in \mathcal{C}(\sigma) \cap \mathcal{C}(\tau)$ be a nontrivial common cluster and prune σ and τ at cluster I yielding subtrees $\sigma_I = \sigma|_I$ and $\tau_I = \tau|_I$. Accordingly, let $\sigma_{\neg I} = \sigma|_{\neg I}$ and $\tau_{\neg I} = \tau|_{\neg I}$, respectively, denote the remaining parts of σ and τ after pruning subtrees rooted at cluster I , where $\neg I = I^C \cup \{i\}$ and $i \in I$. Here, both $\sigma_{\neg I}$ and $\tau_{\neg I}$ contain leaf i as a representative of each associated pruned subtree. Note that $|I| \geq 2$ and $|\neg I| \geq 2$.

Since the NNI control law preserves the common edges and possess the decomposability property (see Section 3.3.3), the length of the NNI navigation path can be written as

$$d_{nav}(\sigma, \tau) = d_{nav}(\sigma_I, \tau_I) + d_{nav}(\sigma_{\neg I}, \tau_{\neg I}). \quad (3.137)$$

Similarly, by definition of d_{RF} , we have

$$d_{RF}(\sigma, \tau) = d_{RF}(\sigma_I, \tau_I) + d_{RF}(\sigma_{\neg I}, \tau_{\neg I}). \quad (3.138)$$

Let $\alpha = d_{RF}(\sigma_I, \tau_I)$ and $\beta = d_{RF}(\sigma_{\neg I}, \tau_{\neg I})$, and so $d_{RF}(\sigma, \tau) = \alpha + \beta$. We may apply the induction hypothesis in \mathcal{BT}_I and $\mathcal{BT}_{\neg I}$ to conclude that

$$d_{nav}(\sigma_I, \tau_I) \leq \frac{1}{2}\alpha(\alpha + 1) \quad \text{and} \quad d_{nav}(\sigma_{\neg I}, \tau_{\neg I}) \leq \frac{1}{2}\beta(\beta + 1). \quad (3.139)$$

It then follows from (3.137) that

$$d_{nav}(\sigma, \tau) \leq \frac{1}{2}\alpha(\alpha + 1) + \frac{1}{2}\beta(\beta + 1) \leq \frac{1}{2}(\alpha + \beta)(\alpha + \beta + 1), \quad (3.140)$$

$$= \frac{1}{2}d_{RF}(\sigma, \tau)(d_{RF}(\sigma, \tau) + 1). \quad (3.141)$$

Finally, Proposition 3.15 ensures that this bound is tight. ■

Proposition 3.17 *Over \mathcal{BT}_S one has $d_{nav}(\sigma, \tau) \leq \frac{3}{2}d_{CM}(\sigma, \tau)$.*

Proof Consider the closed form expression of d_{nav} (3.118a) in terms of the special crossing matrix $\mathbf{S}(\sigma, \tau)$ (3.104),

$$d_{nav}(\sigma, \tau) = \|\mathbf{S}(\sigma, \tau)\|_1 = \sum_{\substack{I \in \mathcal{C}(\sigma) \\ J \in \mathcal{C}(\tau)}} (\eta \circ \kappa)(\text{Ch}(I, \sigma)|_J, \text{Ch}(J, \tau)|_I). \quad (3.142)$$

Note that $\kappa(\mathcal{A}, \mathcal{B}) \in \{0, 1, 2\}$ for any $\mathcal{A}, \mathcal{B} \subseteq \mathcal{P}(S)$ with the property that $|\mathcal{A}| = 2$, and $\eta(x) = \frac{1}{2}x^2 + \frac{1}{2}x \leq \frac{3}{2}x$ for all $x \in [0, 2]$. Hence, d_{nav} can be bounded from above as

$$d_{nav}(\sigma, \tau) \leq \frac{3}{2} \sum_{\substack{I \in \mathcal{C}(\sigma) \\ J \in \mathcal{C}(\tau)}} \kappa(\text{Ch}(I, \sigma)|_J, \text{Ch}(J, \tau)|_J) , \quad (3.143)$$

$$= \frac{3}{2} \sum_{\substack{I \in \mathcal{C}(\sigma) \\ J \in \mathcal{C}(\tau)}} \sum_{A \in \text{Ch}(I, \sigma)|_J} \underbrace{\mathbb{1}(A \not\bowtie \text{Ch}(J, \tau)|_J)}_{\sum_{B \in \text{Ch}(J, \tau)|_I} \mathbb{1}(A \not\bowtie B)} , \quad (3.144)$$

$$\leq \frac{3}{2} \sum_{\substack{I \in \mathcal{C}(\sigma) \\ J \in \mathcal{C}(\tau)}} \mathbb{1}\left((I \cap \text{Pr}(J, \tau)) \not\bowtie (J \cap \text{Pr}(I, \sigma))\right) . \quad (3.145)$$

Now, let A, B, C be sets satisfying $B \subseteq C$. Then $A \bowtie B$ implies $(A \cap C) \bowtie B$ (the reverse may not be true), producing $\mathbb{1}(A \bowtie B) \leq \mathbb{1}((A \cap C) \bowtie B)$. Using the contra-positive, the upper bound (3.145) on d_{nav} may be rewritten as

$$d_{nav}(\sigma, \tau) \leq \frac{3}{2} \sum_{\substack{I \in \mathcal{C}(\sigma) \\ J \in \mathcal{C}(\tau)}} \mathbb{1}(I \not\bowtie J) = \frac{3}{2} \|\mathbf{X}(\sigma, \tau)\|_1 = \frac{3}{2} d_{CM}(\sigma, \tau) , \quad (3.146)$$

which completes the proof. ■

The overall ordering of tree dissimilarities in Corollary 3.1, Proposition 3.8 and Proposition 3.17 can be combined as:

Theorem 3.3 *For nondegenerate hierarchies,*

$$\frac{2}{3} d_{RF} \leq \frac{2}{3} d_{NNI} \leq \frac{2}{3} d_{nav} \leq d_{CM} \leq d_{CC} . \quad (3.147)$$

Finally, we remark that the NNI navigation dissimilarity d_{nav} can be generalized to a pair of trees, σ and τ , in \mathcal{T}_S as

$$d_{nav}(\sigma, \tau) = \frac{1}{2} (\|\mathbf{S}(\sigma, \tau)\|_1 + \|\mathbf{S}(\tau, \sigma)\|_1) , \quad (3.148)$$

which is nonnegative and symmetric. For nondegenerate trees $\sigma, \tau \in \mathcal{BT}_S$, one has $\mathbf{S}(\sigma, \tau) = \mathbf{S}(\tau, \sigma)^T$ (which is evident from Lemma 3.10), so that d_{nav} in (3.148) simplifies back to (3.118a). Although the closed form expression of d_{nav} in Theorem 3.2 enables the generalization of d_{nav} to degenerate trees as above, the notion of NNI moves (Definition 3.4) is generally not valid in \mathcal{T}_S .

As for nondegenerate trees in Proposition 3.17, the generalized d_{nav} in \mathcal{T}_S can be bounded above by d_{CM} as follows:

Proposition 3.18 *Over \mathcal{T}_S one has $d_{nav} \leq \left(\frac{1}{8}|S|^2 + \frac{1}{4}|S|\right) d_{CM}$.*

Proof Note that the number of nontrivial children of a cluster in a tree can be at most $\frac{1}{2}|S|$. Hence one can verify the result following similar steps as in the proof of Proposition 3.17. ■

3.5 Discussion and Statistical Analysis

3.5.1 Consensus Models and Median Trees

Let us recall a definition: a *median tree* of a set of sample trees is a tree whose sum of distances to the sample trees is minimum. Although the notion of a median tree is simple and well-defined, finding a median tree of a set of trees is generally a hard combinatorial problem. On the other hand, a consensus model of a set of sample trees is a computationally efficient tool to identify common structures of sample trees. In particular, a remark relating d_{CM} and d_{nav} to commonly used consensus models of a set of trees and their median tree(s) is:

Proposition 3.19 *Both the strict and loose consensus trees, T_* and T^* , of any set of trees T in \mathcal{T}_S (Definition 3.3) are median trees with respect to both the crossing, d_{CM} (3.39), and NNI navigation, d_{nav} (3.118), dissimilarities. In fact, for any $d \in \{d_{CM}, d_{nav}\}$ one has:*

$$\sum_{\tau \in T} d(\tau, T_*) = \sum_{\tau \in T} d(\tau, T^*) = 0. \quad (3.149)$$

Moreover, the loose consensus tree is the maximal (finest) median tree sharing each of its clusters with at least one sample tree.

Proof By Definition 3.3, both strict and loose consensus trees only contain clusters that are compatible with the clusters of every tree in T , and the loose consensus tree is the finest median tree containing only clusters from the sample trees. Thus, the result follows for both d_{CM} and d_{nav} due their relation in Proposition 3.18. ■

3.5.2 Sample Distribution of Dissimilarities

To compare their discriminative power, we use a standard statistical analysis of empirical distributions of different tree measures. The shape of the distribution of a tree measure tells how informative it is; for example, a highly concentrated distribution means that the associated tree measure behaves like the discrete metric¹⁵ as in the case of the Robinson-Foulds distance — see Figure 3.9. Finding a closed form expression for the distribution of a tree measure is a hard problem, and so extensive numerical simulations are generally applied to obtain its sample distribution. In particular, using the uniform and Yule model [197] for generating random trees, we compute the empirical distributions of d_{RF} , d_{MS} , d_{CC} , d_{CM} , and d_{nav} as illustrated in Figure 3.9.¹⁶ Moreover, in Table 3.1 we present two commonly used statistical measures, skewness and kurtosis, for describing the shapes of the probability distributions of all these tree measures. Here, recall that the skewness of a probability distribution measures its tendency on one side of the mean, and the concept of kurtosis measures the peakedness of the distribution [181]. In addition to their computational advantage over d_{MS} , as illustrated in both Figure 3.9 and Table 3.1, like d_{MS} , our tree measures, d_{CC} , d_{CM} and d_{nav} , are significantly more discriminative, with wider ranges of values and symmetry, than d_{RF} .

¹⁵The discrete metric $d : X \times X \rightarrow \mathbb{R}_{\geq 0}$ on a set X is defined as for any $x \neq y \in X$, $d(x, x) = 0$ and $d(x, y) = 1$.

¹⁶In our numerical simulations for any chosen tree measure we observe the same pattern of sample distribution for different numbers of leaves, and so here we only include results for $\mathcal{BT}_{[25]}$.

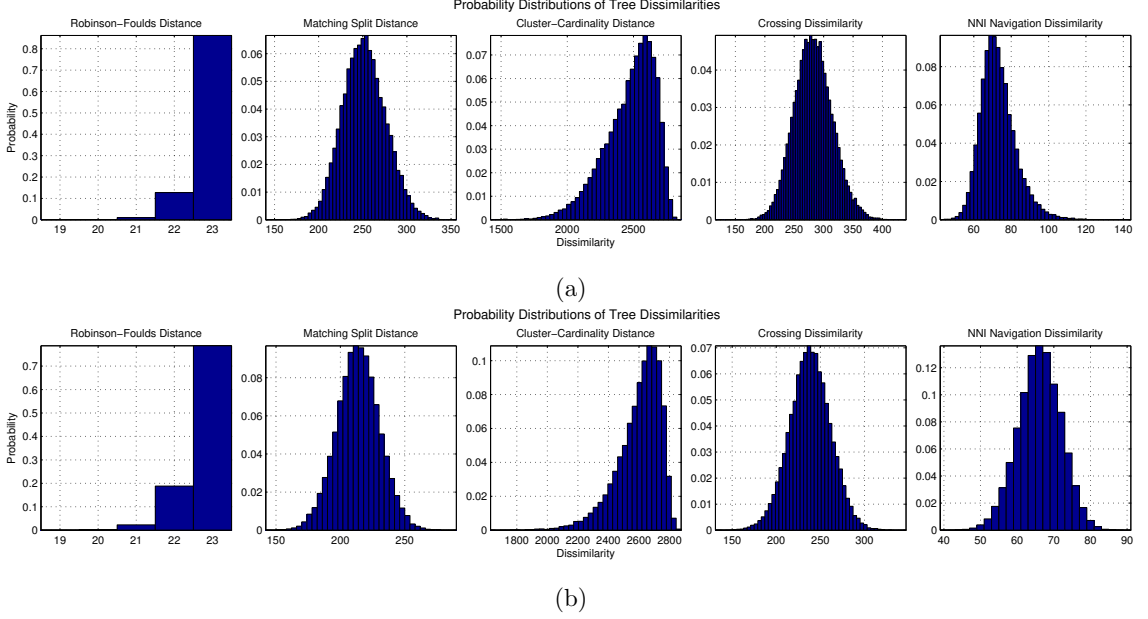


Figure 3.9: Empirical distribution of tree dissimilarities in $\mathcal{BT}_{[25]}$: (from left to right) the Robinson-Foulds distance d_{RF} (3.17), the matching split distance d_{MS} (Def.3.8), the cluster-cardinality distance d_{CC} (3.23), the crossing dissimilarity d_{CM} (3.39), and the NNI navigation dissimilarity d_{nav} (3.118). 100000 sample hierarchies are generated using (a) the uniform and (b) Yule model [197]. The resolutions of histograms of tree measures, from left to right, are 1, 4, 32, 4, 2 unit(s), respectively.

Table 3.1: Skewness and Kurtosis Values for the Distributions of Tree Measures in $\mathcal{BT}_{[25]}$

	Skewness		Kurtosis	
	Uniform	Yule	Uniform	Yule
d_{RF} (3.17)	-2.6162	-2.0740	9.8609	7.3998
d_{MS} (Def. 3.8)	0.1293	-0.0117	3.0060	3.1136
d_{CC} (3.23)	-0.9294	-1.2507	3.8601	5.2724
d_{CM} (3.39)	0.1390	-0.0405	3.1275	3.2103
d_{nav} (3.118)	0.8809	-0.1195	4.8707	3.0746

3.6 Summary

In this chapter, we present three new tree measures for efficient discriminative comparison of trees. First, using the well known relation between trees and ultrametrics, the cluster-cardinality metric d_{CC} is constructed as the pullback of matrix norms along an embedding of trees into the space of matrices. Second, we present the crossing dissimilarity d_{CM} that counts the pairwise incompatibilities of trees. Third, the NNI navigation dissimilarity d_{nav} while presented in closed form is constructed as the length of a navigation path in the space of trees.

All of our dissimilarities can be computed in $O(n^2)$ with the number of leaves n , and they generalize to the degenerate trees as well. Moreover, we provide a closed form expression for each proposed dissimilarity and present an ordering relation between these tree dissimilarities and related tree metrics in the literature (Theorem 3.3). Our numerical studies, summarized in Figure 3.9, suggest that the proposed tree measures are significantly more informative and discriminative than the Robinson-Foulds distance d_{RF} , while maintaining a computational advantage over distances such as the matching-split distance d_{MS} [36, 143].

Finally, an NNI navigation path joining a pair of nondegenerate hierarchies is compatible with d_{RF} , d_{CM} and d_{CC} in the sense of Theorem 3.1 and Proposition 3.10, and can be efficiently computed with the same cost of d_{nav} in $O(n^2)$ time (Proposition 3.14). Consequently, NNI navigation paths are likely of some significance for consensus/average models or statistical analysis of trees.



Chapter 4

Sensor-Based Reactive Navigation in Unknown Convex Sphere Worlds

Agile navigation in dense human crowds [212, 106], or in natural forests, such as now negotiated by rapid flying [124, 170] and legged [224, 119] robots, strongly motivates the development of sensor-based reactive motion planners. By the term *reactive* [55, 140], we mean that motion is generated by a vector field arising from some closed-loop feedback policy issuing online force or velocity commands in real time as a function of instantaneous robot state. By the term *sensor-based*, we mean that information about the location of the environmental clutter to be avoided is limited to structure perceived within some local neighborhood of the robot’s instantaneous position — its sensor footprint.

In this chapter, we propose a new reactive motion planner taking the form of a feedback law for a first-order (velocity-controlled), perfectly, relatively (to a fixed goal location) sensed and actuated disk-shaped robot, that can be computed using only information about the robot’s instantaneous position and structure within its local sensor footprint. We assume the a priori unknown environment is a static topological sphere world [132], whose obstacles are convex and possess smooth boundaries whose curvature is “reasonably” high relative to their mutual separation. Under these assumptions, the proposed closed-loop vector field is guaranteed to bring all but a measure zero set of initial conditions to the desired goal. To the best of our knowledge, this is the first time a sensor-based reactive motion planner has been shown to be provably correct with respect to a general class of environments.

Motivation and Related Work Prior literature on reactive feedback motion planning has embraced the simple, computationally efficient artificial potential field ¹ approach to real-time obstacle avoidance [128] that incurs topologically necessary critical points [129], which, in practice, with no further remediation often include (topologically unnecessary) spurious local minima. Even in topologically simple settings such as the sphere worlds addressed here, constructions that eliminate these spurious attractors — e.g., navigation functions [183], or other methods ² [58] — have largely come at the price of complete prior information.

¹We adopt standard usage to denote by this term the use of the negative gradient field of a scalar valued function as the force or velocity control law for a fully actuated, kinematic (first order dynamics) robot.

²Although harmonic functions are utilized to design potential functions without local minima [58], such intrinsically numerical constructions forfeit the reactive nature of feedback motion planners under discussion here.

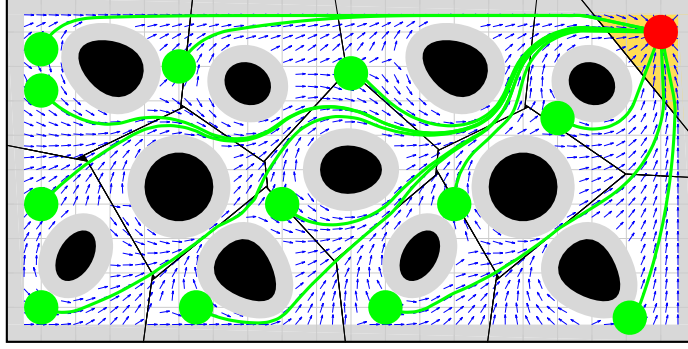


Figure 4.1: Exact navigation of a disk-shaped robot using separating hyperplanes of the robot body (red at the goal) and convex obstacles (black solid shapes). Separating hyperplanes between the robot and obstacles define an obstacle free convex neighborhood (the yellow region when the robot at the goal) of the robot, and the continuous feedback motion towards the metric projection of a given desired goal (red) onto this convex set asymptotically steers almost all robot configurations (green) to the goal without collisions along the way. The grey regions represent the augmented workspace boundary and obstacles, and the arrows show the direction of the resulting vector field.

Extensions to the navigation function framework partially overcoming the necessity of global prior knowledge of (and consequent parameter tuning for) a topologically and metrically simple environment have appeared in the last decade. Adjustable navigation functions are proposed to gradually update the tuning parameter upon the discovery of new obstacles [83], and locally computable navigation functions are introduced by restricting the effect of each obstacle in its immediate vicinity such that a robot is required to deal with at most one obstacle at a time [145]. What is more, sequential composition [45] has been used to cover metrically complicated environments with convex cell-based local potential decompositions [56] (and extended to nonholonomically constrained finite size robots [57]), but still necessitating prior global knowledge of the environment.

Contributions We herein abandon the smooth potential field approach to reactive planning, achieving an algorithm that is “doubly reactive” in the sense that not merely the integrated robot trajectory, but also its generating vector field can be constructed on the fly in real time using only local knowledge of the environment. Our piecewise smooth vector field combines some of the ideas of sensor-based exploration [53] with those of hybrid reactive control [56]. We use separating hyperplanes of convex bodies [37] to identify an obstacle free convex neighborhood of a robot configuration, and build our safe robot navigation field by control action towards the metric projection of the designated point destination onto this convex set.

Our construction requires no parameter tuning and requires only local knowledge of the environment in the sense that the robot needs only locate those proximal obstacles determining its collision free convex neighborhood. When the obstacles are sufficiently separated (Assumption 4.1 stipulates that the robot must be able to pass in between them) and sufficiently strongly convex at their “antipode” (Assumption 4.2 stipulates that they curve away from the enclosing sphere centered at the destination which just touches their

boundary at the most distant point), the proposed vector field generates a smooth flow with a unique attractor at the specified goal location along with (the topologically necessary number of) saddles — at least one associated with each obstacle. Since all of its critical points are nondegenerate, our vector field is guaranteed to steer almost all collision free robot configurations to the goal, while avoiding collisions along the way, as illustrated in Figure 4.1.

It proves most convenient to develop the theoretical properties of this construction under the assumption that the robot can identify and locate those nearby obstacles whose associated separating hyperplanes define the robot’s obstacle free convex neighborhood (a capability termed *Voronoi-adjacent obstacle sensing* in Section 4.2.2), no matter how physically distant they may be. Thus, to accommodate more physically realistic sensors, we adapt the initial construction (and the proof) to the case of two different limited range sensing modalities, while extending the same formal guarantees as in the erstwhile (local but unbounded range) idealized sensor model. Similarly, in the interest of greater practicability, we further extend the construction (and the proof) to the case of the commonly encountered kinematic differential drive vehicle model (retaining the convergence and collision avoidance guarantees, at the necessary cost of a discontinuous feedback law).

This chapter is based on the papers [13, 9]. In the conference paper [13], we propose a different construction based on power diagrams [20] navigating among spherical obstacles using knowledge of Voronoi-adjacent¹² obstacles to construct the robot’s local workspace [13, Eqn. (9)]. The submitted conference paper [9] introduces a new construction, presented in detail in this chapter, for that set in (4.9) based on separating hyperplanes, permitting an extension of the navigable obstacles to the broader class of convex bodies specified by Assumption 4.2, while providing the same guarantee of almost global asymptotic convergence (Theorem 4.3) to a given goal location. From the view of applications, the new appeal to separating hyperplanes permits the central advance of a purely reactive construction from limited range sensors (4.30), e.g., in the planar case from immediate line-of-sight appearance (4.40), with the same global guarantees.

4.1 Problem Formulation

Consider a disk-shaped robot, of radius $r \in \mathbb{R}_{>0}$ centered at $\mathbf{x} \in \mathcal{W}$, operating in a closed compact convex environment \mathcal{W} in the n -dimensional Euclidean space \mathbb{R}^n , where $n \geq 2$, punctured with $m \in \mathbb{N}$ open convex sets $\mathcal{O} := \{O_1, O_2, \dots, O_m\}$ with twice differentiable boundaries, representing obstacles.³ Hence, the free space \mathcal{F} of the robot is given by

$$\mathcal{F} := \left\{ \mathbf{x} \in \mathcal{W} \mid \overline{B(\mathbf{x}, r)} \subseteq \mathcal{W} \setminus \bigcup_{i=1}^m O_i \right\} . \quad (4.1)$$

where $B(\mathbf{x}, r) := \{\mathbf{q} \in \mathbb{R}^n \mid \|\mathbf{q} - \mathbf{x}\| < r\}$ is the open ball centered at \mathbf{x} with radius r , and $\overline{B(\mathbf{x}, r)}$ denotes its closure, and $\|\cdot\|$ denotes the standard Euclidean norm.

³Here, \mathbb{N} is the set of all natural numbers; \mathbb{R} and $\mathbb{R}_{>0}$ ($\mathbb{R}_{\geq 0}$) denote the set of real and positive (nonnegative) real numbers, respectively.

To maintain the local convexity of obstacle boundaries in the free space \mathcal{F} , we assume that our disk-shaped robot can freely fit in between (and thus freely circumnavigate) any of the obstacles throughout the workspace \mathcal{W} :⁴

Assumption 4.1 *Obstacles are separated from each other by clearance of at least*

$$d(O_i, O_j) > 2r, \quad \forall i \neq j, \quad (4.2)$$

and from the boundary $\partial\mathcal{W}$ of the workspace \mathcal{W} as

$$d(O_i, \partial\mathcal{W}) > 2r, \quad \forall i. \quad (4.3)$$

where $d(A, B) := \inf \{ \|a - b\| \mid a \in A, b \in B \}$.

Before formally stating our navigation problem, it is useful to recall a known topological limitation of reactive planners: if a continuous vector field planner on a generalized sphere world has a unique attractor, then it must have at least as many saddles as obstacles [132]. In consequence, the robot navigation problem that we seek to solve is stated as:

Reactive Navigation Problem *Assuming the first order (completely actuated single integrator) robot dynamics,*

$$\dot{\mathbf{x}} = \mathbf{u}(\mathbf{x}), \quad (4.4)$$

find a Lipschitz continuous vector field controller, $\mathbf{u} : \mathcal{F} \rightarrow \mathbb{R}^n$, that leaves the robot's free space \mathcal{F} positively invariant and asymptotically steers almost all robot configurations in \mathcal{F} to any given goal location $\mathbf{x}^ \in \mathcal{F}$.*

4.2 Encoding Collisions via Separating Hyperplanes

In this section, we briefly recall a separating hyperplane theorem of disjoint convex sets, and then adapt it to identify a collision free neighborhood of a disk-shaped robot.

4.2.1 Separating Hyperplane Theorem

A fundamental theorem of convex sets states that any two disjoint convex sets can be separated by a hyperplane such that they lie on opposite sides of this hyperplane:

Theorem 4.1 (Separating Hyperplane Theorem [219, 37]) *For any two nonintersecting convex sets $A, B \in \mathbb{R}^n$ (i.e., $A \cap B = \emptyset$), there exists $\mathbf{a} \in \mathbb{R}^n$ and $b \in \mathbb{R}$ such that $\mathbf{a}^T \mathbf{x} \geq b$ for all $\mathbf{x} \in A$ and $\mathbf{a}^T \mathbf{x} \leq b$ for all $\mathbf{x} \in B$.*

For example, a usual choice of such a hyperplane is [37]:

Definition 4.1 *The maximum margin separating hyperplane of any two disjoint convex sets $A, B \subset \mathbb{R}^n$, with $d(A, B) > 0$, is defined to be*⁵

$$H(A, B) := \left\{ \mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x} - \mathbf{a}\| = \|\mathbf{x} - \mathbf{b}\|, \|\mathbf{a} - \mathbf{b}\| = d(A, B), \mathbf{a} \in \overline{A}, \mathbf{b} \in \overline{B} \right\}, \quad (4.5)$$

where $d(\mathbf{x}, H(A, B)) \geq \frac{d(A, B)}{2}$ for all $\mathbf{x} \in A \cup B$.

⁴Assumption 4.1 is equivalent to the “isolated” obstacles assumption of [183].

It is useful to remark that although there can be more than one pair of points $a \in \overline{A}$ and $b \in \overline{B}$ achieving $\|a - b\| = d(A, B)$, they all define the same maximum margin separating hyperplane (Lemma C.1).

Another useful tool for finding a separating hyperplane between a point and a convex set is metric projection:

Theorem 4.2 ([219]) *Let $A \subset \mathbb{R}^n$ be a closed convex set and $x \in \mathbb{R}^n$ be a point. Then there exists a unique point $a^* \in A$ such that*

$$a^* = \Pi_A(x) := \arg \min_{a \in A} \|a - x\|, \quad (4.6)$$

and one has $(x - \Pi_A(x))^T (\Pi_A(x) - a) \geq 0$ for all $a \in A$.

The map $\Pi_A(x)$ is called the metric projection of x onto the closed convex set A .

Note that the separating hyperplane $\{y \in \mathbb{R}^n \mid (y - \Pi_A(x))^T (\Pi_A(x) - x) = 0\}$ between point $x \in \mathbb{R}^n \setminus A$ and convex set A is referred to as the *support hyperplane* of A at $\Pi_A(x)$, which is tangent to the surface of A at $\Pi_A(x)$ [37].

It is also straightforward to observe that:

Lemma 4.1 *The maximum margin separating hyperplane of a convex set $A \subset \mathbb{R}^n$ and the ball $B(x, r)$ of radius $r \in \mathbb{R}_{>0}$ centered at $x \in \mathbb{R}^n$, satisfying $d(x, A) \geq r$, is given by*

$$H(A, B(x, r)) = \left\{ y \in \mathbb{R}^n \mid \left\| y - (\Pi_{\overline{B(x, r)}} \circ \Pi_{\overline{A}})(x) \right\| = \|y - \Pi_{\overline{A}}(x)\| \right\}, \quad (4.7)$$

where $(\Pi_{\overline{B(x, r)}} \circ \Pi_{\overline{A}})(x) = x - r \frac{x - \Pi_{\overline{A}}(x)}{\|x - \Pi_{\overline{A}}(x)\|}$.

Proof By definition (4.6), the metric projection $\Pi_{\overline{A}}(x)$ of the ball's centroid x onto the convex set \overline{A} is the unique closest point of \overline{A} to x . Hence, due to the symmetry of the ball, the closest point of $\overline{B(x, r)}$ to \overline{A} lies on the line segment joining x and $\Pi_{\overline{A}}(x)$, and is given by $(\Pi_{\overline{B(x, r)}} \circ \Pi_{\overline{A}})(x) = x - r \frac{x - \Pi_{\overline{A}}(x)}{\|x - \Pi_{\overline{A}}(x)\|}$, and so the closest point of \overline{A} to $\overline{B(x, r)}$ is $\Pi_{\overline{A}}(x)$. Thus, the result follows. \blacksquare

A common application of separating hyperplanes of a set of convex bodies is to discover their organizational structure. For instance, to model its topological structure, we define the generalized Voronoi diagrams $\mathcal{V} = \{V_1, V_2, \dots, V_m\}$ of a convex environment \mathcal{W} in \mathbb{R}^n populated with disjoint convex obstacles $\mathcal{O} = \{O_1, O_2, \dots, O_m\}$ (i.e., $d(O_i, O_j) > 0 \forall i \neq j$), based on maximum margin separating hyperplanes, to be ^{6 7}

$$V_i := \left\{ q \in \mathcal{W} \mid \|q - p_i\| \leq \|q - p_j\|, \|p_i - p_j\| = d(O_i, O_j), p_i \in \overline{O_i}, p_j \in \overline{O_j} \quad \forall j \neq i \right\}, \quad (4.8)$$

which yields a convex cell decomposition of a subset of \mathcal{W} such that, by construction, each obstacle is contained in its Voronoi cell, i.e., $O_i \subset V_i$, see Figure 4.2. Note that for point

⁵Note that one can equivalently represent the maximum margin separating hyperplane of any two distinct points $a \neq b \in \mathbb{R}^n$ as $\left\{ x \in \mathbb{R}^n \mid \|x - a\| = \|x - b\| \right\} = \left\{ x \in \mathbb{R}^n \mid (a - b)^T \left(x - \frac{a+b}{2} \right) = 0 \right\}$.

obstacles, say $O_i = \{p_i\}$ for some $p_i \in \mathbb{R}^n$, the generalized Voronoi diagram of \mathcal{W} in (4.8) simplifies back to the standard Voronoi diagram of \mathcal{W} , generated by points $\{p_1, \dots, p_m\}$, i.e., $V_i = \{q \in \mathcal{W} \mid \|q - p_i\| \leq \|q - p_j\|, \forall j \neq i\}$ [167].

4.2.2 The Safe Neighborhood of a Disk-Shaped Robot

Throughout the sequel, we consider a disk-shaped robot, centered at $x \in \mathcal{W}$ with radius $r \in \mathbb{R}_{>0}$, moving in a closed compact convex environment $\mathcal{W} \subseteq \mathbb{R}^n$ populated with open convex obstacles, $\mathcal{O} = \{O_1, O_2, \dots, O_m\}$, satisfying Assumption 4.1. Since the workspace, obstacles, and the robot radius are fixed, we suppress all mention of the associated terms wherever convenient, in order to simplify the notation.

Using the robot body and obstacles as generators of a generalized Voronoi diagram of \mathcal{W} , we define the robot's *local workspace*, $\mathcal{LW}(x)$, illustrated in Figure 4.2(left), as,⁸

$$\mathcal{LW}(x) := \left\{ q \in \mathcal{W} \mid \left\| q - x + r \frac{x - \Pi_{\overline{O}_i}(x)}{\|x - \Pi_{\overline{O}_i}(x)\|} \right\| \leq \|q - \Pi_{\overline{O}_i}(x)\|, \forall i \right\}. \quad (4.9)$$

Note that we here take the advantage of having a disk-shaped robot and construct the maximum margin separating hyperplane between the robot and each obstacle using the robot's centroid (Lemma 4.1), which will become more significant in the sequel when we extend this construction to a fixed radius sensory footprint and a limited range line-of-sight sensor.

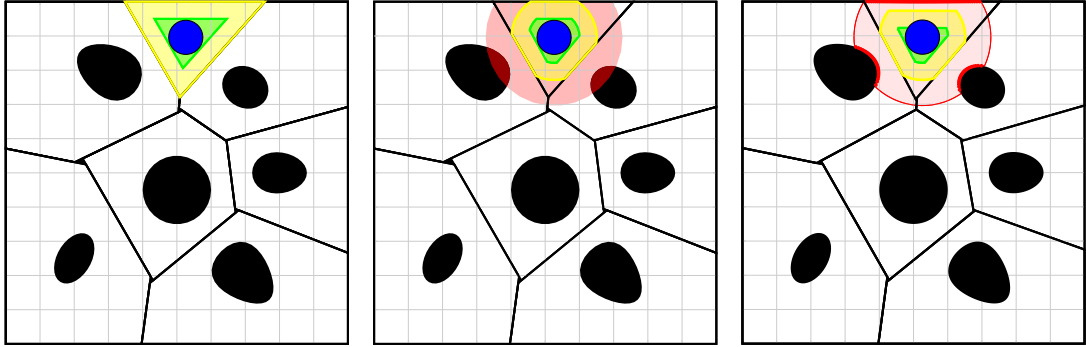


Figure 4.2: Local workspace \mathcal{LW} (yellow) and local free space \mathcal{LF} (green) of a disk-shaped robot (blue) for different sensing modalities: (left) Voronoi-adjacent¹² obstacle sensing, (middle) a fixed radius sensory footprint (red), (right) a limited range line-of-sight sensor (red). The boundary of each generalized Voronoi cell is defined by the maximum margin separating hyperplanes of the robot body (blue) and obstacles (black).

⁶Generalized Voronoi diagrams and cell decomposition methods are traditionally encountered in the design of roadmap methods [140, 165, 53]. A major distinction between our construction and these roadmap algorithms is that the latter typically seek a global, one-dimensional graphical representation of a robot's environment (independent of any specific configuration), whereas our approach uses the local open interior cells of the robot-centric Voronoi diagram to determine a locally safe neighborhood of a given free configuration.

⁷It seems worth noting that our use of generalized Voronoi diagrams is motivated by another application of Voronoi diagrams in robotics for coverage control of distributed mobile sensor networks [60, 137, 178, 12].

⁸Here, to solve the indeterminacy, we set $\frac{x}{\|x\|} = 0$ whenever $x = 0$.

A critical property of the local workspace \mathcal{LW} is:

Proposition 4.1 *A robot placement $\mathbf{x} \in \mathcal{W} \setminus \bigcup_{i=1}^m O_i$ is collision free, i.e., $\mathbf{x} \in \mathcal{F}$, if and only if the robot body is contained in its local workspace $\mathcal{LW}(\mathbf{x})$, i.e.,⁹*

$$\mathbf{x} \in \mathcal{F} \iff \overline{B(\mathbf{x}, r)} \subseteq \mathcal{LW}(\mathbf{x}) . \quad (4.10)$$

Proof To prove the result, it is convenient to rewrite (4.9), in terms of the intersection of the half-spaces defined by separating hyperplanes of the robot and the obstacles, as $\mathcal{LW}(\mathbf{x}) = \mathcal{W} \cap \bigcap_i HS_i$, where

$$HS_i := \left\{ \mathbf{q} \in \mathbb{R}^n \mid \left\| \mathbf{q} - \mathbf{x} + r \frac{\mathbf{x} - \Pi_{\overline{O_i}}(\mathbf{x})}{\|\mathbf{x} - \Pi_{\overline{O_i}}(\mathbf{x})\|} \right\| \leq \left\| \mathbf{q} - \Pi_{\overline{O_i}}(\mathbf{x}) \right\| \right\} . \quad (4.11)$$

Note that for any $\mathbf{x} \in \mathcal{F}$, HS_i is the half space defined by the maximum margin separating hyperplane between the robot body $\overline{B(\mathbf{x}, r)}$ and obstacle O_i (Lemma 4.1), and contains the robot. Moreover, since O_i is open, we have $O_i \cap HS_i = \emptyset$ for any $\mathbf{x} \in \mathcal{F}$. On the other hand, for any colliding configuration $\mathbf{x} \in \mathcal{W} \setminus (\mathcal{F} \cup \bigcup_{i=1}^m O_i)$, there is no separating hyperplane between the robot body and some obstacles, and so $\overline{B(\mathbf{x}, r)} \not\subseteq HS_i$ and $O_i \cap HS_i \neq \emptyset$ for some $i \in \{1, \dots, m\}$.

Hence, using (4.1), one can verify the result as follows: for any $\mathbf{x} \in \mathcal{W} \setminus \bigcup_{i=1}^m O_i$,

$$\mathbf{x} \in \mathcal{F} \iff \overline{B(\mathbf{x}, r)} \subseteq \mathcal{W} , \text{ and } \overline{B(\mathbf{x}, r)} \cap O_i = \emptyset \quad \forall i , \quad (4.12)$$

$$\iff \overline{B(\mathbf{x}, r)} \subseteq \mathcal{W} , \text{ and } \overline{B(\mathbf{x}, r)} \subseteq HS_i \quad \forall i , \quad (4.13)$$

$$\iff \overline{B(\mathbf{x}, r)} \subseteq \mathcal{LW}(\mathbf{x}) , \quad (4.14)$$

which completes the proof. ■

Accordingly, we define the robot's *local free space*, $\mathcal{LF}(\mathbf{x})$, by eroding $\mathcal{LW}(\mathbf{x})$, removing the volume swept along its boundary, $\partial\mathcal{LW}(\mathbf{x})$, by the robot body radius, illustrated on the left in Figure 4.2, as¹⁰ [103]

$$\mathcal{LF}(\mathbf{x}) := \mathcal{LW}(\mathbf{x}) \setminus (\partial\mathcal{LW}(\mathbf{x}) \oplus B(\mathbf{0}, r)) = \left\{ \mathbf{q} \in \mathcal{LW}(\mathbf{x}) \mid \overline{B(\mathbf{q}, r)} \subseteq \mathcal{LW}(\mathbf{x}) \right\} . \quad (4.15)$$

Note that, for any $\mathbf{x} \in \mathcal{F}$, $\mathcal{LF}(\mathbf{x})$ is a nonempty closed convex set, because $\mathbf{x} \in \mathcal{LF}(\mathbf{x})$ and the erosion of a closed convex set by an open ball is a closed convex set.¹¹

⁹Note that $\mathcal{F} \subsetneq \mathcal{W} \setminus \bigcup_{i=1}^m O_i$ for a disk-shaped robot of radius $r > 0$; and one can generalize the same result in (4.10) for any $\mathbf{x} \in \mathcal{W}$ if the robot's local workspace $\mathcal{LW}(\mathbf{x})$ is defined to be

$$\text{cl} \left(\left\{ \mathbf{q} \in \mathcal{W} \mid \left\| \mathbf{q} - \mathbf{x} + r \frac{\mathbf{x} - \Pi_{\overline{O_i}}(\mathbf{x})}{\|\mathbf{x} - \Pi_{\overline{O_i}}(\mathbf{x})\|} \right\| < \left\| \mathbf{q} - \Pi_{\overline{O_i}}(\mathbf{x}) \right\| , \quad \forall i \right\} \right) ,$$

which is empty whenever $\mathbf{x} \in O_i$ for some $i = 1, \dots, m$; otherwise, is equal to (4.9). Here, $\text{cl}(A)$ denotes the closure of a set A .

¹⁰Here, $\mathbf{0}$ is a vector of all zeros with the appropriate size, and $A \oplus B$ denotes the Minkowski sum of sets A and B defined as $A \oplus B = \{a + b \mid a \in A, b \in B\}$.

¹¹The erosion of a closed half-space by an open ball is a closed half-space. Hence, since the erosion operation is distributed over set intersection [103], and a closed convex set can be defined as (possibly infinite) intersection of closed half-spaces [37], and an arbitrary intersection of closed sets is closed [161], the erosion of a closed convex set by an open ball is a closed convex set.

An immediate consequence of Proposition 4.1 is:

Corollary 4.1 *Any robot placement in the local free space $\mathcal{LF}(x)$ of a collision free robot location $x \in \mathcal{F}$ is also collision free, i.e.,*

$$\mathcal{LF}(x) \subseteq \mathcal{F}, \quad \forall x \in \mathcal{F}. \quad (4.16)$$

Finally, it is useful to emphasize that to construct its local workspace, the robot requires only local knowledge of the environment in the sense that the robot only needs to locate proximal obstacles — those whose Voronoi cells are adjacent¹² to the robot’s (local workspace). This can be achieved by assuming an adjustable radius sensory footprint and gradually increasing its sensing range until the set of obstacles in the sensing range satisfies a certain geometric criterion guaranteeing that the detected obstacles exactly define the robot’s local workspace [60]. We will refer to this sensing model as *Voronoi-adjacent obstacle sensing*.

4.3 Robot Navigation via Separating Hyperplanes

In this section, first assuming Voronoi-adjacent obstacle sensing, we introduce a new provably correct vector field controller for safe robot navigation in a convex sphere world, and list its important qualitative properties. Then we present its extensions for two more realistic sensor models (illustrated, respectively, in the middle and the right panels of Figure 4.2): a fixed radius sensory footprint and a limited range line-of-sight sensor. We further adapt our construction to the widely used nonholonomically constrained differential drive vehicle.

4.3.1 Feedback Robot Motion Planner

Assuming the fully-actuated single-integrator robot dynamics in (4.4), for a choice of a desired goal location $x^* \in \mathcal{F}$, we propose a robot navigation strategy, called the “*move-to-projected-goal*” law, $u : \mathcal{F} \rightarrow \mathbb{R}^n$ that steers the robot at location $x \in \mathcal{F}$ towards the global goal x^* through the “*projected goal*”, $\Pi_{\mathcal{LF}(x)}(x^*)$, as follows:¹³

$$u(x) = -k(x - \Pi_{\mathcal{LF}(x)}(x^*)) , \quad (4.17)$$

where $k \in \mathbb{R}_{>0}$ is a fixed control gain and Π_A (4.6) is the metric projection onto a closed convex set $A \subset \mathbb{R}^n$, and $\mathcal{LF}(x)$ is continuously updated using the Voronoi-adjacent obstacle sensing and its relation with $\mathcal{LW}(x)$ in (4.15).

¹² A pair of Voronoi cells in \mathbb{R}^n is said to be *adjacent* if they share a $n - 1$ dimensional face.

¹³In general, the metric projection of a point onto a convex set can be efficiently computed using a standard convex programming solver [37]. If W is a convex polytope, then the robot’s local free space, $\mathcal{LF}(x)$, is also a convex polytope and can be written as a finite intersection of half-spaces. Hence, the metric projection onto a convex polytope can be recast as quadratic programming and can be solved in polynomial time [133]. In the case of a convex polygonal environment, $\mathcal{LF}(x)$ is a convex polygon and the metric projection onto a convex polygon can be solved analytically because the solution lies on one of its edges, unless the input point is inside the polygon.

4.3.2 Qualitative Properties

We now continue with a list of its qualitative (continuity, existence & uniqueness, invariance and stability) properties.

Proposition 4.2 *The “move-to-projected-goal” law in (4.17) is piecewise continuously differentiable.*

Proof An important property of generalized Voronoi diagrams in (4.8) inherited from the standard Voronoi diagrams of point generators is that the boundary of each Voronoi cell is a piecewise continuously differentiable function of generator locations [44, 186]. In particular, for any $x \in \mathcal{F}$, the boundary of the robot’s local workspace $\mathcal{LW}(x)$ is piecewise continuously differentiable, because it is defined by the boundary of the workspace and separating hyperplanes between the robot and the obstacles, parametrized by x and $\Pi_{\overline{O}_i}(x)$, and metric projections onto convex cells are piecewise continuously differentiable [135]. Hence, the boundary of the local free space $\mathcal{LF}(x)$ is also piecewise continuously differentiable, because $\mathcal{LF}(x)$ is the nonempty erosion of $\mathcal{LW}(x)$ by a fixed open ball. Therefore, one can conclude using the sensitivity analysis of metric projections onto moving convex sets [199, 146] that the “move-to-projected-goal” law is Lipschitz continuous and piecewise continuously differentiable. ■

Proposition 4.3 *The robot’s free space \mathcal{F} in (4.1) is positively invariant under the “move-to-projected” law (4.17).*

Proof Since x and $\Pi_{\mathcal{LF}(x)}(x^*)$ are both in $\mathcal{LF}(x)$ for any $x \in \mathcal{F}$, and $\mathcal{LF}(x)$ is an obstacle free convex neighborhood of x (Corollary 4.1), the line segment joining x and $\Pi_{\mathcal{LF}(x)}(x^*)$ is free of collisions. Hence, at the boundary of \mathcal{F} , the robot under the “move-to-projected-goal” law either stays on the boundary or moves towards the interior of \mathcal{F} , but never crosses the boundary, and so the result follows. ■

Proposition 4.4 *For any initial $x \in \mathcal{F}$, the “move-to-projected-goal” law (4.17) has a unique continuously differentiable flow in \mathcal{F} (4.1) defined for all future time.*

Proof The existence, uniqueness and continuous differentiability of its flow follow from the Lipschitz continuity of the “move-to-projected-goal” law in its compact domain \mathcal{F} , because a piecewise continuously differentiable function is locally Lipschitz on its domain [50], and a locally Lipschitz function on a compact set is globally Lipschitz on that set [127]. ■

Proposition 4.5 *The set of stationary points of the “move-to-projected-goal” law (4.17) is $\{x^*\} \cup \bigcup_{i=1}^m \mathfrak{S}_i$, where*

$$\mathfrak{S}_i := \left\{ x \in \mathcal{F} \left| d(x, O_i) = r, \frac{(x - \Pi_{\overline{O}_i}(x))^T (x - x^*)}{\|x - \Pi_{\overline{O}_i}(x)\| \|x - x^*\|} = 1 \right. \right\}. \quad (4.18)$$

Proof It follows from (4.6) that the goal location x^* is a stationary point of (4.17), because $x^* \in \mathcal{LF}(x^*)$. In fact, for any $x \in \mathcal{F}$, one has $\Pi_{\mathcal{LF}(x)}(x^*) = x^*$ whenever $x^* \in \mathcal{LF}(x)$. Hence, in the sequel of the proof, we only consider the set of robot locations satisfying $x^* \notin \mathcal{LF}(x)$.

Let $x \in \mathcal{F}$ such that $x^* \notin \mathcal{LF}(x)$. Recall from (4.9) and (4.15) that $\mathcal{LW}(x)$ is determined by the maximum margin separating hyperplanes of the robot body and the obstacles, and $\mathcal{LF}(x)$ is obtained by eroding $\mathcal{LW}(x)$ by an open ball of radius r . Hence, x lies in the interior of $\mathcal{LF}(x)$ if and only if $d(x, O_i) > r$ for all i . As a result, since $x^* \notin \mathcal{LF}(x)$, one has $x = \Pi_{\mathcal{LF}(x)}(x^*)$ only if $d(x, O_i) = r$ for some i .

Note that if $d(x, O_i) = r$, then, since $d(O_i, O_j) > 2r$ (Assumption 4.1), $d(x, O_j) > r$ for all $j \neq i$. Therefore, there can be only one obstacle index i such that $x = \Pi_{\mathcal{LW}(x)}(x^*)$ and $d(x, O_i) = r$. Further, given $d(x, O_i) = r$, since $\Pi_{\mathcal{LF}(x)}(x^*)$ is the unique closest point of the closed convex set $\mathcal{LF}(x)$ to the goal x^* (Theorem 4.2), its optimality [37] implies that one has $x = \Pi_{\mathcal{LW}(x)}(x^*)$ if and only if the maximum margin separating hyperplane between the robot and obstacle O_i is tangent to the level curve of the squared Euclidean distance to the goal, $\|x - x^*\|^2$, at $\Pi_{\overline{O}_i}(x)$, and separates x and x^* , i.e.,

$$\frac{(x - \Pi_{\overline{O}_i}(x))^T (x - x^*)}{\|x - \Pi_{\overline{O}_i}(x)\| \|x - x^*\|} = 1. \quad (4.19)$$

Thus, one can locate the stationary points of the “move-to-projected-goal” law in (4.17) associated with obstacle O_i as in (4.18), and so the result follows. ■

Remark 4.1. For any equilibrium point $s_i \in \mathfrak{S}_i$ in (4.18) associated with obstacle O_i , one has that the equilibrium s_i , its projection $\Pi_{\overline{O}_i}(s_i)$ and the goal x^* are all collinear.

Lemma 4.2 *The “move-to-projected-goal” law (4.17) in a small neighborhood of the goal x^* is given by*

$$u(x) = -k(x - x^*), \quad \forall x \in B(x^*, \epsilon), \quad (4.20)$$

for some $\epsilon > 0$; and around any stationary point $s_i \in \mathfrak{S}_i$ (4.18), associated with obstacle O_i , it is given by

$$u(x) = -k \left(x - x^* + \frac{(x - \Pi_{\overline{O}_i}(x))^T (x^* - h_i)}{\|x - \Pi_{\overline{O}_i}(x)\|^2} (x - \Pi_{\overline{O}_i}(x)) \right), \quad (4.21)$$

for all $x \in B(s_i, \varepsilon)$ and some $\varepsilon > 0$, where

$$h_i := \frac{x + \Pi_{\overline{O}_i}(x)}{2} + \frac{r}{2} \frac{x - \Pi_{\overline{O}_i}(x)}{\|x - \Pi_{\overline{O}_i}(x)\|}. \quad (4.22)$$

Proof The result for the goal location x^* follows from the continuity of Voronoi diagrams in (4.8) and $x^* \in \mathcal{LF}(x^*)$.

To see the result for any stationary point $s_i \in \mathfrak{S}_i$, recall from the proof of Proposition 4.5 that s_i lies on the boundary segment of $\mathcal{LF}(s_i)$ defined by the separating hyperplane between the robot and i th obstacle, and s_i has a certain nonzero clearance from the boundary segment of $\mathcal{LF}(s_i)$ defined by the separating hyperplane between the robot and any other obstacle. Hence, using the continuity of Voronoi diagrams, for any $x \in B(s_i, \varepsilon)$, the “projected-goal” $\Pi_{\mathcal{LF}(x)}(x^*)$ can be located by taking the projection of x^* onto (a shifted version of) the

maximum margin separating hyperplane between the robot and obstacle O_i as

$$\Pi_{\mathcal{LF}(x)}(x^*) = x^* - \frac{\left(x - \Pi_{\overline{O}_i}(x)\right)^T (x^* - h_i)}{\left\|x - \Pi_{\overline{O}_i}(x)\right\|^2} \left(x - \Pi_{\overline{O}_i}(x)\right), \quad (4.23)$$

where h_i is defined as in (4.22), and this completes the proof. \blacksquare

Since our “move-to-projected-goal” law strictly decreases the (squared) Euclidean distance to the goal x^* away from its stationary points (Proposition 4.7), to guarantee the existence of a unique stable attractor at x^* we require the following assumption¹⁴, whose geometric interpretation is discussed in detail in Appendix C.1.

Assumption 4.2 (*Obstacle Curvature Condition*) *The Jacobian matrix $\mathbf{J}_{\Pi_{\overline{O}_i}}(s_i)$ of the metric projection of any stationary point $s_i \in \mathfrak{S}_i$ onto the associated obstacle O_i satisfies*¹⁵

$$\mathbf{J}_{\Pi_{\overline{O}_i}}(s_i) \prec \frac{\left\|x^* - \Pi_{\overline{O}_i}(s_i)\right\|}{r + \left\|x^* - \Pi_{\overline{O}_i}(s_i)\right\|} \mathbf{I} \quad \forall i, \quad (4.24)$$

where \mathbf{I} is the identity matrix of appropriate size.

Proposition 4.6 *If Assumption 4.2 holds for the goal x^* and for all obstacles, then x^* is the only locally stable equilibrium of the “move-to-projected-goal” law (4.17), and all the stationary points, $s_i \in \mathfrak{S}_i$ (4.18), associated with obstacles, O_i , are nondegenerate saddles.*

Proof It follows from (4.20) that the goal x^* is a locally stable point of the “move-to-projected-goal” law, because its Jacobian matrix, $\mathbf{J}_u(x^*)$, at x^* is equal to $-k\mathbf{I}$.

