CHARACTERIZING AND FACILITATING HUMAN INTERACTIONS WITH SWARMS OF MOBILE ROBOTS

A Dissertation
Presented to
The Academic Faculty

by

Jean-Pierre de la Croix

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

School of Electrical and Computer Engineering
Georgia Institute of Technology
May 2015

CHARACTERIZING AND FACILITATING HUMAN INTERACTIONS WITH SWARMS OF MOBILE ROBOTS

Approved by:

Dr. Magnus Egerstedt, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Ayanna M. Howard
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Bonnie H. Ferri
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Andrea L. Thomaz
School of Interactive Computing
Georgia Institute of Technology

Dr. Eric Feron
School of Aerospace Engineering
Georgia Institute of Technology

Date Approved: 5 February 2015

To my Lauren for her love, support, and everything else.

ACKNOWLEDGEMENTS

First of all, I would like to acknowledge my advisor, Dr. Magnus Egerstedt, for his enthusiastic support over the years. This adventure started with an opportunity for a young undergraduate to make robots move in his lab, and it culminated in this dissertation seven years later. I am very thankful for the environment he fosters in his lab, which has allowed me to grow by leaps and bounds in many new directions. I would also like to thank Dr. Ayanna Howard and Dr. Bonnie Ferri for providing me with valuable feedback to shape the story of this dissertation, and Dr. Andrea Thomaz and Dr. Eric Feron for being part of my Ph.D. committee.

I owe a debt of gratitude to many members and alumni of the Georgia Robotics and InTelligent Systems (GRITS) lab for their support and friendship. I have thoroughly enjoyed our technical collaborations, helpful discussions, and camaraderie over the years. I will truly miss all of these exceptional individuals.

Finally, I want to acknowledge the unwavering love and support of my family. Thank you to my parents for encouraging me to always take on new and challenging adventures, and my parents-in-law for believing that I can make it through each adventure with great success. Most of all, thank you to my wife, Lauren, for everything. We've braved this adventure together, and now it's on to the next!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Many applications require human intervention to guide autonomous robots through complicated tasks. For example, we often rely on and benefit from a human operator's ability to decide where robots should focus their efforts in a search and rescue operation, or to instruct where and how they should position equipment on a manufacturing floor. Even when autonomous robots do not require human guidance, humans and robots continue to share workspaces and interact with each other, as is the case in automated warehouses and assembly lines. Much of the focus has been centered on supporting human interactions with one or a few robots (i.e., human-robot interaction, or simply HRI); however, as the number of robots involved in a task grows large, scalable abstractions are needed to support interactions with larger numbers of robots. Consequently, there has been a growing effort to understand human-*swarm* interactions (HSI) and devise abstractions that are amenable to having humans interact with swarms of robots easily and effectively. In this dissertation, we investigate what it means to impose a control structure on a swarm of robots for the purpose of supporting a specific HSI, when such a control structure is suitable for allowing a user to solve a particular task with a swarm of robots, how one can evaluate attention and effort required to interact with a swarm of robots through a particular control structure, how well attention and effort scale as the number of robots in the swarm increases, why some swarms of robots are easier to interact with than others under the same control structure, how to select an appropriate swarm size, and how to design new input controllers for interacting with swarm of mobile robots. Consequently, this dissertation provides a comprehensive framework for characterizing, understanding, and designing the control structures of new abstractions that will be amenable to humans interacting with swarms of networked mobile robots, as well as, a number of examples of such old and new abstractions investigated under this framework.

CHAPTER I

INTRODUCTION AND BACKGROUND

1.1 Introduction

Many applications require human intervention to guide autonomous robots through complicated tasks. For example, we often rely on and benefit from a human operator's ability to decide where robots should focus their efforts in a search and rescue operation [4, 20], or to instruct where and how they should position equipment on a manufacturing floor [48]. Even when autonomous robots do not require human guidance, humans and robots continue to share workspaces and interact with each other, as is the case in automated warehouses and assembly lines. Much of the focus has been centered on supporting human interactions with one or a few robots (i.e., human-robot interaction, or simply HRI); however, as the number of robots involved in a task grows large, scalable abstractions are needed to support interactions with larger numbers of robots [43]. Consequently, there has been a growing effort to understand human-*swarm* interactions (HSI) and devise abstractions that are amenable to having humans interact with swarms of robots easily and effectively.

Much of the recent work on HSI has focused on *developing* new abstractions for enabling useful interactions with a swarm of robots. For example, some of the novel abstractions that have been developed include articulating gestures [71], composing music [21], stirring fluids [57], manipulating densities [61], and our own: molding clay [33] and broadcasting instructions [29]. However, this beckons the question, *why is it that these abstractions provide a control structure amenable to interacting with swarms of robots?* We answer this question in this dissertation by discussing the following:

- a formal definition of the underlying control structures that support HSI abstractions,
- a tool for demonstrating that a control structure facilitates completing a geometric task with a swarm of robots,

- a set of tools for characterizing attention, effort, scalability, and other factors involved in interacting with a swarm of robots through a particular control structure,
- and a number of examples of old and new abstractions evaluated with this set of tools.

Consequently, this dissertation provides a comprehensive framework for *understanding*, *characterizing*, and *designing* the control structures of HSI abstractions (such as our own new abstractions) that are amenable to humans interacting with swarms of mobile robots. We support this framework with discussions around three central topics.

1.1.1 Control Structures for Interacting with Swarms of Robots

Suppose a user is required to guide a mobile robot to a specific stack of packages in a warehouse, such that the robot can sort these packages for an expedited order. A reasonable way for the user to interact with this robot is to drive it with a joystick to the stack of packages. This HRI is simple and the task can easily be completed, but suppose that a collection of ten smaller robots can sort all packages ten times quicker. The user could individually guide each robot as before, but with ten robots this interaction becomes a tedious task. Instead, suppose that we let the robots autonomously coordinate to meet up with nearby robots, and provide the user with a single joystick to guide one of the robots in the swarm to the stack of packages. This deliberate combination of an autonomous cooperative behavior and a simple input controller creates an HSI that is amenable to a user guiding this small swarm of ten robots to the stack of packages.

The package sorting example demonstrates that HSIs are unique compared to HRIs not only due to the number of robots involved in the interaction, but also, because the user's interaction with the swarm of robots occurs concurrently with the interactions between the robots in the swarm. Specifically, this example illustrates a type of control structure called the *single-leader network*, in which the user interacts with the swarm of robots through a single designated leader robot. Suppose that these package sorting robots are single integrators, such that the position of any robot, $x_i(t)$, can be directly controlled, i.e., $\dot{x}_i(t) = u_i(t)$. Let the control input of each follower robot be a function of the distance to other nearby robots (neighbors),

$$u_i(t) = \sum_{j \in N(i)} (x_j(t) - x_i(t)), \tag{1}$$

where $N(i)$ is the set of all neighbors of robot i, while the position of the leader robot, $x_l(t)$, is directly controlled by the user, i.e., $\dot{x}_l(t) = v(t)$. This control structure purposefully reduces the complexity of interacting with the swarm down to controlling a single robot. In Chapter 2, we discuss how these types of control structures can be formally defined from a multi-agent systems perspective in terms of state feedback (i.e., local, inter-robot interactions) and a set of admissible inputs from the user. This definition is pivotal in not only understanding how an HSI is structured, but in constructing characterization tools for such HSIs.

1.1.2 Characterization Tools for Human-Swarm Interactions

Can the user complete the task with the swarm? and *Is this a usable interaction structure?* are two fundamental questions that we would like to answer for a human-swarm interaction. The former question can be answered objectively given the control structure of the HSI and the set of admissible user inputs. Specifically, we discuss in Section 2.2.1 how to use control Lyapunov functions (CLFs) to show that a particular control structure facilitates a user guiding a swarm of mobile robots to some desired specification set (i.e., a set of geometric configurations that the swarm should achieve). If the user can complete the task with the swarm, then the latter question takes on a subjective tone: *Was it easy to complete the task with this swarm? How much did you have to concentrate on the task? Were you frustrated?* These questions can be answered by a user study, which we have done for single-leader networks to tie the answer to these subjective questions to properties of the multi-agent system as is discussed in Chapter 3. However, we do not want to solely rely on running a study for every HSI. In Section 2.3, we discuss tools for objectively characterizing the attention, effort, and scalability of interacting with a swarm of robots through a particular HSI control structure. These characterization tools presented in Section 2.3 and the results of the user study presented in Chapter 3 allow us to make concrete recommendations on how to design HSI control structures.

1.1.3 New Abstraction for Interacting with Swarms of Robots

Given the tools to characterize HSI control structures, we are capable of designing and developing new abstractions for interacting with swarms of mobile robots. In Chapter 4, we discuss three novel abstractions: broadcast control, biologically-inspired parameterized interactions, and manipulating deformable media. These abstractions allow a user to interact with a swarm of mobile robots to achieve specific geometric configurations. For example, a deformable medium, such as clay, can be molded into a shape that the swarm should form, or a single broadcast signal can separate two types of robots into separate clusters. Our discussion partly focuses on demonstrating that these abstractions allow a user to complete these specific task with a swarm, and the advantages and disadvantages of some of these abstractions with respect to the characterization tools discussed in Chapters 2 and 3. We also focus our discussion on other important HSI questions, such as *How many robots are needed for a particular interaction to succeed?* and *Is there an input controller that affords guiding a swarm of robots into specific shapes?*

1.2 Background Research

We described a package sorting example in Section 1.1.1, where a user is interested in guiding a swarm of ten mobile robots to a stack of packages in a warehouse for sorting. Suppose that we program the robots to autonomously meet up with their neighbors, and provide the user with a single joystick to guide one of the robots in the swarm to the stack of packages. This so-called *single-leader network* (as described in detail in [79, 81]) is a control structure that is likely amenable to guiding the small swarm of ten robots to the stack of packages as is illustrated in Figure 1. In this section, we will use the single-leader network and this example of a human-swarm interaction to introduce the mathematical language required to discuss such HSIs from a control and graph theory perspective. Graph and control theory are instrumental in modeling the interactions between robots in the swarm, defining the control structure imposed on a swarm, and most importantly, characterizing the factors contributing to an amenable HSI control structure.

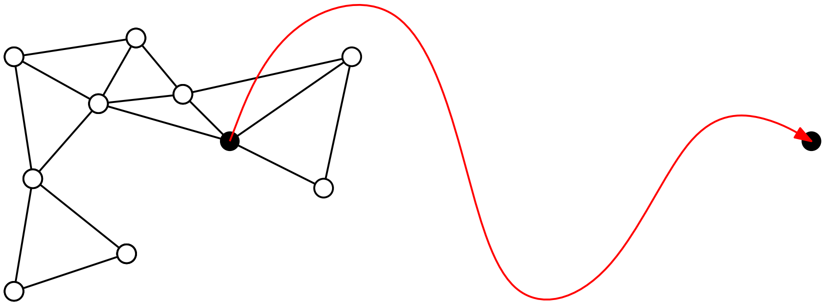


Figure 1: A user guides a single-leader network to a specific rendezvous location from its initial configuration. The solid red line is the path along which the user guides the leader. All robots share a common location in the final configuration.

1.2.1 Graphs, the Graph Laplacian, and Control Structures

Interactions between robots in a swarm typically occur in two different ways: robots communicate to each other about their states (via acoustic, optical, or radio channels) or sense each other's physical states (via tactile, infrared, laser, ultrasonic, or vision sensors). Graphs are an encoding of these interactions, where each robot is a vertex and each interaction with another robot is an edge. These edges may be directed if the interaction is one sided, or undirected if both robots partake in the interaction. In Figure 1, each circular object is a vertex (robot) and each straight line between two circles is an undirected edge (interaction). Encoding the interactions of robots in a swarm allows us, for example, to create other mathematical objects (e.g., the graph Laplacian) to model the dynamics of the swarm and characterize the importance of vertices (i.e., node centrality). Consequently, understanding how to model the interaction topology of the swarm with a graph is a prerequisite to defining and characterizing the control structure of an HSI.

1.2.1.1 Graphs

A graph can be formally defined as $\mathcal{G} = (V, E)$, where V is the set of vertices (or nodes), and $V \times V \subseteq E$ is the set of edges between vertices. Each edge between two vertices models an interaction between two robots, meaning that if the robots are sensing each other or communicating over a network, then the graph will contain an edge between the two vertices representing the robots. Consequently, if robot $i \in V$ and robot $j \in V$ are interacting, then there exists an edge $(i, j) \in E$. The set of robots that share an edge with robot i is the so-called neighborhood,

$$N(i) = \{j \in V \mid (i, j) \in E\}, \tag{2}$$

which is implicitly a function of a graph $\mathcal{G} = (V, E)$. If an edge $(i, j) \in E$ is undirected, then $j \in N(i)$ and $i \in N(j)$. If this edge is directed, then $j \in N(i)$, but $i \notin N(j)$.

Graphs can be static or dynamic. For example, an edge $(i, j) \in E$ in a Δ-disk graph exists if and only if $\|x_j - x_i\| \leq \Delta$, $\Delta \in \mathbf{R}^+$, where x_i and x_j are the positions of robot i and j and Δ represents a maximum interaction distance. Figure 2 illustrates a Δ-disk around robot i, which includes robots j and k, but does not include robot l; therefore, $N(i) = \{j, k\}$. Since mobile robots frequently change their positions, a particular edge $(i, j) \in E$ exists whenever robots i and j are close to each other, but is removed from the graph when the robots move too far apart to interact. Consequently, the set of edges and neighborhoods in a Δ-disk graph change. The vertex set of a graph may also change over time if robots are added or removed from the swarm. Graphs with vertex or edge sets that are not constant are dynamic. Dynamic graphs, such as the Δ-disk graph, are typically used to model swarms, because they capture dynamic nature of interactions between robots in a swarm induced by mobility and limited communication and sensing ranges. However, some situations are not bound by these limitations (e.g., a swarm of robots in a small space) or it could be advantageous to impose a static network topology on the swarm. For example, we may insist on a static graph in the package sorting example, such that the interaction topology of the swarm does not change while the user is interacting with this single-leader network. Consequently, a static graph models an interaction topology with a constant edge set, i.e., a robot will always interact with the same set of robots.

1.2.1.2 Graph Laplacian

The graph is essential in modeling the dynamics of the robots in the swarm. For example, if we suppose that the package sorting robots are single integrators, $\dot{x}_i(t) = u_i(t)$, where $x_i(t) \in \mathbf{R}$ (without loss of generality) is the position of robot i at time t and $u_i(t) \in \mathbf{R}$ is the control input to robot i at time t, then the so-called consensus dynamics (introduced

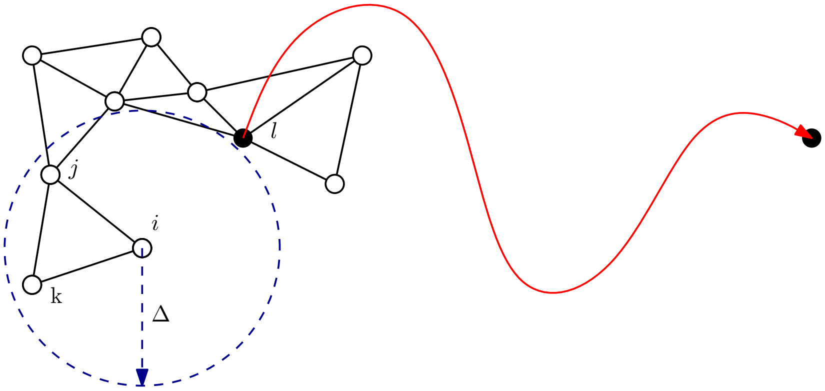


Figure 2: Since the Δ-disk around robot i includes robots j and k, $(i,j) \in E$ and $(i,k) \in E$, but the Δ-disk does not include robot l, so $(i,l) \notin E$.

in [68]) are

$$\dot{x}_i(t) = u_i(t) = \sum_{j \in N(i)} \left(x_j(t) - x_i(t)\right), \tag{3}$$

where the change in position of a robot is implicitly a function of the graph. The consensus dynamics model a robot i moving towards the centroid of the closure of its *neighborhood*, i.e. the average position of itself and its neighbors. If we expand the sum in Equation 3, then

$$x_i(t) = \sum_{j \in N(i)} x_j(t) - \deg(i) x_i(t), \tag{4}$$

where $\deg(i)$ is equal to $|N(i)|$, i.e., the number of robots in the neighborhood of robot i. Let us define $\Delta(\mathcal{G})$ to be a $n \times n$ diagonal matrix ($n = 10$ in the package sorting example), where

$$\Delta(\mathcal{G})_{ii} = \deg(i), \tag{5}$$

which is the so-called *degree matrix*. Another symmetric $n \times n$ matrix is the so-called *adjacency matrix*, $A(\mathcal{G})$, where

$$A(\mathcal{G})_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

If we stack all $x_i(t)$ into a vector $x(t)$, then it follows from Equation 4 and the definitions of the degree and adjacency matrices that

$$\begin{aligned}\dot{x}(t) &= (A(\mathcal{G}) - \Delta(\mathcal{G}))x(t) \\ &= -L(\mathcal{G})x(t),\end{aligned} \tag{7}$$

where $L(\mathcal{G}) = \Delta(\mathcal{G}) - A(\mathcal{G})$ is the so-called *graph Laplacian.* This definition of the graph Laplacian makes the implicit assumption that the graph $\mathcal{G}$ is undirected. The graph Laplacian can also be defined for undirected graphs, such that $L(\mathcal{G}) = D(\mathcal{G})D(\mathcal{G})^T$, where $D(\mathcal{G})$ is the $m \times n$ *incidence matrix* $(m = |E|, n = |V|)$.

It is known (see for example, [65]) that for any undirected, connected graph $\mathcal{G}$, Equation 7 asymptotically converges to

$$x(t) \to \frac{\mathbf{1}^T x(t_0)}{n}\mathbf{1} \text{ as } t \to \infty, \tag{8}$$

which is the centroid of the robots' initial position. In general, it can be shown that the consensus dynamics applied to a connected graph (undirected or directed with a rooted outward branching tree) will converge to span$\{\mathbf{1}\}$, which is precisely the null space of the graph Laplacian.

Suppose a user input, $v(t)$, is applied to the n-th robot in addition to Equation 3, then the multi-agent system (swarm) dynamics are

$$\dot{x}(t) = u(t) = -Lx(t) + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} v(t), \tag{9}$$

where L is the aforementioned graph Laplacian matrix. We show in Chapter 2 that this single-leader network converges to the position of the n-th robot, the leader. In general, it is known (see, for example, [19]) that in multi-leader networks, the follower robots will converge to a point in the convex hull of the leaders.

We can construct the controllability matrix (or Gramian, see [55] for a standard definition) from the state transition and input matrices in Equation 9 to evaluate if a user can guide the swarm from its initial position to a desired final position. Similarly, we discuss in

Section 2.2.1 how to use control Lyapunov functions (introduced in [78]) to show if a user can guide the swarm from its initial position to some desired specification set, and we provide examples where $L(\mathcal{G})$ is essential to this analysis. Consequently, the graph Laplacian is a useful mathematical object to describe the consensus dynamics of a swarm of robots and demonstrating, for example, if there exists a user input to guide all the robots into some desired geometric configuration.

1.2.1.3 Control Structures

Equation 9 describes a single-leader network, which is one example of a control structure for a swarm of robots. For example, if the dynamics of all robots (again, modeled as single integrators) were

$$x_i(t) = u_i(t) = \sum_{j \in N(i)} (x_j(t) - x_i(t)) + v(t), \tag{10}$$

then $v(t) \in \mathbf{R}$ is a broadcast input signal. The user interacts with the swarm by broadcasting an input signal $v(t)$ to all robots, rather than only interacting with the leader robot. The graph Laplacian can once again be used to described the dynamics of the entire swarm,

$$\dot{x}(t) = u(t) = -Lx(t) + \mathbf{1}v(t), \tag{11}$$

where $\mathbf{1} \in \mathbf{R}^n$ is a vector of all ones. Equation 11 is another example of a control structure used to facilitate the package sorting task as we will demonstrate in Section 2.4.2.

We focus in this dissertation on control structures for multi-agent systems (such as swarms of robots) that are characterized by the kinematics of the robots, the dynamics of their interactions, and an exogenous input signal. Specifically,

1. The kinematics of an individual robot are a function of its own state and the *control input*, $u_i(t)$. For example, single integrator kinematics are modeled as $\dot{x}_i(t) = u_i(t)$, while the kinematics of a unicycle are modeled as

$$\begin{bmatrix} \dot{x}_i(t) \\ \dot{y}_i(t) \\ \dot{\theta}_i(t) \end{bmatrix} = \begin{bmatrix} \cos(\theta_i(t)) & 0 \\ \sin(\theta_i(t)) & 0 \\ 0 & 1 \end{bmatrix} u_i(t), \tag{12}$$

where the two vector components of $u_i(t) \in \mathbf{R}^2$ are equal to the linear and angular velocity of the unicycle at time t, respectively.

2. The control input, $u_i(t)$, for robot i is a function of the stacked state vector $x(t)$ (i.e., the states of all robots in the swarm) and some exogenous user control input $v(t)$. Since the interactions between robots are limited to an interaction topology (e.g., a Δ-disk graph), the control input to robot i, $u_i(t)$, can only be a function of its own state and the state of any of its neighbors.

The distinction between control input, $u(t)$, and some exogenous input, $v(t)$, is common when modeling multi-agent systems, but not always explicitly stated. We discuss in Chapter 2 one formal definition of such control structure for multi-agent systems in the context of HSIs. Specifically, the exogenous input signal $v(t)$ is the *user input* restricted to some set of admissible inputs, $\mathcal{V}$.

These control structures are often created to support multi-robot applications focused on geometric tasks, such as foraging [3], coverage [5], exploration [18], rendezvous [62], surveillance [70], and transport [77]. For example, a control structure can force robots to maintain an equal distance to all nearby robots to form triangulations that cover an area larger than what can be covered by a single robot. While these robots can self-organize into triangulations without any guidance, an HSI control structure permits a user to externally guide these robots to an area of interest by including the exogenous input, $v(t)$. This dissertation focuses on such control structures that are geared towards facilitating a user in achieving a particular geometric configuration with the swarm of robots. For example, we discuss the package sorting example throughout this dissertation, where the user-desired geometric swarm configuration is for all robots to be located together at the rendezvous location, i.e., the stack of packages to be sorted.

1.2.2 Attention, Effort, Scalability, and other Factors in HSIs

The previous section was dedicated to introducing the graph, graph Laplacian, and control structures. These concepts are essential to the discussion of controllability, node centrality, and network centralization as objective metrics of how a swarm of robots will perform under input in Chapters 2 and 3. This section introduces three new metrics, attention, effort, and complexity, to objectively characterize the user's interactions with a swarm of mobile robots through an HSI control structure. However, we are also interested in measuring subjective factors, such as perceived difficulty, frustration, temporal demand, and other self-reported measures that can be captured through a user study. The novelty is that we tie objective and subjective metrics together in Chapter 3 to construct recommendations on how interaction topologies, and HSIs in general, should be designed.

1.2.2.1 Attention, Effort, and Scalability

Since the point of robotics is partly to automate tedious task, a user should naturally not be required to pay a lot of attention or expend a lot of effort to guiding a swarm of robots. At its simplest, one can imagine that the user is only responsible for pressing a button to start or stop the robots, but typically, robots require a little more guidance in tasks that require greater decision making. We want to measure the attention and effort required to guide a swarm of robots into a specific geometric configuration. For example, how much attention is required for the user to guide the package sorting robots to their rendezvous location?

Roger Brockett introduced the notion of a minimum attention controller [15] by solving an optimal control problem that minimized the total variation of the control signal over time as well as over the state of the system. For example, the attention functional would take the form

$$\int_{\mathcal{T}} \int_{\mathcal{X}} \phi\left(x, t, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial t}\right) dx dt, \tag{13}$$

where $\mathcal{X}$ and $\mathcal{T}$ are the state space and time domains over which the controller is defined, and u is the control signal. However, attention is only part of the story. Optimization problems often minimize u itself (see, for example, [16]), meaning that it is desirable to complete some task with minimal effort. Consequently, we propose in Section 2.3 that a cost on the user input should be both in terms of attention and effort. One way to formulate this attention-effort cost for an HSI control structure is

$$\int_0^\infty \frac{1}{2}\left(\|v(t)\|^2 + \|\dot{v}(t)\|^2\right) dt, \tag{14}$$

which encodes a cost on the squared magnitude of the user input v (effort) and the variation in v over time (attention). It should be noted that such a functional computes the attention-effort cost for a particular $v(t)$ and not for a particular HSI control structure. In fact, it becomes evident in Chapter 2 that most, if not all, metrics are a function of the *task*. Tasks are a unique combination of a robotic swarm, HSI control structure, initial condition, and a specification set. For example, a user input, $v(t)$, will typically be specific to a task and not to an HSI control structure. Consequently, we can gain insight into an HSI control structure in general by computing these metrics across a variety of tasks (i.e., variations in initial conditions and specification sets), as we have done in Chapter 3.

If attention and effort are computed for some task with n robots, then we can also compute the attention and effort required to complete a similar task with $(n + 1)$ robots. Computing the attention-effort cost over a range of n can provide insight into the scalability of the HSI control structure with respect to attention and effort. Our notion of scalability is consequently similar to computational complexity for algorithms (see, for example, [23]).

1.2.2.2 Subjective Factors

While attention, effort, and scalability are objective metrics computed either from an optimal input or sampled user input, we can also assess the effectiveness of an HSI control structure with subjective factors. For example, we could ask the user to perform the package sorting task with the single leader network organized over a variety of different interaction topologies. Then, we could inquire about the perceived difficulty of the task, whether the user felt frustrated, and how much attention they felt that they needed to invest in the task. These subjective factors are typically gathered through a user study. User studies are a common tool (see, for example, their use in [6]) to measure these subjective factors by asking a large number of participants (a sample population) to complete tasks and answer a targeted questionnaire afterwards. Most user studies are used to decide whether control structure A is effective, or if control structure B is better, as is explored in [60], but not *why* they are effective. Consequently, we discuss a user study in Chapter 3, where we tied the subjective factors measured in the user study to objective metrics, such as controllability, node centrality, and network centralization. The results of this analysis is an answer to a more profound question, *what characteristics of a particular control structure make it more or less amenable to user input?* Answering this question allowed us to make recommendations in Chapter 2 on which HSI control structure to select for a particular task, or in Chapter 3 how to design interaction topologies that improve interactions with a swarm of robots through an HSI control structure.

1.2.3 Current and New Abstractions for HSIs

Given the discovered connections between attention, effort, scalability, the subjective factors, and the graph and control theoretic properties of the control structures, we are able to make an informed decision on how to design an HSI control structure that will be amenable to users guiding a swarm of robots in a task. It also provides us with tools to reevaluate some of the existing abstractions for HSIs in the new framework presented in this dissertation, as is discussed in Chapters 2 and 3. It is often the case that these abstractions are presented as novel control structures for swarm robotics, but rely on the fact that "it just works" to justify that it is an appropriate abstraction. While this is partly true, we have already alluded that one can be far more specific in one's justification.

We discuss in Chapter 4 three novel HSI abstractions: molding clay [33], chasing prey, and broadcasting instructions [29]. We have already discussed that the most prevalent approach is to use leader-based multi-agent coordination [53], such as single-leader networks, where the user input is applied directly to a single leader robot. This is an effective control strategy if a small number of robots is used, but it becomes less effective as the number of robots in the team grows, as we discuss in Chapter 3. Alternative approaches that have been proposed include articulating gestures [71], composing music [21], stirring fluids [57], and manipulating densities [61]; however, these approaches may require additional distributed infrastructure or complex interfaces for generating input signals. Consequently, a related question focuses on the appropriate design of input controllers such that a user is not overloaded with deciding control inputs. This question has been addressed, for example, in [25, 35]; however, the operator typically uses an input controller (like a joystick) that does not afford all of the actions needed to collectively manipulate a multi-robot team. We mitigate this drawback in Section 4.3 by using a deformable medium as an input controller that specifically affords the actions required to form geometric shapes.

Significant inspiration for designing HSI abstractions has been drawn from the many examples of cooperative behaviors in nature. Foraging and predator-prey models have inspired the design of cooperative capture strategies for multi-agent robotic systems in [3, 8, 49, 82, 83]. In Section 4.2, we demonstrate that a small group of predators can use a simple geometric formation to capture a member of a larger collection of prey. Nature provides us with numerous examples of such few-to-many (or one-to-many if we consider solitary hunters) strategies [39, 72], which we can adapt into HSI control structures by, for example, handing control of the predators to the user. This approach may lead to more natural interpretation of leader-based interactions. More importantly, we addressed the rarely answered question (see, for example, [51]) of how many robots are needed for the interaction to succeed.

We can also draw inspiration from physical processes. For example, the Brazil nut effect (granular convection) describes how a granular mixture of differently sized particles separates under direct, external vibrations [2, 22, 75]. The objective is to separate all particles with a single external input signal, rather than separating each type of particle separately, which is akin to a user shaking a box of cereal. Sensorless manipulation [9] uses external vibration and basins of attraction to sort and orient parts of different shapes and sizes. The viscosity of magnetorheological fluids can be controlled by applying an external magnetic field to align magnetic particles in the non-magnetic fluid [37]. These applications are examples of controlling passive components using a broadcast input signal, but this concept can also be extended to robots. For example, Becker et al. demonstrated in [7] and [14] that it is possible to broadcast a uniform control signal to steer a collection of robots with small variations in their turn rate to arbitrary positions. The drawback of this approach is that it does not leverage the ability of robots to interact with each other. Azuma et al. demonstrated in [1] that a global broadcast controller can achieve coverage with local controllers that switch between random and deterministic walks. A global controller broadcasts the value of an objective function that corresponds to the task of the collection of homogeneous robots. This value measures the collection's performance and is used locally to control each robot's motion. Similarly, the focus of our work in Section 4.1 is to leverage the local interactions (like in Azuma et al.) between heterogeneous robots (like in Becker et al.) under a broadcast signal from a user, such that the user can interact with the swarm collectively with a single input controller.

CHAPTER II

OBJECTIVE METRICS FOR THE ANALYSIS OF HSI CONTROL STRUCTURES

In this chapter, we discuss how to define, analyze, and objectively characterize HSI control structures, which require users to interact with a swarm of robots by applying an exogenous input signal. As we have done in [31, 32], we begin with a formal definition of what it means to impose a control structure on the dynamics of the multi-agent (or, more specifically, multi-robot) system that represents a swarm of robots. We use a control Lyapunov function (CLF) approach to show convergence of the multi-robot system to some geometric configuration to demonstrate that it is feasible for the user to complete the task of achieving this desired geometric configuration with such a swarm of robots. Additionally, we propose attention, effort, and scalability as metrics for objectively characterizing a user's interactions with an HSI-enabled swarm of robots during a specific task. These metrics can be measured after the task, or approximated before the task to characterize and improve an HSI. We will demonstrate that in the latter approach, we can use optimal control tools to generate an approximation of user control input under the assumption that users with training will act approximately optimally. Consequently, we are able to quantify the answer to the questions, *if we are given a particular HSI control structure for a swarm of robots, does it provide an interaction that requires low attention and effort?* and *Does this interaction scale well as the swarm increases in size?*, which provides us with deeper insight into the viability of a control structure than proofs of convergence alone.

2.1 Definitions

Our first objective is to determine whether it is feasible for a human operator to use a particular human-swarm interaction (HSI) to achieve some geometric configuration with a swarm of mobile robots. To establish feasibility, we first need to define what a HSI represents in terms of the control structure it imposes on a multi-robot system, and what it means for a human operator to complete a task with the robotic swarm.

2.1.1 Human-Swarm Interaction Control Structures

In general, we consider continuous-time, time-invariant systems with inputs, which represent robotic swarms that can be externally controlled (or interacted with) by a user. The dynamics of such multi-robot systems can be defined as $\dot{x}(t) = f(x(t), u(t))$, where $x(t) \in \mathcal{X}$ is the state of the system at time t and $u(t) \in \mathcal{U}$ is the input to the system at time t. In fact, $x(t)$ will represent the stacked vector of the states belonging to all robots at time t, while $x_i(t)$ will refer to the state of robot i at time t. For example, $x(t)$ will typically represent the position or pose of all robots together at time t, i.e., their geometric configuration.

More importantly, the differentiable function $f : \mathcal{X} \times \mathcal{U} \to T\mathcal{X}$, where $T\mathcal{X}$ is a tangent space, is structured according to the interaction topology of the multi-robot system. The interaction topology is given by a graph $\mathcal{G} = (V, E)$, where V is the set of vertices representing the robots, and E is the set of edges representing information exchange between robots via communication links or due to sensor footprints (see, for example, [65]). Specifically, $f \in \text{sparse}_{\mathcal{X}}(\mathcal{G})$ conveys that state information in the multi-robot system can only flow between robots that are linked in the interaction topology. Consequently,

$$f \in \text{sparse}_{\mathcal{X}}(\mathcal{G}) \Leftrightarrow \left(j \notin \overline{N}(i) \Rightarrow \frac{\partial f_i(x, u)}{\partial x_j} = 0,\ \forall x, u \right), \tag{15}$$

where $N(i)$ is the so-called neighborhood of robot i, i.e., $j \in N(i)$ if $(i, j) \in E, i, j \in V$, and $\overline{N}(i) = N(i) \cup \{i\}$ is its closure.

By picking a particular HSI control structure, we are being specific about the structure of $\mathcal{U}$, i.e., how the user can interact with the robotic swarm and how the robots in the swarm interact with each other. Our definition is as follows:

Definition 2.1.1. *An HSI control structure is a map*

$$H : \mathcal{X} \times \mathcal{V} \to \mathcal{U}, \tag{16}$$

where $\mathcal{V}$ is some set of admissible inputs to make the corresponding robotic swarm accessible to human control. Additionally,

$$f(x, H(x, v)) = f_H(x, v) \in \mathit{sparse}_{\mathcal{X}}(\mathcal{G}), \tag{17}$$

which means that the dynamics f under this map H needs to observe the sparsity structure imposed by the interaction topology.

This definition of an HSI control structure implies that the control input to the system is really a combination of state feedback and a restricted set of inputs from the user, which respects the constraints imposed by the interaction topology. Consequently, the dynamics of a multi-robot system under such an HSI control structure are

$$\begin{aligned}\dot{x}(t) &= f(x(t), u(t)) \\ &= f(x(t), H(x(t), v(t)) \\ &= f_H(x(t), v(t)).\end{aligned} \tag{18}$$

Therefore, an HSI control structure is a very specific way in which the user controls the multi-robot system, i.e., interacts with the robotic swarm.

For example, suppose that a robotic swarm consists of n mobile robots positioned on a rail ($x_i(t) \in \mathbf{R}$) with single-integrator dynamics,

$$\dot{x}_i(t) = u_i(t),\ i = \{1, \ldots, n\}, \tag{19}$$

where the control input for the first $n-1$ robots is

$$u_i(t) = \sum_{j \in N(i)} (x_j(t) - x_i(t)). \tag{20}$$

$N(i)$ denotes the neighborhood of robot i, which is the set of all its immediate neighbors in the interaction topology derived from communication links or sensor footprints.

The control input for the n-th robot is

$$u_n(t) = \sum_{j \in N(i)} (x_j(t) - x_i(t)) + v(t),\ v(t) \in \mathcal{V}, \tag{21}$$

which corresponds to the user directly controlling the position of the n-th robot. This HSI control structure is commonly referred to as a *single-leader network* (see, for example, [68]), because the user interacts with the swarm of robots by guiding a "leader" robot, while the other robots follow the leader and each other according to the consensus dynamics in (20) (see [67] for more on consensus).

If we stack all $x_i(t)$'s into a state vector $x(t) \in \mathbf{R}^n$ and all $u_i(t)$'s into an input vector $u(t) \in \mathbf{R}^n$, then the ensemble dynamics of our example system are

$$\dot{x}(t) = u(t) = -Lx(t) + lv(t),\ l = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \in \mathbf{R}^n \tag{22}$$

where L is the graph Laplacian L as defined in [65] (and commonly used in multi-robot control). Consequently, the single-leader network HSI control structure is a particular structuring of the control input $u(t)$ in (22) given by the function H, such that

$$u(t) = H(x(t), v(t)) = -Lx(t) + lv(t), \tag{23}$$

where $v(t) \in \mathbf{R}$ is the user input.

2.1.2 Tasks

Definition 2.1.2. *When a multi-robot system under some HSI control structure can asymptotically converge to a state, a subset of states, or all states in a specification set $\mathcal{S}$ and stay in this set, then*

$$\limsup_{t\to\infty} d(x(t), \mathcal{S}) = 0, \tag{24}$$

where,

$$d(x(t), \mathcal{S}) = \inf_{s\in\mathcal{S}} \|x(t) - s\|. \tag{25}$$

If this is true, then we say that the user can achieve some or all of the geometric configurations described by $\mathcal{S}$ with the robotic swarm.

The *specification set* is the set of geometric configurations that we want the user to achieve with the robotic swarm, in the sense that the user should be able to form a geometric configuration with the swarm and keep it in this configuration. For example, a specification set could be defined as

$$\mathcal{S} = \{x \in \mathbf{R}^n \mid x_i = x_j,\ i, j = 1, \ldots, n\}, \tag{26}$$

which merely states that all components of the state should be equal, or $\mathcal{S} = \text{span}\,\{\mathbf{1}\}$. For example, the specification set for consensus problems with multi-robot teams is typically defined in this way. Or, we may want the user to guide a single-leader network, such that all robots in the swarm rendezvous at a specific location, i.e., $\alpha \in \mathbf{R}, \mathcal{S} = \alpha\mathbf{1}$. More succinctly, these are examples of *geometric tasks*, which for our purposes we define as follows:

Definition 2.1.3. *A geometric task is a 2-tuple,*

$$\mathcal{T} = (x_0, \mathcal{S})\,, \tag{27}$$

where x_0 is the initial state of the swarm, and $\mathcal{S}$ is the specification set.

Definition 2.1.4. *A task supported by a control structure H is a 2-tuple,*

$$\mathcal{T}_H = (H, \mathcal{T}), \tag{28}$$

where $\mathcal{T} = (x_0, \mathcal{S})$ is a geometric task.

Consequently, a task $\mathcal{T}_H$ encodes that a user is required to guide swarm of robots from their initial configuration x_0 to a specific geometric configuration $\mathcal{S}$ by interacting with this swarm through a control structure H.

2.2 Feasibility

We have shown that the function $H : \mathcal{X} \times \mathcal{V} \to \mathcal{U}$ encodes a particular HSI control structure into the dynamics of a multi-robot system, and that if this combination of multi-robot system and control structure can asymptotically converge to a specification set $\mathcal{S}$ (or a subset thereof), then we say that it is feasible for a user to complete this task $\mathcal{T}_H = (H, (x_0, \mathcal{S}))$. More formally,

Definition 2.2.1. *It is feasible to achieve a specification set $\mathcal{S}$ under an HSI control structure defined by H if there exists $v(t)$ such that, $v(t) \in \mathcal{V}$, $\forall t \geq t_0$, and*

$$\limsup_{t \to \infty} d(x(t), \mathcal{S}) = 0,$$

when $\dot{x}(t) = f_H(x(t), v(t)), x(t_0) = x_0$.

We will use control Lyapunov functions (CLFs) [78] to determine this feasibility.

2.2.1 Control Lyapunov Functions

Let us denote $D \subset \mathcal{X}$ as a domain of the state space containing the quasi-static equilibrium point z for some $w \in \mathcal{V}$, such that $\dot{x}(t) = f_H(z, w) = 0$.

Definition 2.2.2. *A continuously differentiable $V : D \to \mathbf{R}$ with*

$$V(z) = 0 \textit{ and } V(x) > 0 \textit{ in } D - \{z\}$$

is a control Lyapunov function (CLF), if there exists a $v \in \mathcal{V}$ for each $x \in D$, such that

$$\dot{V}(x, v) = \langle \nabla V(x), f_H(x, v) \rangle < 0 \textit{ in } D - \{z\} \tag{29}$$

and $\dot{V}(z, w) = 0$.

If such a control Lyapunov function exists, then any trajectory starting in some compact subset $\Omega_c = \{x \in \mathcal{X} \mid V(x) \leq c,\ c > 0\} \subset D$ will approach z as $t \to \infty$.

Theorem 2.2.3. *If there exists a CLF as defined in Definition 2.2.2 for the system described by $\dot{x}(t) = f_H(x(t), v(t)),\ x(0) = x_0$, and the specification set $\mathcal{S}$ is some quasi-static equilibrium point $z \in D$, then it is feasible to converge to z as $t \to \infty$.*

Proof. By Definition 2.2.2, the existence of a CLF guarantees that if $x_0 \in \Omega_c$, then there exists $v(t) \in \mathcal{V}$, such that the multi-robot system converges to z asymptotically, i.e. $\lim_{t\to\infty} x(t) = z$. Since $z = \mathcal{S}$, it is true that

$$\left(\limsup_{t\to\infty} d(x(t), z) = 0 \right) \Rightarrow \left(\limsup_{t\to\infty} d(x(t), \mathcal{S}) = 0 \right),$$

which by Definition 2.2.1 confirms that for this particular multi-robot system and HSI control structure, the user can achieve the geometric configuration in the specification set $\mathcal{S}$ with the corresponding robotic swarm. □

Using this formulation of CLFs allows us to test the feasibility of achieving, for example, rendezvous at a specific location or a formation at a specific location with a specific rotation and assignment to positions. However, we would also like to capture formations that can translate and rotate, like cyclic pursuit, or rendezvous at any arbitrary location. Therefore, our definition of CLFs needs to include sets of quasi-static equilibrium points and limit cycles. Suppose that $D \subset \mathcal{X}$ is a domain of the state space that contains all or part of the specification set $\mathcal{S}$.

Definition 2.2.4. *A continuously differentiable $V : D \to \mathbf{R}$ (and locally positive definite as before) is a control Lyapunov function, if there exists $v \in \mathcal{V}$ such that*

$$\dot{V}(x, v) = \langle \nabla V(x), f_H(x, v) \rangle \leq 0 \tag{30}$$

for each x in some compact set $\Omega \subset D$, for example, Ω_c. By LaSalle's invariance principle [56], if M is the largest invariant set in $\left\{ x \in \Omega \;\middle|\; \dot{V}(x, v) = 0,\ v \in \mathcal{V} \right\}$, then any trajectory starting in Ω will approach M as $t \to \infty$.

Consequently, we must ensure that our choice of CFL satisfies $M \subseteq \mathcal{S}$, otherwise we cannot prove that it is feasible to achieve any of the geometric configurations in the specification set $\mathcal{S}$.

Theorem 2.2.5. *If there exists a CLF as defined in Definition 2.2.4 for the system defined by $(\mathcal{X}, \mathcal{V}, f_H, x_0)$ and $M \subseteq \mathcal{S}$, then it is feasible to asymptotically converge to M from any $x(t_0) \in \Omega$.*

Proof. The proof is similar to what was shown in the first theorem. By Definition 2.2.4, the existence of a CLF guarantees that if $x_0 \in \Omega$, then there exists $v(t) \in \mathcal{V}$, such that the multi-robot system converges to the invariant set M asymptotically. Therefore,

$$\limsup_{t \to \infty} d(x(t), M) = 0$$

$$\limsup_{t \to \infty} \inf_{m \in M} \|x(t) - m\| = 0.$$

If $M \subseteq \mathcal{S}$, then any $m \in M$ is also in $\mathcal{S}$, which means that

$$\limsup_{t \to \infty} \inf_{m \in \mathcal{S}} \|x(t) - m\| = 0$$

$$\limsup_{t \to \infty} d(x(t), \mathcal{S}) = 0,$$

which satisfies our definition of feasibility. □

2.3 Attention, Effort, and Scalability

Feasibility does not immediately imply that it is possible for a user to complete a task easily and effectively. A user's experience is typically explored through user studies, where measures such as attention, effort, or frustration are evaluated. Unfortunately, such user studies tend to be time-consuming. Consequently, we circumvent this problem by objectively investigating how much attention and effort are required to accomplish a given task. We also investigate the scalability of an HSI control structure with respect to attention and effort across similar tasks with larger numbers of robots in the swarm.

2.3.1 Attention and Effort

Attention and effort are common metrics by which one can characterize the user's performance in some task [54]. These metrics can be gathered through experiments in a user study, but the concept of attention and effort can also be formulated in a control theory context. Roger Brockett introduced the notion of a minimum attention controller [15] by solving an optimal control problem that minimized the total variation of the control signal over time as well as over the state of the system, e.g., the attention functional would take the form

$$\int_{\mathbf{T}} \int_{\mathbf{X}} \phi \left(x, t, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial t} \right) dx dt, \tag{31}$$

where $\mathbf{X}$ and $\mathbf{T}$ are the state space and time domains over which the controller is defined, and u is the control signal.

However, attention is only part of the story. Optimization problems often minimize $u(t)$ itself (see, for example, [16]), meaning that it is desirable to complete some task with minimal effort. Consequently, we propose that a cost on the user control input should be both in terms of attention and effort. One way to define this attention-effort cost is

$$J(v) = \int_0^\infty \left(\|v(t)\| + \|\dot{v}(t)\| \right) dt, \tag{32}$$

which encodes a cost on the magnitude of the user input v (effort) and the variation in v over time (attention). It should be noted that such a functional computes the attention-effort cost for a particular $v(t)$ and not for a particular HSI control structure. To overcome this problem, we focus instead on a particular choice of control signal – the optimal one, v^* – as a proxy for the signal a trained human operator might indeed employ.

While a user is likely to try to complete a task as fast and accurately as possible, a user also likely choses to minimize attention and effort. Too much attention or effort required to complete a task is likely undesirable. Consequently, we propose to compute the optimal control v^* using a cost function that encodes accuracy, effort, and attention simultaneously. For example, in Section 2.4.1.1 we minimize the following cost with respect to $w = \dot{v}$:

$$\begin{aligned} \min_{w} \quad & J(w) = \frac{1}{2}\int_0^\infty \left((x-\alpha\mathbf{1})^T(x-\alpha\mathbf{1}) + v^Tv + w^Tw\right) dt \\ \text{s.t.} \quad & \dot{x} = -L_f x + lv \\ & \dot{v} = w \\ & x(0) = x_0,\ v(0) = 0 \end{aligned} \tag{33}$$

The first term penalizes any swarm configuration that is not in the specification set, while the second and third terms penalize effort and attention. Computing $\dot{v}^*(t)$ allows us to construct $v^*(t)$, which we will use in evaluating the attention-effort cost. We will demonstrate this example in full in Section 2.4 for three different HSI control structures under the same geometric task.

Optimal control solutions are a function of the initial conditions; therefore, different tasks (with respect to initial conditions) are likely to result in different attention-effort costs. Consequently, we recommend to either average the cost over a sampling of the initial conditions, or use attention-effort cost to compare two different HSI control structures in the same task with the same initial conditions.

2.3.2 Scalability

Suppose that if $J_n(v_1^*)$ is the attention-effort cost for n robots in some task $\mathcal{T}_H$, then $J_{n+1}(v_2^*)$ is the attention-effort cost for $(n+1)$ robots in a similar task $\tilde{\mathcal{T}}_H$. $\mathcal{T}_H$ and $\tilde{\mathcal{T}}_H$ are not equivalent, because a task is implicitly a function of the number of robots in the swarm. Regardless, scalability approximates the increase in attention and effort required to interact with a swarm of more robots. We choose to formulate scalability similar to computational complexity for algorithms: $\Sigma(n)$ is a function that encodes the change in cost for some type of task as a function of number of robots in the swarm, n. For example, we show in Section 2.4.4 that one can linearly approximate the growth of the attention-effort cost over a range of swarm size, for example, from N to $10N$. The slope of a linear fit to this data can be used for approximating $\Sigma(n)$.

2.4 *Examples*

In this section, we will provide several examples of HSI control structures imposed on multi-robot systems for which we can find CLFs and show that a user can achieve a particular geometric configuration with a swarm of robots. We will revisit our previous example of a single-leader network, where the user guides a swarm of robots to a common rendezvous location, and compare it to using broadcast control and controlling all robots simultaneously.

2.4.1 Rendezvous with a Single-Leader Network

Rendezvous is similar to consensus in that all robots meet up at the same location; however, let us suppose rendezvous captures the additional constraint that all robots should meet up at a particular location. The specification set that encodes this objective is $\mathcal{S} = \{x \in \mathbf{R}^n \mid x_i = \alpha,\ \alpha \in \mathbf{R},\ i = \{1, 2, \ldots, n\}\}$, or more concisely, $\mathcal{S} = \alpha\mathbf{1}$, where α is the rendezvous location.

We chose a candidate CLF [68] given by

$$V(x) = \frac{1}{2}\|x - \alpha\mathbf{1}\|^2, \tag{34}$$

which captures the disagreement between the current state of the robotic swarm and the rendezvous location. $V(x)$ is positive definite everywhere except at the desired equilibrium point $x = \alpha\mathbf{1}$ and is radially unbounded ($\|x\| \to \infty \Rightarrow V(x) \to \infty$).

Next, we need to compute $\dot{V}(x, v)$, which is defined by

$$\begin{aligned} \dot{V}(x, v) &= \langle \nabla V(x), f_H(x, v) \rangle \\ &= (x - \alpha\mathbf{1})^T(-Lx + lv) \\ &= -(x - \alpha\mathbf{1})^T Lx - (\alpha - x_n)v \\ &= -x^T Lx - (\alpha - x_n)v. \end{aligned} \tag{35}$$

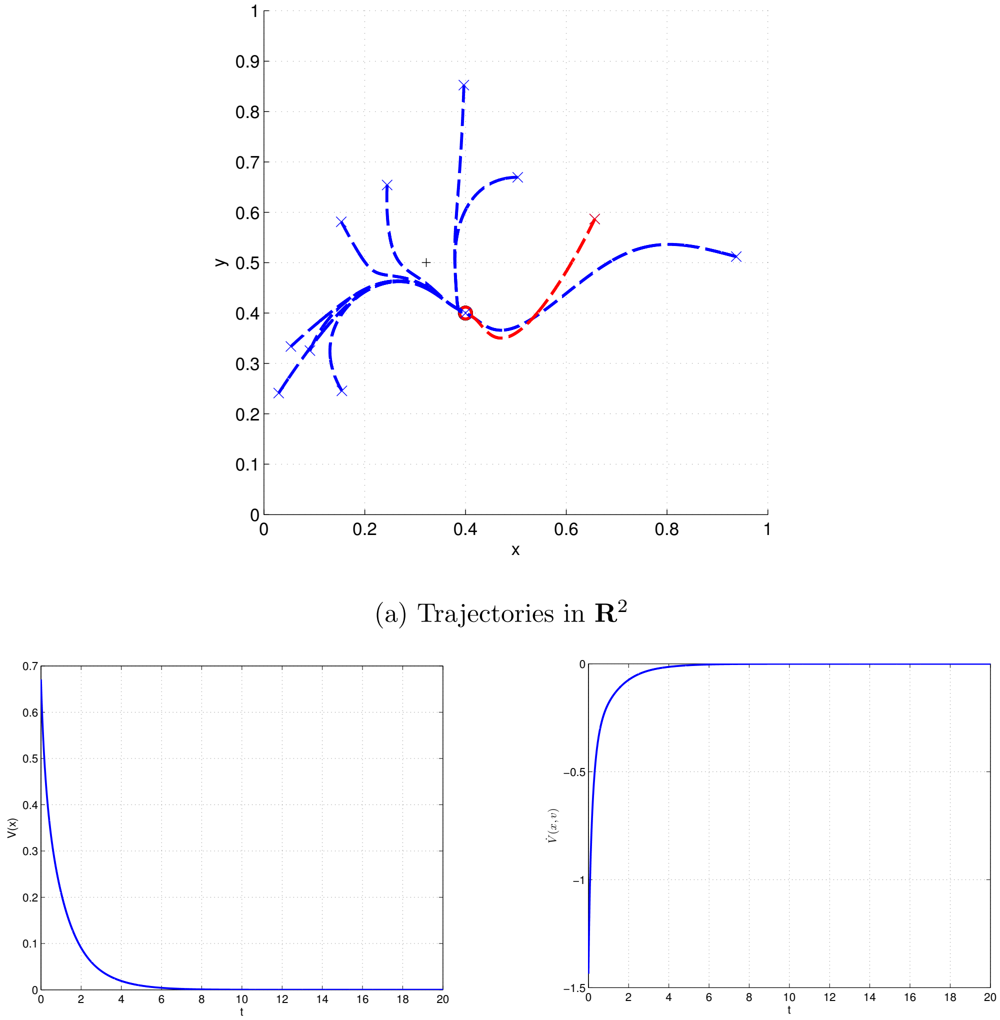
If $V(x)$ is a CLF, then it must be true that for each $x \in \mathbf{R}^n$, there exists $v \in \mathcal{V}, \mathcal{V} = \mathbf{R}$ such that $\dot{V}(x, v) < 0$ when $x \neq \alpha\mathbf{1}$ and $\dot{V}(x, v) = 0$ when $x = \alpha\mathbf{1}$. In Equation (35), we can see that since $-x^T Lx$ is semi-definite positive, we can always chose $v \in \mathbf{R}$, such that $\dot{V}(x, v) < 0$. Therefore, $V(x)$ is a CLF that guarantees that there exists $v(t) \in \mathcal{V}$, such that the user can guide the swarm of robots from $x(t_0) \in \mathbf{R}^n$ to $x = \alpha\mathbf{1}$ as $t \to \infty$.

Figure 3a is a demonstration of rendezvous with a single-leader network. To aid in the visualization, the above candidate CLF and the single-leader network system have been extended to $\mathbf{R}^2$. Since the robots are single integrators, the dynamics along each dimension, x and y, are decoupled. The user applies a control input $v \in \mathbf{R}^2$ to guide the leader robot to the origin. All robots that are organized over an arbitrary connected, static interaction topology. The solid, red trajectory belongs to the leader robot that is controlled by the user, while the dashed, blue trajectories belong to all other robots in the swarm. $\times$ denotes their starting location, while $+$ denotes the rendezvous location if $v(t) = 0, \forall t$, and $\circ$ illustrates the robots' actual final position. Figure 3b shows that the CLF $V(x, y)$ is positive, but "energy" dissipates as robots converge on the rendezvous location, while Figure 3c shows that $\dot{V}(x, y, v)$ remains negative during the interaction. Consequently, it is feasible for the user to use this HSI control structure to choose the rendezvous location of a swarm of robots. Similarly, this combination of multi-robot system and HSI control structure would be effective in setting the flocking direction if the state x were the orientation θ of each robot, rather than its position.

2.4.1.1 Attention, Effort, and Scalability

We discussed in a previous section that in place of measuring $v(t)$, we compute $v^*(t)$ using optimal control tools. We solve the following optimization problem:

$$\begin{aligned} \min_{w} \quad & J(w) = \frac{1}{2} \int_0^\infty \left((x - \alpha\mathbf{1})^T (x - \alpha\mathbf{1}) + v^T v + w^T w \right) dt \\ \text{s.t.} \quad & \dot{x} = -Lx + lv \\ & \dot{v} = w \\ & x(0) = x_0, \ v(0) = 0 \end{aligned} \tag{36}$$



(a) Trajectories in $\mathbf{R}^2$

(b) $V(x, y)$

(c) $\dot{V}(x, y, v)$

Figure 3: A user is guiding a swarm of ten robots to rendezvous at $(0.4, 0.4)$ by interacting with the leader robot (red).

This is a continuous-time, infinite horizon linear quadratic regulator-like (LQR-like) problem, which can be solved in the following manner. First, the first order necessary conditions (FONC) for optimality are:

$$\begin{aligned}
H &= \frac{1}{2}\left((x-\alpha\mathbf{1})^T(x-\alpha\mathbf{1}) + v^Tv + w^Tw\right) + \lambda^T\dot{x} + \mu^T\dot{v} \\
\frac{\partial H}{\partial w} &= w^T + \mu^T = 0 \Rightarrow w = -\mu \\
\dot{\lambda} &= -\frac{\partial H}{\partial x} = L^T\lambda - x + \alpha\mathbf{1} \\
\dot{\mu} &= -\frac{\partial H}{\partial v} = -l^T\lambda - v
\end{aligned} \tag{37}$$

It is important to note here that the co-state dynamics, $\dot{\lambda}$, include an extra affine term that is typically not present in a standard LQR problem. For convenience, let us stack states and co-states into single variables in the following way:

$$\begin{aligned} z &= \begin{bmatrix} x \\ v \end{bmatrix}, \; \dot{z} = \begin{bmatrix} -L & l \\ 0 & 0 \end{bmatrix} z + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w = A_z z + B_z w \\ \eta &= \begin{bmatrix} \lambda \\ \mu \end{bmatrix}, \; \dot{\eta} = \begin{bmatrix} -L^T & 0 \\ l^T & 0 \end{bmatrix} \eta - z + \begin{bmatrix} \alpha\mathbf{1} \\ 0 \end{bmatrix} = -A_z^T \eta - z + \Psi \end{aligned} \tag{38}$$

We propose that $\eta(t) = S(t)z(t) + P(t)$ is the solution to the stacked co-state equations. The affine component, $P(t)$, is to account for the affine component that is tracked in the cost. If we start from the proposed solution, then

$$\begin{aligned} \eta &= Sz + P \\ \dot{\eta} &= \dot{S}z + S\dot{z} + \dot{P} \\ -A_z^T \eta - z + \Psi &= \dot{S}z + SA_z z + SB_z w + \dot{P} \\ -A_z^T Sz - A_z^T P - z + \Psi &= \dot{S}z + SA_z z - SB_z B_z^T Sz - SB_z B_z^T P + \dot{P} \\ -\dot{P} - (A_z^T - SB_z B_z^T)P + \Psi &= \left(\dot{S} + SA_z + A_z^T S - SB_z B_z^T S + I\right) z \end{aligned} \tag{39}$$

Since this LQR-like problem is computed over an infinite horizon, we can compute the steady state $\hat{S}$ and $\hat{P}$, when $\dot{S} = 0$ and $\dot{P} = 0$. Consequently, to satisfy Equation 101, we must solve

$$\begin{aligned} \hat{P} &= (A_z^T - SB_z B_z^T)^{-1} \Psi \\ 0 &= \hat{S}A_z + A_z^T \hat{S} - \hat{S}B_z B_z^T \hat{S} + I \end{aligned} \tag{40}$$

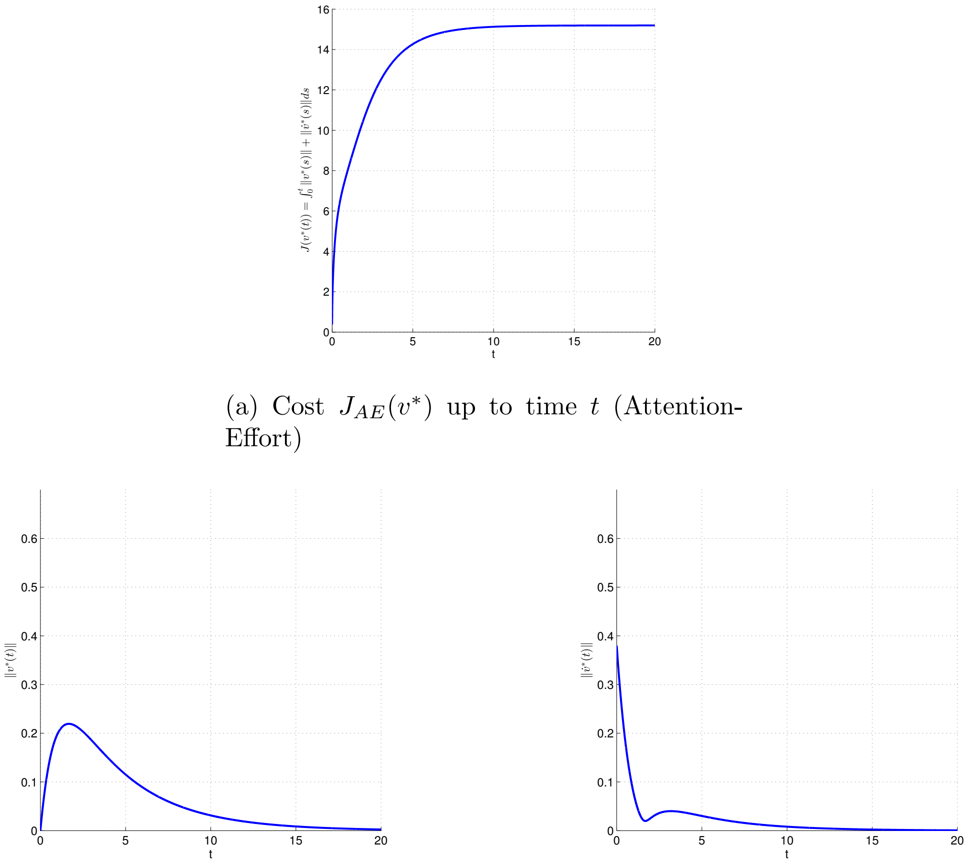
The second equation is the continuous time algebraic Ricatti equation, while P can be solved for directly. Finally, we are able to compute $\dot{v}^* = w$,

$$\begin{aligned} w &= -\mu \\ &= -B_z^T(\hat{S}z + \hat{P}). \end{aligned} \tag{41}$$

Consequently, the optimal user control input signal is

$$v^*(t) = \int_0^t w(\tau) d\tau, \; v^*(0) = 0. \tag{42}$$

Figure 3a was generated by controlling the single-leader network with the optimal user control input $v^*(t)$. The attention-effort cost for a swarm of ten robots is illustrated in Figure 4a, while Figure 4b and 4c illustrate the instantaneous effort and attention. The attention-effort cost caps out once the swarm converges to the rendezvous location. The instantaneous effort ramps up shortly to drive the leader robot to the rendezvous location, which also requires some attention. Once the swarm is at the rendezvous location, both (instantaneous) attention and effort are zero.



(a) Cost $J_{AE}(v^*)$ up to time t (Attention-Effort)

(b) $\|v^*(t)\|^2$ (Effort)

(c) $\|\dot{v}^*(t)\|^2$ (Attention)

Figure 4: A user's approximated instantaneous attention and effort and cumulative attention-effort cost while guiding a swarm of ten robots to a rendezvous location.

Scalability can be calculated by augmenting the task by adding more robots. In this example, the new robot is added to the swarm in a random location. Figure 5 illustrates the increased attention, effort, and attention-effort cost of completing the "same" task with an more robots. The increase in cost is mainly attributed to an increase in effort as shown by the red dashed line in Figure 5, while attention has only marginally increased as shown by the black dash-dotted line in the same figure. The scalability metric for this particular task is approximated by a linear fit to the attention-effort cost. The slope of this linear fit is $\Sigma(n) = 1.05n$, which is an increase in the attention-effort cost for every robot. However, the exact coefficient of $\Sigma(n)$ is only meaningful once we will compare $\Sigma(n)$ for different HSI control structures under this same geometric task.

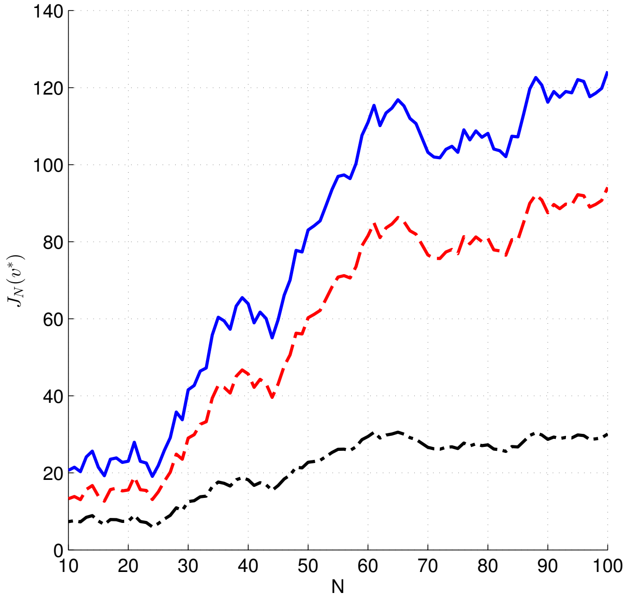


Figure 5: Growth of attention (black dash-dotted), effort (red dashed), and attention-effort (blue solid) cost for guiding a single-leader network of N robots to the rendezvous location.

2.4.2 Rendezvous with a Broadcast Signal

In Section 4.1, we discuss that it is possible to use a broadcast signal to separate a swarm of heterogeneous robots, but let us first examine this HSI control structure in the context of this chapter. Suppose that broadcasting an input signal is a HSI for the same swarm of ten package sorting robots, and we would like to know if it is feasible to rendezvous at a particular location by broadcasting an input signal. The dynamics of the swarm are

$$\dot{x}(t) = -Lx(t) + \mathbf{1}v(t) \tag{43}$$

where L is once again the graph Laplacian, and $\mathbf{1} \in \mathbf{R}^n$ is a vector of all ones.

We again choose the candidate CLF given by

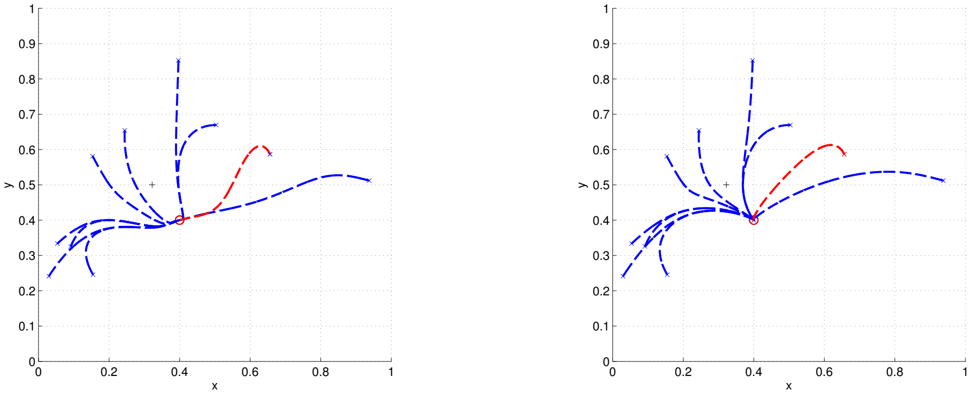
$$V(x) = \frac{1}{2}\|x - \alpha\mathbf{1}\|^2, \tag{44}$$

which captures the disagreement between the current state of the robotic swarm and the rendezvous location. $V(x)$ is positive definite everywhere except at the desired equilibrium point $x = \alpha\mathbf{1}$ and is radially unbounded ($\|x\| \to \infty \Rightarrow V(x) \to \infty$).

Following the same procedure as before, we can show that for each $x \in \mathcal{X},\ \mathcal{X} = [0, 1]$, there exists a $v \in \mathcal{V},\ \mathcal{V} = \mathcal{R}$, such that

$$\dot{V}(x, v) = -x^T Lx + (x - \alpha\mathbf{1})^T \mathbf{1}v < 0, \tag{45}$$

when $x \neq \alpha\mathbf{1}$ and $\dot{V}(x, v) = 0$ when $x = \alpha\mathbf{1}$. We can also solve for v^* as before. The LQR-like optimal control problem is largely unchanged with the state constrain being equal to Equation 43 being the only difference. Figure 6a illustrates rendezvous using the optimal broadcast control $v^* \in \mathbf{R}$.



(a) Broadcast control.

(b) Concurrent control.

Figure 6: These trajectories in $\mathbf{R}^2$ illustrate rendezvous at $(0.4, 0.4)$ using broadcast and concurrent control.

2.4.3 Rendezvous with a Concurrent Controller

When we first introduced the package sorting example, we spoke of the merits of using single-leader networks over controlling all ten robots concurrently, because the latter approach seems tedious for a single user. However, it is worthwhile to compute the attention, effort, and scalability of this control structure for comparison. The control structure of this *concurrent control* approach is

$$\dot{x}(t) = -Lx(t) + v(t), \tag{46}$$

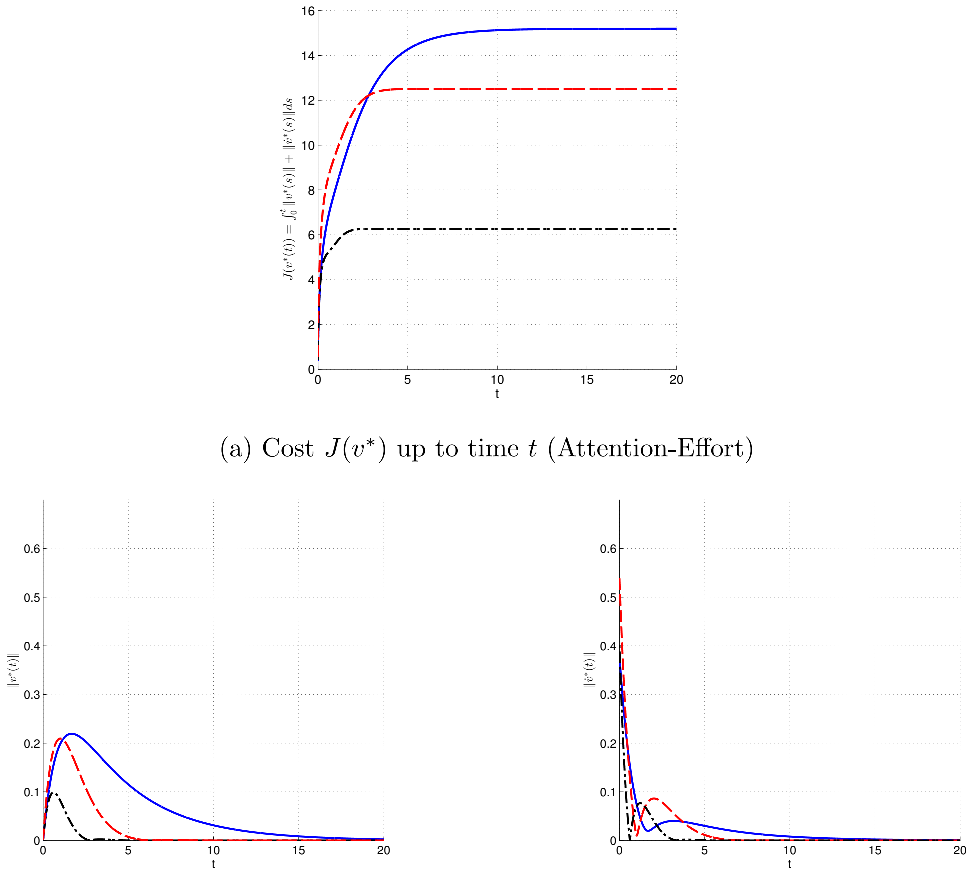
where $v(t)$ is a N-dimensional input vector, because then input to each robot is computed separately. The same procedures as before can be used to show that there exist a CLF that proves it is possible to guide the swarm to the rendezvous location $\alpha\mathbf{1}$, and that it is possible to solve the LQR-like optimal control problem to compute $v^*(t)$. Once again, the difference is in the constraints, and additionally $v \in \mathbf{R}^n$ instead of $v \in \mathbf{R}$, which affects the dimensions of matrices, but not does not change the methodology. Figure 6b illustrates rendezvous using $v^* \in \mathbf{R}^n$.

2.4.4 Comparison

We have shown in the previous section that all three control structures support the rendezvous task, which precludes us from choosing one control structure over the other. However, we can use attention, effort, and scalability as metrics for making this decision. We can compute an optimal control input v^* for each task, $\mathcal{T}_{CC}$, $\mathcal{T}_{SLN}$, and $\mathcal{T}_{BC}$, which correspond to the rendezvous problem tied to concurrent control, single leader networks, and broadcast control respectively. Figure 7 contains plots of the attention-effort cost, attention, and effort on the interval $t \in [0, 20]$ of all three tasks, where $\mathcal{T}_{CC}$ is solid blue, $\mathcal{T}_{SLN}$ is dashed red, and $\mathcal{T}_{BC}$ is dash-dotted black.

If we focus on Figure 7a, then it is evident that using a single leader network for rendezvous incurred the greatest attention-effort cost, while broadcast control incurred the least attention-effort cost. The effort required for rendezvous under concurrent control and a single leader network is almost the same at its greatest in Figure 7b, but the effort is sustained longer for the single leader network. On the other hand, attention is less for the single leader network than concurrent control as shown in Figure 7c. Broadcast control required less attention and effort compared to the other two control structures.

If the broadcast control-based task requires less attention and effort compared to concurrent the control-based and single-leader network-based task, is this also true for a larger



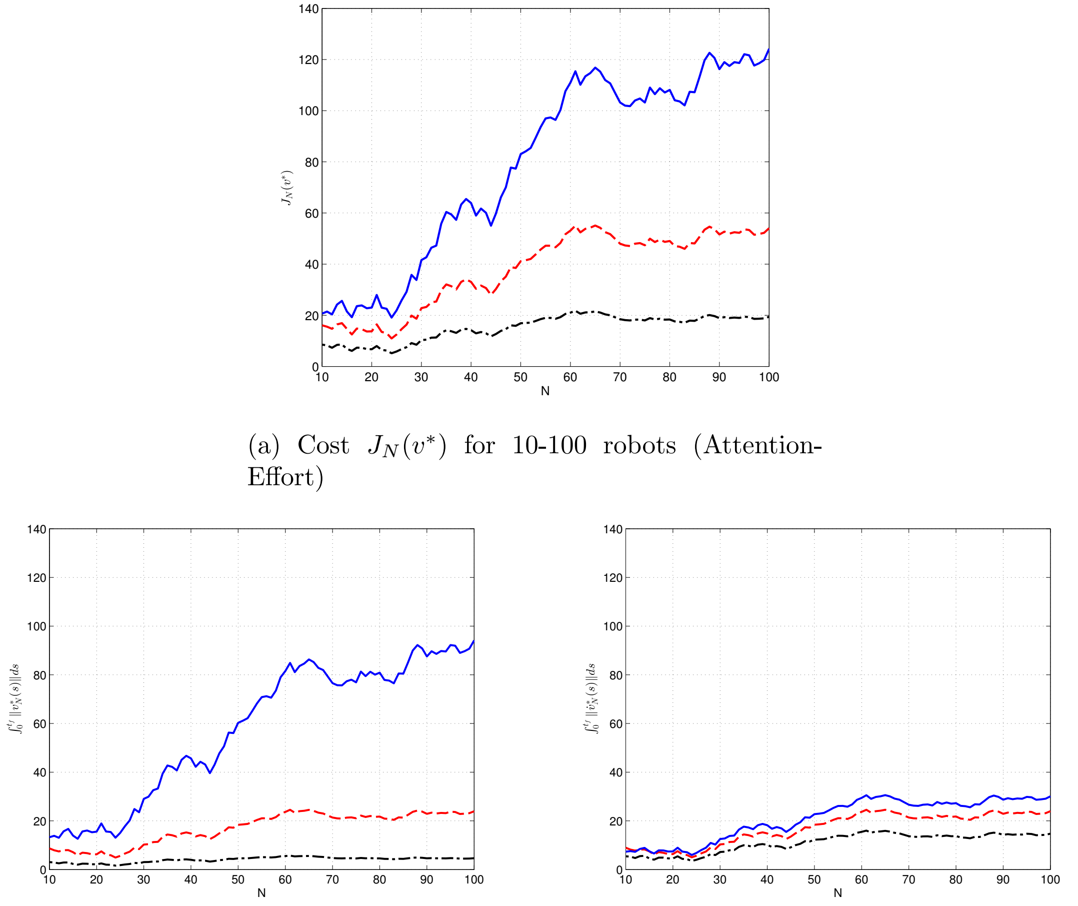
(a) Cost $J(v^*)$ up to time t (Attention-Effort)

(b) $\|v^*(t)\|$ (Effort)

(c) $\|\dot{v}^*(t)\|$ (Attention)

Figure 7: A user's estimated attention and effort while guiding a swarm of 10 robots to rendezvous at a specific location with different control structures: single-leader network (blue solid), broadcast control (black dash-dotted), and concurrent control (red dashed).

number of robots? Scalability describes the growth rate of the attention-effort cost when modifying the task by adding more robots to the swarm. Figure 8 illustrates the effect of increasing the swarm size from ten to 100 robots on attention, effort, and the combined attention-effort cost. The procedure for increasing the swarm size was to add each new robot to the workspace by choosing its location from a uniform distribution that covers the entire workspace, $\mathcal{X} = \{x, y \mid x \in [0,1],\ y \in [0,1]\}$. Figure 8a includes a linear fit to the attention-effort cost data, i.e., $\Sigma(n) = cn$, where $c = 0.31$ for concurrent control, $c = 1.05$ for single leader networks, and $c = 0.13$ for broadcast control. Comparing Figure 8b and 8c reveals that these control structures are more differentiated by effort than attention, and that attention levels off after $N \approx 60$.



(a) Cost $J_N(v^*)$ for 10-100 robots (Attention-Effort)

(b) Cumulative effort for 10-100 robots

(c) Cumulative attention for 10-100 robots

Figure 8: Growth of a user's estimated attention and effort while guiding a swarm of 10-100 robots to rendezvous at a specific location with different control structures: single-leader network (blue solid), broadcast control (black dash-dotted), and concurrent control (red dashed).

The result of this comparison is that broadcast control outperforms concurrent control, while concurrent control outperforms single leader networks with respect to attention, effort, and scalability. However, this comparison omits one important factor that differentiates concurrent control from single-leader networks (and broadcast control), which is the fact that the dimension of the former control structure grows linearly in the size of the swarm, while the latter control structures require the user to decide only a two dimensional input (akin to a joystick). Consequently, if a single user could yield 100 joystick, or gather 100 co-operators, or rely on a computer (perhaps, the user simply specifies a goal location with a point-and-click interface), then concurrent control is better than a single-leader network.

Regardless, broadcast control is the better control structure for this task.