Now, to determine the type of any stationary point $s_i \in \mathfrak{S}_i$ associated with obstacle O_i , define

$$g(x) := \frac{\left(x^* - \Pi_{\overline{O}_i}(x)\right)^T \left(x - \Pi_{\overline{O}_i}(x)\right)}{\left\|x - \Pi_{\overline{O}_i}(x)\right\|^2} - \frac{r}{2\left\|x - \Pi_{\overline{O}_i}(x)\right\|} - \frac{1}{2}, \quad (4.25)$$

and so the “move-to-projected-goal” law in a small neighborhood of s_i in (4.21) can be rewritten as

$$\dot{u}(x) = -k \left(x - x^* + g(x) \left(x - \Pi_{\overline{O}_i}(x) \right) \right). \quad (4.26)$$

Hence, using $\left\|s_i - \Pi_{\overline{O}_i}(s_i)\right\| = r$, one can verify that its Jacobian matrix at s_i is given by

$$\mathbf{J}_u(s_i) = -kg(s_i) \left(\frac{\left\|x^* - \Pi_{\overline{O}_i}(s_i)\right\|}{r + \left\|x^* - \Pi_{\overline{O}_i}(s_i)\right\|} \mathbf{Q} - \mathbf{J}_{\Pi_{\overline{O}_i}}(s_i) \right) - \frac{k}{2} (\mathbf{I} - \mathbf{Q}), \quad (4.27)$$

¹⁴A similar obstacle curvature condition is necessarily made in the design of navigation functions for spaces with convex obstacles in [174].

¹⁵For any two symmetric matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{N \times N}$, $\mathbf{A} \prec \mathbf{B}$ (and $\mathbf{A} \preceq \mathbf{B}$) means that $\mathbf{B} - \mathbf{A}$ is positive definite (positive semidefinite, respectively).

where $g(s_i) = -\frac{\|x^* - \Pi_{\overline{O}_i}(s_i)\|}{r} - 1 < -2$, and

$$\mathbf{Q} = \mathbf{I} - \frac{\left(s_i - \Pi_{\overline{O}_i}(s_i)\right) \left(s_i - \Pi_{\overline{O}_i}(s_i)\right)^T}{\left\|s_i - \Pi_{\overline{O}_i}(s_i)\right\|^2}. \quad (4.28)$$

Note that $\mathbf{J}_{\Pi_{\overline{O}_i}}(x) \left(x - \Pi_{\overline{O}_i}(x)\right) = 0$ for all $x \in \mathbb{R}^n \setminus \overline{O}_i$ [109, 85]. Hence, if Assumption 4.2 holds, then one can conclude from $g(s_i) < -2$ and (4.27) that the only negative eigenvalue of $\mathbf{J}_u(s_i)$ and the associated eigenvector are $-\frac{k}{2}$ and $\left(s_i - \Pi_{\overline{O}_i}(s_i)\right)$, respectively; and all other eigenvalues of $\mathbf{J}_u(s_i)$ are positive. Thus, s_i is a nondegenerate saddle point of the “move-to-projected-goal” law associated with O_i . ■

Proposition 4.7 *Given that the goal location x^* and the obstacles satisfy Assumption 4.2, the goal x^* is an asymptotically stable equilibrium of the “move-to-projected-goal” law (4.17), whose basin of attraction includes \mathcal{F} , except a set of measure zero.*

Proof Consider the squared Euclidean distance to the goal as a smooth Lyapunov function candidate, i.e., $V(x) := \|x - x^*\|^2$, and it follows from (4.6) and (4.17) that

$$\begin{aligned} \dot{V}(x) &= -k \underbrace{2(x - x^*)^T (x - \Pi_{\mathcal{L}\mathcal{F}(x)}(x^*))}_{\geq \|x - \Pi_{\mathcal{L}\mathcal{F}(x)}(x^*)\|^2} \leq -k \|x - \Pi_{\mathcal{L}\mathcal{F}(x)}(x^*)\|^2 \leq 0, \quad (4.29) \\ &\text{since } x \in \mathcal{L}\mathcal{F}(x) \text{ and } \|x - x^*\|^2 \geq \|\Pi_{\mathcal{L}\mathcal{F}(x)}(x^*) - x^*\|^2 \end{aligned}$$

which is zero iff x is a stationary point. Hence, we have from LaSalle’s Invariance Principle [127] that all robot configurations in \mathcal{F} asymptotically reach the set of equilibria of (4.17). Therefore, the result follows from Proposition 4.2 and Proposition 4.6, because, under Assumption 4.2, x^* is the only stable stationary point of the piecewise continuous “move-to-projected-goal” law (4.17), and all other stationary points are nondegenerate saddles whose stable manifolds have empty interiors [108]. ■

Finally, we find it useful to summarize important qualitative properties of the “move-to-projected-goal” law as:¹⁶

Theorem 4.3 *The piecewise continuously differentiable “move-to-projected-goal” law in (4.17) leaves the robot’s free space \mathcal{F} (4.1) positively invariant; and if Assumption 4.2 holds, then its unique continuously differentiable flow, starting at almost any configuration $x \in \mathcal{F}$, asymptotically reaches the goal location x^* , while strictly decreasing the squared Euclidean distance to the goal, $\|x - x^*\|^2$, along the way.*

4.3.3 Extensions for Limited Range Sensing Modalities

Navigation using a Fixed Radius Sensory Footprint

A crucial property of the “move-to-projected-goal” law (4.17) is that it only requires the knowledge of the robot’s Voronoi-adjacent¹² obstacles to determine the robot’s lo-

¹⁶Since the “move-to-projected-goal” law is piecewise continuously differentiable, it can be lifted to higher order dynamical models [130, 131, 82].

cal workspace and so the robot's local free space. We now exploit that property to relax our construction so that it can be put to practical use with commonly available sensors that have bounded radius footprint.¹⁷ We will present two specific instances, pointing out along the way how they nevertheless preserve the sufficient conditions for the qualitative properties listed in Section 4.3.2.

Suppose the robot is equipped with a sensor with a fixed sensing range, $R \in \mathbb{R}_{>0}$, whose sensory output, denoted by $\mathcal{S}_R(\mathbf{x}) := \{S_1, S_2, \dots, S_m\}$, at a location, $\mathbf{x} \in \mathcal{W}$, returns some computationally effective dense representation of the perceptible portion, $S_i := O_i \cap B(\mathbf{x}, R)$, of each obstacle, O_i , in its sensory footprint, $B(\mathbf{x}, R)$. Note that S_i is always open and might possibly be empty (if O_i is outside the robot's sensing range), see Figure 4.2(middle); and we assume that the robot's sensing range is greater than the robot body radius, i.e., $R > r$.

As in (4.9), using the maximum margin separating hyperplanes of the robot and sensed obstacles, we define the robot's *sensed local workspace*, illustrated in Figure 4.2(middle), as,

$$\mathcal{LW}_s(\mathbf{x}) := \left\{ \mathbf{q} \in \mathcal{W} \cap \overline{B\left(\mathbf{x}, \frac{r+R}{2}\right)} \mid \left\| \mathbf{q} - \mathbf{x} + r \frac{\mathbf{x} - \Pi_{\overline{S_i}}(\mathbf{x})}{\|\mathbf{x} - \Pi_{\overline{S_i}}(\mathbf{x})\|} \right\| \leq \left\| \mathbf{q} - \Pi_{\overline{S_i}}(\mathbf{x}) \right\|, \forall i \text{ s.t. } S_i \neq \emptyset \right\}. \quad (4.30)$$

Note that $\overline{B\left(\mathbf{x}, \frac{r+R}{2}\right)}$ is equal to the intersection of the closed half-spaces containing the robot body and defined by the maximum margin separating hyperplanes of the robot body, $\overline{B(\mathbf{x}, r)}$, and all individual points, $\mathbf{q} \in \mathbb{R}^n \setminus B(\mathbf{x}, R)$, outside its sensory footprint.¹⁸

An important observation revealing a critical connection between the robot's local workspace \mathcal{LW} in (4.9) and its sensed local workspace \mathcal{LW}_s in (4.30) is:

Proposition 4.8 $\mathcal{LW}_s(\mathbf{x}) = \mathcal{LW}(\mathbf{x}) \cap \overline{B\left(\mathbf{x}, \frac{r+R}{2}\right)}$ for all $\mathbf{x} \in \mathcal{W}$.

Proof As discussed in the proof of Proposition 4.1, for any $\mathbf{x} \in \mathcal{W}$, we have $\mathcal{LW}(\mathbf{x}) = \mathcal{W} \cap \bigcap_i HS_i$, where HS_i is defined as in (4.11). Similarly, one can rewrite (4.30) as $\mathcal{LW}_s(\mathbf{x}) = \mathcal{W} \cap \overline{B\left(\mathbf{x}, \frac{r+R}{2}\right)} \cap \bigcap_i \widehat{HS_i}$, where

$$\widehat{HS_i} := \left\{ \mathbf{q} \in \mathbb{R}^n \mid \left\| \mathbf{q} - \mathbf{x} + r \frac{\mathbf{x} - \Pi_{\overline{S_i}}(\mathbf{x})}{\|\mathbf{x} - \Pi_{\overline{S_i}}(\mathbf{x})\|} \right\| \leq \left\| \mathbf{q} - \Pi_{\overline{S_i}}(\mathbf{x}) \right\| \right\}. \quad (4.31)$$

Note that if $S_i = \emptyset$, then the predicate in (4.31) is trivially holds and so $\widehat{HS_i} = \mathbb{R}^n$; otherwise, since $S_i = O_i \cap B(\mathbf{x}, R)$, we have $\Pi_{\overline{S_i}}(\mathbf{x}) = \Pi_{\overline{O_i}}(\mathbf{x})$ and so $\widehat{HS_i} = HS_i$. Moreover, if $S_i = \emptyset$ (i.e., $d(\mathbf{x}, O_i) > R$), then we also have from Definition 4.1 and Lemma 4.1 that $\overline{B\left(\mathbf{x}, \frac{r+R}{2}\right)} \subset HS_i$. Thus, we obtain that

$$\widehat{HS_i} \cap \overline{B\left(\mathbf{x}, \frac{r+R}{2}\right)} = HS_i \cap \overline{B\left(\mathbf{x}, \frac{r+R}{2}\right)}, \quad \forall i. \quad (4.32)$$

¹⁷ This extension results from the construction of the robot's local workspace (4.9) in terms of the maximum margin separating hyperplanes of convex sets. In consequence, because the intersection of convex sets is a convex set [37], perceived obstacles in the robot's (convex) sensory footprint are, in turn, themselves always convex.

¹⁸ Since $d(\mathbf{q}, \overline{B(\mathbf{x}, r)}) \geq R - r$ for any $\mathbf{q} \in \mathbb{R}^n \setminus B(\mathbf{x}, R)$, it follows from Theorem 4.1 that the distance of $\overline{B(\mathbf{x}, r)}$ from the associated maximum margin separating hyperplane $H(\mathbf{q}, \overline{B(\mathbf{x}, r)})$ (defined in (4.5) and (4.7)), between the robot body and point \mathbf{q} , is at least $\frac{R-r}{2}$, i.e., $d(B(\mathbf{x}, r), H(\mathbf{q}, \overline{B(\mathbf{x}, r)})) \geq \frac{R-r}{2}$, and so the robot's centroidal distance to the separating hyperplane is $\frac{R+r}{2}$.

Therefore, one can verify the result as follows:

$$\mathcal{LW}_s(x) = \mathcal{W} \cap \overline{B(x, \frac{r+R}{2})} \cap \bigcap_i \widehat{HS}_i = \mathcal{W} \cap \bigcap_i \left(\widehat{HS}_i \cap \overline{B(x, \frac{r+R}{2})} \right), \quad (4.33)$$

$$= \mathcal{W} \cap \bigcap_i \left(HS_i \cap \overline{B(x, \frac{r+R}{2})} \right) = \left(\mathcal{W} \cap \bigcap_i HS_i \right) \cap \overline{B(x, \frac{r+R}{2})}, \quad (4.34)$$

$$= \mathcal{LW}(x) \cap \overline{B(x, \frac{r+R}{2})}. \quad (4.35)$$

■

In accordance with its local free space $\mathcal{LF}(x)$ in (4.15), we define the robot's *sensed local free space* $\mathcal{LF}_s(x)$ by eroding $\mathcal{LW}_s(x)$ by the robot body, illustrated in Figure 4.2(middle), as,

$$\mathcal{LF}_s(x) := \left\{ q \in \mathcal{LW}_s(x) \mid \overline{B(q, r)} \subseteq \mathcal{LW}_s(x) \right\} = \mathcal{LF}(x) \cap \overline{B(x, \frac{R-r}{2})}, \quad (4.36)$$

where the latter follows from Proposition 4.8 and that the erosion operation is distributed over set intersection [103]. Note that, for any $x \in \mathcal{F}$, $\mathcal{LF}_s(x)$ is a nonempty closed convex set containing x as is $\mathcal{LF}(x)$.

To safely steer a single-integrator disk-shaped robot towards a given goal location $x^* \in \mathcal{F}$ using a fixed radius sensory foot-print, we propose the following “move-to-projected-goal” law,

$$u(x) = -k(x - \Pi_{\mathcal{LF}_s(x)}(x^*)), \quad (4.37)$$

where $k > 0$ is a fixed control gain, and $\Pi_{\mathcal{LF}_s(x)}$ (4.6) is the metric projection onto the robot's sensed local free space $\mathcal{LF}_s(x)$, and $\mathcal{LF}_s(x)$ is assumed to be continuously updated.

Due to the nice relations between the robot's different local neighborhoods in Proposition 4.8 and (4.36), the revised “move-to-projected-goal” law for a fixed radius sensory footprint inherits all qualitative properties of the original one presented in Section 4.3.2, summarized as:

Proposition 4.9 *The “move-to-projected-goal” law of a disk-shaped robot equipped with a fixed radius sensory footprint in (4.37) is piecewise continuously differentiable; and if Assumption 4.2 holds, then its unique continuously differentiable flow asymptotically steers almost all configurations in its positively invariant domain \mathcal{F} towards any given goal location $x^* \in \mathcal{F}$, while strictly decreasing the (squared) Euclidean distance to the goal along the way.*

Proof The proof of the result follows patterns similar to those of Proposition 4.2 - Proposition 4.7, because of the relations between the robot's local neighborhoods in Proposition 4.8 and (4.36), and so it is omitted for the sake of brevity. ■

Navigation using a 2D LIDAR Range Scanner

We now present another practical extension of the “move-to-projected-goal” law for safe robot navigation using a 2D LIDAR range scanner in an unknown convex planar environment $\mathcal{W} \subseteq \mathbb{R}^2$ populated with convex obstacles $\mathcal{O} = \{O_1, O_2, \dots, O_m\}$, satisfying Assumption 4.1. Assuming an angular scanning range of 360 degrees and a fixed radial range of

$R \in \mathbb{R}_{>0}$, we model the sensory measurement of the LIDAR scanner at location $\mathbf{x} \in \mathcal{W}$ by a polar curve [206] $\rho_{\mathbf{x}} : (-\pi, \pi] \rightarrow [0, R]$, defined as,

$$\rho_{\mathbf{x}}(\theta) := \min \left(\begin{array}{c} R, \\ \min \left\{ \|\mathbf{p} - \mathbf{x}\| \mid \mathbf{p} \in \partial\mathcal{W}, \text{atan2}(\mathbf{p} - \mathbf{x}) = \theta \right\}, \\ \min_i \left\{ \|\mathbf{p} - \mathbf{x}\| \mid \mathbf{p} \in O_i, \text{atan2}(\mathbf{p} - \mathbf{x}) = \theta \right\} \end{array} \right). \quad (4.38)$$

We further assume that the LIDAR sensing range is greater than the robot body radius, i.e., $R > r$.

Suppose $\rho_i : (\theta_{l_i}, \theta_{u_i}) \rightarrow [0, R]$ is a convex curve segment of the LIDAR scan $\rho_{\mathbf{x}}$ (4.38) at location $\mathbf{x} \in \mathcal{W}$ (please refer to Appendix C.4 for the notion of convexity in polar coordinates which we use to identify convex polar curve segments in a LIDAR scan, corresponding to the obstacle and workspace boundary), then we define the associated *line-of-sight obstacle* as the open epigraph of ρ_i whose pole is located at \mathbf{x} [206],^{10 19}

$$L_i := \{\mathbf{x}\} \oplus \text{epi} \rho_i = \{\mathbf{x}\} \oplus \left\{ (\varrho \cos \theta, \varrho \sin \theta) \mid \theta \in (\theta_{l_i}, \theta_{u_i}), \varrho > \rho_i(\theta) \right\}, \quad (4.39)$$

which is an open convex set. Accordingly, we assume the availability of a sensor model $\mathcal{L}_R(\mathbf{x}) := \{L_1, L_2, \dots, L_t\}$ that returns the list of convex line-of-sight obstacles detected by the LIDAR scanner at location \mathbf{x} , where t denotes the number of detected obstacles and changes as a function of robot location.

Following the lines of (4.9) and (4.15), we define the robot's *line-of-sight local workspace* and *line-of-sight local free space*, illustrated in Figure 4.2(right), respectively, as

$$\mathcal{LW}_{\mathcal{L}}(\mathbf{x}) := \left\{ \mathbf{q} \in L_{ft}(\mathbf{x}) \cap \overline{B\left(\mathbf{x}, \frac{r+R}{2}\right)} \mid \left\| \mathbf{q} - \mathbf{x} + r \frac{\mathbf{x} - \Pi_{\overline{L_i}}(\mathbf{x})}{\|\mathbf{x} - \Pi_{\overline{L_i}}(\mathbf{x})\|} \right\| \leq \left\| \mathbf{q} - \Pi_{\overline{L_i}}(\mathbf{x}) \right\|, \forall i \right\}. \quad (4.40)$$

$$\mathcal{LF}_{\mathcal{L}}(\mathbf{x}) := \left\{ \mathbf{q} \in \mathcal{LW}_{\mathcal{L}}(\mathbf{x}) \mid \overline{B(\mathbf{q}, r)} \subseteq \mathcal{LW}_{\mathcal{L}}(\mathbf{x}) \right\}, \quad (4.41)$$

where $L_{ft}(\mathbf{x})$ denotes the LIDAR sensory footprint at \mathbf{x} , given by the hypograph of the LIDAR scan $\rho_{\mathbf{x}}$ (4.38) at \mathbf{x} , i.e.,

$$L_{ft}(\mathbf{x}) := \{\mathbf{x}\} \oplus \text{hyp} \rho_{\mathbf{x}} = \{\mathbf{x}\} \oplus \left\{ (\varrho \cos \theta, \varrho \sin \theta) \mid \theta \in (-\pi, \pi], 0 \leq \varrho \leq \rho_{\mathbf{x}}(\theta) \right\}. \quad (4.42)$$

Similar to Proposition 4.1 and Corollary 4.1, we have:

Proposition 4.10 *For any $\mathbf{x} \in \mathcal{F}$, $\mathcal{LW}_{\mathcal{L}}(\mathbf{x})$ is an obstacle free closed convex subset of \mathcal{W} and contains the robot body $B(\mathbf{x}, r)$. Therefore, $\mathcal{LF}_{\mathcal{L}}(\mathbf{x})$ is a nonempty closed convex subset of \mathcal{F} and contains \mathbf{x} .*

Proof For any $\mathbf{x} \in \mathcal{F}$, the LIDAR sensory footprint in (4.42) can be equivalently rewritten using the global knowledge of the robot's workspace as

$$L_{ft}(\mathbf{x}) = \mathcal{W} \cap \overline{B(\mathbf{x}, R)} \setminus \bigcup_i A_i. \quad (4.43)$$

¹⁹Here, $\overset{\circ}{A}$ denotes the interior of a set A .

where A_i is the augmented line-of-sight obstacle associated with obstacle O_i , defined as

$$A_i := \left\{ \alpha (\mathbf{p} - \mathbf{x}) + \mathbf{p} \mid \mathbf{p} \in O_i, \alpha \in [0, \infty) \right\} . \quad (4.44)$$

Hence, since $R > r$, it follows from (4.40) that

$$\mathcal{LW}_{\mathcal{L}}(\mathbf{x}) = \widehat{\mathcal{LW}}_{\mathcal{L}}(\mathbf{x}) \setminus \bigcup_i A_i , \quad (4.45)$$

where

$$\widehat{\mathcal{LW}}_{\mathcal{L}}(\mathbf{x}) := \left\{ \mathbf{q} \in \mathcal{W} \cap \overline{B(\mathbf{x}, \frac{r+R}{2})} \mid \left\| \mathbf{q} - \mathbf{x} + r \frac{\mathbf{x} - \Pi_{\mathcal{L}_i}(\mathbf{x})}{\|\mathbf{x} - \Pi_{\mathcal{L}_i}(\mathbf{x})\|} \right\| \leq \|\mathbf{q} - \Pi_{\mathcal{L}_i}(\mathbf{x})\|, \forall i \right\} . \quad (4.46)$$

Note that, as discussed in the proof of Proposition 4.1, since $\mathbf{x} \in \mathcal{F}$, $\widehat{\mathcal{LW}}_{\mathcal{L}}(\mathbf{x})$ is a closed convex set and free of any line-of-sight obstacle L_i , i.e., $\widehat{\mathcal{LW}}_{\mathcal{L}}(\mathbf{x}) \cap L_i = \emptyset$ for all i ; and it contains the robot body, i.e., $\overline{B(\mathbf{x}, r)} \subseteq \widehat{\mathcal{LW}}_{\mathcal{L}}(\mathbf{x})$.

Now observe that if obstacle O_i is in the LIDAR's sensing range, i.e, $O_i \cap B(\mathbf{x}, R) \neq \emptyset$, then $A_i \cap B(\mathbf{x}, R) = L_j \cap B(\mathbf{x}, R)$ for some j . Hence, since $\widehat{\mathcal{LW}}_{\mathcal{L}}(\mathbf{x})$ is free of line-of-sight obstacles, we have from (4.45) that $\mathcal{LW}_{\mathcal{L}}(\mathbf{x}) = \widehat{\mathcal{LW}}_{\mathcal{L}}(\mathbf{x})$. Recall that $\widehat{\mathcal{LW}}_{\mathcal{L}}(\mathbf{x})$ is an obstacle-free closed convex set, and so is $\mathcal{LW}_{\mathcal{L}}(\mathbf{x})$. Thus, the result follows because $\mathcal{LF}_{\mathcal{L}}(\mathbf{x})$ is the erosion of $\mathcal{LW}_{\mathcal{L}}(\mathbf{x})$ by the robot body radius r . ■

Accordingly, to navigate a fully-actuated single-integrator robot using a LIDAR scanner towards a desired goal location $\mathbf{x}^* \in \mathcal{F}$, with the guarantee of no collisions along the way, we propose the following “move-to-projected-goal” law

$$\mathbf{u}(\mathbf{x}) = -k(\mathbf{x} - \Pi_{\mathcal{LF}_{\mathcal{L}}(\mathbf{x})}(\mathbf{x}^*)) , \quad (4.47)$$

where $k > 0$ is fixed, and $\Pi_{\mathcal{LF}_{\mathcal{L}}(\mathbf{x})}$ (4.6) is the metric projection onto the robot's line-of-sight free space $\mathcal{LF}_{\mathcal{L}}(\mathbf{x})$ (4.41), which is assumed to be continuously updated.

We summarize important properties of the “move-to-projected-goal” law for navigation using a LIDAR scanner as:

Proposition 4.11 *The “move-to-projected-goal” law of a LIDAR-equipped disk-shaped robot in (4.47) leaves the robot's free space \mathcal{F} (4.1) positively invariant; and if Assumption 4.2 holds, then its unique, continuous and piecewise differentiable flow asymptotically brings all but a measure zero set of initial configurations in \mathcal{F} to any designated goal location $\mathbf{x}^* \in \mathcal{F}$, while strictly decreasing the (squared) Euclidean distance to the goal along the way.*

Proof As discussed in the proof of Proposition 4.3, the positive invariance of \mathcal{F} under the “move-to-projected-goal” law in (4.47) follows from that for any $\mathbf{x} \in \mathcal{F}$, the robot's line-of-sight local free space $\mathcal{LF}_{\mathcal{L}}(\mathbf{x})$ (4.41) is an obstacle free closed convex subset of \mathcal{F} , and contains both \mathbf{x} and $\Pi_{\mathcal{LW}_{\mathcal{L}}(\mathbf{x})}(\mathbf{x}^*)$ (Proposition 4.10 and Theorem 4.2). Hence, $-k(\mathbf{x} - \Pi_{\mathcal{LF}_{\mathcal{L}}(\mathbf{x})}(\mathbf{x}^*)) \in T_{\mathbf{x}}\mathcal{F}$ is either interior directed or, at worst, tangent to the boundary of \mathcal{F} .

The existence, uniqueness and continuity of its flow can be observed using a partitioning of \mathcal{F} such that the “move-to-projected-goal” law is piecewise continuously differentiable in

each connected component of any partition element. Let \mathcal{D}_t denote the set of collision free robot locations at which the number of detected line-of-sight obstacles is equal to $t \in \mathbb{N}$, i.e.,

$$\mathcal{D}_t := \left\{ x \in \mathcal{F} \mid |\mathcal{L}_R(x)| = t \right\}. \quad (4.48)$$

Recall that $\mathcal{L}_R(x) = \{L_1, L_2, \dots, L_t\}$ is our sensor model that returns the list of convex line-of-sight obstacles detected by the LIDAR at location x . Hence, the collection of \mathcal{D}_t 's defines a partition of \mathcal{F} .

Note that \mathcal{D}_t is generally disconnected, because each line-of-sight obstacle is associated with an open convex segment of a LIDAR scan. Further, observe that the “move-to-projected-goal” law is piecewise continuously differentiable when its domain is restricted to any connected component of \mathcal{D}_t , because each connected component of \mathcal{D}_t is associated with a certain collection of obstacles and workspace boundary segments, i.e, the robot persistently perceives the same set of environmental clutter in every connected component of \mathcal{D}_t . Hence, since a piecewise continuously differentiable function is Lipschitz continuous on a compact set [50, 127], the “move-to-projected-goal” law has a unique continuously differentiable flow in every connected component of \mathcal{D}_t . Further, when the robot enters a connected component of \mathcal{D}_t , it stays in that connected component for a nonzero time, because a line-of-sight obstacle L_i is an open set and can not instantaneously appear or disappear under any continuous motion. Thus, the unique, continuous and piecewise differentiable flow of the move-to-projected-goal” law in \mathcal{F} is constructed by piecing together its unique, continuously differentiable trajectories in every connected component of \mathcal{D}_t 's.

Finally, using a similar pattern to the proofs of Proposition 4.5 and Proposition 4.6, one can verify that the set of stationary points of (4.47) is $\{x^*\} \cup \bigcup_{i=1}^m \mathfrak{S}_i$, where \mathfrak{S}_i is defined as in (4.18); and if Assumption 4.2 holds, then the goal x^* is the only locally stable point of (4.47), and all the stationary points, \mathfrak{S}_i , associated with obstacles, O_i , are nondegenerate saddles. Moreover, as discussed in the proof of Proposition 4.7, the “move-to-projected-goal” law in (4.47) strictly decreases the (squared) Euclidean distance to x^* away from its stationary points, and so x^* is the unique attractor of (4.47), whose basin of attraction includes all but a measure zero set of \mathcal{F} . \blacksquare

As a final remark, it is useful to note that the “move-to-projected-goal” law in (4.47) might have discontinuities because of possible occlusions between obstacles. If there is no occlusion between obstacles in the LIDAR's sensing range, then the LIDAR scanner provides exactly the same information about obstacles as does the fixed radius sensory footprint of Section 4.3.3, and so the “move-to-projected-goal” law in (4.47) is piecewise continuously differentiable as is its version in (4.37). In this regard, one can avoid occlusions between obstacles by properly selecting the LIDAR's sensing range so that $A_i \cap A_j \cap B(x, R) = \emptyset$ for all $x \in \mathcal{F}$ and $i \neq j$, where A_i is the augmented line-of-sight obstacle associated with O_i , defined as in (4.44). For example, since $d(x, O_i) \geq r$ for any $x \in \mathcal{F}$ and $d(O_i, O_j) > 2r$ for any $i \neq j$ (Assumption 4.1), a conservative choice of R that prevents occlusions between obstacles is $r < R \leq 3r$.

4.3.4 An Extension for Differential Drive Robots

Maintaining the specialization to the plane, $\mathcal{W} \subset \mathbb{R}^2$, we now consider a disk-shaped differential drive robot described by state $(\mathbf{x}, \theta) \in \mathcal{F} \times (-\pi, \pi]$, centered at $\mathbf{x} \in \mathcal{F}$ with body radius $r \in \mathbb{R}_{>0}$ and orientation $\theta \in (-\pi, \pi]$, moving in \mathcal{W} . The kinematic equations describing its motion are

$$\dot{\mathbf{x}} = v \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix}, \quad \text{and} \quad \dot{\theta} = \omega, \quad (4.49)$$

where $v \in \mathbb{R}$ and $\omega \in \mathbb{R}$ are, respectively, the linear (tangential) and angular velocity inputs of the robot.

In contrary to the “move-to-projected-goal” law of a fully actuated robot in (4.17), a differential drive robot can not directly move towards the projected goal $\Pi_{\mathcal{LF}(\mathbf{x})}(\mathbf{x}^*)$ of a given goal location²⁰ $\mathbf{x}^* \in \mathring{\mathcal{F}}$, unless it is perfectly aligned with $\Pi_{\mathcal{LF}(\mathbf{x})}(\mathbf{x}^*)$, because it is underactuated due to the nonholonomic constraint $\begin{bmatrix} -\sin \theta \\ \cos \theta \end{bmatrix}^T \dot{\mathbf{x}} = 0$. In consequence, to determine the robot’s linear motion, we restrict the robot’s local free space $\mathcal{LF}(\mathbf{x})$ (4.15) to conform to the nonholonomic constraint as

$$\mathcal{LF}_v(\mathbf{x}) := \mathcal{LF}(\mathbf{x}) \cap H_N, \quad (4.50)$$

where $H_N := \left\{ \mathbf{q} \in \mathbb{R}^n \mid \begin{bmatrix} -\sin \theta \\ \cos \theta \end{bmatrix}^T (\mathbf{q} - \mathbf{x}) = 0 \right\}$ is the straight line motion range due to the nonholonomic constraint. Note that $\mathcal{LF}(\mathbf{x}) \cap H_N$ is a closed line segment in \mathcal{W} and contains \mathbf{x} . Similarly, to determine the robot’s angular motion, we define

$$\mathcal{LF}_\omega(\mathbf{x}) := \mathcal{LF}(\mathbf{x}) \cap H_G, \quad (4.51)$$

where $H_G := \{ \omega \mathbf{x} + (1 - \omega) \mathbf{x}^* \in \mathbb{R}^n \mid \omega \in \mathbb{R} \}$ is the line going through \mathbf{x} and \mathbf{x}^* .

Accordingly, based on a standard differential drive controller [19], we propose the following “move-to-projected-goal” law for a differential drive robot,^{21 22}

$$v = -k \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix}^T (\mathbf{x} - \Pi_{\mathcal{LF}_v(\mathbf{x})}(\mathbf{x}^*)) , \quad (4.52a)$$

$$\omega = k \operatorname{atan} \left(\frac{\begin{bmatrix} -\sin \theta \\ \cos \theta \end{bmatrix}^T \left(\mathbf{x} - \frac{\Pi_{\mathcal{LF}_\omega(\mathbf{x})}(\mathbf{x}^*) + \Pi_{\mathcal{LF}(\mathbf{x})}(\mathbf{x}^*)}{2} \right)}{\begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix}^T \left(\mathbf{x} - \frac{\Pi_{\mathcal{LF}_\omega(\mathbf{x})}(\mathbf{x}^*) + \Pi_{\mathcal{LF}(\mathbf{x})}(\mathbf{x}^*)}{2} \right)} \right) , \quad (4.52b)$$

where $k > 0$ is fixed, and $\mathcal{LF}_v(\mathbf{x})$, $\mathcal{LF}_\omega(\mathbf{x})$ and $\mathcal{LF}(\mathbf{x})$ are assumed to be continuously updated.

²⁰Here, we require the goal to be in the interior $\mathring{\mathcal{F}}$ of \mathcal{F} to guarantee that the robot can nearly align its orientation with the (local) goal in finite time.

²¹We follow the paper [19] by resolving the indeterminacy through setting angular velocity $\omega = 0$ whenever $\mathbf{x} = \frac{\Pi_{\mathcal{LF}_\omega(\mathbf{x})}(\mathbf{x}^*) + \Pi_{\mathcal{LF}(\mathbf{x})}(\mathbf{x}^*)}{2}$. Note that this introduces the discontinuity necessitated by Brockett’s condition [40].

We summarize some important properties of the “move-to-projected-goal” law of a differential drive robot as:

Proposition 4.12 *Given the goal and obstacles satisfy Assumption 4.2, the “move-to-projected-goal” law of a disk-shaped differential drive robot in (4.52) asymptotically steers almost all configurations in its positively invariant domain $\mathcal{F} \times (-\pi, \pi]$ towards any given goal location $\mathbf{x}^* \in \mathring{\mathcal{F}}$, without increasing the Euclidean distance to the goal along the way.*

Proof The positive invariance of $\mathcal{F} \times (-\pi, \pi]$ under the “move-to-projected-goal” law (4.52) and the existence and uniqueness of its flow can be established using similar patterns of the proofs of Proposition 4.2, Proposition 4.3 and Proposition 4.4, and the flow properties of the differential drive controller in [19].

As in the proof of Proposition 4.7, using the squared distance to goal, $V(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}^*\|^2$, as a smooth Lypunov function candidate, one can verify the stability properties from (4.6), (4.49), and (4.52) as follows: for any $(\mathbf{x}, \theta) \in \mathcal{F} \times (-\pi, \pi]$,

$$\begin{aligned} \dot{V}(\mathbf{x}) = -k \underbrace{2(\mathbf{x} - \mathbf{x}^*)^T (\mathbf{x} - \Pi_{\mathcal{LF}_v(\mathbf{x})}(\mathbf{x}^*))}_{\geq \|\mathbf{x} - \Pi_{\mathcal{LF}_v(\mathbf{x})}(\mathbf{x}^*)\|^2} &\leq -k \|\mathbf{x} - \Pi_{\mathcal{LF}_v(\mathbf{x})}(\mathbf{x}^*)\|^2 \leq 0. \quad (4.53) \\ \text{since } \mathbf{x} \in \mathcal{LF}_v(\mathbf{x}) \text{ and } \|\mathbf{x} - \mathbf{x}^*\|^2 &\geq \|\Pi_{\mathcal{LF}_v(\mathbf{x})}(\mathbf{x}^*) - \mathbf{x}^*\|^2 \end{aligned}$$

Hence, it follows from LaSalle Invariance Principle [127] that all configurations in $\mathcal{F} \times (-\pi, \pi]$ asymptotically reach the set of configurations, where the robot is located at the associated projected goal $\Pi_{\mathcal{LF}_v(\mathbf{x})}(\mathbf{x}^*)$ at any arbitrary orientation,

$$\left\{ (\mathbf{x}, \theta) \in \mathcal{F} \times (-\pi, \pi] \mid \mathbf{x} = \Pi_{\mathcal{LF}_v(\mathbf{x})}(\mathbf{x}^*) \right\}. \quad (4.54)$$

Note that if $\Pi_{\mathcal{LF}_v(\mathbf{x})}(\mathbf{x}^*)$, $\Pi_{\mathcal{LF}_\omega(\mathbf{x})}(\mathbf{x}^*)$ and $\Pi_{\mathcal{LF}(\mathbf{x})}(\mathbf{x}^*)$ are assumed to be fixed, then the standard differential drive controller asymptotically aligns the robot with $\frac{\Pi_{\mathcal{LF}_\omega(\mathbf{x})}(\mathbf{x}^*) + \Pi_{\mathcal{LF}(\mathbf{x})}(\mathbf{x}^*)}{2}$, i.e., $\begin{bmatrix} -\sin \theta \\ \cos \theta \end{bmatrix}^T \left(\mathbf{x} - \frac{\Pi_{\mathcal{LF}_\omega(\mathbf{x})}(\mathbf{x}^*) + \Pi_{\mathcal{LF}(\mathbf{x})}(\mathbf{x}^*)}{2} \right) = 0$. Hence, using the optimality of metric projection in (4.6), one can conclude that $\Pi_{\mathcal{LF}_v(\mathbf{x})}(\mathbf{x}^*) = \Pi_{\mathcal{LF}_\omega(\mathbf{x})}(\mathbf{x}^*) = \Pi_{\mathcal{LF}(\mathbf{x})}(\mathbf{x}^*)$ whenever $\mathbf{x} = \Pi_{\mathcal{LF}_v(\mathbf{x})}(\mathbf{x}^*)$ and $\begin{bmatrix} -\sin \theta \\ \cos \theta \end{bmatrix}^T \left(\mathbf{x} - \frac{\Pi_{\mathcal{LF}_\omega(\mathbf{x})}(\mathbf{x}^*) + \Pi_{\mathcal{LF}(\mathbf{x})}(\mathbf{x}^*)}{2} \right) = 0$.

Therefore, using a similar approach to the proofs of Proposition 4.5, Lemma 4.2 and Proposition 4.6, one can verify that the set of stationary points of (5.44) is given by

$$\{\mathbf{x}^*\} \times (-\pi, \pi] \bigcup \left\{ (s_i, \theta) \in \mathcal{F} \times (-\pi, \pi] \mid s_i \in \mathfrak{S}_i, \begin{bmatrix} -\sin \theta \\ \cos \theta \end{bmatrix}^T (s_i - \mathbf{x}^*) = 0 \right\}, \quad (4.55)$$

where \mathfrak{S}_i is defined as in (4.18); and every robot configuration located at \mathbf{x}^* is locally stable and all stationary points associated with obstacles are nondegenerate saddles with stable manifolds of measure zero. Thus, the result follows. \blacksquare

²²In the design of angular motion, we particularly select a local target location, $\frac{\Pi_{\mathcal{LF}_\omega(\mathbf{x})}(\mathbf{x}^*) + \Pi_{\mathcal{LF}(\mathbf{x})}(\mathbf{x}^*)}{2} \in \mathring{\mathcal{F}}$ given $\mathbf{x}^* \in \mathring{\mathcal{F}}$, in the interior $\mathring{\mathcal{F}}$ of \mathcal{F} to increase the convergence rate of the resulting vector field. One can consider other convex combinations of $\Pi_{\mathcal{LF}_\omega(\mathbf{x})}(\mathbf{x}^*)$ and $\Pi_{\mathcal{LF}(\mathbf{x})}(\mathbf{x}^*)$, and the resulting vector field retains qualitative properties.

Note that the “move-to-projected-goal” law of a differential drive robot in (4.52) can be extended to limited range sensing models by using the robot’s sensed local free space \mathcal{LF}_s (4.36) or the robot’s line-of-sight local free space \mathcal{LF}_L (4.41) instead of the local free space \mathcal{LW} (4.15), and the resulting vector field planner maintains qualitative properties.

4.4 Numerical Simulations

To demonstrate the motion pattern generated by our “move-to-projected-goal” law around and far away from the goal, we consider a 10×10 and a 50×10 environment cluttered with convex obstacles and a desired goal located at around the upper right corner, as illustrated in Figure 4.3 and Figure 4.4, respectively.²³ We present in these figures example navigation trajectories of the “move-to-projected-goal” law for different sensing and actuation modalities. We observe a significant consistency between the resulting trajectories of the “move-to-projected-goal” law and the boundary of the Voronoi diagram of the environment, where the robot balances its distance to all proximal obstacles while navigating towards its destination — a desired autonomous behaviour for many practical settings instead of follow-

²³For all simulations, we set $r = 0.5$, $R = 2$ and $k = 1$, and all simulations are obtained through numerical integration of the associated “move-to-projected-goal” law using the `ode45` function of MATLAB.

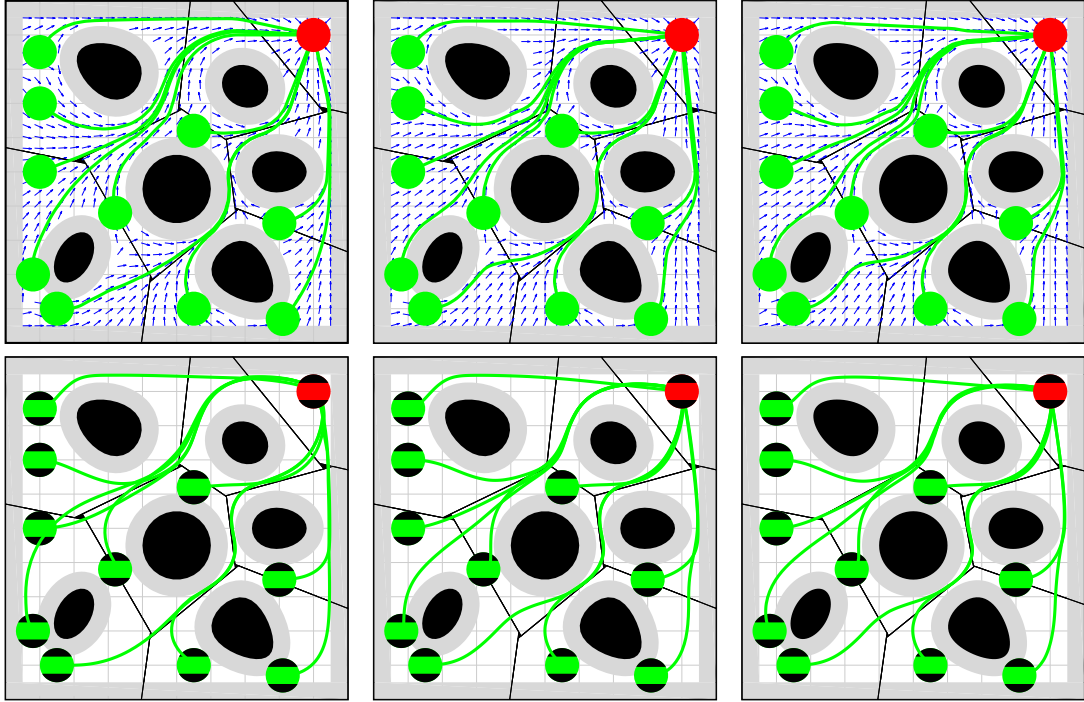


Figure 4.3: Example navigation trajectories of the “move-to-projected-goal” law, starting at a set of initial configurations (green) towards a designated point goal (red), for different sensing and actuation models: (top) a fully actuated robot, (bottom) a differential drive robot, (left) Voronoi-adjacent¹² obstacle sensing, (center) a fixed radius sensory footprint, (right) a limited range LIDAR sensor.

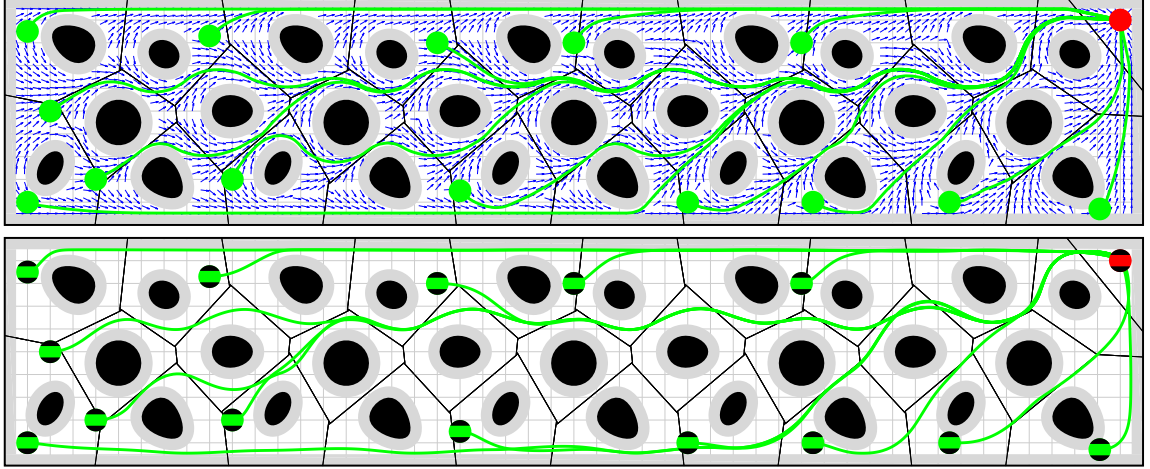


Figure 4.4: Example navigation trajectories of the “move-to-projected-goal” law in (4.17) starting at a set of initial positions (green) far away from the goal (red).²⁴

ing the obstacle boundary tightly. In our simulations, we avoid occlusions between obstacles by properly selecting the LIDAR’s sensing range, and in so doing both limited range sensing models provide the same information about the environment away from the workspace boundary, and the associated “move-to-projected-goal” laws yield almost the same navigation paths. As observed in Figure 4.3, although they are initiated at the same location, a fully actuated and a differential drive robot may follow significantly different trajectories due to their differences in system dynamics and controller design. It is also useful to note that the “move-to-projected-goal” law decreases not only the Euclidean distance, $\|x - x^*\|$, to the goal, but also the Euclidean distance, $\|\Pi_{\mathcal{LF}(x)}(x^*) - x^*\|$, between the projected goal, $\Pi_{\mathcal{LF}(x)}(x^*)$, and the global goal, x^* , illustrated in Figure 4.5.

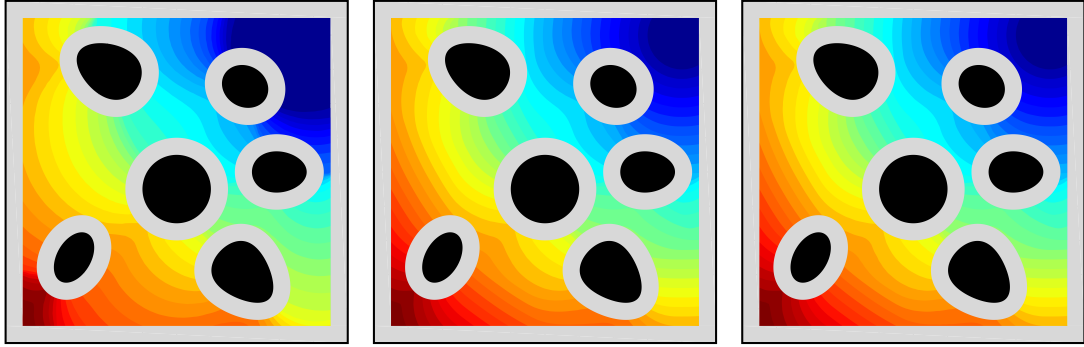


Figure 4.5: The Euclidean distance, $\|\Pi_{\mathcal{LF}(x)}(x^*) - x^*\|$, between the projected goal, $\Pi_{\mathcal{LF}(x)}(x^*)$, and the global goal, x^* , for different sensing modalities: (left) Voronoi-adjacent¹² obstacle sensing, (middle) a fixed radius sensory footprint, (right) a limited range line-of-sight sensor.

²⁴ Please refer to Appendix C.6 for additional figures illustrating the navigation pattern far away from the goal for different sensing and actuation models.

4.5 Summary

In this chapter, we construct a sensor-based feedback law that provably solves the real-time collision free robot navigation problem in a domain cluttered with unknown but sufficiently separated and strongly convex obstacles. Our algorithm introduces a novel use of separating hyperplanes for identifying the robot’s local obstacle free convex neighborhood, affording a piecewise smooth velocity command instantaneously pointing toward the metric projection of the designated goal location onto this convex set. Given sufficiently separated (Assumption 4.1) and appropriately “strongly” convex (Assumption 4.2) obstacles, we show that the resulting vector field has a smooth flow with a unique attractor at the goal location (along with the topologically inevitable saddles — at least one for each obstacle). Since all of its critical points are nondegenerate, our vector field asymptotically steers almost all configurations in the robot’s free space to the goal, with the guarantee of no collisions along the way. We also present its practical extensions for two limited range sensing models and the widely used differential drive model, while maintaining formal guarantees. We illustrate the effectiveness of the proposed navigation algorithm in numerical simulations.



Chapter 5

Voronoi-Based Coverage Control of Heterogeneous Disk-Shaped Robots

Among the many proposed multiple mobile sensor coordination strategies [196], Voronoi-based coverage control [60] uniquely combines both deployment and allocation in an intrinsically distributed manner [167] via gradient descent (the “move-to-centroid” law) down a utility function minimizing the expected event sensing cost to adaptively achieve a *centroidal Voronoi configuration* (depicted on the left in Figure 5.1). Since the original application to homogeneous point robots [60], a growing literature considers the extension to heterogeneous groups of robots differing variously in their sensorimotor capabilities [178, 122, 177, 137] by recourse to *power diagrams* — generalized Voronoi diagrams with additive weights [20].

Although it inherits many nice properties of a standard Voronoi diagram such as convexity and dual triangulability, a power diagram may possibly have empty cells associated with some (unassigned) robots and/or some robots may not be contained in their nonempty cells [20], as situation depicted on the middle in Figure 5.1. Such *occupancy defects* (Definition 5.1) generally cost resource inefficiency or redundancy¹, and, crucially, they re-introduce the problem of collision avoidance — the chief motivation for the present chapter.

Voronoi-based coverage control implicitly entails collision avoidance for point robots since robots move in their pairwise disjoint Voronoi cells [60], but an additional collision avoidance strategy is mandatory for safe navigation of finite size robots. Existing work on combining coverage control and collision avoidance generally uses (i) either heuristic approaches based on repulsive fields [71, 179] and reciprocal velocity obstacles [38] causing robots to converge to configurations far from optimal sensing configurations; or (ii) the projection of a vector field whenever a robot reaches the boundary of its partition cell [178, 121] introducing a source of discontinuity. An important observation made in [178] is that it is sufficient to restrict robot bodies to respective Voronoi regions for collision avoidance, but this is a conservative assumption for robot groups with different body sizes (as illustrated on the right in Figure 5.1).

¹Note that a power diagram with an occupancy defect can be beneficial in certain applications to save/balance energy across a mobile network of power limited agents [137].

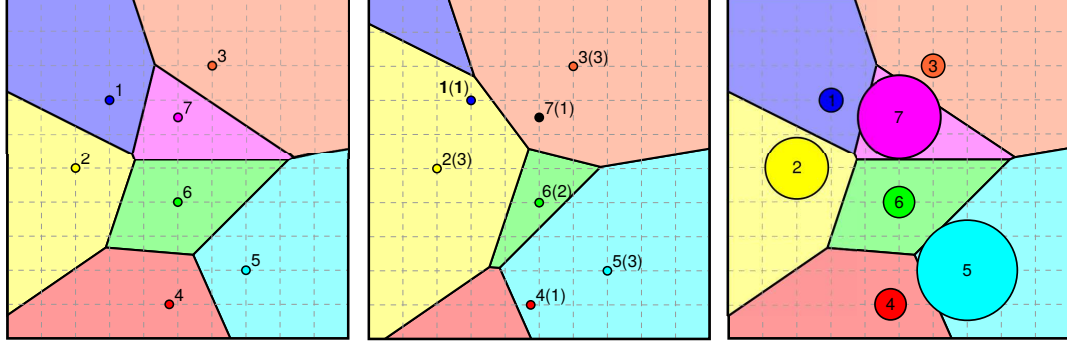


Figure 5.1: An illustration of (left) the Voronoi and (middle) power diagrams of an environment based on a noncolliding placement of point robots, where the weights of power cells are shown in parentheses. Although each point robot is always contained in its Voronoi cell, power cells associated with some robots (e.g. the 7th robot) may be empty and/or some robots (e.g. the 1st and 4th robots) may not be contained in their nonempty power cells. (right) A collision free disk configuration does not necessarily have Voronoi cells containing respective robot bodies.

In this chapter, we provide a necessary and sufficient condition for identifying collision-free configurations of finite size robots in terms of their power diagrams, and accordingly propose a constrained coverage control (“move-to-constrained-centroid”) law, whose continuous and piecewise smooth flow asymptotically converges to an optimal sensing configuration, while avoiding any collisions along the way. We extend the practicability of the result by adding a congestion management heuristic for unassigned robots that hastens the assigned robots’ progress, and, finally, adapt the fully actuated point particle vector field planner to the widely used kinematic differential drive vehicle model (retaining the convergence and collision avoidance guarantees in both extensions).

This chapter is based on the conference paper [12], and presents a new robot-centric application of Voronoi diagrams to exactly encode (multi)robot collisions.

5.1 Coverage Control of Point Robots

5.1.1 Location Optimization of Homogeneous Robots

Let Q be a convex environment in the n -dimensional Euclidean space \mathbb{R}^n with a priori known event distribution function $\phi : Q \rightarrow \mathbb{R}_{>0}$ that models the probability of some event occurs in Q , and $\mathbf{p} := (p_1, p_2, \dots, p_m) \in Q^m$ be a (noncolliding) placement of $m \in \mathbb{N}$ point robots in Q .² Suppose that the event detection (sensing) cost of i th robot at location $q \in Q$ is a nondecreasing differentiable function, $f : \mathbb{R} \rightarrow \mathbb{R}$, of the Euclidean distance, $\|q - p_i\|$, between q and p_i . Further assume that robots are assigned to events based on a partition of Q yielding a cover, $\mathcal{W} := \{W_1, W_2, \dots, W_m\}$, a collection of subsets (“cells”), W_i ,

²Here, \mathbb{N} is the set of all natural numbers; \mathbb{R} and $\mathbb{R}_{>0}$ ($\mathbb{R}_{\geq 0}$) denote the set of real and positive (nonnegative) real numbers, respectively.

whose union returns Q but whose cells have mutually disjoint interiors.³ A well established approach (arising in both facility location [101, 167] and quantization [149, 73] problems) achieves such a cover by minimizing the expected event sensing cost,

$$\mathcal{H}(\mathbf{p}, \mathcal{W}) := \sum_{i=1}^m \int_{W_i} f(\|\mathbf{q} - \mathbf{p}_i\|) \phi(\mathbf{q}) d\mathbf{q} . \quad (5.1)$$

Now observe that, for any fixed \mathbf{p} , the optimal task assignment minimizing \mathcal{H} is the standard Voronoi diagram $\mathcal{V}(\mathbf{p}) := \{V_1, \dots, V_m\}$ of Q based on the configuration \mathbf{p} ,

$$V_i = \left\{ \mathbf{q} \in Q \mid \|\mathbf{q} - \mathbf{p}_i\| \leq \|\mathbf{q} - \mathbf{p}_j\|, \forall j \neq i \right\} . \quad (5.2)$$

Thus, given the optimal task assignment of robots, the objective function \mathcal{H} takes the following form

$$\mathcal{H}_{\mathcal{V}}(\mathbf{p}) := \mathcal{H}(\mathbf{p}, \mathcal{V}(\mathbf{p})) = \sum_{i=1}^m \int_{V_i} f(\|\mathbf{q} - \mathbf{p}_i\|) \phi(\mathbf{q}) d\mathbf{q} , \quad (5.3)$$

and it is common knowledge that [73, 167, 60]

$$\frac{\partial \mathcal{H}_{\mathcal{V}}(\mathbf{p})}{\partial \mathbf{p}_i} = \int_{V_i} \frac{\partial}{\partial \mathbf{p}_i} f(\|\mathbf{q} - \mathbf{p}_i\|) \phi(\mathbf{q}) d\mathbf{q} . \quad (5.4)$$

In the special case of $f(x) = x^2$, the partial derivative of $\mathcal{H}_{\mathcal{V}}$ has a simple physical interpretation as follows:

$$\frac{\partial \mathcal{H}_{\mathcal{V}}(\mathbf{p})}{\partial \mathbf{p}_i} = 2\mu_{V_i} (\mathbf{p}_i - \mathbf{c}_{V_i}) , \quad (5.5)$$

where μ_{V_i} and \mathbf{c}_{V_i} , respectively, denote the mass and the center of mass of V_i according to the mass density function ϕ ,

$$\mu_{V_i} := \int_{V_i} \phi(\mathbf{q}) d\mathbf{q} , \quad \mathbf{c}_{V_i} := \int_{V_i} \mathbf{q} \phi(\mathbf{q}) d\mathbf{q} . \quad (5.6)$$

Hence, at a critical point of $\mathcal{H}_{\mathcal{V}}$, the robots are located at the centroid of their respective Voronoi cells, called a centroidal Voronoi configuration.

Assuming first-order (completely actuated single-integrator) robot dynamics,

$$\dot{\mathbf{p}}_i = \mathbf{u}_i , \quad (5.7)$$

the standard “move-to-centroid” law asymptotically steering point robots to a centroidal Voronoi configuration with the guarantee of no collision along the way is

$$\mathbf{u}_i = -k (\mathbf{p}_i - \mathbf{c}_{V_i}) , \quad (5.8)$$

where $k \in \mathbb{R}_{>0}$ is a fixed control gain and the Voronoi diagram $\mathcal{V}(\mathbf{p})$ of Q is assumed to be continuously updated. Note that μ_{V_i} and \mathbf{c}_{V_i} are both continuously differentiable

³We will generally refer to such decompositions as “diagrams” but also occasionally allow the slight abuse of language to follow tradition and refer to \mathcal{W} as a *partition*.

functions of \mathbf{p} as are both \mathcal{H}_V and u_i [44]. Finally, observe, again, that the coverage control u_i supports a distributed implementation whose local communications structure is specified by the associated Delaunay graph [60].

5.1.2 Location Optimization of Heterogeneous Robots

In distributed sensing applications, heterogeneity of robotic networks in sensing and actuation [178, 122, 177, 137] is often modelled by recourse to *power diagrams*, generalized Voronoi diagrams with additive weights [20]. More precisely, for a given multirobot configuration $\mathbf{p} \in Q^m$, the event sensing cost of i th robot at location $\mathbf{q} \in Q$ is assumed to be given by the *power distance*, $\|\mathbf{q} - \mathbf{p}_i\|^2 - \rho_i^2$ where $\rho_i \in \mathbb{R}_{\geq 0}$ is the *power radius* of i th robot. Accordingly, the task assignment of robots are determined by the power diagram $\mathcal{P}(\mathbf{p}, \boldsymbol{\rho}) := \{P_1, P_2, \dots, P_m\}$ of Q based on the configuration \mathbf{p} and the associated power radii $\boldsymbol{\rho} := (\rho_1, \rho_2, \dots, \rho_m)$,

$$P_i := \left\{ \mathbf{q} \in Q \mid \|\mathbf{q} - \mathbf{p}_i\|^2 - \rho_i^2 \leq \|\mathbf{q} - \mathbf{p}_j\|^2 - \rho_j^2, \forall j \neq i \right\}, \quad (5.9)$$

and the location optimization function becomes

$$\mathcal{H}_{\mathcal{P}}(\mathbf{p}, \boldsymbol{\rho}) = \sum_{i=1}^m \int_{P_i} \left(\|\mathbf{q} - \mathbf{p}_i\|^2 - \rho_i^2 \right) \phi(\mathbf{q}) d\mathbf{q}. \quad (5.10)$$

Note that in the special case of $\rho_i = \rho_j$ for all $i \neq j$ the power diagram $\mathcal{P}(\mathbf{p}, \boldsymbol{\rho})$ and the Voronoi diagram $\mathcal{V}(\mathbf{p})$ of Q are identical, i.e., $P_i = V_i$, and the utility functions $\mathcal{H}_{\mathcal{P}}$ and \mathcal{H}_V are related by a constant offset, i.e., $\mathcal{H}_{\mathcal{P}}(\mathbf{p}, \boldsymbol{\rho}) = \mathcal{H}_V(\mathbf{p}) + c(\boldsymbol{\rho})$ for some $c(\boldsymbol{\rho}) \in \mathbb{R}$.

Similar to (5.5), for fixed $\boldsymbol{\rho}$, the partial derivative of $\mathcal{H}_{\mathcal{P}}$ takes the following simple form [178, 137, 179],

$$\frac{\partial \mathcal{H}_{\mathcal{P}}(\mathbf{p}, \boldsymbol{\rho})}{\partial \mathbf{p}_i} = 2\mu_{P_i} (\mathbf{p}_i - \mathbf{c}_{P_i}), \quad (5.11)$$

where μ_{P_i} and \mathbf{c}_{P_i} are the mass and the center of mass ⁴ of P_i , respectively, as defined in (5.6).

For the single integrator robot model (5.7), the standard “move-to-centroid” law of heterogeneous robotic networks asymptotically driving robots to a critical point of $\mathcal{H}_{\mathcal{P}}(\cdot, \boldsymbol{\rho})$, where robots are located at the centroids of their respective power cells, is defined as

$$\mathbf{u}_i = -k (\mathbf{p}_i - \mathbf{c}_{P_i}), \quad (5.12)$$

for some fixed $k \in \mathbb{R}_{>0}$ and the power diagram $\mathcal{P}(\mathbf{p}, \boldsymbol{\rho})$ of Q is assumed to be continuously updated. Notwithstanding its welcome inheritance of many standard Voronoi properties (e.g., convexity, dual triangulability), a power diagram may yield empty cells associated with some robots and/or some robots may not be contained in their nonempty power cells, illustrated in Figure 5.1. In consequence, contrary to the case of homogeneous robots, the “move-to-centroid” law of heterogeneous point robots is discontinuous and it cannot guarantee collision-free navigation. Thus, in past literature, for robots of finite but hetero-

⁴ To be well defined we set $\mathbf{c}_{P_i} = \mathbf{p}_i$ whenever P_i has an empty interior.

geneous size, the standard “move-to-centroid” law inevitably requires an additional heuristic collision avoidance strategy for safe navigation.

5.2 Occupancy Defects of Power Diagrams

Definition 5.1 (Occupancy Defect) *The power partition, $\mathcal{P}(\mathbf{p}, \boldsymbol{\rho})$, associated with configuration $\mathbf{p} \in Q^m$ and radii $\boldsymbol{\rho} \in (\mathbb{R}_{\geq 0})^m$ is said to have an occupancy defect if $\mathbf{p}_i \notin P_i$ for some $i \in \{1, 2, \dots, m\}$.*

Configurations incurring this unfortunate occupancy defects introduce a number of problems. First of all, empty partition cells cause resource redundancy, because some robots may never be assigned to any event happening around them. Such robots do not only become redundant, but also complicate collision avoidance as (moving or stationary) obstacles and limit the mobility of others. In general, robots that are not contained in their respective cells require an extra care for collision avoidance.

A straightforward characterization of an occupancy defective configuration is: ⁵

Proposition 5.1 *Given radii $\boldsymbol{\rho} \in (\mathbb{R}_{\geq 0})^m$, configuration $\mathbf{p} \in Q^m$ does not incur an occupancy defective power diagram if and only if $\|\mathbf{p}_i - \mathbf{p}_j\|^2 \geq |\rho_i^2 - \rho_j^2|$ for all $i \neq j$.*

Proof By Definition 5.1, $\mathcal{P}(\mathbf{p}, \boldsymbol{\rho})$ has no occupancy defect if and only if $\mathbf{p}_i \in P_i$ for all i , which is the case if and only if

$$\|\mathbf{p}_i - \mathbf{p}_i\|^2 - \rho_i^2 \leq \|\mathbf{p}_i - \mathbf{p}_j\|^2 - \rho_j^2, \quad (5.13)$$

$$\|\mathbf{p}_j - \mathbf{p}_j\|^2 - \rho_j^2 \leq \|\mathbf{p}_j - \mathbf{p}_i\|^2 - \rho_i^2, \quad (5.14)$$

for all $i \neq j$. Thus, the result follows. ■

In what follows, we present a novel use of power diagrams for identifying collision-free multirobot configurations and propose a constrained optimization combining coverage control and collision avoidance for heterogeneous robots.

5.3 Combining Coverage Control and Collision Avoidance

Throughout the rest of paper, we consider heterogeneous disk-shaped multirobot configurations, $\mathbf{p} = (\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_m) \in Q^m$, in Q with associated vectors of nonnegative body radii $\boldsymbol{\beta} := (\beta_1, \beta_2, \dots, \beta_m) \in (\mathbb{R}_{\geq 0})^m$ and sensory footprint radii $\boldsymbol{\sigma} := (\sigma_1, \sigma_2, \dots, \sigma_m) \in (\mathbb{R}_{\geq 0})^m$, where i th robot is centered at $\mathbf{p}_i \in Q$ and has body radius $\beta_i \geq 0$ and sensory footprint radius $\sigma_i \geq 0$. Accordingly, we will denote by $\mathcal{B}(\mathbf{p}, \boldsymbol{\beta}) = \{B_1, B_2, \dots, B_m\}$, a cover we term the *body diagram* of Q , solving the power problem (5.9), (5.10), defined from $\mathcal{H}_{\mathcal{B}}(\mathbf{p}, \boldsymbol{\beta})$; and we will denote by $\mathcal{S}(\mathbf{p}, \boldsymbol{\sigma}) = \{S_1, S_2, \dots, S_m\}$, a cover we term the *sensor diagram* of

⁵In [122], the authors note the issue of empty power cells and give a similar sufficient condition for each robot to be contained in its power cell, but this sufficiency condition is neither used in the design of the proposed coverage control law nor guaranteed to hold during the evaluation of the proposed coverage control algorithm.

Q , solving the corresponding problem defined by $\mathcal{H}_s(\mathbf{p}, \boldsymbol{\sigma})$. We also find it convenient to denote the configuration space of body-noncolliding disks of radii $\boldsymbol{\beta}$ in Q as

$$\mathbf{Conf}(Q, \boldsymbol{\beta}) := \left\{ \mathbf{p} \in Q^m \mid \|\mathbf{p}_i - \mathbf{p}_j\| > \beta_i + \beta_j \quad \forall i \neq j, D(\mathbf{p}_i, \beta_i) \subset \overset{\circ}{Q} \quad \forall i \right\}, \quad (5.15)$$

where $D(\mathbf{x}, r) := \{\mathbf{y} \in \mathbb{R}^n \mid \|\mathbf{x} - \mathbf{y}\| \leq r\}$ is the closed disk in \mathbb{R}^n centered at $\mathbf{x} \in \mathbb{R}^n$ with radius $r \geq 0$, and $\overset{\circ}{Q}$ is the interior of Q . Note that the vectors of body radii $\boldsymbol{\beta}$ and sensory footprint radii $\boldsymbol{\sigma}$ are not necessary equal, because $\boldsymbol{\beta}$ models the heterogeneity of robots in body size, $\boldsymbol{\sigma}$ models their heterogeneity in sensing and actuation.

5.3.1 Encoding Collisions via Body Diagrams

A geometric characterization of collision-free multirobot configurations in Q via their body diagrams is:

Proposition 5.2 *Let $\mathcal{B}(\mathbf{p}, \boldsymbol{\beta})$ be the body diagram of Q associated with configuration $\mathbf{p} \in Q^m$ (such that $\mathbf{p}_i \neq \mathbf{p}_j$ for all $i \neq j$) and body radii $\boldsymbol{\beta} \in (\mathbb{R}_{\geq 0})^m$. Then \mathbf{p} is collision-free if and only if every robot body is contained in the interior of its body cell, i.e.,*

$$\mathbf{p} \in \mathbf{Conf}(Q, \boldsymbol{\beta}) \iff D(\mathbf{p}_i, \beta_i) \subset \overset{\circ}{B}_i \quad \forall i. \quad (5.16)$$

Proof The sufficiency (\Leftarrow) follows because $\mathcal{B}(\mathbf{p}, \boldsymbol{\beta})$ is a cover of Q whose elements have disjoint interiors. Hence, given $D(\mathbf{p}_i, \beta_i) \subset \overset{\circ}{B}_i$ for all i , we have $D(\mathbf{p}_i, \beta_i) \subset \overset{\circ}{Q}$ and $D(\mathbf{p}_i, \beta_i) \cap D(\mathbf{p}_j, \beta_j) = \emptyset$ for all $i \neq j$, and so $\|\mathbf{p}_i - \mathbf{p}_j\| > \beta_i + \beta_j$. Thus, $\mathbf{p} \in \mathbf{Conf}(Q, \boldsymbol{\beta})$.

To see the necessity (\Rightarrow), for any $\mathbf{p} \in \mathbf{Conf}(Q, \boldsymbol{\beta})$, we will show that $\mathbf{p}_i \in B_i$ for all i , and the distance between \mathbf{p}_i and the boundary ∂B_i of B_i is greater than β_i , i.e., $\min_{\mathbf{x} \in \partial B_i} \|\mathbf{x} - \mathbf{p}_i\| > \beta_i$, and so $D(\mathbf{p}_i, \beta_i) \subset \overset{\circ}{B}_i$.

It follows from Proposition 5.1 that for any $\mathbf{p} \in \mathbf{Conf}(Q, \boldsymbol{\beta})$, $\mathcal{B}(\mathbf{p}, \boldsymbol{\beta})$ has no occupancy defect (Definition 5.1), i.e., $\mathbf{p}_i \in B_i \quad \forall i$.

The boundary ∂B_i of B_i is defined by the boundary ∂Q of Q and the separating separating hyperplane between body cells B_i and B_j for some $j \neq i$ [20]. By definition (5.15), we have $\min_{\mathbf{x} \in \partial Q} \|\mathbf{x} - \mathbf{p}_i\| > \beta_i$ for any $\mathbf{p} \in \mathbf{Conf}(Q, \boldsymbol{\beta})$.

Now observe that, for any $i \neq j$, the separating hyperplane between body cells B_i and B_j is perpendicular to the line joining \mathbf{p}_i and \mathbf{p}_j and is given by [20]

$$H_{ij} := \left\{ \mathbf{x} \in \mathbb{R}^n \mid 2\mathbf{x}^T (\mathbf{p}_i - \mathbf{p}_j) = \beta_j^2 - \beta_i^2 + \|\mathbf{p}_i\|^2 - \|\mathbf{p}_j\|^2 \right\}, \quad (5.17)$$

and the perpendicular distance of \mathbf{p}_i to H_{ij} is given by

$$d(\mathbf{p}_i, H_{ij}) := \frac{\|\mathbf{p}_i - \mathbf{p}_j\|}{2} + \frac{\beta_i^2 - \beta_j^2}{2\|\mathbf{p}_i - \mathbf{p}_j\|}. \quad (5.18)$$

Note that $d(\mathbf{p}_i, H_{ij})$ is negative when $\mathcal{B}(\mathbf{p}, \boldsymbol{\beta})$ has an occupancy defect; and we have from Proposition 5.1 that $\mathcal{B}(\mathbf{p}, \boldsymbol{\beta})$ is free of such a defect for any $\mathbf{p} \in \mathbf{Conf}(Q, \boldsymbol{\beta})$ and so $d(\mathbf{p}_i, H_{ij}) \geq 0$. One can further show that for any $i \neq j$,

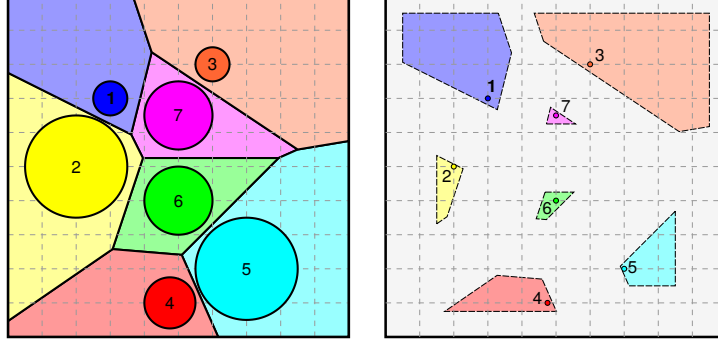


Figure 5.2: (left) Encoding collision free configurations via body diagrams: A configuration of disks is nonintersecting iff each disk is contained in the interior of its body cell. (right) Free subcells, obtained by eroding each body cell with its associated disk radius.

$$\begin{aligned}
 d(\mathbf{p}_i, H_{ij}) &= \beta_i + \frac{\|\mathbf{p}_i - \mathbf{p}_j\|^2 + \beta_i^2 - \beta_j^2 - 2\beta_i \|\mathbf{p}_i - \mathbf{p}_j\|}{2\|\mathbf{p}_i - \mathbf{p}_j\|}, \\
 &= \beta_i + \underbrace{\frac{(\|\mathbf{p}_i - \mathbf{p}_j\| - \beta_i)^2 - \beta_j^2}{2\|\mathbf{p}_i - \mathbf{p}_j\|}}_{>0, \text{ since } \mathbf{p} \in \text{Conf}(Q, \boldsymbol{\beta})} > \beta_i,
 \end{aligned} \tag{5.19}$$

which completes the proof. ■

To determine a collision-free neighborhood of a configuration $\mathbf{p} \in \text{Conf}(Q, \boldsymbol{\beta})$ with a vector of body radii $\boldsymbol{\beta} \in (\mathbb{R}_{\geq 0})^m$, we define a *free subdiagram* $\mathcal{F}(\mathbf{p}, \boldsymbol{\beta}) := \{F_1, F_2, \dots, F_m\}$ of the body diagram $\mathcal{B}(\mathbf{p}, \boldsymbol{\beta}) = \{B_1, B_2, \dots, B_m\}$ by eroding each cell removing the volume swept along its boundary, ∂B_i , by the associated body radius,⁶ see Figure 5.2, as [103]

$$F_i := B_i \setminus (\partial B_i \oplus D(\mathbf{0}, \beta_i)) = \left\{ \mathbf{q} \in B_i \mid \min_{\mathbf{x} \in \partial B_i} \|\mathbf{x} - \mathbf{q}\| > \beta_i \right\}. \tag{5.20}$$

Note that F_i is a nonempty convex set because $\mathbf{p}_i \in F_i$ and the erosion of a convex set by a ball is convex.⁷

The following observation yields a (possibly conservative) convex inner approximation of the free configuration space neighborhood surrounding free configuration as

$$\mathbf{p} \in \text{Conf}(Q, \boldsymbol{\beta}) \Rightarrow \prod \mathcal{F}(\mathbf{p}, \boldsymbol{\beta}) \subset \text{Conf}(Q, \boldsymbol{\beta}), \tag{5.21}$$

where $\prod \mathcal{F}(\mathbf{p}, \boldsymbol{\beta}) = F_1 \times F_2 \times \dots \times F_m$.

Lemma 5.1 *Let $\mathbf{p} \in \text{Conf}(Q, \boldsymbol{\beta})$ be a multirobot configuration with a vector of body radii $\boldsymbol{\beta} \in (\mathbb{R}_{\geq 0})^n$, and $\mathcal{F}(\mathbf{p}, \boldsymbol{\beta})$ be the free subdiagram of the body diagram $\mathcal{B}(\mathbf{p}, \boldsymbol{\beta})$.*

⁶Here, $\mathbf{0}$ is a vector of all zeros with the appropriate size, and $A \oplus B := \{a + b \mid a \in A, b \in B\}$ is the Minkowski sum of sets A and B .

⁷It is obvious that the erosion of a half-space by a ball is a half-space. Hence, since the erosion operation is distributed over set intersection [103], and a convex set can be defined as (possibly infinite) intersection of half-spaces [37], the erosion of a convex set by a ball is convex.

Then $\mathbf{q} \in Q^m$ is a collision-free multirobot configuration in $\mathbf{Conf}(Q, \beta)$ if $\mathbf{q}_i \in F_i$ (i.e., $D(\mathbf{q}_i, \beta_i) \subset \overset{\circ}{B}_i$) for all i .

Proof The results directly follows from $\mathcal{B}(\mathbf{p}, \beta)$ covering a partition of Q , as discussed in the proof of Proposition 5.2. \blacksquare

5.3.2 Coverage Control of Heterogeneous Disk-Shaped Robots

Based on our observations in Proposition 5.2 and Lemma 5.1, we now introduce a constrained optimization framework combining coverage control and collision avoidance.

Consider a heterogeneous multirobot configuration $\mathbf{p} \in \mathbf{Conf}(Q, \beta + \epsilon)$ with associated vectors of body radii $\beta \in (\mathbb{R}_{\geq 0})^m$, safety margins $\epsilon \in (\mathbb{R}_{> 0})^m$ and sensory footprint radii $\sigma \in (\mathbb{R}_{\geq 0})^m$, and let $\mathcal{S}(\mathbf{p}, \sigma) = \{S_1, S_2, \dots, S_m\}$ be the sensory diagram of Q based on robot locations \mathbf{p} and sensory footprint radii σ , and $\mathcal{F}(\mathbf{p}, \beta + \epsilon) = \{F_1, F_2, \dots, F_m\}$ be the free subdiagram associated with configuration \mathbf{p} and enlarged body radii $\beta + \epsilon$. Here we use ϵ to guarantee the clearance between any pair $i \neq j$ of robots to be at least $\epsilon_i + \epsilon_j$.⁸

Now, in contrast to the standard “move-to-centroid” law that steers each robot directly towards the centroid, c_{S_i} , of its sensory cell, S_i , we propose a coverage control policy that selects a safe target location, called the *constrained centroid* of S_i , that solves the following convex programming⁹

$$\begin{aligned} & \text{minimize} && \|\mathbf{q}_i - c_{S_i}\|^2 \\ & \text{subject to} && \mathbf{q}_i \in \overline{F}_i \end{aligned} \tag{5.22}$$

where \overline{F}_i is a closed convex set. It is well known that the unique solution of (5.22) is given by [37, Section 8.1.1]¹⁰

$$\overline{c}_{S_i} := \begin{cases} c_{S_i}, & \text{if } c_{S_i} \in \overline{F}_i, \\ \Pi_{\overline{F}_i}(c_{S_i}), & \text{otherwise,} \end{cases} \tag{5.23}$$

where $\Pi_C(\mathbf{x})$ denotes the metric projection of $\mathbf{x} \in \mathbb{R}^n$ onto a convex set $C \subset \mathbb{R}^n$, and note that Π_C is piecewise continuously differentiable [135, 199, 146].¹¹ Accordingly, for the single integrator robot dynamics (5.7), our “move-to-constrained-centroid” law is defined as

$$\mathbf{u}_i = -k(\mathbf{p}_i - \overline{c}_{S_i}), \tag{5.24}$$

where $k \in \mathbb{R}_{> 0}$ is a fixed control gain, and we assume that $\mathcal{S}(\mathbf{p}, \sigma)$ and $\mathcal{F}(\mathbf{p}, \beta + \epsilon)$ are continuously updated. We find it convenient to have $G_s(Q, \beta + \epsilon, \sigma)$ denote the set of

⁸Having a positive vector of safety margins ϵ enables us to consider collision-free multirobot configurations in $\mathbf{Conf}(Q, \beta + \epsilon) \subset \mathbf{Conf}(Q, \beta)$. Throughout the rest of the paper, in order to simplify the notation, we will abuse the notation and use $\mathbf{Conf}(Q, \beta + \epsilon)$ to refer to the closure of the configuration space in (5.15).

⁹Here, \overline{A} is the closure of set A .

¹⁰In general, the metric projection of a point onto a convex set can be efficiently computed using a standard convex programming solver [37]. If Q is a convex polytope, then a free subcell, F_i , is also a convex polytope and can be written as a finite intersection of half-spaces. Hence, the metric projection onto a convex polytope can be recast as quadratic programming and can be solved in polynomial time [133]. In the case of a convex polygonal environment, F_i is a convex polygon and the metric projection onto a convex polygon can be solved analytically since the solution lies on one of its edges unless the input point is inside the polygon.

¹¹Note that c_{S_i} is well defined (see footnote 4), hence \overline{c}_{S_i} must be as well given $F_i \neq \emptyset$.

equilibria of our “move-to-constrained-centroid” law, where robots are located at the constrained centroid of their respective sensory cells,¹²

$$G_s(Q, \beta + \epsilon, \sigma) := \left\{ \mathbf{p} \in \text{Conf}(Q, \beta + \epsilon) \mid \mathbf{p}_i = \bar{\mathbf{c}}_{S_i} \quad \forall i \right\}. \quad (5.25)$$

In the special case of identical sensory footprint radii, i.e., $\sigma_i = \sigma_j$ for all $i \neq j$, these stationary configurations are called the constrained centroidal Voronoi configurations [74]. Also note that for homogeneous point robots, our “move-to-constrained-centroid” law in (5.24) simplifies back to the standard “move-to-centroid” law in (5.8).

We summarize the qualitative properties of our “move-to-constrained-centroid” law as follows:

Theorem 5.1 *For any choice of vectors of body radii $\beta \in (\mathbb{R}_{\geq 0})^m$, safety margin $\epsilon \in (\mathbb{R}_{> 0})^m$ and sensory footprint radii $\sigma \in (\mathbb{R}_{\geq 0})^m$, the configuration space of nonintersecting disks $\text{Conf}(Q, \beta + \epsilon)$ (5.15) is positive invariant under the “move-to-constrained-centroid” law in (5.24) whose unique, continuous and piecewise differentiable flow, starting at any configuration in $\text{Conf}(Q, \beta + \epsilon)$, asymptotically reaches a locally optimal sensing configuration in $G_s(Q, \beta + \epsilon, \sigma)$, while strictly decreasing the utility function $\mathcal{H}_s(\cdot, \sigma)$ (5.10) along the way. If an equilibrium in $G_s(Q, \beta + \epsilon, \sigma)$ is isolated, then it is locally asymptotically stable.*

Proof The instantaneous “target” in (5.24) lies in the closure of the convex inner approximation to the freespace neighborhood of any free configuration, $\bar{\mathbf{c}}_{s(\mathbf{p}, \sigma)} \in \prod \mathcal{F}(\mathbf{p}, \beta + \epsilon) \subset \text{Conf}(Q, \beta + \epsilon)$, hence, according to Lemma 5.1, the configuration space tangent vector defined by (5.24), $-k(\mathbf{p} - \bar{\mathbf{c}}_{s(\mathbf{p}, \sigma)}) \in T_{\mathbf{p}}\text{Conf}(Q, \beta + \epsilon)$, is either interior directed or, at worse, tangent to the boundary of $\prod \mathcal{F}(\mathbf{p}, \beta + \epsilon)$. Therefore, by construction (5.22), the “move-to-constrained-centroid” law leaves $\text{Conf}(Q, \beta + \epsilon)$ positively invariant.

The existence, uniqueness and continuity of its flow can be observed using an equivalent hybrid system consisting of a family of piecewise continuously differentiable local vector fields as follows. Let $\mathbf{u}^I : \mathcal{D}_I \rightarrow (\mathbb{R}^n)^m$ be a local controller associated with a subset I of $\{1, 2, \dots, m\}$ defined as

$$\mathbf{u}_i^I = \begin{cases} -k(\mathbf{p}_i - \bar{\mathbf{c}}_{S_i}), & \text{if } i \in I, \\ \mathbf{0}, & \text{otherwise,} \end{cases} \quad (5.26)$$

where its domain is

$$\mathcal{D}_I := \left\{ \mathbf{p} \in \text{Conf}(Q, \beta + \epsilon) \mid \dot{S}_i \neq \emptyset \quad \forall i \in I \right\}. \quad (5.27)$$

Note that for a given configuration in its domain, \mathcal{D}_I , a local policy index, I , indicates which robots are assigned to sensory cells with nonempty interiors, and so the domains, \mathcal{D}_I , of local controllers defines a finite open cover of $\text{Conf}(Q, \beta + \epsilon)$. Hence, since all unassigned robots are stationary under the “move-to-constrained-centroid” law and every robot whose sensory cell has a nonempty interior is assigned to the coverage task, one can

¹² Note that this set cannot be empty since it contains the minima of a smooth function over a compact set (5.22).

further conclude that these local controllers can be composed using the policy selection strategy, $g : \mathbf{Conf}(Q, \beta + \epsilon) \rightarrow \mathbb{P}(m)$ maximizing the number of assigned robots,¹³

$$g(\mathbf{p}) := \arg \max_{\substack{I \subseteq \{1, \dots, m\}, \\ \mathbf{p} \in \mathcal{D}_I}} |I|, \quad (5.28)$$

such that the resulting hybrid vector field is the same as the “move-to-constrained-centroid” law in (5.24), i.e., for any $\mathbf{p} \in \mathbf{Conf}(Q, \beta + \epsilon)$,

$$\mathbf{u}(\mathbf{p}) = \mathbf{u}^{g(\mathbf{p})}(\mathbf{p}). \quad (5.29)$$

Note that, since a sensory cell with a nonempty interior can not instantaneously appear or disappear under any continuous motion, each time when a local controller is selected by g , the local controller steers the robots for a nonzero time.

Now the continuity properties of each local control policy can be observed as follows. As in the case of Voronoi diagrams [44], we have that the boundary of a sensory cell with a nonempty interior is a piecewise continuously differentiable function of robot locations, and its centroid is continuously differentiable with respect to robot locations. Similarly, the boundary of each element of $\mathcal{F}(\mathbf{p}, \beta + \epsilon)$ is piecewise continuously differentiable, because each free cell is a nonempty erosion of an element of the body diagram $\mathcal{B}(\mathbf{p}, \beta + \epsilon)$ by a fixed closed ball. Hence, one can conclude that each local control policy is piecewise continuously differentiable, because metric projections onto (moving) convex cells are piecewise continuously differentiable [135, 199, 146], and the composition of piecewise continuously differentiable functions are also piecewise continuously differentiable [50].

Therefore, the existence, uniqueness and continuously differentiability of the flow of each local controller \mathbf{u}^I follow from the Lipschitz continuity of \mathbf{u}^I in its compact domain \mathcal{D}_I , because a piecewise continuously differentiable function is also locally Lipschitz on its domain [50] and a locally Lipschitz function on a compact set is globally Lipschitz on that set [127]. Hence, since their domains define a finite open cover of $\mathbf{Conf}(Q, \beta + \epsilon)$, the unique, continuous and piecewise differentiable flow of the “move-to-constrained-centroid” law is constructed by piecing together trajectories of these local policies.

Finally, a natural choice of a Lyapunov function candidate for the stability analysis is the continuously differentiable location optimization function \mathcal{H}_s (5.10), and one can verify from (5.11), (5.22) and (5.24) that for any $\mathbf{p} \in \mathbf{Conf}(Q, \beta + \epsilon)$ ¹⁴

$$\begin{aligned} \dot{\mathcal{H}}_s(\mathbf{p}, \boldsymbol{\sigma}) = -k \sum_{i=1}^m \mu_{S_i} \underbrace{2(\mathbf{p}_i - \mathbf{c}_{S_i})^T (\mathbf{p}_i - \bar{\mathbf{c}}_{S_i})}_{\substack{\geq \|\mathbf{p}_i - \bar{\mathbf{c}}_{S_i}\|^2, \\ \text{since } \mathbf{p}_i \in F_i \text{ and } \|\mathbf{p}_i - \mathbf{c}_{S_i}\|^2 \geq \|\bar{\mathbf{c}}_{S_i} - \mathbf{c}_{S_i}\|^2}} \leq -k \sum_{i=1}^m \mu_{S_i} \|\mathbf{p}_i - \bar{\mathbf{c}}_{S_i}\|^2 \leq 0, \end{aligned} \quad (5.30)$$

which is equal to 0 only if $\mathbf{p}_i = \bar{\mathbf{c}}_{S_i}$ for all i , i.e., $\mathbf{p} \in G_s(Q, \beta + \epsilon, \boldsymbol{\sigma})$. Thus, it follows from LaSalle’s Invariance Principle [127] that all multirobot configurations in $\mathbf{Conf}(Q, \beta + \epsilon)$ asymptotically reach $G_s(Q, \beta + \epsilon, \boldsymbol{\sigma})$. If an equilibrium \mathbf{p}^* in $G_s(Q, \beta + \epsilon, \boldsymbol{\sigma})$ is isolated, then it is guaranteed that $\dot{\mathcal{H}}_s(\mathbf{p}, \boldsymbol{\sigma}) < 0$ in a neighborhood of \mathbf{p}^* , and so it is locally

¹³Here $\mathbb{P}(m)$ denotes the set of all subsets of $\{1, 2, \dots, m\}$.

¹⁴ \mathbf{A}^T is the transpose of matrix \mathbf{A} .

asymptotically stable [108]. ■

As a final remark, we would like to emphasize that, since its construction is only based on power partitions, our “move-to-constrained-centroid” law is distributed in the sense of planar triangulations [20, 137].

5.3.3 Congestion Control of Unassigned Robots

In this subsection, we shall present a heuristic congestion management strategy for unassigned robots that improves assigned robots’ progress.

For a choice of vectors of body radii $\beta \in (\mathbb{R}_{\geq 0})^m$, safety margins $\epsilon \in (\mathbb{R}_{> 0})^m$ and sensory footprint radii $\sigma \in (\mathbb{R}_{\geq 0})^m$, let $\mathbf{p} \in \text{Conf}(Q, \beta + \epsilon)$ be a multirobot configuration in Q with the associated body diagram $\mathcal{B}(\mathbf{p}, \beta + \epsilon) = \{B_1, B_2, \dots, B_m\}$, free subdiagram $\mathcal{F}(\mathbf{p}, \beta + \epsilon) = \{F_1, F_2, \dots, F_m\}$ and sensory diagram $\mathcal{S}(\mathbf{p}, \sigma) = \{S_1, S_2, \dots, S_m\}$.

Consider the following heuristic management of robots: if i th robot has a sensory cell P_i with a nonempty interior, then it is assigned to the coverage task with sensory cell S_i ; otherwise, since the robot becomes redundant for the coverage task, it is assigned to move towards a safe location in B_i . We therefore define the set of “active” domains $\mathcal{A}(\mathbf{p}, \beta + \epsilon, \sigma) = \{A_1, A_2, \dots, A_n\}$ of robots as

$$A_i := \begin{cases} S_i, & \text{if } \overset{\circ}{S}_i \neq \emptyset, \\ B_i, & \text{otherwise.} \end{cases} \quad (5.31)$$

Note that $\mathcal{A}(\mathbf{p}, \beta + \epsilon, \sigma)$ defines a cover of Q and its elements have nonempty interior for all $\mathbf{p} \in \text{Conf}(Q, \beta + \epsilon)$ (Proposition 5.2).

For the first-order robot dynamics (5.7), we propose the following “move-to-constrained-active-centroid” law

$$\mathbf{u}_i = -k(\mathbf{p}_i - \bar{\mathbf{c}}_{A_i}), \quad (5.32)$$

that steers each robot towards the constrained centroid, $\bar{\mathbf{c}}_{A_i}$ as defined in (5.23), of its active domain, A_i , which is the closest point in \overline{F}_i to the centroid \mathbf{c}_{A_i} and so uniquely solves [37]

$$\begin{aligned} & \text{minimize} \quad \|\mathbf{q}_i - \mathbf{c}_{A_i}\|^2 \\ & \text{subject to} \quad \mathbf{q}_i \in \overline{F}_i \end{aligned} \quad (5.33)$$

where \overline{F}_i is convex and $k \in \mathbb{R}_{> 0}$ is a fixed control gain. Once again, we assume that $\mathcal{A}(\mathbf{p}, \beta + \epsilon, \sigma)$ and $\mathcal{F}(\mathbf{p}, \beta + \epsilon)$ are continuously updated. It is also useful to have $G_{\mathcal{A}}(Q, \beta + \epsilon, \sigma)$ denote the set of equilibria of the “move-to-constrained-active-centroid” law where robots are located at the constrained centroid of their active domains,

$$G_{\mathcal{A}}(Q, \beta + \epsilon, \sigma) := \left\{ \mathbf{p} \in \text{Conf}(Q, \beta + \epsilon) \mid \mathbf{p}_i = \bar{\mathbf{c}}_{A_i} \ \forall i \right\}. \quad (5.34)$$

We summarize some important properties of our “move-to-constrained-active-centroid” law as follows:

Proposition 5.3 *For any choice of vectors of body radii $\mathbf{r} \in (\mathbb{R}_{\geq 0})^m$, safety margins $\alpha \in (\mathbb{R}_{> 0})^m$ and sensory radii $\sigma \in (\mathbb{R}_{\geq 0})^m$, the “move-to-constrained-active-centroid” law in (5.32) leaves the configuration space of nonintersecting disks $\text{Conf}(Q, \beta + \epsilon)$ positively*

invariant; and its unique, continuous and piecewise differentiable flow, starting at any configuration in $\text{Conf}(Q, \beta + \epsilon)$, asymptotically reaches $G_A(Q, \beta + \epsilon, \sigma)$ without increasing the utility function $\mathcal{H}_s(\cdot, \sigma)$ (5.10) along the way.

Proof The positive invariance of $\text{Conf}(Q, \beta + \epsilon)$ under the “move-to-constrained-active-centroid” law and the existence, uniqueness and continuity properties of its flow follow the same pattern as established in Theorem 5.1.

For the stability analysis, using (5.11), (5.32) and (5.33), one can show that the continuously differentiable utility function $\mathcal{H}_s(\cdot, \sigma)$ (5.10) is nonincreasing along the trajectory of the “move-to-constrained-active-centroid” law as follows:

$$\begin{aligned} \dot{\mathcal{H}}_s(\mathbf{p}, \sigma) &= -k \sum_{\substack{i \in \{1, \dots, m\} \\ \dot{S}_i \neq \emptyset}} \mu_{S_i} \underbrace{2(\mathbf{p}_i - \mathbf{c}_{S_i})^T (\mathbf{p}_i - \bar{\mathbf{c}}_{S_i})}_{\substack{\geq \|\mathbf{p}_i - \bar{\mathbf{c}}_{S_i}\|^2, \\ \text{since } \mathbf{p}_i \in F_i \text{ and } \|\mathbf{p}_i - \mathbf{c}_{S_i}\|^2 \geq \|\bar{\mathbf{c}}_{S_i} - \mathbf{c}_{S_i}\|^2}} - k \sum_{\substack{i \in \{1, \dots, m\} \\ \dot{S}_i = \emptyset}} \underbrace{\mu_{S_i}}_{=0} 2(\mathbf{p}_i - \mathbf{c}_{S_i})^T (\mathbf{p}_i - \bar{\mathbf{c}}_{B_i}), \\ &\leq -k \sum_{\substack{i \in \{1, \dots, m\} \\ \dot{S}_i \neq \emptyset}} \mu_{S_i} \|\mathbf{p}_i - \bar{\mathbf{c}}_{S_i}\|^2 \leq 0. \end{aligned} \quad (5.35)$$

Hence, we have from Lasalle’s Invariance Principle [127] that, at an equilibrium point of the “move-to-constrained-active-centroid” law, a robot is located at the constrained centroid, $\bar{\mathbf{c}}_{S_i}$, of its sensory cell, S_i , if it has a nonempty interior, i.e., $\dot{S}_i \neq \emptyset$. Given that $\mathbf{p}_i = \bar{\mathbf{c}}_{S_i}$ for all $i \in \{1, \dots, m\}$ with $\dot{S}_i \neq \emptyset$, using (5.11), (5.32) and (5.33), one can further show that

$$\begin{aligned} \dot{\mathcal{H}}_B(\mathbf{p}, \beta + \epsilon) &= -k \sum_{\substack{i \in \{1, \dots, n\} \\ \dot{S}_i \neq \emptyset}} \mu_{B_i} \underbrace{2(\mathbf{p}_i - \mathbf{c}_{B_i})^T (\mathbf{p}_i - \bar{\mathbf{c}}_{S_i})}_{\substack{=0, \\ \text{since } \mathbf{p}_i = \bar{\mathbf{c}}_{S_i}}} - k \sum_{\substack{i \in \{1, \dots, n\} \\ \dot{S}_i = \emptyset}} \mu_{B_i} \underbrace{2(\mathbf{p}_i - \mathbf{c}_{B_i})^T (\mathbf{p}_i - \bar{\mathbf{c}}_{B_i})}_{\substack{\geq \|\mathbf{p}_i - \bar{\mathbf{c}}_{B_i}\|^2, \\ \text{since } \mathbf{p}_i \in F_i \text{ and } \|\mathbf{p}_i - \mathbf{c}_{B_i}\|^2 \geq \|\bar{\mathbf{c}}_{B_i} - \mathbf{c}_{B_i}\|^2}}, \\ &\leq -k \sum_{\substack{i \in \{1, \dots, n\} \\ \dot{S}_i = \emptyset}} \mu_{B_i} \|\mathbf{p}_i - \bar{\mathbf{c}}_{B_i}\|^2 \leq 0. \end{aligned} \quad (5.36)$$

Therefore, at a stationary point of (5.32) i th robot is located at the constrained centroid, $\bar{\mathbf{c}}_{B_i}$, of its body cell B_i if $\dot{S}_i = \emptyset$. Overall, by Lasalle’s Invariance Principle, we have that any multirobot configuration starting in $\text{Conf}(Q, \beta + \epsilon)$ asymptotically converges to a locally optimal sensing configuration in $G_A(Q, \beta + \epsilon, \sigma)$, which completes the proof. ■

5.3.4 Coverage Control of Differential Drive Robots

We now present another extension of our “move-to-constrained-centroid” law of Section 5.3.2 for the widely used kinematic differential drive model.

Consider a noncolliding placement of a heterogeneous group of disk-shaped differential drive robots $(\mathbf{p}, \theta) \in \text{Conf}(Q, \beta + \epsilon) \times (-\pi, \pi]^n$ in a convex planar environment $Q \subset \mathbb{R}^2$ with associated vectors of body radii $\beta \in (\mathbb{R}_{\geq 0})^n$, safety margins $\epsilon \in (\mathbb{R}_{> 0})^n$ and sensory footprint radii $\sigma \in (\mathbb{R}_{\geq 0})^n$, where $\theta = (\theta_1, \theta_2, \dots, \theta_n)$ is the vector of robot orientations.

The kinematic equations describing the motion of each differential drive robot are

$$\dot{\mathbf{p}}_i = v_i \begin{bmatrix} \cos \theta_i \\ \sin \theta_i \end{bmatrix}, \quad \text{and} \quad \dot{\theta}_i = \omega_i, \quad (5.37)$$

where $v_i \in \mathbb{R}$ and $\omega_i \in \mathbb{R}$ are, respectively, the linear (tangential) and angular velocity inputs of i th robot. Note that the differential drive model is underactuated due to the nonholonomic constraint $\begin{bmatrix} -\sin \theta_i \\ \cos \theta_i \end{bmatrix}^T \dot{\mathbf{p}}_i = 0$.

Let $\mathcal{S}(\mathbf{p}, \boldsymbol{\sigma}) = \{S_1, S_2, \dots, S_m\}$ (5.9) be the sensory diagram of Q based on robot locations \mathbf{p} and sensory footprint radii $\boldsymbol{\sigma}$, and $\mathcal{F}(\mathbf{p}, \boldsymbol{\beta} + \boldsymbol{\epsilon}) = \{F_1, F_2, \dots, F_m\}$ (5.20) be the free subdiagram associated with configuration \mathbf{p} and enlarged body radii $\boldsymbol{\beta} + \boldsymbol{\epsilon}$. For a choice of $\boldsymbol{\epsilon} \in (\mathbb{R}_{>0})^n$ with $\epsilon_i > \epsilon_i$ for all i , we further define $\mathcal{T}(\mathbf{p}, \boldsymbol{\beta} + \boldsymbol{\epsilon}) = \{T_1, T_2, \dots, T_m\}$ to be

$$T_i := \text{conv}(\{\mathbf{p}_i\} \cup F'_i), \quad (5.38)$$

where $\mathcal{F}(\mathbf{p}, \boldsymbol{\beta} + \boldsymbol{\epsilon}) = \{F'_1, F'_2, \dots, F'_m\}$ and $\text{conv}(A)$ denotes the convex hull of set A . Note that, since $F'_i \subset F_i$, $\mathbf{p}_i \in F_i$ and F_i is convex, every element of $\mathcal{T}(\mathbf{p}, \boldsymbol{\beta} + \boldsymbol{\epsilon})$ is contained in the associated element of $\mathcal{F}(\mathbf{p}, \boldsymbol{\beta} + \boldsymbol{\epsilon})$, i.e., $T_i \subseteq F_i$. It is useful to remark that we particularly require $\mathbf{p}_i \in T_i$ to guarantee an optimal choice of a local target position in (5.42) relative to \mathbf{p}_i , and we construct subset T_i of F_i to increase the convergence rate of our proposed coverage control law in (5.44), by guaranteeing that the robot can nearly align its orientation with its target location in finite time.

As in the case of “move-to-constrained-centroid” law of fully actuated robots in (5.24), for optimal coverage, each differential drive robot will intent to move towards the constrained centroid, $\bar{\mathbf{c}}_{S_i}$ (5.23), of its sensory cell, S_i , but with a slight difference due to the nonholonomic constraint. To determine a linear velocity input that guarantees collision avoidance and conforms to the nonholonomic constraint, we select a safe target location that solves the following convex programming,

$$\begin{aligned} & \text{minimize} \quad \|\mathbf{q}_i - \mathbf{c}_{S_i}\|^2 \\ & \text{subject to} \quad \mathbf{q}_i \in \bar{F}_i \cap H_i \end{aligned} \quad (5.39)$$

where

$$H_i := \left\{ \mathbf{x} \in Q \mid \begin{bmatrix} -\sin \theta_i \\ \cos \theta_i \end{bmatrix}^T (\mathbf{x} - \mathbf{p}_i) = 0 \right\}, \quad (5.40)$$

is the straight line motion range due to the nonholonomic constraint. Note that $\bar{F}_i \cap H_i$ is a closed line segment in Q . Hence, once again, the unique solution of (5.39) is given by

$$\bar{\mathbf{c}}_{S_i}^v := \begin{cases} \mathbf{c}_{S_i}, & \text{if } \mathbf{c}_{S_i} \in \bar{F}_i \cap H_i, \\ \Pi_{\bar{F}_i \cap H_i}(\mathbf{c}_{S_i}), & \text{otherwise,} \end{cases} \quad (5.41)$$

where Π_C is the metric projection map onto a convex set C . Similarly, to determine robot's angular motion, we select another safe target location that solves

$$\begin{aligned} & \text{minimize} \quad \|\mathbf{q}_i - \mathbf{c}_{S_i}\|^2 \\ & \text{subject to} \quad \mathbf{q}_i \in \bar{T}_i \end{aligned} \quad (5.42)$$

where $\overline{T}_i \subset \overline{F}_i$ is convex, and the unique solution of (5.42) is

$$\overline{c}_{S_i}^\omega := \begin{cases} c_{S_i}, & \text{if } c_{S_i} \in \overline{T}_i, \\ \Pi_{\overline{T}_i}(c_{S_i}), & \text{otherwise.} \end{cases} \quad (5.43)$$

Accordingly, based on a standard differential drive controller [19], we propose the following “move-to-constrained-centroid” law for differential drive robots,¹⁵

$$v_i = -k \begin{bmatrix} \cos \theta_i \\ \sin \theta_i \end{bmatrix}^T (\mathbf{p}_i - \overline{c}_{S_i}^v), \quad (5.44a)$$

$$\omega_i = k \operatorname{atan} \left(\frac{\begin{bmatrix} -\sin \theta_i \\ \cos \theta_i \end{bmatrix}^T (\mathbf{p}_i - \overline{c}_{S_i}^\omega)}{\begin{bmatrix} \cos \theta_i \\ \sin \theta_i \end{bmatrix}^T (\mathbf{p}_i - \overline{c}_{S_i}^\omega)} \right), \quad (5.44b)$$

where $k > 0$ is fixed. Note that the linear velocity control v_i is a function of the constrained centroid $\overline{c}_{S_i}^v$ (5.41) of the associated sensory cell handling both collision avoidance and the nonholonomic constraint whereas the angular velocity control ω_i is a function of the constrained centroid $\overline{c}_{S_i}^\omega$ (5.43) of the associated power cell only handling collision avoidance.

Having $G_{\mathcal{D}}(Q, \beta, \epsilon, \varepsilon, \sigma)$ denote its set of stationary points, where the constrained centroids $\overline{c}_{S_i}^v$ and $\overline{c}_{S_i}^\omega$ coincide, and i th robot is located at $\overline{c}_{S_i}^v = \overline{c}_{S_i}^\omega$,

$$G_{\mathcal{D}}(Q, \beta, \epsilon, \varepsilon, \sigma) := \left\{ \mathbf{p} \in \operatorname{Conf}(Q, \beta + \epsilon) \mid \mathbf{p}_i = \overline{c}_{S_i}^v = \overline{c}_{S_i}^\omega \ \forall i \right\},$$

we summarize important qualitative properties of the “move-to-constrained-centroid” law of differential drive robots as:

Proposition 5.4 *For any $\beta, \sigma \in (\mathbb{R}_{\geq 0})^m$ and $\epsilon, \varepsilon \in (\mathbb{R}_{> 0})^m$ with $\epsilon_i < \varepsilon_i$ for all i , the “move-to-constrained-centroid” law of differential drive robots in (5.44) asymptotically steers all configurations in its positively invariant domain $\operatorname{Conf}(Q, \beta + \epsilon) \times (-\pi, \pi]^m$ towards the set of optimal sensing configurations $G_{\mathcal{D}}(Q, \beta, \epsilon, \varepsilon, \sigma) \times (-\pi, \pi]^m$ without increasing the utility function $\mathcal{H}_S(\cdot, \sigma)$ (5.10) along the way.*

Proof The configuration space $\operatorname{Conf}(Q, \beta + \epsilon) \times (-\pi, \pi]^m$ is positively invariant under the “move-to-constrained-centroid” law in (5.44), because, by construction (5.39), each robot’s motion is constrained to the associated safe partition subcell in Q . The existence and uniqueness of its flow can be established using the pattern of the proof of Theorem 5.1 and the flow properties of the differential drive controller in [19].

Now, using $\mathcal{H}_S(\cdot, \sigma)$ (5.10) as a continuously differentiable Lyapunov function candidate, we obtain the stability properties as follows: for any $(\mathbf{p}, \theta) \in \operatorname{Conf}(Q, \beta + \epsilon) \times (-\pi, \pi]^m$

$$\begin{aligned} \dot{\mathcal{H}}_S(\mathbf{p}, \sigma) &= -k \sum_{i=1}^m \mu_{S_i} \underbrace{2(\mathbf{p}_i - c_{S_i})^T (\mathbf{p}_i - \overline{c}_{S_i}^v)}_{\geq \|\mathbf{p}_i - \overline{c}_{S_i}^v\|^2} \leq -k \sum_{i=1}^m \mu_{S_i} \|\mathbf{p}_i - \overline{c}_{S_i}^v\|^2 \leq 0, \quad (5.45) \\ &\text{since } \mathbf{p}_i \in F_i \cap H_i \text{ and } \|\mathbf{p}_i - c_{S_i}\|^2 \geq \|\overline{c}_{S_i}^v - c_{S_i}\|^2 \end{aligned}$$

¹⁵To resolve indeterminacy we set $\omega_i = 0$ whenever $\mathbf{p}_i = \overline{c}_{S_i}^\omega$.

where $\dot{\mathbf{p}}_i = -k(\mathbf{p}_i - \bar{\mathbf{c}}_{S_i}^v)$. Hence, by LaSalle's Invariance Principle [127], at a stationary point of (5.44) i th robot is located at $\bar{\mathbf{c}}_{S_i}^v$. Since, for fixed $\bar{\mathbf{c}}_{S_i}^v$ and $\bar{\mathbf{c}}_{S_i}^\omega$, the standard differential drive controller asymptotically aligns each robot with the constrained centroid $\bar{\mathbf{c}}_{S_i}^\omega$, i.e., $\begin{bmatrix} -\sin \theta_i \\ \cos \theta_i \end{bmatrix}^T (\mathbf{p}_i - \bar{\mathbf{c}}_{S_i}^\omega) = 0$ [19], it is guaranteed by (5.39) and (5.42) that $\bar{\mathbf{c}}_{S_i}^v = \bar{\mathbf{c}}_{S_i}^\omega$ whenever $\|\mathbf{p}_i - \bar{\mathbf{c}}_{S_i}^v\| = 0$ and $\begin{bmatrix} -\sin \theta_i \\ \cos \theta_i \end{bmatrix}^T (\mathbf{p}_i - \bar{\mathbf{c}}_{S_i}^\omega) = 0$. Therefore, we have from LaSalle's Invariance Principle that all configurations in $\text{Conf}(Q, \beta + \epsilon) \times (-\pi, \pi]^m$ asymptotically reach $G_{\mathcal{D}}(Q, \beta, \epsilon, \epsilon, \sigma) \times (-\pi, \pi]^m$. ■

Finally, note that the “move-to-constrained-active-centroid” law of Section 5.3.3 can be utilized for congestion control of differential drive robots by using active domains in (5.31) instead of the sensory diagram $\mathcal{S}(\mathbf{p}, \sigma)$, and the resulting coverage law maintains qualitative properties.