2.4.5 Recommendations

The three examples in the section show that a CLF approach is useful to show convergence of the multi-robot system to a specification set. In fact, our definition of a HSI control structure allows us to use CLFs directly, and the CLFs themselves can typically be constructed by inspecting the specification set. The specification set is also useful when adding a tracking cost to the optimal control problem. The optimal control problems may be different for each HSI control structure and task; however, we have shown that a general guideline is to include a tracking, effort, and attention cost when solving for v^*. Consequently, v^* will likely serve as a good proxy for the user control input, v, when evaluating attention, effort, and scalability. Attention, effort, and scalability can consequently be used to compare three tasks that differentiate in the control structure that is used.

2.5 Conclusions

In this chapter, we have provided a precise definition for what it means to impose a human-swarm interaction (HSI) control structure on a multi-robot system and to achieve a geometric configuration with a swarm of robots. With these two definitions in hand, we defined that feasibility in this context implies that a user can successfully guide a swarm of robots into some desired geometric configuration. We have also shown that finding a control Lyapunov function (CLF) implies feasibility, such that CLFs can be used to show that a particular combination of multi-robot system and HSI control structure is appropriate for achieving a particular geometric configuration or set of configurations as demonstrated by the included examples. Additionally, we proposed attention, effort, and scalability as metrics for evaluating a user's interactions with an HSI-enabled swarm of robots during a specific task. We demonstrated how to use optimal control tools to generate an approximation of the user control input, which allowed us to evaluate and compare HSI control structures before users have to interact with the swarm of mobile robots.

CHAPTER III

COMBINING OBJECTIVE AND SUBJECTIVE METRICS FOR THE CHARACTERIZATION OF HSI CONTROL STRUCTURES

In the previous chapter we discussed the use of optimal control to approximate the attention and effort required of a trained user to complete a particular task. Alternatively, we could have chosen to invite a number of participants to complete the task, capture each participant's input signal $v(t)$, and analyze the attention, effort, and scalability in the same way with the actual $v(t)$ in place of $v^*(t)$. Additionally, we could have queried the participants about how much effort and attention they felt that the task required. This approach is a so-called user study, which is typically employed to gather a set of objective and subjective data to evaluate and compare. A quantitative comparison of this data is useful, for example, to decide which control structure to choose for the task amongst a set of control structures. In this chapter, we will instead demonstrate how to correlate the user study data to control and graph theoretic properties of the control structure. This correlation allows us to understand how the characteristics of a control structure influence the difficulty of completing the task, such that we can improve its design.

The main focus of this chapter is on the role of the underlying *interaction topology* (graph), since the characteristics of a control structure are strongly dependent on it. For example, all three control structures introduced in Section 2.4 depended on the fact that the underlying interaction topology was connected. If an interaction topology had any disconnected components, then rendezvous of the entire swarm would no longer be possible (see, for example, [65] for a proof). We can also observe variability in the attention and effort required to complete a task whenever the topology is changed. Consequently, we are justified in focusing on the effect of the interaction topology on the efficacy of different types of control structures.

Since there exist a variety of different ways in which users can interact with a swarm of mobile robots (see, for example, [21, 33, 46, 64, 66, 76]), we decided to once again focus on single-leader networks, since this type of control structure has been well studied (see, for example, [40, 74, 79]) from a control-theoretic vantage point. A number of works have connected *controllability* properties of such control structures to the underlying interaction topology (see, for example, [63, 65]). However, controllability or the control Lyapunov function (CLF) approach in Section 2.2.1 only establish what can theoretically be achieved through a control structure. But, these properties do not quantify how easy or difficult it is for users to *actually* interact with a swarm of robots. Consequently, we introduced attention, effort, and scalability in Section 2.3; however, these metrics were tied to the user input signal, rather than the properties of the control structures themselves. Therefore, we are faced with the question, *How are the interaction topology dependent control structure properties correlated to a user's perceived difficult of interacting with a swarm of robots through this control structure?* In this chapter, we attempt to answer this question by conducting a user study, as we have done in [27, 38], where people are to control teams of simulated mobile robots through single-leader network control structures with different interaction topologies. This user study is described in Section 3.1. In particular, the participants are asked to rate the difficulty of the control task across different interaction topologies. These topologies are connected to control and graph theoretic properties in Section 3.2, including properties such as controllability, node centrality, and network centralization. The findings are reported in Section 3.3, and the main result is that the user study strongly indicates that the established controllability properties do tell a part of, but not the whole story.

3.1 User Study

We describe the user study in three parts. First, we describe our selection of interaction topologies, which forms the basis of the different tasks in the user study. Then, we follow up with a description of the experimental setup and the results of the user study.

3.1.1 Interaction Topologies

We have previously discussed that an interaction topology, i.e., a graph, describes how information is flowing among the robots for the purpose of coordinating their behaviors.

Since the explicit aim with this chapter is to understand what role the interaction topology plays when users control single-leader networks, we first need to discuss which of the many possible interaction topologies should be considered. We restrict our discussion to a smaller set of common interaction topologies to generate results that are both representative and practically useful.

The least connected (in terms of the so-called algebraic connectivity) of the connected graphs over n nodes (vertices) is the so-called line graph, L_n, where each node, with the exception of the terminating nodes, is connected to two other nodes. This is a very natural organization, found for example in single-file military columns. Consequently, we include L_n in the list of candidate topologies under consideration. However, choosing the topology itself is not enough. We also need to decide which node in the graph should be the leader-node, where the user can apply an input signal. For L_n, we consider three different leader locations–the head node, a node offset from the head, and the center node of a line graph (consisting of an odd number of nodes), as illustrated in Figure 9.

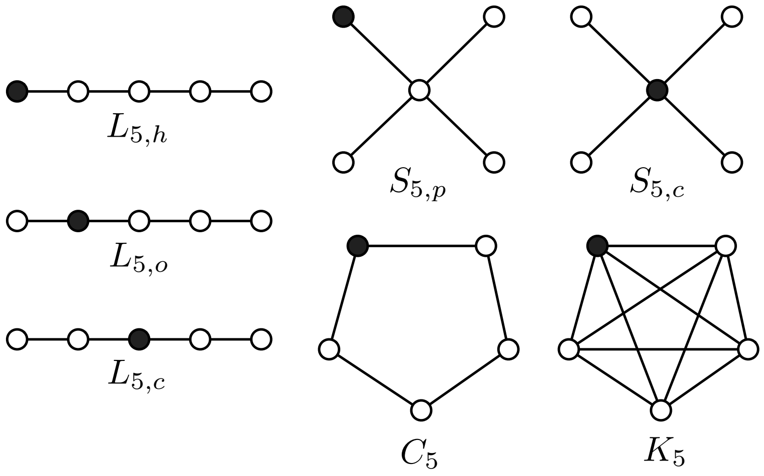


Figure 9: The four canonical interaction topologies used in the user studies: L (line), S (star), C (cycle), and K complete graph. The black node is the leader node while the white nodes are the followers. The subscripts refer to the total number of nodes and which node serves as the leader for the cases.

The line graph is a degenerate example of an acyclic (or tree) graph. Another degenerate, yet typical, example of an acyclic graph is the star graph S_n. This graph consists of a central node connected to all other nodes in the graph. These peripheral nodes do not share edges with any additional nodes, and this topology is found, for example, in certain communication networks where a central hub shares and receives information from the additional nodes. As the star graph S_n constitutes one extreme tree graph and the line graph L_n the other, we include this topology in our list of candidate topologies, and we let the central as well as a peripheral node be the leader in the user study, as illustrated in Figure 9.

The simplest cyclic graph is the cycle graph, which is denoted by C_n. It is a “closed” line graph, where each node shares an edge with two other nodes in the graph. Cycle graphs are found naturally in certain social contexts (such as games). Since all nodes are equal, we can assign any node as the leader.

Finally, the complete graph K_n is a graph where each nodes is connected to all other nodes. This type of structure is common in communication networks (broadcast-based) or when a small number of mobile robots are coordinating their behaviors. Since all nodes are connected to all other nodes, it again does not matter which node is assigned as the leader. Table 1 summarizes the interaction topologies used in the user study and defines a new notation we use to encode the interaction topology with a leader location, e.g., $S_{7,p}$ for a star graph with a peripheral leader node.

Table 1: Network configuration, leader location, and target configuration for each task.

Tasks	Network	Leader	Notation	Targets
1, 8	L_7	Head	$L_{7,h}$	Ellipse, Wedge
2, 9	L_7	Offset	$L_{7,o}$	Ellipse, Wedge
3, 10	L_7	Center	$L_{7,c}$	Ellipse, Wedge
4, 11	C_7	Any	C_7	Ellipse, Wedge
5, 12	K_7	Any	K_7	Ellipse, Wedge
6, 13	S_7	Center	$S_{7,c}$	Ellipse, Wedge
7, 14	S_7	Periphery	$S_{7,p}$	Ellipse, Wedge

3.1.2 Experimental Setup

The purpose of the user study is to measure the perceived difficulty of controlling a particular interaction topology with a single leader; therefore, 18 participants were presented with 14 such tasks. The order in which the tasks were given was randomized, such that any learning and order biases in the data were minimized. Each task consisted of moving the single-leader network via the leader from its initial configuration to a target geometric configuration. For example, participants were asked to guide a L_7 network from its initial configuration into an ellipse. Table 1 provides a detailed list of the 14 tasks. The tasks were selected such that all networks were paired with each target formation, and all target formations were significantly different from a network's initial configuration. These two conditions ensured that none of the tasks were trivial to complete (e.g., form circle with a C_7 network).

The experiment was structured such that the participant was shown the interaction topology only prior to the start of the task. Communication links were not visible to the eye and it was up to the participant to infer the behavior of the network from the interactions of the robots. The participants were only able to directly control the movement of the leader during the experiments using a joystick.

During the tasks, the participants received no feedback (e.g., a scoring meter). This choice was made so that focus was entirely on matching the network to the target formation. During each task, a score was calculated from a least square fit of the network's current configuration to its target formation. And, since we were only concerned about the participants matching the *shape* of the formation, the least square (LSQ) fit was translation, rotation, and assignment invariant, meaning that neither the location of the formation in the workspace, nor the assignment of robots to specific positions in the formation mattered, following the developments in [52].

After each task, the participants' experiences were recording, by rating the perceived difficulty of the tasks on a scale from very easy (0) to very difficult (20) in conjunction with a NASA Task Load Index (TLX) workload survey [50]. The workload survey consisted of six questions that covered physical, mental, and temporal demands, as well as levels of performance, effort, and frustration. Consequently, the mean least squares error, difficulty rating, and total raw score of the TLX workload survey were repeated for each task. The duration of the task as well as the total distance traveled by the robot network were also recorded.

To get a sense for the way the experiment is experienced, see Figure 10, where a participant controls a collection of simulated robots through a graphical user interface as shown in Figure 11. The interface is split into two areas, where the left area is the current state of the network and the right area is the desired target configuration. The participant is moving the leader using the analog joystick on a gamepad. Figure 12 shows the graphical user interface used by the participant to rate the difficulty and the workload for each task.

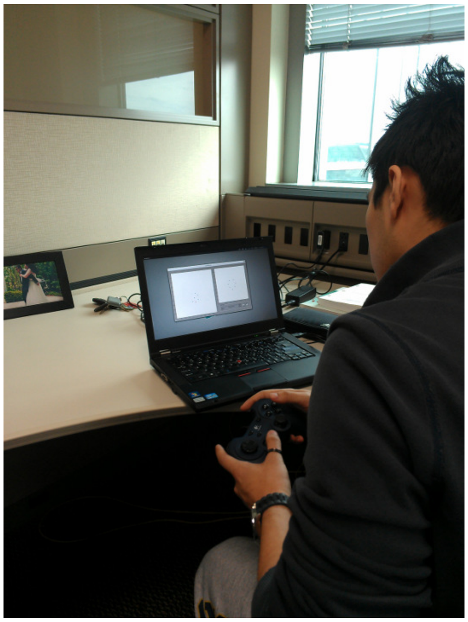


Figure 10: Photo of a participant performing one of the tasks in the user study. The participant is moving the network around using the analog joystick on a gamepad to match the target formation.

3.1.3 Experimental Results

A repeated-measures, one-way ANOVA statistical test [45] on the collected user-study data reveals that the LSQ score ($p < 0.0000001$), rating score ($p = 0.00138$), and workload score ($p = 0.0256$) are all *statistically significant* at a 0.05 (or 95%) confidence level. We are justified in applying the ANOVA statistical test, since the user-study data in most cases satisfies the underlying assumptions of the test, which state that any sampled data should be independent and normally distributed, and the variances across tasks should be equal. For example, Figure 13 illustrates that the sampled NASA TLX workload scores are approximately normally distributed. The normal distribution is somewhat skewed in the

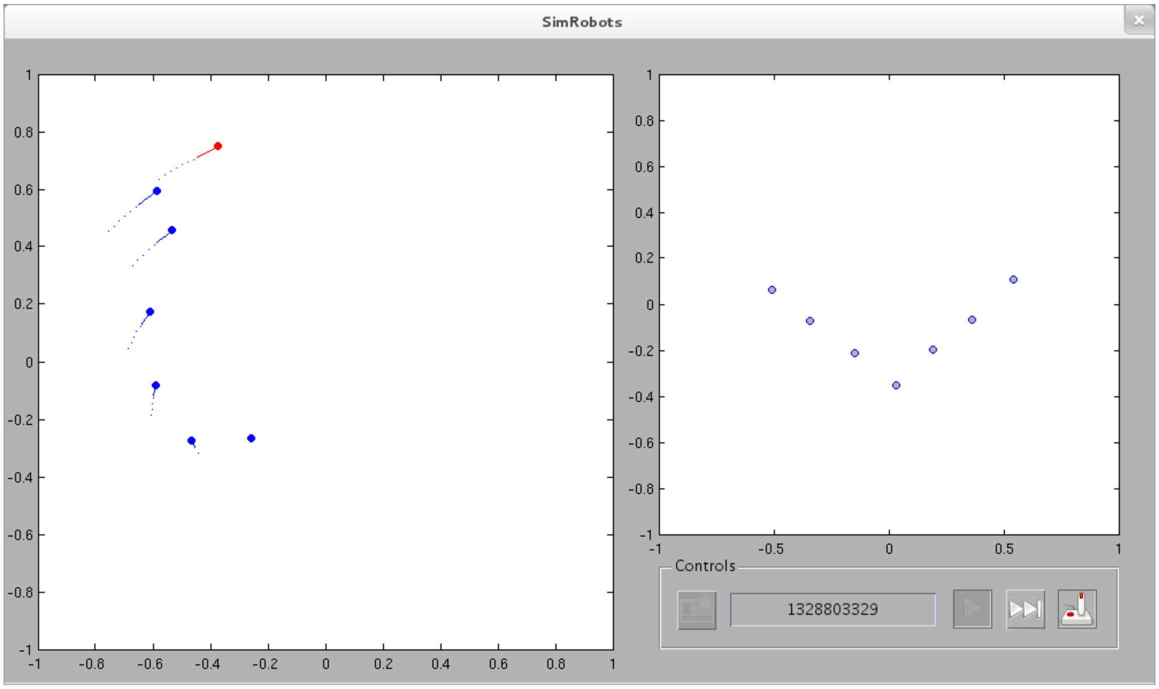
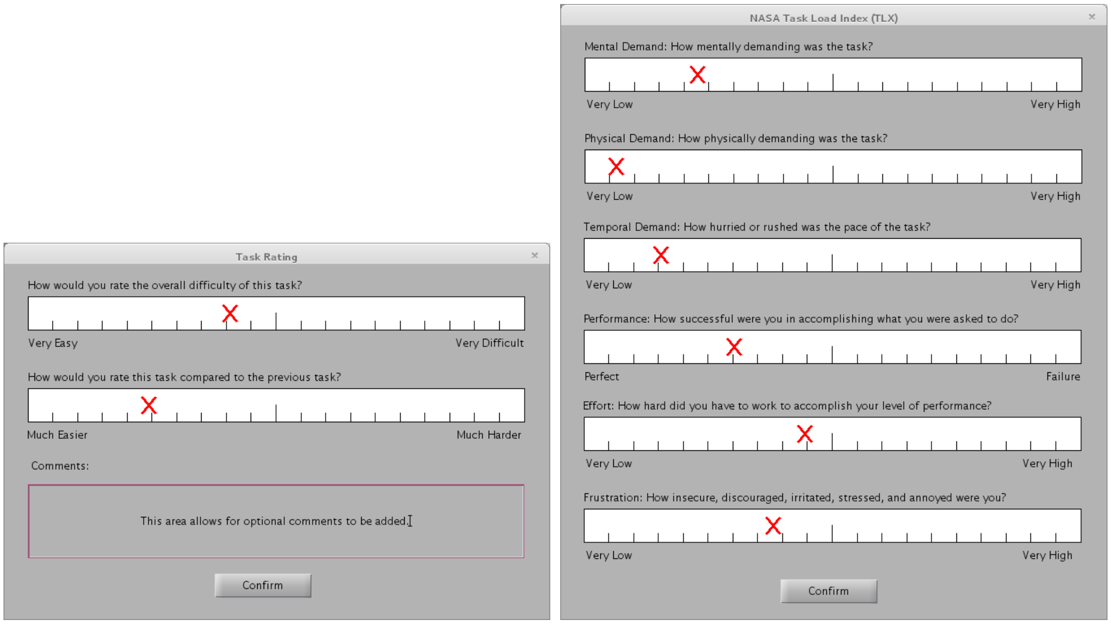


Figure 11: Screen capture of the graphical user interface used by the participant to control a L_7 network.



How would you rate the overall difficulty of this task?

How would you rate this task compared to the previous task?

This area allows for optional comments to be added

Mental Demand: How mentally demanding was the task?

Physical Demand: How physically demanding was the task?

Temporal Demand: How hurried or rushed was the pace of the task?

Performance: How successful were you in accomplishing what you were asked to do?

Effort: How hard did you have to work to accomplish your level of performance?

Frustration: How insecure, discouraged, irritated, stressed, and annoyed were you?

(a) Task rating interface.

(b) NASA TLX workload interface.

Figure 12: Screen capture of the graphical user interface used by the participant to rate the task. Figure 12a corresponds to questions about the overall difficulty of the task by itself and in comparison to the previous task. Figure 12b corresponds to questions about mental, physical, and temporal demands, as well as, levels of performance, effort, and frustration.

case of the LSQ scores; however, the ANOVA statistical test is relatively robust to such violations of its normality assumption [58].

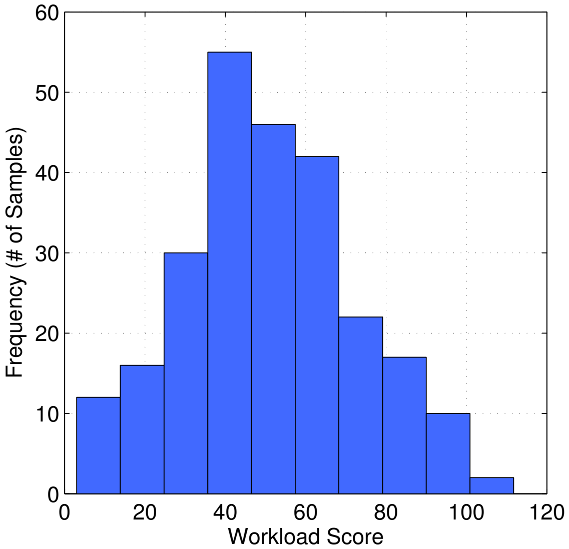


Figure 13: The sampled NASA TLX workload scores are approximately normally distributed.

We use the LSQ, rating, and NASA TLX workload scores to make comparisons between the different tasks and the different interaction topologies, and for decide which topologies are easier for users to guide the single-leader network to the target geometric configuration. While the time data ($p = 0.012$) is also statistically significant, the distance data ($p = 0.262$) is not statistically significant enough to use as a measure to distinguish between the tasks. Since we did not ask participants to minimize time or distance, we chose to omit both metrics from the analysis.

Before describing the interaction protocols, the robots dynamics, or how the leader-follower teams were formally defined, we first discuss the findings from the user studies in terms of the captured data. These findings will be related to formal properties in the robot swarm in subsequent sections. Figure 14 is a histogram of the mean LSQ scores for each task, with error bars denoting the standard error, which is computed by normalizing the standard deviation with the square root of the number of samples. The standard error expresses the region in which we can be confident that the true population mean lies (see [24]). If we want to claim that one task received a lower score than another task, we have to check for a statistically significant difference in the regions denoted by the error bars of the respective tasks. The repeated-measures ANOVA (analysis of variance) test performs these pairwise comparisons and reports the significance level of any difference. If a difference is statistically significant, we are justified in claiming that one task received a lower (or higher) score than another task.

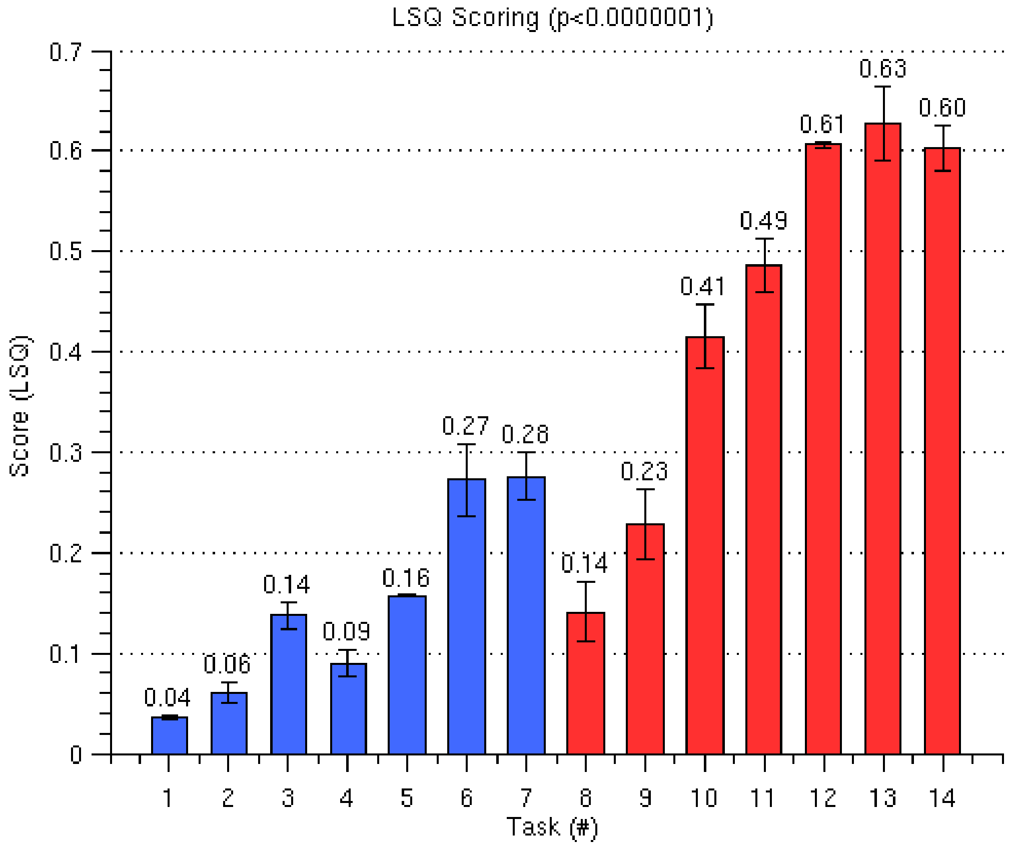


Figure 14: Mean LSQ score for each task. The task numbers correspond to those in Table 1, and the best scores were obtained for line graphs, with the leader node as the head of the line, forming ellipse formations (Task 1). The worst was when trying to form a wedge with a star network, with the central node as the leader node (Task 13).

It can be inferred from Figure 14 that the task of moving a interaction topology to an ellipse formation is generally easier than moving the same interaction topology to a wedge. The first seven bars correspond to the ellipse while the last seven correspond to the wedge. The LSQ scores suggest that the difference in scores between two interaction topologies is largely independent of the target formation. Note that we have to be careful and use the modifier "largely" here, since not all pairwise comparisons yield statistically significant differences. However, almost without exception L_7 networks have a statistically significantly lower (better) score than C_7, K_7, and S_7 networks regardless of target formation. Similarly, S_7 networks have in almost all cases a statistically significant higher (worse) score than all other networks. As a consequence, one conclusion that can be drawn is that in terms of LSQ scores, line formations are to be preferred and star formations are to be avoided.

Figure 15 is a histogram of the mean rating scores for each task. We observe a similar trend as before, where, independent of the target formation, line topologies are mostly rated as being easier to control than all other topologies. Star topologies are mostly rated as the hardest topologies to move into a particular formation. The $p = 0.0138$ value is larger than the p-value of the LSQ scores, so we see less statistically significant differences between the tasks. For example, the $L_{7,c}$ network does not have a statistically significant advantage over C_7, K_7, or S_7 networks with respect to the rating scores.

Finally, Figure 16 is a histogram of the mean workload scores for each task, which encodes how difficult the task was in terms of the effort involved. Each bar is divided into six parts encoding (starting from the bottom) the mental, physical, and time demands, as well as the levels of performance, effort, and frustration reported by the participant. The size of each sub-bar corresponds to the magnitude that each measured response contributes to the total workload score. We observe a similar pattern in the workload scores compared to the rating scores. However, the $p = 0.0256$ value is larger for the workload score than for the rating and LSQ scores, so we, again, see less statistically significant differences between the tasks.

From Figures 14, 15, and 16, it is clear that, as expected, some interaction topologies were significantly more difficult to control than others. For example, we directly see that line topologies are easier to control than star graphs. However, to make these types of observations stand on a more firm mathematical footing, we first need to discuss the actual single-leader network dynamics used in the experiments and their corresponding controllability properties. We will return to the results of the user study once this has been done.

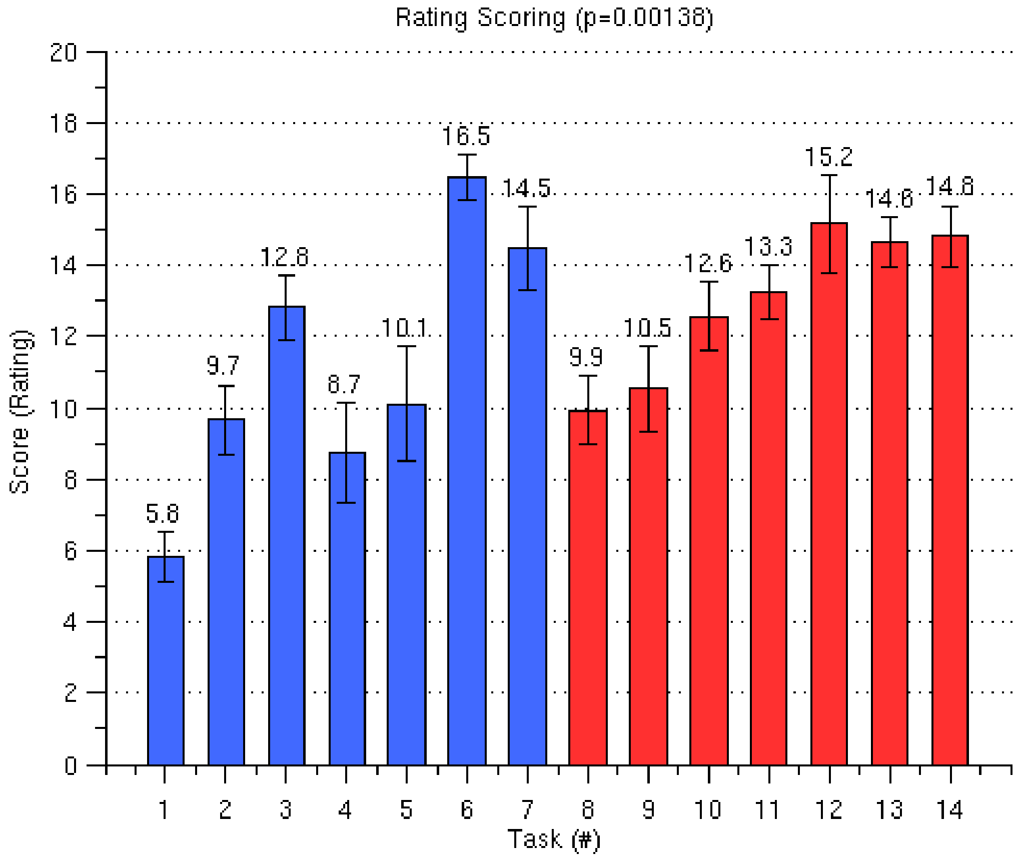


Figure 15: Mean rating score of the perceived difficulty for each task, as enumerated in Table 1. Line graphs (Tasks 1,2,3,8,9,10) are generally perceived to be easier to control while star graphs (Tasks 6,7,13,14) are harder. The differences between target formations (ellipses and wedges) are not very pronounced.

3.2 Properties of the Single-leader Network

Ultimately, we want to tie the results of the user study to established control and graph theoretic properties. Since these properties are directly tied to how the single-leader network behaves under input, a strong connection between these properties and how amenable the network is to user guidance would endow us with a set of tools for enabling effective human-swarm interactions. As a first step, we must fully describe the single-leader network dynamics, i.e., the HSI control structure. Next, we define three fundamental control and graph theoretic properties–the rank of the controllability matrix, the node centrality measures, and the network centralization coefficient–such that we can correlate these properties to the results of the user study.

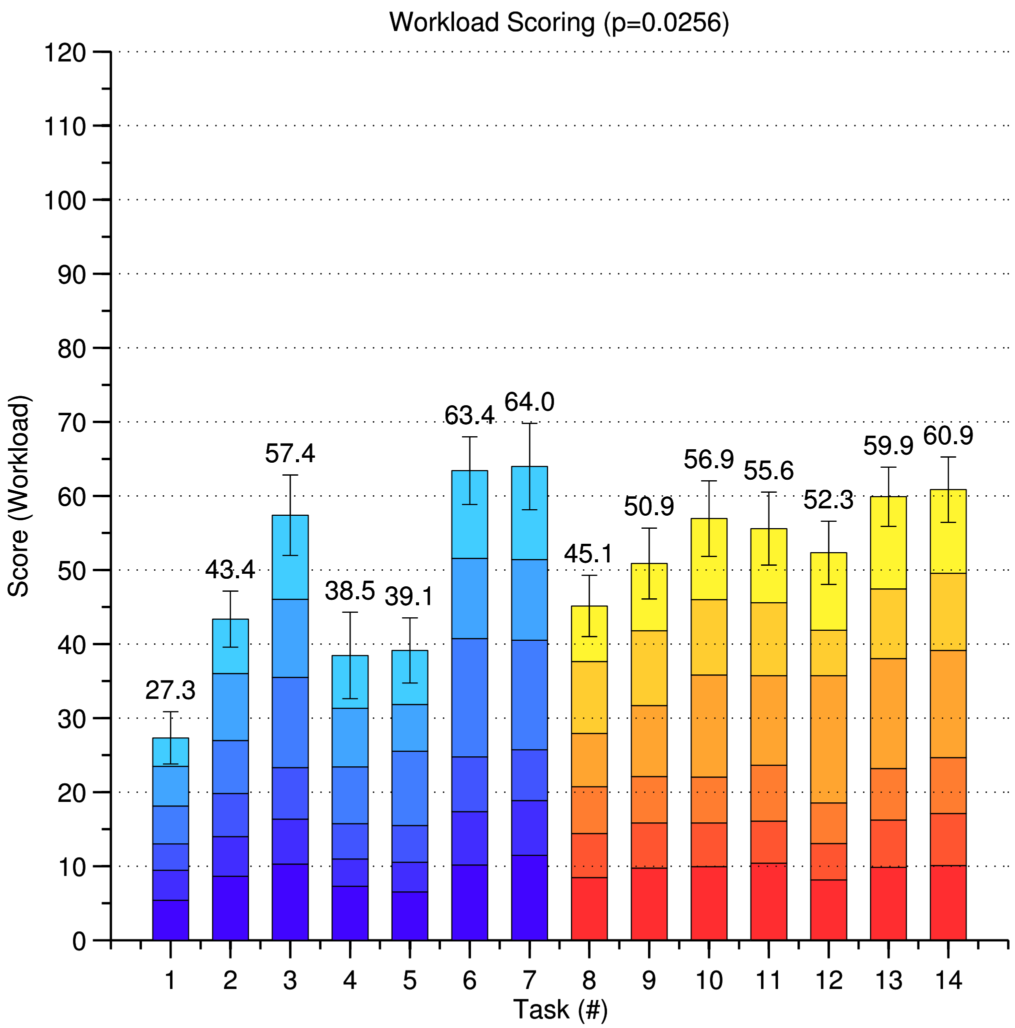


Figure 16: Mean workload score for each task. The differences between tasks are less pronounced than for the LSQ results and the rating scores.

3.2.1 Single-leader Network Dynamics

We begin our description of the single-leader network dynamics by defining the low level dynamics that each robot executes locally. Given n robots, let $p_i(t) \in \mathbf{R}^2$ be the planar position of robot i at time t, $i = 1, \ldots, n$. As the robots can typically measure the relative displacements to adjacent robots, i.e., robot i can measure $p_j(t) - p_i(t)$ if robot j is a neighbor to robot i in the network, a standard interaction law, that has been proposed repeatedly in the community (see, for example [17, 65]), is

$$\dot{p}_i(t) = u_i(t) = -\sum_{j \in N(i)} w(p_i(t), p_j(t))(p_i(t) - p_j(t)), \ i = 1, \ldots, N-1, \tag{47}$$

for all robots but the leader. We use the convention that the first $N-1$ robots are followers and robot n is the leader. $N(i)$ is once again the set of neighbors to robot i and $w(p_i, p_j)$ is a scalar weight that scales the contribution of robot j's position to the movement of robot i. Since we are interested in formation control, a typical choice of this weight is

$$w(p_i(t), p_j(t)) = \frac{\|p_i(t) - p_j(t)\| - \delta_{ij}}{\|p_i(t) - p_j(t)\|}, \tag{48}$$

and δ_{ij} is the desired distance between robots i and j. Consequently, robot i will move towards robot j if their distance is greater than δ_{ij}, while they will move apart from each other if it is less than δ_{ij}.

The underlying interaction topology is modeled as an undirected, static graph $\mathcal{G} = (V, E)$. Although we only consider undirected networks in this chapter, it can be useful to associate an orientation with the underlying graph. The way this construction works is by defining a mapping $\sigma : E \to \{-1, 1\}$ with each edge, thus assigning it an orientation. v_i is the tail of edge $(v_i, v_j) \in E$ if $\sigma((v_i, v_j)) = -1$ while it is the head if $\sigma((v_i, v_j)) = 1$, with the interpretation that $\sigma((v_i, v_j)) = -\sigma((v_j, v_i))$. The corresponding, directed graph is $\mathcal{G}_\sigma = (V, E_\sigma)$ and if the edges in this graph are numbered from 1 to m, then the $n \times m$ incidence matrix, $D(\mathcal{G}_\sigma)$, is given by

$$[D(\mathcal{G}_\sigma)]_{ij} = \begin{cases} 1 & \text{if } v_i \text{ is the head to edge } j \\ -1 & \text{if } v_i \text{ is the tail to edge } j \\ 0 & \text{otherwise.} \end{cases} \tag{49}$$

Moreover, if we associate a weight with each edge, we can let W be the $m \times m$ diagonal weight matrix, where m is the number of edges, and each entry along the diagonal corresponds to the corresponding edge weight. The weighted graph Laplacian L_w then takes the following form:

$$L_w(\mathcal{G}, p) = D(\mathcal{G}_\sigma) W(p) D(\mathcal{G}_\sigma)^T. \tag{50}$$

Note here that L_w does not depend on σ even though the incidence matrix does, meaning that the graph Laplacian is orientation independent.

The ensemble dynamics of the followers in the single-leader network can be derived by partitioning L_w as

$$L_w(\mathcal{G}, p) = \begin{bmatrix} L_f(p) & l(p) \\ l(p)^T & \lambda(p) \end{bmatrix}, \tag{51}$$

where $L_f(p)$ is a $(n-1) \times (n-1)$ matrix, $l(p)$ is a $(n-1)$ vector, and $\lambda(p)$ is a scalar. Using this notation, the stacked follower dynamics in Equation 47 are

$$\dot{x}_j = -L_f(p)x_j - l(p)v_j, \ j = 1, 2, \tag{52}$$

where $x_j \in \mathbf{R}^{n-1}$ is the state of all follower robots and $v_j \in \mathbf{R}$ is the exogenous user input in dimension j. Given these single-leader network dynamics, we are ready to answer two questions about a given network: *Is the network controllable?* and *How controllable is the network?*. We will use the rank of the controllability matrix and the node centrality measures applied to the leader to answer these questions.

3.2.2 Controllability

To understand what is meant my controllability, we first need to take a short foray into the field of linear systems theory. A linear, time-invariant (LTI) system is given by

$$\dot{z} = Az + Bv, \tag{53}$$

where $z \in \mathbf{R}^q$ is the state of the system and $v \in \mathbf{R}^k$ is the input. Such a system is completely controllable if and only if it is possible to drive the system from any initial state to any final state.

The controllability of LTI systems can be analyzed through the controllability matrix

$$\Gamma = \begin{bmatrix} B & AB & \dots & A^{(q-1)}B \end{bmatrix}, \tag{54}$$

The rank of this matrix tells us how controllable the system is. If the rank is q then the system is completely controllable, while if it is rank deficient, then the rank is equal to the dimension of the so-called *controllable subspace* which essentially is the subspace in which the control inputs can drive the system between arbitrary states. Consequently, the rank of the controllability matrix seems like a promising candidate for understanding which single-leader networks are easily controlled.

In fact, if we (for now) let the edge weights be identically equal to one, we get the ensemble dynamics

$$\dot{x}_j = -L_f x_j - l u_j,\ j = 1, 2, \tag{55}$$

which is indeed a linear, time-invariant system. Consequently, we can construct the network controllability matrix

$$\Gamma_L = \left[\begin{array}{cccc} -l & (-L_f)(-l) & \dots & (-L_f)^{(n-2)}(-l) \end{array} \right], \tag{56}$$

and we are going to use the rank of this matrix, $\rho(\Gamma_L)$, as one of the candidate measures of how easy the network is to control. The reason why this is a valid notion is that even though our system has non-unity weights on the edges, the linearized dynamics around the desired inter-robot distances δ_{ij} is closely related to Equation 55.