5.4 Numerical Simulations

A common source of collisions between robots performing a distributed sensing task is a concentrated event distribution, which generally causes robots to move towards the same small region of the environment.¹⁶ We therefore consider the following event distribution, $\phi : [0, 10]^2 \rightarrow \mathbb{R}_{>0}$, for a homogeneous group of disk-shaped robots operating in a 10×10 square environment,

$$\phi(\mathbf{q}) = e^{-\left\| \mathbf{q} - \begin{bmatrix} 7 \\ 7 \end{bmatrix} \right\|^2}. \quad (5.46)$$

In Figure 5.3, we present the resulting trajectories of our proposed coverage control algorithms for a group of robots starting at around the bottom left corner of the environment. Since the event distribution is concentrated around a small region, as expected, the standard “move-to-centroid” law steers robots to a centroidal Voronoi configuration where robots collide. On the other hand, since a Voronoi partition has no occupancy defect, our “move-to-constrained-centroid” and “move-to-constrained-active-centroid” laws yield the same trajectory that asymptotically converges a collision-free constrained centroidal Voronoi configuration. It is also well known that minimizing the location optimization function \mathcal{H}_S (5.10) generally results in a locally optimal sensing configuration, and we observe in Figures 5.3.(c) and 5.3.(e) that, although they are initiated at the same location, fully actuated and differential drive robots asymptotically reach different constrained centroidal Voronoi configurations.

To demonstrate how unassigned robots may limit the mobility of others, we consider a heterogeneous group of disk-shaped robots operating in a 10×10 environment with the following event distribution function, $\phi : [0, 10]^2 \rightarrow \mathbb{R}_{>0}$,

$$\phi(\mathbf{q}) = 1 + 10e^{-\frac{1}{9}\left\| \mathbf{q} - \begin{bmatrix} 8 \\ 8 \end{bmatrix} \right\|^2} + e^{-\frac{1}{2}\left\| \mathbf{q} - \begin{bmatrix} 8 \\ 2 \end{bmatrix} \right\|^2} + e^{-\frac{1}{2}\left\| \mathbf{q} - \begin{bmatrix} 8 \\ 4 \end{bmatrix} \right\|^2} + e^{-\left\| \mathbf{q} - \begin{bmatrix} 3 \\ 7 \end{bmatrix} \right\|^2}, \quad (5.47)$$

¹⁶For all simulations, we set $k = 1$, $\epsilon_i = 0.05$ and $\varepsilon_i = 0.1$ for all $i \in \{1, 2, \dots, m\}$, and all simulations are obtained through numerical integration of the associated coverage control law using the `ode45` function of MATLAB, and the computation of the centroid of a power cell in (5.6) is approximated by discretizing the power cell by a 20×20 grid.

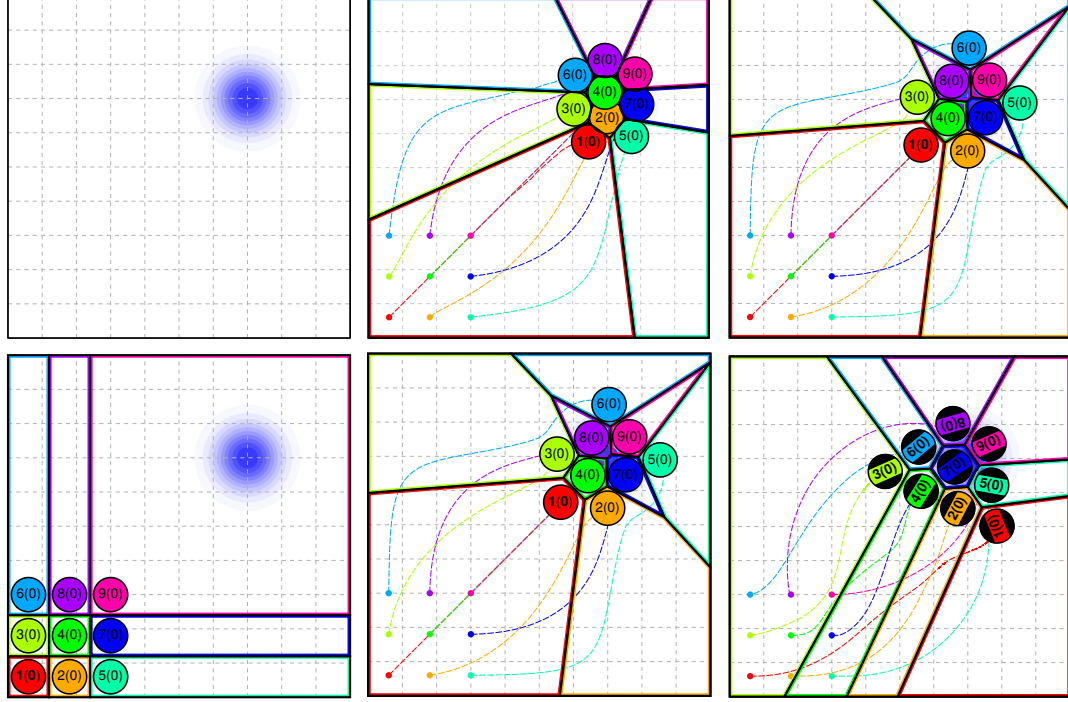


Figure 5.3: Avoiding collisions around a concentrated event distribution. (top, left) Spatial event distribution, (bottom, left) Initial configuration of a homogeneous robot network, where the weight of sensory cell are shown in the parenthesis, and the resulting trajectories of (top, middle) the standard “move-centroid” law (5.12), (bottom, middle) the “move-to-constrained-centroid” law (5.24), (top, right) the “move-to-constrained-active-centroid” law (5.32), (bottom, right) the “move-to-constrained-centroid” law of differential drive robots (5.44) which are initially aligned with the horizontal axis.

which is also used in [137]. In Figure 5.4, we illustrate the resulting trajectories of our safe coverage control algorithms for a heterogeneous groups of robots starting at around the left bottom corner of the environment. As seen in Figure 5.4.(a), the 2nd robot is initially not assigned to any region. It stays stationary for a certain finite time under the the standard “move-to-centroid” law during which the 1st robot moves through it. Also notice that the 3rd robot violates the workspace boundary before converging a safe location. In summary, the “move-to-centroid” law steers disk-shaped robots to a locally optimal sensing configuration without avoiding collisions along the way. Our “move-to-constrained-centroid” law prevents any possible self-collisions and collisions with the boundary of the environment. However, since the 2nd robot stays unassigned for all future time, the 1st robot is blocked and it can not move to a better coverage location. Fortunately, while guaranteeing collision avoidance, our “move-to-constrained-active-centroid” law steers unassigned robots to improve assigned robots’ progress for both fully actuated and differential drive robots, as illustrated in Figures 5.4.(d) and 5.4.(e), respectively.

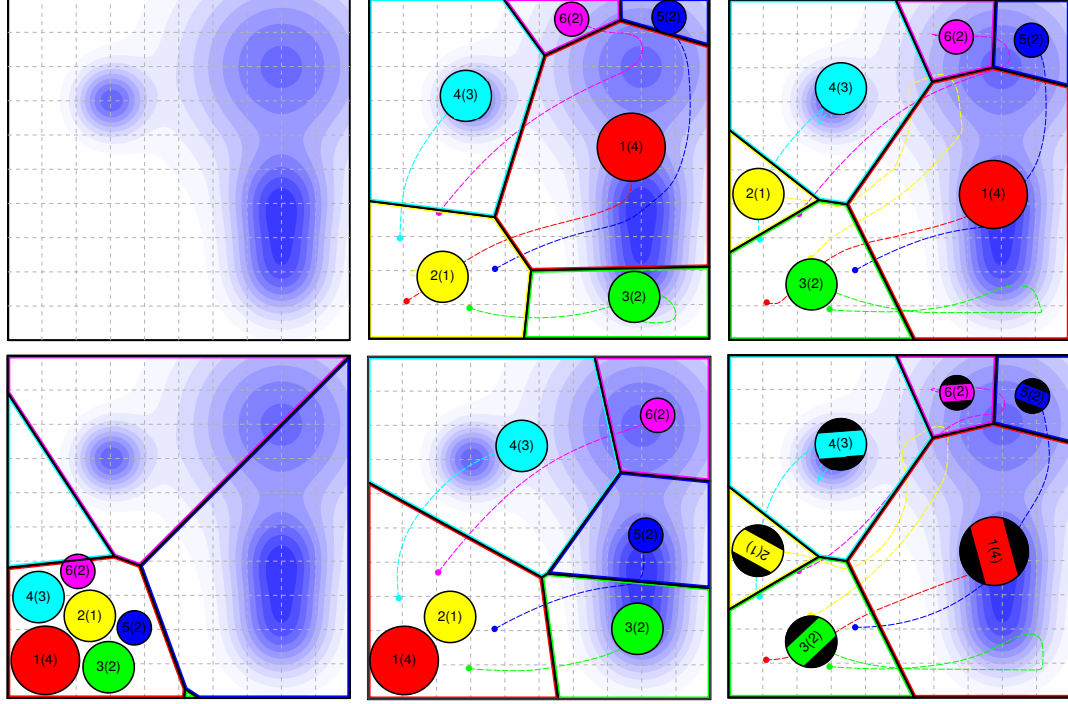


Figure 5.4: Safe coverage control of heterogeneous disk-shaped robots with a heuristic management of unassigned robots. (top, left) Spatial event distribution, (top, bottom) Initial configuration of a heterogeneous robot network, where the weight of sensory cell are shown in the parenthesis, and the resulting trajectories of (top, middle) the standard “move-centroid” law (5.12), (bottom, middle) the “move-to-constrained-centroid” law (5.24), (top, right) the “move-to-constrained-active-centroid” law (5.32), (bottom, right) the “move-to-constrained-active-centroid” law of differential drive robots which are initially aligned with the horizontal axis.

5.5 Summary

In distributed mobile sensing applications, networks of agents that are heterogeneous, respecting both actuation as well as body and sensory footprint, are often modelled by recourse to power diagrams — generalized Voronoi diagrams with additive weights [20]. In this chapter, we introduce a novel use of power diagrams for identifying collision-free multi-robot configurations, and propose a constrained optimization framework combining coverage control and collision avoidance for fully actuated disk-shaped robots, comprising the main contributions of the chapter. We also present its extensions for the widely used differential drive model and for heuristic congestion management of unassigned robots, while maintaining the convergence and collision guarantees. Numerical simulations demonstrate the effectiveness of the proposed coverage control algorithms.



Chapter 6

Conclusions and Future Work

While it is essential to model and understand the topologies of configuration spaces in order to design provably correct motion planners, configuration spaces arising in robotics generally have complex shapes and high dimensions making it very difficult, if not impossible, to explicitly model and study the underlying space, which is the technical challenge that inspired this dissertation. We herein argued that automatically discovered intrinsic structures in configuration spaces via clustering has the potential to afford computationally efficient tools for modelling configuration spaces and for the design of provably correct motion planners. This new philosophy for modelling configuration spaces, still in its infancy, yields promising results for closing the gap between configuration space and sampling-based motion planning approaches — while the former is based on an explicit construction of a configuration space, the latter is based on a dense collection of sample configurations connected by simple paths.

Chapter 2 introduced the use of hierarchical clustering for relaxed, deterministic coordination and control of multiple robots. Hierarchical clustering [115] offers a natural abstraction for ensemble task encoding and control in terms of precise yet flexible organizational specifications at different resolutions, by relating the continuous space of configurations to the combinatorial space of trees. This hierarchical abstraction intrinsically suggests a two-level navigation strategy for coordinated motion design: (i) at the low-level, perform finer adjustments on configurations by using hierarchy-preserving vector fields; and (ii) at the high-level, resolve structural conflicts between configurations by using a discrete transition policy in tree space. Accordingly, we proposed a provably correct generic hierarchical navigation framework for collision-free coordinated motion design towards any given destination via a sequence of hierarchy-preserving controllers recursively deployed on-the-fly according to a discrete navigation rule in the space of cluster hierarchies. For a choice of a hierarchical clustering algorithm, we demonstrated a computationally efficient instantiation of our hierarchical navigation framework for coordinated control of an arbitrary number of disk-shaped robots operating in an ambient Euclidean space. The contributions of Chapter 2 that enable an efficient solution to the coordinated motion planning problem are: (i) a topologically nontrivial abstraction that relates complex-shaped and high-dimensional continuous space of multirobot configurations to the combinatorial space of trees with the nearest neighbor interchange (NNI) connectivity, (ii) an explicit characterization of the topology of configurations sharing the same cluster hierarchy, (iii) a recursively defined vec-

tor field for hierarchy-preserving navigation, and (iv) a recursive algorithm that iteratively constructs a navigation path in tree space.

Chapter 3 presented a comprehensive study of navigation in tree space, because, besides our interest motivated by independent applications in coordinated robot navigation, navigation in tree space is a fundamental classification problem common to both computational biology and engineering, for construction of descriptive hierarchical organizational models and for efficient informative comparison of hierarchical structures. Using the fact that a binary cluster hierarchy is a maximal collection of compatible clusters [195] and two distinct binary hierarchies always have some incompatible clusters, we proposed an NNI navigation algorithm that recursively identifies and fixes incompatibilities between clusters of trees. We showed that for trees with n leaves, our NNI navigation algorithms require $O(n)$ time for each iteration and the length of a navigation path, defined as the NNI navigation dissimilarity d_{nav} , between any pair of binary trees is at most $\frac{1}{2}(n-1)(n-2)$. Hence, d_{nav} turns out to be an efficient approximation to the (NP-hard) NNI distance d_{NNI} in terms of “edit length”. At the same time, a closed form formula we derived for d_{nav} presents it as a weighted count of pairwise incompatibilities between clusters of trees, lending it the character of an edge dissimilarity measure as well. A relaxation of this formula to a simple count of pairwise incompatibilities of clusters of trees yields another measure, the crossing dissimilarity d_{CM} . Both dissimilarities are symmetric and positive definite (vanish only between identical trees) on binary hierarchies but they fail to satisfy the triangle inequality. Nevertheless, both are bounded below by the widely used Robinson-Foulds metric d_{RF} and bounded above by a newly proposed true metric, the cluster-cardinality metric d_{CC} . We showed that each of the proposed new dissimilarities is computable in time $O(n^2)$ in the number of leaves n , and we revealed the following order relation between these tree measures,

$$\frac{2}{3}d_{RF} \leq \frac{2}{3}d_{NNI} \leq \frac{2}{3}d_{nav} \leq d_{CM} \leq d_{CC} . \quad (6.1)$$

Chapter 4 introduced a novel application of separating hyperplanes of convex bodies to the problem of sensor-based reactive robot navigation in unknown convex sphere worlds, i.e., to construct a vector field over a compact convex Euclidean subset, populated with unknown but sufficiently separated and strongly convex obstacles, whose flow brings a disk-shaped robot from all but a zero measure set of initial conditions to a designated point destination, with the guarantee of no collisions along the way. We utilized separating hyperplanes of convex bodies to identify a collision-free convex neighborhood of the robot, and proved that the continuous feedback motion towards the metric projection of the goal location onto this convex neighborhood solves the specified robot navigation problem. We also proposed its practical extensions to two limited range sensing models and the nonholonomically constrained kinematics of the standard differential drive vehicle, while maintaining the provable properties.

Chapter 5 presented a constrained coverage control law for heterogeneous disk-shaped robots that solves the combined sensory coverage and collision avoidance problem. In addition to their standard usages in distributed mobile sensing applications for solving sensory task assignment and for modelling group heterogeneity in actuation, sensing, computation

and energy sources [60, 137, 178], in this chapter we tailored Voronoi diagrams to precisely encode multirobot collisions. We further presented practical extensions (a heuristic congestion manager that speeds convergence and a lift of the point particle controller to the more practical differential drive kinematics) that maintain the convergence and collision avoidance guarantees.

We believe that the potential applications of clustering to feedback motion design presented in this dissertation only scratch the surface of its potential in the long run. Some future extensions to these results are summarized below.

An interesting direction for future research is to extend our hierarchical navigation framework presented in Chapter 2 to more practical settings in the field of robotics including navigating around obstacles in compact spaces and a distributed implementation of our navigation framework. Especially, when the scalability and efficiency of hierarchical protocols in sensor networks for information routing and aggregation is of concern [3], these hierarchical methods suggest a promising unifying framework to simultaneously handle control [18], communication[11] and information aggregation (fusion)[10] in multi-agent systems. What is more, as opposed to rigidly imposed goal specifications, future work should investigate clustering-based task specifications of multirobot systems, affording precise yet flexible formation specifications while ignoring fine details of individual positioning.

Promising future research based on the results presented in Chapter 3 includes applications in perception, learning and anomaly detection where navigation in tree space might play a key role for construction of acyclic graphical models and their informative comparison. We believe that discriminative comparison of cluster hierarchies are likely of some significance for learning graphical models and contextual scene understanding.

A potential extension to the current results presented in Chapter 4 is a fully smoothed version of our sensor-based feedback motion planner, permitting its lift to more complicated dynamical models such as force-controlled (second-order) and more severely underactuated systems. This will enable its empirical demonstration for safe, high-speed navigation in a forest-like environments and in human crowds. Future work can also focus on the extension of these ideas for coordinated, decentralized feedback control of multirobot swarms. More generally, we seek to identify fundamental limits on navigable environments for a memoryless greedy robotic agent with a limited range sensing capability. Another exciting future direction is combining a discrete-time version of our feedback motion planner (see Appendix C.5) with another (e.g., sampling-based) motion planning algorithm to solve online robot navigation problem in nonconvex environments.

Future extensions of safe Voronoi-based coverage control presented in Chapter 5 can target hierarchical settings, based on nested partitions of convex environments [16], and nonconvex environments. Combining coverage control and other collective behaviours (for example, circular formations [198]) for active sensing is an interesting direction for future research.

In the longer term, clustering-based robot motion planning and control is a rich research environment with a large number of open problems, because while clustering offers various algorithms to identify different structural patterns in configuration spaces, robotics presents diverse types of robots and workspaces with varied topological properties. Moreover, abstract relations between continuous space of configurations and the combinatorial

space of clustering models via clustering might have a significant value for the systematic enumeration of intrinsic structural patterns in configuration spaces. For example, although the continuous configuration space of a centipede-like multiple legged robot with rotating legs (e.g., RHex [192]) is a torus [26], its locomotion pattern is usually described in terms of combinatorial gait specifications based on the contact state of each leg [105, 118]. Hence, an interesting research problem is how clustering can be exploited for descriptive high-level modelling of gait transition and coordination in such robots.

Moreover, increasing interest in data-driven methods in robotics makes clustering a key factor in robot motion design. In particular, automatic learning of motor skills and control from large unlabelled data sets inevitably requires clustering for discovering hidden structural motion patterns within data. Spatio-temporal consistency of many sensory phenomena in robotics also motivates the use of clustering in robot perception for anomaly detection, video segmentation and contextual scene understanding, which we want to explore in future.



Appendix A

Coordinated Robot Navigation: Proof Details

A.1 Properties of the Hierarchy-Invariant Vector Field

Although the recursive definition of the hierarchy-preserving navigation policy $f_{\tau, \mathbf{y}}$ in Table 2.5 expresses an efficient encoding of intracluster and intercluster interactions and dependences 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, as anticipated from the proof structure of Theorem 2.4 in Table A.1. 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 [45] over the covering family.

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

Proposition A.1 *The recursion in Table 2.5 results in a well defined function, $f_{\tau, \mathbf{y}} : \mathfrak{S}(\tau) \rightarrow (\mathbb{R}^n)^S$, that can be computed for each configuration $\mathbf{x} \in \mathfrak{S}(\tau)$ in $O(|S|^2)$ time.*

Proof See Appendix A.2.2. ■

A.1.1 An Equivalent System Model

Key for understanding the hierarchy-preserving navigation policy $f_{\tau, \mathbf{y}}$ in Table 2.5 is the observation that for any configuration $\mathbf{x} \in \mathfrak{S}(\tau)$, the list of visited clusters of τ satisfying base conditions during the recursive computation of $f_{\tau, \mathbf{y}}$ defines a partition \mathcal{S} of S compatible with τ , i.e., $\mathcal{S} \subset \mathcal{C}(\tau)$.¹

¹Note that the recursions in Table 2.5 and Table A.3 have the same base and recursion conditions, and the recursion in Table A.3 returns the list of clusters satisfying base conditions, which defines a partition of S (see Proposition A.2). Hence, using the relation between these recursions in Proposition A.4, one can conclude this observation.

Table A.1: Proof Structure of Theorem 2.4 : Logical Dependences

<ul style="list-style-type: none"> • Proposition A.1 (Quadratic Time Function) [A.1, p.123 \Leftarrow A.2.2, p.134] • Proposition A.2 (Continuous & Piecewise Smooth) [A.1.1, p.125 \Leftarrow A.2.3,p.136] <ul style="list-style-type: none"> – Lemma A.4 (Child Partition Block) [A.1.6, p.133] • Proposition A.3 (Domain Covering Induced Partition) [A.1.1, p.125 \Leftarrow A.2.4, p.137] • Proposition A.4 (Equivalent Vector Field) [A.1.1, p.126 \Leftarrow A.2.5, p.138] • Proposition A.11 (Stratum Positive Invariance) [A.1.4, p.130] <ul style="list-style-type: none"> – Recalls Proposition A.3, Proposition A.4 – Proposition A.7 (Substratum Positive Invariance) [A.1.3, p.129 \Leftarrow A.2.9, p.140] <ul style="list-style-type: none"> * Lemma A.6 (Invariance - Base Case 1) [A.2.9, p.140 \Leftarrow A.2.13, p.145] * Lemma A.7 (Invariance - Base Case 2) [A.2.9, p.141 \Leftarrow A.2.14, p.147] * Lemma A.8 (Invariance - Recursion) [A.2.9, p.141 \Leftarrow A.2.15, p.148] • Proposition A.12 (Stratum Existence & Uniqueness) [A.1.4, p.130] <ul style="list-style-type: none"> – Recalls Proposition A.3, Proposition A.4 – Proposition A.8 (Substratum Existence Uniqueness) [A.1.3, p.129 \Leftarrow A.2.8, p.140] <ul style="list-style-type: none"> * Recalls Proposition A.2, Proposition A.7. * Lemma A.2 (Relative Centroidal Dynamics) [A.1.6, p.132 \Leftarrow A.2.11,p.142] * Lemma A.3 (Configuration Bound Radius) [A.1.6, p.A.3 \Leftarrow A.2.12, p.143] <ul style="list-style-type: none"> • Recalls Lemma A.2. – Proposition A.13 (Stratum Stability) [A.1.4, p.131] <ul style="list-style-type: none"> * Recalls Proposition A.4, Proposition A.7. * Proposition A.6 (Substratum Policy Selection) [A.1.3, p.129 \Leftarrow A.2.7, p.139] <ul style="list-style-type: none"> • Recalls Proposition A.3. • Lemma A.5 (Partition Refinement) [A.1.6, p.133] * Proposition A.9 (Finite Time Prepares Relation) [A.1.3, p.130 \Leftarrow A.2.10, p.141] <ul style="list-style-type: none"> • Lemma A.9 (Case (i) in Definition A.2) [A.2.10, p.141 \Leftarrow A.2.16, p.A.2.16] • Lemma A.10 (Case (ii) in Definition A.2) [A.2.10, p.142 \Leftarrow A.2.17, p.A.2.17] • Lemma A.11 (Case (iii) in Definition A.2) [A.2.10, p.142 \Leftarrow A.2.18, p.A.2.18] • Proposition A.5 (Substratum Prepares Graph) [A.1.2, p.130 \Leftarrow A.2.6, p.138] <ul style="list-style-type: none"> – Recalls Lemma A.4. • Proposition A.10 (Nondegenerate Execution) [A.1.3, p.130] <ul style="list-style-type: none"> – Recalls Proposition A.6, Proposition A.7. – Lemma A.1 (Closed Substratum Domain) [A.1.3, p.132]
--

Now observe, depending on which base condition holds (see cases 2 and 4 in Table 2.5), every block I of partition \mathcal{S} , associated with any given configuration $\mathbf{x} \in \mathcal{D}_A(I) \cup (\mathfrak{S}(\tau) \setminus \mathcal{D}_H(I))$, can be related to a binary scalar $\hat{b}_I(\mathbf{x}) \in \{-1, +1\}$ such that ²

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

where $\mathcal{D}_A(I)$ and $\mathcal{D}_H(I)$ are defined as in (2.15) and (2.18), 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

²Observe from Table 2.5 that any configuration $\mathbf{x} \in \mathfrak{S}(\tau)$ satisfies a base condition (see cases 2 or 4 in Table 2.5) at cluster $I \in \mathcal{C}(\tau)$ if $\mathbf{x} \in \mathcal{D}_A(I) \cup (\mathfrak{S}(\tau) \setminus \mathcal{D}_H(I))$. Also note that $\mathcal{D}_A(I) \cup (\mathfrak{S}(\tau) \setminus \mathcal{D}_H(I)) = \mathcal{D}_A(I) \cup (\mathfrak{S}(\tau) \setminus (\mathcal{D}_A(I) \cup \mathcal{D}_H(I)))$, and $\mathcal{D}_A(I)$ and $\mathfrak{S}(\tau) \setminus (\mathcal{D}_A(I) \cup \mathcal{D}_H(I))$ are disjoint.

controllers indexed by partitions of S compatible with τ and associated binary vectors as follows.

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

$$\mathcal{SP}_S(\tau) := \left\{ (\mathcal{S}, \mathbf{b}) \mid \mathcal{S} \in \mathcal{P}_S(\tau), \mathbf{b} \in \{-1, +1\}^{\mathcal{S}} \right\}. \quad (\text{A.2})$$

For any partition $\mathcal{S} \in \mathcal{P}_S(\tau)$ of S and $\mathbf{b} := (b_I)_{I \in \mathcal{S}} \in \{-1, +1\}^{\mathcal{S}}$, the domain $\mathcal{D}(\mathcal{S}, \mathbf{b})$ of a local control policy $h_{\mathcal{S}, \mathbf{b}}$, presented in Table A.2, is defined to be

$$\mathcal{D}(\mathcal{S}, \mathbf{b}) := \bigcap_{I \in \mathcal{S}} \left(\mathcal{D}_B(I, b_I) \cap \bigcap_{K \in \text{Anc}(I, \tau)} \mathcal{D}_H(K) \right), \quad (\text{A.3})$$

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

$$\mathcal{D}_B(I, b_I) := \begin{cases} \mathcal{D}_A(I), & \text{if } b_I = +1, \\ \mathfrak{S}(\tau), & \text{if } b_I = -1, \end{cases} \quad (\text{A.4})$$

and all ancestors $K \in \text{Anc}(I, \tau)$ of I in τ satisfy the recursion condition of having properly separated children clusters described by $\mathcal{D}_H(K)$ (2.18). Accordingly, let $\mathcal{V}_\tau(\mathcal{S})$ denote the set of clusters of τ visited during the recursive computation of $h_{\mathcal{S}, \mathbf{b}}$ in Table A.2,

$$\mathcal{V}_\tau(\mathcal{S}) := \{K \in \mathcal{C}(\tau) \mid K \supseteq I, I \in \mathcal{S}\}. \quad (\text{A.5})$$

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

Observe that each local control policy $h_{\mathcal{S}, \mathbf{b}}$ is a recursive composition of continuous functions of \mathbf{x} ; therefore, it is continuous.

Proposition A.2 *The recursion in Table A.2 defines a continuous and piecewise smooth function,³ $h_{\mathcal{S}, \mathbf{b}} : \mathfrak{S}(\tau) \rightarrow (\mathbb{R}^n)^S$.*

Proof See Appendix A.2.3. ■

To conclude our introduction of the family of covering fields in Table A.2, we now observe that the vector field $f_{\tau, \mathbf{y}}$ in Table 2.5 is an online concatenation of continuous local controllers, $h_{\mathcal{S}, \mathbf{b}}$, of Table A.2 using a policy selection method described in Table A.3, summarized as:

Proposition A.3 *For any given configuration $\mathbf{x} \in \mathfrak{S}(\tau)$, the policy selection algorithm p in Table A.3 always returns a valid policy index $(\mathcal{S}, \mathbf{b}) = p(\mathbf{x})$ in $\mathcal{SP}_S(\tau)$ (A.2) such that the domain $\mathcal{D}(\mathcal{S}, \mathbf{b})$ (A.3) of the associated local control policy $h_{\mathcal{S}, \mathbf{b}}$ (Table A.2) contains \mathbf{x} , i.e.,*

$$\mathbf{x} \in (\mathcal{D} \circ p)(\mathbf{x}). \quad (\text{A.6})$$

Proof See Appendix A.2.4. ■

³Note that if $f : U \rightarrow \mathbb{R}^n$ is continuous and piecewise smooth on an open set $U \subset \mathbb{R}^n$, then it is locally Lipschitz on U [135].

Table A.2: Local Control Policies in a Hierarchical Stratum

Let \mathcal{S} be a partition of S with $\mathcal{S} \subset \mathcal{C}(\tau)$, and $\mathbf{b} = (b_I)_{I \in \mathcal{S}} \in \{-1, +1\}^{\mathcal{S}}$. For any desired $\mathbf{y} \in \mathfrak{S}(\tau)$, supporting $\tau \in \mathcal{BT}_S$, and initial $\mathbf{x} \in \mathcal{D}(\mathcal{S}, \mathbf{b})$ (A.3), the local control policy, $h_{\mathcal{S}, \mathbf{b}} : \mathcal{D}(\mathcal{S}, \mathbf{b}) \rightarrow (\mathbb{R}^n)^S$,

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

is recursively computed starting at the root cluster S with the zero control input $\mathbf{0} \in (\mathbb{R}^n)^S$ as follows: for any $\mathbf{u} \in (\mathbb{R}^n)^S$ and $I \in \mathcal{V}_\tau(\mathcal{S})$ (A.5),

Base Cases	{	<ol style="list-style-type: none"> 1. function $\hat{\mathbf{u}} = \hat{h}_{\mathcal{S}, \mathbf{b}}(\mathbf{x}, \mathbf{u}, I)$ 2. if $I \in \mathcal{S}$, 3. if $b_I = +1$ 4. $\hat{\mathbf{u}} \leftarrow f_A(\mathbf{x}, \mathbf{u}, I)$ (2.14), % Attracting Field 5. else 6. $\hat{\mathbf{u}} \leftarrow f_S(\mathbf{x}, \mathbf{u}, I)$ (2.24), % Split Separation Field 7. end 8. else 	
Recursion	{	<ol style="list-style-type: none"> 9. $\{I_L, I_R\} \leftarrow \text{Ch}(I, \tau)$, 10. $\hat{\mathbf{u}}_L \leftarrow \hat{h}_{\mathcal{S}, \mathbf{b}}(\mathbf{x}, \mathbf{u}, I_L)$, % Recursion for Left Child 11. $\hat{\mathbf{u}}_R \leftarrow \hat{h}_{\mathcal{S}, \mathbf{b}}(\mathbf{x}, \hat{\mathbf{u}}_L, I_R)$, % Recursion for Right Child 12. $\hat{\mathbf{u}} \leftarrow f_H(\mathbf{x}, \hat{\mathbf{u}}_R, I)$ (2.19), % Split Preserving Field 13. end 14. return $\hat{\mathbf{u}}$ 	

Proposition A.4 *For any given $\mathbf{x} \in \mathfrak{S}(\tau)$, the vector field $f_{\tau, \mathbf{x}}$ (see Table 2.5) and the local control policy $h_{\mathcal{S}, \mathbf{b}}(\mathbf{x})$ (see Table A.2) selected as $(\mathcal{S}, \mathbf{b}) = p(\mathbf{x})$ (see Table A.3) generate the same control (velocity) inputs, i.e.,*

$$f_{\tau, \mathbf{y}}(\mathbf{x}) = h_{p(\mathbf{x})}(\mathbf{x}) . \quad (\text{A.7})$$

Proof See Appendix A.2.5. ■

Since the vector field $f_{\tau, \mathbf{y}}$ is defined for entire $\mathfrak{S}(\tau)$, it is useful to remark that the domains $\mathcal{D}(\mathcal{S}, \mathbf{b})$ (A.3) of substratum policies $h_{\mathcal{S}, \mathbf{b}}$ define a cover of $\mathfrak{S}(\tau)$ indexed by partitions of S compatible with τ and associated binary vectors.

A.1.2 Online Sequential Composition of Substratum Policies

We now briefly describe the logic behind online sequential composition [45] of substratum policies.

Table A.3: Substratum Policy Selection Algorithm

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

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

recursively generates a local policy index in $\mathcal{SP}_S(\tau)$ (A.2) starting at the root cluster S as follows: for any $I \in \mathcal{C}(\tau)$,

- | | | |
|------------|---|--|
| Base Cases | { | 1. function $(\hat{\mathcal{J}}, \hat{\mathbf{b}}) = \hat{p}(\mathbf{x}, I)$ |
| | | 2. if $\mathbf{x} \in \mathcal{D}_A(I)$ (2.15), |
| | | 3. $\hat{\mathcal{J}} \leftarrow \{I\}$, |
| | | 4. $\hat{\mathbf{b}} \leftarrow +1$, |
| | | 5. else if $\mathbf{x} \notin \mathcal{D}_H(I)$ (2.18), |
| | | 6. $\hat{\mathcal{J}} \leftarrow \{I\}$, |
| | | 7. $\hat{\mathbf{b}} \leftarrow -1$, |
| Recursion | { | 8. else |
| | | 9. $\{I_L, I_R\} \leftarrow \text{Ch}(I, \tau)$, |
| | | 10. $(\hat{\mathcal{J}}_L, \hat{\mathbf{b}}_L) \leftarrow \hat{p}(\mathbf{x}, I_L)$, |
| | | 11. $(\hat{\mathcal{J}}_R, \hat{\mathbf{b}}_R) \leftarrow \hat{p}(\mathbf{x}, I_R)$, |
| | | 12. $\hat{\mathcal{J}} \leftarrow \hat{\mathcal{J}}_L \cup \hat{\mathcal{J}}_R$, |
| | | 13. $\hat{\mathbf{b}} \leftarrow \hat{\mathbf{b}}_L \parallel \hat{\mathbf{b}}_R$, ⁴ |
| | | 14. end |
| | | 15. return $(\hat{\mathcal{J}}, \hat{\mathbf{b}})$ |

To characterize our policy selection strategy, we first define a priority measure⁵ for each local controller $h_{\mathcal{S}, \mathbf{b}}$ associated with a partition $\mathcal{S} \in \mathcal{P}_S(\tau)$ of S and a binary vector $\mathbf{b} \in \{-1, +1\}^{\mathcal{S}}$ to be

$$\text{priority}(\mathcal{S}, \mathbf{b}) := \sum_{I \in \mathcal{S}} b_I |I|^2. \quad (\text{A.8})$$

Note that the maximum and minimum of the priority measure is attained at the coarsest partition $\{S\}$ of S , and $b_S = +1$ and $b_S = -1$, respectively,

$$\text{priority}(\{S\}, +1) = |S|^2, \quad (\text{A.9a})$$

$$\text{priority}(\{S\}, -1) = -|S|^2. \quad (\text{A.9b})$$

⁴Here, $\mathbf{p} \parallel \mathbf{q}$ 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 $\mathbf{r} = \mathbf{p} \parallel \mathbf{q}$, if and only if $r_a = p_a$ and $r_b = q_b$ for all $a \in A$ and $b \in B$.

⁵In past literature, such a priority assignment of local controllers is done by using backchaining of the prepares graph in an offline manner [45].

Accordingly, we shall refer to the local control policy with index $(\{S\}, +1)$ as the goal policy, because it has the highest priority and asymptotically steers all configurations in its domain $\mathcal{D}(\{S\}, +1)$ (A.3) to \mathbf{y} following the negated gradient of $V(\mathbf{x}) = \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|$, i.e., for any $\mathbf{x} \in \mathcal{D}(\{S\}, +1)$

$$h_{\{S\}, +1}(\mathbf{x}) = -\nabla V(\mathbf{x}) = -(\mathbf{x} - \mathbf{y}) . \quad (\text{A.10})$$

Note that since the root cluster S has no ancestor, i.e., $\text{Anc}(S, \tau) = \emptyset$, by definition (A.3), $\mathcal{D}(\{S\}, +1) = \mathcal{D}_A(S)$, and $\mathcal{D}_A(S)$ (2.15) contains the goal configuration \mathbf{y} .

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

Definition A.1 *Let $(\mathcal{S}, \mathbf{b}), (\mathcal{S}', \mathbf{b}') \in \mathcal{SP}_S(\tau)$ be two distinct substratum policy indices. Then, $h_{\mathcal{S}, \mathbf{b}}$ is said to prepare $h_{\mathcal{S}', \mathbf{b}'}$ if and only if all trajectories of $h_{\mathcal{S}, \mathbf{b}}$ starting in its domain $\mathcal{D}(\mathcal{S}, \mathbf{b})$, possibly excluding a set of measure zero, reach $\mathcal{D}(\mathcal{S}', \mathbf{b}')$ in finite time.*⁶

Accordingly, define the prepares graph $\mathcal{PG} = (\mathcal{SP}_S(\tau), \mathcal{E}_{\mathcal{PG}})$ to have vertex set $\mathcal{SP}_S(\tau)$ (A.2) with a policy index $(\mathcal{S}, \mathbf{b}) \in \mathcal{SP}_S(\tau)$ connected to another policy index $(\mathcal{S}', \mathbf{b}')$ by a directed edge in $\mathcal{E}_{\mathcal{PG}}$ if and only if $h_{\mathcal{S}, \mathbf{b}}$ prepares $h_{\mathcal{S}', \mathbf{b}'}$.

Although the prepares graph \mathcal{PG} is the most critical component of the sequential composition framework [45] that defines a discrete abstraction of continuous control policies, the exponentially growing cardinality of substratum policies, discussed in Appendix A.1.5, 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 $(\{S\}, +1)$ through a directed path, as follows.

Definition A.2 *Let $\widehat{\mathcal{PG}} = (\mathcal{SP}_S(\tau), \widehat{\mathcal{E}}_{\mathcal{PG}})$ be a graph with vertex list $\mathcal{SP}_S(\tau)$, and a policy index $(\mathcal{S}, \mathbf{b}) \in \mathcal{SP}_S(\tau)$ that is connected to another policy index $(\mathcal{S}', \mathbf{b}') \in \mathcal{SP}_S(\tau)$ by a directed edge in $\widehat{\mathcal{E}}_{\mathcal{PG}}$ if and only if at least one of the following properties holds:*⁷

- (i) *Complement:* There exists a singleton cluster $I \in \mathcal{S}$ such that $\mathbf{b}_I = -1$, and $\mathcal{S}' = \mathcal{S}$ and $\mathbf{b}' \in \{-1, +1\}^{\mathcal{S}'}$ with $\mathbf{b}'_I = +1$ and $\mathbf{b}'_D = \mathbf{b}_D$ for all $D \in \mathcal{S} \setminus \{I\}$.
- (ii) *Split:* There exists a nonsingleton cluster $I \in \mathcal{S}$ such that $\mathbf{b}_I = -1$, and $\mathcal{S}' = \mathcal{S} \setminus \{I\} \cup \text{Ch}(I, \tau)$ and $\mathbf{b}' \in \{-1, +1\}^{\mathcal{S}'}$ with $\mathbf{b}'_K = -1$ for all $K \in \text{Ch}(I, \tau)$ and $\mathbf{b}'_D = \mathbf{b}_D$ for all $D \in \mathcal{S} \setminus \text{Ch}(I, \tau)$.
- (iii) *Merge:* There exists a nonsingleton cluster $I \in \mathcal{C}(\tau)$ such that $\text{Ch}(I, \tau) \subset \mathcal{S}$ and $\mathbf{b}_K = +1$ for all $K \in \text{Ch}(I, \tau)$, and $\mathcal{S}' = \mathcal{S} \setminus \text{Ch}(I, \tau) \cup \{I\}$ and $\mathbf{b}' \in \{-1, +1\}^{\mathcal{S}'}$ with $\mathbf{b}'_I = +1$ and $\mathbf{b}'_D = \mathbf{b}_D$ for all $D \in \mathcal{S} \setminus \text{Ch}(I, \tau)$.

Note that, since \mathcal{S} is compatible with τ , i.e., $\mathcal{S} \subset \mathcal{C}(\tau)$, if $|\mathcal{S}| > 1$, then there exists a cluster $I \in \mathcal{C}(\tau)$ such that $\text{Ch}(I, \tau) \subset \mathcal{S}$ (Lemma A.4). Hence, for any policy index

⁶ Here, we slightly relax the original definition of the prepares relation in [45] by not requiring the knowledge of goal sets, globally asymptotically stable states of local control policies in advance.

⁷One 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 2.1.4.

$(\mathcal{S}, \mathbf{b}) \neq (\{S\}, +1)$, there always exists a policy index $(\mathcal{S}', \mathbf{b}') \neq (\mathcal{S}, \mathbf{b})$ that satisfies one of these conditions, (i)-(iii) above. Thus, the out-degree of a policy index $(\mathcal{S}, \mathbf{b}) \neq (\{S\}, +1)$ in $\widehat{\mathcal{P}\mathcal{G}}$ is at least one, whereas the goal policy index $(\{S\}, +1)$ in $\widehat{\mathcal{P}\mathcal{G}}$ has an out-degree of zero. We summarize some important properties of $\widehat{\mathcal{P}\mathcal{G}}$ as follows:

Proposition A.5 *The graph $\widehat{\mathcal{P}\mathcal{G}} = (\mathcal{SP}_S(\tau), \widehat{\mathcal{E}}_{\mathcal{P}\mathcal{G}})$, as defined in Definition A.2, is an acyclic subgraph of the prepares graph $\mathcal{P}\mathcal{G} = (\mathcal{SP}_S(\tau), \mathcal{E}_{\mathcal{P}\mathcal{G}})$ (see Definition A.1) such that all policy indices in $\mathcal{SP}_S(\tau)$ are connected to the goal policy index $(\{S\}, +1)$ through directed paths in $\widehat{\mathcal{E}}_{\mathcal{P}\mathcal{G}}$, of length at most $O(|S|^2)$ hops, along which **priority** (A.8) is strictly increasing, i.e., for any $((\mathcal{S}, \mathbf{b}), (\mathcal{S}', \mathbf{b}')) \in \widehat{\mathcal{E}}_{\mathcal{P}\mathcal{G}}$*

$$\text{priority}(\mathcal{S}', \mathbf{b}') > \text{priority}(\mathcal{S}, \mathbf{b}). \quad (\text{A.11})$$

Proof See Appendix A.2.6. ■

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 A.6 *For any given $\mathbf{x} \in \mathfrak{S}(\tau)$, the policy selection method, p , in Table A.3 always returns the index of a local controller with the maximum priority among all local controllers whose domain contains \mathbf{x} , i.e.,*

$$p(\mathbf{x}) = \arg \max_{\substack{(\mathcal{S}', \mathbf{b}') \in \mathcal{SP}_S(\tau) \\ \mathbf{x} \in \mathcal{D}(\mathcal{S}', \mathbf{b}')}} \text{priority}(\mathcal{S}', \mathbf{b}') \quad (\text{A.12})$$

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

Proof See Appendix A.2.7. ■

A.1.3 Qualitative Properties of Substratum Policies

We now list important qualitative (existence, uniqueness, invariance, and stability) properties of the substratum control policies of Table A.2. Let \mathcal{S} be a partition of S compatible with τ , i.e., $\mathcal{S} \subset \mathcal{C}(\tau)$, and \mathbf{b} is a binary vector in $\{-1, 1\}^{\mathcal{S}}$.

Proposition A.7 *The domain $\mathcal{D}(\mathcal{S}, \mathbf{b})$ (A.3) of a substratum policy $h_{\mathcal{S}, \mathbf{b}}$ (see Table A.2) is positive invariant.*

Proof See Appendix A.2.9. ■

Proposition A.8 *(Substratum Existence and Uniqueness) The vector field $h_{\mathcal{S}, \mathbf{b}}$ (see Table A.2) is locally Lipschitz in $\mathfrak{S}(\tau)$, and for any initial $\mathbf{x} \in \mathcal{D}(\mathcal{S}, \mathbf{b}) \subset \mathfrak{S}(\tau)$, there always exists a compact (bounded and closed) subset W of $\mathcal{D}(\mathcal{S}, \mathbf{b})$ (A.3) such that all trajectories of $h_{\mathcal{S}, \mathbf{b}}$ starting at \mathbf{x} remain in W for all future time.*

Therefore, there is a unique continuous and piecewise smooth flow of $h_{\mathcal{S}, \mathbf{b}}$ in $\mathcal{D}(\mathcal{S}, \mathbf{b})$ that is defined for all future time.

Proof See Appendix A.2.8. ■

Proposition A.9 (*Finite Time Prepares Relation*) Each local control policy $h_{\mathcal{S}, \mathbf{b}}$, with the exception of the goal controller $h_{\{S\}, +1}$, steers (almost) all configurations in its domain $\mathcal{D}(\mathcal{S}, \mathbf{b})$ to the domain $\mathcal{D}(\mathcal{S}', \mathbf{b}')$ of another local controller $h_{\mathcal{S}', \mathbf{b}'}$ at a higher priority (A.8) in finite time.

Proof See Appendix A.2.10. ■

Proposition A.10 (*Nonzero Execution Time*) Let \mathbf{x}^t be a trajectory of the local control policy $h_{\mathcal{S}, \mathbf{b}}$ starting at $\mathbf{x}^0 \in \mathcal{D}(\mathcal{S}, \mathbf{b})$ such that $p(\mathbf{x}^0) = (\mathcal{S}, \mathbf{b})$.

Then, the local controller is guaranteed to steer the group for a nonzero time until reaching the domain of a local controller at a higher priority (A.8), i.e.,

$$\inf_t \{t \geq 0 \mid p(\mathbf{x}^t) \neq (\mathcal{S}, \mathbf{b})\} > 0. \quad (\text{A.13})$$

Proof Recall that for any configuration $\mathbf{x} \in \mathfrak{S}(\tau)$, the policy selection method in Table A.3 always yields the index of the local controller with the highest priority among all local controllers whose domains contain \mathbf{x} (Proposition A.6). Hence, since the initial configuration \mathbf{x}^0 is not included in the domain of any other local controller with a higher priority than $\text{priority}(\mathcal{S}, \mathbf{b})$ and domains of local controllers are closed relative to $\mathfrak{S}(\tau)$ (Lemma A.1), there always exists an open set around \mathbf{x}^0 which does not intersect with the domain of any local controller at a higher priority than $\text{priority}(\mathcal{S}, \mathbf{b})$. Thus, since its domain is positively invariant (Proposition A.7), $h_{\mathcal{S}, \mathbf{b}}$ is guaranteed to steer the configuration in the intersection of this open set and $\mathcal{D}(\mathcal{S}, \mathbf{b})$ for a nonzero time. ■

A.1.4 Qualitative Properties of Stratum Policies

We now proceed with a list of important qualitative properties of the hierarchy-preserving navigation policy of Table 2.5.

Proposition A.11 *The stratum $\mathfrak{S}(\tau)$ is positive invariant under the hierarchy-invariant control policy $f_{\tau, \mathbf{y}}$ (see Table 2.5).*

Proof Recall that the domains \mathcal{D} (A.3) of local control policies in Table A.2 define a cover of $\mathfrak{S}(\tau)$ (see Proposition A.3), each of whose elements is positively invariant under the flow of the associated local policy (see Proposition A.7). Thus, the result follows because the hierarchy preserving vector field $f_{\tau, \mathbf{y}}$ is equivalent to online sequential composition of local control policies of Table A.2 based on the policy selection algorithm in Table A.3 (see Proposition A.4). ■

Proposition A.12 (*Stratum Existence and Uniqueness*) *The hierarchy invariance control policy $f_{\tau, \mathbf{y}}$ (Table 2.5) has a unique, continuous, and piecewise smooth flow, φ^t , in $\mathfrak{S}(\tau)$, defined for all $t \geq 0$.*

Proof Recall from Proposition A.4 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 (see Proposition A.8). Since their domains define a finite closed cover of $\mathfrak{S}(\tau)$ (see Proposition A.3), the unique, continuous, and piecewise flow of $f_{\tau, \mathbf{y}}$ is constructed by piecing together trajectories of these substratum policies. ■

Proposition A.13 *Any $\mathbf{y} \in \mathfrak{S}(\tau)$ is an asymptotically stable equilibrium point of the hierarchy-invariant control policy, $f_{\tau, \mathbf{y}}$ (see Table 2.5), whose basin of attraction includes $\mathfrak{S}(\tau)$, except a set of measure zero.*

Proof Using the equivalence (Proposition A.4) of the hierarchy preserving field $f_{\tau, \mathbf{y}}$ and the sequential composition of substratum control policies of Table A.2 based on the policy selection method in Table A.3, the result can be obtained as follows.

Since **priority** (A.8) is an integer-valued function with bounded range $[-|S|^2, |S|^2]$, using Proposition A.6 and Proposition A.9, one can conclude that the disks starting at almost any configuration in $\mathfrak{S}(\tau)$ reach the domain $\mathcal{D}(\{S\}, +1)$ of the goal policy $h_{\{S\}, +1}$ in finite time after visiting at most $O(|S|^2)$ of other local control policies. Note that $\mathbf{y} \in \mathcal{D}(\{S\}, +1)$. Then, the goal policy $h_{\{S\}, +1}$,

$$h_{\{S\}, +1}(\mathbf{x}) = -\nabla_{\frac{1}{2}} \|\mathbf{x} - \mathbf{y}\|_2^2 = -(\mathbf{x} - \mathbf{y}), \quad (\text{A.14})$$

asymptotically steers all configuration in $\mathcal{D}(\{S\}, +1)$ to \mathbf{y} while keeping its domain of attraction $\mathcal{D}_A(S)$ positively invariant (see Proposition A.7), which completes the proof ■

A.1.5 On the Cardinality of Substratum Policies

To gain an appreciation for the computational efficiency of hierarchy preserving vector field in Table 2.5, we find it useful to have a brief discussion without proofs on the cardinality of the family of local control policies of Table A.2. The number of partitions $\mathcal{P}_S(\tau)$ of S ⁸ compatible with a cluster hierarchy $\tau \in \mathcal{BT}_S$ is recursively given by⁹

$$|\mathcal{P}_S(\tau)| = 1 + |\mathcal{P}_S(\tau_L)| |\mathcal{P}_S(\tau_R)|, \quad (\text{A.16})$$

where τ_L, τ_R denote the left and right subtrees of τ , respectively. For any caterpillar tree¹⁰ $\sigma \in \mathcal{BT}_S$, we have $|\mathcal{P}_S(\sigma)| = |S|$, because one of two subtrees of σ is always one-leaf tree. On the other hand, for a balanced tree $\gamma \in \mathcal{BT}_S$, the cardinality of partitions of S compatible

⁸The number of partitions of a set with n elements is given by the Bell number, B_n , recursively defined as: for any $n \in \mathbb{N}$ [189]

$$B_{n+1} = \sum_{k=0}^n \binom{n}{k} B_k, \quad (\text{A.15})$$

where $B_0 = 1$. The Bell number, B_n , grows super exponentially with the set size, n ; however, in our case we require partitions of S to be compatible with τ and this restricts the growth of number of such partitions of S to at most exponential with $|S|$, depending on the structure of τ .

⁹Let $\{S_L, S_R\} = \text{Ch}(S, \tau)$ be the root split of τ , and τ_L and τ_R are the associated subtrees of τ rooted at S_L and S_R , respectively. Then, any partition of S compatible with τ , except the trivial partition $\{S\}$, can be written as the union of a partition of S_L compatible with τ_L and a partition of S_R compatible with τ_R . Hence, one can conclude the recursion in (A.16).

¹⁰A caterpillar tree is a rooted tree in which at most one of the children of every interior cluster is nonsingleton.

with γ grows exponentially,¹¹

$$\sqrt{2}^{|S|} \leq |\mathcal{P}_S(\gamma)| \leq \frac{4}{5} \sqrt{\frac{5}{2}}^{|S|}, \quad (\text{A.21})$$

for $|J| = 2^k$, $k \in \mathbb{N}_+ = \{1, 2, 3, \dots\}$; for example, $|\mathcal{P}_{[2]}(\gamma)| = 2$, $|\mathcal{P}_{[4]}(\gamma)| = 5$, $|\mathcal{P}_{[8]}(\gamma)| = 26$ and $|\mathcal{P}_{[16]}(\gamma)| = 677$. In addition to a partition \mathcal{S} of S compatible with τ , every local control policy $h_{\mathcal{S}, \mathbf{b}}$ is indexed by a binary variable of size $|S|$ with a possible choice of $2^{|S|}$ values. Thus, the number of local control policies $h_{\mathcal{S}, \mathbf{b}}$ grows exponentially with the group size, $|S|$.

A.1.6 A Set of Useful Observations on Substratum Policies

Here we introduce a set of useful lemmas that constitute building blocks for proving some qualitative properties of substratum policies presented in Appendix A.1.3. Let \mathcal{S} be a partition of S compatible with τ , i.e. $\mathcal{S} \subset \mathcal{C}(\tau)$, and \mathbf{b} is a binary vector in $\{-1, +1\}^{\mathcal{S}}$.

Lemma A.1 *The domain, $\mathcal{D}(\mathcal{S}, \mathbf{b})$ (A.3), of each substratum policy, $h_{\mathcal{S}, \mathbf{b}}$, is closed relative to $\mathfrak{S}(\tau)$.*

Proof Using the continuity of functions¹² in the predicates used to define them, one can conclude that for any $I \in \mathcal{C}(\tau)$, sets $\mathcal{D}_A(I)$ (2.15) and $\mathcal{D}_H(I)$ (2.18) are closed relative to $\mathfrak{S}(\tau)$. Hence, since the intersection of arbitrary many closed sets are closed [161], the domain $\mathcal{D}(\mathcal{S}, \mathbf{b})$ (A.3) of each local controller $h_{\mathcal{S}, \mathbf{b}}$ is closed relative to $\mathfrak{S}(\tau)$. ■

A critical observation used for bounding the centroidal configuration radius (Lemma A.3) and the range of a trajectory of a substratum policy (Proposition A.8) is:

¹¹Let F_n denote the number of partitions of $[n] = \{1, 2, \dots, n\}$ compatible with a balanced rooted binary tree with n leaves, where $n = 2^k$ for some $k \in \mathbb{N}_+$, and by (A.16) it satisfies

$$F_{2n} = 1 + F_n^2, \quad (\text{A.17})$$

subject to the base condition $F_2 = 2$. Define G_n and H_n , for $n = 2^k$ and $k \in \mathbb{N}_+$, to be, respectively,

$$G_{2n} = G_n^2 \quad \text{and} \quad H_{2n} = \frac{5}{4} H_n^2 \quad (\text{A.18})$$

where $G_2 = H_2 = 2$. Note that $G_n = \sqrt{2}^n$ and $H_n = \frac{4}{5} \sqrt{\frac{5}{2}}^n$ for $n = 2^k$ and $k \in \mathbb{N}_+$. Now observe that for any $n = 2^k$ and $k \in \mathbb{N}_+$

$$G_n \leq F_n \leq H_n, \quad (\text{A.19})$$

and so

$$\sqrt{2}^n \leq F_n \leq \frac{4}{5} \sqrt{\frac{5}{2}}^n. \quad (\text{A.20})$$

¹²A function $f : X \rightarrow Y$ between two topological spaces, X and Y , is continuous if the inverse image of every open subset of Y of f is an open subset of X [161].

Lemma A.2 (*Relative Centroidal Dynamics*) Let $\mathbf{x} \in \mathfrak{S}(\tau)$ and $\mathbf{u} = h_{\mathfrak{S}, \mathbf{b}}(\mathbf{x})$. Then, the centroidal dynamics of any cluster $I \in \mathcal{V}_\tau(\mathfrak{S})$ (A.5) visited during recursive computation of $h_{\mathfrak{S}, \mathbf{b}}$ (Table A.2), except the root S , satisfies¹³

$$c(\mathbf{u}|I) = -c(\mathbf{x} - \mathbf{y}|I) + 2\alpha_P(\mathbf{x}, \mathbf{v}_P) \frac{|I^{-\tau}|}{|P|} \frac{s_I(\mathbf{x})}{\|s_I(\mathbf{x})\|} + c(\mathbf{u}|P) + c(\mathbf{x} - \mathbf{y}|P) , \quad (\text{A.22})$$

for some $\mathbf{v}_P \in (\mathbb{R}^d)^S$ associated with parent cluster $P = \text{Pr}(I, \tau)$; whereas we have for the root cluster S

$$c(\mathbf{u}|S) = -c(\mathbf{x} - \mathbf{y}|S) . \quad (\text{A.23})$$

Proof See Appendix A.2.11. ■

Lemma A.3 (*Upper Bound on Configuration Radius*) Let \mathbf{x}^t denote a trajectory of $h_{\mathfrak{S}, \mathbf{b}}$ (Table A.2) starting at any initial $\mathbf{x}^0 \in \mathcal{D}(\mathfrak{S}, \mathbf{b})$ (A.3) for $t \geq 0$.

Then, the centroidal configuration radius, $r(\mathbf{x}^t|S)$ (2.12), is bounded above for all $t \geq 0$ by a certain finite value, $R(\mathbf{x}^0, \mathbf{y})$, depending on \mathbf{x}^0 and \mathbf{y} , i.e.,

$$r(\mathbf{x}^t|S) \leq R(\mathbf{x}^0, \mathbf{y}) < \infty, \quad \forall t \geq 0. \quad (\text{A.24})$$

Proof See Appendix A.2.12. ■

Lemma A.4 If \mathfrak{S} is not the trivial partition, i.e., $|\mathfrak{S}| > 1$, then there always exists a cluster $I \in \mathcal{C}(\tau)$ such that $\text{Ch}(I, \tau) \subset \mathfrak{S}$.

Proof Define the depth of cluster $I \in \mathcal{C}(\tau)$ in τ to be the number of its ancestors, $|\text{Anc}(I, \tau)|$. Let $K \in \mathfrak{S}$ be a cluster in \mathfrak{S} with the maximal depth, i.e.,

$$|\text{Anc}(K, \tau)| = \arg \max_{D \in \mathfrak{S}} |\text{Anc}(D, \tau)| . \quad (\text{A.25})$$

Then, we will now show that $K^{-\tau}$ is also in \mathfrak{S} , and so $I = \text{Pr}(K, \tau)$ satisfies the lemma.

Proof by a contradiction. Suppose that $K^{-\tau}$ is not in \mathfrak{S} . Since \mathfrak{S} is a partition of S compatible with τ , then some descendant $D \in \text{Des}(K^{-\tau}, \tau)$ is in \mathfrak{S} . Note that $|\text{Anc}(K, \tau)| = |\text{Anc}(K^{-\tau})| < |\text{Anc}(D, \tau)|$, which contradicts (A.25). ■

Lemma A.5 Let \mathfrak{S} and \mathfrak{S}' be two distinct partitions of S compatible with τ , i.e., $\mathfrak{S} \neq \mathfrak{S}' \subset \mathcal{C}(\tau)$. Then, at least one of the following cases always holds

- (i) (\mathfrak{S} Partially Refines \mathfrak{S}') There exists a cluster $K' \in \mathfrak{S}'$ with a nontrivial partition \mathcal{K}' such that $\mathcal{K}' \subset \mathfrak{S}$.
- (ii) (\mathfrak{S}' Partially Refines \mathfrak{S}) There exists a cluster $K \in \mathfrak{S}$ with a nontrivial partition \mathcal{K} such that $\mathcal{K} \subset \mathfrak{S}'$.

Proof For any $j \in S$, let $\mathfrak{S}(j)$ denote the unique element of \mathfrak{S} containing j .

Since $\mathfrak{S} \neq \mathfrak{S}'$, let $K' \in \mathfrak{S}' \setminus \mathfrak{S}$ be an unshared cluster. Since both \mathfrak{S} and \mathfrak{S}' are partitions of S compatible with τ , we have either $\mathfrak{S}(k') \subsetneq K'$ or $\mathfrak{S}(k') \supsetneq K'$ for all $k' \in K'$. If $\mathfrak{S}(k') \subsetneq K'$ for all $k' \in K'$, then $\mathcal{K}' = \{\mathfrak{S}(k')|k' \in K'\}$ defines a partition of K' and we obtain Lemma A.5.(i). Otherwise, by symmetry, we have Lemma A.5.(ii). Thus, the lemma follows. ■

¹³Here, for any $I \in \mathcal{C}(\tau)$ we use $c : (\mathbb{R}^d)^I \rightarrow \mathbb{R}^d$ (2.8).

A.2 Proofs

A.2.1 Proof of Theorem 2.6

Proof To prove the first part of the result, we shall consider $\text{Port}_{\sigma,\tau}$ as a mapping from $\mathfrak{S}(\sigma)$ to $(\mathbb{R}^n)^S$ and verify that $\text{Port}_{\sigma,\tau}(\mathfrak{S}(\sigma)) \subseteq \text{Portal}(\sigma, \tau)$.

By definition, the restriction of $\text{Port}_{\sigma,\tau}$ to $\text{Portal}(\sigma, \tau)$ is the identity map on $\text{Portal}(\sigma, \tau)$. Hence, we only need to show that $\text{Port}_{\sigma,\tau}(\mathfrak{S}(\sigma) \setminus \text{Portal}(\sigma, \tau)) \subseteq \text{Portal}(\sigma, \tau)$.

Let $\mathbf{y} = \text{Ctr}(\mathbf{x})$ and $\mathbf{z} = \text{Sc1}(\mathbf{y})$ be intermediate configurations for the portal transformation of a configuration $\mathbf{x} \in \mathfrak{S}(\sigma) \setminus \text{Portal}(\sigma, \tau)$ into $\mathbf{w} = \text{Mrg}(\mathbf{z}) = \text{Port}(\mathbf{x})$.

First, recall that rigid transformations and scaling of partial configurations preserve their clustering structure [15]. Hence, the common subtrees of σ and τ rooted at A , B and C are preserved after each transformation by Ctr (2.34), Sc1 (2.35) and Mrg (2.37).

Second, each partial configuration of the symmetric configuration $\mathbf{y} \in \text{Sym}(\sigma, \tau)$ associated with (σ, τ) is properly translated by Sc1 (2.35) so that each of them lies in the corresponding consensus ball, i.e., $r(\mathbf{z}|Q) < r_Q(\mathbf{z})$ for all $Q \in (A, B, C)$. Hence, the partial configuration $\mathbf{z}|P$ supports both of the subtrees of σ and τ rooted at P .

Finally, if $P = S$, then the result simply follows, because $\mathbf{z} = \mathbf{w} \in \text{Portal}(\sigma, \tau)$. Otherwise, for every $I \in \{P\} \cup \text{Anc}(P, \sigma)$, Mrg (2.37) iteratively separates the common complementary clusters I and $I^{-\sigma}$ of σ and τ , in a bottom up fashion starting at cluster P , to support the subtrees of σ and τ rooted at $\text{Pr}(I, \sigma)$. Note that in the base case, \mathbf{z} supports both of the subtrees of σ and τ rooted at P and $P^{-\sigma}$; and at the termination at root cluster S , \mathbf{w} supports both trees σ and τ , i.e., $\mathbf{w} \in \text{Portal}(\sigma, \tau)$.

We now proceed with the computational properties of $\text{Port}_{\sigma,\tau}$. As stated in the proof of Proposition A.1, the inclusion test of a configuration for being in a hierarchical stratum can be computed in $O(|S|^2)$ time, from which one conclude that the inclusion test for being in $\text{Portal}(\sigma, \tau)$ can also be computed in $O(|S|^2)$ time. If the given configuration is not a portal configuration, then the computation of $\text{Port}_{\sigma,\tau}$ (2.33) requires cluster centroids of σ , which can be computed in linear, $O(|S|)$, time as described in the proof of Proposition A.1. Given cluster centroids, one can compute Ctr (2.34) and Sc1 (2.35) in linear, $O(|S|)$, time, because the Napoleon transformation NT of an arbitrary triangle can be computed in constant, $O(1)$, time (refer to Appendix B). Finally, given the cluster centroids, each iteration of Mrg (2.37) can be computed in linear $O(|S|)$ time; and so all iterations of Mrg can be performed in $O(|S|^2)$ time since it may require at most $|S|$ iterations. Thus, the result follows. \blacksquare

A.2.2 Proof of Proposition A.1

Proof Recall from (2.15) that for any singleton cluster $I \in \mathcal{C}(\tau)$, we have $\mathcal{D}_A(I) = \mathfrak{S}(\tau)$. Hence, for any given $\mathbf{x} \in \mathfrak{S}(\tau)$, the base condition $\mathbf{x} \in \mathcal{D}_A(I)$ (case 2 in Table 2.5) always holds at any singleton cluster $I \in \mathcal{C}(\tau)$. Moreover, the cardinality of any cluster passed as an argument in a recursive step of the evaluation must decrease relative to the calling cluster size (case 7 in Table 2.5). Therefore, the recursion in Table 2.5 terminates, in the worst case, after visiting all clusters of τ only once.

Since all vector fields (f_A (2.14), f_H (2.19) and f_S (2.24)) used in Table 2.5 are well defined over the entirety of their domain $\mathfrak{S}(\tau)$ with codomain $(\mathbb{R}^n)^S$, the recursion in Table 2.5 results in a true function, $f_{\tau, \mathbf{y}} : \mathfrak{S}(\tau) \rightarrow (\mathbb{R}^n)^S$, with well defined evaluation for each configuration $\mathbf{x} \in \mathfrak{S}(\tau)$.

We now assess the computational complexity of the recursion in Table 2.5. Centroids of clusters of τ can be computed all at once in $O(|S|)$ time using the post-order traversal of τ and the following recursive relation of cluster centroids: for any disjoint $A, B \subset S$,

$$c(\mathbf{x}|A \cup B) = \frac{|A|}{|A|+|B|}c(\mathbf{x}|A) + \frac{|B|}{|A|+|B|}c(\mathbf{x}|B). \quad (\text{A.26})$$

Given cluster centroids, $\eta_{i,I,\tau}(\mathbf{x})$ (2.7) can be computed in constant, $O(1)$, time for any $i \in I$ and $I \in \mathcal{C}(\tau)$. Hence, since $|\mathcal{C}(\tau)| = 2|S| - 1$ for any $\tau \in \mathcal{BT}_S$ and $I = \{k \mid k \in K, K \in \text{Ch}(I, \tau)\}$ for any nonsingleton cluster $I \in \mathcal{C}(\tau)$, we conclude:

- The inclusion test for being in $\mathfrak{S}(\tau)$ (2.6) can be computed in $O(|S|^2)$.
- Given $\mathbf{x} \in \mathfrak{S}(\tau)$, the inclusion test for being in $\mathcal{D}_H(I)$ (2.18) for any cluster $I \in \mathcal{C}(\tau)$ can be computed in $O(|S|)$ time; and the recursion in Table 2.5 requires such inclusion tests at most only once for all clusters of τ , all of which can be computed in $O(|S|^2)$ time.
- The vector fields f_A (2.14), f_H (2.19) and f_S (2.24) at any cluster $I \in \mathcal{C}(\tau)$ can be computed in $O(|S|)$ time; and, once again, the recursion in Table 2.5 requires such computation at most at every cluster of τ , all of which can be performed in $O(|S|^2)$ time.

Finally, to conclude that $f_{\tau, \mathbf{y}}$ is computable in $O(|S|^2)$ time, we show that the inclusion test for being in $\mathcal{D}_A(I)$ (2.15) for all clusters $I \in \mathcal{C}(\tau)$ can be efficiently computed in $O(|S|^2)$ time as follows. Given cluster centroids, $\mathcal{L}_{\vec{\mathbf{y}}}(\mathbf{x}_k - \mathbf{m}_K(\mathbf{x}))^T \mathbf{s}_K(\mathbf{x})$ (2.17) can be computed in constant, $O(1)$, time for any $k \in K$, $K \in \mathcal{C}(\tau)$; and, likewise, $\mathcal{L}_{\vec{\mathbf{y}}} \frac{1}{2} \|\mathbf{x}_i - \mathbf{x}_j\|^2$ (2.16) can be computed in constant $O(1)$ time for any given pair $i \neq j \in S$. Further, using (2.15) and hierarchical relations of clusters, observe the following recursive relation of $\mathcal{D}_A(I)$: for any nonsingleton $I \in \mathcal{C}(\tau)$ and $\{I_L, I_R\} = \text{Ch}(I, \tau)$,

$$\mathcal{D}_A(I) = \mathcal{D}_A(I_L) \cap \mathcal{D}_A(I_R) \cap \widehat{\mathcal{D}}_A(I_L, I_R), \quad (\text{A.27})$$

subject to the base condition $\mathcal{D}_A(I) = \mathfrak{S}(\tau)$ for any singleton cluster $I \in \mathcal{C}(\tau)$, where

$$\begin{aligned} \widehat{\mathcal{D}}_A(I_L, I_R) := \left\{ \mathbf{x} \in \mathfrak{S}(\tau) \mid \mathcal{L}_{\vec{\mathbf{y}}} \frac{1}{2} \|\mathbf{x}_i - \mathbf{x}_j\|^2 \geq (r_i + r_j)^2, \quad \forall i \in I_L, j \in I_R, \right. \\ \left. \mathcal{L}_{\vec{\mathbf{y}}}(\mathbf{x}_k - \mathbf{m}_K(\mathbf{x}))^T \mathbf{s}_K(\mathbf{x}) \geq 0, \quad \forall k \in K, K \in \{I_L, I_R\} \right\}. \quad (\text{A.28}) \end{aligned}$$

Note that, given $\mathbf{x} \in \mathfrak{S}(\tau)$, the inclusion test for being in $\widehat{\mathcal{D}}_A(I_L, I_R)$ for the children $\{I_L, I_R\} = \text{Ch}(I, \tau)$ of any nonsingleton cluster $I \in \mathcal{C}(\tau)$ can be computed in $O(|I_L| |I_R| + |I_L| + |I_R|)$ time.

Hence, given $\mathbf{x} \in \mathfrak{S}(\tau)$, the inclusion test for being in \mathcal{D}_A (2.15) for any cluster $I \in \mathcal{C}(\tau)$ and all its descendants in $\text{Des}(I, \tau)$ can be computed at once in $O(|I|^2)$ time using the post-order traversal of the subtree of τ rooted at I and the recursive formulation (A.27) of $\mathcal{D}_A(I)$. This can be verified as follows. First, observe that the cluster set $\mathcal{C}(\tau)$ of τ can be

recursively defined as:

$$\bullet \text{ (Base Step) } \{j\} \in \mathcal{C}(\tau) \text{ for all } j \in S. \quad (\text{A.29a})$$

$$\bullet \text{ (Recursion) If } I, I^{-\tau} \in \mathcal{C}(\tau) \setminus \{S\}, \text{ then } \Pr(I, \tau) \in \mathcal{C}(\tau). \quad (\text{A.29b})$$

Accordingly, we provide a proof by structural induction [187]. For any $I \in \mathcal{C}(\tau)$:

- (Base Case) If I is singleton, then the result simply holds since any singleton cluster $I \in \mathcal{C}(\tau)$ has no descendant in τ and satisfies $\mathcal{D}_A(I) = \mathfrak{S}(\tau)$.
- (Induction) Otherwise ($|I| \geq 2$), let $\{I_L, I_R\} = \text{Ch}(I, \tau)$. (Induction hypothesis) Suppose that the inclusion test for being in \mathcal{D}_A for any child $K \in \text{Ch}(I, \tau)$ and all its descendant in $\text{Des}(K, \tau)$ is computable in $O(|K|^2)$. Then, by the recursion in (A.27), the inclusion test for being in \mathcal{D}_A for cluster I and all its descendants in $\text{Des}(I, \tau)$ only requires the extra test for being in $\widehat{\mathcal{D}}_A(I_L, I_R)$ for the children $\{I_L, I_R\} = \text{Ch}(I, \tau)$ in addition to the inclusion test for every child $K \in \text{Ch}(I, \tau)$ and its descendants in $\text{Des}(K, \tau)$. Hence, the total computation cost for cluster I and its descendants in τ is $O(|I_L|^2) + O(|I_R|^2) + O(|I_L||I_R| + |I_R| + |I_L|) = O(|I|^2)$.

Therefore, since $\mathcal{C}(\tau) = \{S\} \cup \text{Des}(S, \tau)$, given $\mathbf{x} \in \mathfrak{S}(\tau)$, the inclusion test for being in $\mathcal{D}_A(I)$ for all clusters $I \in \mathcal{C}(\tau)$ can be computed at once in $O(|S|^2)$ time, and this completes the proof. \blacksquare

A.2.3 Proof of Proposition A.2

Proof To demonstrate how the recursion in Table A.2 recursively composes continuous vector fields, we shall recast f_A (2.14), f_H (2.19), f_S (2.24) and the recursion $\hat{h}_{s,b}$ (Table A.2) as follows: for any cluster $I \in \mathcal{V}_\tau(S)$ (A.5) visited during recursive computation of $h_{s,b}$,

$$\begin{aligned} f_A^I : \mathfrak{S}(\tau) \times (\mathbb{R}^n)^S &\rightarrow \mathfrak{S}(\tau) \times (\mathbb{R}^n)^S \\ (\mathbf{x}, \mathbf{u}) &\mapsto (\mathbf{x}, f_A(\mathbf{x}, \mathbf{u}, I)) \end{aligned} \quad (\text{A.30})$$

$$\begin{aligned} f_H^I : \mathfrak{S}(\tau) \times (\mathbb{R}^n)^S &\rightarrow \mathfrak{S}(\tau) \times (\mathbb{R}^n)^S \\ (\mathbf{x}, \mathbf{u}) &\mapsto (\mathbf{x}, f_H(\mathbf{x}, \mathbf{u}, I)) \end{aligned} \quad (\text{A.31})$$

$$\begin{aligned} f_S^I : \mathfrak{S}(\tau) \times (\mathbb{R}^n)^S &\rightarrow \mathfrak{S}(\tau) \times (\mathbb{R}^n)^S \\ (\mathbf{x}, \mathbf{u}) &\mapsto (\mathbf{x}, f_S(\mathbf{x}, \mathbf{u}, I)) \end{aligned} \quad (\text{A.32})$$

$$\begin{aligned} \hat{h}_{s,b}^I : \mathfrak{S}(\tau) \times (\mathbb{R}^n)^S &\rightarrow \mathfrak{S}(\tau) \times (\mathbb{R}^n)^S \\ (\mathbf{x}, \mathbf{u}) &\mapsto (\mathbf{x}, \hat{h}_{s,b}(\mathbf{x}, \mathbf{u}, I)) \end{aligned} \quad (\text{A.33})$$

Note that, by definition, $f_A^I(\mathbf{x}, \mathbf{u})$ is smooth in both \mathbf{x} and \mathbf{u} , and $f_H^I(\mathbf{x}, \mathbf{u})$ and $f_S^I(\mathbf{x}, \mathbf{u})$ are continuous and piecewise smooth functions of \mathbf{x} and \mathbf{u} , because functions defined by the maximum of a finite collection of smooth functions are continuous and piecewise smooth, and the product of continuous and piecewise smooth functions are also continuous and piecewise smooth [50].

We now show that, for any $I \in \mathcal{V}_\tau(S)$, $\hat{h}_{s,b}^I(\mathbf{x}, \mathbf{u})$ is continuous and piecewise smooth in \mathbf{x} and \mathbf{u} . First, observe from Lemma A.4 that the set $\mathcal{V}_\tau(S)$ (A.5) can be recursively

defined as

$$\bullet \text{ (Base Step) } I \in \mathcal{V}_\tau(\mathcal{S}) \text{ for all } I \in \mathcal{S}. \quad (\text{A.34a})$$

$$\bullet \text{ (Recursion) If } I, I^{-\tau} \in \mathcal{V}_\tau(\mathcal{S}) \setminus \{S\}, \text{ then } \Pr(I, \tau) \in \mathcal{V}_\tau(\mathcal{S}). \quad (\text{A.34b})$$

Accordingly, we provide a proof by structural induction [187]. For any cluster $I \in \mathcal{V}_\tau(\mathcal{S})$:

- (Base Case) If $I \in \mathcal{S}$, then we have

$$\hat{h}_{\mathcal{S}, \mathbf{b}}^I(\mathbf{x}, \mathbf{u}) = \begin{cases} f_A^I(\mathbf{x}, \mathbf{u}), & \text{if } b_I = +1, \\ f_S^I(\mathbf{x}, \mathbf{u}), & \text{if } b_I = -1, \end{cases} \quad (\text{A.35})$$

which is continuous and piecewise smooth in \mathbf{x} and \mathbf{u} .

- (Induction) Else, we have $|I| \geq 2$ and so let $\{I_L, I_R\} = \text{Ch}(I, \tau)$. (Induction hypothesis) Suppose $\hat{h}_{\mathcal{S}, \mathbf{b}}^{I_L}(\mathbf{x}, \mathbf{y})$ and $\hat{h}_{\mathcal{S}, \mathbf{b}}^{I_R}(\mathbf{x}, \mathbf{y})$ are continuous and piecewise smooth. Then, one can verify from Table A.2 that

$$\hat{h}_{\mathcal{S}, \mathbf{b}}^I(\mathbf{x}, \mathbf{u}) = \left(f_H^I \circ \hat{h}_{\mathcal{S}, \mathbf{b}}^{I_R} \circ \hat{h}_{\mathcal{S}, \mathbf{b}}^{I_L} \right)(\mathbf{x}, \mathbf{u}). \quad (\text{A.36})$$

Hence, $\hat{h}_{\mathcal{S}, \mathbf{b}}^I$ is a composition of continuous and piecewise smooth functions, and it must remain so as well [50].

Thus, the result follows since $(\mathbf{x}, h_{\mathcal{S}, \mathbf{b}}(\mathbf{x})) = \hat{h}_{\mathcal{S}, \mathbf{b}}^S(\mathbf{x}, \mathbf{0})$. \blacksquare

A.2.4 Proof of Proposition A.3

Proof Since the recursion in Table A.3 uses only clusters of τ and guarantees, in cases 4, 7 and 13 in Table A.3, that the dimension of \mathbf{b} is equal to the cardinality of \mathcal{S} , the output $(\mathcal{S}, \mathbf{b}) = p(\mathbf{x})$ associated with any configuration $\mathbf{x} \in \mathfrak{S}(\tau)$ always satisfies that $\mathcal{S} \subset \mathcal{C}(\tau)$ and $\mathbf{b} \in \{-1, +1\}^{\mathcal{S}}$.

To prove that \mathcal{S} is a partition of S , we shall show that, for any $\mathbf{x} \in \mathfrak{S}(\tau)$ and $I \in \mathcal{C}(\tau)$, $(\hat{\mathcal{J}}, \hat{\mathbf{b}}) = \hat{p}(\mathbf{x}, I)$ yields a partition $\hat{\mathcal{J}}$ of I . Based on the recursive definition (A.29) of $\mathcal{C}(\tau)$, we now provide a proof by structural induction. For any $\mathbf{x} \in \mathfrak{S}(\tau)$ and $I \in \mathcal{C}(\tau)$, let $(\hat{\mathcal{J}}, \hat{\mathbf{b}}) = \hat{p}(\mathbf{x}, I)$, then:

- (Base Case) If I is singleton, then $\mathcal{D}_A(I) = \mathfrak{S}(\tau)$ and the base condition in case 2 in Table A.3 holds. Hence, we have $\hat{\mathcal{J}} = \{I\}$, the trivial partition of I , and so the result follows.
- (Induction) Otherwise ($|I| \geq 2$), we have two possibilities.
 - If I satisfies any base condition in cases 2 and 5 in Table A.3, i.e., $\mathbf{x} \in \mathcal{D}_A(I) \cup (\mathfrak{S}(\tau) \setminus \mathcal{D}_H(I))$, then we have $\hat{\mathcal{J}} = \{I\}$, and the result directly follows.
 - Else (the recursion condition in cases 8-14 in Table A.3 holds), since $|I| \geq 2$, let $\{I_L, I_R\} = \text{Ch}(I, \tau)$ and $(\hat{\mathcal{J}}_L, \hat{\mathbf{b}}_L) = \hat{p}(\mathbf{x}, I_L)$ and $(\hat{\mathcal{J}}_R, \hat{\mathbf{b}}_R) = \hat{p}(\mathbf{x}, I_R)$. (Induction Hypothesis) Suppose that $\hat{\mathcal{J}}_L$ and $\hat{\mathcal{J}}_R$ are partitions of I_L and I_R , respectively. Then, since $\hat{\mathcal{J}} = \hat{\mathcal{J}}_L \cup \hat{\mathcal{J}}_R$ (see case 12 in Table A.3) and $\text{Ch}(I, \tau)$ is a bipartition of I , we observe that $\hat{\mathcal{J}}$ is a partition of I .

Hence, since $(\mathcal{S}, \mathbf{b}) = p(\mathbf{x}) = \hat{p}(\mathbf{x}, S)$, the recursion in Table A.3 terminates with a partition \mathcal{S} of S . Thus, since the policy selection algorithm is deterministic, p is a well-defined function from $\mathfrak{S}(\tau)$ to $\mathcal{SP}_S(\tau)$ (A.2).

Finally, we shall show that $(\mathcal{S}, \mathbf{b}) = p(\mathbf{x})$ is the index of a local control policy whose domain $\mathcal{D}(\mathcal{S}, \mathbf{b})$ contains \mathbf{x} , i.e., $\mathbf{x} \in (\mathcal{D} \circ p)(\mathbf{x})$. Using the base conditions in cases 2-7 in Table A.3, one can verify that for any $I \in \mathcal{S}$, if $b_I = +1$, then $\mathbf{x} \in \mathcal{D}_A(I)$; and if $b_I = -1$, then $\mathbf{x} \in \mathfrak{S}(\tau) \setminus (\mathcal{D}_A(I) \cup \mathcal{D}_H(I)) \subset \mathfrak{S}(\tau)$. Hence, the base conditions guarantee that $\mathbf{x} \in \mathcal{D}_B(I, b_I)$ (A.4) for any $I \in \mathcal{S}$. Observe that during the recursive computation of p in Table A.3, to reach any cluster $I \in \mathcal{S}$ satisfying a base condition, every ancestor $K \in \text{Anc}(I, \tau)$ of I must have been recursively visited. A recursion (see cases 8-14 in Table A.3) at any ancestor $K \in \text{Anc}(I, \tau)$ of I in τ implies that $\mathbf{x} \in \mathcal{D}_H(K) \setminus \mathcal{D}_A(K) \subset \mathcal{D}_H(K)$. Thus, by definition (A.3), we have $\mathbf{x} \in \mathcal{D}(\mathcal{S}, \mathbf{b})$, and the result follows. ■

A.2.5 Proof of Proposition A.4

Proof For any given $\mathbf{x} \in \mathfrak{S}(\tau)$, the recursions in Table 2.5 and Table A.3 traverse the same clusters of τ in the same order, because both recursions have identical base and recursion conditions.

Now observe that the tree traversal pattern used by the recursion in Table A.2 is fixed for a given policy index $(\mathcal{S}, \mathbf{b}) \in \mathcal{SP}_{\mathcal{S}}(\tau)$: a base condition is satisfied at any cluster $I \in \mathcal{S}$, and to reach such cluster I all its ancestors $\text{Anc}(I, \tau)$ must have been recursively visited starting from the root cluster S . Recall from the proof of Proposition A.3 that $(\mathcal{S}, \mathbf{b}) = p(\mathbf{x})$ yields a partition \mathcal{S} of S such that a base condition in Table A.3 holds for every block $I \in \mathcal{S}$ and all its ancestors in $\text{Anc}(I, \tau)$ are recursively visited. Hence, if the policy index is selected as $(\mathcal{S}, \mathbf{b}) = p(\mathbf{x})$, the recursion in Table A.2 computing $h_{\mathcal{S}, \mathbf{b}}(\mathbf{x})$ always follows the tree traversal pattern used by the recursion in Table A.3 computing $p(\mathbf{x})$.

Thus, for a given configuration, all recursions in Table 2.5, Table A.2 and Table A.3 share a common tree traversal pattern.

Let $\mathbf{x} \in \mathfrak{S}(\tau)$ and $(\mathcal{S}, \mathbf{b}) = p(\mathbf{x})$, and observe from Table A.3 that for any $I \in \mathcal{S}$, if $b_I = +1$, then $\mathbf{x} \in \mathcal{D}_A(I)$; and if $b_I = -1$, then $\mathbf{x} \in \mathfrak{S}(\tau) \setminus (\mathcal{D}_A(I) \cup \mathcal{D}_H(I))$; and $\mathbf{x} \in \mathcal{D}_H(K) \setminus \mathcal{D}_A(K)$ for all $K \in \text{Anc}(I, \tau)$. Using this relation between policy indices and domains, one can conclude that the recursions in Table 2.5 and Table A.2 use the same vector fields for the identical base and recursive steps. Thus, the result follows. ■

A.2.6 Proof of Proposition A.5

Proof According to Definition A.2, any pair $((\mathcal{S}, \mathbf{b}), (\mathcal{S}', \mathbf{b}'))$ of policy indices in $\widehat{\mathcal{E}}_{\mathcal{PG}}$ satisfies at least one of Lemmas A.9 - A.11. Hence, $h_{\mathcal{S}, \mathbf{b}}$ prepares $h_{\mathcal{S}', \mathbf{b}'}$ in finite time, and $\text{priority}(\mathcal{S}', \mathbf{b}') > \text{priority}(\mathcal{S}, \mathbf{b})$. Thus, $((\mathcal{S}, \mathbf{b}), (\mathcal{S}', \mathbf{b}'))$ is also an edge of the prepares graph \mathcal{PG} .

Moreover, for any $(\mathcal{S}, \mathbf{b}) \neq (\{S\}, +1)$, there always exists a policy index $(\mathcal{S}', \mathbf{b}') \neq (\mathcal{S}, \mathbf{b})$ such that $((\mathcal{S}, \mathbf{b}), (\mathcal{S}', \mathbf{b}'))$ is an edge of $\widehat{\mathcal{PG}}$. This can be observed as follows. Since \mathcal{S} is compatible with τ , i.e., $\mathcal{S} \subset \mathcal{C}(\tau)$, if $|\mathcal{S}| > 1$, then there exists a nonsingleton cluster $I \in \mathcal{C}(\tau)$ such that $\text{Ch}(I, \tau) \subset \mathcal{S}$ (Lemma A.4). Hence, at least one of the following always holds:

- (a) There exists a cluster $I \in \mathcal{S}$ with $b_I = -1$.
- (b) There exists a cluster $I \in \mathcal{C}(\tau)$ such that $\text{Ch}(I, \tau) \subset \mathcal{S}$ and $b_K = +1$ for all $K \in \text{Ch}(I, \tau)$.

And $(\mathcal{S}', \mathbf{b}')$ can be selected accordingly to satisfy one of the connectivity conditions of $\widehat{\mathcal{PG}}$ (cases (i)-(iii) in Definition A.2).

Since every policy index $(\mathcal{S}, \mathbf{b}) \neq (\{\mathcal{S}\}, +1)$ has an adjacent policy index $(\mathcal{S}', \mathbf{b}')$ in $\widehat{\mathcal{PG}}$ and $\text{priority}(\mathcal{S}', \mathbf{b}') > \text{priority}(\mathcal{S}, \mathbf{b})$, $\widehat{\mathcal{PG}}$ has no cycle and all of its nodes connected to the goal policy index $(\{\mathcal{S}\}, +1)$ through directed paths along which **priority** is strictly increasing. Note that the goal policy index has the highest **priority** value, which is $|\mathcal{S}|^2$ (A.9). Further, since **priority** (A.8) is integer valued function whose range (A.9) is $[-|\mathcal{S}|^2, |\mathcal{S}|^2]$, the length of a directed path in $\widehat{\mathcal{PG}}$ is bounded above by $O(|\mathcal{S}|^2)$ hops, and so the result follows. ■

A.2.7 Proof of Proposition A.6

Proof If there is only one local controller whose domain contains \mathbf{x} , then the result follows from Proposition A.3.

Otherwise, we shall provide a proof by contradiction. Let $(\mathcal{S}, \mathbf{b}) = p(\mathbf{x})$, and $(\mathcal{S}', \mathbf{b}') \neq (\mathcal{S}, \mathbf{b})$ be the index of a local controller whose domain $\mathcal{D}(\mathcal{S}', \mathbf{b}')$ (A.3) contains \mathbf{x} . Suppose that the local controller $h_{\mathcal{S}', \mathbf{b}'}$ has the maximum priority among all local controllers whose domains contain \mathbf{x} . We shall show below that there always exists another local controller whose domain contains \mathbf{x} and it has a higher priority than $\text{priority}(\mathcal{S}', \mathbf{b}')$, which is a contradiction.

It follows from Lemma A.5 that at least one of the followings always holds:

- Case 1 (\mathcal{S} Partially Refines \mathcal{S}'): There exists a cluster $K' \in \mathcal{S}'$ with a nontrivial partition \mathcal{K}' (i.e., $|\mathcal{K}'| \geq 2$) such that $\mathcal{K}' \subset \mathcal{S}$. Since $\mathbf{x} \in \mathcal{D}(\mathcal{S}, \mathbf{b})$ and all the elements of \mathcal{K}' are descendants of K' in τ , the recursive tree traversal in Table A.3 requires that $\mathbf{x} \in \mathcal{D}_H(K') \setminus \mathcal{D}_A(K')$. Hence, $b'_{K'} = -1$. Since $(a+b)^2 > a^2 + b^2$ for any $a, b \in \mathbb{R}_+$, one can observe that replacing K' of \mathcal{S}' with the elements of \mathcal{K}' and updating \mathbf{b}' with the associated binary values from \mathbf{b} yields the index $(\mathcal{S}'', \mathbf{b}'')$ of another local controller,

$$\mathcal{S}'' = \mathcal{K}' \cup \mathcal{S}' \setminus \{K'\}, \quad (\text{A.37})$$

$$\mathbf{b}'' = (b''_I)_{I \in \mathcal{S}''} \text{ s.t. } b''_I = \begin{cases} b_I, & \text{if } I \in \mathcal{K}', \\ b'_I, & \text{if } I \in \mathcal{S}' \setminus \{K'\}, \end{cases} \quad (\text{A.38})$$

at a strictly higher priority,

$$\text{priority}(\mathcal{S}'', \mathbf{b}'') = \text{priority}(\mathcal{S}', \mathbf{b}') + \underbrace{|\mathcal{K}'|^2 + \sum_{I' \in \mathcal{K}'} b_{I'} |I'|^2}_{> 0} > \text{priority}(\mathcal{S}', \mathbf{b}'). \quad (\text{A.39})$$

Note that we still have $\mathbf{x} \in \mathcal{D}(\mathcal{S}'', \mathbf{b}'')$, because $\mathbf{x} \in \mathcal{D}(\mathcal{S}, \mathbf{b}) \cap \mathcal{D}(\mathcal{S}', \mathbf{b}')$.

- Case 2 (\mathcal{S}' Partially Refines \mathcal{S}): There exists a cluster $K \in \mathcal{S}$ with a nontrivial partition \mathcal{K} (i.e., $|\mathcal{K}| \geq 2$) such that $\mathcal{K} \subset \mathcal{S}'$. Since $K \in \mathcal{S}$, one of the base conditions in Table A.3 at cluster K holds, and we therefore have either $\mathbf{x} \in \mathcal{D}_A(K)$ or $\mathbf{x} \notin \mathcal{D}_H(K)$. Further, since $\mathbf{x} \in \mathcal{D}(\mathcal{S}', \mathbf{b}')$ and K is an ancestor of all the elements of \mathcal{K} in τ , we also have $\mathbf{x} \in \mathcal{D}_H(K)$. Thus, $\mathbf{x} \in \mathcal{D}_A(K)$ and $b_K = +1$. Once again, since $(a+b)^2 > a^2 + b^2$ for any $a, b \in \mathbb{R}_+$ and $\mathbf{x} \in \mathcal{D}(\mathcal{S}, \mathbf{b}) \cap \mathcal{D}(\mathcal{S}', \mathbf{b}')$,

one can verify that the following local policy index

$$\mathcal{S}'' = \{K\} \cup \mathcal{S}' \setminus \mathcal{K} , \quad (\text{A.40})$$

$$\mathbf{b}'' = (\mathbf{b}_I'')_{I \in \mathcal{S}''} \text{ s.t. } \mathbf{b}_I'' = \begin{cases} +1, & \text{if } I = K , \\ \mathbf{b}_I', & \text{if } I \in \mathcal{S}' \setminus \mathcal{K} , \end{cases} \quad (\text{A.41})$$

has a strictly higher priority,

$$\text{priority}(\mathcal{S}'', \mathbf{b}'') = \text{priority}(\mathcal{S}', \mathbf{b}') + \underbrace{|K|^2 - \sum_{I \in \mathcal{K}} \mathbf{b}_I |I|^2}_{> 0} > \text{priority}(\mathcal{S}', \mathbf{b}') , \quad (\text{A.42})$$

and its domain contains \mathbf{x} , i.e., $\mathbf{x} \in \mathcal{D}(\mathcal{S}'', \mathbf{b}'')$.

- Case 3 (Identical Resolution): $\mathcal{S}' = \mathcal{S}$ and $\mathbf{b}' \neq \mathbf{b}$. Since $\mathcal{S}' = \mathcal{S}$, one can maximize $\text{priority}(\mathcal{S}', \mathbf{b}')$ (A.8) by maximizing the binary vector \mathbf{b}' , which is achieved by setting $\mathbf{b}_I' = +1$ for any $I \in \mathcal{S}'$ whenever $\mathbf{x} \in \mathcal{D}_A(I)$. The base conditions in Table A.3 guarantee such an optimal selection of \mathbf{b}' . However, since $\mathbf{b}' \neq \mathbf{b}$, we have

$$\text{priority}(\mathcal{S}, \mathbf{b}) > \text{priority}(\mathcal{S}', \mathbf{b}') , \quad (\text{A.43})$$

which completes the proof. ■

A.2.8 Proof of Proposition A.8

Proof The continuity and piecewise smoothness of $h_{\mathcal{S}, \mathbf{b}}$ (Proposition A.2) implies its locally Lipschitz continuity in $\mathfrak{S}(\tau)$ [135]; and the existence of at least one trajectory of $h_{\mathcal{S}, \mathbf{b}}$ starting at \mathbf{x} follows from its continuity.

Let \mathbf{x}^t denote a trajectory of $h_{\mathcal{S}, \mathbf{b}}$ starting at any $\mathbf{x}^0 \in \mathcal{D}(\mathcal{S}, \mathbf{b})$ for all $t \geq 0$. We have from Proposition A.7 that \mathbf{x}^t remains in $\mathcal{D}(\mathcal{S}, \mathbf{b})$ for all $t \geq 0$. Further, by Lemma A.2, the centroidal trajectory $\mathbf{c}(\mathbf{x}^t | \mathcal{S})$ is guaranteed to lie on the line segment joining $\mathbf{c}(\mathbf{x}^0 | \mathcal{S})$ and $\mathbf{c}(\mathbf{y} | \mathcal{S})$; and, by Lemma A.3, the centroidal configuration radius $r(\mathbf{x}^t | \mathcal{S})$ (2.12) is bounded above by a certain finite value, $R(\mathbf{x}^0, \mathbf{y})$, depending only on the initial and desired configurations, \mathbf{x}^0 and \mathbf{y} , respectively. Thus, all trajectories of $h_{\mathcal{S}, \mathbf{b}}$ stay in a compact subset W of $\mathcal{D}(\mathcal{S}, \mathbf{b})$ and the compact set defined by the Minkowski sum of the line segment joining $\mathbf{c}(\mathbf{x}^0 | \mathcal{S})$ and $\mathbf{c}(\mathbf{y} | \mathcal{S})$ and the closed ball centered at the origin with radius of $R(\mathbf{x}^0, \mathbf{y})$.

Given that all trajectories of $h_{\mathcal{S}, \mathbf{b}}$ starting at any $\mathbf{x} \in \mathcal{D}(\mathcal{S}, \mathbf{b})$ lie in a compact subset W of $\mathcal{D}(\mathcal{S}, \mathbf{b})$, the uniqueness of its flow follows from the Lipschitz continuity of $h_{\mathcal{S}, \mathbf{b}}$ in W , because a locally Lipschitz function on $\mathfrak{S}(\tau)$ is Lipschitz on every compact subset of $\mathfrak{S}(\tau)$, also refer to Theorem 3.3 in [127]. Moreover, this unique flow is continuous and piecewise smooth since it is the integral of the continuous and piecewise smooth vector field $h_{\mathcal{S}, \mathbf{b}}$ [200], which completes the proof. ■

A.2.9 Proof of Proposition A.7

Before proceeding with the proof of Proposition A.7, we find it useful to emphasize some critical properties of a trajectory \mathbf{x}^t of $h_{\mathcal{S}, \mathbf{b}}(\mathbf{x})$ starting at any $\mathbf{x}^0 \in \mathcal{D}(\mathcal{S}, \mathbf{b})$.

Lemma A.6 A trajectory \mathbf{x}^t of $h_{\mathcal{S}, \mathbf{b}}(\mathbf{x})$ (see Table A.2) starting at any initial configuration $\mathbf{x}^0 \in \mathcal{D}(\mathcal{S}, \mathbf{b})$ (A.3) satisfies the following properties for any $I \in \mathcal{S}$ with $b_I = +1$ and $t \geq 0$,

- (i) $\mathcal{L}_{\vec{y}} \frac{1}{2} \|\mathbf{x}_i^t - \mathbf{x}_j^t\|^2 \geq (r_i + r_j)^2, \quad \forall i \neq j \in I,$
- (ii) $\mathcal{L}_{\vec{y}} (\mathbf{x}_k^t - \mathbf{m}_K(\mathbf{x}^t))^T \mathbf{s}_K(\mathbf{x}^t) \geq 0, \quad \forall k \in K, K \in \text{Des}(I, \tau),$
- (iii) $\eta_{k,K}(\mathbf{x}^t) \geq 0, \quad \forall k \in K, K \in \text{Des}(I, \tau),$
- (iv) $\|\mathbf{x}_i^t - \mathbf{x}_j^t\|^2 > (r_i + r_j)^2, \quad \forall i \neq j \in I.$

Proof See Appendix A.2.13. ■

Lemma A.7 A trajectory \mathbf{x}^t of $h_{\mathcal{S}, \mathbf{b}}(\mathbf{x})$ (see Table A.2) starting at any initial configuration $\mathbf{x}^0 \in \mathcal{D}(\mathcal{S}, \mathbf{b})$ (A.3) satisfies the following properties for any $I \in \mathcal{S}$ with $b_I = -1$ and $t \geq 0$,

- (i) $\eta_{k,K}(\mathbf{x}^t) \geq 0, \quad \forall k \in K, K \in \text{Des}(I, \tau),$
- (ii) $\|\mathbf{x}_i^t - \mathbf{x}_j^t\|^2 > (r_i + r_j)^2, \quad \forall i \neq j \in I.$

Proof See Appendix A.2.14. ■

Lemma A.8 Let $\mathcal{V}_\tau(\mathcal{S})$ (A.5) be the set of clusters visited during the recursive computation of $h_{\mathcal{S}, \mathbf{b}}(\mathbf{x})$ in Table A.2.

Then a trajectory \mathbf{x}^t of $h_{\mathcal{S}, \mathbf{b}}(\mathbf{x})$ starting at any initial configuration $\mathbf{x}^0 \in \mathcal{D}(\mathcal{S}, \mathbf{b})$ (A.3) satisfies the following properties for any $I \in \mathcal{V}_\tau(\mathcal{S}) \setminus \mathcal{S}$ and $t \geq 0$,

- (i) $\eta_{k,K}(\mathbf{x}^t) \geq r_k + \alpha, \quad \forall k \in K, K \in \text{Ch}(I, \tau),$
- (ii) $\|\mathbf{x}_i^t - \mathbf{x}_j^t\|^2 > (r_i + r_j)^2, \quad \forall i \in K, j \in I \setminus K, K \in \text{Ch}(I, \tau).$

Proof See Appendix A.2.15. ■

Accordingly, we conclude the positive invariance of the domain $\mathcal{D}(\mathcal{S}, \mathbf{b})$ of $h_{\mathcal{S}, \mathbf{b}}$ as follows:

Proof of Proposition A.7 By cases (iii)-(iv) in Lemma A.6 and cases (i)-(ii) in Lemmas A.7-A.8, a trajectory \mathbf{x}^t of $h_{\mathcal{S}, \mathbf{b}}$ starting at any $\mathbf{x}^0 \in \mathcal{D}(\mathcal{S}, \mathbf{b})$ is guaranteed to remain in $\mathfrak{S}(\tau)$ for all future time. Given $\mathbf{x}^t \in \mathfrak{S}(\tau)$ for all $t \geq 0$, cases (i)-(ii) in Lemma A.6 imply $\mathbf{x}^t \in \mathcal{D}_A(I)$ for any $I \in \mathcal{S}$ with $b_I = +1$; and case (iii) in Lemma A.8 implies $\mathbf{x}^t \in \mathcal{D}_H(K)$ for every ancestor $K \in \text{Anc}(I, \tau)$ of any $I \in \mathcal{S}$. Thus, by definition (A.3), we have $\mathbf{x}^t \in \mathcal{D}(\mathcal{S}, \mathbf{b})$ for all $t \geq 0$. ■

A.2.10 Proof of Proposition A.9

Here we first establish finite-time prepares relations between pairs of local policies whose indices are related to each other in a certain way as specified in Definition A.2; and then we continue with the proof of Proposition A.9.

Lemma A.9 (Case (i) in Definition A.2) Let $\mathcal{S} \in \mathcal{P}_S(\tau)$ be a partition of S and $\mathbf{b}, \mathbf{b}' \in \{-1, +1\}^{\mathcal{S}}$. If $b_I = b'_I$ for all $I \in \mathcal{S}$ but a singleton cluster $D \in \mathcal{S}$ where $b_D = -1$ and $b'_D = +1$, then the domains (A.3) of local control policies $h_{\mathcal{S}, \mathbf{b}}$ and $h_{\mathcal{S}, \mathbf{b}'}$ are identical, i.e.,

$$\mathcal{D}(\mathcal{S}, \mathbf{b}') = \mathcal{D}(\mathcal{S}, \mathbf{b}), \quad (\text{A.44})$$

and their priorities (A.8) satisfy

$$\text{priority}(\mathcal{S}, \mathbf{b}') = \text{priority}(\mathcal{S}, \mathbf{b}) + 2. \quad (\text{A.45})$$

Proof See Appendix A.2.16. ■

Lemma A.10 (Case (ii) in Definition A.2) *Let $\mathcal{S} \in \mathcal{P}_S(\tau)$ be a partition of S and $\mathbf{b} \in \{-1, +1\}^{\mathcal{S}}$ such that $b_I = -1$ for a nonsingleton cluster $I \in \mathcal{S}$; and let $\mathcal{S}' = \mathcal{S} \setminus \{I\} \cup \text{Ch}(I, \tau)$ and $\mathbf{b}' \in \{-1, +1\}^{\mathcal{S}'}$ with $b'_K = -1$ for all $K \in \text{Ch}(I, \tau)$ and $b'_D = b_D$ for all $D \in \mathcal{S} \setminus \{I\}$.*

Then all trajectories of the local control policy $h_{\mathcal{S}, \mathbf{b}}$ starting in its domain $\mathcal{D}(\mathcal{S}, \mathbf{b})$ reach in finite time the domain $\mathcal{D}(\mathcal{S}', \mathbf{b}')$ of the local controller $h_{\mathcal{S}', \mathbf{b}'}$ which has a higher priority (A.8) than $h_{\mathcal{S}, \mathbf{b}}$ does, i.e.,

$$\text{priority}(\mathcal{S}', \mathbf{b}') > \text{priority}(\mathcal{S}, \mathbf{b}). \quad (\text{A.46})$$

Proof See Appendix A.2.17. ■

Lemma A.11 (Case (iii) in Definition A.2) *Let $\mathcal{S} \in \mathcal{P}_S(\tau)$ be a partition of S and $\mathbf{b} \in \{-1, +1\}^{\mathcal{S}}$ such that $\text{Ch}(I, \tau) \subset \mathcal{S}$ for a nonsingleton cluster $I \in \mathcal{C}(\tau)$ and $b_K = +1$ for all $K \in \text{Ch}(I, \tau)$; and let $\mathcal{S}' = \mathcal{S} \setminus \text{Ch}(I, \tau) \cup \{I\}$ and $\mathbf{b}' \in \{-1, +1\}^{\mathcal{S}'}$ with $b'_I = +1$ and $b'_D = b_D$ for all $D \in \mathcal{S} \setminus \text{Ch}(I, \tau)$.*

Then the local control policy $h_{\mathcal{S}, \mathbf{b}}$ steers (almost) all configurations in its domain $\mathcal{D}(\mathcal{S}, \mathbf{b})$ in finite time to the domain $\mathcal{D}(\mathcal{S}', \mathbf{b}')$ of the local controller $h_{\mathcal{S}', \mathbf{b}'}$ which has a higher priority (A.8) than $h_{\mathcal{S}, \mathbf{b}}$ does, i.e.,

$$\text{priority}(\mathcal{S}', \mathbf{b}') > \text{priority}(\mathcal{S}, \mathbf{b}). \quad (\text{A.47})$$

Proof See Appendix A.2.18. ■

Proof of Proposition A.9 Since \mathcal{S} is a partition of S compatible with τ , i.e., $\mathcal{S} \subset \mathcal{C}(\tau)$, observe that if $|\mathcal{S}| > 1$, then there exists a cluster $I \in \mathcal{C}(\tau)$ such that $\text{Ch}(I, \tau) \subset \mathcal{S}$ (Lemma A.4). Hence, since $(\mathcal{S}, \mathbf{b}) \neq (\{S\}, +1)$, at least one of the followings always holds:

- (a) There exists $I \in \mathcal{S}$ such that $b_I = -1$. If $|I| = 1$, then we have the result by Lemma A.9; otherwise ($|I| > 1$), the results follows from Lemma A.10.
- (b) There exist a cluster $I \in \mathcal{C}(\tau)$ such that $\text{Ch}(I, \tau) \subset \mathcal{S}$ and $b_K = +1$ for all $K \in \text{Ch}(I, \tau)$. Accordingly, the results follows from Lemma A.11 and this completes the proof. ■

A.2.11 Proof of Lemma A.2

Proof For any cluster $I \in \mathcal{S}$, the recursion in Table A.2 employs a vector field satisfying the associated base condition, and then recursively constructs an additive repulsion field at every ancestor $\text{Anc}(I, \tau)$ of I , which can be explicitly written as follows: for any $i \in I$ and $I \in \mathcal{S}$,

- if $b_I = +1$, then we have

$$u_i = f_A(\mathbf{x}, \mathbf{0}, I)_i + \sum_{\substack{K \in \text{Anc}(I, \tau) \cup \{I\} \setminus \{S\} \\ R = \text{Pr}(K, \tau)}} 2\alpha_R(\mathbf{x}, \mathbf{v}_R) \frac{|K^{-\tau}|}{|R|} \frac{s_K(\mathbf{x})}{\|s_K(\mathbf{x})\|}, \quad (\text{A.48})$$

- else ($b_I = -1$),

$$u_i = f_S(\mathbf{x}, \mathbf{0}, I)_i + \sum_{\substack{K \in \text{Anc}(I, \tau) \cup \{I\} \setminus \{S\} \\ R = \text{Pr}(K, \tau)}} 2\alpha_R(\mathbf{x}, \mathbf{v}_R) \frac{|K^{-\tau}|}{|R|} \frac{s_K(\mathbf{x})}{\|s_K(\mathbf{x})\|}, \quad (\text{A.49})$$

for some $\mathbf{v}_R \in (\mathbb{R}^d)^S$ associated with cluster $R \in \text{Anc}(I, \tau)$.