To see this, assume that we decouple the weights along the different dimensions. In other words, consider the dimensionally-decoupled system dynamics

$$\dot{x}_j = -L_f(p_j) x_j - l(p_j) u_j,\ j = 1, 2. \tag{57}$$

This equation is different from Equation 118, because L_f and l depend on p_j instead of, as before, on the full state vector p. The decoupled weights are now assumed to be given by

$$w(p_{i,j}(t), p_{k,j}(t)) = \frac{|p_{i,j}(t) - p_{k,j}(t)| - \delta_{ijk}}{|p_{i,j}(t) - p_{k,j}(t)|},\ j = 1, 2. \tag{58}$$

Linearizing this system along the state-input pair $(\hat{x}_j, 0)$, where $\hat{x}_j$ is such that the edge distances are exactly equal to to the desired distances, yields

$$\begin{aligned} \dot{\tilde{x}}_j \quad = \quad & - \left.\frac{\partial(L_f(x_j)x_j + l(x_j)u_j)}{\partial x_j}\right|_{(\hat{x}_j,0)} \tilde{x}_j \\ & - \left.\frac{\partial(L_w(x_j)x_j + l(x_j)u_j)}{\partial u_j}\right|_{(\hat{x}_j,0)} u_j,\ j = 1, 2. \end{aligned} \tag{59}$$

With a fair amount of algebra, we can show that this is equal to the linear system in Equation 55:

Theorem 1. *Let $\dot{x}(t) = -L_w x(t)$ be the dynamics of the unforced, non-linear system, where for the purposes of this theorem only $x(t) = [x_1(t), \dots, x_n(t)]^\top$. If and only if the following two properties hold:*

1. *Consensus in the j-th dimension of the d-dimensional Euclidean space is given by,*

$$\dot{x}_{i,j}(t) = -\sum_{k \in N(i)} w(x_{i,j}(t), x_{k,j}(t))(x_{i,j}(t) - x_{k,j}(t)) \tag{60}$$

   and,

2. *The following property for the partial derivative of $w(x_i(t), x_j(t))$ holds:*

$$\frac{\partial w(x_{i,j}(t), x_{k,j}(t))}{\partial x_{i,j}(t)}(x_{i,j}(t) - x_{k,j}(t))\bigg|_{\hat{x}_{0,j}} = 1. \tag{61}$$

Then the linearized system will be equivalent to the linear system, $\dot{x}(t) = -Lx(t)$.

Proof. Property 1 holds, because we already assumed that we can decouple the weights along the two dimensions. Property 2 also holds if the edge weights are defined as in Equation 48 and decoupled along each dimension. Recall that we chose the equilibrium point $(\hat{x}_0)$ to be where each robot is in the target configuration with $\|x_i - x_k\| = \delta$. Similarly, the equilibrium point in the j-th dimension is $(\hat{x}_{0,j})$ with $\|x_{i,j} - x_{k,j}\| = \delta_j$.

$$\begin{aligned}
(61) &= \frac{\partial w(x_{i,j}, x_{k,j})}{\partial x_{i,j}}(x_{i,j} - x_{k,j})\bigg|_{(\hat{x}_{0,j})} \\
&= \frac{\partial}{\partial x_{i,j}}\left(\frac{\|x_{i,j} - x_{k,j}\| - \delta_j}{\|x_{i,j} - x_{k,j}\|}\right)(x_{i,j} - x_{k,j})\bigg|_{(\hat{x}_{0,j})} \\
&= \left(\frac{\delta_j(x_{i,j} - x_{k,j})}{\|x_{i,j} - x_{k,j}\|^3}\right)(x_{i,j} - x_{k,j})\bigg|_{(\hat{x}_{0,j})} \\
&= \frac{\|x_{i,j} - x_{k,j}\|(x_{i,j} - x_{k,j})^2}{\|x_{i,j} - x_{k,j}\|^3} \\
&= \frac{(x_{i,j} - x_{k,j})^2}{|x_{i,j} - x_{k,j}|^2} = \frac{(x_{i,j} - x_{k,j})^2}{(x_{i,j} - x_{k,j})^2} = 1
\end{aligned} \tag{62}$$

We have shown that properties 1 and 2 hold; therefore, we have to show that consequently the linearized system is equal to the linear system of the form $\dot{x}(t) = -Lx(t)$. We can achieve this by linearizing the system along each dimension k around the equilibrium point $(\hat{x}_{0,j})$ in that dimension.

First we compute the Jacobian matrix (J) of the system at ($\hat{x}_{0,j}$),

$$\begin{aligned}
\dot{\tilde{x}}_j &= -\left.\frac{\partial(L_w x_j)}{\partial x_j}\right|_{(\hat{x}_{0,j})} \tilde{x}_j \\
&= -\frac{\partial}{\partial x_j}\left[DWD^T x_j\right]\tilde{x}_j \\
&= -\frac{\partial}{\partial x_j}\begin{bmatrix} f_1(x_{0,j}) \\ \vdots \\ f_n(x_{0,j}) \end{bmatrix}\tilde{x}_j = -\begin{bmatrix} \frac{\partial f_1(x_{0,j})}{\partial x_{1,j}} & \cdots & \frac{\partial f_1(x_{0,j})}{\partial x_{n,j}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n(x_{0,j})}{\partial x_{1,j}} & \cdots & \frac{\partial f_n(x_{0,j})}{\partial x_{n,j}} \end{bmatrix}\tilde{x}_j
\end{aligned} \tag{63}$$

The entries on the diagonal of J are all the partial derivatives $\frac{\partial}{\partial x_{i,j}} f_i(\hat{x}_{0,j})$ for $i = 1, \ldots, n$. We may isolate f_i from the graph Laplacian by premultiplying it by e_i^T, where the unit vector e_i is a zero vector with a 1 at the i-th position. We apply property 2, as well as, the fact that $w(x_{i,j}, x_{k,j})|_{(\hat{x}_{0,j})} = 0$, since $\|x_{i,j} - x_{k,j}\| = \delta_j$

$$\begin{aligned}
\frac{\partial f_i(\hat{x}_{0,j})}{\partial x_{i,j}} &= \frac{\partial}{\partial x_{i,j}}\left[e_i^T DWD^T x_j\right] \\
&= \frac{\partial}{\partial x_{i,j}}\left.\left[\sum_{k\in N(i)} w(x_{i,j}, x_{k,j})(x_{i,j} - x_{k,j})\right]\right|_{(\hat{x}_{0,j})} \\
&= \sum_{k\in N(i)} \frac{\partial}{\partial x_{i,j}}\left.\left[w(x_{i,j}, x_{k,j})(x_{i,j} - x_{k,j})\right]\right|_{(\hat{x}_{0,j})} \\
&= \sum_{k\in N(i)} \left[\frac{\partial w(x_{i,j}, x_{k,j})}{\partial x_{i,j}}(x_{i,j} - x_{k,j})\right. \\
&\qquad + \left.\left. w(x_{i,j}, x_{k,j})\frac{\partial(x_{i,j} - x_{k,j})}{\partial x_{i,j}}\right]\right|_{(\hat{x}_{0,j})} \\
&= \sum_{k\in N(i)} (1 + 0) = [\Delta(\mathcal{G})]_{ik}
\end{aligned} \tag{64}$$

The i-th entry of the diagonal of J corresponds to the degree of robot i in $\mathcal{G}$. The off-diagonal entries of J are the partial derivatives $\frac{\partial}{\partial x_{i,j}} f_k(\hat{x}_{0,j})$, where $i \neq k$. We note that a contribution from the derivative with respect to $x_{i,j}$ appears if and only if $i \in N(k)$.

$$\begin{aligned}
\frac{\partial f_k(\hat{x}_{0,j})}{\partial x_{i,j}} &= \frac{\partial}{\partial x_{i,j}} \left[e_j^T DWD^T x_j\right] \\
&= \frac{\partial}{\partial x_{i,j}} \left[\sum_{l \in N(k)} w(x_{k,j}, x_{l,j})(x_{k,j} - x_{l,j})\right]\Bigg|_{(\hat{x}_{0,j})} \\
&= \sum_{l \in N(k)} \frac{\partial}{\partial x_{i,j}} \left[w(x_{k,j}, x_{l,j})(x_{k,j} - x_{l,j})\right]\Bigg|_{(\hat{x}_{0,j})} \\
&= \sum_{l \in N(k)} \left[\frac{\partial w(x_{k,j}, x_{l,j})}{\partial x_{i,j}}(x_{k,j} - x_{l,j})\right. \\
&\quad \left. + \; w(x_{k,j}, x_{l,j})\frac{\partial (x_{k,j} - x_{l,j})}{\partial x_{i,j}}\right]\Bigg|_{(\hat{x}_{0,j})} \\
&= \begin{cases} -1 & \text{if } i \in N(k), \\ 0 & \text{otherwise.} \end{cases} = -[A(\mathcal{G})]_{ik}
\end{aligned} \tag{65}$$

Each off-diagonal entry of J corresponds to the negative of the same entry in $A(\mathcal{G})$. Given the results for the diagonal and off-diagonal entries of J, we can write the following equation:

$$\begin{aligned}
\dot{\tilde{x}}_k &= -J\tilde{x}_j \\
&= -\left[\Delta(\mathcal{G}) - A(\mathcal{G})\right]\tilde{x}_j \\
&= -L\tilde{x}_j
\end{aligned} \tag{66}$$

We have shown that the Jacobian matrix from the linearization around the equilibrium point is equivalent to the unweighted graph Laplacian $L(\mathcal{G})$. □

Corollary 1. *Similarly, the system $\dot{x}_j = -L_f(p)x_j - l(p)u_j$, if the two properties holds, is equivalent to the system $\dot{x}_j = -L_f x_j - l u_j$ when linearized.*

The linearization of this system immediately follows from the theorem. Let $x(t) = [x_1(t), \ldots, x_{(n-1)}(t)]^\top$ be the vector of followers as before and recall that $u = x_n$.

$$\begin{aligned}\dot{\tilde{x}}_j =& -\left.\frac{\partial(L_{w,f}x_j + l_w u_j)}{\partial x_j}\right|_{(\hat{x}_{0,j},\hat{u}_{0,j})}\tilde{x}_j \\ & -\left.\frac{\partial(L_{w,f}x_j + l_w u_j)}{\partial u_j}\right|_{(\hat{x}_{0,j},\hat{u}_{0,j})}\tilde{u}_j \\ =& -\left.\frac{\partial(L_{w,f}x_j)}{\partial x_j}\right|_{(\hat{x}_{0,j})}\tilde{x}_j - \left.\frac{\partial(l_w u_j)}{\partial u_j}\right|_{(\hat{x}_{0,j},\hat{u}_{0,j})}\tilde{u}_j \\ =& -L_f\tilde{x}_j - l\tilde{u}_j,\ j = 1, 2\end{aligned} \tag{67}$$

We have shown that the dynamics of the linearized single leader network are equivalent to the dynamics of the unweighted single leader network; therefore, we are almost (recall, we have neglected the coupling terms in edge-weights) justified in considering the rank of the controllability matrix associated with the linearized system. The reason why we do not consider the controllability properties of the nonlinear system is that a number of connections have already been proven between the interaction topology and the linear dynamics in Equation 55 when it comes to establishing the controllability properties of the network dynamics, e.g. [79, 81]. And, as for the "almost" modifier, as the purpose of this chapter is ultimately about what users *considers* to be easy to control, the question whether or not $\rho(\Gamma_L)$ is an appropriate measure is ultimately an empirical question. And, as will be seen in the next section, $\rho(\Gamma_L)$ will indeed turn out to be a very strong measure of how easy it is for human users to control the underlying network.

3.2.3 Measures of Node Centrality

As an alternative to controllability, we will also use four classic centrality measure for simple graphs–degree centrality [13], closeness [26], betweenness [41], and eigenvector centrality [10]–to quantify the importance of the leader in each of the networks. Degree centrality is equal to the node degree (i.e., number of edges shared with other nodes),

$$C_D(v) = \deg(v), \text{where } v \in V. \tag{68}$$

$C_D(v)$ is simple to calculate, but it only measures the importance of the leader with respect to its immediate neighbors. For example, Figure 17 illustrates degree centrality in red. The black leader node has two neighbors; therefore, $C_D(v) = 2$.

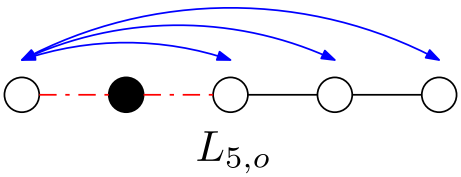


Figure 17: Degree centrality (dash-dotted) and betweeness centrality (solid) of the black leader node of this $L_{5,c}$ interaction topology are illustrated.

Closeness is defined in terms of the shortest paths from a node to all other nodes on the network.

$$C_C(v) = \sum_{u \in V \setminus v} 2^{-d_{\mathcal{G}}(v,u)}, \text{where } u, v \in V. \tag{69}$$

Closeness penalizes a node with long paths to other nodes. For the black leader node in Figure 17,

$$C_C(v) = \frac{1}{2^1} + \frac{1}{2^1} + \frac{1}{2^2} + \frac{1}{2^3} = 1.375,$$

but if the leader were assigned to be the middle node in the $L_{5,o}$ interaction topology, then the leader's closeness would be larger, $C_C(v) = 1.5$.

Betweenness is a measure of the fraction of shortest paths between any two nodes that passes through a particular node.

$$C_B(v) = \sum_{u \neq w \in V \setminus v} \frac{\sigma_{u,w}(v)}{\sigma_{u,w}}, \tag{70}$$

where $\sigma_{u,w}(v)$ is the total number of shortest paths between u and w that intersect v and $\sigma_{u,v}$ is the total number of shortest paths between u and w. Figure 17 illustrates in purple all shortest paths that pass through the black leader node. In the case of this $L_{5,o}$ interaction topology, the shortest paths are also the only paths; therefore, the fraction in Equation 70 is always one, such that $C_B(v) = 6$ for this particular interaction topology.

Eigenvector centrality measures the influence of a node on the network. This centrality measure is computed by first solving the eigenvalue problem, $A\mathbf{x} = \lambda_{max}\mathbf{x}$, where A is the adjacency matrix and λ_{max} its largest eigenvalue. The i-th entry of the vector $\mathbf{x}$ is the centrality score given to the i-th node in the network.

$$C_E(v) = x_i, \text{where } x_i \text{ is the } i\text{-th entry of } \mathbf{x}. \tag{71}$$

For the interaction topology shown in Figure 17, the second entry of the eigenvector corresponding to $\lambda_{\max} = 1.732$ is 0.5; therefore, for this particular interaction topology, $C_E(v) = 0.5$.

3.2.4 Newer Measures of Centrality

In addition to these four centrality measures, newer centrality measures have been developed for complex networks, such as social networks. We can use Bonacich's power centrality [11], Kleinberg's centrality [59], and Bonacich's alpha centrality [12] to quantify the importance of the leader in the network. If A is the adjacency matrix for some network, then Bonacich's power centrality is defined as,

$$C_P(A, \alpha, \beta) = \alpha(I - \beta A)^{-1}A\mathbf{1}, \tag{72}$$

where α is a scaling factor such that the sum of the centralities of all nodes is equal to the number nodes, and β is an attenuation parameters that needs to be less than the reciprocal of the largest eigenvalue of A. The power centrality of the i-th node is the i-th entry of $C_P(A, \alpha, \beta)$.

Kleinberg's centrality is computed in similar way as eigenvector centrality, except that the i-th node's centrality is equal to the i-th component of the principal eigenvector of $A^T A$. A is, as before, the adjacency matrix representing the network. For the interaction topology shown in Figure 17, $C_K(v) = 0.707$, where v is the leader node.

Finally, Bonacich's alpha centrality is a unique centrality measure, which allows us to externally influence the importance of a node. Our selection of a leader in a network implicitly makes the corresponding node more important than other nodes on the network. The importance of a node is given by some scalar, and all these scalars are stacked into the vector $\vec{e}$. For example, suppose that seventh node of a L_7 network is the leader, then one possible $\vec{e}$ is $[1, 1, 1, 1, 1, 1, 2]^T$. If α is a scalar that indicates how important the endogenous factors are compared to the exogenous factors given by $\vec{e}$, then alpha centrality is

$$C_\alpha(A, \alpha) = (I - \alpha A^T)^{-1}\vec{e}. \tag{73}$$

For the interaction topology shown in Figure 17, $\alpha = 0.25$, and $\vec{e} = [1, 1, 1, 1, 1, 1, 2]^T$, the seventh entry of C_α is 3.004, which is the leader's alpha centrality.

Given the importance of the leader in the leader-follower structure, we can expect that the measures of node centrality for the leader are another indicator of how difficult it is for a user to control a network.

3.2.5 Centralization

So far, we have focused on the centrality of the leader in a leader-follower network, but it may be useful to investigate centrality as a property of the whole network. Network centralization is a measure of how central the node with the highest centrality score is compared to all other nodes [42]. Suppose, we are given some centrality measure, $C(v)$ (for example, $C_D(v)$), then $v^* = \operatorname{argmax}_{v \in V} C(v)$ is the node with the highest centrality measure. If the sum of differences between $C(v^*)$ and the centrality score of each node on a particular network is $\sum_{v \in V} C(v^*) - C(v)$, then let C^* be the largest of such sum of differences over all possible networks with the same number of nodes. Consequently, network centralization, $\mathcal{C}$, is defined as

$$\mathcal{C} = \frac{\sum_{v \in V} C(v^*) - C(v)}{C^*}. \tag{74}$$

As an example, suppose we want to calculate the network centralization of a L_7 network with respect to degree centrality. The largest centrality measure on L_7 is $C_D(v^*) = 2$. The sum of differences is maximized on a S_7 network, where the center node has a degree centrality of 6, while all peripheral nodes have a degree centrality of 1. Therefore, the largest sum of differences is $C_D^* = 30$. At this point, we can compute the network centralization,

$$\mathcal{C}_D = \frac{\sum_{v \in V} C_D(v^*) - C_D(v)}{C_D^*} = \frac{2}{30} = 0.0\bar{6} \tag{75}$$

A small $\mathcal{C}_D$ indicates that the nodes of a L_7 network have similar degree centrality.

While network centralization is computed independent of our choice of leader, networks with little centralization may uniformly propagate input signals, which may reduce the difficulty in interacting with the network as a whole. Consequently, we can expect it to be another metric for how difficult it is for a user to control a network.

3.3 Analysis and Conclusions

We are now ready to connect the single-leader network characterization – the rank of the controllability matrix, the node centrality measures, and the network centralizations – to the user study results. The average rating, average LSQ error, total workload score, the rank of the controllability matrix, and the first four node centrality measures, C_D, C_C, C_B, and C_E are summarized in Table 2, while Table 3 adds the three other node centralities, C_P, C_K, and C_α. The line graph, L_7, with the leader node located at the head of the network, is completely controllable and, intuitively, it should be easy to move the followers into position by pulling the leader around. This observation is supported by the user study data. It is important to note that controlling this particular L_7 graph can be easily accomplished by the user independent of the target formation. In fact, if the leader in the L_7 graph is offset from the head of the network, the score, ratings, and workload measures slightly increase in comparison, even though the controllability remains constant. In this case, examining the measures of node centrality helps us explain for the difference. The leader in the $L_{7,h}$ network has a lower node centrality score than the leader in the $L_{7,o}$ network. The results indicate that a less important (or influential) leader in the network is beneficial for controlling networks in tasks require robots to be moved into a specific formation (as opposed to driving the network from point A to point B collectively).

Selecting a leader in the center of the L_7 graph cuts the rank of the controllability matrix in half, while again increasing the reported measures in comparison to the $L_{7,o}$ network. We can conclude from this observation that a decrease in rank results in an increase in the reported measures. The rank of its controllability matrix is the same as that of the C_7 network; however, its leader's centrality score is larger or equal than the centrality score of the C_7 network's leader. Therefore, we can expect that the C_7 network is easier for a user to control than the $L_{7,c}$ network. This conclusion is validated by the user study data in those cases where the difference is statistically significant.

The complete graph K_7 is rank deficient due to its high degree of symmetry. In fact, the rank of the controllability matrix is 1; meaning that the only the network's center of mass can be controlled in this configuration. Consequently, it is impossible for the participant

Table 2: Mean LSQ, rating, and workload scores with controllability matrix rank, ρ, and node centrality measures for each task.

Task	Network	Target	ρ	C_D	C_C	C_B	C_E	LSQ	Rating	Workload
1	$L_{7,h}$	Ellipse	6	1	0.984	0	0.191	0.035	5.83	27.33
2	$L_{7,o}$	Ellipse	6	2	1.469	10	0.354	0.061	9.65	43.37
3	$L_{7,c}$	Ellipse	3	2	1.750	18	0.500	0.137	12.82	57.40
4	C_7	Ellipse	3	2	1.750	6	0.378	0.090	8.72	38.46
5	K_7	Ellipse	1	6	3.000	0	0.378	0.157	10.11	39.14
6	$S_{7,c}$	Ellipse	1	6	3.000	30	0.707	0.273	16.47	63.42
7	$S_{7,p}$	Ellipse	2	1	1.750	0	0.289	0.276	14.46	63.98
8	$L_{7,h}$	Wedge	6	1	0.984	0	0.191	0.141	9.93	45.14
9	$L_{7,o}$	Wedge	6	2	1.469	10	0.354	0.229	10.54	50.88
10	$L_{7,c}$	Wedge	3	2	1.750	18	0.500	0.415	12.57	56.94
11	C_7	Wedge	3	2	1.750	6	0.378	0.486	13.26	55.59
12	K_7	Wedge	1	6	3.000	0	0.378	0.606	15.16	52.32
13	$S_{7,c}$	Wedge	1	6	3.000	30	0.707	0.627	14.64	59.90
14	$S_{7,p}$	Wedge	2	1	1.750	0	0.289	0.602	14.81	60.86

Table 3: Mean LSQ, rating, and workload scores with Bonacich ($\beta = 1$), Kleinberg, and alpha centrality ($\alpha = 0.25$) measures for each task.

Task	Network	Target	C_P	C_K	C_α	LSQ	Rating	Workload
1	$L_{7,h}$	Ellipse	0.000	0.354	2.536	0.035	5.83	27.33
2	$L_{7,o}$	Ellipse	0.441	0.707	3.004	0.061	9.65	43.37
3	$L_{7,c}$	Ellipse	1.764	1.000	3.134	0.137	12.82	57.40
4	C_7	Ellipse	1.000	1.000	3.155	0.090	8.72	38.46
5	K_7	Ellipse	1.000	1.000	1.600	0.157	10.11	39.14
6	$S_{7,c}$	Ellipse	1.517	1.000	5.600	0.273	16.47	63.42
7	$S_{7,p}$	Ellipse	0.885	0.166	3.100	0.276	14.46	63.98
8	$L_{7,h}$	Wedge	0.000	0.354	2.536	0.141	9.93	45.14
9	$L_{7,o}$	Wedge	0.441	0.707	3.004	0.229	10.54	50.88
10	$L_{7,c}$	Wedge	1.764	1.000	3.134	0.415	12.57	56.94
11	C_7	Wedge	1.000	1.000	3.155	0.486	13.26	55.59
12	K_7	Wedge	1.000	1.000	1.600	0.606	15.16	52.32
13	$S_{7,c}$	Wedge	1.517	1.000	5.600	0.627	14.64	59.90
14	$S_{7,p}$	Wedge	0.885	0.166	3.100	0.602	14.81	60.86

to move K_7 into a wedge formation. The results from the user study confirm this fact. In contrast, from Table 2 demonstrates that the reported measures are low for tasks 4 and 5, where the user has to move a C_7 and K_7 network into an elliptical formation. We can conclude that the perceived difficulty is only low for such rank deficient networks, if the target formation is analogous to the natural formation of the network (e.g., C_7, a cycle, to an ellipse) and the user can avoid driving the system into an uncontrollable subspace.

In contrast, all tasks involving a S_7 network using either a peripheral or center leader are perceived as being very difficult. Not only are these networks rank deficient, but any input quickly drives the system into an uncontrollable subspace. Interestingly, the extra rank of the $S_{7,p}$ network has no advantage over the fully rank deficient $S_{7,c}$ network.

In fact, the rank of the controllability matrix is negatively correlated to the scores as shown in Table 4, which supports the claim that *a configuration with a higher rank was almost without exceptions given a better score than a configuration with a lower rank.* We can conclude that the rank of the controllability matrix is a strong predictor of how easy it is to control a team of mobile robots. Consequently, it is the first property one should consider when choosing an easily user-guided single-leader network. As a corollary, symmetric configurations (e.g., star graphs and complete graphs) are not particularly well-suited for human control. The node centrality measures of the leader are positively correlated (e.g., for C_E, $r^2_{\text{Rating}} = 0.58$, $r^2_{\text{Workload}} = 0.54$) to the scores as shown in Table 4. In other words, a small leader-node centrality is another good indicator that a particular network of mobile robots is easier to control. In fact, *given two configurations with the same ranks, all centrality measures serve as reasonable tie breakers for which network is easiest to control.* Finally, Table 5 shows that there is also a positive correlation between network centralization and the scores, which means we can use the network centralization as another reasonable tiebreaker. It is important to note, however, that rank, node centrality, and network centralization are by no means *absolute* metrics for the difficulty of controlling a given network, but good predictors of the perceived difficulty.

Table 4: Correlation coefficient of mean LSQ, rating, and workload scores versus controllability matrix rank and each node centrality measure.

Score	ρ	C_D	C_C	C_B	C_E	C_P	C_K	C_α
LSQ	-0.604	0.352	0.495	0.181	0.356	0.441	0.123	0.241
Rating	-0.733	0.432	0.616	0.420	0.588	0.648	0.172	0.456
Workload	-0.539	0.164	0.366	0.472	0.541	0.634	0.124	0.524

Table 5: Correlation coefficient (r^2) of mean LSQ, rating, and workload scores versus centralization measure with respect to degree centrality, closeness, betweenness, and eigenvector centrality.

Score	$\mathcal{C}_D$	$\mathcal{C}_C$	$\mathcal{C}_B$	$\mathcal{C}_E$
LSQ	0.437	0.342	0.324	0.163
Rating	0.644	0.565	0.548	0.345
Workload	0.660	0.637	0.630	0.537

3.3.1 Recommendations

The results of this user study and their analysis allow us to make recommendations about how to construct single-leader networks that are amenable to human control. First of all, the single-leader network needs to be controllable, which means that interaction topologies that are symmetric with respect to the leader node should be avoided. Once a set of controllable single-leader networks has been constructed, we recommend using either closeness centrality or Bonacich's power centrality to evaluate the centrality of the leader on each of these networks. Choosing a network with a lower leader centrality is beneficial to reducing the difficulty of controlling a single-leader network. Alternatively, it is also beneficial to pick a network with a small (non-zero) network centralization with respect to degree centrality. Ultimately, choosing a single-leader network that is controllable, has small leader centrality, and has small network centralization seems to ensure that the network will be amenable to human control for the tasks in this user study.

CHAPTER IV

NEW INTERACTIONS WITH SWARMS OF ROBOTS

This fourth chapter introduces three new HSI abstractions: broadcast control, predator-prey interactions, and deformable media. First, we demonstrate how users can interact with a swarm of mobile robots by broadcasting as single input signal to all robots. This type of human-swarm interaction (HSI) has been previously introduced in Section 2.4.2 for guiding a swarm of robots to a stack of packages for sorting. In Section 4.1, we will demonstrate that users can also use broadcast input signal to separate a heterogeneous swarm of robots into homogeneous clusters. The focus will also be on applying the framework from Chapter 2 to this HSI control structure. Section 4.2 introduces a new task: a user needs to deploy a parameterized formation of N robots (predators) to capture an evasive, parameterized robot (prey). The focus of this work is to provide the user with an algorithm to pick the minimum number of predators to capture the prey based on the parameters picked for the hunt. Consequently, we introduce and provide a preliminary answer to an important HSI question, *How many robots are required for the user to be able to successfully complete a task?* We introduce another important question in Section 4.3: *What type of input controllers affords guiding robotic swarms into specific shapes?* In Chapter 3, we discussed how single-leader networks are generally difficult to guide into specific geometric shapes. Consequently, we introduce a new HSI abstraction that is based on providing users with a deformable medium–clay. Deformable media, such as clay, afford actions associated with forming shapes, such as stretching, splitting, bending, and merging unlike traditional input controllers, such as joysticks.

4.1 Separating Heterogeneous Swarms with a Broadcast Signal

Broadcast control was previously introduced in Section 2.4.2 as an example of an HSI control structure to guide a swarm of robots to a rendezvous location. In this section, we focus on the task of separating the swarm into separate, connected components or “clusters” that correspond to all robots of a particular class (where this class may correspond to a unique capability) being co-located, but completely separated from robots belonging to another class. This problem has not been widely studied in robotics aside from [47], but it is well known that in physical processes involving granular mixtures, granules of different sizes segregate under external perturbations, which is our source of inspiration for broadcast control [2, 22, 75]. If a user can achieve granular separation by shaking a cereal box, then this interaction may be useful for swarms as well.

Once again, we let the robots execute weighted consensus dynamics, where the robots are attracted to each other, rather than repulsed (cohesion). Robots within the same class have the same weight, while robots from different classes have different weights. Different weights in the dynamics are an analog for the robots having different sizes. We will show that a user can apply an exogenous control signal to this swarm that will completely separate robots from each other if they are in different classes, while robots from the same class remain together. The key is that these multi-agent systems are heterogeneous, meaning that robots with different characteristics will react differently to the same input signal. This difference in response to an input signal is exploited to achieve a global behavior for the swarm.

As we have done in [28], we derive an external signal for separating two classes of robots and then show that it is also possible to separate three classes of robots, while assuming that the initial position of all robots is the same and that the separation happens along a single dimension. Next, we generalize to separating M classes of robots. Once we have made this generalization, we will continue to generalize our results to show that we can also compute a separation signal for robots moving in multiple dimensions and to the case when the robots start separating while co-located in a small ball, i.e. not coincident as previously assumed. Last, we conclude with some experimental verification in the form of simulations. Along the way, we will also use attention, effort, and scalability, as we have done in [32], to characterize the user's interaction with the swarm during this separation task .

4.1.1 Separating Two, Three, and M Classes

Suppose that a network of robots is comprised of M types of robots, belonging to one of the classes in $\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_M\}$. The problem is how to separate these M classes of robots from each other without separating robots from the *same* class. As an initial way of approaching this problem, let us assume that the robots in the different class somehow carry different weights, such that they respond differently to an external signal. In addition, assume that these robots are all running a local, forced agreement protocol, in the sense that

$$\dot{x}_i(t) = v_i(t) = \gamma_{\pi(i)} \left(\sum_{j \in N(i)} (x_j(t) - x_i(t)) + v(t) \right), \tag{76}$$

where x_i is the position of robot i, $\gamma_{\pi(i)}$ is a scalar weight, $N(i)$ is the set of neighbors that robot i has in the network, and v is an exogenous user input signal. The set of neighbors is defined by the condition $\|x_j - x_i\| \leq \Delta$, i.e. $j \in N(i)$ and $i \in N(j)$ if robots i and j are close enough to each other. It is important to note that a neighborhood, $N(i)$, is not a function of class, since an robot is not aware of its neighbor's class. The key object in Equation (76) is the class membership function $\pi : \mathcal{N} \to \mathcal{C}$, where $\mathcal{N} = \{1, \ldots, N\}$ is the set of all robots, i.e the function π maps robot i into one of the M classes with weights $\gamma_1, \gamma_2, \ldots, \gamma_M$. The notion that these classes weigh differently is encoded in the following two properties:

1. Each class has a unique weight; otherwise, any two classes with the same weight can be merged into a single class.

2. The weights can be ordered in ascending order,

   $$0 < \gamma_1 < \gamma_2 < \ldots < \gamma_M,$$

   by simply relabeling $\mathcal{C}$ if necessary.

We aim to generate a separating signal v that ensures that the robots are separated from each other, as is illustrated in Figure 18. This signal will be broadcast simultaneously to all robots. As a first step towards deriving such a separating signal, let us first assume that a) the robots are all scalar, i.e., that $x_i \in \mathbb{R}$, and b) that all robots start at the same position, i.e., $x_i(0) = x_j(0), \forall (i,j)$. We will first consider the two class case, then the three class

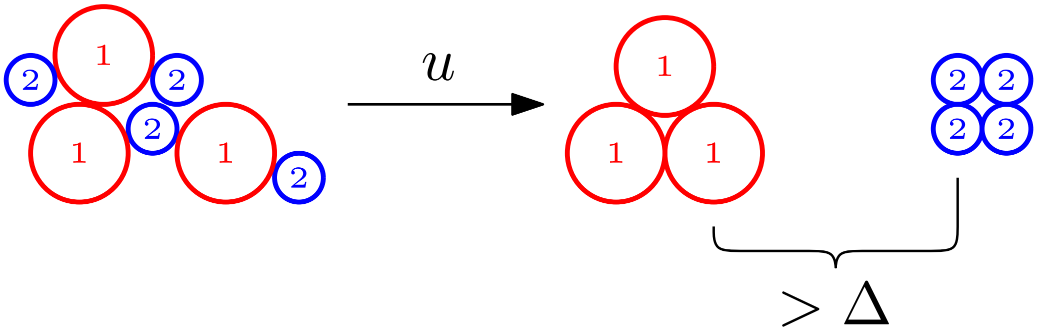


Figure 18: Separation of Δ between two classes using an external signal v applied to all robots simultaneously.

case, and finally, we will generalize to M classes. The two class case will provide us with a way to separate two classes, while the three class case will provide us with a strategy for separating more than two classes by using the results from the two class case. It is then possible for us to use the results from these two cases to generalize to M classes.

4.1.1.1 Two Classes

Under the two assumption, $x_i(0) = x_j(0), \forall(i,j)$ and $x_i \in \mathbb{R}$, we will derive a constant, scalar separating signal that is guaranteed to achieve a desired separation of greater than Δ between the two classes. In fact, assume that there are N_1 robots of $\mathcal{C}_1$ and N_2 robots of $\mathcal{C}_2$. Since all robots within a class start at the same position and execute the same dynamics, they will always stay together under any input signal v. As such, if we let χ_i be the position of any member of $\mathcal{C}_i$, $i = 1, 2$, we can let $d_{12} = \chi_2 - \chi_1$ denote the distance separating the two classes. And, the swarm dynamics for the two classes become

$$\dot{\chi}_1 = \gamma_1(N_2 d_{12} + v), \quad \dot{\chi}_2 = \gamma_2(-N_1 d_{12} + v)$$

or

$$\dot{d}_{12} = \dot{\chi}_2 - \dot{\chi}_1 = (\gamma_2 - \gamma_1)v - (\gamma_2 N_1 + \gamma_1 N_2)d_{12}. \tag{77}$$

Now, assume that the robots are no longer connected if an inter-robot distance is greater than Δ, i.e. if $i \in \mathcal{C}_1, j \in \mathcal{C}_2$, then $j \notin N(i), i \notin N(j)$ when $|x_j - x_i| > \Delta$.

Theorem 4.1.1. *v is a scalar, constant separating signal if*

$$v > \frac{(\gamma_2 N_1 + \gamma_1 N_2)\Delta}{\gamma_2 - \gamma_1}, \tag{78}$$

which ensures that $\dot{d}_{12} > 0$ when $d_{12} \in [0, \Delta]$, i.e., guarantees that all robots belonging to different classes are completely separated by a distance greater than Δ.

Proof. The proof follows directly from applying $\dot{d}_{12} > 0$ to Equation (77) and solving for v to get

$$v > \frac{(\gamma_2 N_1 + \gamma_1 N_2) d_{12}}{\gamma_2 - \gamma_1}.$$

$u > 0$ is needed to start the separation process when $d_{12} = 0$. Since d_{12} increases monotonically on the interval $[0, \Delta]$, it is sufficient to suppose that $d_{12} = \Delta$ and require that

$$v > \frac{(\gamma_2 N_1 + \gamma_1 N_2)\Delta}{\gamma_2 - \gamma_1}$$

is applied on the interval.

Then, a scalar, constant separating signal v is

$$v = \frac{(\gamma_2 N_1 + \gamma_1 N_2)(\Delta + \epsilon)}{\gamma_2 - \gamma_1}, \tag{79}$$

where $\epsilon > 0$. □

Moreover, we can explicitly compute the duration T_s for which v needs to be applied to ensure that no two robots from different classes are connected.

Corollary 2. *To separate the two classes, v needs to be applied for a duration of*

$$T_s = \frac{\ln\left(\frac{\Delta + \epsilon}{\epsilon}\right)}{\gamma_1 N_2 + \gamma_2 N_1}, \tag{80}$$

where $\epsilon > 0$.

Proof. The distance separating the two classes at time t is

$$\begin{aligned} d_{12}(t) &= \int_0^t e^{-(\gamma_2 N_1 + \gamma_1 N_2)(t-s)} (\gamma_2 N_1 + \gamma_1 N_2)(\Delta + \epsilon) ds \\ &= (\Delta + \epsilon)\left(1 - e^{-(\gamma_2 N_1 + \gamma_1 N_2)t}\right), \end{aligned} \tag{81}$$

where $d_{12}(0) = 0$ is the initial condition. At time T_s, we know that $d_{12}(T_s) = \Delta$, so we can solve Equation (81) in terms of T_s to get Equation (80). Since $\dot{d}_{12}(T_s) > 0$, the two classes will be separated by a distance greater than Δ after T_s time. □

4.1.1.2 Three Classes

The next step is to find a separating signal v, which can completely separate robots belonging to three classes, $\mathcal{C}_1$, $\mathcal{C}_2$, and $\mathcal{C}_3$. We will use this result as a stepping stone for generalizing our results to M classes. With the addition of $\mathcal{C}_3$, we define d_{23} to be distance between robots in $\mathcal{C}_2$ and $\mathcal{C}_3$ and d_{13} to be the distance between robots in $\mathcal{C}_1$ and $\mathcal{C}_3$. Let $d_{13} = d_{12} + d_{23}$, such that $d_{13} > d_{12}$ and $d_{13} > d_{23}$ during the separation process, i.e. $t > 0$. Recall that the initial conditions are $d_{13}(0) = d_{12}(0) = d_{23}(0) = 0$.

The swarm dynamics for the three classes before separation are

$$\begin{aligned} \dot{\chi}_1 &= \gamma_1(N_2 d_{12} + N_3 d_{13} + v), \\ \dot{\chi}_2 &= \gamma_2(-N_1 d_{12} + N_3 d_{23} + v), \\ \dot{\chi}_3 &= \gamma_3(-N_1 d_{13} - N_2 d_{23} + v). \end{aligned} \tag{82}$$

Instead of finding a single separating signal v, we are going to be strategic and find a series of separating signals $v_M, \ldots, v_2$ that separate the classes by peeling off classes in descending order of their weights. For example, in the three class problem, v_3 will separate $\mathcal{C}_3$ from $\mathcal{C}_2$ and $\mathcal{C}_1$, and v_2 will separate $\mathcal{C}_2$ from $\mathcal{C}_1$[1].

First, we want to find v_3 that separates $\mathcal{C}_3$ from $\mathcal{C}_2$. v_3 will also separate $\mathcal{C}_3$ from $\mathcal{C}_1$, since when $d_{23} > \Delta$, then $d_{13} > \Delta$ must be true, because $d_{13} > d_{23}$ by definition. As before, we want $\dot{d}_{23} > 0$ when $d_{23} = \Delta$:

$$\begin{aligned} \dot{d}_{23} &= \dot{\chi}_3 - \dot{\chi}_2 > 0 \\ &= \gamma_3(-d_{13}N_1 - d_{23}N_2 + v_3) \\ &\quad - \gamma_2(-d_{12}N_1 + d_{23}N_3 + v_3) > 0 \end{aligned} \tag{83}$$

Suppose we find a v_3 that satisfies this inequality and achieves a separation of $\mathcal{C}_3$ from $\mathcal{C}_1$ and $\mathcal{C}_2$. The dynamics are now slightly different:

$$\begin{aligned} \dot{\chi}_1 &= \gamma_1(N_2 d_{12} + v_2), \\ \dot{\chi}_2 &= \gamma_2(-N_1 d_{12} + v_2), \\ \dot{\chi}_3 &= \gamma_3 v_2. \end{aligned} \tag{84}$$

[1] When we state that we want to separate $\mathcal{C}_i$ from $\mathcal{C}_j$, what we really mean is that we want to separate the robots in $\mathcal{C}_i$ from the robots in $\mathcal{C}_j$.