Now, using (2.14) and (2.24), one can verify that for any $I \in \mathcal{S}$,

$$c(\mathbf{u}|I) = -c(\mathbf{x} - \mathbf{y}|I) + \sum_{\substack{K \in \text{Anc}(I, \tau) \cup \{I\} \setminus \{S\} \\ R = \text{Pr}(K, \tau)}} 2\alpha_R(\mathbf{x}, \mathbf{v}_R) \frac{|K^{-\tau}|}{|R|} \frac{s_K(\mathbf{x})}{\|s_K(\mathbf{x})\|}, \quad (\text{A.50})$$

which can be generalized to other clusters in $\mathcal{V}_\tau(\mathcal{S}) \setminus \mathcal{S}$. That is to say, we now show that for any $I \in \mathcal{V}_\tau(\mathcal{S})$, the centroidal dynamics $c(\mathbf{u}|I)$ satisfies (A.50). Based on the recursive definition (A.34) of $\mathcal{V}_\tau(\mathcal{S})$, we provide a proof by structural induction. For any $I \in \mathcal{V}_\tau(\mathcal{S})$,

- (Base Case) If $I \in \mathcal{S}$, then the result is shown above in (A.50).
- (Induction) Otherwise, $|I| > 2$ and let $\{I_L, I_R\} = \text{Ch}(I, \tau)$. (Induction hypothesis) Suppose that $c(\mathbf{u}|I_L)$ and $c(\mathbf{u}|I_R)$ satisfy (A.50). Then using

$$c(\mathbf{u}|I) = \frac{|I_L|}{|I|} c(\mathbf{u}|I_L) + \frac{|I_R|}{|I|} c(\mathbf{u}|I_R), \quad (\text{A.51})$$

one can obtain (A.50) for cluster I as well.

Observe that for the root cluster S the equation (A.50) simplifies and yields (A.23). Further, using (A.50), we obtain (A.22) for any $I \in \mathcal{V}_\tau(\mathcal{S}) \setminus \{S\}$ with parent $P = \text{Pr}(I, \tau)$ as follows:

$$\begin{aligned} c(\mathbf{u}|I) &= -c(\mathbf{x} - \mathbf{y}|I) + 2\alpha_P(\mathbf{x}, \mathbf{v}_P) \frac{|I^{-\tau}|}{|P|} \frac{s_I(\mathbf{x})}{\|s_I(\mathbf{x})\|} + \underbrace{\sum_{\substack{K \in \text{Anc}(I, \tau) \setminus \{S\} \\ R = \text{Pr}(K, \tau)}} 2\alpha_R(\mathbf{x}, \mathbf{v}_R) \frac{|K^{-\tau}|}{|R|} \frac{s_K(\mathbf{x})}{\|s_K(\mathbf{x})\|}}_{=c(\mathbf{u}|P) + c(\mathbf{x} - \mathbf{y}|P)}, \\ &= -c(\mathbf{x} - \mathbf{y}|I) + 2\alpha_P(\mathbf{x}, \mathbf{v}_P) \frac{|I^{-\tau}|}{|P|} \frac{s_I(\mathbf{x})}{\|s_I(\mathbf{x})\|} + c(\mathbf{u}|P) + c(\mathbf{x} - \mathbf{y}|P), \end{aligned} \quad (\text{A.52})$$

which completes the proof. ■

A.2.12 Proof of Lemma A.3

Proof Since the domain $\mathcal{D}(\mathcal{S}, \mathbf{b})$ of $h_{\mathcal{S}, \mathbf{b}}$ is positive invariant (Proposition A.7), the existence of \mathbf{x}^t for $t \geq 0$ simply follows from the continuity of $h_{\mathcal{S}, \mathbf{b}}$ (Proposition A.2). We now show that for any cluster $I \in \mathcal{V}_\tau(\mathcal{S})$ (A.5) that is visited during the recursive computation of $h_{\mathcal{S}, \mathbf{b}}$ in Table A.2, the centroidal radius $r(\mathbf{x}^t|I)$ is bounded above by a certain value, $R_I(\mathbf{x}^0, \mathbf{y})$, depending only on \mathbf{x}^0 and \mathbf{y} .

Based on the recursive definition of $\mathcal{V}_\tau(\mathcal{S})$ in (A.34), we now provide a proof of the result by structural induction. For any $I \in \mathcal{V}_\tau(\mathcal{S})$,

- (Base Case 1) If $I \in \mathcal{S}$ and $|I| = 1$, then the result simply follows, because $r(\mathbf{x}^t|I) = r_i$ for all $t \geq 0$, where $I = \{i\}$.
- (Base Case 2) If $I \in \mathcal{S}$, $|I| \geq 2$ and $b_I = +1$, then, using Table A.2, one can verify that for any $i \in I$

$$\dot{\mathbf{x}}_i = h_{\mathcal{S}, \mathbf{b}}(\mathbf{x})_i = f_A(\mathbf{x}, \mathbf{u}, I)_i + \mathbf{v}_I = -(\mathbf{x}_i - \mathbf{y}_i) + \mathbf{v}_I, \quad (\text{A.53})$$

for some $\mathbf{u} \in (\mathbb{R}^d)^S$ and $\mathbf{v}_I \in \mathbb{R}^d$, where \mathbf{v}_I represents the accumulated rigid translation due to all ancestors of I in τ .

Accordingly, we obtain for any $i \in I$ that

$$\frac{d}{dt} \|\mathbf{x}_i - \mathbf{c}(\mathbf{x}|I)\|^2 = -2 \|\mathbf{x}_i - \mathbf{c}(\mathbf{x}|I)\|^2 + (\mathbf{x}_i - \mathbf{c}(\mathbf{x}|I))^T (\mathbf{y}_i - \mathbf{c}(\mathbf{y}|I)) , \quad (\text{A.54})$$

from which one can conclude that

$$\|\mathbf{x}_i^t - \mathbf{c}(\mathbf{x}^t|I)\| \leq \max(\|\mathbf{x}_i^0 - \mathbf{c}(\mathbf{x}^0|I)\|, \|\mathbf{y}_i - \mathbf{c}(\mathbf{y}|I)\|) . \quad (\text{A.55})$$

Thus, by definition, it follows that the centroidal radius $r(\mathbf{x}^t|I)$ is bounded above as

$$r(\mathbf{x}^t|I) \leq R_I(\mathbf{x}^0, \mathbf{y}) = \max(r(\mathbf{x}^0|I), r(\mathbf{y}|I)) . \quad (\text{A.56})$$

- (Base Case 3) If $I \in \mathcal{S}$, $|I| \geq 2$ and $b_I = -1$, then, using Table A.2, one can verify that for any $k \in K$ and $K \in \text{Ch}(I, \tau)$,

$$\dot{\mathbf{x}}_k = h_{\mathcal{S}, \mathbf{b}}(\mathbf{x})_k = f_S(\mathbf{x}, \mathbf{u}, I)_k + \mathbf{v}_I = -\mathbf{c}(\mathbf{x} - \mathbf{y}|I) + 2\beta_I(\mathbf{x}) \frac{|K^{-\tau}|}{|I|} \frac{s_K(\mathbf{x})}{\|s_K(\mathbf{x})\|} + \mathbf{v}_I, \quad (\text{A.57})$$

for some $\mathbf{u} \in (\mathbb{R}^d)^S$ and $\mathbf{v}_I \in \mathbb{R}^d$.

Accordingly, we obtain for any $K \in \text{Ch}(I, \tau)$ that

$$\frac{d}{dt} r(\mathbf{x}|K) = 0, \quad \text{and} \quad \frac{d}{dt} \|s_K(\mathbf{x})\|^2 = 2\beta_I(\mathbf{x}) . \quad (\text{A.58})$$

Observe from (2.25) that

$$\|s_K(\mathbf{x})\| \geq 2 \left(\beta + \max_{D \in \text{Ch}(I, \tau)} r(\mathbf{x}|D) \right) \implies \min_{\substack{d \in D \\ D \in \text{Ch}(I, \tau)}} (\eta_{d,D}(\mathbf{x}) - r_d) \geq \beta \implies \beta_I(\mathbf{x}) = 0, \quad (\text{A.59})$$

Thus, since $r(\mathbf{x}^t|K) = r(\mathbf{x}^0|K)$ for all $t \geq 0$, it follows that

$$\|s_K(\mathbf{x}^t)\| \leq \max \left(\|s_K(\mathbf{x}^0)\|, 2 \left(\beta + \max_{D \in \text{Ch}(I, \tau)} r(\mathbf{x}^0|D) \right) \right), \quad (\text{A.60})$$

and, since $r(\mathbf{x}|I) \leq \max_{K \in \text{Ch}(I, \tau)} \|s_K(\mathbf{x})\| + r(\mathbf{x}, K)$ for any $\mathbf{x} \in \mathfrak{S}(\tau)$, we have

$$r(\mathbf{x}^t|K) \leq R_I(\mathbf{x}^0, \mathbf{y}), \quad (\text{A.61})$$

where

$$R_I(\mathbf{x}^0, \mathbf{y}) = \max_{K \in \text{Ch}(I, \tau)} \max \left(\|s_K(\mathbf{x}^0)\|, 2 \left(\beta + r(\mathbf{x}^0|K) \right) \right) + \max_{K \in \text{Ch}(I, \tau)} r(\mathbf{x}^0|K), \quad (\text{A.62})$$

- (Induction) Otherwise, $|I| \geq 2$ and suppose that $r(\mathbf{x}^t|K) \leq R_K(\mathbf{x}^0, \mathbf{y})$ for all $K \in \text{Ch}(I, \tau)$. Then, using Lemma A.2, one can obtain for any $K \in \text{Ch}(I, \tau)$ that

$$\frac{d}{dt} \|s_K(\mathbf{x})\|^2 = -2 \|s_K(\mathbf{x})\|^2 + 2 s_K(\mathbf{x})^T s_K(\mathbf{y}) + 2 \alpha_I(\mathbf{x}, \mathbf{v}_I), \quad (\text{A.63})$$

for some $\mathbf{v}_I \in \mathbb{R}^d$. Now observe from (2.20) that

$$\|s_K(\mathbf{x})\| \geq 2 \left(\beta + \max_{D \in \text{Ch}(I, \tau)} r(\mathbf{x}|D) \right) \implies \min_{\substack{d \in D \\ D \in \text{Ch}(I, \tau)}} (\eta_{d,D}(\mathbf{x}) - r_d) \geq \beta \implies \alpha_I(\mathbf{x}, \mathbf{v}_I) = 0, \quad (\text{A.64})$$

Hence, using (A.63) and (A.64), one can conclude that

$$\|s_K(\mathbf{x}^t)\| \leq \max \left(\|s_K(\mathbf{x}^0)\|, \|s_K(\mathbf{y})\|, 2 \left(\beta + \max_{D \in \text{Ch}(I, \tau)} R_D(\mathbf{x}^0, \mathbf{y}) \right) \right) \quad (\text{A.65})$$

and since $r(\mathbf{x}|I) \leq \max_{K \in \text{Ch}(I, \tau)} \|s_K(\mathbf{x})\|_2 + r(\mathbf{x}, K)$ for any $\mathbf{x} \in \mathfrak{S}(\tau)$, we have

$$r(\mathbf{x}^t|I) \leq R_I(\mathbf{x}^0, \mathbf{y}), \quad (\text{A.66})$$

where

$$R_I(\mathbf{x}^0, \mathbf{y}) = \max_{K \in \text{Ch}(I, \tau)} \max \left(\|s_K(\mathbf{x}^0)\|, \|s_K(\mathbf{y})\|, 2 \left(\beta + R_K(\mathbf{x}^0, \mathbf{y}) \right) \right) + \max_{K \in \text{Ch}(I, \tau)} R_K(\mathbf{x}^0, \mathbf{y}). \quad (\text{A.67})$$

Thus, the result follows with $R(\mathbf{x}^0, \mathbf{y}) = R_I(\mathbf{x}^0, \mathbf{y})$. ■

A.2.13 Proof of Lemma A.6

Proof By definition of $\mathcal{D}(\mathcal{S}, \mathbf{b})$ (A.3), $\mathbf{x}^0 \in \mathcal{D}_A(I)$ for any $I \in \mathcal{S}$ with $b_I = +1$, and one can verify using Table A.2 that for any $i \in I$ and $I \in \mathcal{S}$ with $b_I = +1$,

$$\dot{x}_i = h_{\mathcal{S}, \mathbf{b}}(\mathbf{x})_i = f_A(\mathbf{x}, \mathbf{u}, I)_i + v_I = -(x_i - y_i) + v_I, \quad (\text{A.68})$$

for some $\mathbf{u} \in (\mathbb{R}^n)^{\mathcal{S}}$ and $\mathbf{v}_I \in \mathbb{R}^n$, where \mathbf{v}_I represents the accumulated rigid translation due to ancestors of I in τ .

Accordingly, cases (i)-(iv) in Lemma A.6 can be shown as follows:

(i) Using (2.16) and (A.68), one can verify that for any $i \neq j \in I$,

$$\begin{aligned} \frac{d}{dt} \mathcal{L}_{\vec{y}} \frac{1}{2} \|\mathbf{x}_i - \mathbf{x}_j\|^2 &= -\mathcal{L}_{\vec{y}} \frac{1}{2} \|\mathbf{x}_i - \mathbf{x}_j\|^2 + \underbrace{\|y_i - y_j\|^2}_{> (r_i + r_j)^2, \text{ since } \mathbf{y} \in \mathfrak{S}(\tau)} > -\mathcal{L}_{\vec{y}} \frac{1}{2} \|\mathbf{x}_i - \mathbf{x}_j\|^2 + (r_i + r_j)^2, \quad (\text{A.69}) \\ &> (r_i + r_j)^2, \text{ since } \mathbf{y} \in \mathfrak{S}(\tau) \end{aligned}$$

and so for any $t \geq 0$

$$\begin{aligned} \mathcal{L}_{\vec{y}} \frac{1}{2} \|\mathbf{x}_i^t - \mathbf{x}_j^t\|^2 &\geq e^{-t} \underbrace{\mathcal{L}_{\vec{y}} \frac{1}{2} \|\mathbf{x}_i^0 - \mathbf{x}_j^0\|^2}_{\geq (r_i + r_j)^2, \text{ since } \mathbf{x}^0 \in \mathcal{D}_A(I)} + (1 - e^{-t})(r_i + r_j)^2 \geq (r_i + r_j)^2. \quad (\text{A.70}) \\ &\geq (r_i + r_j)^2, \text{ since } \mathbf{x}^0 \in \mathcal{D}_A(I) \end{aligned}$$

(ii) Similarly, using (2.17) and (A.68), we obtain for any $k \in K, K \in \text{Des}(I, \tau)$,

$$\begin{aligned} \frac{d}{dt} \mathcal{L}_{\vec{y}} (\mathbf{x}_k - s_K(\mathbf{x}))^T s_K(\mathbf{x}) &= -\mathcal{L}_{\vec{y}} (\mathbf{x}_k - s_K(\mathbf{x}))^T s_K(\mathbf{x}) + \underbrace{\eta_{k,K}(\mathbf{y}) \|s_K(\mathbf{y})\|}_{\geq 0, \text{ since } \mathbf{y} \in \mathfrak{S}(\tau)}, \quad (\text{A.71}) \\ &\geq 0, \text{ since } \mathbf{y} \in \mathfrak{S}(\tau) \end{aligned}$$

from which we conclude for any $t \geq 0$ that

$$\begin{aligned} \mathcal{L}_{\vec{y}} (\mathbf{x}_k^t - s_K(\mathbf{x}^t))^T s_K(\mathbf{x}^t) &\geq e^{-t} \underbrace{\mathcal{L}_{\vec{y}} (\mathbf{x}_k^0 - s_K(\mathbf{x}^0))^T s_K(\mathbf{x}^0)}_{\geq 0, \text{ since } \mathbf{x}^0 \in \mathcal{D}_A(I)} \geq 0. \quad (\text{A.72}) \\ &\geq 0, \text{ since } \mathbf{x}^0 \in \mathcal{D}_A(I) \end{aligned}$$

(iii) Now observe from (2.17), (2.23) and (A.68) that for any $k \in K$ and $K \in \text{Des}(I, \tau)$,

$$\begin{aligned} \frac{d}{dt} \eta_{k,K}(\mathbf{x}) &= -\eta_{k,K}(\mathbf{x}) \left(1 + \frac{s_K(\mathbf{x})^T s_K(\mathbf{y})}{\|s_K(\mathbf{x})\|^2} \right) + \underbrace{\frac{\mathcal{L}_{\vec{y}} (\mathbf{x}_k - m_K(\mathbf{x}))^T s_K(\mathbf{x})}{\|s_K(\mathbf{x})\|}}_{\geq 0 \text{ by Lemma A.6.(ii)}}. \quad (\text{A.73}) \\ &\geq 0 \text{ by Lemma A.6.(ii)} \end{aligned}$$

As a result, since $\frac{d}{dt} \eta_{k,K}(\mathbf{x}) \geq 0$ whenever $\eta_{k,K}(\mathbf{x}) = 0$, we have the invariance of local cluster structure, i.e., $\eta_{k,K}(\mathbf{x}^t) \geq 0$ for all $t \geq 0$.

(iv) The relative displacement of any pair of agents, $i \neq j \in I$, satisfies

$$\dot{\mathbf{x}}_i - \dot{\mathbf{x}}_j = -(\mathbf{x}_i - \mathbf{x}_j) + (\mathbf{y}_i - \mathbf{y}_j). \quad (\text{A.74})$$

whose solution for $t \geq 0$ is explicitly given by

$$\mathbf{x}_i^t - \mathbf{x}_j^t = e^{-t} (\mathbf{x}_i^0 - \mathbf{x}_j^0) + (1 - e^{-t}) (\mathbf{y}_i - \mathbf{y}_j). \quad (\text{A.75})$$

Hence, since $\mathbf{x}^0 \in \mathcal{D}_A(I)$ and $\mathbf{y} \in \mathfrak{S}(\tau)$, one can verify the intra-cluster collision avoidance as follows:

$$\begin{aligned} \|\mathbf{x}_i^t - \mathbf{x}_j^t\|^2 &= e^{-2t} \underbrace{\|\mathbf{x}_i^0 - \mathbf{x}_j^0\|^2}_{> (r_i + r_j)^2} + (1 - e^{-t})^2 \underbrace{\|\mathbf{y}_i - \mathbf{y}_j\|^2}_{> (r_i + r_j)^2} + e^{-t} (1 - e^{-t}) \underbrace{\mathcal{L}_{\vec{y}} \|\mathbf{x}_i^0 - \mathbf{x}_j^0\|^2}_{\geq 2(r_i + r_j)^2}, \quad (\text{A.76}) \\ &> (r_i + r_j)^2, \quad (\text{A.77}) \end{aligned}$$

and this completes the proof. ■

A.2.14 Proof of Lemma A.7

Proof For any singleton $I \in \mathcal{S}$, the results simply follow, because a singleton cluster contains no pair of indices and has an empty set of descendants. Otherwise, for any nonsingleton $I \in \mathcal{S}$ with $b_I = -1$, one can obtain from Table A.2 that for any $k \in K$ and $K \in \text{Ch}(I, \tau)$,

$$\dot{\mathbf{x}}_k = h_{\mathcal{S}, \mathbf{b}}(\mathbf{x})_k = f_S(\mathbf{x}, \mathbf{u}, I)_k + \mathbf{v}_I = -c(\mathbf{x} - \mathbf{y}|I) + 2\beta_I(\mathbf{x}) \frac{|K^{-\tau}|}{|I|} \frac{s_K(\mathbf{x})}{\|s_K(\mathbf{x})\|} + \mathbf{v}_I, \quad (\text{A.78a})$$

for some $\mathbf{u} \in (\mathbb{R}^n)^{\mathcal{S}}$ and $\mathbf{v}_I \in \mathbb{R}^n$, where \mathbf{v}_I models the overall rigid translation due to ancestors of I in τ .

Accordingly, using (A.78), we will show the results as follows:

- (i) The preservation of local cluster structure can be observed in two steps. First, since $\dot{\mathbf{x}}_i - \dot{\mathbf{x}}_j = 0$ for any $i \neq j \in K$ and $K \in \text{Ch}(I, \tau)$, we have $\eta_{d,D}(\mathbf{x}^t) = \eta_{d,D}(\mathbf{x}^0) \geq 0$ for all $t \geq 0$ and $d \in D$, $D \in \text{Des}(K, \tau)$ and $K \in \text{Ch}(K, \tau)$. Second, using (2.23) and (2.25), we obtain that for any $k \in K$, $K \in \text{Ch}(I, \tau)$,

$$\frac{d}{dt} \eta_{k,K}(\mathbf{x}) = \beta_I(\mathbf{x}) \geq -(\eta_{k,K}(\mathbf{x}) - r_k - \beta) , \quad (\text{A.79})$$

where $\beta > 0$. Hence, $\frac{d}{dt} \eta_{k,K}(\mathbf{x}) > 0$ whenever $\eta_{k,K}(\mathbf{x}) = 0$, and so $\eta_{k,K}(\mathbf{x}^t) \geq 0$ for any $t \geq 0$.

- (ii) Likewise, we conclude the intra-cluster collision avoidance between individuals in I in two steps. First, we have for any $i \neq j \in K$, $K \in \text{Ch}(I, \tau)$,

$$\dot{\mathbf{x}}_i - \dot{\mathbf{x}}_j = 0 , \quad (\text{A.80})$$

guaranteeing that for all $t \geq 0$,

$$\|\mathbf{x}_i^t - \mathbf{x}_j^t\|^2 = \|\mathbf{x}_i^0 - \mathbf{x}_j^0\|^2 > (r_i + r_j)^2 . \quad (\text{A.81})$$

Second, for any $i \in K$, $j \in I \setminus K$ and $K \in \text{Ch}(I, \tau)$, we have

$$\dot{\mathbf{x}}_i - \dot{\mathbf{x}}_j = 2\beta_I(\mathbf{x}) \frac{s_K(\mathbf{x})}{\|s_K(\mathbf{x})\|} , \quad (\text{A.82})$$

yielding

$$\frac{d}{dt} \|\mathbf{x}_i - \mathbf{x}_j\|^2 = 2 \underbrace{\beta_I(\mathbf{x})}_{\geq 0} \underbrace{(\mathbf{x}_i - \mathbf{x}_j)^T \frac{s_K(\mathbf{x})}{\|s_K(\mathbf{x})\|}}_{=\eta_{i,K}(\mathbf{x}) + \eta_{j,I \setminus K}(\mathbf{x}) \geq 0} \geq 0 , \quad (\text{A.83})$$

and so for $t \geq 0$

$$\|\mathbf{x}_i^t - \mathbf{x}_j^t\|^2 \geq \|\mathbf{x}_i^0 - \mathbf{x}_j^0\|^2 > (r_i + r_j)^2 . \quad (\text{A.84})$$

■

A.2.15 Proof of Lemma A.8

Proof By definition of $\mathcal{D}(\mathcal{S}, \mathbf{b})$ (A.3), for any $I \in \mathcal{V}_\tau(\mathcal{S}) \setminus \mathcal{S}$, we have $\mathbf{x}^0 \in \mathcal{D}_H(I)$ (2.18) and one can verify from Table A.2 that for any $k \in K$ and $K \in \text{Ch}(I, \tau)$,

$$\dot{\mathbf{x}}_k = h_{\mathcal{S}, \mathbf{b}}(\mathbf{x})_k = f_H(\mathbf{x}, \mathbf{u}, I)_k + \mathbf{v}_I = \mathbf{u}_k + 2\alpha_I(\mathbf{x}, \mathbf{u}) \frac{|K^{-\tau}|}{|I|} \frac{s_K(\mathbf{x})}{\|s_K(\mathbf{x})\|} + \mathbf{v}_I, \quad (\text{A.85})$$

for some $\mathbf{u} \in (\mathbb{R}^n)^S$ and $\mathbf{v}_I \in \mathbb{R}^n$. Here, \mathbf{v}_I represents the total rigid translation due to ancestors of I in τ .

With these observations in place, we now achieve claimed results as follows:

- (i) The maintenance of cluster separation (case (i) in Lemma A.8) can be observed, using (2.23) and (A.85), as follows: for any $k \in K$ and $K \in \text{Ch}(I, \tau)$,

$$\frac{d}{dt}\eta_{k,K}(\mathbf{x}) = \mathcal{L}_{\vec{\mathbf{u}}}\eta_{k,K}(\mathbf{x}) + \alpha_I(\mathbf{x}, \mathbf{u}), \quad (\text{A.86})$$

and, since $\mathbf{x}^0 \in \mathcal{D}_H(I)$ and $\alpha_I(\mathbf{x}, \mathbf{u}) \geq -\mathcal{L}_{\vec{\mathbf{u}}}\eta_{k,K}(\mathbf{x})$ whenever $\eta_{k,K}(\mathbf{x}) = r_k + \alpha$, we have $\eta_{k,K}(\mathbf{x}^t) \geq r_k + \alpha$ for all $t \geq 0$.

- (ii) The inter-cluster collision avoidance (case (ii) in Lemma A.8) directly follows from the maintenance of certain cluster separation (case (i) in Lemma A.8), because

$$\eta_{k,K}(\mathbf{x}^t) \geq r_k, \quad \implies \quad \|\mathbf{x}_i^t - \mathbf{x}_j^t\|^2 > (r_i + r_j)^2, \quad (\text{A.87})$$

■

A.2.16 Proof of Lemma A.9

Proof Since $\mathcal{D}_A(I) = \mathfrak{S}(\tau)$ for any singleton cluster $I \in \mathcal{C}(\tau)$, we have from (A.4) that $\mathcal{D}_B(I, -1) = \mathcal{D}_B(I, +1) = \mathfrak{S}(\tau)$ for any singleton cluster $I \in \mathcal{C}(\tau)$. Hence, by definition (A.3), the first part of the result holds.

Likewise, using (A.8), one can observe the second part of the result, because the binary vectors \mathbf{b} and $\hat{\mathbf{b}}$ only differ at a singleton cluster $D \in \mathcal{S}$ where $b_D = -1$ and $b'_D = +1$. ■

A.2.17 Proof of Lemma A.10

Proof For any nonsingleton $I \in \mathcal{S}$ with $b_I = -1$, one can verify from Table A.2 that for any $k \in K$ and $K \in \text{Ch}(I, \tau)$

$$\dot{\mathbf{x}}_k = h_{\mathcal{S}, \mathbf{b}}(\mathbf{x})_k = f_S(\mathbf{x}, \mathbf{u}, I) = -c(\mathbf{x} - \mathbf{y}|I) + 2\beta_I(\mathbf{x}) \frac{|K^{-\tau}|}{|I|} \frac{s_K(\mathbf{x})}{\|s_K(\mathbf{x})\|} + \mathbf{v}_I, \quad (\text{A.88})$$

for some $\mathbf{u} \in (\mathbb{R}^n)^S$ and $\mathbf{v}_I \in \mathbb{R}^n$.

Accordingly, using (2.23) and (2.25), we obtain that

$$\frac{d}{dt}\eta_{k,K}(\mathbf{x}) = \beta_I(\mathbf{x}) \geq -\eta_{k,K}(\mathbf{x}) + r_k + \beta. \quad (\text{A.89})$$

Hence, a trajectory \mathbf{x}^t of $h_{\mathcal{S}, \mathbf{b}}$, starting at any $\mathbf{x}^0 \in \mathcal{D}(\mathcal{S}, \mathbf{b})$, satisfies

$$\eta_{k,K}(\mathbf{x}^t) \geq e^{-t} \eta_{k,K}(\mathbf{x}^0) + (1 - e^{-t})(r_k + \beta), \quad (\text{A.90})$$

for all $t \geq 0$. Thus, since $\beta > \alpha > 0$ and $\frac{d}{dt} \eta_{k,K}(\mathbf{x}) > 0$ whenever $\eta_{k,K}(\mathbf{x}) < r_k + \beta$, using LaSalle's Invariance Principle [138], one can conclude that the local policy $h_{\mathcal{S}, \mathbf{b}}$ asymptotically steers all the configurations in its domain $\mathcal{D}(\mathcal{S}, \mathbf{b})$ to a subset $\mathcal{D}_H^\beta(I)$ of the interior $\mathring{\mathcal{D}}_H(I)$ of $\mathcal{D}_H(I)$ (2.18),

$$\mathcal{D}_H^\beta(I) := \left\{ \mathbf{x} \in \mathfrak{S}(\tau) \mid \eta_{k,K}(\mathbf{x}) \geq r_k + \beta, \quad \forall k \in K, K \in \text{Ch}(I, \tau) \right\} \subset \mathcal{D}_H(I). \quad (\text{A.91})$$

In particular, since $\beta > \alpha$, the system in (A.88) starting at any configuration in $\mathcal{D}(\mathcal{S}, \mathbf{y})$ enters $\mathcal{D}_H(I)$ in finite time.

Now observe from (A.3) and (A.8) that

$$\text{priority}(\mathcal{S}', \mathbf{b}') = \text{priority}(\mathcal{S}, \mathbf{b}) + \underbrace{|I|^2 - \sum_{D \in \text{Ch}(I, \tau)} |D|^2}_{>0} > \text{priority}(\mathcal{S}, \mathbf{b}), \quad (\text{A.92})$$

and

$$\mathcal{D}(\mathcal{S}', \mathbf{b}') = \mathcal{D}(\mathcal{S}, \mathbf{b}) \cap \mathcal{D}_H(I) \supset \mathcal{D}(\mathcal{S}, \mathbf{b}) \cap \mathcal{D}_H^\beta(I). \quad (\text{A.93})$$

Thus, since $\mathcal{D}_H^\beta(I) \subset \mathring{\mathcal{D}}_H(I)$ and its domain $\mathcal{D}(\mathcal{S}, \mathbf{b})$ is positively invariant (Proposition A.7), $h_{\mathcal{S}, \mathbf{b}}$ prepares $h_{\mathcal{S}', \mathbf{b}'}$ in finite time, and the result follows. \blacksquare

A.2.18 Proof of Lemma A.11

Proof Since $\text{Ch}(I, \tau) \subset \mathcal{S}$ and $\mathbf{b}_K = +1$ for any $K \in \text{Ch}(I, \tau)$, every child $K \in \text{Ch}(I, \tau)$ of I in τ satisfies the base condition in cases 2-4 in Table A.2 whereas cluster I satisfies the recursion conditions in cases 9-12 in Table A.2. Hence, using Table A.2, one can verify that for any $k \in K$ and $K \in \text{Ch}(I, \tau)$,

$$\dot{\mathbf{x}}_k = h_{\mathcal{S}, \mathbf{b}}(\mathbf{x})_k = (f_H \circ f_A)(\mathbf{x}, \mathbf{u}, I)_k + \mathbf{v}_I = -(\mathbf{x}_k - \mathbf{y}_k) + 2\alpha_I(\mathbf{x}, \mathbf{u}) \frac{|K^{-\tau}|}{|I|} \frac{s_K(\mathbf{x})}{\|s_K(\mathbf{x})\|} + \mathbf{v}_I, \quad (\text{A.94})$$

for some $\mathbf{u} \in (\mathbb{R}^n)^S$ and $\mathbf{v}_I \in \mathbb{R}^n$.

We now show in three steps that $h_{\mathcal{S}, \mathbf{b}}$ asymptotically steers (almost) all configuration in its domain $\mathcal{D}(\mathcal{S}, \mathbf{b})$ to

$$\mathcal{G}(I) := \left\{ \mathbf{x} \in \mathfrak{S}(\tau) \mid \frac{s_K(\mathbf{x})}{\|s_K(\mathbf{x})\|} = \frac{s_K(\mathbf{y})}{\|s_K(\mathbf{y})\|}, \|s_K(\mathbf{x})\| \geq \|s_K(\mathbf{y})\|, \mathbf{x}_k - \mathbf{c}(\mathbf{x}|K) = \mathbf{y}_k - \mathbf{c}(\mathbf{y}|K), \right. \\ \left. \forall k \in K, K \in \text{Ch}(I, \tau) \right\}, \quad (\text{A.95})$$

which is a subset of $\widehat{\mathcal{D}}_A(I_L, I_R)$ (A.28) associated with children clusters $\{I_L, I_R\} = \text{Ch}(I, \tau)$,

because for any $\mathbf{x} \in \mathcal{G}(I)$ and $i \in K$, $j \in I \setminus K$ and $K \in \text{Ch}(I, \tau)$

$$\mathcal{L}_{\vec{\mathbf{y}}} \frac{1}{2} \|\mathbf{x}_i - \mathbf{x}_j\|^2 = (\mathbf{x}_i - \mathbf{x}_j)^\top (\mathbf{y}_i - \mathbf{y}_j), \quad (\text{A.96})$$

$$= \underbrace{(\mathbf{x}_i - \mathbf{x}_j - \mathbf{s}_K(\mathbf{x}) + \mathbf{s}_K(\mathbf{y}))^\top}_{=\mathbf{y}_i - \mathbf{y}_j} (\mathbf{y}_i - \mathbf{y}_j) + \underbrace{(\mathbf{s}_K(\mathbf{x}) - \mathbf{s}_K(\mathbf{y}))^\top}_{=\epsilon \mathbf{s}_K(\mathbf{y}) \text{ for some } \epsilon \geq 0} (\mathbf{y}_i - \mathbf{y}_j), \quad (\text{A.97})$$

$$\geq 0 \text{ since } \mathbf{y} \in \mathfrak{C}(\tau)$$

$$\geq \|\mathbf{y}_i - \mathbf{y}_j\|^2 > (r_i + r_j)^2, \quad (\text{A.98})$$

and for any $\mathbf{x} \in \mathcal{G}(I)$ and $k \in K$ and $K \in \text{Ch}(I, \tau)$

$$\mathcal{L}_{\vec{\mathbf{y}}} (\mathbf{x}_k - \mathbf{m}_K(\mathbf{x}))^\top \mathbf{s}_K(\mathbf{x}) = \underbrace{(\mathbf{y}_k - \mathbf{m}_K(\mathbf{y}))^\top \mathbf{s}_K(\mathbf{x})}_{\geq 0 \text{ since } \mathbf{x} \in \mathcal{G}(I) \text{ and } \mathbf{y} \in \mathfrak{C}(\tau)} + \underbrace{(\mathbf{x}_k - \mathbf{m}_K(\mathbf{x}))^\top \mathbf{s}_K(\mathbf{y})}_{\geq 0 \text{ since } \mathbf{x} \in \mathcal{G}(I)} \geq 0. \quad (\text{A.99})$$

Likewise, one can observe that $\mathcal{G}(I) \cap \mathcal{D}_H(I)$ is a subset of the interior of $\hat{\mathcal{D}}_A(I_L, I_R)$, i.e., $\mathcal{G}(I) \cap \mathcal{D}_H(I) \subset \hat{\mathcal{D}}_A(I_L, I_R)$.

First, using (A.94), we obtain that for any $K \in \text{Ch}(I, \tau)$

$$\frac{d}{dt} \frac{\mathbf{s}_K(\mathbf{x})^\top \mathbf{s}_K(\mathbf{y})}{\|\mathbf{s}_K(\mathbf{x})\| \|\mathbf{s}_K(\mathbf{y})\|} = \frac{\|\mathbf{s}_K(\mathbf{y})\|}{\|\mathbf{s}_K(\mathbf{x})\|} - \frac{\mathbf{s}_K(\mathbf{x})^\top \mathbf{s}_K(\mathbf{y})}{\|\mathbf{s}_K(\mathbf{x})\|^3 \|\mathbf{s}_K(\mathbf{y})\|} = \frac{\|\mathbf{s}_K(\mathbf{y})\|}{\|\mathbf{s}_K(\mathbf{x})\|} \left(1 - \frac{\mathbf{s}_K(\mathbf{x})^\top \mathbf{s}_K(\mathbf{y})}{\|\mathbf{s}_K(\mathbf{x})\| \|\mathbf{s}_K(\mathbf{y})\|} \right)^2, \quad (\text{A.100})$$

$$\geq 0,$$

where the equality only holds if $\frac{\mathbf{s}_K(\mathbf{x})}{\|\mathbf{s}_K(\mathbf{x})\|} = \pm \frac{\mathbf{s}_K(\mathbf{y})}{\|\mathbf{s}_K(\mathbf{y})\|}$. Thus, $h_{\mathcal{S}, \mathbf{b}}$ asymptotically aligns the separating hyperplane normals of complementary clusters $\text{Ch}(I, \tau)$ of (almost) any configuration in $\mathcal{D}(\mathcal{S}, \mathbf{b})$ with the desired ones. Note that the set of configurations $\mathbf{x} \in \mathcal{D}(\mathcal{S}, \mathbf{b})$ with $\frac{\mathbf{s}_K(\mathbf{x})}{\|\mathbf{s}_K(\mathbf{x})\|} = -\frac{\mathbf{s}_K(\mathbf{y})}{\|\mathbf{s}_K(\mathbf{y})\|}$ has measure zero and are saddle points.

Next, let $\mathbf{x} \in \mathcal{D}(\mathcal{S}, \mathbf{b})$ with $\frac{\mathbf{s}_K(\mathbf{x})}{\|\mathbf{s}_K(\mathbf{x})\|} = \frac{\mathbf{s}_K(\mathbf{y})}{\|\mathbf{s}_K(\mathbf{y})\|}$ for all $K \in \text{Ch}(I, \tau)$. Then, using (A.94), observe that

$$\frac{d}{dt} \|\mathbf{s}_K(\mathbf{x})\|^2 = -2 \|\mathbf{s}_K(\mathbf{x})\|^2 + 2 \mathbf{s}_K(\mathbf{x})^\top \mathbf{s}_K(\mathbf{y}) + 4 \underbrace{\alpha_I(\mathbf{x}, \mathbf{u})}_{\geq 0} \|\mathbf{s}_K(\mathbf{x})\|, \quad (\text{A.101})$$

$$\geq -2 \|\mathbf{s}_K(\mathbf{x})\|^2 + 2 \mathbf{s}_K(\mathbf{x})^\top \mathbf{s}_K(\mathbf{y}) = -2 \|\mathbf{s}_K(\mathbf{x})\| (\|\mathbf{s}_K(\mathbf{x})\| - \|\mathbf{s}_K(\mathbf{y})\|). \quad (\text{A.102})$$

Hence, $\frac{d}{dt} \|\mathbf{s}_K(\mathbf{x})\|^2 > 0$ whenever $\|\mathbf{s}_K(\mathbf{x})\| < \|\mathbf{s}_K(\mathbf{y})\|_2$. Thus, the stable configurations of $h_{\mathcal{S}, \mathbf{b}}$ also satisfies $\|\mathbf{s}_K(\mathbf{x})\| \geq \|\mathbf{s}_K(\mathbf{y})\|$.

Finally, we have from (A.94) that for any $k \in K$, $K \in \text{Ch}(K, \tau)$,

$$\frac{d}{dt} (\mathbf{x}_k - \mathbf{c}(\mathbf{x}|K)) = -(\mathbf{x}_k - \mathbf{c}(\mathbf{x}|K)) + (\mathbf{y}_k - \mathbf{c}(\mathbf{y}|K)), \quad (\text{A.103})$$

and so a trajectory \mathbf{x}^t of $h_{\mathcal{S}, \mathbf{b}}$, starting at any $\mathbf{x}^0 \in \mathcal{D}(\mathcal{S}, \mathbf{b})$, satisfies

$$\mathbf{x}_k^t - \mathbf{c}(\mathbf{x}^t|K) = e^{-t} (\mathbf{x}_k^0 - \mathbf{c}(\mathbf{x}^0|K)) + (1 - e^{-t}) (\mathbf{y}_k - \mathbf{c}(\mathbf{y}|K)), \quad (\text{A.104})$$

for all $t \geq 0$. Hence, the centroidal displacements, $\mathbf{x}_k^t - \mathbf{c}(\mathbf{x}^t|K)$, of any configuration $\mathbf{x} \in \mathcal{D}(\mathcal{S}, \mathbf{b})$ asymptotically matches the centroidal displacement, $\mathbf{y}_k - \mathbf{c}(\mathbf{y}|K)$, of the desired configuration \mathbf{y} .

Thus, it follows from LaSalle's Invariance Principle [138] that (almost) all configurations in the domain $\mathcal{D}(\mathcal{S}, \mathbf{b})$ of $h_{\mathcal{S}, \mathbf{b}}$ asymptotically reach $\mathcal{G}(I)$.

Now observe from (A.3) and (A.8) that

$$\text{priority}(\mathcal{S}', \mathbf{b}') = \text{priority}(\mathcal{S}, \mathbf{b}) + \underbrace{|I|^2 - \sum_{D \in \text{Ch}(I, \tau)} |D|^2}_{> 0} > \text{priority}(\mathcal{S}, \mathbf{b}), \quad (\text{A.105})$$

and

$$\mathcal{D}(\mathcal{S}', \mathbf{b}') \supset \mathcal{D}(\mathcal{S}, \mathbf{b}) \cap \widehat{\mathcal{D}}_A(I_L, I_R) \supset \mathcal{D}(\mathcal{S}, \mathbf{b}) \cap \mathcal{G}(I), \quad (\text{A.106})$$

which follows from that $\mathcal{G}(I) \subset \widehat{\mathcal{D}}_A(I_L, I_R)$ and $\mathcal{D}_A(I) = \mathcal{D}_A(I_L) \cap \mathcal{D}_A(I_R) \cap \widehat{\mathcal{D}}_A(I_L, I_R)$ (A.27).

Thus, one can conclude from (A.106) and $\mathcal{G}(I) \cap \mathcal{D}_H(I) \subset \overset{\circ}{\widehat{\mathcal{D}}}_A(I_L, I_R)$ that the disks starting at almost any configuration in the positively invariant $\mathcal{D}(\mathcal{S}, \mathbf{b})$ reach $\mathcal{D}(\mathcal{S}', \mathbf{b}')$ in finite time, and this completes the proof. \blacksquare

Appendix B

On the Optimality of Napoleon Triangles

In elementary geometry, one way of constructing an equilateral triangle from any given triangle is as follows: in a plane the centres of equilateral triangles erected, either all externally or all internally, on the sides of the given triangle form an equilateral triangle, illustrated in Figure B.1 (also see [62, Chapter 3.3] and the survey [155]). This result is generally referred to as *Napoleon's theorem*, notwithstanding its dubious origins — see [155] and [95] for a detailed history of the theorem. We will refer to these constructions as *the outer and inner Napoleon transformations*, and the associated equilateral triangles as *the outer and inner Napoleon triangles* of the original triangle, respectively. Conversely, given its outer and inner Napoleon triangles in position (i.e., they are oppositely oriented and have the same centroid), the original triangle is uniquely determined; for this and related results we refer to [220] and [98]. In other words, the converse of Napoleon's theorem offers a parametrization of a triangle in terms of equilateral triangles. A fascinating application of Napoleon triangles is the planar tessellation used by Escher: a plane can be tiled using congruent copies of the hexagon, defined by the vertices of any triangle and its uniquely paired outer Napoleon triangle, known as *Escher's theorem* [182].

Equilaterals built on the sides of a triangle make a variety of appearances in the classical literature. Torricelli uses this construction to solve Fermat's problem: locate a point minimizing the sum of distances to the vertices of a given triangle — one of the first problems of location science [136]. The unique solution of this problem is known as *the Torricelli point* of the given triangle, located as follows [96]. If an internal angle of the triangle is greater than 120° , then the Torricelli point is at that obtuse vertex. Otherwise, the three lines joining opposite vertices of the original triangle and externally erected triangles are concurrent, and they intersect at the Torricelli point, see Figure B.1. The figure, defined by the original triangle and the erected equilateral triangles, is referred to as *the Torricelli configuration* (see [156, 99]), and the new vertices of this figure form the so-called *vertex set of the Torricelli configuration*. It also bears mentioning that explicit solutions in nonlinear optimization are very rare. The Fermat problem for three points is such a special case, and its generalization to more points has no explicit solution [136].

In this appendix, we demonstrate some remarkable, but not immediately obvious, optimality properties of twice iterated Napoleon triangles. First, two composed inner Napoleon

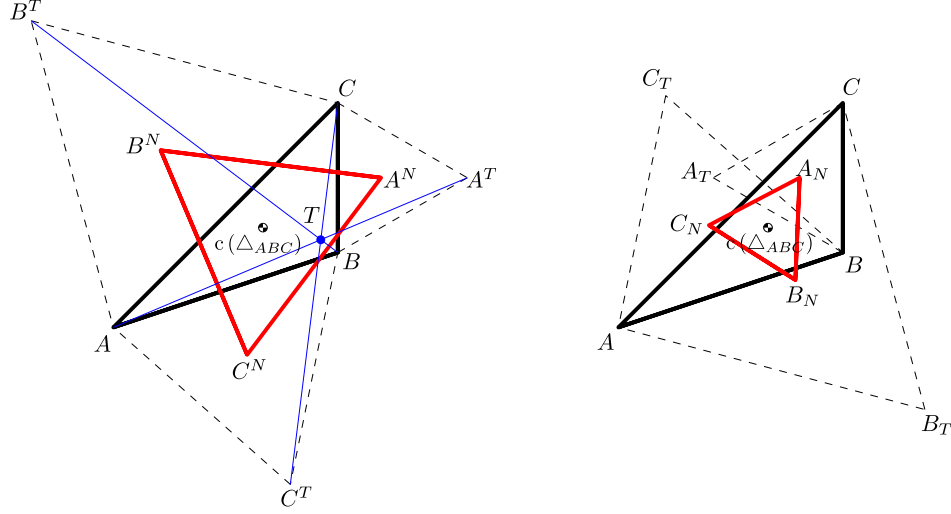


Figure B.1: An illustration of (left) the Torricelli point T , the outer Torricelli configuration with $\triangle_{A^T B^T C^T}$ and the outer Napoleon triangle $\triangle_{A^N B^N C^N}$, and (right) the inner Torricelli configuration with $\triangle_{A_T B_T C_T}$ and the inner Napoleon triangle $\triangle_{A_N B_N C_N}$ of a triangle \triangle_{ABC} . Note that centroids of the vertices of Torricelli configurations, Napoleon triangles and the original triangle all coincide, i.e., $c(\triangle_{ABC}) = c(\triangle_{A^T B^T C^T}) = c(\triangle_{A^N B^N C^N}) = c(\triangle_{A_T B_T C_T}) = c(\triangle_{A_N B_N C_N})$.

transformations of a triangle collapse the original one to a point located at its centroid which, by definition, minimizes the sum of squared distances to the vertices of the given triangle. Surprisingly, two composed outer Napoleon transformations yield an equilateral triangle, optimally aligned with the original triangle by virtue of minimizing the sum of squared distances between the paired vertices (Theorem B.1). More precisely, for any triangle \triangle_{ABC} in the n -dimensional Euclidean space \mathbb{R}^n with the vertices $A, B, C \in \mathbb{R}^n$, we will say that the triangle $\triangle_{A'B'C'}$ is an *optimally aligned* equilateral triangle of \triangle_{ABC} if it solves the following constrained optimization problem:

$$\begin{aligned} & \text{minimize} && \|A - A'\|^2 + \|B - B'\|^2 + \|C - C'\|^2 \\ & \text{subject to} && \|A' - B'\|^2 = \|A' - C'\|^2 = \|B' - C'\|^2 \end{aligned} \tag{B.1}$$

where $A', B', C' \in \mathbb{R}^n$ and $\|\cdot\|$ denotes the standard Euclidean norm on \mathbb{R}^n . As we show below, this optimization problem has a unique solution so long as A, B, C are not collinear. It is also important to emphasize that this is another rare instance of a nonlinear optimization problem that admits an explicit solution.

B.1 Torricelli and Napoleon Transformations

For any ordered triple $\mathbf{x} = [x_1, x_2, x_3]^T \in \mathbb{R}^{3n}$ of vectors of the n -dimensional Euclidean space \mathbb{R}^n , let $\mathbf{R}_{\mathbf{x}}$, denote the *rotation matrix* corresponding to a counter-clockwise rotation by $\pi/2$ in the plane, defined by orthonormal vectors \mathbf{n} and \mathbf{t} , in which the triangle $\triangle_{\mathbf{x}}$

formed by \mathbf{x} is positively oriented (i.e., its vertices in counter-clockwise order follow the sequence $\dots \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 1 \rightarrow \dots$),

$$\mathbf{R}_{\mathbf{x}} := [\mathbf{n}, \mathbf{t}] \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} [\mathbf{n}, \mathbf{t}]^T, \quad (\text{B.2})$$

where

$$\mathbf{n} := \begin{cases} \frac{\mathbf{x}_2 - \mathbf{x}_1}{\|\mathbf{x}_2 - \mathbf{x}_1\|}, & \text{if } \mathbf{x}_1 \neq \mathbf{x}_2, \\ \frac{\mathbf{x}_3 - \mathbf{x}_2}{\|\mathbf{x}_3 - \mathbf{x}_2\|}, & \text{otherwise,} \end{cases} \quad \mathbf{t} := \begin{cases} \in \{z \in \mathbb{S}^{n-1} \mid \mathbf{n}^T z = 0\}, & \text{if } \mathbf{x} \text{ is collinear,} \\ \mathbf{P}(\mathbf{n}) \frac{\mathbf{x}_3 - \mathbf{x}_1}{\|\mathbf{x}_3 - \mathbf{x}_1\|}, & \text{otherwise.} \end{cases} \quad (\text{B.3})$$

Here, $\mathbf{P}(\mathbf{n}) := \mathbf{I}_n - \mathbf{n}\mathbf{n}^T$ is the projection onto $T_{\mathbf{n}}\mathbb{S}^{n-1}$ (the tangent space of the $(n-1)$ -sphere \mathbb{S}^{n-1} at point $\mathbf{n} \in \mathbb{S}^{n-1}$), and \mathbf{I}_n is the $n \times n$ identity matrix, and \mathbf{A}^T denotes the transpose of a matrix \mathbf{A} . For any trivial triangle $\Delta_{\mathbf{x}}$, all of whose vertices are located at the same point, we fix $\mathbf{R}_{\mathbf{x}} = \mathbf{0}$ by setting $\frac{\mathbf{x}}{\|\mathbf{x}\|} = 0$ whenever $\mathbf{x} = 0$. Note that $\Delta_{\mathbf{x}}$ is both positively and negatively oriented if \mathbf{x} is collinear. Consequently, to define a plane containing such \mathbf{x} , we select an arbitrary vector \mathbf{t} perpendicular to vector \mathbf{n} in (B.3). It is also convenient to denote by $c(\mathbf{x})$ the *centroid* of $\Delta_{\mathbf{x}}$, i.e., $c(\mathbf{x}) := \frac{1}{3} \sum_{i=1}^3 \mathbf{x}_i$.

In general, the Torricelli and Napoleon transformations of three points in Euclidean n -space can be defined based on their original planar definitions in a 2-dimensional subspace of \mathbb{R}^n containing \mathbf{x} . That is to say, for any $\mathbf{x} \in \mathbb{R}^{3n}$, select a 2-dimensional subspace of \mathbb{R}^n containing \mathbf{x} , and then construct the erected triangles on the side of $\Delta_{\mathbf{x}}$ in this subspace to obtain the Torricelli and Napoleon transformations of \mathbf{x} , as illustrated in Figure B.1. Accordingly, let $T_{\pm} : \mathbb{R}^{3n} \rightarrow \mathbb{R}^{3n}$ and $N_{\pm} : \mathbb{R}^{3n} \rightarrow \mathbb{R}^{3n}$ denote the *Torricelli and Napoleon transformations* where the sign, $+$ and $-$, determines the type of the transformation, inner and outer, respectively. Denoting by \otimes the Kronecker product [94], one can write closed form expressions of the Torricelli and Napoleon transformations as follows.

Lemma B.1 *The Torricelli and Napoleon transformations of any triple $\mathbf{x} \in \mathbb{R}^{3n}$ on a plane containing \mathbf{x} are, respectively, given by*

$$T_{\pm}(\mathbf{x}) = \left(\frac{1}{2} \mathbf{K} \pm \frac{\sqrt{3}}{2} (\mathbf{I}_3 \otimes \mathbf{R}_{\mathbf{x}}) \mathbf{L} \right) \mathbf{x}, \quad (\text{B.4})$$

$$N_{\pm}(\mathbf{x}) = \frac{1}{3} (\mathbf{K}\mathbf{x} + T_{\pm}(\mathbf{x})), \quad (\text{B.5})$$

where

$$\mathbf{K} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \otimes \mathbf{I}_n, \quad \text{and} \quad \mathbf{L} = \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix} \otimes \mathbf{I}_n. \quad (\text{B.6})$$

Proof One can locate the new vertex of an equilateral triangle, inwardly or outwardly, constructed on one side of $\Delta_{\mathbf{x}}$ in the plane containing \mathbf{x} using different geometric properties of equilateral triangles. We find it convenient to use the perpendicular bisector of the corresponding side of $\Delta_{\mathbf{x}}$, the line passing through its midpoint and being perpendicular to it, such that the new vertex is on this bisector and at a proper distance away from the side of $\Delta_{\mathbf{x}}$.

For instance, let $\mathbf{y} = [y_1, y_2, y_3]^T = T_+(\mathbf{x})$. Consider the side of $\Delta_{\mathbf{x}}$ joining x_1 and x_2 , using the midpoint $m_{12} := \frac{1}{2}(x_1 + x_2)$, to locate the new vertex y_3 of inwardly erected triangle on this side as

$$y_3 = m_{12} + \frac{\sqrt{3}}{2} \mathbf{R}_{\mathbf{x}} (x_2 - x_1) , \quad (\text{B.7})$$

where $\mathbf{R}_{\mathbf{x}}$ (see (B.2)) is a counter-clockwise rotation by $\frac{\pi}{2}$ in the plane where \mathbf{x} is positively oriented. Note that the height of an equilateral triangle from any side is $\frac{\sqrt{3}}{2}$ times its side length. Hence, by symmetry, one can conclude (B.4).

Given a Torricelli configuration $\mathbf{y} = [y_1, y_2, y_3]^T = T_{\pm}(\mathbf{x})$, by definition, the vertices of the associated Napoleon triangle $\mathbf{z} = [z_1, z_2, z_3]^T = N_{\pm}(\mathbf{x})$ are given by

$$z_1 = \frac{1}{3}(y_1 + x_2 + x_3) , \quad z_2 = \frac{1}{3}(x_1 + y_2 + x_3) \quad \text{and} \quad z_3 = \frac{1}{3}(x_1 + x_2 + y_3) , \quad (\text{B.8})$$

which is equal to (B.5), and so the result follows. \blacksquare

Note that the Torricelli and Napoleon transformations of \mathbf{x} are unique if and only if $\mathbf{x} \in \mathbb{R}^{3n}$ is non-collinear. If, contrarily, \mathbf{x} is collinear, then $\Delta_{\mathbf{x}}$ is both positively and negatively oriented, and for $n \geq 3$ there is more than one 2-dimensional subspace of \mathbb{R}^n containing \mathbf{x} .

Remark B.1. ([220]) For any $\mathbf{x} = [x_1, x_2, x_3]^T \in \mathbb{R}^{3n}$, the centroid of the Torricelli configuration $\mathbf{y} = [y_1, y_2, y_3]^T = T_{\pm}(\mathbf{x})$, the Napoleon configuration $\mathbf{z} = N_{\pm}(\mathbf{x})$ and the original triple \mathbf{x} all coincide, i.e.,

$$c(\mathbf{x}) = c(\mathbf{y}) = c(\mathbf{z}) , \quad (\text{B.9})$$

and the distances between the associated elements of \mathbf{x} and \mathbf{y} are all the same, i.e., for any $i \neq j \in \{1, 2, 3\}$

$$\|y_i - x_i\|_2 = \|y_j - x_j\|_2 . \quad (\text{B.10})$$

An observation key to all further results is that Napoleon transformations of equilateral triangles are very simple.

Lemma B.2 *The inner Napoleon transformation N_+ of any triple $\mathbf{x} = [x_1, x_2, x_3]^T \in \mathbb{R}^{3n}$ comprising the vertices of an equilateral triangle $\Delta_{\mathbf{x}}$ collapses it to the trivial triangle all of whose vertices are located at its centroid $c(\mathbf{x})$,*

$$N_+(\mathbf{x}) = \mathbf{1}_3 \otimes c(\mathbf{x}) , \quad (\text{B.11})$$

whereas the outer Napoleon transformation N_- reflects the vertices of $\Delta_{\mathbf{x}}$ with respect to its centroid $c(\mathbf{x})$,

$$N_-(\mathbf{x}) = 2 \cdot \mathbf{1}_3 \otimes c(\mathbf{x}) - \mathbf{x} . \quad (\text{B.12})$$

Here, $\mathbf{1}_3$ is the \mathbb{R}^3 column vector of all ones, and \cdot denotes the standard entrywise (or Hadamard [112, Section 5.7]) product.

Proof Observe that the inwardly erected triangle on any side of an equilateral triangle is equal to the equilateral triangle itself, i.e., $T_+(\mathbf{x}) = \mathbf{x}$, and so, by definition, one has (B.11).

Alternatively, using (B.5), one can obtain

$$N_+(\mathbf{x}) = \frac{1}{3}(\mathbf{K}\mathbf{x} + T_+(\mathbf{x})) = \frac{1}{3}(\mathbf{K}\mathbf{x} + \mathbf{x}) = \mathbf{1}_3 \otimes c(\mathbf{x}) , \quad (\text{B.13})$$

where \mathbf{K} is defined as in (B.6).

Now consider outwardly erected equilateral triangles on the sides of an equilateral triangle, and let $\mathbf{y} = [y_1, y_2, y_3]^T = T_-(\mathbf{x})$. Note that each erected triangle has a common side with the original triangle. Since $\triangle_{\mathbf{x}}$ is equilateral, observe that the midpoint of the unshared vertices of an erected triangle and the original triangle is equal to the midpoint of their common sides, i.e., $\frac{1}{2}(y_1 + x_1) = \frac{1}{2}(x_2 + x_3)$ and so on. Hence, we have $T_-(\mathbf{x}) = \mathbf{K}\mathbf{x} - \mathbf{x}$. Thus, one can verify the result using (B.5) as

$$N_-(\mathbf{x}) = \frac{1}{3}(\mathbf{K}\mathbf{x} + T_-(\mathbf{x})) = \frac{1}{3}(\mathbf{K}\mathbf{x} + \mathbf{K}\mathbf{x} - \mathbf{x}) = 2 \cdot \mathbf{1}_3 \otimes c(\mathbf{x}) - \mathbf{x} . \quad (\text{B.14})$$

■

Since the Napoleon transformation of any triangle results in an equilateral triangle, motivated from Lemma B.2, we now consider the iterations of the Napoleon transformation. For any $k \geq 0$, let $N_{\pm}^k : \mathbb{R}^{3n} \rightarrow \mathbb{R}^{3n}$ denote the k -th Napoleon transformation defined to be

$$N_{\pm}^{k+1} := N_{\pm} \circ N_{\pm}^k , \quad (\text{B.15})$$

where we set $N_{\pm}^0 := id$, and $id : \mathbb{R}^{3n} \rightarrow \mathbb{R}^{3n}$ is the *identity map* on \mathbb{R}^{3n} .

It is evident from Lemma B.2 that the following lemma holds.

Lemma B.3 *For any $\mathbf{x} \in \mathbb{R}^{3n}$ and $k \geq 1$,*

$$N_+^{k+1}(\mathbf{x}) = \mathbf{1}_3 \otimes c(\mathbf{x}) , \quad \text{and} \quad N_-^{k+2}(\mathbf{x}) = N_-^k(\mathbf{x}) . \quad (\text{B.16})$$

As a result, the basis of iterations of the Napoleon transformations consists of N_{\pm} and N_{\pm}^2 , whose explicit forms, except N_-^2 , are given above. Using (B.5) and (B.12), the closed form expression of the double outer Napoleon transformation N_-^2 can be obtained as

Lemma B.4 *An arbitrary triple $\mathbf{x} = [x_1, x_2, x_3]^T \in \mathbb{R}^{3n}$ gives rise to the double outer Napoleon triangle, $N_-^2 : \mathbb{R}^{3n} \rightarrow \mathbb{R}^{3n}$, according to the formula*

$$N_-^2(\mathbf{x}) = \frac{2}{3}\mathbf{x} + \frac{1}{3}T_+(\mathbf{x}) . \quad (\text{B.17})$$

Proof By Napoleon's theorem, $N_-(\mathbf{x})$ is an equilateral triangle. Using (B.5) and Lemma B.2, one can obtain the result as follows:

$$N_-^2(\mathbf{x}) = N_-(N_-(\mathbf{x})) = 2 \cdot \mathbf{1}_3 \otimes c(\mathbf{x}) - N_-(\mathbf{x}) = 2 \cdot \mathbf{1}_3 \otimes c(\mathbf{x}) - \frac{1}{3}(\mathbf{K}\mathbf{x} + T_-(\mathbf{x})) , \quad (\text{B.18})$$

$$= \frac{2}{3}(\mathbf{K}\mathbf{x} + \mathbf{x}) - \frac{1}{3}(\mathbf{K}\mathbf{x} + T_-(\mathbf{x})) = \frac{2}{3}\mathbf{x} + \frac{1}{3}(\mathbf{K}\mathbf{x} - T_-(\mathbf{x})) = \frac{2}{3}\mathbf{x} + \frac{1}{3}T_+(\mathbf{x}) , \quad (\text{B.19})$$

where \mathbf{K} is defined as in (B.6).

■

Note that $N_-^2(\mathbf{x})$ is a convex combination of \mathbf{x} and $T_+(\mathbf{x})$, see Figure B.2.

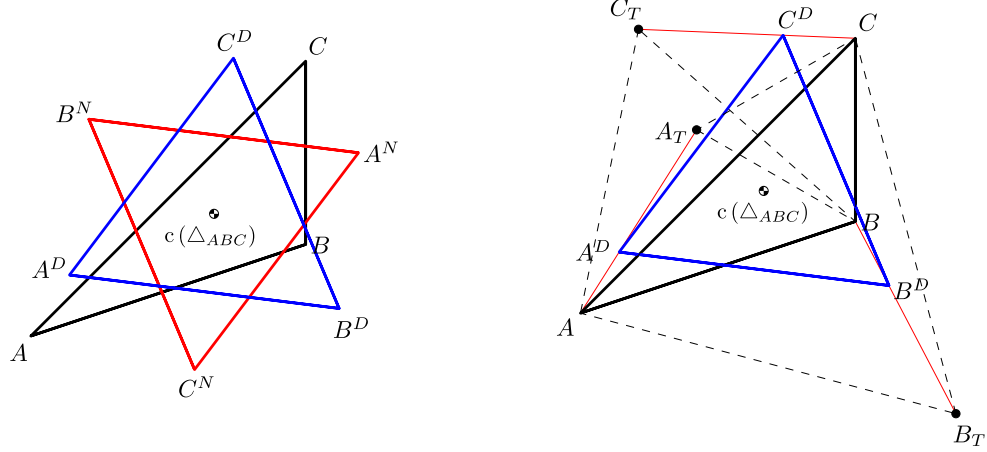


Figure B.2: (left) Outer, $\triangle A^N B^N C^N$, and double outer, $\triangle A^D B^D C^D$, Napoleon transformations of a triangle $\triangle ABC$. (right) The double outer Napoleon triangle $\triangle A^D B^D C^D$ is a convex combination of the original triangle $\triangle ABC$ and the vertex set of its inner Torricelli configuration $\triangle A_T B_T C_T$.

B.2 Optimality of Napoleon Transformations

To best of our knowledge, the Napoleon transformation N_{\pm} is mostly recognized as being a function into the space of equilateral triangles. In addition to this inherited property, N_{\pm}^2 has an optimality property that is not immediately obvious. Although the double inner Napoleon transformation N_+^2 is not really that interesting to work with, it gives a hint about the optimality of N_-^2 : for any given triangle N_+^2 yields a trivial triangle, all of whose vertices are located at the centroid of the given triangle which, by definition, minimizes the sum of squared distances to the vertices of the original triangle. Surprisingly, one has a similar optimality property for N_-^2 :

Theorem B.1 *The double outer Napoleon transformation $N_-^2(\mathbf{x})$ given in (B.17) yields the equilateral triangle most closely aligned with $\triangle_{\mathbf{x}}$ in the sense that it minimizes the total sum of squared distances between corresponding vertices. That is to say, for any $\mathbf{x} = [x_1, x_2, x_3]^T \in \mathbb{R}^{3n}$, $N_-^2(\mathbf{x})$ is an optimal solution of the following problem:*

$$\begin{aligned} & \text{minimize} \quad \sum_{i=1}^3 \|\mathbf{x}_i - \mathbf{y}_i\|^2 \\ & \text{subject to} \quad \|\mathbf{y}_1 - \mathbf{y}_2\|^2 = \|\mathbf{y}_1 - \mathbf{y}_3\|^2 = \|\mathbf{y}_2 - \mathbf{y}_3\|^2, \end{aligned} \tag{B.20}$$

where $\mathbf{y} = [y_1, y_2, y_3]^T \in \mathbb{R}^{3n}$. Furthermore, if \mathbf{x} is non-collinear, then (B.20) has a unique solution.

Proof Using the method of Lagrange multipliers [32], we first show that an optimal solution of (B.20) lies in the plane containing the triangle $\triangle_{\mathbf{x}}$. Then, to show the result, we solve (B.20) using a proper parametrization of equilateral triangles in \mathbb{R}^2 .

The Lagrangian formulation of (B.20) minimizes

$$L(\mathbf{y}, \lambda_1, \lambda_2) = \sum_{i=1}^3 \|\mathbf{x}_i - \mathbf{y}_i\|_2^2 + \lambda_1 \left(\|\mathbf{y}_1 - \mathbf{y}_2\|_2^2 - \|\mathbf{y}_1 - \mathbf{y}_3\|_2^2 \right) + \lambda_2 \left(\|\mathbf{y}_1 - \mathbf{y}_2\|_2^2 - \|\mathbf{y}_2 - \mathbf{y}_3\|_2^2 \right), \quad (\text{B.21})$$

where $\lambda_1, \lambda_2 \in \mathbb{R}$ are Lagrange multipliers. A necessary condition for optimality in (B.20) is that the gradient $\nabla_{\mathbf{y}} L(\mathbf{y}, \lambda_1, \lambda_2)$ of the Lagrangian with respect to \mathbf{y} at any locally optimal solution is zero,

$$\nabla_{\mathbf{y}} L(\mathbf{y}, \lambda_1, \lambda_2) = 2 \begin{bmatrix} (\mathbf{y}_1 - \mathbf{x}_1) + \lambda_1 (\mathbf{y}_3 - \mathbf{y}_2) + \lambda_2 (\mathbf{y}_1 - \mathbf{y}_2) \\ (\mathbf{y}_2 - \mathbf{x}_2) + \lambda_1 (\mathbf{y}_2 - \mathbf{y}_1) + \lambda_2 (\mathbf{y}_3 - \mathbf{y}_1) \\ (\mathbf{y}_3 - \mathbf{x}_3) - \lambda_1 (\mathbf{y}_3 - \mathbf{y}_1) - \lambda_2 (\mathbf{y}_3 - \mathbf{y}_2) \end{bmatrix} = 0, \quad (\text{B.22})$$

from which one can conclude that an optimal solution of (B.20) lies in the plane containing $\triangle_{\mathbf{x}}$. Accordingly, without any loss of generality, suppose that $\triangle_{\mathbf{x}}$ is a positively oriented triangle in \mathbb{R}^2 , i.e., its vertices are in counter-clockwise order in \mathbb{R}^2 .

In general, an equilateral triangle $\triangle_{\mathbf{y}}$ in \mathbb{R}^2 with vertices $\mathbf{y} = [y_1, y_2, y_3]^T \in \mathbb{R}^6$ can be uniquely parametrized using two of its vertices, say y_1 and y_2 , and a binary variable $k \in \{-1, +1\}$ specifying the orientation of $\triangle_{\mathbf{y}}$; for instance, $k = +1$ if $\triangle_{\mathbf{y}}$ is positively oriented, and so on. Consequently, the remaining vertex, y_3 , can be located as

$$y_3 = \frac{1}{2} (y_1 + y_2) + k \frac{\sqrt{3}}{2} \mathbf{R}_{\pi/2} (y_2 - y_1), \quad (\text{B.23})$$

where $\mathbf{R}_{\pi/2} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ is the rotation matrix defining a rotation by $\pi/2$.

Hence, one can rewrite the optimization problem (B.20) in terms of new parameters as an unconstrained optimization problem: for $y_1, y_2 \in \mathbb{R}^2$ and $k \in \{-1, 1\}$,

$$\text{minimize} \quad \|\mathbf{x}_1 - y_1\|_2^2 + \|\mathbf{x}_2 - y_2\|_2^2 + \|\mathbf{x}_3 - \mathbf{M}y_1 - \mathbf{M}^T y_2\|_2^2, \quad (\text{B.24})$$

where $\mathbf{M} := \frac{1}{2}\mathbf{I} - k \frac{\sqrt{3}}{2} \mathbf{R}_{\pi/2}$, and \mathbf{I} is the 2×2 identity matrix. Note that $\mathbf{M} + \mathbf{M}^T = \mathbf{I}$, $\mathbf{M}^T \mathbf{M} = \mathbf{M} \mathbf{M}^T = \mathbf{I}$ and $\mathbf{M}^2 = -\mathbf{M}^T$.

For a fixed $k \in \{-1, 1\}$, (B.24) is a convex optimization problem of y_1 and y_2 , because every norm on \mathbb{R}^n is convex, and compositions of convex functions with affine transformations preserve convexity [37]. Hence, a global optimal solution of (B.24) occurs where the gradient of the objective function is zero at

$$\begin{bmatrix} (\mathbf{I} + \mathbf{M}^T \mathbf{M}) & (\mathbf{M}^2)^T \\ \mathbf{M}^2 & (\mathbf{I} + \mathbf{M} \mathbf{M}^T) \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1 + \mathbf{M}^T \mathbf{x}_3 \\ \mathbf{x}_2 + \mathbf{M} \mathbf{x}_3 \end{bmatrix}, \quad (\text{B.25})$$

which simplifies to

$$\begin{bmatrix} 2\mathbf{I} & -\mathbf{M} \\ -\mathbf{M}^T & 2\mathbf{I} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1 + \mathbf{M}^T \mathbf{x}_3 \\ \mathbf{x}_2 + \mathbf{M} \mathbf{x}_3 \end{bmatrix}. \quad (\text{B.26})$$

Note that the objective function, $f(\mathbf{y})$, is strongly convex, because its Hessian, $\nabla^2 f(\mathbf{y})$, satisfies

$$\nabla^2 f(\mathbf{y}) = \begin{bmatrix} 2\mathbf{I} & -\mathbf{M} \\ -\mathbf{M}^T & 2\mathbf{I} \end{bmatrix} \succeq \mathbf{I}, \quad (\text{B.27})$$

which means that for a fixed $k \in \{-1, +1\}$ the optimal solution of (B.24) is unique.

Now observe that

$$\frac{1}{3} \begin{bmatrix} 2\mathbf{I} & \mathbf{M} \\ \mathbf{M}^T & 2\mathbf{I} \end{bmatrix} \begin{bmatrix} 2\mathbf{I} & -\mathbf{M} \\ -\mathbf{M}^T & 2\mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad (\text{B.28})$$

hence the solution of the linear equation (B.26) is

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 2\mathbf{I} & \mathbf{M} \\ \mathbf{M}^T & 2\mathbf{I} \end{bmatrix} \begin{bmatrix} x_1 + \mathbf{M}^T x_3 \\ x_2 + \mathbf{M} x_3 \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 2x_1 + 2\mathbf{M}^T x_3 + \mathbf{M} x_2 + \mathbf{M}^2 x_3 \\ \mathbf{M}^T x_1 + (\mathbf{M}^2)^T x_3 + 2x_2 + 2\mathbf{M} x_3 \end{bmatrix}, \quad (\text{B.29})$$

$$= \frac{1}{3} \begin{bmatrix} 2x_1 + \mathbf{M}^T x_3 + \mathbf{M} x_2 \\ 2x_2 + \mathbf{M}^T x_1 + \mathbf{M} x_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(x_1 + \frac{x_1 + x_2 + x_3}{3}) + k \frac{1}{2\sqrt{3}} \mathbf{R}_{\pi/2}(x_3 - x_2) \\ \frac{1}{2}(x_2 + \frac{x_1 + x_2 + x_3}{3}) + k \frac{1}{2\sqrt{3}} \mathbf{R}_{\pi/2}(x_1 - x_3) \end{bmatrix}. \quad (\text{B.30})$$

Here, substituting y_1 and y_2 back into (B.23) yields

$$y_3 = \frac{1}{2} \left(x_3 + \frac{x_1 + x_2 + x_3}{3} \right) + k \frac{1}{2\sqrt{3}} \mathbf{R}_{\pi/2}(x_2 - x_1). \quad (\text{B.31})$$

Thus, overall, we have

$$\mathbf{y} = \frac{2}{3} \mathbf{x} + \frac{1}{3} \left(\frac{1}{2} \mathbf{K} \mathbf{x} + k \frac{\sqrt{3}}{2} (\mathbf{I}_3 \otimes \mathbf{R}_{\pi/2}) \mathbf{L} \mathbf{x} \right) = \begin{cases} \frac{2}{3} \mathbf{x} + \frac{1}{3} \mathbf{T}_+(\mathbf{x}), & \text{if } k = +1, \\ \frac{2}{3} \mathbf{x} + \frac{1}{3} \mathbf{T}_-(\mathbf{x}), & \text{if } k = -1, \end{cases} \quad (\text{B.32})$$

where \mathbf{K} and \mathbf{L} are defined as in (B.6). Recall that $\Delta_{\mathbf{x}}$ is assumed to be positively oriented, i.e., $\mathbf{R}_{\mathbf{x}} = \mathbf{R}_{\pi/2}$, and so it is convenient to have the results in terms of Torricelli transformations \mathbf{T}_{\pm} , see (B.4). As a result, the difference of \mathbf{y} and \mathbf{x} is simply given by

$$\mathbf{y} - \mathbf{x} = \begin{cases} \frac{1}{3} (\mathbf{T}_+(\mathbf{x}) - \mathbf{x}), & \text{if } k = +1, \\ \frac{1}{3} (\mathbf{T}_-(\mathbf{x}) - \mathbf{x}), & \text{if } k = -1. \end{cases} \quad (\text{B.33})$$

Finally, one can easily verify that the optimum value of k is equal to $+1$, since the distance of \mathbf{x} to the vertices of its inner Torricelli configuration $\mathbf{T}_+(\mathbf{x})$ is always less than or equal to its distance to the vertices of its outer Torricelli configuration $\mathbf{T}_-(\mathbf{x})$. Here, the equality only holds if \mathbf{x} is collinear. Thus, an optimal solution of (B.20) coincides with the double outer Napoleon transformation, $\mathbf{N}_-^2(\mathbf{x})$ (B.17), and it is the unique solution of (B.20) if \mathbf{x} is non-collinear. \blacksquare

As a final remark, we would like to note that our particular interest in the optimality of Napoleon triangles comes from our results on coordinated robot navigation, where a group of robots require to interchange their (structural) adjacencies through a minimum cost configuration determined by the double outer Napoleon transformation, refer to Chapter 2 for more details.

Appendix C

Sensor-Based Reactive Navigation: Supplementary Material

C.1 Geometric Interpretation of the Curvature Condition

A convenient way of characterizing metric limitations, such as the obstacle curvature condition in Assumption 4.2, of the “move-to-projected-goal” law is in terms of the enclosing balls of the goal \mathbf{x}^* , defined as:

Definition C.1 *The enclosing ball, $B_{\mathbf{x}} := B(\mathbf{x}^*, \|\mathbf{x} - \mathbf{x}^*\| - r)$, of the goal \mathbf{x}^* associated with a robot location $\mathbf{x} \in \mathbb{R}^n \setminus \overline{B(\mathbf{x}^*, r)}$ is the largest open ball, centered at \mathbf{x}^* , that does not intersect with the robot body $\overline{B(\mathbf{x}, r)}$.*

In other words, the enclosing ball $B_{\mathbf{x}}$ is the largest ball, centered at the goal \mathbf{x}^* , which can be circumnavigated by a disk-shaped robot of radius r , starting at location \mathbf{x} , without increasing the Euclidean distance to the goal.

Observe that, for any stationary point $\mathbf{s}_i \in \mathfrak{S}_i$ (4.18) associated with obstacle O_i , one has $B_{\mathbf{s}_i} = B(\mathbf{x}^*, \|\mathbf{x}^* - \Pi_{\overline{O}_i}(\mathbf{s}_i)\|)$ and $\Pi_{\overline{B_{\mathbf{s}_i}}}(\mathbf{s}_i) = \Pi_{\overline{O}_i}(\mathbf{s}_i)$, because \mathbf{s}_i , $\Pi_{\overline{O}_i}(\mathbf{s}_i)$ and \mathbf{x}^* are all collinear (Proposition 4.5). That is to say, $B_{\mathbf{s}_i}$ is tangent to (i.e, the osculating ball of) O_i at $\Pi_{\overline{O}_i}(\mathbf{s}_i)$. Hence, we have

$$\Pi_{\overline{B_{\mathbf{s}_i}}}(\mathbf{x}) = \left\| \mathbf{x}^* - \Pi_{\overline{O}_i}(\mathbf{s}_i) \right\| \frac{\mathbf{x} - \mathbf{x}^*}{\|\mathbf{x} - \mathbf{x}^*\|}, \quad \forall \mathbf{x} \in \mathbb{R}^n \setminus \overline{B_{\mathbf{s}_i}}, \quad (\text{C.1})$$

and so the Jacobian matrix $\mathbf{J}_{\Pi_{\overline{B_{\mathbf{s}_i}}}}(\mathbf{s}_i)$ of the metric projection of \mathbf{s}_i onto the associated enclosing ball $B_{\mathbf{s}_i}$ is given by

$$\mathbf{J}_{\Pi_{\overline{B_{\mathbf{s}_i}}}}(\mathbf{s}_i) = \frac{\left\| \mathbf{x}^* - \Pi_{\overline{O}_i}(\mathbf{s}_i) \right\|}{r + \left\| \mathbf{x}^* - \Pi_{\overline{O}_i}(\mathbf{s}_i) \right\|} \mathbf{Q}_i(\mathbf{s}_i), \quad (\text{C.2})$$

where

$$\mathbf{Q}_i(\mathbf{x}) := \mathbf{I} - \frac{\left(\mathbf{x} - \Pi_{\overline{O}_i}(\mathbf{x})\right)\left(\mathbf{x} - \Pi_{\overline{O}_i}(\mathbf{x})\right)^T}{\left\|\mathbf{x} - \Pi_{\overline{O}_i}(\mathbf{x})\right\|^2}, \quad \forall \mathbf{x} \in \mathbb{R}^n \setminus \overline{O}_i. \quad (\text{C.3})$$

Therefore, since $\mathbf{Q}_i(s_i) \preceq \mathbf{I}$, one can conclude that the upper bound in (4.24) of Assumption 4.2 is due to the enclosing ball B_{s_i} of the goal \mathbf{x}^* associated with s_i . Because any path, starting at $\mathbf{x} \in \mathbb{R}^n$, along which the distance to the goal \mathbf{x}^* is strictly decreasing should stay in $B_{\mathbf{x}}$ for all future time; and the “move-to-projected-goal” law yields such navigation paths (Theorem 4.3).

More precisely, the geometric connection between enclosing balls of the goal and the curvature condition in Assumption 4.2 can be established as follows:

Proposition C.1 *Let $s_i \in \mathfrak{S}_i$ (4.18) be a critical point associated with obstacle O_i . If $\overline{O}_i \setminus \Pi_{\overline{O}_i}(s_i) \subset B_{s_i}$, then*

$$\mathbf{J}_{\Pi_{\overline{O}_i}}(s_i) \prec \frac{\left\|\mathbf{x}^* - \Pi_{\overline{O}_i}(s_i)\right\|}{\left\|\mathbf{x}^* - \Pi_{\overline{O}_i}(s_i)\right\| + r} \mathbf{I}. \quad (\text{C.4})$$

Thus, if $\overline{O}_i \setminus \Pi_{\overline{O}_i}(s_i) \subset B_{s_i}$ for all $i \in \{1, 2, \dots, m\}$ and $s_i \in \mathfrak{S}_i$, then Assumption 4.2 holds.

Proof Since $\Pi_{\overline{B}_{s_i}}(s_i) = \Pi_{\overline{O}_i}(s_i)$, the result can be verified using a similar pattern of the proof of Lemma C.6; here the only difference is that the entire \overline{O}_i , except $\Pi_{\overline{O}_i}(s_i)$, is strictly contained in B_{s_i} . ■

Alternatively, using functional representations of obstacles, one can verify Assumption 4.2 as follows:

Proposition C.2 *Let each obstacle O_i be associated with a convex function $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ such that $O_i = f_i^{-1}(-\infty, c_i)$ for some $c_i \in \mathbb{R}$. Then, Assumption 4.2 holds if*

$$\frac{\nabla^2 f_i(\Pi_{\overline{O}_i}(s_i))}{\left\|\nabla f_i(\Pi_{\overline{O}_i}(s_i))\right\|} \succ \frac{1}{\left\|\mathbf{x}^* - \Pi_{\overline{O}_i}(s_i)\right\|}, \quad (\text{C.5})$$

for all $i \in \{1, 2, \dots, m\}$ and $s_i \in \mathfrak{S}_i$ (4.18).

Proof Consider the enclosing ball B_{s_i} of the goal \mathbf{x}^* associated with $s_i \in \mathfrak{S}_i$. We have from Definition C.1 that $B_{s_i} = \beta^{-1}\left(-\infty, \left\|\mathbf{x}^* - \Pi_{\overline{O}_i}(s_i)\right\|\right)$, where $\beta(\mathbf{x}) := \|\mathbf{x} - \mathbf{x}^*\|^2$. Hence, it follows that

$$\frac{\nabla^2 \beta(\Pi_{\overline{B}_{s_i}}(s_i))}{\left\|\nabla \beta(\Pi_{\overline{B}_{s_i}}(s_i))\right\|} = \frac{1}{\left\|\mathbf{x}^* - \Pi_{\overline{O}_i}(s_i)\right\|}. \quad (\text{C.6})$$

Therefore, since $\Pi_{\overline{B}_{s_i}}(s_i) = \Pi_{\overline{O}_i}(s_i)$, one can conclude the result from Lemma C.4 and Lemma C.5. ■

Two immediate corollaries of Proposition C.1 and Proposition C.2 for the case of spherical and ellipsoidal obstacles are:

Corollary C.1 *If all obstacles are open balls, then Assumption 4.2 holds for any goal $\mathbf{x}^* \in \mathcal{F}$.*

Corollary C.2 *Let each obstacle O_i be an open ellipsoid defined as $O_i = f_i^{-1}(-\infty, c_i)$ for some $c_i \in \mathbb{R}$, where $f_i := (\mathbf{x} - \mathbf{p}_i)^T \mathbf{A}_i (\mathbf{x} - \mathbf{p}_i)$ and $\mathbf{A}_i \in \mathbb{R}^{n \times n}$ is symmetric positive definite. Then, Assumption 4.2 holds if*

$$\frac{\lambda_{\min}(\mathbf{A}_i)}{\lambda_{\max}(\mathbf{A}_i)} > \frac{\|\mathbf{p}_i - \Pi_{\overline{O}_i}(\mathbf{s}_i)\|}{\|\mathbf{x}^* - \Pi_{\overline{O}_i}(\mathbf{s}_i)\|}, \quad (\text{C.7})$$

for all $i \in \{1, 2, \dots, m\}$ and $\mathbf{s}_i \in \mathfrak{S}_i$, where $\lambda_{\min}(\mathbf{A}_i)$ and $\lambda_{\max}(\mathbf{A}_i)$ are, respectively, the minimum and maximum eigenvalues of \mathbf{A}_i .

Proof The results follows from Proposition C.2 and

$$\frac{\nabla^2 f_i(\Pi_{\overline{O}_i}(\mathbf{s}_i))}{\|\nabla f_i(\Pi_{\overline{O}_i}(\mathbf{s}_i))\|} = \frac{\mathbf{A}}{\|\mathbf{A}(\mathbf{p}_i - \Pi_{\overline{O}_i}(\mathbf{s}_i))\|} \succcurlyeq \frac{\lambda_{\min}(\mathbf{A}_i)}{\lambda_{\max}(\mathbf{A}_i)} \frac{1}{\|\mathbf{p}_i - \Pi_{\overline{O}_i}(\mathbf{s}_i)\|} \mathbf{I}. \quad (\text{C.8})$$

■

In consequence, one can briefly conclude that it is easier for a robot to navigate around obstacles more spherical (i.e., not too flat) and towards goal locations away from obstacles, while strictly decreasing the Euclidean distance to the goal.

C.2 Uniqueness of Maximum Margin Separating Hyperplanes

For any two disjoint convex sets $A, B \in \mathbb{R}^n$, there can be more than one pair of points $\mathbf{a} \in \overline{A}$ and $\mathbf{b} \in \overline{B}$ achieving $\|\mathbf{a} - \mathbf{b}\| = d(A, B)$; however, they all have the same maximum margin separating hyperplane:

Lemma C.1 *Let $A, B \subset \mathbb{R}^n$ be two disjoint convex sets, and $\mathbf{a}_1, \mathbf{a}_2 \in A$ and $\mathbf{b}_1, \mathbf{b}_2 \in B$ be points with $\|\mathbf{a}_1 - \mathbf{b}_1\| = \|\mathbf{a}_2 - \mathbf{b}_2\| = d(A, B)$. Then, for any $\mathbf{x} \in \mathbb{R}^n$, the following equality always holds*

$$(\mathbf{a}_1 - \mathbf{b}_1)^T \left(\mathbf{x} - \frac{\mathbf{a}_1 + \mathbf{b}_1}{2} \right) = (\mathbf{a}_2 - \mathbf{b}_2)^T \left(\mathbf{x} - \frac{\mathbf{a}_2 + \mathbf{b}_2}{2} \right). \quad (\text{C.9})$$

Proof First, to see that $\mathbf{a}_1 - \mathbf{b}_1 = \mathbf{a}_2 - \mathbf{b}_2$, consider

$$(\mathbf{a}_1 - \mathbf{b}_1)^T (\mathbf{a}_2 - \mathbf{b}_2) = (\mathbf{a}_1 - \mathbf{b}_1)^T \left(\mathbf{a}_2 - \frac{\mathbf{a}_1 + \mathbf{b}_1}{2} \right) + (\mathbf{b}_1 - \mathbf{a}_1)^T \left(\mathbf{b}_2 - \frac{\mathbf{a}_1 + \mathbf{b}_1}{2} \right), \quad (\text{C.10})$$

$$= d(A, B)^2 + \frac{1}{2} \underbrace{(\mathbf{a}_1 - \mathbf{b}_1)^T (\mathbf{a}_2 - \mathbf{a}_1)}_{\geq 0, \text{ by Theorem 4.2}} + \frac{1}{2} \underbrace{(\mathbf{b}_1 - \mathbf{a}_1)^T (\mathbf{b}_2 - \mathbf{b}_1)}_{\geq 0, \text{ by Theorem 4.2}}, \quad (\text{C.11})$$

$$\geq d(A, B)^2. \quad (\text{C.12})$$

where the inequality follows from Theorem 4.2, because $\|a_1 - b_1\| = d(A, B) = d(a_1, B) = d(A, b_1)$. Moreover, it follows from the Cauchy-Schwartz inequality that

$$(a_1 - b_1)^T (a_2 - b_2) \leq \|a_1 - b_1\| \|a_2 - b_2\| = d(A, B)^2. \quad (\text{C.13})$$

Hence, since $(a_1 - b_1)^T (a_2 - b_2) = \|a_1 - b_1\|^2 = \|a_2 - b_2\|^2$, one always has

$$a_1 - b_1 = a_2 - b_2. \quad (\text{C.14})$$

Also observe from (C.11) that

$$(a_1 - b_1)^T (a_1 - a_2) = 0, \quad \text{and} \quad (a_1 - b_1)^T (b_1 - b_2) = 0. \quad (\text{C.15})$$

Therefore, the result can be verified as follows:

$$(a_2 - b_2)^T \left(x - \frac{a_2 + b_2}{2} \right) = (a_1 - b_1)^T \left(x - \frac{a_2 + b_2}{2} \right), \quad (\text{C.16})$$

$$= (a_1 - b_1)^T \left(x - \frac{a_1 + b_1}{2} \right) + \underbrace{(a_1 - b_1)^T \left(\frac{a_1 + b_1}{2} - \frac{a_2 + b_2}{2} \right)}_{=0, \text{ by (C.15)}}, \quad (\text{C.17})$$

$$= (a_1 - b_1)^T \left(x - \frac{a_1 + b_1}{2} \right). \quad (\text{C.18})$$

■

C.3 On the Jacobian of Metric Projection

A well known property of metric projections is being nonexpansive:

Lemma C.2 ([219]) *The metric projection onto a closed convex set $A \subseteq \mathbb{R}^n$ is Lipschitz continuous with Lipschitz constant 1, i.e., $\|\Pi_A(x) - \Pi_A(y)\| \leq \|x - y\|$ for all $x, y \in \mathbb{R}^n$.*

Note that a Lipschitz function in \mathbb{R}^n is differentiable almost everywhere, and Π_A is piecewise continuously differentiable [135].

Lemma C.3 ([109, 85]) *The Jacobian $\mathbf{J}_{\Pi_K}(x)$ of the metric projection onto a closed convex set $K \subseteq \mathbb{R}^n$ with twice continuously differentiable (\mathcal{C}^2) boundary is a positive semi-definite and symmetric operator of norm at most unity, i.e.,*

$$\mathbf{0} \preceq \mathbf{J}_{\Pi_K}(x) \preceq \mathbf{I}, \quad \forall x \in \mathbb{R}^n \setminus K, \quad (\text{C.19})$$

and one has $\mathbf{J}_{\Pi_K}(x)(x - \Pi_K(x)) = 0$.

The Jacobian matrix of the metric projection onto a convex set can be analytically obtained using its functional representation in terms of a level set of a convex function:

Lemma C.4 Let $K \in \mathbb{R}^n$ be a closed convex set associated with a twice continuously differentiable (\mathcal{C}^2) convex function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ such that $K = f^{-1}(-\infty, c]$ for some $c \in \mathbb{R}$.

Then, the Jacobian $\mathbf{J}_{\Pi_K}(\mathbf{x})$ of the metric projection of $\mathbf{x} \in \mathbb{R}^n \setminus K$ onto K is given by ¹

$$\mathbf{J}_{\Pi_K}(\mathbf{x}) = \mathbf{Q}(\mathbf{I} + \mathbf{Q}\mathbf{P}\mathbf{Q})^{-1} \mathbf{Q} = \mathbf{Q} - \mathbf{I} + (\mathbf{I} + \mathbf{Q}\mathbf{P}\mathbf{Q})^{-1} , \quad (\text{C.20})$$

where

$$\mathbf{Q} := \mathbf{I} - \frac{(\mathbf{x} - \Pi_K(\mathbf{x}))(\mathbf{x} - \Pi_K(\mathbf{x}))^T}{\|\mathbf{x} - \Pi_K(\mathbf{x})\|^2} , \quad (\text{C.21})$$

$$\mathbf{P} := \frac{\|\mathbf{x} - \Pi_K(\mathbf{x})\|}{\|\nabla f(\Pi_K(\mathbf{x}))\|} \nabla^2 f(\Pi_K(\mathbf{x})) . \quad (\text{C.22})$$

Proof Using the relation between K and f , one can rewrite the metric project onto K as

$$\Pi_K(\mathbf{x}) = \arg \min_{\mathbf{y} \in K} \|\mathbf{y} - \mathbf{x}\| = \arg \min_{f(\mathbf{y}) \leq c} \|\mathbf{y} - \mathbf{x}\| . \quad (\text{C.23})$$

Further, due to the optimality of $\Pi_K(\mathbf{x})$, the outward surface normal of K at $\Pi_K(\mathbf{x})$ is given by $\frac{\mathbf{x} - \Pi_K(\mathbf{x})}{\|\mathbf{x} - \Pi_K(\mathbf{x})\|} = \frac{\nabla f(\Pi_K(\mathbf{x}))}{\|\nabla f(\Pi_K(\mathbf{x}))\|}$, and we have

$$\mathbf{x} = \Pi_K(\mathbf{x}) + \|\mathbf{x} - \Pi_K(\mathbf{x})\| \frac{\nabla f(\Pi_K(\mathbf{x}))}{\|\nabla f(\Pi_K(\mathbf{x}))\|} . \quad (\text{C.24})$$

Hence, using $\mathbf{J}_{\Pi_K}(\mathbf{x})(\mathbf{x} - \Pi_K(\mathbf{x})) = 0$ (Lemma C.3), the derivative of (C.24) yields

$$\mathbf{J}_{\Pi_K}(\mathbf{x}) = (\mathbf{I} - \mathbf{Q}\mathbf{P})^{-1} \mathbf{Q} . \quad (\text{C.25})$$

Note that it is not straightforward to observe that the closed form of $\mathbf{J}_{\Pi_K}(\mathbf{x})$ in (C.25) is positive definite and symmetric (Lemma C.3). Alternatively, using the matrix identity $(\mathbf{I} + \mathbf{A}\mathbf{B})^{-1} \mathbf{A} = \mathbf{A}(\mathbf{I} + \mathbf{B}\mathbf{A})^{-1}$ [176] and $\mathbf{Q}\mathbf{Q} = \mathbf{Q}$, a more informative closed form of $\mathbf{J}_{\Pi_K}(\mathbf{x})$ can be obtained as follows:

$$\mathbf{J}_{\Pi_K}(\mathbf{x}) = (\mathbf{I} - \mathbf{Q}\mathbf{P})^{-1} \mathbf{Q} = \underbrace{(\mathbf{I} - \mathbf{Q}\mathbf{Q}\mathbf{P})^{-1}}_{=\mathbf{Q}(\mathbf{I} - \mathbf{Q}\mathbf{P}\mathbf{Q})^{-1}} \mathbf{Q}\mathbf{Q} = \mathbf{Q}(\mathbf{I} - \mathbf{Q}\mathbf{P}\mathbf{Q})^{-1} \mathbf{Q} . \quad (\text{C.26})$$

Alternatively, using a special case of Woodbury matrix identity (a.k.a. the matrix inversion lemma) [176],

$$(\mathbf{I} + \mathbf{Q}\mathbf{P})^{-1} = \mathbf{I} - \mathbf{Q}(\mathbf{I} + \mathbf{P}\mathbf{Q})^{-1} \mathbf{P} , \quad (\text{C.27})$$

we also have

$$\mathbf{J}_{\Pi_K}(\mathbf{x}) = (\mathbf{I} - \mathbf{Q}\mathbf{P})^{-1} \mathbf{Q} = \left(\mathbf{I} - \mathbf{Q}(\mathbf{I} + \mathbf{P}\mathbf{Q})^{-1} \mathbf{P} \right) \mathbf{Q} , \quad (\text{C.28})$$

¹Here, $\nabla f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and $\nabla^2 f : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$ denote the gradient and Hessian of a twice continuously differentiable function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, respectively.

$$= \mathbf{Q} - \mathbf{I} + \underbrace{\mathbf{I} - \mathbf{Q}(\mathbf{I} + \mathbf{P}\mathbf{Q}\mathbf{Q})^{-1}\mathbf{P}\mathbf{Q}}_{=(\mathbf{I} + \mathbf{Q}\mathbf{P}\mathbf{Q})^{-1}}, \quad (\text{C.29})$$

$$= \mathbf{Q} - \mathbf{I} + (\mathbf{I} + \mathbf{Q}\mathbf{P}\mathbf{Q})^{-1}. \quad (\text{C.30})$$

Recall that $\mathbf{Q}\mathbf{Q} = \mathbf{Q}$. Thus, the lemma follows. \blacksquare

Lemma C.5 *Let $K_1, K_2 \in \mathbb{R}^n$ be two closed convex sets associated with twice differentiable convex functions $f_1 : \mathbb{R}^n \rightarrow \mathbb{R}$ and $f_2 : \mathbb{R}^n \rightarrow \mathbb{R}$, respectively, such that $K_1 = f_1^{-1}(-\infty, c_1]$ and $K_2 = f_2^{-1}(-\infty, c_2]$ for some $c_1, c_2 \in \mathbb{R}$. And let $\mathbf{x} \in \mathbb{R}^n \setminus (K_1 \cup K_2)$ with $\Pi_{K_1}(\mathbf{x}) = \Pi_{K_2}(\mathbf{x})$.*

Then the following equivalence holds

$$\frac{\nabla^2 f_1(\Pi_{K_1}(\mathbf{x}))}{\|\nabla f_1(\Pi_{K_1}(\mathbf{x}))\|} \preceq \frac{\nabla^2 f_2(\Pi_{K_2}(\mathbf{x}))}{\|\nabla f_2(\Pi_{K_2}(\mathbf{x}))\|} \iff \mathbf{J}_{\Pi_{K_1}}(\mathbf{x}) \succcurlyeq \mathbf{J}_{\Pi_{K_2}}(\mathbf{x}). \quad (\text{C.31})$$

Proof The result directly follows from Lemma C.4 and the following matrix relation of positive definite matrices, \mathbf{A} and \mathbf{B} , [34]

$$\mathbf{A} \preceq \mathbf{B} \iff \mathbf{A}^{-1} \succcurlyeq \mathbf{B}^{-1}. \quad (\text{C.32})$$

Lemma C.6 *Let $K_1, K_2 \subseteq \mathbb{R}^n$ be two convex sets with twice continuously differentiable (\mathcal{C}^2) boundary. If $K_1 \supseteq K_2$, then the Jacobians $\mathbf{J}_{\Pi_{K_1}}(\mathbf{x})$ and $\mathbf{J}_{\Pi_{K_2}}(\mathbf{x})$ of metric projections onto K_1 and K_2 , respectively, satisfy*

$$\mathbf{J}_{\Pi_{K_1}}(\mathbf{x}) \succcurlyeq \mathbf{J}_{\Pi_{K_2}}(\mathbf{x}), \quad (\text{C.33})$$

for all $\mathbf{x} \in \mathbb{R}^n \setminus K_1$ with $\Pi_{K_1}(\mathbf{x}) = \Pi_{K_2}(\mathbf{x})$.

Proof For any $\mathbf{x} \in \mathbb{R}^n \setminus K_1$ with $\Pi_{K_1}(\mathbf{x}) = \Pi_{K_2}(\mathbf{x})$ and $\mathbf{y} \in \mathbb{R}^n$, one can write the metric projection of $\mathbf{x} + \mathbf{y}$ onto K_1 and K_2 , respectively, as

$$\Pi_{K_1}(\mathbf{x} + \mathbf{y}) = \Pi_{K_1}(\mathbf{x}) + \mathbf{J}_{\Pi_{K_1}}(\mathbf{x})\mathbf{y} + o(\mathbf{y}), \quad (\text{C.34a})$$

$$\Pi_{K_2}(\mathbf{x} + \mathbf{y}) = \Pi_{K_2}(\mathbf{x}) + \mathbf{J}_{\Pi_{K_2}}(\mathbf{x})\mathbf{y} + o(\mathbf{y}), \quad (\text{C.34b})$$

where $\lim_{\|\mathbf{y}\| \rightarrow 0} \frac{o(\mathbf{y})}{\|\mathbf{y}\|} = 0$. Further, since $K_1 \supseteq K_2$, by the monotonicity of metric projections, we have

$$\|\mathbf{x} + \mathbf{y} - \Pi_{K_1}(\mathbf{x} + \mathbf{y})\|^2 \leq \|\mathbf{x} + \mathbf{y} - \Pi_{K_2}(\mathbf{x} + \mathbf{y})\|^2. \quad (\text{C.35})$$

Now it follows from (C.34), (C.35) and Lemma C.3 that

$$\begin{aligned} & \frac{\left\| \left(\mathbf{I} - \mathbf{J}_{\Pi_{K_2}}(\mathbf{x}) \right) \mathbf{y} \right\|^2}{\|\mathbf{y}\|^2} - \frac{\left\| \left(\mathbf{I} - \mathbf{J}_{\Pi_{K_1}}(\mathbf{x}) \right) \mathbf{y} \right\|^2}{\|\mathbf{y}\|^2} \geq \\ & \frac{\|\mathbf{x} - \Pi_{K_1}(\mathbf{x}) - o(\mathbf{y})\|^2}{\|\mathbf{y}\|^2} - \frac{\|\mathbf{x} - \Pi_{K_2}(\mathbf{x}) - o(\mathbf{y})\|^2}{\|\mathbf{y}\|^2} + \frac{2\mathbf{y}^T \left(\mathbf{J}_{\Pi_{K_1}}(\mathbf{x}) - \mathbf{J}_{\Pi_{K_2}}(\mathbf{x}) \right) o(\mathbf{y})}{\|\mathbf{y}\|^2}, \end{aligned} \quad (\text{C.36})$$

where the right hand side converges to zero as $\|y\| \rightarrow 0$. Therefore, for any $y \in \mathbb{R}^n$, one always has

$$\left\| \left(I - \mathbf{J}_{\Pi_{K_2}}(\mathbf{x}) \right) y \right\|^2 \geq \left\| \left(I - \mathbf{J}_{\Pi_{K_1}}(\mathbf{x}) \right) y \right\|^2. \quad (\text{C.37})$$

Thus, the result follows since $\mathbf{0} \preccurlyeq \mathbf{J}_{\Pi_{K_1}}(\mathbf{x}), \mathbf{J}_{\Pi_{K_2}}(\mathbf{x}) \preccurlyeq \mathbf{I}$ (Lemma C.3). \blacksquare

C.4 Convexity in Polar Coordinates

Similar to the notion of convexity in Cartesian coordinates, a polar curve $\rho : (\theta_l, \theta_u) \rightarrow \mathbb{R}_{\geq 0}$ is said to be *convex* with respect to the pole if and only if its epigraph,² $\text{epi}\rho := \{(\theta, \varrho) \mid \theta \in (\theta_l, \theta_u), \varrho \geq \rho(\theta)\}$, is a convex set; and, likewise, ρ is said to be *concave* if and only if its hypograph, $\text{hyp}\rho := \{(\theta, \varrho) \mid \theta \in (\theta_l, \theta_u), 0 \leq \varrho \leq \rho(\theta)\}$ is a convex set [157, 75], see Figure C.1.

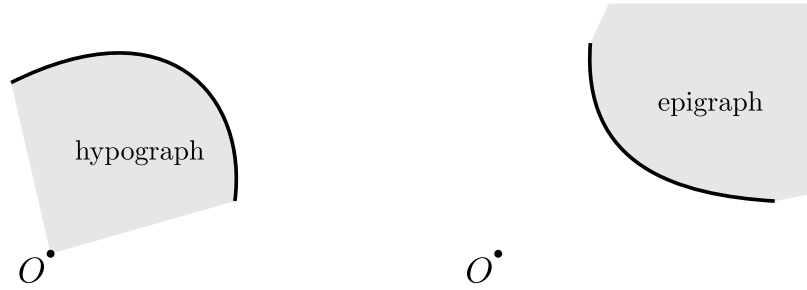


Figure C.1: Convexity in polar coordinates. A polar curve is convex (concave) with respect to the pole iff its epigraph (hypograph) is a convex set, as illustrated on the right (left, respectively).