Since we now want $\dot{d}_{12} \geq 0$ when $d_{12} = \Delta$, the separating signal v_2 that separates $\mathcal{C}_1$ and $\mathcal{C}_2$ is exactly Equation (79).

It is important to note that this strategy seems to ignore the change in dynamics that occurs when $\mathcal{C}_1$ and $\mathcal{C}_3$ separate before $\mathcal{C}_2$ and $\mathcal{C}_3$ have separated. Similarly, depending on the choice of parameters γ_1, γ_2, and γ_3, as well as, N_1, N_2, and N_3, $\mathcal{C}_1$ and $\mathcal{C}_2$ may have separated before $\mathcal{C}_2$ and $\mathcal{C}_3$ separate. This scenario renders applying v_2 unnecessary. Therefore, this strategy will not be optimal; however, we will show that this strategy will still successfully separate the three class. Our motivation is to define a *simple* strategy, which will guarantee the separation of the different classes of robots independent of the parameters (with respect to the strategy only, not with respect to how the input signals are defined) and can easily be applied by a user.

Theorem 4.1.2. *v_3 is a scalar, constant separating signal if*

$$v_3 > \frac{(\gamma_3(N_1 + N_2) + \gamma_2 N_3)\Delta}{\gamma_3 - \gamma_2}, \tag{85}$$

which ensures that $\dot{d}_{23} > 0$ when $d_{23} \in [0, \Delta]$, i.e. guarantees that all robots belonging to $\mathcal{C}_3$ are separated from all robots in $\mathcal{C}_1$ and $\mathcal{C}_2$. Once this separation has occurred, the scalar, constant separating signal v_2 equal to Equation (79) can be applied to separate robots belonging to $\mathcal{C}_2$ from robots belonging to $\mathcal{C}_1$.

Proof. We want to find v_3 that guarantees $\dot{d}_{23} > 0$ when $d_{23} = \Delta$. First we solve for v_3 in Equation (83).

$$v_3 > \frac{(\gamma_3 N_2 + \gamma_2 N_3)d_{23} + (\gamma_3 d_{13} - \gamma_2 d_{12})N_1}{\gamma_3 - \gamma_2}$$

We select a sufficiently large v_3 by applying the fact that $d_{13} > d_{23}$ and $\gamma_2 d_{12} > 0$:

$$\begin{aligned} v_3 &> \frac{(\gamma_3(N_1 + N_2) + \gamma_2 N_3)d_{13}}{\gamma_3 - \gamma_2} \\ &= \frac{(\gamma_3(N_1 + N_2) + \gamma_2 N_3)(\Delta + \epsilon)}{\gamma_3 - \gamma_2}, \end{aligned}$$

where $\epsilon > 0$.

This is not an airtight upper bound, since the contribution from robots in $\mathcal{C}_1$ will be zero sometime before $d_{23} = \Delta$ and separation between $\mathcal{C}_2$ and $\mathcal{C}_3$ is achieved. However, it still guarantees that $\dot{d}_{23} > 0$ when $d_{23} \in [0, \Delta]$.

Once $\mathcal{C}_3$ is separated from $\mathcal{C}_1$ and $\mathcal{C}_2$, we are justified in using v_2 as defined by Equation (79) to separate $\mathcal{C}_1$ and $\mathcal{C}_2$ if and only if the inequality $\dot{d}_{23} \geq 0$ still holds when applying v_2. Otherwise, we cannot guarantee that $\mathcal{C}_2$ and $\mathcal{C}_3$ remain separated. After separation,

$$\dot{d}_{23} = \gamma_3 v_2 - \gamma_2(-d_{12}N_1 + v_2) \geq 0,$$

so we need to plug in v_2 and make sure the inequality holds.

$$v_2 = \frac{(\gamma_2 N_1 + \gamma_1 N_2)(\Delta + \epsilon)}{\gamma_2 - \gamma_1} \geq -\frac{\gamma_2 d_{12} N_1}{\gamma_3 - \gamma_2}$$

We can directly see that this inequality will hold, because a) $\gamma_3 > \gamma_2 > 0$, and b) $\gamma_2 d_{12} N_1 > 0$. Therefore, if we use v_3 to separate $\mathcal{C}_3$ from $\mathcal{C}_1$ and $\mathcal{C}_2$ and then v_2 to separate $\mathcal{C}_2$ from $\mathcal{C}_1$, we are able to completely separate all three classes from each other. □

Moreover, we can compute a duration for applying v_3, $T_{s,3}$, that is sufficient to separate $\mathcal{C}_3$ from the other two classes and a duration of applying v_2, $T_{s,2}$, that is sufficient to separate $\mathcal{C}_2$ from $\mathcal{C}_1$. These durations will be relaxed upper bounds, because of the following two assumptions:

1. We assume that the dynamics in Equation (82) are unchanged on the interval $[0, T_{s,3}]$, which is not accurate, since $\mathcal{C}_1$ separates from $\mathcal{C}_3$ before $\mathcal{C}_2$, and $\mathcal{C}_1$ may even separate from $\mathcal{C}_2$ before then depending on the weights and the sizes of the classes.
2. We assume that $d_{12}(T_{s,3}) = 0$, which is not accurate, since $d_{12} > 0$ is guaranteed by the fact that $\gamma_1 \neq \gamma_2$.

Under these assumptions, we can compute a simple schedule for applying v_3 and v_2 to achieve separation of the three classes.

Corollary 3. *To separate the three classes, $\mathcal{C}_1$, $\mathcal{C}_2$, and $\mathcal{C}_3$, we will apply the signal v_3 for a duration of $T_{s,3}$ and at time $T_{s,3}$, we will apply the signal v_2 for a duration of $T_{s,2}$, where $T_{s,3}$ and $T_{s,2}$ are defined as,*

$$\begin{aligned} T_{s,3} &= \frac{\ln\left(\frac{\Delta+\epsilon}{\epsilon}\right)}{\gamma_3(N_1 + N_2) + \gamma_2 N_3} \\ T_{s,2} &= \frac{\ln\left(\frac{\Delta+\epsilon}{\epsilon}\right)}{\gamma_1 N_2 + \gamma_2 N_1}, \end{aligned} \tag{86}$$

and $\epsilon > 0$.

Proof. The proof follows the same process as Corollary 2 by deriving $d_{23}(t)$ on the interval $t \in [0, T_{s,3}]$ and solving $d_{23}(T_{s,3}) = \Delta$ for $T_{s,3}$. $d_{23}(t)$ is derived from Equation (82) assuming that these dynamics are unchanged on the interval $[0, T_{s,3}]$. $d_{12}(t)$ is derived from Equation (84) on the interval $t \in (T_{s,3}, T_{s,2}]$. We solve $d_{12}(T_{s,2}) = \Delta$ for $T_{s,2}$ and assume that $d_{12}(T_{s,3}) = 0$ to find the time to separate $\mathcal{C}_1$ and $\mathcal{C}_2$ as in Corollary 2. □

This simple schedule is not optimal. We will hold v_3 for longer than is needed to separate $\mathcal{C}_3$ from $\mathcal{C}_2$, because we assume that the dynamics do not change on the interval $[0, T_{s,3}]$. Similarly, assuming that $d_{12}(T_{s,3}) = 0$ is pessimistic, since $d_{12} > 0$ is guaranteed by the fact that $\gamma_1 \neq \gamma_2$. The consequence is that we will hold v_2 for longer than is needed to separate $\mathcal{C}_2$ from $\mathcal{C}_1$. We could be more exact in computing these durations, since the parameters are known, and thus, we know how the system evolves. However, we are motivated to find a strategy that is *simple* to implement, in the sense that this strategy is independent of the choice in parameters (weights and sizes of the classes) that determine the order of separation[2]. Despite the lack of optimality, this strategy of applying v_3 for $T_{s,3}$ and then applying v_2 for $T_{s,2}$ will separate the three classes successfully.

4.1.1.3 M Classes

The next step is to find a similar strategy to completely separate robots belonging to M classes, $\mathcal{C}_1, \ldots, \mathcal{C}_M$. We define d_{ij} to be the distance between robots in $\mathcal{C}_i$ and $\mathcal{C}_j$. If three classes, $\mathcal{C}_i$, $\mathcal{C}_j$, and $\mathcal{C}_k$, are ordered such that $\gamma_i < \gamma_j < \gamma_k$, then by our previous construction, $d_{ik} = d_{ij} + d_{jk}$, such that $d_{ik} > d_{ij}$ and $d_{ik} > d_{jk}$ (with the exception of the initial conditions, where $d_{ik}(0) = d_{ij}(0) = d_{jk}(0) = 0$).

The swarm dynamics for the M classes before any separation are

$$\dot{\chi}_i = \gamma_i \left(\sum_{k=1}^{M} N_k d_{ik} + u \right),$$

where $d_{ii} = 0$ and $d_{ik} = -d_{ki}$. Instead of finding a single separating signal v, we are going to again be strategic and find a series of separating signals, $v_M, \ldots, v_2$, that separate the

[2] Independence from the parameters does not imply that v or T_s do not depend on the parameters, but rather that we can apply v for T_s time to separate two specific classes without worrying about how exactly these two classes separate from the other classes.

classes by peeling off classes in descending order of their weights.

First, we want to find v_M that separates $\mathcal{C}_M$ from $\mathcal{C}_{M-1}$. v_M will also separate $\mathcal{C}_M$ from $\mathcal{C}_{M-2}, \ldots, \mathcal{C}_1$. Next, we want to find v_{M-1} that separates $\mathcal{C}_{M-1}$ from $\mathcal{C}_{M-2}$, and so on until we have separated all of the M classes from each other. As was the case before, we will assume that the dynamics are unchanged while we separate two classes, even though $\mathcal{C}_{k-2}$ would separate from $\mathcal{C}_k$ before $\mathcal{C}_k$ and $\mathcal{C}_{k-1}$ have separated, and $\mathcal{C}_{k-1}$ may even separate from $\mathcal{C}_{k-2}$ before separation between $\mathcal{C}_k$ and $\mathcal{C}_{k-1}$ has been achieved. As we have shown before, v_k will still be an input signal that separates $\mathcal{C}_k$ and $\mathcal{C}_{k-1}, \ldots, \mathcal{C}_1$ successfully.

Theorem 4.1.3. *$v_M, \ldots, v_2$ is a series of scalar, constant separating signals that separate M classes completely, i.e. $d_{ij} > \Delta, \forall i \neq j$, and $i, j \in \{1, \ldots, M\}$, if*

$$v_k > \frac{\left(\gamma_k \sum_{j=1}^{k-1} N_j + \gamma_{k-1} N_k\right) \Delta}{\gamma_k - \gamma_{k-1}}, \tag{87}$$

where $k \in \{2, \ldots, M\}$.

Proof. We want to find v_k that separates robots in $\mathcal{C}_k$ from the robots in $\mathcal{C}_1, \ldots, \mathcal{C}_{k-1}$. Assuming that when v_k is applied, $\mathcal{C}_{k+1}, \ldots, \mathcal{C}_M$ are already separated from $\mathcal{C}_1, \ldots, \mathcal{C}_k$, then we know from the developments in the previous sections that v_k is of the form Equation (87). However, we have to make sure that applying v_k does not result in the merging of any of the already separated classes $\mathcal{C}_{k+1}, \ldots, \mathcal{C}_M$. Therefore, the following inequality must be satisfied,

$$v_k \geq -\frac{\gamma_k \sum_{j=1}^{k-1} d_{jk} N_j}{\gamma_{k+1} - \gamma_k},$$

for all $k = 2, \ldots, (M-1)$ such that $\mathcal{C}_k$ and $\mathcal{C}_{k+1}$ do not to merge, as well as, that $v_k \geq 0$ such that none of the separated classes $\mathcal{C}_{k+1}, \ldots, \mathcal{C}_M$ merge. Both of these inequalities are satisfied by the fact that v_k is always positive by inspection of Equation (87). □

Moreover, we can again compute a duration of applying v_M, $T_{s,M}$, that is sufficient to separate $\mathcal{C}_M$ from the other classes, a duration of applying v_{M-1}, $T_{s,M-1}$, that is sufficient to separate $\mathcal{C}_{M-1}$ from the other classes, and so on. As before, we apply the assumption that the dynamics do not change while a class $\mathcal{C}_k$ is separated from $\mathcal{C}_{k-1}$ and the other classes, and that when we start separating $\mathcal{C}_k$ from $\mathcal{C}_{k-1}$, that distance separating $\mathcal{C}_k$ from

$\mathcal{C}_{k-1}, \ldots, \mathcal{C}_1$ is zero. These assumptions, as before, lead to conservative upper bounds on the durations that the input signals are applied to separate the classes.

Corollary 4. *To separate the M classes, $\mathcal{C}_1, \ldots, \mathcal{C}_M$, we will apply the signal v_M for a duration of $T_{s,M}$, and at time $T_{s,M}$, we will apply the signal v_{M-1} for a duration of $T_{s,M-1}$, and so on, where v_k is held for a duration of $T_{s,k}$,*

$$T_{s,k} = \frac{\ln\left(\frac{\Delta+\epsilon}{\epsilon}\right)}{\gamma_k \sum_{j=1}^{k-1} N_j + \gamma_{k-1} N_k}, \tag{88}$$

where $k = M, \ldots, 2$ and $\epsilon > 0$. Let $T_{s,M+1} = 0$.

Proof. Suppose that we pick the separating signal such that

$$v_k = \frac{\left(\gamma_k \sum_{j=1}^{k-1} N_j + \gamma_{k-1} N_k\right)(\Delta + \epsilon)}{\gamma_k - \gamma_{k-1}}.$$

The proof follows directly from the proof of Corollary 2 by deriving $d_{(k-1)k}(t)$ on the interval $t \in [T_{s,k+1}, T_{s,k}]$ and solving $d_{(k-1)k}(T_{s,k}) = \Delta$ for $T_{s,k}$, as if $d_{(k-1)k}(T_{s,k+1}) = 0$. □

This strategy is not optimal for the same reasons as before. Assuming that $d_{(k-1)k}(T_{s,k+1}) = 0$ is pessimistic, since $d_{(k-1)k} > 0$ is guaranteed by the fact that $\gamma_{k-1} \neq \gamma_k$. Similarly, assuming that the dynamics do not change when peeling a class away from the rest of the classes is not accurate. The consequence is that we will hold v_k for longer than is needed to separate $\mathcal{C}_k$ from $\mathcal{C}_1, \ldots, \mathcal{C}_{k-1}$, $\forall k = 2, \ldots, M$. However, complete separation of all M classes is achieved under this strategy.

4.1.1.4 Other Generalizations

In the previous sections we assumed that $x_i \in \mathbb{R}$; however, the exact same arguments apply to non-scalar robots ($x_i \in \mathbb{R}^n, n \geq 2$) under the same dynamics. The only difference is that we still insist on a constant v_k, where the separation condition becomes

$$\|v_k\| > \frac{(\gamma_k \sum_{j=1}^{k-1} N_j + \gamma_{k-1} N_k)\Delta}{\gamma_k - \gamma_{k-1}}.$$

This condition follows from the fact that the dynamics are decoupled along all dimensions and that the magnitude of v_k is independent of its direction in $\mathbb{R}^n$.

We want to be able to remove our assumption about the initial conditions, i.e., that $x_i(0) = x_j(0),\ \forall(i,j)$. These initial condition could be achieved by simply running the unforced version of the dynamics, which we know will asymptotically drive all robots to a common location (as long as the network stays connected). However for practical purposes, it may be too long to wait for all robots to converge to exactly the same location, so what we will do is see how the argument needs to change when we insist that $\|x_i(0) - x_j(0)\| \leq 2\delta,\ \forall(i,j)$ for a given, small $\delta > 0$.

To show that separation is possible between two classes, $\mathcal{C}_1$ and $\mathcal{C}_2$, we need to show that v is a constant, separating signal that completely separates all pairs of robots $(i,j), i \in \mathcal{C}_1, j \in \mathcal{C}_2$. In fact, we will show that if we assume that $\Delta > 4\delta$, $x_i \in \mathbb{R}$, and pick a v that separates the centroids of the two classes by $\Delta + 2\delta$, then

1. While the network is completely connected, the centroids are separating, i.e. $\dot{\bar{x}}_2 - \dot{\bar{x}}_1 > 0$, and robots are moving towards the centroid of their class.

2. Once some of the robots from the two classes start separating, $\dot{\bar{x}}_2 - \dot{\bar{x}}_1 > 0$ holds and the closest pair $(i,j), i \in \mathcal{C}_1, j \in \mathcal{C}_2$ is separating.

Lemma 4.1.4. *If the robots are initially co-located in a δ-ball, i.e. $|x_i(0) - x_j(0)| \leq 2\delta, \forall(i,j)$ and $\Delta > 4\delta$, then an input signal*

$$u > \frac{(\gamma_1 N_2 + \gamma_2 N - 1)(\Delta + 2\delta)}{\gamma_2 - \gamma_1},$$

will ensure that $\dot{\bar{x}}_2 - \dot{\bar{x}}_1 > 0$ on the interval $(\bar{x}_2 - \bar{x}_1) \in [-2\delta, \Delta + 2\delta]$.

Proof. Suppose $\bar{x}_k(t)$ is the centroid of the positions of the robots in class $\mathcal{C}_k$, i.e.

$$\bar{x}_k(t) = \frac{1}{N_k} \sum_{j \in \mathcal{C}_k} x_j(t),$$

then, the first step is to find the derivative of the two centroids, $\dot{\bar{x}}_1$ and $\dot{\bar{x}}_2$. We can rewrite Equation (76) for an robot $x_i \in \mathcal{C}_1$ as,

$$\begin{aligned} \dot{x}_i &= \gamma_1 \left(\sum_{j \in N(1)} (x_j - x_i) + \sum_{j \in N(2)} (x_j - x_i) + u \right), \\ &= \gamma_1 \left(N_1(\bar{x}_1 - x_i) + N_2(\bar{x}_2 - x_i) + u \right), \end{aligned}$$

under the assumption that all robots of $\mathcal{C}_1$ and $\mathcal{C}_2$ are connected. This assumption is certainly true while all robots are inside the δ-ball and before the distance between any two robots from different classes exceeds Δ. Then,

$$\begin{aligned}\dot{\bar{x}}_1 &= \frac{1}{N_1}\sum_{j\in\mathcal{C}_1}\dot{x}_j\\ &= \frac{\gamma_1}{N_1}\sum_{j\in\mathcal{C}_1}\left(N_1(\bar{x}_1 - x_j) + N_2(\bar{x}_2 - x_j) + u\right)\\ &= \gamma_1\left(N_2(\bar{x}_2 - \bar{x}_1) + u\right)\end{aligned}$$

Following the same procedure, we can compute $\dot{\bar{x}}_2$,

$$\dot{\bar{x}}_2 = \gamma_2\left(N_1(\bar{x}_1 - \bar{x}_2) + u\right),$$

and in turn we can compute,

$$\begin{aligned}\dot{\bar{x}}_2 - \dot{\bar{x}}_1 &= \gamma_2\left(N_1(\bar{x}_1 - \bar{x}_2) + u\right) - \gamma_1\left(N_2(\bar{x}_2 - \bar{x}_1) + u\right)\\ &= -(\gamma_2 N_1 + \gamma_1 N_2)(\bar{x}_2 - \bar{x}_1) + (\gamma_2 - \gamma_1)u\end{aligned}$$

Without an external input, $u = 0$, the distance between the centroid decays to zero asymptotically; however, if we were to apply

$$u = \frac{(\gamma_2 N_1 + \gamma_1 N_2)(\Delta + 2\delta + \epsilon)}{\gamma_2 - \gamma_1}, \tag{89}$$

where $\epsilon > 0$, then we ensure that the distance between the centroids is always increasing.

One of the assumptions we made is that all robots of $\mathcal{C}_1$ and $\mathcal{C}_2$ are connected during the separation process; however, we know that not all robots of $\mathcal{C}_1$ will separate from all robots of $\mathcal{C}_2$ simultaneously. In fact, the dynamics will change as robots start to separate, but we will show that v will still ensure complete separation of the two classes.

Suppose that an robot, $x_i \in \mathcal{C}_1$, starts to separate from some of the robots in $\mathcal{C}_2$, and therefore, this robot's dynamics change to

$$\dot{\bar{x}}_i = \gamma_1\left(\sum_{j\in N(1)}(x_j - x_i) + \sum_{j\in\tilde{N}(2,i)}(x_j - x_i) + u\right),$$

where $\tilde{N}(2,i)$ is the set of $\tilde{N}_{2,i}$ robots from class $\mathcal{C}_2$ that are still connected to robot x_i from class $\mathcal{C}_1$. As a consequence, the dynamics of the centroid of class $\mathcal{C}_1$, now denoted $\tilde{\bar{x}}_1$, are also changed to

$$\begin{aligned}\dot{\bar{x}}_1 &= \frac{\gamma_1}{N_1}\sum_{j\in\mathcal{C}_1}\left(\sum_{k\in\tilde{N}(2,j)}(x_k - x_j) + u\right)\\ &\le \gamma_1\left(N_2(\bar{x}_2 - \bar{x}_1) + u\right),\end{aligned}$$

and similarly,

$$\begin{aligned}\dot{\bar{x}}_2 &= \frac{\gamma_2}{N_2}\sum_{j\in\mathcal{C}_2}\left(\sum_{k\in\tilde{N}(1,j)}(x_k - x_j) + u\right)\\ &\ge \gamma_2\left(N_1(\bar{x}_1 - \bar{x}_2) + u\right).\end{aligned}$$

Therefore,

$$\begin{aligned}\dot{\bar{x}}_2 - \dot{\bar{x}}_1 &\ge \gamma_2\left(N_1(\bar{x}_1 - \bar{x}_2) + u\right) - \gamma_1\left(N_2(\bar{x}_2 - \bar{x}_1) + u\right)\\ &= -(\gamma_2 N_1 + \gamma_1 N_2)(\bar{x}_2 - \bar{x}_1) + (\gamma_2 - \gamma_1)u.\end{aligned}$$

If we apply v as defined in Equation (89), then $\dot{\bar{x}}_2 - \dot{\bar{x}}_1 > 0$ continues to hold even if some of the pairs $(i, j), i \in \mathcal{C}_1, j \in \mathcal{C}_2$, have separated. □

Lemma 4.1.5. *If the robots are initially co-located in a δ-ball, i.e. $|x_i(0) - x_j(0)| \le 2\delta, \forall(i, j)$, $\Delta > 4\delta$, and while $(\bar{x}_2 - \bar{x}_1) \in [-2\delta, 2\delta]$, the network is completely connected and each robot is moving towards the centroid of their class.*

Proof. We want to be able to show that while the networks is still completely connected (which is certainly true while $|\bar{x}_2 - \bar{x}_1| \le 2\delta$, because $\Delta > 4\delta$), robot $i \in \mathcal{C}_1$ is moving towards $\bar{x}_1$ and robot $j \in \mathcal{C}_2$ is moving towards $\bar{x}_2$:

$$\begin{aligned}\dot{\bar{x}}_1 - \dot{x}_i &= \gamma_1\left(N_2(\bar{x}_2 - \bar{x}_1) + u\right)\\ &\quad - \gamma_1\left(N_1(\bar{x}_1 - x_i) + N_2(\bar{x}_2 - x_i) + u\right)\\ &= -\gamma_1(N_1 + N_2)(\bar{x}_1 - x_i),\end{aligned}$$

and similarly,

$$\dot{\bar{x}}_2 - \dot{x}_j = -\gamma_2(N_1 + N_2)(\bar{x}_2 - x_j).$$

These equations show that each robot is moving towards the centroid of their class. The result is that if we have separated the centroids by 2δ, we can be sure that all robots $j \in \mathcal{C}_2$ are to the right of all robots $i \in \mathcal{C}_1$, i.e. $x_j > x_i,\ \forall(i, j)$. □

Unfortunately, once some of the robots start to separate, we can no longer guarantee that robots are moving towards the centroid of their class. However, we can show that the closest pair $(i, j), i \in \mathcal{C}_1, j \in \mathcal{C}_2$ continues to separate.

Lemma 4.1.6. *Once some of the robots from the two different classes have started to separate, the closest pair $(i, j), i \in \mathcal{C}_1, j \in \mathcal{C}_2$ is separating if we continue to apply*

$$u > \frac{(\gamma_1 N_2 + \gamma_2 N - 1)(\Delta + 2\delta)}{\gamma_2 - \gamma_1},$$

i.e. $\dot{x}_j - \dot{x}_i > 0$.

Proof. Suppose $\tilde{N}_{2,i}$ is the number of robots of $\mathcal{C}_2$ that robot i can detect, then $\tilde{\bar{x}}_{2,i}$ is the centroid of those robots. Similarly, $\tilde{N}_{1,j}$ is the number of robots of $\mathcal{C}_1$ that robot j can detect, and $\tilde{\bar{x}}_{1,j}$ is the centroid of those robots. Recall that we have separated the two classes in such a way that $\bar{x}_2 > \bar{x}_1$, $\bar{x}_2 > \tilde{\bar{x}}_{2,i}$, $\bar{x}_1 < \tilde{\bar{x}}_{1,j}$, and $x_j > x_i$. If $(i, j), i \in \mathcal{C}_1, j \in \mathcal{C}_2$ is the closest pair, then $\bar{x}_1 < x_i$ and $\bar{x}_2 > x_j$. We will use these inequalities to show that $\dot{x}_j - \dot{x}_i > 0$:

$$\begin{aligned}
\dot{x}_j - \dot{x}_i &= \gamma_2(\tilde{N}_{1,j}(\tilde{\bar{x}}_{1,j} - x_j) + N_2(\bar{x}_2 - x_j) + u) \\
&- \gamma_1(N_1(\bar{x}_1 - x_i) + \tilde{N}_{2,i}(\tilde{\bar{x}}_{2,i} - x_i) + u) \\
&= (\gamma_2 - \gamma_1)u + \gamma_2 N_2(\bar{x}_2 - x_j) - \gamma_1 N_1(\bar{x}_1 - x_i) \\
&+ \gamma_2 \tilde{N}_{1,j}(\tilde{\bar{x}}_{1,j} - x_j) - \gamma_1 \tilde{N}_{2,i}(\tilde{\bar{x}}_{2,i} - x_i) \\
&> (\gamma_2 - \gamma_1)u + \gamma_2 N_2(\bar{x}_2 - x_j) - \gamma_1 N_1(\bar{x}_1 - x_i) \\
&+ \gamma_2 N_1(\bar{x}_1 - x_j) - \gamma_1 N_2(\bar{x}_2 - x_i) \\
&= (\gamma_2 - \gamma_1)(u + N_1\bar{x}_1 + N_2\bar{x}_2) \\
&- \gamma_2(N_1 + N_2)x_j + \gamma_1(N_1 + N_2)x_i \\
&> (\gamma_2 - \gamma_1)(u + N_1\bar{x}_1 + N_2\bar{x}_2) \\
&- \gamma_2(N_1 + N_2)\bar{x}_2 + \gamma_1(N_1 + N_2)\bar{x}_1 \\
&= (\gamma_2 - \gamma_1)u - (\gamma_2 N_1 + \gamma_1 N_2)(\bar{x}_2 - \bar{x}_1) \\
&= \dot{\bar{x}}_2 - \dot{\bar{x}}_1 > 0
\end{aligned}$$

□

Since the centroids and the closest pair $(i, j), i \in \mathcal{C}_1, j \in \mathcal{C}_2$ are separating under v from Equation (89), we know that the two classes continue to separate even as some of the robots in each class have already separated. However, we have no guarantee that the centroids are not separating significantly faster than the closest pair, such that when the centroids are separated by $\Delta + 2\delta$, the closest pair is separated by distance less than Δ. Therefore, there may exist a permutation of γ_1, γ_2, N_1, and N_2, for which v is not sufficient to separate the two classes. Despite this possibility, we can make the following conjecture based on these lemmas and our simulations:

Conjecture 1. *If the robots are initially co-located in a δ-ball, i.e. $\|x_i(0) - x_j(0)\| \leq \delta,\ \forall(i, j)$, then it is possible to completely separate two classes of robots, $\mathcal{C}_1$ and $\mathcal{C}_2$, by separating the centroids of the two classes by a distance greater than $\Delta + 2\delta$. v is a separating signal if*

$$u > \frac{(\gamma_1 N_2 + \gamma_2 N_1)(\Delta + 2\delta)}{\gamma_2 - \gamma_1}, \tag{90}$$

where $\delta > 0$ and $\Delta > 4\delta$.

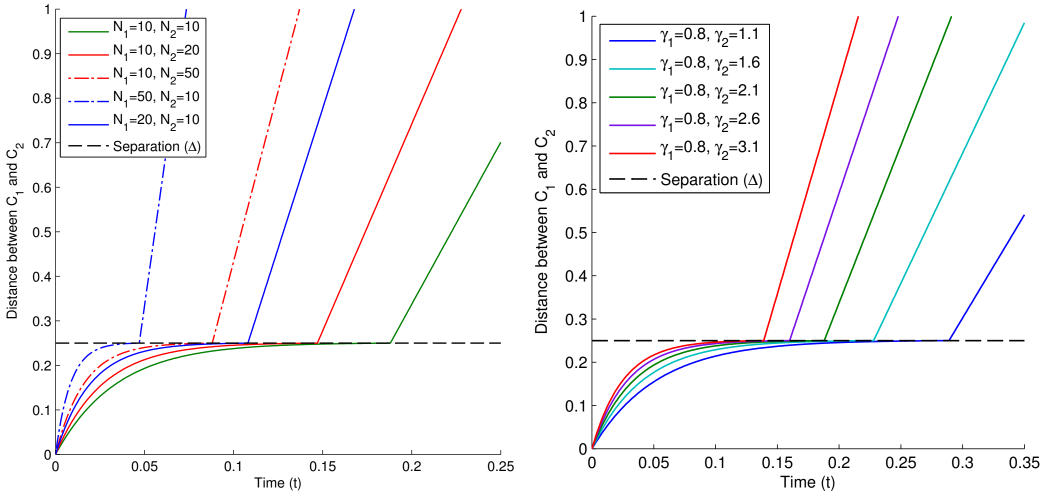
4.1.2 Simulations

We want to verify numerically in simulation whether our results hold, and demonstrate the effect of varying the parameters γ_k and N_k for each of the M classes. First, let us consider the two class case, where we are interested in separating $\mathcal{C}_1$ from $\mathcal{C}_2$. Figure 19 demonstrates a successful separation using the control signal

$$u = \frac{(\gamma_2 N_1 + \gamma_2 N_2)(\Delta + \epsilon)}{\gamma_2 - \gamma_1},$$

for some $\epsilon > 0$. The separation distance Δ is indicated by the black dashed line. The distance between the two classes logarithmically approaches Δ until separation occurs, after which the distance that separates the two classes increases quickly. If it were the case that v was not sufficient to separate the two classes, we would see that the distance between $\mathcal{C}_1$ and $\mathcal{C}_2$ in the plots would stay under the black dashed line.

Figure 19a illustrates the effect of varying N_1 and N_2, while Figure 19b illustrates the effect of varying γ_1 and γ_2. In all cases, the distance between $\mathcal{C}_1$ and $\mathcal{C}_2$ eventually exceeds the separation distance Δ (the black dashed line in the figures).



(a) Distance separating $\mathcal{C}_1$ and $\mathcal{C}_2$, when $\gamma_1 = 0.8, \gamma_2 = 2.1$ and N_1, N_2 are varied.

(b) Distance separating $\mathcal{C}_1$ and $\mathcal{C}_2$, when $N_1 = 10, N_2 = 10$ and γ_1, γ_2 are varied.

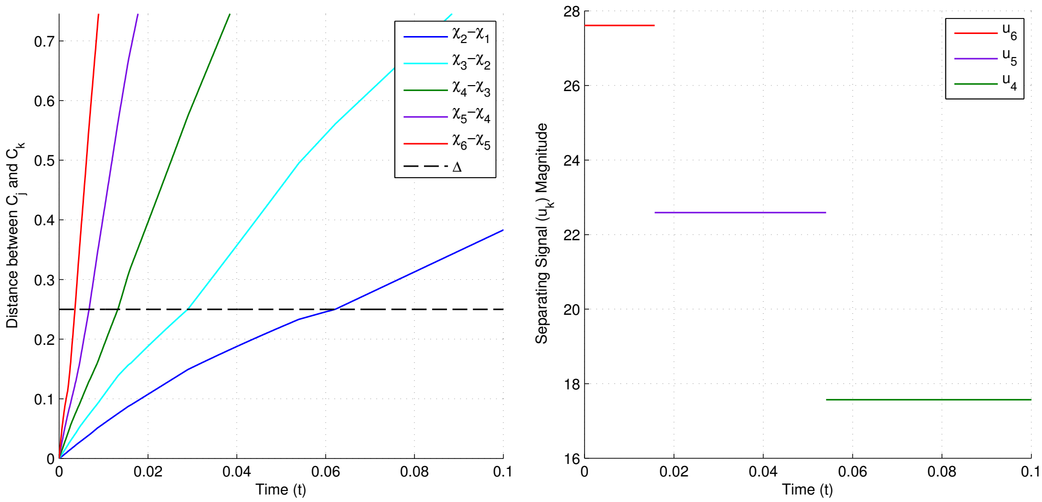
Figure 19: Successful separation of $\mathcal{C}_1$ and $\mathcal{C}_2$ for a variety of parameters.

We can also demonstrate that our choice of $v_M, \ldots, v_2$ applied to the case when we want to separates six classes, $\mathcal{C}_1, \ldots, \mathcal{C}_6$ is also successful. Again, a failure to separate a pair of classes (i.e., v_k is not sufficient for separation) would have been indicated by one of the separation distances (lines in the plot) staying under the black dashed line. Figure 20 illustrates that we can successfully separate the six classes from each other. In both cases, all classes have the same number of robots, i.e. $N_1 = \ldots = N_6 = 10$. Figure 20a specifically considers the case when the inter-class difference in γ increases, i.e. $\gamma_2 - \gamma_1 < \gamma_3 - \gamma_2 < \ldots < \gamma_6 - \gamma_5$. In this scenario, $\mathcal{C}_6$ and $\mathcal{C}_5, \ldots, \mathcal{C}_1$ completely separate first, then $\mathcal{C}_5$ and $\mathcal{C}_4, \ldots, \mathcal{C}_1$ separate completely, and so on. Figure 20b illustrates the simple schedule used to separate the classes. In this scenario, v_6, v_5, and v_4 are sufficient to actually separate all M classes.

Last, we want to demonstrate that if the robots move in $\mathbf{R}^2$ and do not start in the same location, namely they are co-located in some δ-ball, then we can still achieve separation using the control signal

$$u = \frac{(\gamma_2 N_1 + \gamma_2 N_2)(\Delta + 2\delta + \epsilon)}{\gamma_2 - \gamma_1},$$

for some $\delta, \epsilon > 0$.



(a) Distances separating $\mathcal{C}_1,\ldots,\mathcal{C}_6$, when $\gamma_1 = 0.2, \gamma_2 = 0.4, \gamma_3 = 0.8, \gamma_4 = 1.6, \gamma_5 = 3.2, \gamma_6 = 6.4$, $N_1 = \ldots = N_6 = 10$

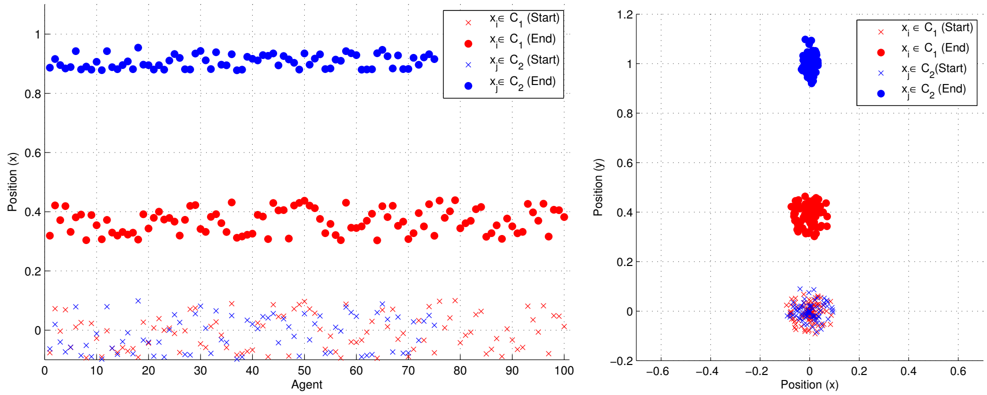
(b) Each v_k is applied for a duration of $T_{s,k}$, where $k = \{4,5,6\}$ on the interval $t \in (0, 0.1)$.

Figure 20: Successful separation of $\mathcal{C}_1,\ldots,\mathcal{C}_6$ with a simple schedule.

Figure 21 illustrates the case where 100 robots in $\mathcal{C}_1$ separate from 75 robots in $\mathcal{C}_2$. The separation distance when any two robots disconnect is $\Delta = 0.4$, and all robots start from a location within a δ-ball, where $\delta = 0.1$. The centroids of the two classes are separated by a distance greater than $\Delta + 2\delta$ when the simulation ends. The minimum separation between two robots of each class is Δ_{min}, and since $\Delta_{\text{min}} > \Delta$, the two classes are completely separated. Figure 21a illustrates the case when $x_i \in \mathbb{R}$, while Figure 21b illustrates the case when $x_i \in \mathbb{R}^2$. The signal v separates the two classes in both cases.

4.1.3 Feasibility Revisited

Before we conclude our discussion of using a broadcast input signal to separate a swarm of heterogeneous robots into homogeneous clusters of robots with the same class, let us revisit this HSI control structure in the context of this separation task. First, we can demonstrate using a CLF that there exists a broadcast input signal that can separate a heterogeneous swarm of two difference classes of robots. Once again, we use the initial conditions $x_i(t_0) = x_j(t_0),\ \forall i,j \in \mathcal{C}_1$ and $x_i(t_0) = x_j(t_0),\ \forall i,j \in \mathcal{C}_2$, which corresponds all



(a) Separating $\mathcal{C}_1$ and $\mathcal{C}_2$ in $\mathbb{R}$, $\Delta = 0.4, \delta = 0.1$. $\Delta_{\text{min}} = 0.446$.

(b) Separating $\mathcal{C}_1$ and $\mathcal{C}_2$ in $\mathbb{R}^2$, $\Delta = 0.4, \delta = 0.1$. $\Delta_{\text{min}} = 0.457$.

Figure 21: Successful separation of $\mathcal{C}_1$ and $\mathcal{C}_2$ when robots start in a δ-ball, $N_1 = 100, N_2 = 75$, $\gamma_1 = 0.2, \gamma_2 = 0.7$.

robots of same class starting together. We can simplify the dynamics, as before, to

$$\begin{aligned}\dot{\chi}_1 &= -\gamma_1(N_2(\chi_1 - \chi_2) - v) = f_{H,1}(\chi, v)\\ \dot{\chi}_2 &= \gamma_2(N_1(\chi_1 - \chi_2) + v) = f_{H,2}(\chi, v),\end{aligned} \tag{91}$$

where $\chi_i \in \mathbf{R}$ represents the shared position of all robots of class $\mathcal{C}_i$, and $\chi = [\chi_1, \chi_2]^T$.

A specification set that encodes a separation distance of Δ between the two classs of robots is $\mathcal{S} = \{x \in \mathbf{R} \mid \|x_i - x_j\| = \Delta,\ \forall i, j,\ i \in \mathcal{C}_1, j \in \mathcal{C}_2\}$. Consequently, we pick a candidate CLF [36]

$$V(\chi) = \frac{1}{4}\left(\|\chi_1 - \chi_2\|^2 - \Delta^2\right)^2, \tag{92}$$

which is positive definite everywhere except at the quasi-static equilibrium points, where $\|\chi_1 - \chi_2\| = \Delta$. Next, we need to show that

$$\dot{V}(\chi, v) = \begin{bmatrix} \frac{\partial V}{\partial \chi_1} \\ \\ \frac{\partial V}{\partial \chi_2} \end{bmatrix}^T \begin{bmatrix} f_{H,1}(\chi, v) \\ f_{H,2}(\chi, v) \end{bmatrix} < 0 \tag{93}$$

Suppose that in this example the domain is $D = \left\{\chi \in \mathbf{R}^2 \mid 0 \le \|\chi_1 - \chi_2\| \le \Delta, \chi_1 \le \chi_2\right\}$, that all robots of the same class start at the same location $[\chi_1(t_0), \chi_2(t_0)]^T \in D$, that the “weights” of the two classes of robots are ordered $0 < \gamma_1 < \gamma_2$, and that

$$\begin{aligned}\dot{V}(\chi, v) &= -(\gamma_1 N_2 + \gamma_2 N_1)(\chi_1 - \chi_2)^2(\|\chi_1 - \chi_2\|^2 - \Delta^2) \\ &\quad - (\gamma_2 - \gamma_1)(\chi_1 - \chi_2)(\|\chi_1 - \chi_2\|^2 - \Delta^2)v,\end{aligned} \tag{94}$$

then for every $\chi \in D$,

$$v \geq \frac{\gamma_1 N_2 + \gamma_2 N_1}{\gamma_2 - \gamma_1}(\chi_2 - \chi_1) \tag{95}$$

will ensure that $\dot{V}(\chi, v) \leq 0$, where $\dot{V}(\chi, v) = 0$ only whenever $\|\chi_1 - \chi_2\| = \Delta$ or $\chi_1 = \chi_2$. By LaSalle’s invariance principle, this system will converge to the largest invariant set M in $\left\{\chi \in \Omega \;\middle|\; \dot{V}(\chi, v) = 0\right\}$ as $t \to \infty$, where Ω is the compact subset

$$\left\{\chi \in \mathbf{R}^2 \;\middle|\; V(\chi) \leq \frac{1}{4}\Delta^4 - \epsilon, \epsilon > 0\right\} \subset D. \tag{96}$$

The largest invariant set M is

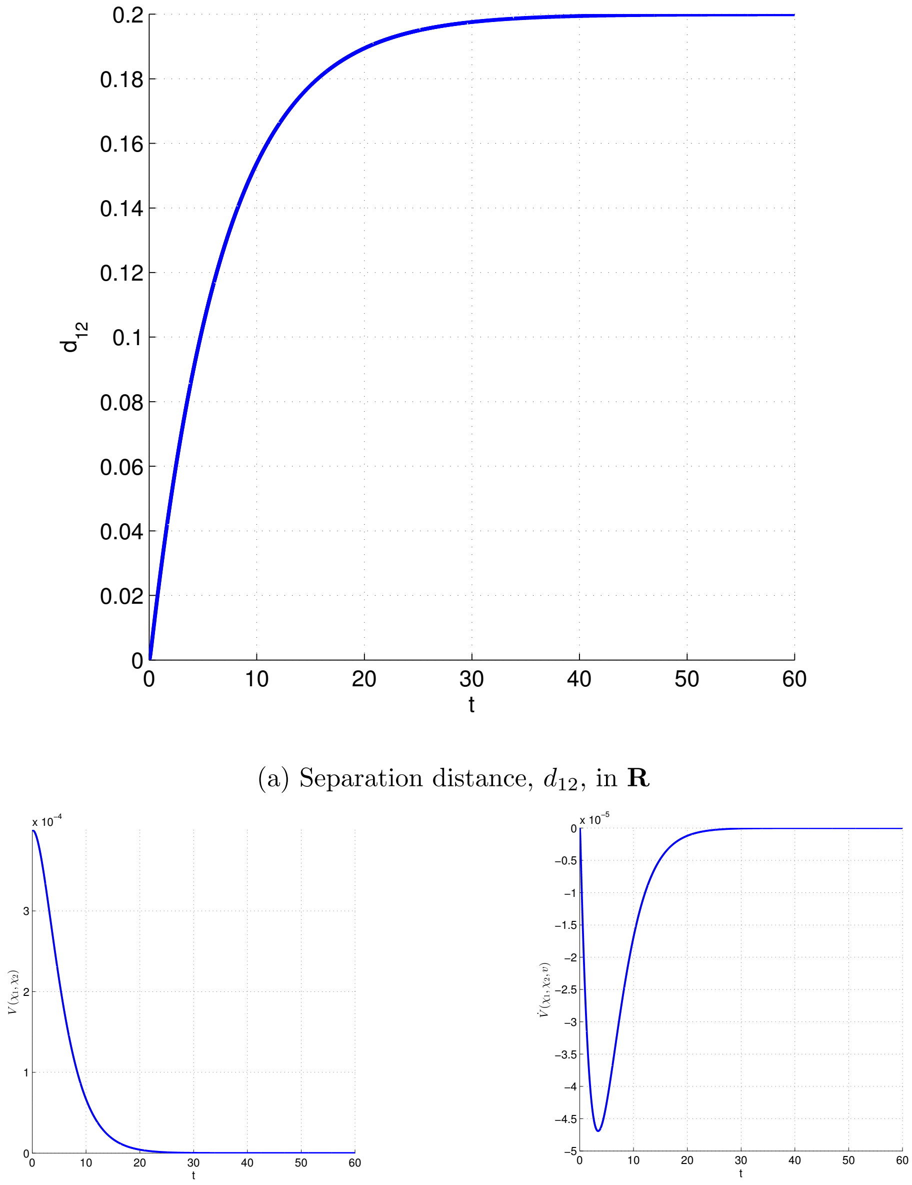
$$\left\{\chi \in \Omega \;\middle|\; \dot{V}(\chi, v) = 0,\ \|\chi_1 - \chi_2\| = \Delta,\ v = \frac{\gamma_1 N_2 + \gamma_2 N_1}{\gamma_2 - \gamma_1}\Delta\right\}, \tag{97}$$

because for this particular $v \in \mathcal{V}$, $\dot{\chi}_2 - \dot{\chi}_1 = 0$, such that $\|\chi_1 - \chi_2\| = \Delta$ will always hold and thus $\dot{V}(\chi, v) = 0$ and $V(\chi) = 0$. $M \subseteq S$; therefore, it is feasible for the user to use this broadcast control HSI structure to separate the two classes of robots by a distance Δ if the system starts at $\chi(t_0)$ in Ω.

Figure 22 illustrates separation of a swarm of ten robots of $\mathcal{C}_1$ and five robots of $\mathcal{C}_2$ by a distance $\Delta = 0.2$. The user is applying a constant, positive broadcast signal

$$v = \frac{(\gamma_1 N_2 + \gamma_1 N_1)}{\gamma_2 - \gamma_1}\Delta,$$

analogous to using a wind tunnel to move robots (on a rail) with mass inversely proportional to γ_i. Figure 22a indicates the starting location of the $\mathcal{C}_1$ (blue) robots and $\mathcal{C}_2$ (red) robots by × and their final positions by ∘. Initially, the separation between the two classes of robots is less than Δ, but eventually, their separation equals Δ. This plot is confirmed by Figure 22b which shows that the CLF $V(\chi)$ is positive, but “energy” dissipates as the desired separation distance is achieved, while Figure 22c shows that $\dot{V}(\chi, v)$ remains negative during the interaction. Consequently, it is feasible for the user to use a strong enough broadcast signal to separate the two classes of robots in the example by a distance of Δ.



(a) Separation distance, d_{12}, in $\mathbf{R}$

(b) $V(\chi)$

(c) $\dot{V}(\chi, v)$

Figure 22: A user is separating a swarm of ten robots of $\mathcal{C}_1$ ($\gamma_1 = 0.2$) and 30 robots of $\mathcal{C}_2$ ($\gamma_2 = 0.9$) by a distance $\Delta = 0.2$ with a broadcast signal v.

4.1.4 Attention, Effort, and Scalability

Similar to our discussion of broadcast control in the context of rendezvous in Section 2.4.4, we compute v^* for this separation problem using optimal control tools. We solve the following optimization problem:

$$\begin{aligned}\min_{w} \quad & J(w) = \frac{1}{2}\int_0^\infty \big(\kappa(d_{12}-\Delta)^T(d_{12}-\Delta) + w^T w\big)\, dt \\ \text{s.t.} \quad & \dot{x} = -(\gamma_1 N_2 + \gamma_2 N_1)d_{12} + (\gamma_2 - \gamma_1)v = Ad_{12} + Bv \\ & \dot{v} = w \\ & d_{12}(0) = d_{12,0},\ v(0) = 0,\end{aligned} \tag{98}$$

where d_{12} denotes the separation between the two classes of robots, and $\kappa > 1$ weighs the tracking error stronger than the cost on attention. The effort cost, $v^T v$, is missing, because in this particular task, it is crucial to exert enough effort to separate the two classes of robots.

This is again a continuous-time, infinite horizon linear quadratic regulator-like (LQR-like) problem, which can be solved in the following manner. First, the first order necessary conditions (FONC) for optimality are:

$$\begin{aligned} H &= \frac{1}{2}\big((d_{12}-\Delta)^T(d_{12}-\Delta) + w^T w\big) + \lambda^T \dot{x} + \mu^T \dot{v} \\ \frac{\partial H}{\partial w} &= w^T + \mu^T = 0 \Rightarrow w = -\mu \\ \dot{\lambda} &= -\frac{\partial H}{\partial x} = A^T \lambda - \kappa d_{12} + \kappa \Delta \\ \dot{\mu} &= -\frac{\partial H}{\partial v} = -B^T \lambda \end{aligned} \tag{99}$$

It is important to note here that the co-state dynamics, $\dot{\lambda}$, include an extra affine term that is typically not present in a standard LQR problem. For convenience, let us stack states and co-states into single variables in the following way:

$$\begin{aligned} z &= \begin{bmatrix} d_{12} \\ v \end{bmatrix},\ \dot{z} = \begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} z + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w = A_z z + B_z w \\ \eta &= \begin{bmatrix} \lambda \\ \mu \end{bmatrix},\ \dot{\eta} = \begin{bmatrix} A^T & 0 \\ B^T & 0 \end{bmatrix} \eta - \begin{bmatrix} \kappa & 0 \\ 0 & 0 \end{bmatrix} z + \begin{bmatrix} \kappa\Delta \\ 0 \end{bmatrix} = -A_z^T \eta - C_z z + \Psi \end{aligned} \tag{100}$$

We propose that $\eta(t) = S(t)z(t) + P(t)$ is the solution to the stacked co-state equations. The affine component, $P(t)$, is to account for the affine component that is tracked in the cost. If we start from the proposed solution, then

$$\begin{aligned}
\eta &= Sz + P \\
\dot{\eta} &= \dot{S}z + S\dot{z} + \dot{P} \\
-A_z^T\eta - C_z z + \Psi &= \dot{S}z + SA_z z + SB_z w + \dot{P} \\
-A_z^T Sz - A_z^T P - C_z z + \Psi &= \dot{S}z + SA_z z - SB_z B_z^T Sz - SB_z B_z^T P + \dot{P} \\
-\dot{P} - (A_z^T - SB_z B_z^T)P + \Psi &= \left(\dot{S} + SA_z + A_z^T S - SB_z B_z^T S + C_z\right) z
\end{aligned} \tag{101}$$

Since this LQR-like problem is computed over an infinite horizon, we can compute the steady state $\hat{S}$ and $\hat{P}$, when $\dot{S} = 0$ and $\dot{P} = 0$. Consequently, to satisfy Equation 101, we must solve

$$\begin{aligned}
\hat{P} &= (A_z^T - SB_z B_z^T)^{-1}\Psi \\
0 &= \hat{S}A_z + A_z^T\hat{S} - \hat{S}B_z B_z^T\hat{S} + C_z
\end{aligned} \tag{102}$$

The second equation is the continuous time algebraic Ricatti equation, while P can be solved for directly. Finally, we are able to compute $\dot{v}^* = w$,

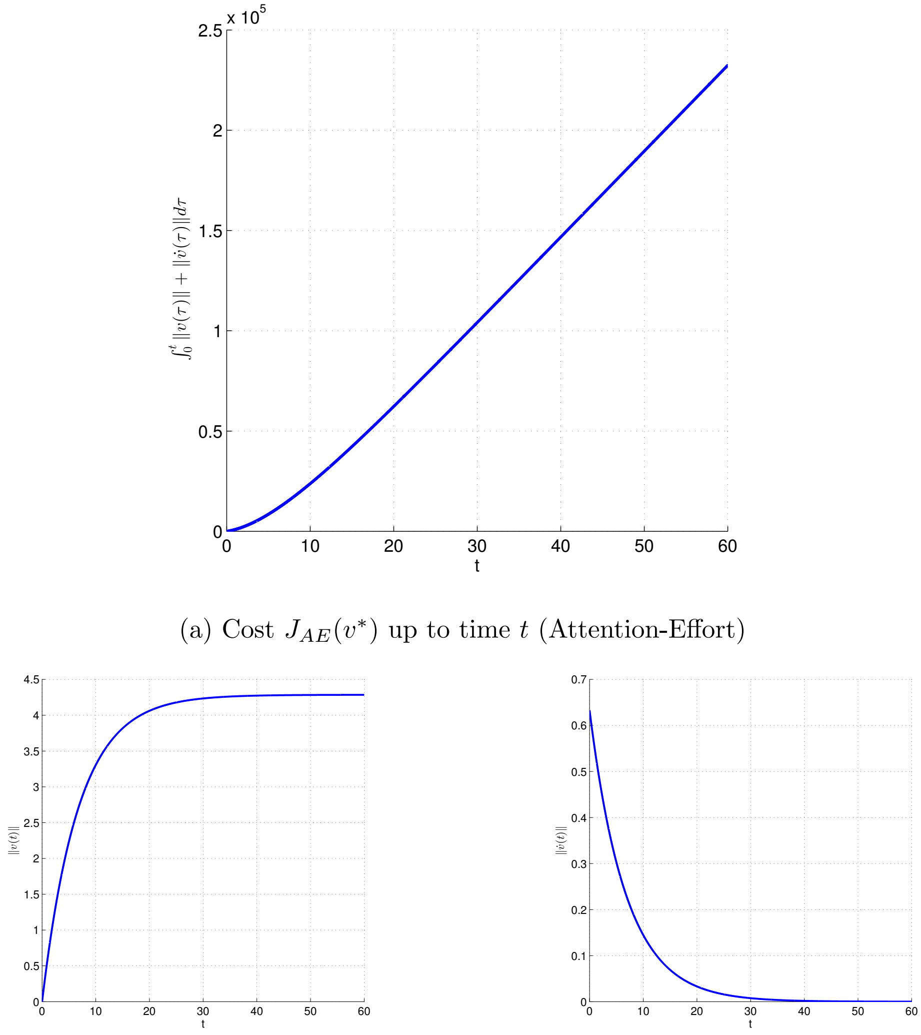
$$\begin{aligned}
w &= -\mu \\
&= -B_z^T(\hat{S}z + \hat{P}).
\end{aligned} \tag{103}$$

Consequently, the optimal user control input signal is

$$v^*(t) = \int_0^t w(\tau)d\tau,\ v^*(0) = 0. \tag{104}$$

Figure 23 was generated by separating a swarm of two classes with the optimal broadcast user input $v^*(t)$. The attention-effort cost is illustrated in Figure 23a, while Figure 23b and 23c illustrate the instantaneous effort and attention. The attention-effort cost increases steadily, because a constant input signal is required to keep the two classes of robots separated. The instantaneous effort ramps up to separate the two classes of robots by a distance of $\Delta = 0.2$, which also requires some attention. Once the two classes are separated, the (instantaneous) attention is zero, while effort is constant, but non-zero, because maintaining the separation distance is a dynamic equilibrium.

Scalability can be calculated by consecutively adding one more robot to each class. In this example, the new robots are initially located at the centroid of the other robots in their



(a) Cost $J_{AE}(v^*)$ up to time t (Attention-Effort)

(b) $\|v^*(t)\|$ (Effort)

(c) $\|\dot{v}^*(t)\|$ (Attention)

Figure 23: A user's estimated attention and effort while separating swarm of $|\mathcal{C}_1| = 10, |\mathcal{C}_2| = 30$ robots (solid) and $|\mathcal{C}| = 11, |\mathcal{C}_2| = 31$ robots (dashed).

class. Figure 24 illustrates the increased attention-effort cost of completing the "same" task with an extra robot uniformly randomly added to either class. The increase in cost is mainly attributed to an increase in effort as illustrated by the dashed, red line, while attention has only marginally increased as illustrated by the dash-dotted, black line . The scalability metric for this particular task has to be approximated, because the constant non-zero effort continuously increases the attention-effort cost. Therefore, we will approximate scalability by examining the attention-effort costs at a point in time when separation has been achieved, attention is zero, and effort is constant. Once again, we approximate $\Sigma(n)$

by the slope of a linear fit on the attention-effort cost. Consequently, $\Sigma(n) = 5010.4n$, which translates to a 2.2% cost increase over the original cost for every new robot. In comparison, broadcast control in the rendezvous task in Section 2.4.4 incurred a 1.65% cost increase over the original cost for every new robot.

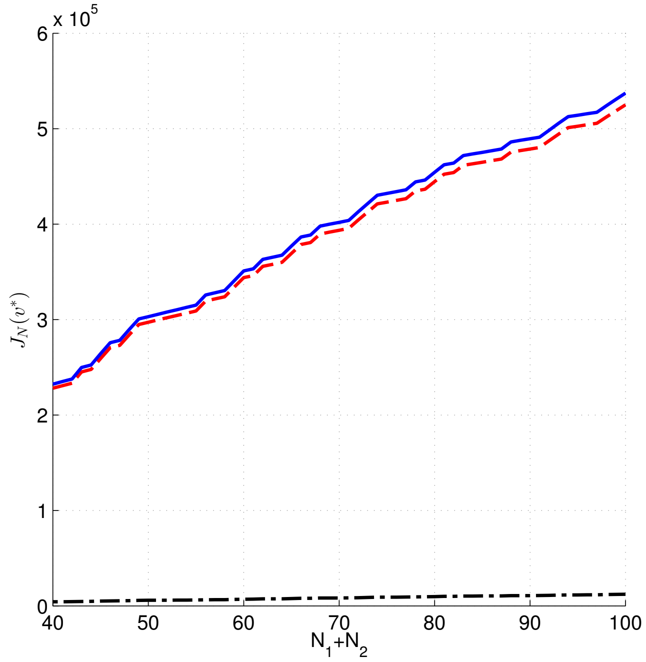


Figure 24: Growth of attention (black dash-dotted), effort (red dashed), and attention-effort (blue solid) cost for separating a swarm of N_1 robots of class $\mathcal{C}_1$ and N_2 robots of class $\mathcal{C}_2$.

4.1.5 Conclusions

We have demonstrated that it is possible to separate a heterogeneous swarm of mobile robots into homogeneous clusters with a broadcast input signal. This heterogeneous swarm can be composed of two, three, or an arbitrary number of different classes of robots, which interact with other robots independent of their class, but react differently to the broadcast input signal. Since we are controlling the swarm using a single broadcast input signal, this control structure lends itself to human-swarm interactions. A user can interact with the entire swarm of robots by pushing a joystick with sufficient force in a direction along which the separation should be achieved. However, we have also demonstrated that broadcast control in this separation task scales poorly with effort, while scaling well with attention.

Consequently, we either need to design a different HSI control structure for this task, or add scaling to the input controller to compensate for the increased effort required to separate the swarm of robots.

4.2 Computing Group Sizes for Successful Predator-Prey Interactions

All HSI examples introduced up to this point involved an arbitrary number of robots in the swarm. Why, for example, are ten robots used to sort a stack of packages, or why are we separating a swarm of 100 robots of $\mathcal{C}_1$ and 75 robots of $\mathcal{C}_2$? In general, if we have a task $\mathcal{T}_H$, how many robots are needed for the user to successfully complete this task? In this section, we answer this question in the context of a pursuit-evasion task (as we have done in [30]), where a user is required to deploy a team of N robots to capture an evasive robot. Since nature provides us with numerous pursuit-evasion example in the form of predator-prey interactions, we have decided to take inspiration from one specific example: Lions, unlike Cheetahs [39], form a so-called "catcher's mitt" formation to capture gazelles and other prey with superior agility. Typically, three to five lionesses are involved in the hunt, with the dominant female positioned at the center, while the remaining females flank her in the wing positions [39]. Consequently, our pursuit-evasion task will resemble lions hunting gazelles.

To be able to predict how many predators are needed, we need models that are simple enough to analyze, yet expressive enough to allow for a parametrization that captures different predator strategies. It should already be stressed at at this point, however, that our aim is *not* bio-mimicry despite our robot's resemblance to lions and gazelles as illustrated in Figure 25. We characterize a prey's behavior in the presence of predators with a set of parameters. These parameters allow us to adjust the behavior of the prey, such that the predators can hunt a variety of strong, weak, brave, or skittish prey. Similarly, we define parameters that describe the hunting strategy of the predators, such as their speed and formation as a group during the hunt. These models also allow us to analytically and algorithmically answer whether a capture strategy for a multi-agent robot team will be successful against a particular prey. Very few results (for example, [51]) exist that relate the number of robots in a swarm or multi-robot team to the success at achieving the task at hand. This work can thus be thought of as one particular attempt at addressing this previously neglected question concerning how large a team or swarm of mobile robots needs to be to achieve a particular task.

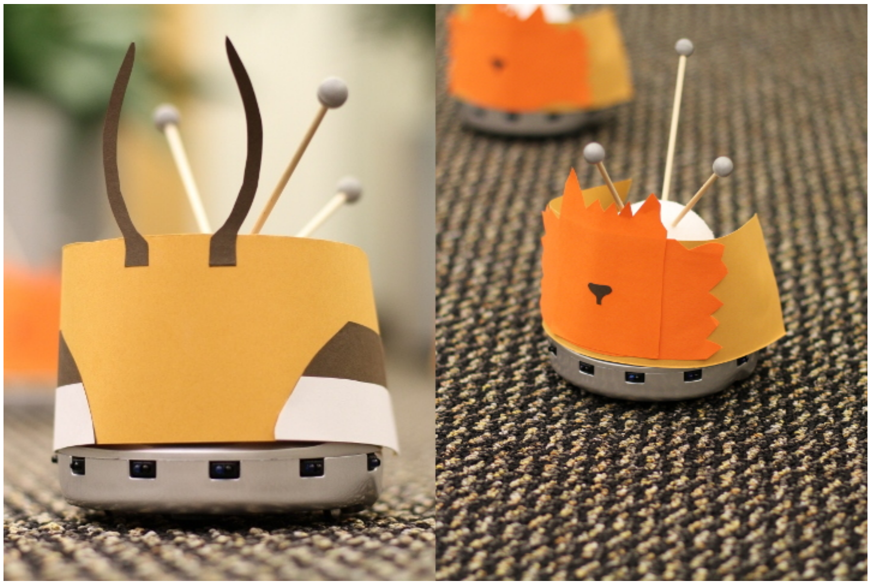


Figure 25: Mobile robots are surrogates for the prey (gazelles) and predators (lions).

4.2.1 Geometric Hunting Strategies

Since we are inspired by the predator-prey interactions between lions and gazelles, where the predators hunt as a team in a strategic geometric configuration (i.e., a formation) to overcome the evasive abilities of the prey, we focus our attention on "catcher's mitt"-like formations of N predators, as illustrated in Figure 26. The dominant predator (to borrow from the lion terminology) in the team takes the central position on the x-axis, while the remaining predators are spread symmetrically to either side in the "wing" positions. Each predator is separated from its nearest team member by a distance Δ, and the predators can be thought of being uniformly distributed on a circle of radius r centered on the x-axis. Furthermore, ℓ is the distance separating the outermost wing predators and the prey along the x-axis (denoted by a cross in the figure).

Counting from the center predator, the zeroth level is the center predator, the first set of wing predators is the first level, the next set of wing predators is the second level, and

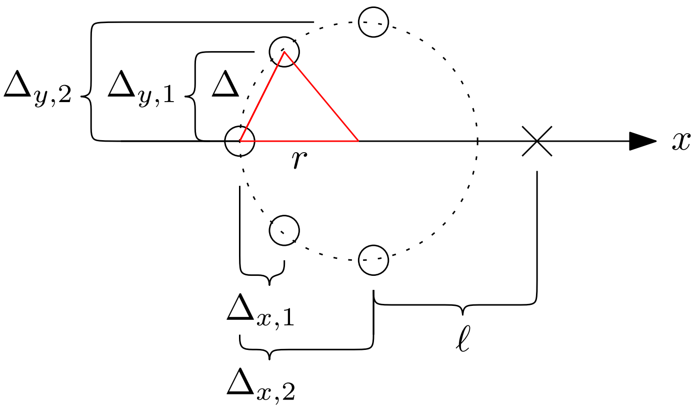


Figure 26: A group of $N = 5$ predators (circles) hunting a single prey (cross).

so on. There are $m = (N - 1)/2$ levels of wing predators. And, for each predator, we can compute its x and y position in the formation with respect to the center predator:

$$\begin{aligned}\Delta_{x,k} &= r - r\cos\left(2k\sin^{-1}\left(\frac{\Delta}{2r}\right)\right)\\ |\Delta_{y,k}| &= r\sin\left(2k\sin^{-1}\left(\frac{\Delta}{2r}\right)\right),\end{aligned} \tag{105}$$

where $k = \{0, 1, \ldots, m\}$, and where a k-th level wing predator is located at $(\Delta_{x,k}, \Delta_{y,k})$ with respect to the center predator. We can also compute how far a predator at the k-th level is from the prey along the x-axis, i.e.,

$$\lambda_k(\ell) = \ell + \Delta_{x,m} - \Delta_{x,k}. \tag{106}$$

Then, the total distance separating the prey and a predator at the k-th level is,

$$\delta_k(\ell) = \sqrt{\lambda_k(\ell)^2 + \Delta_{y,k}^2}. \tag{107}$$

The center predator is the zeroth level, such that $\lambda_0(\ell) = \delta_0(\ell) = \ell + \Delta_{x,m}$, since $\Delta_{x,0} = \Delta_{y,0} = 0$.

Now that we have described the initial *geometry* of the hunt, we are ready to describe its *dynamics*. The predators hunt together along the x-axis in the direction of the prey with a constant, scalar velocity v (incidentally, also the exogenous user input) and without deviating from their configuration. If $p_l(t) \in \mathbf{R}^2$ is the position of a particular predator at time t (where we somewhat facetiously use the subscript l to denote "lion"), then

$$\dot{p}_l(t) = \begin{bmatrix}\dot{x}_l(t)\\ \dot{y}_l(t)\end{bmatrix} = \begin{bmatrix}v\\ 0\end{bmatrix}. \tag{108}$$

The dynamics of the prey with respect to a particular predator at the k-th level are encoded by a continuous, monotonically non-increasing function $\Gamma : \mathbf{R} \to \mathbf{R}$, such that it starts off at some value $\Gamma_{\max}$ and ends up at 0 after a finite interval, i.e.,

$$\Gamma(\psi) = \begin{cases} \Gamma_{\max} & \text{if } \psi \leq \Delta_{\max} \\ 0 & \text{if } \psi \geq \Delta_d \end{cases}, \tag{109}$$

where $\Delta_{\max}$ is the distance from a predator when the prey will start evading with maximum effort, and Δ_d is the distance from a predator when the prey can detect the predator, and where we assume that $\Delta_{\max} < \Delta_d$.

If we denote the contribution to the prey dynamics of a predator at the k-th level as

$$\Gamma_k(\ell) = \Gamma(\delta_k(\ell)) = \Gamma\left(\sqrt{\lambda_k(\ell)^2 + \Delta_{y,k}^2}\right), \tag{110}$$

then we can write down the full dynamics of the prey for the N predator hunt. If $p_g(t) = [x_g(t), y_g(t)]^T \in \mathbf{R}^2$ is the position of the prey at time t (where we use the subscript g to denote "gazelle"), then

$$\dot{x}_g = \Gamma_0(\ell) + 2\sum_{k=1}^{m} \frac{\Gamma_k(\ell)\lambda_k(\ell)}{\sqrt{\lambda_k(\ell)^2 + \Delta_{y,k}^2}}, \tag{111}$$

and $\dot{y}_g = 0$. The symmetric configuration of wing predators during the hunt ensures that any component along the y-axis is always equal to zero.

For the purpose of generality we will use Equation (111), but we will ground our results on a slightly modified version of an actual swarm interaction dynamic–originally proposed in [44]. Under this model the contribution from a predator located at the k-th level is

$$\Gamma_k^{'}(\ell) = \begin{cases} \beta\sqrt{\frac{\gamma}{2e}} & \lambda_k(\ell) \leq \sqrt{\frac{\gamma}{2}} \\ \beta\lambda_k(\ell)e^{-\frac{\delta_k(\ell)^2}{\gamma}} & \sqrt{\frac{\gamma}{2}} < \lambda_k(\ell) < \infty \\ 0 & \lambda_k(\ell) \geq \infty \end{cases}, \tag{112}$$

such that the full dynamics of the prey for a N predator hunt can be written as

$$\dot{x}_g = \Gamma_0^{'}(\ell) + 2\sum_{k=1}^{m} \Gamma_k^{'}(\ell), \tag{113}$$

and $\dot{y}_g = 0$, where β, and γ are certain behavioral parameters. In fact, the interaction dynamics described in [44] defines an additional parameter, which dictates whether an agent is drawn towards another agent. Some predators, for example the anglerfish *Lophius piscatorius*, are able to attract their prey. However, lions and most predators do not hunt in this manner, and therefore we exclude this phenomenon from the model. Instead we focus our attention on the two parameters β and γ. These two parameters together characterize with how much effort the prey attempts to escape from a predator as the distance to the predator decreases. The maximum effort with which the prey evades a predator is $\beta\sqrt{\frac{\gamma}{2e}}$, where as its maximum effort is captured by $\Gamma_{\max}$ in Equation (109). Moreover, γ parameterizes how close the predators can approach the prey before it evades with maximum effort. This distance is captured by $\Delta_{\max}$ in Equation (109), while Δ_d in Equation (109) captures the distance at which the prey does not detect the predators.

Although all prey are scared of predators and will attempt to escape them, we do not need to impose that different prey are alike. Therefore, we have two prey parameters that can vary: β and γ (or $\Gamma_{\max}$ and $\Delta_{\max}$). We also consider three additional parameters that characterize the formation and dynamics of the predators. A group of predators will move together in a formation parameterized by Δ and r with a constant velocity v, which gives us three additional "knobs" by which the hunt dynamics can be characterized, namely v, Δ, and r.

4.2.1.1 One Predator

The first predator-prey scenario is a single predator hunting a single prey. Figure 27 shows the predator-prey configuration we consider in the scenario. This particular configuration,

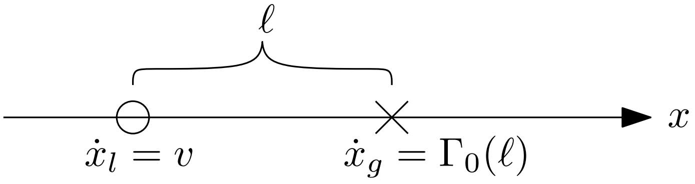


Figure 27: A single predator (circle) hunting a single prey (cross).

which places the predator and the prey on the line $y = 0$, allows us to only consider:

$$\dot{x}_g = \Gamma_0(\ell) = \Gamma(\ell). \tag{114}$$

Since we are interested in hunting, let us consider that a capture occurs when ℓ, the distance separating the predator from the prey, is equal to zero. We want to find the velocity of the predator, v, that guarantees that capture occurs.

Theorem 2. *If Γ_{max} is the maximal effort the prey can use to evade the predator, then the predator is able to the capture the prey if and only if the predator's velocity is $v > \Gamma_{\text{max}}$.*

This result is not particularly surprising. What it simply says is that the predator's hunt velocity v must be greater than the prey's maximal evasion velocity, which is, for example, how solitary Cheetahs hunt. However, the proof of this statement will be useful for more complex situations and, as such, we do include it below:

Proof. Let ℓ^* be the distance separating the predator and the prey that maximizes the derivative $\dot{\ell}$, where

$$\dot{\ell} = \dot{x}_g - \dot{x}_l = \Gamma(\ell) - v. \tag{115}$$

Then the predator is able to capture the prey if and only if $\dot{\ell}(\ell^*) < 0$. By construction of $\Gamma(\psi)$, we know that $\Gamma(\ell^*)$ is maximized when $\ell^* \leq \Delta_{\text{max}}$. Therefore,

$$\dot{\ell}(\ell^*) = \Gamma(\ell^*) - v < 0, \tag{116}$$

which means that

$$\Gamma_{\text{max}} - v < 0 \;\Leftrightarrow\; \Gamma_{\text{max}} < v, \tag{117}$$

which means that given ℓ^*, the predator can capture the prey if $v > \Gamma_{\text{max}}$. □

If we consider the predator-prey interaction dynamic defined in Equation (113), we can apply this theorem to find the velocity v of the predator that is guaranteed to capture a prey parameterized by β and γ. The configuration shown in Figure 27 allows us to formulate the prey dynamics as:

$$\dot{x}_g = \beta \ell e^{-\frac{\ell^2}{\gamma}}.$$

We can detect a capture by checking if $\dot{\ell}(\ell^*)$ is negative, where $\dot{\ell}$ is now defined in the following way:

$$\dot{\ell} = \dot{x}_g - \dot{x}_l = \beta \ell e^{-\frac{\ell^2}{\gamma}} - v.$$

We solve for ℓ^*, the value of ℓ that maximizes $\dot{\ell}$, by setting the derivative of $\dot{\ell}$ with respect to ℓ to zero and simplifying:

$$\begin{aligned} \frac{\partial \dot{\ell}}{\partial \ell} = \beta e^{-\frac{\ell^2}{\gamma}} - \frac{2\beta\ell^2}{\gamma} e^{-\frac{\ell^2}{\gamma}} &= 0 \\ \left(1 - \frac{2\ell^2}{\gamma}\right) \beta e^{-\frac{\ell^2}{\gamma}} &= 0 \\ \left(1 - \frac{2\ell^2}{\gamma}\right) &= 0 \\ \sqrt{\frac{\gamma}{2}} &= \ell^* \end{aligned}$$

We plug ℓ^* back into the equation for $\dot{\ell}$ and set this equation equal to less than zero to solve for v:

$$\dot{\ell}(\ell^*) = \beta \sqrt{\frac{\gamma}{2}} e^{-\frac{\sqrt{\frac{\gamma}{2}}^2}{\gamma}} - v < 0,$$

i.e.,

$$\beta \sqrt{\frac{\gamma}{2e}} < v. \tag{118}$$

As such, the predator can capture the prey if and only if $v > \beta\sqrt{\frac{\gamma}{2e}}$, when the maximal effort of the prey, $\Gamma_{\max} = \beta\sqrt{\frac{\gamma}{2e}}$, is held on the interval $\ell \leq \sqrt{\frac{\gamma}{2}}$.

Let us briefly examine a concrete example. Figure 28 is a graph of $\dot{\ell}$ as a function of $\ell \in [0, 2]$. The maximum of $\dot{\ell}$ occurs at $\ell^* = \sqrt{\frac{\gamma}{2}} \approx 0.274$, which agrees with our derivation. We also show two cases for the predator's velocity: 0.1 m/s and 0.3 m/s. First, if the predator's velocity is 0.1 m/s, then the maximum of $\dot{\ell}$ is above the solid line, meaning that $\dot{\ell}(\ell^*) > 0$ and the predator is not fast enough to capture the prey,

$$v = 0.1 < 1.5\sqrt{\frac{0.15}{2e}} \approx 0.249.$$

In the second case, the predator's velocity is 0.3 m/s, such that $\dot{\ell}(\ell^*) < 0$, and the predator is able to capture the prey, since $v = 0.3 > \Gamma_{\max} = 0.249$.

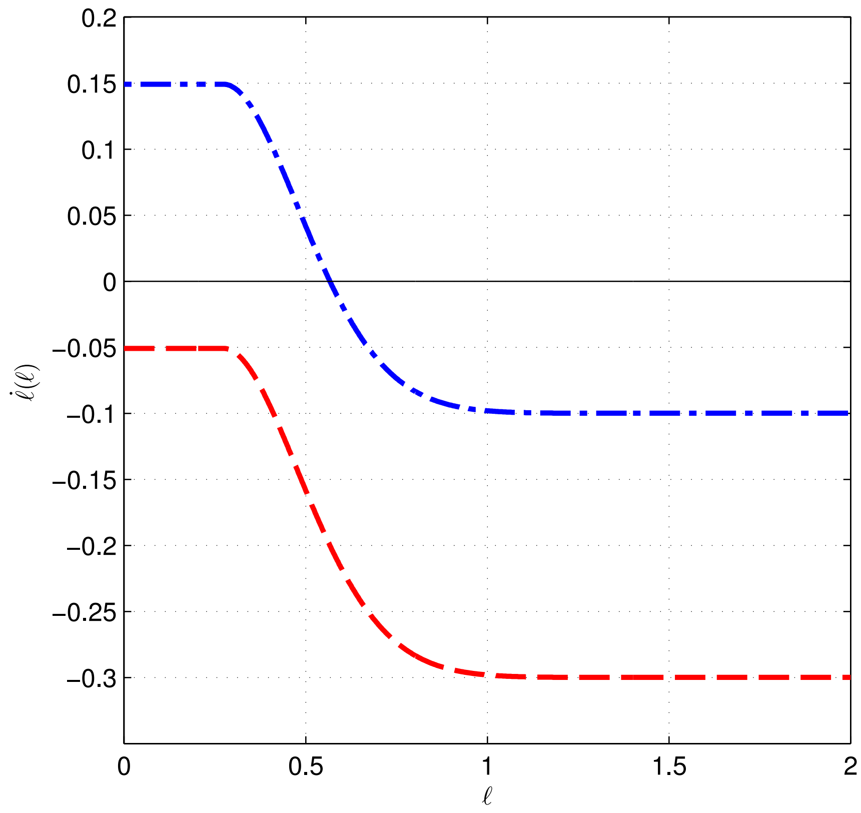


Figure 28: The change in distance between prey and predator, $\dot{\ell}$, as a function of ℓ when $\beta = 1.5$, $\gamma = 0.15$. The blue dashed-dotted curve corresponds to a predator velocity of $v = 0.1$ m/s, while the red dashed curve correspond to a predator velocity of $v = 0.3$ m/s.