Alternatively, like the first- and second-order conditions for convexity of Cartesian functions, one can verify the convexity of a polar curve as follows:

Theorem C.1 (Second-Order Convexity Condition [157]) *A twice differentiable polar curve*

²Here the epigraph and the hypograph of a polar curve are given in polar coordinates, and one can equivalently write them in Cartesian coordinates as

$$\text{epi}\rho = \{(\varrho \cos \theta, \varrho \sin \theta) \mid \theta \in (\theta_l, \theta_u), \varrho \geq \rho(\theta)\}, \quad (\text{C.38})$$

$$\text{hyp}\rho = \{(\varrho \cos \theta, \varrho \sin \theta) \mid \theta \in (\theta_l, \theta_u), 0 \leq \varrho \leq \rho(\theta)\}. \quad (\text{C.39})$$

$\rho : (\theta_l, \theta_u) \rightarrow \mathbb{R}_{>0}$ is said to be convex with respect to the pole if³

$$\Gamma := \rho^2 + 2 \left(\frac{d\rho}{d\theta} \right)^2 - \rho \frac{d^2\rho}{d\theta^2} \leq 0 . \quad (\text{C.42})$$

Theorem C.2 (Three-Point Convexity Condition [75]) *A polar curve $\rho : (\theta_l, \theta_u) \rightarrow \mathbb{R}_{>0}$ is convex to the pole if⁴*

$$\det \left(\begin{bmatrix} \frac{1}{\rho(\theta_1)} & \cos \theta_1 & \sin \theta_1 \\ \frac{1}{\rho(\theta_2)} & \cos \theta_2 & \sin \theta_2 \\ \frac{1}{\rho(\theta_3)} & \cos \theta_3 & \sin \theta_3 \end{bmatrix} \right) \cdot \det \left(\begin{bmatrix} 1 & \cos \theta_1 & \sin \theta_1 \\ 1 & \cos \theta_2 & \sin \theta_2 \\ 1 & \cos \theta_3 & \sin \theta_3 \end{bmatrix} \right) \leq 0 , \quad (\text{C.44})$$

for all $\theta_1, \theta_2, \theta_3 \in (\theta_l, \theta_u)$.

Note that the second determinant term in (C.44) quantifies the circular order of θ_1, θ_2 and θ_3 , i.e., it is positive (negative) if these angles are given in counter-clockwise (clockwise, respectively) order.

In accordance with Theorem C.2, since a LIDAR scanner has a fixed angular resolution in practice, say $\Delta\theta \in (0, \pi)$, to check the convexity of a LIDAR scan in counter-clockwise angular order, we find it convenient to define

$$\Upsilon(\theta) := \det \left(\begin{bmatrix} \frac{1}{\rho(\theta-\Delta\theta)} & \cos(\theta-\Delta\theta) & \sin(\theta-\Delta\theta) \\ \frac{1}{\rho(\theta)} & \cos(\theta) & \sin(\theta) \\ \frac{1}{\rho(\theta+\Delta\theta)} & \cos(\theta+\Delta\theta) & \sin(\theta+\Delta\theta) \end{bmatrix} \right) . \quad (\text{C.45})$$

Therefore, one can identify the convex polar curve segments of a LIDAR scan using the convexity measures Γ (C.42) and Υ (C.45) as illustrated in Figure C.2.

³In [157], the convexity of a polar curve with respect to the pole is characterized based on its tangent lines: a polar curve at a point is convex if and only if the curve in a small neighborhood of that point lies on the opposite side of the tangent at that point to the pole. Accordingly, the second-order convexity condition in (C.42) is derived using the perpendicular distance p of the pole to the tangent line of a polar curve ρ at point $(\theta, \rho(\theta))$, given by

$$\frac{1}{p^2} = u^2 + \left(\frac{du}{d\theta} \right)^2 , \quad (\text{C.40})$$

where $u := \frac{1}{\rho}$; and the polar curve ρ is said to be convex to the pole if and only if $\frac{dp}{d\rho}$ is negative, where

$$\frac{dp}{d\rho} = p^3 u^2 \left(u + \frac{d^2 u}{d\theta^2} \right) = \frac{p^3}{\rho^2} \left(\rho^2 + 2 \left(\frac{d\rho}{d\theta} \right)^2 - \rho \frac{d^2\rho}{d\theta^2} \right) . \quad (\text{C.41})$$

⁴Let $\mathbf{v}_t = (\cos \theta_t, \sin \theta_t)$ and $\mathbf{p}_t = (\rho(\theta_t) \cos \theta_t, \rho(\theta_t) \sin \theta_t)$ for $t = 1, 2, 3$. Then, to have a geometric understanding of the three-point convexity condition one can equivalently rewrite (C.44) as

$$((\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_2)) \cdot ((\mathbf{v}_2 - \mathbf{v}_1) \times (\mathbf{v}_3 - \mathbf{v}_2)) \leq 0 , \quad (\text{C.43})$$

where \times and \cdot denote the cross and dot products, respectively.

⁵ Here, we set the LIDAR's angular resolution to $\Delta\theta = \frac{\pi}{100}$, and approximately compute the first- and second-order derivatives of a simulated LIDAR range data, respectively, using its three-point first- and second-order central differences [87] after smoothing with a five-point Gaussian moving average filter with unit variance, $\sigma^2 = 1$ [208].

⁶ A practical heuristic for identifying convex segments of a LIDAR scan is its segmentation based on

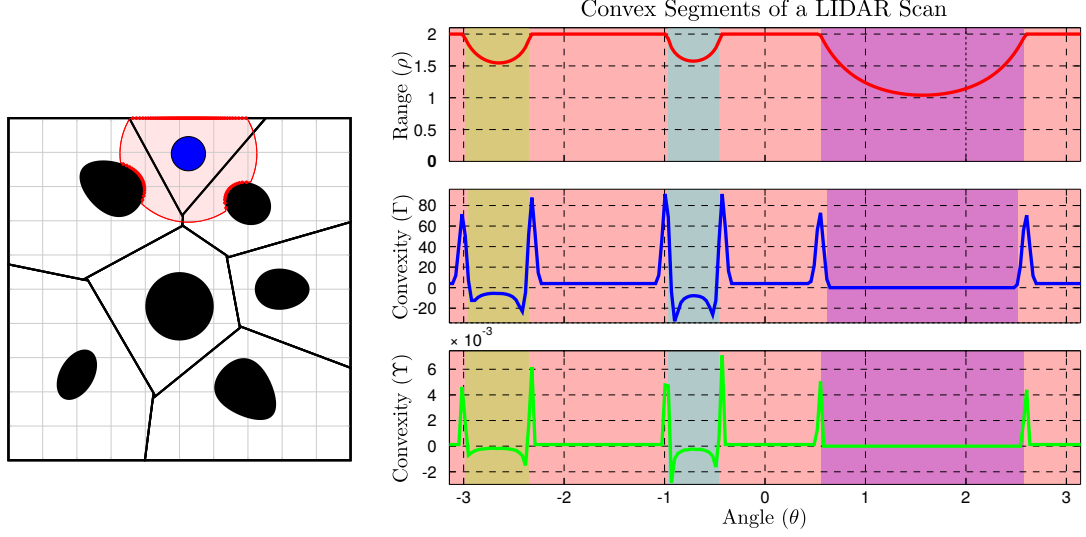


Figure C.2: Segmentation of a LIDAR scan into convex polar curves using convexity measures Γ (C.42) and Υ (C.44).^{5 6}

C.5 An Extension for a Discrete-Time Robot Model

Keeping in mind its potential application to online robot navigation in a nonconvex environment when combined with a standard (e.g., sampling based) motion planning algorithm [140, 55] — a very promising future research direction that we will explore, we now introduce a discrete-time version of the “move-to-projected-goal” law in (4.17) to iteratively navigate towards a designated goal location $\mathbf{x}^* \in \mathcal{F}$ as follows: for any $\mathbf{x}^k \in \mathcal{F}$,

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \left(\mathbf{x}^k - \Pi_{\mathcal{LF}(\mathbf{x}^k)}(\mathbf{x}^*) \right) \Delta t, \quad (\text{C.46})$$

where $k \in \mathbb{N}$ is a discrete time index, $\Delta t \in (0, 1]$ is a fixed sample time (step size), and $\Pi_{\mathcal{LF}(\mathbf{x}^k)}(\mathbf{x}^*)$ (4.6) is the metric projection of the goal \mathbf{x}^* onto the robot’s local free space $\mathcal{LF}(\mathbf{x}^k)$ (4.15). Note that we here avoid collisions along the line segment joining consecutive robot states, \mathbf{x}^k and \mathbf{x}^{k+1} , by limiting the range of values of Δt to $(0, 1]$, because \mathbf{x}^{k+1} becomes a convex combination of the robot state \mathbf{x}^k and the projected goal $\Pi_{\mathcal{LF}(\mathbf{x}^k)}(\mathbf{x}^*)$, i.e., $\mathbf{x}^{k+1} = (1 - \Delta t) \mathbf{x}^k + \Delta t \Pi_{\mathcal{LF}(\mathbf{x}^k)}(\mathbf{x}^*)$, and the line segment joining them is always free of collisions (Corollary 4.1).

Therefore, using the continuity of the move-to-projected-goal law in (4.17) (Proposition 4.2) and the type of its stationary points (Proposition 4.6), one can conclude that:

Corollary C.3 *If Assumption 4.2 holds for the goal and for all obstacles, then the discrete-time “move-to-projected-goal” law in (C.46) starting from almost any robot location in \mathcal{F} (4.1) iteratively reaches a small neighborhood, $B(\mathbf{x}^*, \epsilon)$ for some $\epsilon > 0$, of the goal \mathbf{x}^* in finite steps with the guarantee of no collisions along the line segments joining two consecutive*

local maxima; however, such a heuristic approach might detect some concave curve segments in addition to all convex segments in a LIDAR scan.

robot states, while strictly decreasing the Euclidean distance to the goal.

Note that the discrete-time “move-to-projected-goal” law in (C.46) can be simply adapted to limited range sensing models, by using the robot’s sensed local free space \mathcal{LF}_s (4.36) or the line-of-sight local free space $\mathcal{LF}_\mathcal{L}$ (4.41), as well as to the differential drive model while retaining the convergence and collision avoidance guarantees.

To demonstrate its motion pattern, we present in Figure C.3 the resulting navigation paths of the discrete-time “move-to-projected-goal” law in (C.46) for different sampling times and sensing models.

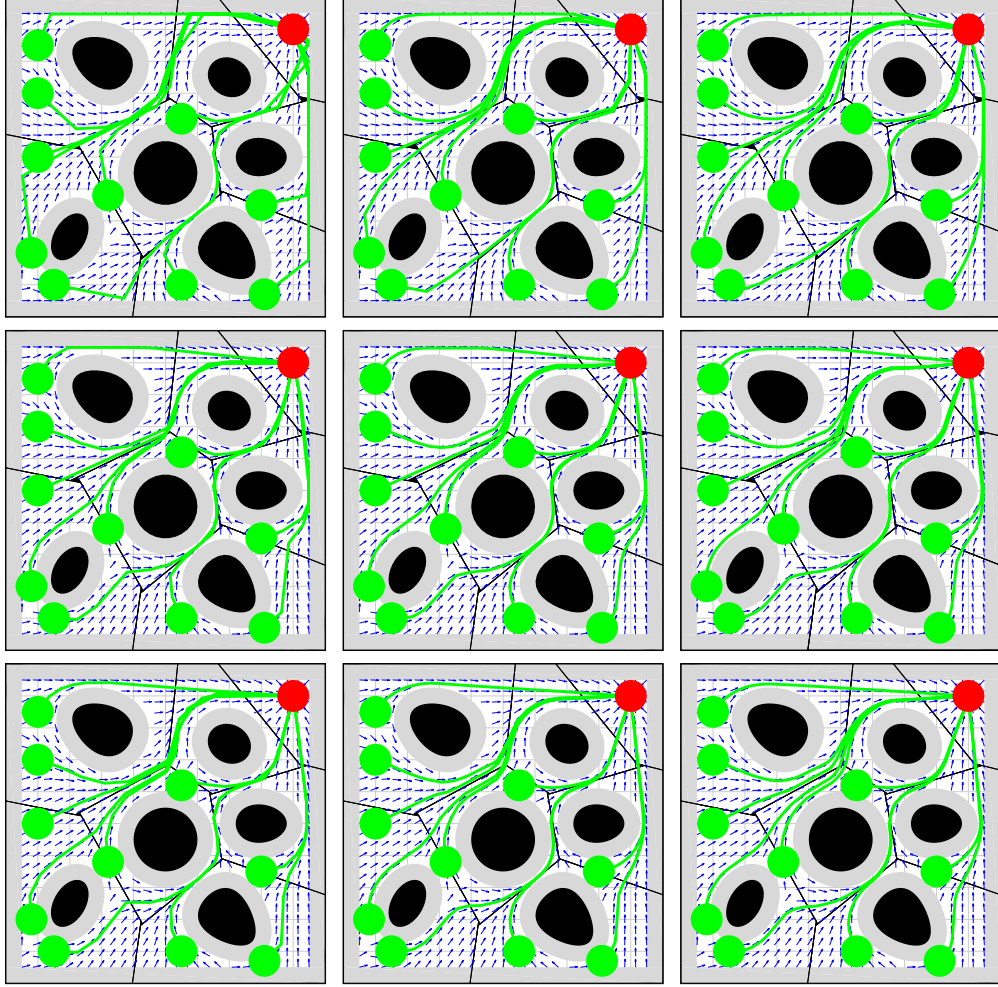


Figure C.3: Example navigation paths of the discrete-time “move-to-projected-goal” law for different sampling times and sensing models: (left) $\Delta t = 1$, (middle) $\Delta t = 0.5$, and (right) $\Delta t = 0.25$; and (top) Voronoi-adjacent ¹² obstacle sensing, and (center) a fixed radius sensory footprint, and (bottom) a limited range line-of-sight sensor.

C.6 Motion Pattern Far Away from the Goal

In Figure C.4, we present the motion pattern generated by the “move-to-projected-goal” law starting at a set of initial robot configurations far away from the goal, located at the upper right conner of a 50×10 environment populated with convex obstacles, for different sensing and actuation models.

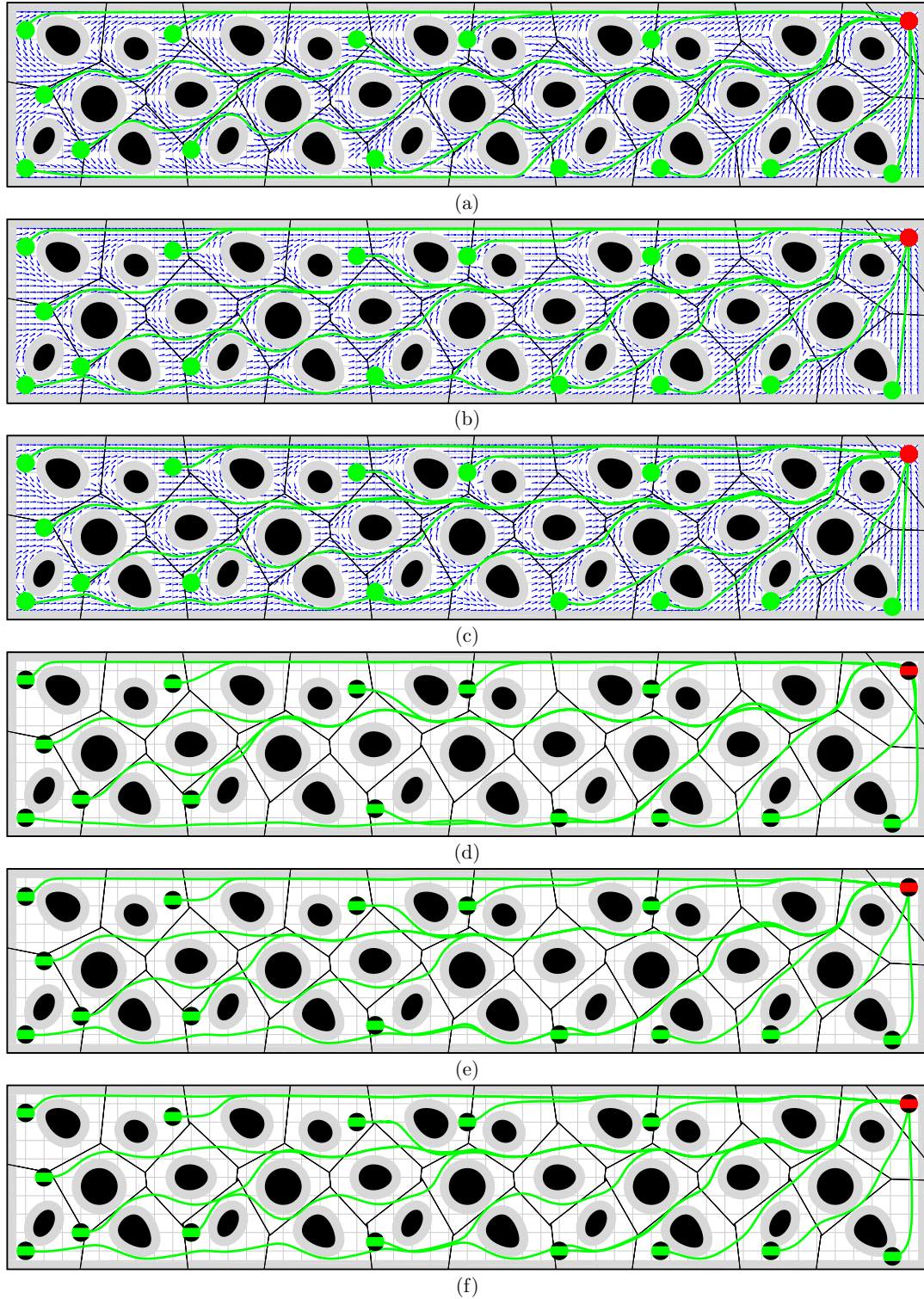


Figure C.4: Example navigation trajectories of the “move-to-projected-goal” law starting at a set of initial conditions (green) far away from the goal (red) for different sensing and actuation models: (a,b,c) a fully actuated robot, (d,e,f) a differential drive robot, (a,d) Voronoi-adjacent¹² obstacle sensing, (b,e) a fixed radius sensory footprint, (c,f) a limited range line-of-sight sensor.

Bibliography

- [1] A. Abrams and R. Ghrist. Finding topology in a factory: configuration spaces. *The American Mathematical Monthly*, 109(2):140–150, 2002.
- [2] 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*, 12(4):1309–1317, 2015.
- [3] K. Akkaya and M. Younis. A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, 3(3):325–349, 2005.
- [4] M. S. Aldenderfer and R. Blashfield. *Cluster Analysis*. Beverly Hill Sage Publications, 1984.
- [5] B. L. Allen and M. Steel. Subtree transfer operations and their induced metrics on evolutionary trees. *Annals of Combinatorics*, 5:1–15, 2001.
- [6] J.-M. Ame, C. Rivault, and J.-L. Deneubourg. Cockroach aggregation based on strain odour recognition. *Animal Behaviour*, 68(4):793–801, 2004.
- [7] C. Anderson and N. R. Franks. Teams in animal societies. *Behavioral Ecology*, 12(5): 534–540, 2001.
- [8] V. I. Arnold. *Ordinary Differential Equations*. MIT Press, 1973.
- [9] O. Arslan and D. E. Koditschek. Sensor-based reactive navigation in unknown convex sphere worlds. *(submitted to) the 12th International Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2016.
- [10] O. Arslan and D. E. Koditschek. Anytime hierarchical clustering. *arXiv preprint arXiv:1404.3439*, 2014.
- [11] O. Arslan and D. E. Koditschek. A recursive, distributed minimum spanning tree algorithm for mobile ad hoc networks. *Poster presented at RSS2014 Workshop on Communication-aware Robotics: New Tools for Multi-Robot Networks, Autonomous Vehicles, and Localization*, 2014. Online Available: <http://kodlab.seas.upenn.edu/0mur/RSS2014CarNetWorkshop>.
- [12] O. Arslan and D. E. Koditschek. Voronoi-based coverage control of heterogeneous disk-shaped robots. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 4259–4266, 2016.

- [13] O. Arslan and D. E. Koditschek. Exact robot navigation using power diagrams. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 1–8, 2016.
- [14] O. Arslan and D. E. Koditschek. On the optimality of Napoleon triangles. *Journal of Optimization Theory and Applications*, 170(1):97–106, 2016.
- [15] O. Arslan, Y. Baryshnikov, D. P. Guralnik, and D. E. Koditschek. Hierarchically clustered navigation of distinct euclidean particles. In *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*, pages 946–953, 2012.
- [16] O. Arslan, D. P. Guralnik, and D. E. Koditschek. Navigation of distinct euclidean particles via hierarchical clustering. In H. L. Akin, N. M. Amato, V. Isler, and A. F. van der Stappen, editors, *Algorithmic Foundations of Robotics XI: Selected Contributions of the Eleventh International Workshop on the Algorithmic Foundations of Robotics*, volume 107, pages 19–36. Springer International Publishing, 2015.
- [17] O. Arslan, D. P. Guralnik, and D. E. Koditschek. Discriminative measures for comparison of phylogenetic trees. (*submitted to*) *Discrete Applied Mathematics*, 2016. Online Available: <http://kodlab.seas.upenn.edu/Omur/NNITechReport2013>.
- [18] O. Arslan, D. P. Guralnik, and D. E. Koditschek. Coordinated robot navigation via hierarchical clustering. *Robotics, IEEE Transactions on*, 32(2):352–371, 2016.
- [19] A. Astolfi. Exponential stabilization of a wheeled mobile robot via discontinuous control. *Journal of dynamic systems, measurement, and control*, 121(1):121–126, 1999.
- [20] F. Aurenhammer. Power diagrams: Properties, algorithms and applications. *SIAM Journal on Computing*, 16(1):78–96, 1987.
- [21] N. Ayanian, V. Kumar, and D. Koditschek. Synthesis of controllers to create, maintain, and reconfigure robot formations with communication constraints. In C. Pradalier, R. Siegwart, and G. Hirzinger, editors, *Robotics Research: The 14th International Symposium (ISRR)*, volume 70, pages 625–642. Springer Berlin Heidelberg, 2011.
- [22] A. Back, J. Guckenheimer, and M. Myers. A dynamical simulation facility for hybrid systems. In *Hybrid Systems*, volume 736 of *Lecture Notes in Computer Science*, pages 255–267. Springer, 1993.
- [23] G. W. Barlow. Hexagonal territories. *Animal Behaviour*, 22:876–878, 1974.
- [24] J.-P. Barthélemy and F. McMorris. The median procedure for n-trees. *Journal of Classification*, 3(2):329–334, 1986.
- [25] Y. Baryshnikov and D. P. Guralnik. Hierarchical clustering and configuration spaces. (*in preparation*).

- [26] Y. Baryshnikov and B. Shapiro. How to run a centipede: a topological perspective. In *Geometric Control Theory and Sub-Riemannian Geometry*, pages 37–51. Springer, 2014.
- [27] Y. Baryshnikov, P. Bubenik, and M. Kahle. Min-type Morse theory for configuration spaces of hard spheres. *International Mathematics Research Notices*, 2014(9):2577–2592, 2014.
- [28] J. C. Bednarz. Cooperative hunting in Harris’ hawks (*Parabuteo unicinctus*). *Science*, 239(4847):1525–1527, 1988.
- [29] C. Belta and V. Kumar. Abstraction and control for groups of robots. *Robotics, IEEE Transactions on*, 20(5):865–875, 2004.
- [30] C. Belta, V. Isler, and G. Pappas. Discrete abstractions for robot motion planning and control in polygonal environments. *Robotics, IEEE Transactions on*, 21(5):864–874, 2005.
- [31] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, 1975.
- [32] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.
- [33] N. P. Bhatia and G. P. Szegő. *Dynamical Systems: Stability Theory and Applications*. Springer-Verlag, 1967.
- [34] R. Bhatia. *Positive Definite Matrices*. Princeton Series in Applied Mathematics. Princeton University Press, 2007.
- [35] L. J. Billera, S. P. Holmes, and K. Vogtmann. Geometry of the space of phylogenetic trees. *Advances in Applied Mathematics*, 27(4):733–767, 2001.
- [36] D. Bogdanowicz and K. Giaro. Matching split distance for unrooted binary phylogenetic trees. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 9(1):150–160, 2012.
- [37] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [38] A. Breitenmoser and A. Martinoli. On combining multi-robot coverage and reciprocal collision avoidance. In *Proceedings of the Twelfth Int. Symp. on Distributed Autonomous Robotic Systems*, 2014.
- [39] K. Bremer. Combinable component consensus. *Cladistics*, 6(4):369–372, 1990.
- [40] R. W. Brockett. *Asymptotic stability and feedback stabilization*. Defense Technical Information Center, 1983.
- [41] R. Brooks and T. Lozano-Perez. A subdivision algorithm in configuration space for findpath with rotation. *Systems, Man and Cybernetics, IEEE Transactions on*, 15(2):224–233, 1985.

- [42] E. K. Brown and W. H. E. Day. A computationally efficient approximation to the nearest neighbor interchange metric. *Journal of Classification*, 1:93–124, 1984.
- [43] D. Bryant. A classification of consensus methods for phylogenetics. *DIMACS series in Discrete Mathematics and Theoretical Computer Science*, 61:163–184, 2003.
- [44] F. Bullo, J. Cortés, and S. Martinez. *Distributed Control of Robotic Networks: A Mathematical Approach to Motion Coordination Algorithms*. Princeton University Press, 2009.
- [45] R. R. Burridge, A. A. Rizzi, and D. E. Koditschek. Sequential composition of dynamically dexterous robot behaviors. *The International Journal of Robotics Research*, 18(6):534–555, 1999.
- [46] J. Canny. *The complexity of robot motion planning*. MIT press, 1988.
- [47] G. Carlsson and F. Mémoli. Characterization, stability and convergence of hierarchical clustering methods. *Journal of Machine Learning Research*, 11:1425–1470, 2010.
- [48] L. Chaimowicz and V. Kumar. Aerial shepherds: Coordination among uavs and swarms of robots. In *Distributed Autonomous Robotic Systems 6*, pages 243–252. Springer Japan, 2007.
- [49] J. Chakerian and S. Holmes. Computational tools for evaluating phylogenetic and hierarchical clustering trees. *Journal of Computational and Graphical Statistics*, 21(3):581–599, 2012.
- [50] R. W. Chaney. Piecewise ck functions in nonsmooth analysis. *Nonlinear Analysis: Theory, Methods & Applications*, 15(7):649–660, 1990.
- [51] B. Chazelle. Approximation and decomposition of shapes. *Algorithmic and Geometric Aspects of Robotics*, 1:145–185, 1987.
- [52] B. Chazelle. The convergence of bird flocking. *Journal of the ACM (JACM)*, 61(4):21, 2014.
- [53] H. Choset and J. Burdick. Sensor-based exploration: The hierarchical generalized Voronoi graph. *The International Journal of Robotics Research*, 19(2):96–125, 2000.
- [54] H. Choset and P. Pignon. Coverage path planning: The boustrophedon cellular decomposition. In *Field and Service Robotics*, pages 203–209. Springer London, 1998.
- [55] 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, Cambridge, MA, 2005.
- [56] D. Conner, H. Choset, and A. Rizzi. Flow-through policies for hybrid controller synthesis applied to fully actuated systems. *Robotics, IEEE Transactions on*, 25(1):136–146, 2009.

- [57] D. Conner, H. Choset, and A. Rizzi. Integrating planning and control for single-bodied wheeled mobile robots. *Autonomous Robots*, 30(3):243–264, 2011.
- [58] C. I. Connolly and R. A. Grupen. The applications of harmonic functions to robotics. *Journal of Robotic Systems*, 10(7):931–946, 1993.
- [59] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition, 2009.
- [60] J. Cortés, S. Martinez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *Robotics and Automation, IEEE Transactions on*, 20(2):243–255, 2004.
- [61] I. Couzin. Collective minds. *Nature*, 445(7129):715, 2007.
- [62] H. S. M. Coxeter and S. L. Greitzer. *Geometry Revisited*, volume 19. Mathematical Association of America, 1996.
- [63] M. C. Crofoot and I. C. Gilby. Cheating monkeys undermine group strength in enemy territory. *Proceedings of the National Academy of Sciences*, 109(2):501–505, 2012.
- [64] K. CulikII and D. Wood. A note on some tree similarity measures. *Information Processing Letters*, 15(1):39–42, 1982.
- [65] 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*, pages 2807–2813, 2012.
- [66] 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*, pages 427–436, 1997.
- [67] W. H. E. Day. Optimal algorithms for comparing trees with labeled leaves. *Journal of Classification*, 2:7–28, 1985.
- [68] M. de Berg, M. van Krefeld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer, 3rd edition, 2008.
- [69] 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*, pages 356–363. MIT Press, 1991.
- [70] 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*, 42(2):229–243, 2006.
- [71] A. Dirafzoon, M. Menhaj, and A. Afshar. Voronoi based coverage control for nonholonomic mobile robots with collision avoidance. In *Control Applications (CCA), IEEE Int. Conf. on*, pages 1755–1760, 2010.

- [72] Q. Du, V. Faber, and M. Gunzburger. Centroidal voronoi tessellations: Applications and algorithms. *SIAM Review*, 41(4):637–676, 1999.
- [73] Q. Du, V. Faber, and M. Gunzburger. Centroidal Voronoi tessellations: Applications and algorithms. *SIAM Review*, 41(4):637–676, 1999.
- [74] Q. Du, M. D. Gunzburger, and L. Ju. Constrained centroidal voronoi tessellations for surfaces. *SIAM Journal on Scientific Computing*, 24(5):1488–1506, 2003.
- [75] H. G. Eggleston. *Convexity*. Cambridge University Press, 1958.
- [76] E. Fadell and L. Neuwirth. Configuration spaces. *Mathematica Scandinavica*, 10: 111–118, 1962.
- [77] E. R. Fadell and S. Y. Husseini. *Geometry and Topology of Configuration Spaces*. Springer, 2001.
- [78] M. Farber. Topological complexity of motion planning. *Discrete and Computational Geometry*, 29(2):211221, 2003.
- [79] B. Faverjon. Obstacle avoidance using an octree in the configuration space of a manipulator. In *Robotics and Automation. Proceedings. 1984 IEEE International Conference on*, volume 1, pages 504–512, 1984.
- [80] J. Felsenstein. *Inferring Phylogenies*. Sinauer Associates, Suderland, USA, 2004.
- [81] A. Feragen, M. Owen, J. Petersen, M. Wille, L. Thomsen, A. Dirksen, and M. Bruijne. Tree-space statistics and approximations for large-scale analysis of anatomical trees. In *Information Processing in Medical Imaging*, volume 7917 of *Lecture Notes in Computer Science*, pages 74–85. Springer Berlin Heidelberg, 2013.
- [82] R. Fierro and F. L. Lewis. Control of a nonholomic mobile robot: Backstepping kinematics into dynamics. *Journal of Robotic Systems*, 14(3):149–163, 1997.
- [83] I. Filippidis and K. Kyriakopoulos. Adjustable navigation functions for unknown sphere worlds. In *Decision and Control and European Control Conference, 2011 50th IEEE Conference on*, pages 4276–4281, 2011.
- [84] R. A. Finkel and J. L. Bentley. Quad trees a data structure for retrieval on composite keys. *Acta Informatica*, 4(1):1–9, 1974.
- [85] S. Fitzpatrick and R. R. Phelps. Differentiability of the metric projection in Hilbert space. *Transactions of the American Mathematical Society*, 270(2):483–501, 1982.
- [86] G. Flierl, D. Grnbaum, S. Levin, and D. Olson. From individuals to aggregations: the interplay between behavior and physics. *Journal of Theoretical Biology*, 196(4): 397–454, 1999.
- [87] B. Fornberg. Generation of finite difference formulas on arbitrarily spaced grids. *Mathematics of Computation*, 51(184):699–706, 1988.

- [88] J. H. Friedman, J. L. Bentley, and R. A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Trans. Math. Softw.*, 3(3):209–226, 1977.
- [89] J. M. Fryxell, A. Mosser, A. R. E. Sinclair, and C. Packer. Group formation stabilizes predator-prey dynamics. *Nature*, 449(7165):1041–1043, 2007.
- [90] S. K. Gazda, R. C. Connor, R. K. Edgar, and F. Cox. A division of labour with role specialization in group-hunting bottlenose dolphins (*Tursiops truncatus*) off Cedar key, Florida. *Proceedings of the Royal Society B: Biological Sciences*, 272(1559):135–140, 2005.
- [91] M. Ghosh and N. M. Amato. Hierarchical distance-based aggregation. Technical Report TR14-006, Texas A&M University, 2014.
- [92] A. Gierer and H. Meinhardt. A theory of biological pattern formation. *Kybernetik*, 12(1):30–39, 1972.
- [93] V. Graciano Santos and L. Chaimowicz. Hierarchical congestion control for robotic swarms. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 4372–4377, 2011.
- [94] A. Graham. *Kronecker Products and Matrix Calculus: With Applications*, volume 108. Horwood Chichester, 1981.
- [95] B. Grünbaum. Is Napoleon’s theorem really Napoleon’s theorem? *The American Mathematical Monthly*, 119(6):495–501, 2012.
- [96] S. Gueron and R. Tessler. The Fermat–Steiner problem. *The American Mathematical Monthly*, 109(5):443–451, 2002.
- [97] E. Guizzo. So, where are my robot servants? *IEEE Spectrum*, 51(6):74–79, 2014.
- [98] M. Hajja, H. Martini, and M. Spirova. On converses of Napoleon’s theorem and a modified shape function. *Beiträge zur Algebra und Geometrie*, 47(2):363–383, 2006.
- [99] M. Hajja, H. Martini, and M. Spirova. New extensions of Napoleon’s theorem to higher dimensions. *Beitr. Algebra Geom*, 49(1):253–264, 2008.
- [100] M. A. Halverson, D. K. Skelly, and A. Caccone. Kin distribution of amphibian larvae in the wild. *Molecular Ecology*, 15(4):1139–1145, 2006.
- [101] H. W. Hamacher and Z. Drezner. *Facility Location: Applications and Theory*. Springer Science & Business Media, 2002.
- [102] 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*, 57(S3):192–202, 2000.
- [103] R. Haralick, S. R. Sternberg, and X. Zhuang. Image analysis using mathematical morphology. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 9(4):532–550, 1987.

- [104] A. Hatcher. *Algebraic Topology*. Cambridge Univ. Press, 2002.
- [105] G. C. Haynes, F. R. Cohen, and D. E. Koditschek. Gait transitions for quasi-static hexapedal locomotion on level ground. In C. Pradalier, R. Siegwart, and G. Hirzinger, editors, *Robotics Research: The 14th International Symposium (ISRR)*, pages 105–121. Springer Berlin Heidelberg, 2011.
- [106] P. Henry, C. Vollmer, B. Ferris, and D. Fox. Learning to navigate through crowded environments. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 981–986, 2010.
- [107] M. Herman. Fast, three-dimensional, collision-free motion planning. In *Robotics and Automation (ICRA), IEEE International Conference on*, volume 3, pages 1056–1063, 1986.
- [108] M. W. Hirsch, S. Smale, and R. L. Devaney. *Differential Equations, Dynamical Systems, and an Introduction to Chaos*. Academic press, 2nd edition, 2003.
- [109] R. B. Holmes. Smoothness of certain metric projections on Hilbert space. *Transactions of the American Mathematical Society*, 184:87–100, 1973.
- [110] H. Honda. Description of cellular patterns by dirichlet domains: The two-dimensional case. *Journal of Theoretical Biology*, 72(3):523–543, 1978.
- [111] 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*, 3(4):76–88, 1984.
- [112] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 2012.
- [113] 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*, pages 1652–1659, 2014.
- [114] A. Jadbabaie, J. Lin, and A. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *Automatic Control, IEEE Transactions on*, 48(6):988–1001, 2003.
- [115] A. K. Jain and R. C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.
- [116] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys (CSUR)*, 31(3):264–323, 1999.
- [117] F. James Rohlf. Consensus indices for comparing classifications. *Mathematical Biosciences*, 59(1):131–144, 1982.
- [118] A. M. Johnson and D. E. Koditschek. Toward a vocabulary of legged leaping. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 2568–2575, 2013.

- [119] A. M. Johnson, M. T. Hale, G. C. Haynes, and D. E. Koditschek. Autonomous legged hill and stairwell ascent. In *Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on*, pages 134–142, 2011.
- [120] S. Kambhampati and L. Davis. Multiresolution path planning for mobile robots. *IEEE Journal on Robotics and Automation*, 2(3):135–145, 1986.
- [121] Y. Kantaros, M. Thanou, and A. Tzes. Visibility-oriented coverage control of mobile robotic networks on non-convex regions. In *Robotics and Automation, IEEE International Conference on*, pages 1126–1131, 2014.
- [122] Y. Kantaros, M. Thanou, and A. Tzes. Distributed coverage control for concave areas by a heterogeneous robotswarm with visibility sensing constraints. *Automatica*, 53: 195–207, 2015.
- [123] C. S. Karagoz, H. I. Bozma, and D. E. Koditschek. Coordinated navigation of multiple independent disk-shaped robots. *Robotics, IEEE Transactions on*, 30(6):1289–1304, 2014.
- [124] S. Karaman and E. Frazzoli. High-speed flight in an ergodic forest. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 2899–2906, 2012.
- [125] P. Kareiva and G. Odell. Swarms of predators exhibit "preytaxis" if individual predators use area-restricted search. *The American Naturalist*, 130(2):233–270, 1987.
- [126] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *Robotics and Automation, IEEE Transactions on*, 12(4):566–580, 1996.
- [127] H. K. Khalil. *Nonlinear Systems*. Prentice Hall, 3rd edition, 2001.
- [128] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, 5(1):90–98, 1986.
- [129] D. Koditschek. Exact robot navigation by means of potential functions: Some topological considerations. In *Robotics and Automation, 1987 IEEE International Conference on*, volume 4, pages 1–6, 1987.
- [130] D. E. Koditschek. Adaptive techniques for mechanical systems. In *Proc. 5th. Yale Workshop on Adaptive Systems*, pages 259–265, 1987.
- [131] D. E. Koditschek. Some applications of natural motion control. *Journal of Dynamic Systems, Measurement, and Control*, 113:552–557, 1991.
- [132] D. E. Koditschek and E. Rimon. Robot navigation functions on manifolds with boundary. *Advances in Applied Mathematics*, 11(4):412–442, 1990.
- [133] M. Kozlov, S. Tarasov, and L. Khachiyan. The polynomial solvability of convex quadratic programming. *USSR Computational Mathematics and Mathematical Physics*, 20(5):223–228, 1980.

- [134] M. Kumar, D. Garg, and V. Kumar. Self-sorting in a swarm of hetero-geneous agents. In *American Control Conference*, pages 117–122, 2008.
- [135] L. Kuntz and S. Scholtes. Structural analysis of nonsmooth mappings, inverse functions, and metric projections. *Journal of Mathematical Analysis and Applications*, 188(2):346–386, 1994.
- [136] Y. S. Kupitz, H. Martini, and M. Spirova. The Fermat–Torricelli problem, part i: A discrete gradient-method approach. *Journal of Optimization Theory and Applications*, 158(2):305–327, 2013.
- [137] A. Kwok and S. Martnez. Deployment algorithms for a power-constrained mobile sensor network. *International Journal of Robust and Nonlinear Control*, 20(7):745–763, 2010.
- [138] J. P. LaSalle. *The Stability of Dynamical Systems*. Society for Industrial Mathematics, 1976.
- [139] S. M. Lavalle. Rapidly-exploring random trees: A new tool for path planning. Technical report, Iowa State University, 1998.
- [140] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, UK, 2006.
- [141] S. M. LaValle and J. J. Kuffner. Rapidly-exploring random trees: Progress and prospects. In *Algorithmic and Computational Robotics: New Directions*, pages 293–308, 2000.
- [142] M. Li, J. Tromp, and L. Zhang. On the nearest neighbour interchange distance between evolutionary trees. *Journal of Theoretical Biology*, pages 463–467, 1996.
- [143] Y. Lin, V. Rajan, and B. Moret. A metric for phylogenetic trees based on matching. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 9(4):1014–1022, 2012.
- [144] S. R. Lindemann and S. M. LaValle. Current issues in sampling-based motion planning. In *Robotics Research. The Eleventh International Symposium*, pages 36–54. Springer Berlin Heidelberg, 2005.
- [145] G. Lionis, X. Papageorgiou, and K. Kyriakopoulos. Locally computable navigation functions for sphere worlds. In *Robotics and Automation (ICRA), 2007 IEEE International Conference on*, pages 1998–2003, 2007.
- [146] J. Liu. Sensitivity analysis in nonlinear programs and variational inequalities via continuous selections. *SIAM Journal on Control and Optimization*, 33(4):1040–1060, 1995.
- [147] 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*, pages 1427–1432, 1989.

- [148] 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*, 8(1): 39–54, 1991.
- [149] S. Lloyd. Least squares quantization in PCM. *Information Theory, IEEE Transactions on*, 28(2):129–137, 1982.
- [150] S. Loizou. The multi-agent navigation transformation: Tuning-free multi-robot navigation. In *Proceedings of Robotics: Science and Systems (RSS)*, 2014.
- [151] S. G. Loizou. Closed form navigation functions based on harmonic potentials. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 6361–6366, 2011.
- [152] T. Lozano-Perez. Spatial planning: A configuration space approach. *Computers, IEEE Transactions on*, C-32(2):108–120, 1983.
- [153] T. Lozano-Perez, M. T. Mason, and R. H. Taylor. Automatic synthesis of fine-motion strategies for robots. *The International Journal of Robotics Research*, 3(1):3–24, 1984.
- [154] T. Margush and F. R. McMorris. Consensus n-trees. *Bulletin of Mathematical Biology*, 43(2):239–244, 1981.
- [155] H. Martini. On the theorem of Napoleon and related topics. *Mathematische Semesterberichte*, 43(1):47–64, 1996.
- [156] H. Martini and B. Weissbach. Napoleon’s theorem with weights in n-space. *Geometriae Dedicata*, 74(2):213–223, 1999.
- [157] J. McMahon and V. Snyder. *Elements of the Differential Calculus*. American Book Company, 1898.
- [158] D. Meagher. Geometric modeling using octree encoding. *Computer Graphics and Image Processing*, 19(2):129–147, 1982.
- [159] B. Mirkin. *Mathematical Classification and Clustering*. Kluwer Academic Publishers, 1996.
- [160] 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*, 38(3):423–457, 1973.
- [161] J. Munkres. *Topology*. Pearson, 2nd edition, 2000.
- [162] B. Nabet, N. E. Leonard, I. D. Couzin, and S. A. Levin. Dynamics of decision making in animal group motion. *Journal of Nonlinear Science*, 19(4):399–435, 2009.
- [163] L. A. Nituch, J. A. Schaefer, and C. D. Maxwell. Fine-scale spatial organization reflects genetic structure in sheep. *Ethology*, 114(7):711–717, 2008.

- [164] D. O'Brien. Analysis of the internal arrangement of individuals within crustacean aggregations (Euphausiacea, Mysidacea). *Journal of Experimental Marine Biology and Ecology*, 128(1):1–30, 1989.
- [165] C. Ó'Dúnlaing and C. K. Yap. A “retraction” method for planning the motion of a disc. *Journal of Algorithms*, 6(1):104–111, 1985.
- [166] P. Ogren. Split and join of vehicle formations doing obstacle avoidance. In *Robotics and Automation (ICRA), 2004 IEEE International Conference on*, pages 1951–1955, 2004.
- [167] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Wiley Series in Probability and Statistics. John Wiley & Sons, New York, 2nd edition, 2000.
- [168] A. Okubo and S. A. Levin. *Diffusion and Ecological Problems: Modern Perspectives*. Springer Science & Business Media, 2001.
- [169] M. Owen and J. S. Provan. A fast algorithm for computing geodesic distances in tree space. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 8(1):2–13, 2011.
- [170] A. A. Paranjape, K. C. Meier, X. Shi, S.-J. Chung, and S. Hutchinson. Motion primitives and 3D path planning for fast flight through a forest. *The International Journal of Robotics Research*, 34(3):357–377, 2015.
- [171] L. E. Parker. Multiple mobile robot systems. In B. Siciliano and O. Khatib, editors, *Springer Handbook of Robotics*, pages 921–941. Springer Berlin Heidelberg, 2008.
- [172] L. E. Parker. Path planning and motion coordination in multiple mobile robot teams. In *Encyclopedia of Complexity and Systems Science*, pages 5783–5800. Springer New York, 2009.
- [173] J. K. Parrish and L. Edelstein-Keshet. Complexity, pattern, and evolutionary trade-offs in animal aggregation. *Science*, 284(5411):99–101, 1999.
- [174] S. Paternain, D. E. Koditschek, and A. Ribeiro. Navigation functions for convex potentials in a space with convex obstacles. *Automatic Control, IEEE Transactions on*, (submitted).
- [175] M. Peasgood, C. Clark, and J. McPhee. A complete and scalable strategy for coordinating multiple robots within roadmaps. *Robotics, IEEE Transactions on*, 24(2):283–292, April 2008.
- [176] K. B. Petersen and M. S. Pedersen. *The Matrix Cookbook*. Technical University of Denmark, 2012.
- [177] A. Pierson, L. C. Figueiredo, L. C. A. Pimenta, and M. Schwager. Adapting to performance variations in multi-robot coverage. In *Robotics and Automation (ICRA), IEEE International Conference on*, 2015.

- [178] L. C. Pimenta, V. Kumar, R. C. Mesquita, and G. A. Pereira. Sensing and coverage for a network of heterogeneous robots. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 3947–3952, 2008.
- [179] L. C. Pimenta, M. Schwager, Q. Lindsey, V. Kumar, D. Rus, R. Mesquita, and G. Pereira. Simultaneous coverage and tracking (scat) of moving targets with robot networks. In G. S. Chirikjian, H. Choset, M. Morales, and T. Murphey, editors, *Algorithmic Foundation of Robotics VIII: Selected Contributions of the Eight International Workshop on the Algorithmic Foundations of Robotics*, volume 57, pages 85–99. Springer Berlin Heidelberg, 2010.
- [180] R. Rammal, G. Toulouse, and M. A. Virasoro. Ultrametricity for physicists. *Reviews of Modern Physics*, 58(3):765, 1986.
- [181] J. A. Rice. *Mathematical Statistics and Data Analysis*. Cengage Learning, 2007.
- [182] J. Rigby. Napoleon, Escher, and tessellations. *Mathematics Magazine*, 64(4):242–246, 1991.
- [183] E. Rimon and D. Koditschek. Exact robot navigation using artificial potential functions. *Robotics and Automation, IEEE Transactions on*, 8(5):501–518, 1992.
- [184] D. Robinson. Comparison of labeled trees with valency three. *Journal of Combinatorial Theory, Series B*, 11(2):105–119, 1971.
- [185] D. F. Robinson and L. R. Foulds. Comparison of phylogenetic trees. *Mathematical Biosciences*, 53(1-2):131–147, 1981.
- [186] R. Rockafellar. Lipschitzian properties of multifunctions. *Nonlinear Analysis: Theory, Methods & Applications*, 9(8):867–885, 1985.
- [187] K. Rosen. *Discrete Mathematics and Its Applications*. McGraw-Hill, 7th edition, 2011.
- [188] P. E. Ross. Robot, you can drive my car. *IEEE Spectrum*, 51(6):60–90, 2014.
- [189] G.-C. Rota. The number of partitions of a set. *The American Mathematical Monthly*, 71(5):498–504, 1964.
- [190] 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*, pages 1112–1117, 2014.
- [191] V. G. Santos and L. Chaimowicz. Cohesion and segregation in swarm navigation. *Robotica*, 32(02):209–223, 2014.
- [192] U. Saranli, M. Buehler, and D. E. Koditschek. Rhex: A simple and highly mobile hexapod robot. *The International Journal of Robotics Research*, 20(7):616–631, 2001.
- [193] 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)*, pages 1–14, 2001.

- [194] D. Scheel and C. Packer. Group hunting behaviour of lions: a search for cooperation. *Animal Behaviour*, 41(4):697–709, 1991.
- [195] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*, volume 24. Springer, 2003.
- [196] M. Schwager, D. Rus, and J.-J. Slotine. Unifying geometric, probabilistic, and potential field approaches to multi-robot deployment. *Int. J. Robotics Research*, 30(3): 371–383, 2011.
- [197] C. Semple and M. Steel. *Phylogenetics*, volume 24. Oxford University Press, 2003.
- [198] R. Sepulchre, D. A. Paley, and N. E. Leonard. Stabilization of planar collective motion: All-to-all communication. *IEEE Transactions on Automatic Control*, 52(5): 811–824, 2007.
- [199] A. Shapiro. Sensitivity analysis of nonlinear programs and differentiability properties of metric projections. *SIAM Journal on Control and Optimization*, 26(3):628–645, 1988.
- [200] G. E. Shilov. *Elementary Real and Complex Analysis*. Courier Corporation, 1996.
- [201] D. D. Sleator, R. E. Tarjan, and W. P. Thurston. Rotation distance, triangulations, and hyperbolic geometry. In *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing (STOC '86)*, pages 122–135, 1986.
- [202] K. Solovey and D. Halperin. k-color multi-robot motion planning. *The International Journal of Robotics Research*, 33(1):82–97, 2014.
- [203] P. Spirakis and C. K. Yap. Strong NP-hardness of moving many discs. *Information Processing Letters*, 19(1):55–59, 1984.
- [204] P. E. Stander. Cooperative hunting in lions: the role of the individual. *Behavioral Ecology and Sociobiology*, 29(6):445–454, 1992.
- [205] M. S. Steinberg. Reconstruction of tissues by dissociated cells. *Science*, 141(3579): pp. 401–408, 1963.
- [206] J. Stewart. *Calculus: Early Transcendentals*. Cengage Learning, 7th edition, 2012.
- [207] C. Strbin, M. Steinegger, and R. Bshary. On group living and collaborative hunting in the yellow saddle goatfish (*parupeneus cyclostomus*)1. *Ethology*, 117(11):961–969, 2011.
- [208] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 2011.
- [209] H. Tanner and A. Boddu. Multiagent navigation functions revisited. *Robotics, IEEE Transactions on*, 28(6):1346–1359, 2012.
- [210] A. Tapus, M. J. Mataric, and B. Scassellati. Socially assistive robotics [grand challenges of robotics]. *IEEE Robotics Automation Magazine*, 14(1):35–42, 2007.

- [211] J. H. Tien, S. A. Levin, and D. I. Rubenstein. Dynamics of fish shoals: identifying key decision rules. *Evolutionary Ecology Research*, 6(4):555–565, 2004.
- [212] P. Trautman, J. Ma, R. M. Murray, and A. Krause. Robot navigation in dense human crowds: Statistical models and experimental studies of humanrobot cooperation. *The International Journal of Robotics Research*, 34(3):335–356, 2015.
- [213] A. M. Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 237(641):37–72, 1952.
- [214] M. Turpin, N. Michael, and V. Kumar. Trajectory planning and assignment in multi-robot systems. In E. Frazzoli, T. Lozano-Perez, N. Roy, and D. Rus, editors, *Algorithmic Foundations of Robotics X: Proceedings of the Tenth Workshop on the Algorithmic Foundations of Robotics*, volume 86, pages 175–190. Springer Berlin Heidelberg, 2013.
- [215] 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*, pages 842–848, 2013.
- [216] K. Vogtmann. Geodesics in the space of trees., 2007. URL www.math.cornell.edu/~vogtmann/papers/TreeGeodesicss. Accessed May 22, 2015.
- [217] M. Waterman and T. Smith. On the similarity of dendrograms. *Journal of Theoretical Biology*, 73(4):789 – 800, 1978.
- [218] D. Watts and J. Mitani. Hunting behavior of chimpanzees at Ngogo, Kibale National Park, Uganda. *International Journal of Primatology*, 23(1):1–28, 2002.
- [219] R. Webster. *Convexity*. Oxford University Press, 1995.
- [220] J. E. Wetzel. Converses of Napoleon’s theorem. *The American Mathematical Monthly*, 99(4):339–351, 1992.
- [221] 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*, pages 17–23, 1991.
- [222] 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*, pages 2186–2191, 1992.
- [223] I. H. Witten, E. Frank, and M. A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers, 2011.
- [224] D. Wooden, M. Malchano, K. Blankespoor, A. Howardy, A. A. Rizzi, and M. Raibert. Autonomous navigation for BigDog. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 4736–4741, 2010.

- [225] A. Yahja, A. Stentz, S. Singh, and B. L. Brumitt. Framed-quadtrees path planning for mobile robots operating in sparse environments. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 1, pages 650–655, 1998.
- [226] A. Yershova and S. M. LaValle. Improving motion-planning algorithms by efficient nearest-neighbor searching. *IEEE Transactions on Robotics*, 23(1):151–157, 2007.
- [227] A. Zelinsky. A mobile robot exploration algorithm. *Robotics and Automation, IEEE Transactions on*, 8(6):707–717, 1992.