4.2.1.2 Three Predators

The previous section gave us an expression for the velocity required for a single predator to capture a prey. However, predators are typically not as fast as the prey, so they hunt cooperatively in groups to increase their chance of success. Let us consider a configuration with three predators, as illustrated in Figure 29. We want to derive a similar capture condition on the velocity of the predators as in the previous case; however, in this scenario $\ell = 0$ implies that capture is achieved when the wing predators at the m-th level achieve "crossover", i.e. these leading wing predators pass the prey. We first show that it is indeed sufficient to just consider the crossover condition and moreover note that this condition is very much in line with the result on cooperative pursuit-evasion games that rely of driving the evader into the convex hull spanned by the pursuers, as was done in [73].

Theorem 3. *If at time t_c, the two leading wing predators reach the prey with respect to the*

x-axis, i.e. $x_l^w(t_c) = x_g(t_c)$, then there exists a strategy which guarantees capture after this "crossover" event.

Proof. If crossover occurs at time t_c, then this event implies that $0 < \dot{x}_g(t_c) \leq v$. We can expand $\dot{x}_g$, such that

$$\dot{x}_g = \dot{x}_g^w + \dot{x}_g^{L\backslash w}, \tag{119}$$

where $\dot{x}_g^w$ is the contribution to $\dot{x}_g$ from the leading wing predators, and $\dot{x}_g^{L\backslash w}$ is the contribution from all predators excluding the leading wing predators.

Since $\dot{x}_g^w(t_c)$, the contribution from the leading wing predators, is zero when they are at the crossover point, we can write $\dot{x}_g(t_c) = \dot{x}_g^{L\backslash w}(t_c)$. At time t_c all predators except the leading wing predators stop, such that $\dot{x}_l^{L\backslash w}(t_c) = 0$. By continuity, there exists $\Delta T > 0$, such that

$$\dot{x}_g(t) > 0, \forall t \in (t_c, t_c + \Delta T]. \tag{120}$$

But on this interval, $\dot{x}_g^{L\backslash w}(t) < \dot{x}_g^{L\backslash w}(t_c) = \dot{x}_g(t_c)$ and since $\dot{x}_l^w(t) = v$, we have $\dot{x}_g^{L\backslash w}(t) < 0$. In other words, the leading wing predators pass the prey and contribute a push towards the rest of the predators.

Let us pick $t' \in (t_c, t_c + \Delta T]$ and let $v' = \dot{x}_g(t') < v$. Since $\dot{x}_g(t) \leq \dot{x}_g(t')$ $\forall t \geq t'$ and $v > v'$, there exists a T such that

$$|x_g(T) - x_l^w(T)| > \frac{v' y_l^w(t_c)}{v}, \tag{121}$$

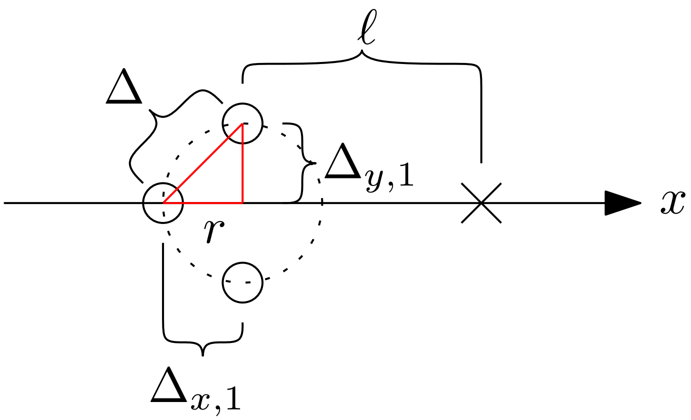


Figure 29: A group of three predators (circles) hunting a single prey (cross).

where $y_l^w(t_c)$ is the position of the leading wing predators in the y-direction at time t_c. Then, at time T, a possible capture strategy is to first stop the motion of the leading wing predators in the x-direction, such that $\dot{x}_l^w(T) = 0$. Next, let these wing predators converge on the x-axis, such that $y_l^w(t) \to 0$, $t > T$. Once they have converged, $y_l^w(t) = 0$ and the leading wing predators can regroup with the rest of the predators, such that $|x_l^{L\backslash w}(t) - x_l^w(t)| \to 0$, $t > T$ and capture is achieved. □

We are now justified in simply using the event that the leading wing predators reach the crossover point, $\ell = 0$, as a condition for guaranteed capture. Consequently, let us return to the three predator scenario as depicted in Figure 29. We define the derivative, $\dot{\ell}(\ell)$, as

$$\begin{aligned}\dot{\ell}(\ell) &= \Gamma_0(\ell) + 2\frac{\Gamma_1(\ell)\lambda_1(\ell)}{\sqrt{\lambda_1(\ell)^2 + \Delta_{y,1}^2}} - v' \\ &= \Gamma(\ell + \Delta_{x,1}) + 2\frac{\Gamma\left(\sqrt{\ell^2 + \Delta_{y,1}^2}\right)}{\sqrt{\ell^2 + \Delta_{y,1}^2}}\ell - v'.\end{aligned} \tag{122}$$

We want to find a configuration of predators, Δ and $v' < v$, such that we can capture a single prey with slower predators.

Theorem 4. *There exists $\Delta_y > 0$, such that captured is guaranteed when $v' < v$, if and only if $\Delta_x > \Delta_{\max}$.*

Proof. For capture to be possible with three predators, we need to satisfy:

$$\Gamma(\ell^* + \Delta_x) + 2\frac{\Gamma\left(\sqrt{(\ell^*)^2 + \Delta_y^2}\right)}{\sqrt{(\ell^*)^2 + \Delta_y^2}}\ell^* < v', \tag{123}$$

where ℓ^* maximizes Equation (122), $\Delta_x = \Delta_{x,1}$, and $\Delta_y = \Delta_{y,1}$. We also want to satisfy that $v' < v$, therefore:

$$\Gamma(\ell^* + \Delta_x) + 2\frac{\Gamma\left(\sqrt{(\ell^*)^2 + \Delta_y^2}\right)}{\sqrt{(\ell^*)^2 + \Delta_y^2}}\ell^* < \Gamma_{\max}. \tag{124}$$

The contribution from the center predator has to satisfy

$$\Gamma(\ell + \Delta_x) < \Gamma_{\max},$$

otherwise it is impossible to satisfy Equation (124). Its contribution is less than $\Gamma_{\max}$ for all $\ell \geq 0$ if and only if $\Delta_x > \Delta_{\max}$ by construction of $\Gamma(\psi)$.

Suppose $\Delta_x > \Delta_{\max}$, then we can rewrite and rearrange Equation (124) as,

$$\begin{aligned} \Gamma_{\max}(1-\epsilon) + 2\frac{\Gamma\left(\sqrt{(\ell^*)^2+\Delta_y^2}\right)}{\sqrt{(\ell^*)^2+\Delta_y^2}}\ell^* &< \Gamma_{\max} \\ \frac{\Gamma\left(\sqrt{(\ell^*)^2+\Delta_y^2}\right)}{\sqrt{(\ell^*)^2+\Delta_y^2}}\ell^* &< \frac{\epsilon}{2}\Gamma_{\max}, \end{aligned} \tag{125}$$

where $\epsilon > 0$ and $\epsilon \in (0,1]$. We can apply one more bound to Equation (125),

$$\begin{aligned} \frac{\Gamma\left(\sqrt{(\ell^*)^2+\Delta_y^2}\right)}{\sqrt{(\ell^*)^2+\Delta_y^2}}\hat{\ell} \leq \frac{\Gamma_{\max}}{\sqrt{(\ell^*)^2+\Delta_y^2}}\ell^* &< \frac{\epsilon}{2}\Gamma_{\max} \\ \frac{\ell^*}{\sqrt{(\ell^*)^2+\Delta_y^2}} &< \frac{\epsilon}{2} \end{aligned} \tag{126}$$

We can satisfy Equation (126) by picking Δ_y sufficiently large, such that the inequality holds independent of ℓ^*. □

Corollary 5. *Given that $\Delta_y > 0$ and $\Delta_x > \Delta_{\max}$, then $\Delta > \Delta_{\max}$ is a lower bound on Δ that must hold for capture to be possible when $v' < \Gamma_{\max} < v$ and Δ_y is sufficiently large.*

Returning to the predator-prey dynamics defined in Equation (113), if $\Delta_x > \sqrt{\frac{\gamma}{2}}$, then three predators are able to cooperatively capture a prey at a velocity v' that is smaller than $\beta\sqrt{\frac{\gamma}{2}}$, i.e the velocity v that is required by a single predator to capture the same prey. The expression for $\dot{\ell}$ with the addition of two wing predators to the single predator is,

$$\dot{\ell} = \beta(\ell+\Delta_x)e^{-\frac{(\ell+\Delta_x)^2}{\gamma}} + 2\beta\ell e^{-\frac{(\ell^2+\Delta_y^2)}{\gamma}} - v' \tag{127}$$

or

$$\dot{\ell} = \Lambda^c + \Lambda^w - v',$$

where,

$$\Lambda^c = \beta(\ell+\Delta_x)e^{-\frac{(\ell+\Delta_x)^2}{\gamma}}$$

is the contribution from the center predator, and

$$\Lambda^w = 2\beta\ell e^{-\frac{(\ell^2+\Delta_y^2)}{\gamma}}$$

is the contribution from the wing predators.

First, the contribution from the center predator, Λ^c, attains its maximum in the region $\ell \geq 0$ when $\ell = 0$, i.e. the maximum push from the center predator occurs when the wing predators achieve crossover.

$$\Lambda^c_{\text{max}} = \Lambda^c\left(\ell = 0\right) = \beta \Delta_x e^{-\frac{\Delta_x^2}{\gamma}}.$$

Second, the contribution from the wing predators, Λ^w is maximized when $\ell = \sqrt{\frac{\gamma}{2}}$.

$$\Lambda^w_{\text{max}} = \Lambda^w\left(\ell = \sqrt{\frac{\gamma}{2}}\right) = 2\beta\sqrt{\frac{\gamma}{2e}}e^{-\frac{\Delta_y^2}{\gamma}}$$

As in the case of the single predator, we again want $\dot{\ell} < 0$, such that $\Lambda^c + \Lambda^w < v'$, and since $\Lambda^c_{\text{max}} + \Lambda^w_{\text{max}} > \Lambda^c + \Lambda^w$ and $v' < v$:

$$\Lambda^c_{\text{max}} + \Lambda^w_{\text{max}} < v' < v$$

If $\Delta_x > \sqrt{\frac{\gamma}{2}}$, then using this inequality and $v > \sqrt{\frac{\gamma}{2e}}$, we can show the following:

$$\begin{aligned} \beta \Delta_x e^{-\frac{\Delta_x^2}{\gamma}} + 2\beta\sqrt{\frac{\gamma}{2e}}e^{-\frac{\Delta_y^2}{\gamma}} &< v \\ \beta \Delta_x e^{-\frac{\Delta_x^2}{\gamma}} + 2e^{-\frac{\Delta_y^2}{\gamma}} v &< v \\ (1-\epsilon)v + 2e^{-\frac{\Delta_y^2}{\gamma}} v &< v \\ 2e^{\frac{-\Delta_y^2}{\gamma}} v &< \epsilon v \\ e^{\frac{-\Delta_y^2}{\gamma}} &< \frac{\epsilon}{2} \end{aligned}$$

There exists a Δ_y sufficiently large for which this inequality will hold, since $\epsilon > 0$ and $\epsilon \in (0, 1]$. As a consequence, there exists Δ_y large enough, such that $v > v'$ holds.

Let us inspect $\dot{\ell}$ from Equation (127) in Figure 30 and check whether we can say something equally useful as in the single predator case. We choose the parameters $\beta = 1.5$, $\gamma = 0.15$, $v = 0.225$, $\Delta = 0.75$, and $\phi = 0.75/(2\sin(\pi/8))$ for which the two conditions hold. The dashed line is the component of $\dot{\ell}$ that is contributed by the center predator, while the dashed-dotted line is the contribution from the two wing predators. The solid line is the total contribution to $\dot{\ell}$ from all predators. A velocity of 0.225m/s is not sufficient for a single predator to capture the prey (i.e., $\dot{\ell}(\ell) > 0$ and recall that the contribution from the center predator is offset by Δ_x); however, it is sufficient for three predators to cooperatively capture this particular prey. The dashed-dotted line has a negative maximum; however, it is unclear whether there exists a capture strategy for two predators passing the prey to either side that implies capture; therefore, we will consider only the cases where we have a center predator.

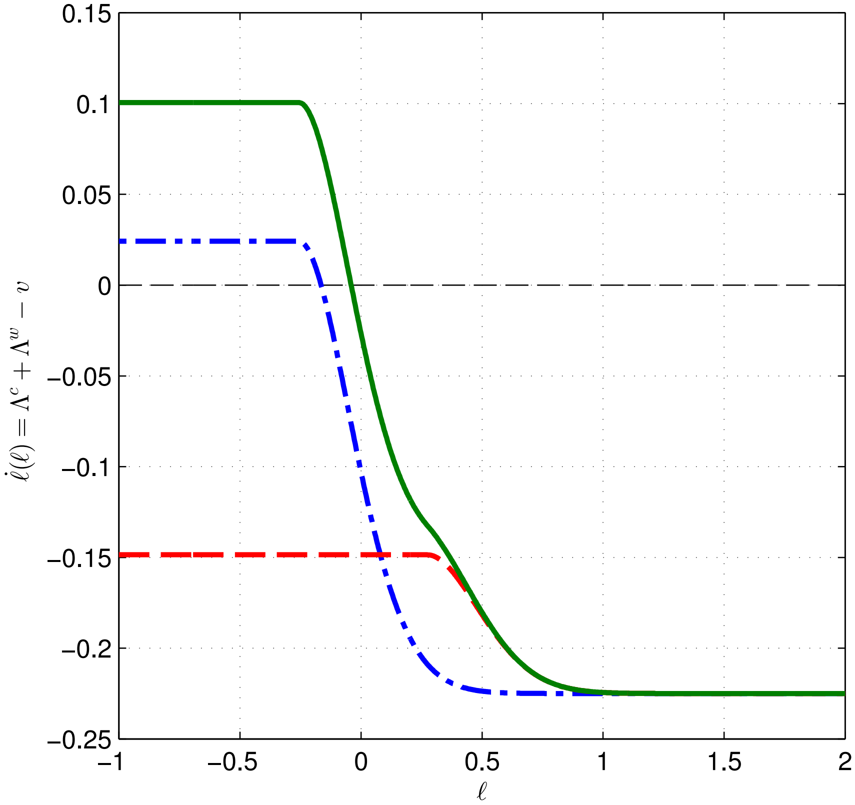


Figure 30: A predator-prey hunt is parameterized by $\beta = 1.5$, $\gamma = 0.15$, $v = 0.225$, $\Delta = 0.75$, $r = 0.75/(2\sin(\pi/8))$. The dashed and dashed-dotted lines are the components contributed to $\dot{\ell}(\ell)$ by the center and wing predators minus v respectively. The solid line is $\dot{\ell}(\ell)$.

4.2.1.3 N Predators

Let us return to the configuration with N predators and a single prey, as illustrated in Figure 26, since we are originally interested in the question of how many predators it takes to capture a particular prey. We want to show that a group of N predators can capture a prey at a velocity v' that is less than the velocity v required for a single predator to capture the same prey.

Theorem 5. *There exists a* $\Delta_y = \min\{\Delta_{y,1}, \ldots, \Delta_{y,m}\}$*, such that* $v' < v$ *and capture is guaranteed, if and only if* $\Delta_{x,m} > \Delta_{\max}$.

Proof. For capture to be possible with N predators, we need to satisfy

$$\Gamma_0(\ell^*) + 2\sum_{k=1}^{m} \frac{\Gamma_k(\ell^*)\lambda_k(\ell^*)}{\sqrt{\lambda_k(\ell^*)^2 + \Delta_{y,k}^2}} < v' \tag{128}$$

where ℓ^* maximizes $\dot{\ell}(\ell)$. We also want to satisfy that $v' < v$, therefore:

$$\Gamma_0(\ell^*) + 2\sum_{k=1}^{m} \frac{\Gamma_k(\ell^*)\lambda_k(\ell^*)}{\sqrt{\lambda_k(\ell^*)^2 + \Delta_{y,k}^2}} < \Gamma_{\max} \tag{129}$$

The contribution from the center predator has to satisfy $\Gamma_0 < \Gamma_{\max}$, otherwise it is impossible to satisfy Equation (129). Its contribution is less than $\Gamma_{\max}$ for all $\ell \geq 0$ if and only if $(r + \Delta_{x,m}) > \Delta_{\max}$ by construction of $\Gamma(\psi)$.

$$\begin{aligned}
\Gamma_{\max}(1-\epsilon) + 2\sum_{k=1}^{m} \frac{\Gamma_k(\ell^*)\lambda_k(\ell^*)}{\sqrt{\lambda_k(\ell^*)^2 + \Delta_{y,k}^2}} &< \Gamma_{\max} \\
\sum_{k=1}^{m} \frac{\Gamma_k(\ell^*)\lambda_k(\ell^*)}{\sqrt{\lambda_k(\ell^*)^2 + \Delta_{y,k}^2}} &< \frac{\epsilon}{2}\Gamma_{\max} \\
\sum_{k=1}^{m} \frac{\Gamma_{\max}\lambda_k(\ell^*)}{\sqrt{\lambda_k(\ell^*)^2 + \Delta_{y,k}^2}} &< \frac{\epsilon}{2}\Gamma_{\max} \\
\sum_{k=1}^{m} \frac{\lambda_k(\ell^*)}{\sqrt{\lambda_k(\ell^*)^2 + \Delta_{y,k}^2}} &< \frac{\epsilon}{2} \\
\sum_{k=1}^{m} \frac{\lambda_k(\ell^*)}{\sqrt{\lambda_k(\ell^*)^2 + \Delta_{y}^2}} &< \frac{\epsilon}{2}
\end{aligned} \tag{130}$$

where $\Delta_y = \min\{\Delta_{y,1}, \ldots, \Delta_{y,m}\}$ and $\epsilon \in (0, 1]$. □

We would also like to show that if there is a configuration of N predators that can capture a prey with a velocity of v', then we can also capture a prey with some configuration of $N+2$ predators at a slower velocity $v'' < v'$.

Corollary 6. *There exists a* $\Delta'_y = \min\{\Delta'_{y,1}, \ldots, \Delta'_{y,m}\}$*, such that* $v'' < v'$ *if and only if* $\Delta'_{x,m} > \Delta_{\max}$.

Proof. We can pick a $\Delta'_y = \min\{\Delta'_{y,1}, \ldots, \Delta'_{y,m}\}$ sufficiently large, such that

$$\sum_{k=1}^{m+1} \frac{\lambda_k(\ell'^*)}{\sqrt{\lambda_k(\ell'^*)^2 + \Delta'^2_y}} < \sum_{k=1}^{m} \frac{\lambda_k(\ell^*)}{\sqrt{\lambda_k(\ell^*)^2 + \Delta^2_y}} \tag{131}$$

where ℓ^* maximizes $\dot{\ell}(\ell)$ for the N predator hunt and ℓ'^* maximizes $\dot{\ell}(\ell)$ for the $N+2$ predator hunt. Satisfying this inequality implies that $N+2$ predators can capture the prey at a velocity $v'' < v'$. □

4.2.2 Algorithm

We proved in the previous section that there exists Δ_y that guarantees that a group of predators can capture a prey under certain conditions (characterized by a set of parameters). For the purposes of the proofs, we have been very conservative with the bounds to show that Δ_y can be made sufficiently large to satisfy the inequalities and thus guarantee capture. Practically speaking, Δ_y can be reasonable (and not necessarily arbitrarily large) depending on the predator and prey parameters selected.

Suppose a user is required to deploy the minimum number of robotic "predators" needed to capture a moving target parameterized by $\Gamma_{\max}$ and $\Delta_{\max}$. The predators are parameterized by v, Δ, r, and T. One way for the user to determine the minimum number of such predators needed to capture this prey can be computed using the algorithm in Table 6.

Table 6: An algorithm for computing the minimum number of predators to capture a prey: Hunting$(\Gamma_{\max}, \Delta_{\max}, v, N_{\max}, \Delta, r, T)$

```
**Input:** Prey parameters $\Gamma_{\max}$ and $\Delta_{\max}$; predator parameters $v$, $\Delta$, $r$, and $T$
**Output:** The minimum $N$ (if it exists) for the given parameters
  **for all** $m = (N_0 - 1)/2$ to $(N_{\max} - 1)/2$ **do**
    **if** $(r - \Delta_{x,m}) > \Delta_{\max}$ **then**
      $[t, \ell(t)] \leftarrow$ ODE45(@dynamics, $[0, T]$, $\ell_0$)
      **if** $\exists t$, $\ell(t) < 0$ **then**
        $N_{\min} \leftarrow 2m + 1$
        **return**
      **end if**
    **end if**
  **end for**
```

T is a new parameter that captures how long the predators will attempt to hunt before running out of energy. Up until now we have assumed $T = \infty$, but we need T to be finite for the algorithm to terminate. $N_{\max}$ is odd and represents the maximum number of predators that can be recruited for the hunt. Since T and $N_{\max}$ are finite, this algorithm may terminate with no successful strategy even if one exists if T or $N_{\max}$ were larger. This limitation is reasonable when these strategies are deployed on robots, because we have a finite amount of time before battery power is expended, or we can only deploy a finite number of robots in the hunt.

4.2.3 Experiment

We validate that is possible to generate a cooperative strategy with the proposed parameterized model that achieves capture by performing an experiment involving robotic "lions" hunting a robotic "gazelle". A differential drive mobile robot platform is used as a robotic surrogate for real predators and prey. These robotic gazelle and lions are provided with positional data from a motion capture system, which allows them to compute the inter-agent distances needed in the dynamics. A low-level controller converts the desired predator and prey motions into the appropriate differential drive velocities needed to actuate the mobile robots.

The experiment consists of two hunting scenarios. The prey and predator parameters, $\beta = 1.5$, $\gamma = 0.15$, and $v = 0.225$m/s are the same for both scenarios, meaning that the same prey and predators participate in both experiments. These parameters were scaled from simulated examples to appropriate values for the hardware environment. In the first scenario, shown in Figure 31, the user deploys a single predator that is unable to capture the prey. We can verify that

$$v = 0.225 < 0.25\sqrt{\frac{0.15}{2e}} \approx 0.25$$

and, therefore, a single robotic lion is not fast enough to capture this particular robotic gazelle.

In the second scenario, shown in Figure 32, the user deploys two more of the same predators, which are able to capture the prey together given the same parameters for β, γ,

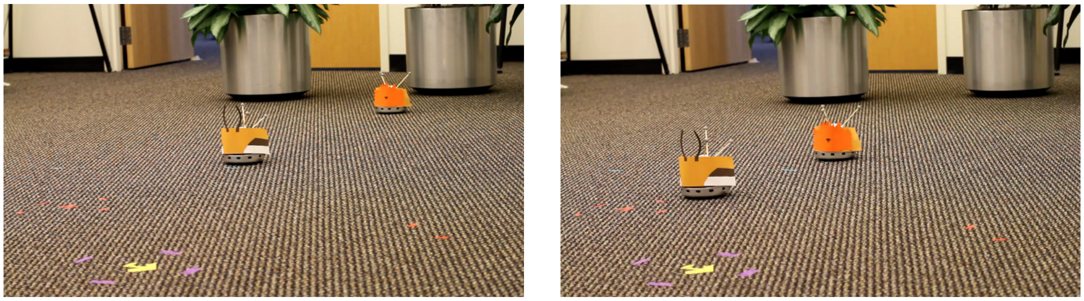


Figure 31: A single robotic “lion” is unable to capture the “gazelle”.

and v, as well as, $\Delta = 0.75$m and $r = 0.75/(2\sin(\pi/8))$. The condition,

$$\Delta = 0.75 > \Delta_x = 0.531 > \sqrt{\frac{0.15}{2}} = 0.274,$$

is satisfied. Figure 33 illustrates the positions of the predators and prey during the hunt. A color gradient is used to denote the progression of time starting with a darker color and ending with a lighter color.

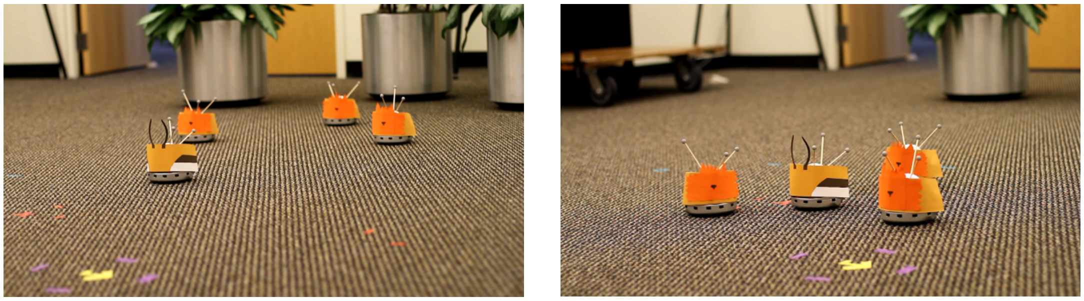


Figure 32: Three robotic “lions” are able to cooperatively capture the “gazelle”.

The prey is not agile enough to escape from the trio of predators; however, it is agile enough to escape a single predator in the first scenario. Such scenarios are often observed in nature when predators (such as lions) sneak up on their prey and the prey is unable to detect the predators early enough and escape. The strategy for capture illustrated in Figures 32 and 33 is for the wing predators to converge (without collision) in front of the prey once they have achieved crossover and allow the center predator to capture the prey.

4.2.4 Conclusion

The experiment validates that we can qualitatively recover hunting strategies from nature and deploy these strategies on mobile multi-agent robot teams. Therefore, we have a valid

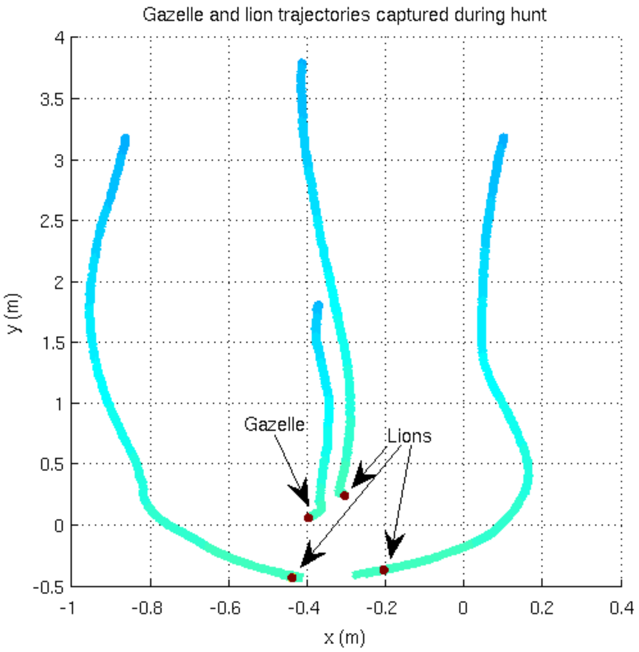


Figure 33: Recorded positions of a robotic gazelle and three robotic lions during the second scenario.

set of parameters that we can use to characterize the dynamics of the predator-prey interactions and decide whether a particular capture strategy will be successful. This abstraction can be naturally extended by incorporating additional, more expressive biologically-inspired parameters, such as a maximum evasion velocity for the prey or variations among predators in the group, which correspond to actuator limits, wheel slip, and battery levels on the robotic platforms. More importantly, we have set the ground work for asking and answering the question, *how many robots are needed to successfully complete a task?* This is a fundamental question to not only ask in the context of HSIs as we have done here, but for multi-robot applications in general.

4.3 Interactions with a Deformable Medium-based Input Controller

We have shown in the Chapter 3 that single-leader networks are often not an effective way to form geometric shapes with a swarm of mobile robots. However, we still want to give a user a single input controller that is amenable to controlling all robots collectively as we have done in Section 4.1. We previously used a traditional joystick to use the leader robot to push and pull the other robots into a formation. Another approach to moving the robots into the correct position is to stretch and bend the swarm of robots into the desired geometric shape. A traditional joystick does not afford these kind of actions, but a deformable medium affords stretching and bending. Therefore, if we can map these affordances to decentralized controllers that achieve these actions with a swarm of mobile robots, then a user can form shapes with the robots by simply molding instrumented clay. We devised one particular implementation of such a deformable medium controller for swarms of mobile robots in [34]. This input controller allows a user to form several geometric shapes with swarms of six and fifteen mobile robots. The framework that implements this human-swarm interaction consists of an image recognition algorithm for classifying the shape of a piece of clay coupled with a library of distributed control laws that globally achieve the specified shape with the robots.

The contribution of this HSI abstraction is not in the application of computer vision for recognizing clay shapes or in the design of decentralized controllers for achieving a geometric formation, but rather in the use of a deformable medium as a joystick to manipulate the swarm of mobile robots. Molding clay is an interaction that is likely more amenable to forming geometric shapes than pushing and pulling on a string. Bending and stretching the swarm of robots collectively consequently seems to be easier than trying to move individual robots into particular positions by pushing and pulling on robots through a single leader robot. This interaction encapsulates the user from the difficult task of moving the robots into position, and it establishes a tractable one-to-one mapping between the shape of the clay and the formation of the mobile robot swarm.

4.3.1 Flowchart

The objective of our HSI abstraction is to provide the user with an input controller for guiding a swarm of mobile robots into a particular formation by molding clay. Rather than instrumenting the clay itself, this abstraction for interacting with the swarm is based on a combination of computer vision algorithms for monitoring the clay interface, and distributed controllers for forming the clay shape with the robots. This process can be divided into two phases–an image recognition phase and a swarm control phase–illustrated in Figure 34. Whenever the user presents a new shape (part of the shape library) by modifying the clay controller, the image recognition algorithm analyzes the observed image and selects the corresponding control law (part of the control law library) to be deployed on the robots n the swarm. Consequently, the swarm of mobile robots converges to the shape formed by the user with the clay.

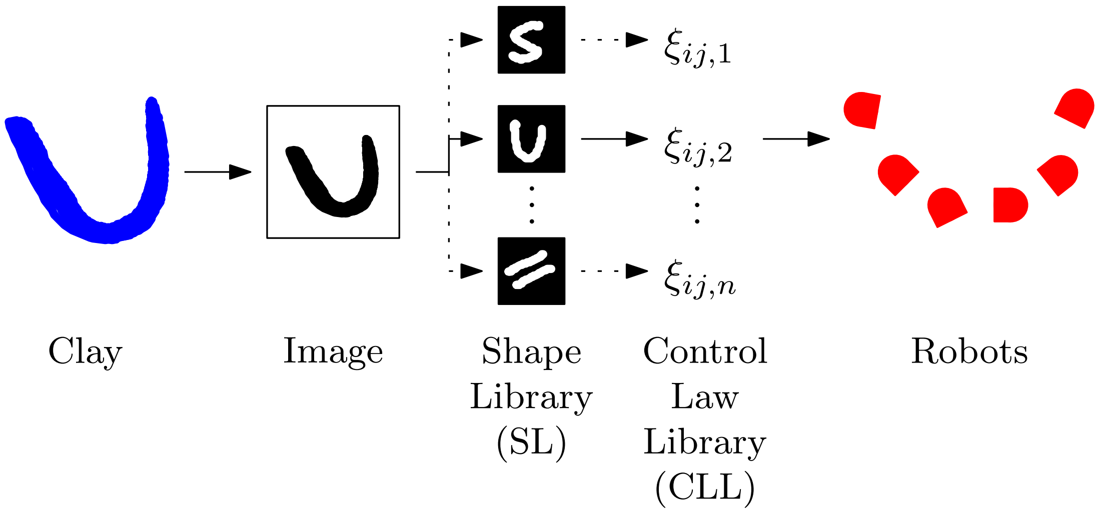


Figure 34: The first three parts of the framework are included in the image recognition phase, while the last two parts are included in the swarm control phase.

Although the main contribution of this paper is the clay-based interaction model for supporting a number of novel affordances for human-swarm interactions, this model only becomes meaningful in the context of actual algorithms. The image recognition phase (described in Section 4.3.2), consists of a segmentation-inspired algorithm that classifies each observed clay shape in the image to a shape class in the precomputed Shapes Library (SL). The swarm control phase (described in Section 4.3.3) maps the output from the image recognition process onto executable control laws. Consequently, we will develop scalable, distributed, and decentralized control laws that guarantee that the swarm converges to the specified shape. This set of control laws forms a precomputed Control Laws Library (CLL). The SL and CLL are created offline and between their elements there is a one-to-one correspondence, meaning that for each shape in the SL there exists a control law in the CLL.

4.3.2 Image Recognition Phase

In this section we describe the image recognition algorithm needed to recognize the shape formed by the user with the clay. The algorithm is divided into two main parts–the offline part described in Sections 4.3.2.1 and 4.3.2.2 and the online part described in Section 4.3.2.3. The objective of the offline part is to extract the features of the boundary of a shape for each class and store these features in the SL. The objective of the online part is to recognize the shape in an image of the clay based on the features stored in the SL.

4.3.2.1 Shape Alignment

For each of the m classes of shapes in the SL, we want to extract the features of the boundary of a shape (which defines that class of shapes) from a sample of images called the training set. Before we are able to extract these features, we need to apply an alignment algorithm. This algorithm is required to remove variations in position, orientation, and scale of the shape (i.e., a shape's pose) in the different images of the training set before feature extraction. The provided raw training set consists of binary (i.e., monochrome) images of a shape from a particular class in different poses. First, coarse alignment is performed by hand, and then the variational approach proposed in [80] is applied to achieve a finer alignment.

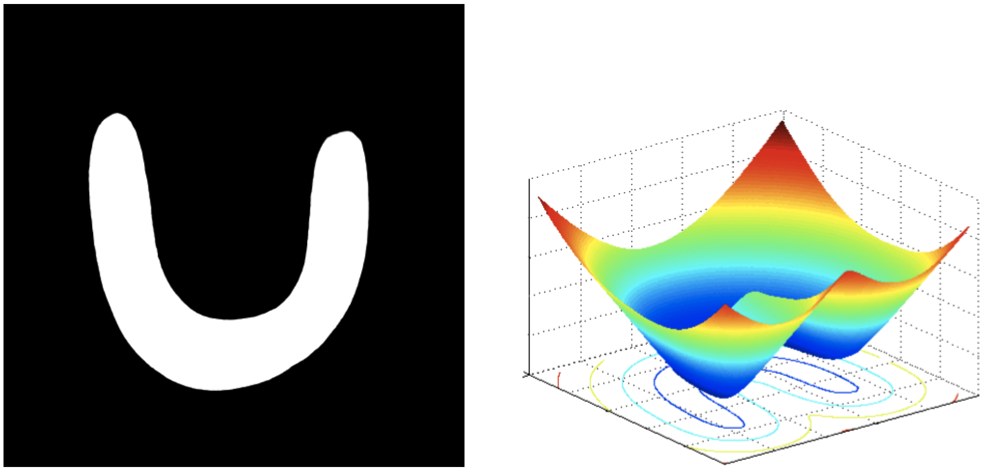
4.3.2.2 Parameterizing the Shape Boundaries

The features of the boundary of a shape from the k-th class can be defined by the parameterized model,

$$\Phi^k[\mathbf{w}^k] = \bar{\Phi}^k + \sum_{i=1}^{q^k} w_i^k \Phi_i^k,$$

where Φ^k is a level set function parameterized by $\mathbf{w}^k = \{w_1^k, w_2^k, \ldots, w_{q^k}^k\}$, which corresponds to the weights of q^k eigenshapes extracted through an eigenvalue decomposition of n signed distance functions and $\bar{\Phi}^k$ is the mean level set of the same n signed distance functions–described in detail in [69, 80]. Specifically, the zero level set of $\Phi^k[\mathbf{w}^k]$ describes the boundaries of the shape associated with the k-th shape class. Those boundaries are directly linked to the variability in the boundary of the shape captured by the q^k eigenshapes, meaning that by varying $\mathbf{w}^k$ it will be possible to try to match a clay shape formed by the user.

Returning to the U-shape class, we applied the level set methods described in [69, 80] to find $\Phi^{\text{U-shape}}$ illustrated in Figure 35b. Figure 35a is its zero level set. We store $\Phi^{\text{U-shape}}$ in the SL and can now use it in the online part of the image recognition framework as way to recognize if the shape of a molded piece of clay belongs to the class of U-shapes.



(a) Zero level-set.

(b) 3D visualization.

Figure 35: The zero level-set of the shape is stored in the Shape Library (SL).

4.3.2.3 Region-based Model For Segmentation and Recognition

The objective of the online part is to segment an image of molded clay and recognize its shape using all parameterized models, $\Phi[\mathbf{w}]$, stored in the SL. We use the Binary Mean Model (BMM) proposed by Yezzi et al. in [84] to compute the parameters of all $\Phi^k[\mathbf{w}^k]$ and then select the one that best segments the image. This is achieved by minimizing a cost function based on the squared difference of the ratios of the total pixel intensity to area between the inner and outer regions. The inner and outer regions are determined by the zero level set of a $\Phi^k[\mathbf{w}^k]$ applied to the image. This is illustrated for the U-shape class in Figure 36.

The algorithm described above fits in our process since our objective is to recognize the shape class of a shape molded by the user with the clay. Since different users likely do not share the same manual skills when deforming the clay, all clay shapes will likely

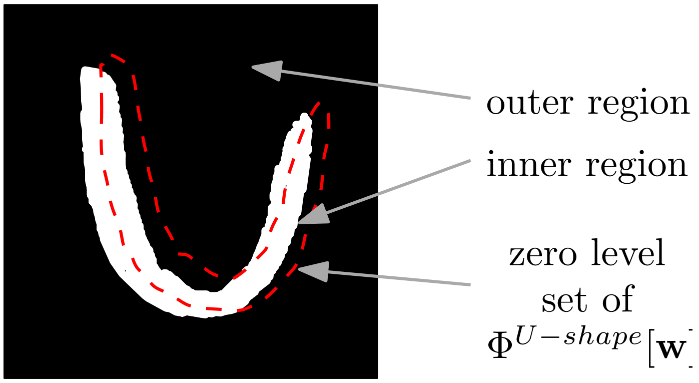


Figure 36: A binary image of the clay with the zero level set of the current $\Phi^{\text{U-shape}}$ superimposed to demonstrate the inner and outer regions used to compute the cost in the BMM.

be nonregular and nonuniform. Moreover, the segmentation has no assumptions on what shape class of the SL (i.e., which $\Phi^k[\mathbf{w}^k]$) should be used to segment the observed shape; therefore, our algorithm segments each newly presented shape using each one of the models available in the SL in parallel.

Within each shape class k, the segmentation process needs time to converge to its final $\Phi^k[\mathbf{w}^k]$. Since our goal is to perform *online* shape recognition, we terminate the computation after a few updates of the weight parameters $\mathbf{w}^k$ for $k = 1, 2, \ldots, m$. Subsequently, we evaluate the BMM cost functional that produced the last $\Phi^k[\mathbf{w}^k]$, for $k = 1, \ldots, m$. The smallest value across the cost functionals will be selected and, thus, determine to which shape class the shape belongs.

For example, suppose that the user molds a U-like shape and that the precomputed SL consists of U-shape and Line-shape classes. The developed algorithm will try to segment the observed U-like shape using both classes. Since the models in the SL capture the characteristics of a particular class of shapes, when the segmentation starts using $\Phi^{U-shape}$, it will result in greater decrease in the BMM cost functional than using $\Phi^{Line-shape}$ for the segmentation. The proposed algorithm handles pose differences in the observed images by matching the latter with the corresponding parameterized shape models after the recognition phase.

4.3.3 Swarm Control Phase

The image recognition phase allows us to match the shape of the deformable medium to a shape in the Shape Library. In this section, we construct the local, decentralized, and distributed control laws that ensure that the robots move into a formation that matches the selected shape. We assume that robots in the swarm do not share a global reference system, do not have a unique ID, and act asynchronously.

4.3.3.1 Robot and Swarm Dynamics

A swarm of N mobile robots is organized over a static, connected Δ-disk graph, the communication graph, $\mathcal{G}_{\text{comm}}$. We use standard, energy-based functions [69], which allow robots to achieve distance-based formations, similar to the formation control introduced in Chapter 3. First, the single integrator dynamics in $\mathbf{R}^2$ of an individual robot is

$$\dot{p}_i(t) = u_i(t), \qquad i = 1,\, 2,\, ,\ldots,\, N \tag{132}$$

where $p_i = [x_i,\, y_i]^T$ is the position of robot i and $u_i = [u_{ix},\, u_{iy}]^T$ is its input. For each edge j incident to robot i in $\mathcal{G}_{\text{comm}}$, we define a nonnegative potential function, $\xi_{ij}(p_i(t), p_j)(t)$, such that the following properties are true:

- ξ_{ij} is convex with a unique minimum,
- ξ_{ij} is monotonically increasing near $d_{ij} = \Delta$.

The edge-tension energy function can be defined as

$$\xi(p(t)) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\xi_{ij}(p_i(t),\, p_j(t)) \tag{133}$$

which is the summation of the local potential functions,

$$\xi_{ij}(p_i(t),\, p_j(t)) = \begin{cases} \frac{1}{2}\{e_{ij}(d_{ij}(t), \theta_{ij}(t))\}^2 & (v_i,\, v_j) \in E_{sens} \\ 0 & (v_i,\, v_j) \notin E_{sens} \end{cases}$$

where $e_{ij} : \mathbb{R}^+ \to \mathbb{R}$ is a strictly increasing function, d_{ij} is the distance between robots i and j, and θ_{ij} is the orientation between robots i and j. Let Δ_{ij} represent the desired distance between robots i and j and Θ_{ij} the desired orientation between robots i and j, then

$e_{ij}(\Delta_{ij}, \Theta_{ij}) = 0$. Each robot minimizes the edge-tension energy in Equation 133 through gradient descent, i.e., each robot's control input is

$$u_i = -\sum_{j \in N_s(i)} \frac{\partial \xi_{ij}(p_i(t),\, p_j(t))^T}{\partial p_i}, \qquad i = 1,\, 2,\, ,\ldots,\, N \tag{134}$$

where $N_s(i) = \{j \mid (i,j) \in E_{sens}\} \subseteq V \smallsetminus \{i\}$ is the neighborhood set of robot i.

A more convenient (for analysis) dynamical system has $\hat{p}$ as its state, where $\hat{p} = (D_{K_N} \otimes I)\, p$ is a stacked vector of all relative positions between robots, D_{K_N} is the oriented incidence matrix of the complete communication graph with N vertices, K_N represents an arbitrary orientation of a complete graph, $\otimes$ denotes the Kronecker matrix product, I is the identity matrix of appropriate dimension, and p is a stacked vector of all robot positions. Consequently, the system dynamics can be written as

$$\dot{\hat{p}} = (D_{K_N} \otimes I_2)u, \tag{135}$$

where u is the stack vector of all inputs defined in Equation 134. Since G_{comm} remains connected for all time, the communication links are fixed and time invariant.

4.3.3.2 Parameterized Edge-Tension Energy Functions

In this section, we demonstrate how different parameterized edge-tension energy functions can be used to achieve deformations in the swarm's formation. If the robots are initially clustered (but not coincident), then we can prove that it is possible to from the shapes in the Shape Library (SL) through a combination of these parameterized functions. Consequently, we construct the Control Laws Library (CCL) by creating control laws for each shape in the SL.

Stretching One example of e_{ij} that allows the swarm formation to be stretched in directions parallel and perpendicular to some desired orientation is

$$e_{ij}^{st}(d_{ij}, \theta_{ij}) = d_{ij}^2 - \big(\cos(\theta_{ij} - \theta_r)\delta_{\|} + \sin(\theta_{ij} - \theta_r)\delta_{\perp}\big)^2, \tag{136}$$

where the parameters $\delta_{\|}$ and $\delta_{\perp}$ represent the stretching factors with respect to the reference orientation θ_r. If $\delta_{\|} = \delta_{\perp}$, then the swarm formation is stretched uniformly into a circular shape, while $\delta_{\|} \neq \delta_{\perp}$ stretches the swarm into an ellipsoid.

Theorem 4.3.1. *Consider a system of N mobile robots with single integrators, each controlled by the input defined in Equation 134, with a local potential function as defined in Equation 136. Then the swarm of robots approaches a formation that minimizes all robots' energy potentials.*

Proof. Consider the function $\xi(p(t))$ defined in Equation 133 that is differentiable and continuous everywhere[3]. Since the communication graph is connected, its diameter cannot be larger than $(N-1)$. Consequently, the largest physical distance between any two robots is smaller than $(N-1)\Delta_{comm}$, which allows us to bound the sum of the interrobot distances such that

$$\sum_{(i,j)\in V\times V} d_{ij} \leq \frac{N(N-1)^2 R_{comm}}{2}. \tag{137}$$

Consequently, all robot trajectories, i.e. $\hat{p}(t)$, evolve in a closed and bounded set. Similarly, the level sets of ξ are compact sets. Since any path connecting two robots i and j on the communication graph has a length of at most $(N-1)$, if we let $\alpha > 0$, then from the properties of ξ it is true that each $d_{ij} \leq \xi_{ij}^{-1}(\alpha(N-1))$ is bounded. Consequently, the set

$$\Omega = \left\{ \hat{p} \;\middle|\; \|\hat{p}\| \leq \frac{N(N-1)^2\Delta_{comm}}{2} \right\} \tag{138}$$

is compact.

The derivative of the edge-tensions energy function, ξ, is negative semidefinite:

$$\begin{aligned}
\dot{\xi} &= \frac{1}{2}\sum_{i=1}^{N}\dot{\xi}_i \\
&= \frac{1}{2}2\sum_{i=1}^{N}\dot{p}_i^T \sum_{j\in N_s(i)} \partial_{ij}\xi_{ij} \\
&= \sum_{i=1}^{N}(\partial_{r_i}\xi_i)^T\dot{p}_i \\
&= -\sum_{i=1}^{N}(\partial_{r_i}\xi_i)^T(\partial_{r_i}\xi_i) \leq 0,\ \forall i.
\end{aligned} \tag{139}$$

This derivative is zero whenever

$$\sum_{j\in N_s(i)} e_{st,ij}(d_{ij},\theta_{ij})\,(p_i - p_j) = 0,\ \forall i = 1,\,\ldots,\,N, \tag{140}$$

[3] Due to discrete variations in the neighboring sets, some discontinuities could arise in the (local) control laws.

which holds if either $p_i = p_j$ or Equation 136 is equal to zero for each pair (i, j). Consequently, the set $\gamma = \left\{ p_i \;\middle|\; \sum_{i=1}^{N} \|\partial_{p_i} \xi_i\|^2 = 0 \right\} \subset \Omega$ is invariant, and we can now prove convergence by applying LaSalle's invariance principle. By this principle, if the initial conditions are in Ω, then the solutions of the system converge to the largest invariant subset $\left\{ p_i \;\middle|\; [x_i, y_i]^T \in \text{span}\,\{\gamma\} \right\} \subseteq \gamma$. Consequently, $\dot{d}_{ij} = 0$ and the system dynamics are $\dot{p} = -(B_{K_N} \otimes I_2)[\ldots \partial_{p_{ij}} \xi_{ij} \ldots]^\top$, such that $[\dot{x}_i,\, \dot{y}_i]^T$ belongs to the range of the oriented incidence matrix D of G_{comm}. For a connected, complete communication graph, range$\,\{D_{K_N}\} = \text{span}\,\{\gamma\}^\perp$; therefore, $[x_i,\, y_i]^T \in \text{span}\,\{\gamma\}$ implies $[\dot{x}_i, \dot{y}_i]^T \in \text{span}\,\{\gamma\}$. This implication holds if and only if $[\dot{x}_i,\, \dot{y}_i]^T \in \text{span}\,\{\gamma\} \cap \text{span}\,\{\gamma\}^\perp \equiv \{0\}$. Consequently, the edge-tension energy function in Equation 133 is zero at steady state, which implies that all ξ_{ij} are locally minimized. If ξ_{ij} is (locally) convex within the communication range, then the extremum is unique and the robots are stabilized to their desired positions and orientations. □

It is important to note that the existence and uniqueness of the solutions of Equation 135 is provided by the boundedness of Equation 134. If one of the two assumptions (connectedness of G_{comm} and initial positions not all coincident, but clustered closely), are violated, then it is possible for $d_{ij} \notin \Omega$ and, consequently, stability is not guaranteed.

Bending A number of different curved formations, such as U-shapes, S-shapes, and L-shapes, are possible by applying a slightly different e_{ij}. For example,

$$e_{ij}^{bend} = \|r_{ij}\|^2 - \big(\cos(\theta_{ij} - \theta_r - \lambda)\delta_\parallel + \sin(\theta_{ij} - \theta_r - \lambda)\delta_\perp\big)^2, \tag{141}$$

where $\lambda = \pm f(\|d_{i,0}\|)\frac{k}{\pi}$. $k > 0$ is a constant, and $f : R \rightarrow \{1, 2, \ldots, N\}$ is a function of $\|d_{i,0}\|$, which is the distance of robot i from its initial position. If $0 \leq \|d_{i,0}\| < \beta_1$, then $f(\|d_{i,0}\|) = 1$, if $\beta_1 \leq \|d_{i,0}\| < \beta_2$, then $f(\|d_{i,0}) = 2$, and so on. The β parameters simply split up some bounded subset of $\mathbf{R}^+ \cup \{0\}$. A proof of convergence follows the same arguments as above and, consequently, can be safely omitted here and for subsequent edge-tension energy functions.

Splitting Our deformable medium also support splitting, which we can support through another edge-tension energy function. Suppose there exists a sensing graph, $\mathcal{G}_{\text{sens}}$. The maximum communication and sensing ranges are ordered, such that $\Delta_{\text{comm}} \gg \Delta_{\text{sens}} > 0$. This ordering implies that $\mathcal{G}_{\text{sens}}$ is dynamic, while $\mathcal{G}_{\text{comm}}$ is static and remains connected at all times. A splitting maneuver implies the swarm separates, which means that G_{sens} is no longer connected. Nevertheless, convergence is still possible, so long as, G_{comm} remains connected. A local edge-tension energy function that can achieve splitting is, for example,

$$\xi_{ij}^{split} = \begin{cases} \pm\frac{1}{2}p_i^2\, k\,\text{sign}\{\cos(\theta_r - \frac{\pi}{2}) \;\; \sin(\theta_r - \frac{\pi}{2})\}, & \|d_{i,loc}\| \leq \beta_1 \\ \xi_{ij}^0, & \text{otherwise,} \end{cases}$$

where ξ_{ij}^0 is a standard edge-tension energy function that simply maintains some specific interrobot distance, and $k > 0$ is a design parameter. The swarm splits in the direction perpendicular to θ_r.

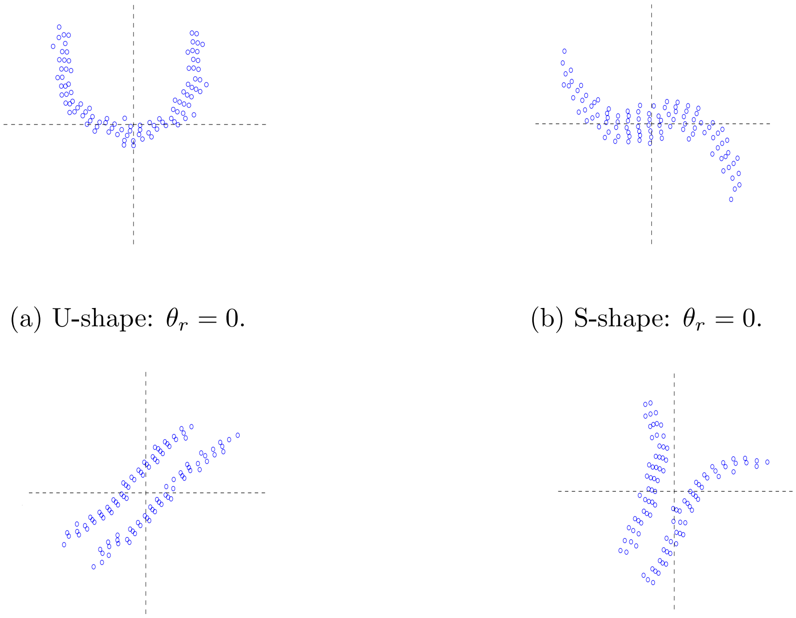
Merging Our deformable medium can be merged back together after a split; therefore, we would like do the same with the swarm. Merging of the swarm can be achieved by forcing all robots to locally agree on their (local) initial positions with an egocentric edge-tension energy, for example,

$$\xi_i^{merge} = \frac{1}{2}\|d_{i,0}\|^2,$$

which is minimized once the robots have returned to their initial conditions, i.e., the initial close clustering of the swarm.

4.3.4 Numerical Simulations and Robot Experiments

We created a number of numerical simulations to demonstrate the efficacy of the control laws. Four different formations are illustrated in Figure 37. These simulations initialized all robots to be clustered uniformly around the origin. The control laws were augmented with a collision avoidance controller to ensure that both simulations and real experiments would avoid collision between robots, while still converging to the target formation. The simulations in Figure 37 included $N = 100$ robots in a 10×10 area with a communication range of $\Delta_{\text{comm}} = 6$. The parameters to the bending and stretching local edge-tension energy functions were $\delta_{\|} = 8$ and $\delta_{\perp} = 0.5$.



(a) U-shape: $\theta_r = 0$.

(b) S-shape: $\theta_r = 0$.

(c) Split and stretch shape: $\theta_r = \frac{\pi}{4}$.

(d) Split and bend shape: $\theta_r = \frac{\pi}{3}$.

Figure 37: Several different swarm formations can be achieved with the different local, edge-tension energy functions.

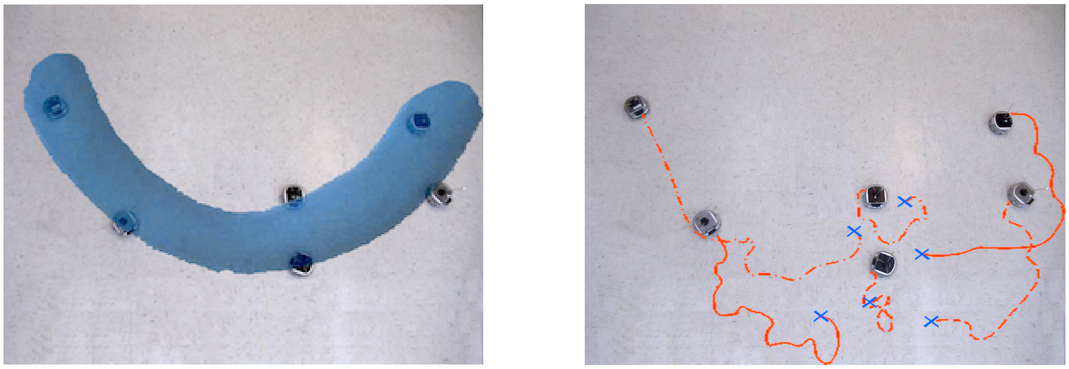
In addition to the numerical simulations, we create experiments with actual mobile robots to demonstrate the efficacy of the entire framework. We used six Khepera III mobile robots, as shown in Figure 38a, for our swarm of robots. An overhead camera, as shown in Figure 38b, was used to capture an image of the shape of the deformable medium. Once a user has formed new shape, the image recognition algorithm computes the appropriate match in the SL and sends the corresponding control law from the CLL to the robots. This process is completed in approximately three seconds on a modern PC (Intel Core 2 Duo @ 2GHz, 4GB 1066MHz RAM). In Figure 39a, a user has formed a U-shape, which as been overlaid on top of the final configuration of the swarm. The trajectories of the robots (captured using a motion capture system) from their initial positions marked by × to their final position are illustrated in Figure 39b. The parameters in this experiment for

(a) Khepera III.

(b) Experimental set up.

Figure 38: The experiments were performed with Khepera III mobile robots and a standard webcam attached to a PC.

the stretching and bending local edge-tension energy functions were $\delta_{\|} = 0.9$, $\delta_{\perp} = 0.13$, and $\Delta_{\text{comm}} = 1$. The robots converge to the U-shape provided by the user in 20-30 seconds. Larger swarm sizes typically require more time to converge.



(a) U-shape is superimposed on the swarm.

(b) Trajectories of the swarm.

Figure 39: The swarm of mobile robots form a U-shape similar to the U-shape formed by the user with the clay.

It is important to note that we were are not actually reproducing the exact shape of the deformable medium. In fact, the shape molded by the user is matched to a template in the SL, which is then approximated by a parameterized control law. The templates in the SL and the matching control laws in the CLL are computed offline, before the user interacts with the swarm. Consequently, the swarm converges, for example, to a U-like shape determined by the parameterization. Figure 40 illustrates several different U-like shapes that can be achieved by the swarm depending on different parameterizations. Consequently, if it is desirable to better approximate a U-like shape molded by the user, then more templates and matching parameterized control laws needs to be added to the shape and control law libraries.

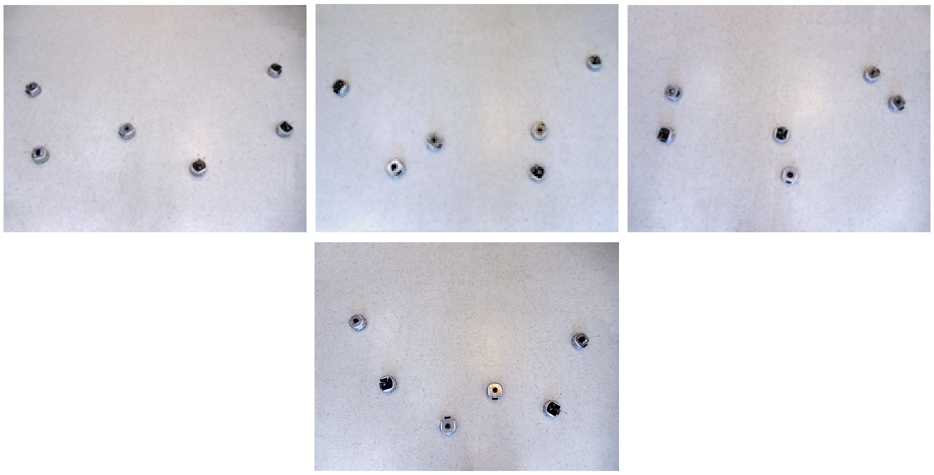


Figure 40: The swarm of robots can form different U-like shapes depending on the selection of parameters.

4.3.5 Conclusions

We have demonstrated that a framework composed of image recognition algorithms and decentralized control laws can be used to provide users with a deformable medium, such as clay, to form geometric shapes with a swarm of mobile robots. The strength of this framework is that a deformable medium, in contrast to joysticks, afford the actions required to form different geometric shapes. For example, a user can bend and stretch a piece of clay in the same way that swarm would bend and stretch into a geometric shape. Consequently, we have demonstrated that by answering the question, *What type of input controllers affords guiding robotic swarms into specific shapes?*, we have designed and developed a better HSI abstraction.

CHAPTER V

CONCLUSIONS AND FUTURE DIRECTIONS

5.1 Conclusions

In this dissertation, we have developed a framework for analyzing, characterizing, and designing human-swarm interaction (HSI) abstractions that are amenable to guiding swarms of mobile robots in geometric tasks. Each technical contribution is a step towards a better understanding of HSI abstractions from a control and graph theory perspective. The main technical contributions of this dissertation are:

- *Control structures*: We provided a formal definition of the control structures underlying HSI abstractions. Consequently, we were able to use control Lyapunov functions (CLFs) to prove that there exist user input signals that can guide a swarm of mobile robots under a control structure to some specification set.

- *Attention, effort, and scalability*: These three metrics characterize a user's interaction with a swarm of mobile robots through some control structure. Optimal control tools allowed us to approximate a trained user's interaction as opposed to capturing these metrics through a user study. Consequently, we were able to compare the attention, effort, and scalability of three different control structures–single-leader networks, broadcast control, and concurrent control– and make a decision on which control structure is best for achieving rendezvous at some desired location with a swarm of mobile robots.

- *User studies*: We have demonstrated how user studies can be used to capture subjective metrics. Specifically, we have shown that these metrics are not only useful in generating qualitative comparisons, but are even more useful when correlating these metrics to properties of the multi-agent system, such as controllability, node centrality, and network centralization. Consequently, we were able to characterize the effect of the interaction topology on the difficulty of guiding a single-leader network into a geometric shape.

- *New HSI abstractions*: Several new HSI abstractions were introduced throughout this dissertation. Specifically, broadcast control was introduced, which focuses on achieving a geometric task with a swarm of mobile robots by interacting with all robots collectively through a broadcast input signal. We demonstrated that broadcast control scaled well in a task geared towards rendezvous, but would require modifications to improve its scalability when separating a heterogeneous swarm of mobile robots into homogeneous clusters.
- *Group size selection*: We have answered the question, *How many robots are required for the user to be able to successfully complete a task?*, in the context of parameterized predator-prey interactions. Experiments with mobile robots as surrogates for lions and gazelles demonstrated that we can generate a formation of a minimum number of predators to successfully capture a particular prey.
- *Deformable media as input controllers*: We investigated the question, *What type of input controllers affords guiding robotic swarms into specific shapes?*, and were able to demonstrate that a deformable medium, such as clay, affords the actions required to complete the task posed in the question. Specifically, users were able to form a variety of different geometric shapes by molding a piece of clay, which was parsed by an image recognition algorithm and matched to an appropriate distributed control law for the mobile robots.

Consequently, these technical contributions comprise a framework that can be applied to future novel HSI abstractions to prove feasibility, assess attention and effort, approximate scalability, design interaction topologies, select swarm size, choose appropriate input controllers, and in general, ensure and understand why a new abstraction is amenable to users interacting with a swarm of mobile robots effectively and easily. For further reference, all technical contributions discussed in this dissertation were published in peer-reviewed journals, conference proceedings, and book chapters [27, 28, 30, 31, 32, 34].

5.2 Future Directions

The framework presented in this dissertation serves as a basis for characterizing and facilitating human interactions with swarms of mobile robots. In its current state it is capable of incorporating a variety of HSI control structures focused on allowing users to solve geometric tasks. These HSI control structures can be characterized, compared, and improved within this framework. The strength of this framework is that it can be naturally extended with further work.

For example, one immediate extension could be applied to our definition of tasks. We have chosen to exemplify tasks with geometric tasks; however, one could also consider dynamic tasks. Such tasks could be an extension of geometric tasks, in the sense that a dynamic task could include an initial configuration and specification set, as well as, state constraints for the trajectories of the robots in the swarm.

Another extension, for example, could be applied to the definition of a HSI control structure. These control structures are currently defined as if the output of the system, i.e., the swarm of mobile robots, is equal to the entire state of the system. However, there may be HSIs, where all states of the robots are not directly observable. This extension would allow this framework to consider HSI control structures that incorporate observers.

Such extensions would allow this framework to incorporate a larger variety of HSI control structures that may be focused on solving different types of tasks. Consequently, this framework can and will continue to serve as a strong basis to characterize, compare, and improve HSI control structures for the purpose of allowing users to interact with swarm of mobile robots effectively.

REFERENCES

[1] Azuma, S.-i., Yoshimura, R., and Sugie, T., "Broadcast Control of Multi-Agent Systems," *Automatica*, 2013.

[2] Bagnold, R. A., "The physics of wind blown sand and desert dunes," *Methuen, London*, vol. 265, 1941.

[3] Balch, T., "The Impact of Diversity on Performance in Multi-Robot Foraging," pp. 92–99, 1999.

[4] Bashyal, S. and Venayagamoorthy, G. K., "Human swarm interaction for radiation source search and localization," in *Swarm Intelligence Symposium, 2008. SIS 2008. IEEE*, pp. 1–8, IEEE, 2008.

[5] Batalin, M. A. and Sukhatme, G. S., "Spreading Out: A Local Approach to Multi-Robot Coverage," vol. 5, pp. 373–382, Springer, 2002.

[6] Becker, A., Ertel, C., and McLurkin, J., "Crowdsourcing swarm manipulation experiments: A massive online user study with large swarms of simple robots," *arXiv preprint arXiv:1402.3653*, 2014.

[7] Becker, A., Onyuksel, C., and Bretl, T., "Feedback Control of Many Differential-Drive Robots with Uniform Control Inputs," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2256–2262, 2012.

[8] Belkhouche, F., Belkhouche, B., and Rastgoufard, P., "Multi-robot hunting behavior," in *2005 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, vol. 3, pp. 2299–2304, IEEE, 2005.

[9] Böhringer, K.-F., Bhatt, V., Donald, B. R., and Goldberg, K., "Algorithms for Sensorless Manipulation using a Vibrating Surface," *Algorithmica*, vol. 26, no. 3-4, pp. 389–429, 2000.

[10] Bonacich, P., "Factoring and weighting approaches to status scores and clique identification," *Journal of Mathematical Sociology*, vol. 2, no. 1, pp. 113–120, 1972.

[11] Bonacich, P., "Power and centrality: A family of measures," *American Journal of Sociology*, pp. 1170–1182, 1987.

[12] Bonacich, P. and Lloyd, P., "Eigenvector-like measures of centrality for asymmetric relations," *Social Networks*, vol. 23, no. 3, pp. 191–201, 2001.

[13] Borgatti, S. P. and Everett, M. G., "A graph-theoretic perspective on centrality," *Social Networks*, vol. 28, no. 4, pp. 466–484, 2006.

[14] Bretl, T., "Control of Many Agents Using Few Instructions," *Robotics: Science and Systems*, 2007.

[15] Brockett, R. W., "Minimum attention control," in *Proceedings of the 36th IEEE Conference on Decision and Control*, vol. 3, pp. 2628–2632, IEEE, 1997.

[16] Bryson, A. E., *Applied optimal control: optimization, estimation and control.* CRC Press, 1975.

[17] Bullo, F., Cortes, J., and Martinez, S., *Distributed Control of Robotic Networks.* Princeton University Press, 2009.

[18] Burgard, W., Moors, M., Stachniss, C., and Schneider, F. E., "Coordinated Multi-Robot Exploration," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 376–386, 2005.

[19] Cao, Y. and Ren, W., "Containment control with multiple stationary or dynamic leaders under a directed interaction graph," in *Proceedings of the 48th IEEE Conference on Decision and Control held jointly with the 2009 28th Chinese Control Conference*, pp. 3014–3019, IEEE, 2009.

[20] Casper, J. and Murphy, R. R., "Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 33, no. 3, pp. 367–385, 2003.

[21] Chopra, S. and Egerstedt, M., "Heterogeneous multi-robot routing," in *American Control Conference (ACC)*, pp. 5390–5395, IEEE, 2014.

[22] Cooke, W., Warr, S., Huntley, J., and Ball, R., "Particle size segregation in a two-dimensional bed undergoing vertical vibration," *Physical Review E*, vol. 53, no. 3, p. 2812, 1996.

[23] Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C., and others, *Introduction to algorithms*, vol. 2. MIT press Cambridge, 2001.

[24] Cumming, G., Fidler, F., and Vaux, D. L., "Error bars in experimental biology," *Journal of Cell Biology*, vol. 177, pp. 7–11, 2007.

[25] Cummings, M. L., *Designing Decision Support Systems for Revolutionary Command and Control Domains.* PhD thesis, University of Virginia, 2004.

[26] Dangalchev, C., "Residual closeness in networks," *Physica A: Statistical Mechanics and its Applications*, vol. 365, no. 2, pp. 556–564, 2006.

[27] de la Croix, J.-P. and Egerstedt, M., "Controllability Characterizations of Leader-Based Swarm Interactions," in *AAAI Fall Symposium Series*, 2012.

[28] de la Croix, J.-P. and Egerstedt, M., "A Separation Signal for Hetereogeneous Networks," in *Allerton Conference on Communication, Control, and Computing*, 2013.

[29] de la Croix, J.-P. and Egerstedt, M., "A separation signal for heterogeneous networks," in *51st Annual Allerton Conference on Communication, Control, and Computing*, pp. 254–261, IEEE, 2013.

[30] de la Croix, J.-P. and Egerstedt, M., "Group-Size Selection for Parameterized Predator-Prey Models," *International Symposium on Mathematical Theory of Networks and Systems*, 2014.

[31] de la Croix, J.-P. and Egerstedt, M., "A Control Lyapunov Function Approach to Human-Swarm Interactions," *American Control Conference*, 2015. To appear.

[32] de la Croix, J.-P. and Egerstedt, M., "Analyzing Human-Swarm Interactions Using Control Lyapunov Functions and Optimal Control," *Networks and Heterogeneous Media*, 2015. To appear.

[33] Diana, M., de la Croix, J.-P., and Egerstedt, M., "Deformable-medium affordances for interacting with multi-robot systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5252–5257, IEEE, 2013.

[34] Diana, M., de la Croix, J.-P., and Egerstedt, M., "Deformable-Medium Affordances for Interacting with Multi-Robot Systems," 2013.

[35] Ding, X. C., Powers, M., Egerstedt, M., Young, R., and Balch, T., "Pilot Decision Support for Controlling Multiple UAVs," *IEEE Robotics and Automation Magazine*, 2009.

[36] Dorfler, F. and Francis, B., "Geometric analysis of the formation problem for autonomous robots," *IEEE Transactions on Automatic Control*, vol. 55, no. 10, pp. 2379–2384, 2010.

[37] Dyke, S., Spencer Jr, B., Sain, M., and Carlson, J., "Modeling and Control of Magnetorheological Dampers for Seismic Response Reduction," *Smart Materials and Structures*, vol. 5, no. 5, p. 565, 1996.

[38] Egerstedt, M., de la Croix, J.-P., Kawashima, H., and Kingston, P., "Interacting with Networks of Mobile Agents," in *Large-Scale Networks in Engineering and Life Sciences*, Springer, 2014.

[39] Estes, R., *The behavior guide to African mammals: including hoofed mammals, carnivores, primates.* University of California Press, 1992.

[40] Fax, J. and Murray, R., "Information flow and cooperative control of vehicle formations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1465–1476, 2004.

[41] Freeman, L. C., "A set of measures of centrality based on betweenness," *Sociometry*, pp. 35–41, 1977.

[42] Freeman, L. C., "Centrality in social networks conceptual clarification," *Social Networks*, vol. 1, no. 3, pp. 215–239, 1979.

[43] Gao, F. and Cummings, M. L., "Barriers to robust and effective human-agent teamwork," in *2014 AAAI Spring Symposium Series*, 2014.

[44] Gazi, V. and Passino, K., "Stability Analysis of Swarms," *IEEE Transactions on Automatic Control*, vol. 48, no. 4, pp. 692–697, 2003.

[45] Girden, E. R., *ANOVA: Repeated Measures.* Sage Publications, Inc., 1991.

[46] Goodrich, M. A., Pendleton, B., Sujit, P., and Pinto, J., "Toward human interaction with bio-inspired robot teams," in *2011 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 2859–2864, IEEE, 2011.

[47] Gross, R., Magnenat, S., and Mondada, F., "Segregation in swarms of mobile robots based on the brazil nut effect," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4349–4356, IEEE, 2009.

[48] Hägele, M., Schaaf, W., and Helms, E., "Robot assistants at manual workplaces: Effective co-operation and safety aspects," in *Proceedings of the 33rd ISR (International Symposium on Robotics)*, pp. 7–11, 2002.

[49] Haque, M., Rahmani, A., and Egerstedt, M., "Geometric Foraging Strategies in Multi-Agent Systems based on Biological Models," *IEEE Conference on Decision and Control*, pp. 6040–6045, 2010.

[50] Hart, S. and Staveland, L., "Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research," in *Human Mental Workload* (Hancock, A. and Meshkati, N., eds.), North Holland Press, 1988.

[51] Hayes, A. T., "How many robots? group size and efficiency in collective search tasks," in *Distributed Autonomous Robotic Systems 5*, pp. 289–298, Springer, 2002.

[52] Ji, M., Azuma, S., and Egerstedt, M., "Role Assignment in Multi-Agent Coordination," *International Journal of Assistive Robotics and Mechatronics*, vol. 7, no. 1, pp. 32–40, 2006.

[53] Ji, M., Muhammad, A., and Egerstedt, M., "Leader-based Multi-Agent Coordination: Controllability and Optimal Control," pp. 6–pp, 2006.

[54] Kahneman, D., "Attention and effort, 1973," *Englewood Cliffs, N7T˜ Prentice-Hall*, 1973.

[55] Kailath, T., *Linear systems*. Prentice-Hall, Englewood Cliffs, NJ, 1980.

[56] Khalil, H. K. and Grizzle, J., *Nonlinear systems*, vol. 3. Prentice hall Upper Saddle River, 2002.

[57] Kingston, P. and Egerstedt, M., "Distributed-infrastructure multi-robot routing using a helmholtz-hodge decomposition," in *50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, pp. 5281–5286, IEEE, 2011.

[58] Kirk, R. E., *Experimental Design*. Wiley Online Library, 1982.

[59] Kleinberg, J. M., "Authoritative sources in a hyperlinked environment," *Journal of the ACM*, vol. 46, no. 5, pp. 604–632, 1999.

[60] Kolling, A., Sycara, K., Nunnally, S., and Lewis, M., "Human swarm interaction: An experimental study of two types of interaction with foraging swarms," *Journal of Human-Robot Interaction*, vol. 2, no. 2, pp. 103–128, 2013.

[61] Lee, S. G. and Egerstedt, M., "Controlled coverage using time-varying density functions," in *IFAC Workshop on Distributed Estimation and Control in Networked Systems*, vol. 4, pp. 220–226, IFAC, 2013.

[62] Lin, J., Morse, A. S., and Anderson, B. D., "The Multi-Agent Rendezvous Problem," *IEEE Conference on Decision and Control*, vol. 2, pp. 1508–1513, 2003.

[63] Lozano, R., Spong, M., Guerrero, J., and Chopra, N., "Controllability and observability of leader-based multi-agent systems," *47th IEEE Conference on Decision and Control*, pp. 3713–3718, 2009.

[64] McLurkin, J., Smith, J., Frankel, J., Sotkowitz, D., Blau, D., and Schmidt, B., "Speaking swarmish: Human-robot interface design for large swarms of autonomous mobile robots.," in *AAAI Spring Symposium: To Boldly Go Where No Human-Robot Team Has Gone Before*, pp. 72–75, 2006.

[65] Mesbahi, M. and Egerstedt, M., *Graph Theoretic Methods in Multi-Agent Networks.* Princeton University Press, 2010.

[66] Nagi, J., Ngo, H., Giusti, A., Gambardella, L. M., Schmidhuber, J., and Di Caro, G. A., "Incremental learning using partial feedback for gesture-based human-swarm interaction," *IEEE Symposium on Human and Robot Interactive Communication*, pp. 898–905, 2012.

[67] Olfati-Saber, R., Fax, J. A., and Murray, R. M., "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[68] Olfati-Saber, R. and Murray, R. M., "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[69] Osher, S. and Sethian, J. A., "Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations," *Journal of computational physics*, vol. 79, no. 1, pp. 12–49, 1988.

[70] Parker, L. E., "Distributed Algorithms for Multi-Robot Observation of Multiple Moving Targets," *Autonomous Robots*, vol. 12, no. 3, pp. 231–255, 2002.

[71] Podevijn, G., O'Grady, R., and Dorigo, M., "Self-organised feedback in human swarm interaction," in *Proceedings of the Workshop on Robot Feedback in Human-robot interaction: How to make a robot readable for a human interaction partner (Ro-Man)*, 2012.

[72] Pryor, K. and Norris, K., *Dolphin societies: Discoveries and puzzles.* University of California Press, 1998.

[73] Pshenichnyi, B., "Simple Pursuit by Several Objects," *Cybernetics and Systems Analysis*, vol. 12, no. 3, pp. 484–485, 1976.

[74] Rahmani, A., Ji, M., Mesbahi, M., and Egerstedt, M., "Controllability of multi-agent systems from a graph-theoretic perspective," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 162–186, 2009.

[75] Rosato, A., Strandburg, K. J., Prinz, F., and Swendsen, R. H., "Why the brazil nuts are on top: Size segregation of particulate matter by shaking," *Physical Review Letters*, vol. 58, no. 10, pp. 1038–1040, 1987.

[76] Schoof, E., Chapman, A., and Mesbahi, M., "Bearing-compass formation control: A human-swarm interaction perspective," *American Control Conference (ACC)d*, pp. 3881–3886, 2014.

[77] Song, P. and Kumar, V., "A Potential Field based Approach to Multi-Robot Manipulation," *IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1217–1222, 2002.

[78] Sontag, E. D., "Control-lyapunov functions," in *Open problems in mathematical systems and control theory*, pp. 211–216, Springer, 1999.

[79] Tanner, H., "On the controllability of nearest neighbor interconnections," *43rd IEEE Conference on Decision and Control*, vol. 3, pp. 2467–2472, 2004.

[80] Tsai, A., Yezzi Jr, A., Wells, W., Tempany, C., Tucker, D., Fan, A., Grimson, W. E., and Willsky, A., "A shape-based approach to the segmentation of medical imagery using level sets," *IEEE Transactions on Medical Imaging*, vol. 22, no. 2, pp. 137–154, 2003.

[81] Twu, P., Egerstedt, M., and Martini, S., "Controllability of homogeneous single-leader networks," pp. 5869–5874, 2010.

[82] Weitzenfeld, A., Vallesa, A., and Flores, H., "A Biologically-inspired Wolf Pack Multiple Robot Hunting Model," *IEEE 3rd Latin American Robotics Symposium*, pp. 120–127, 2006.

[83] Yamaguchi, H., "A cooperative hunting behavior by mobile-robot troops," *The International Journal of Robotics Research*, vol. 18, no. 9, pp. 931–940, 1999.

[84] Yezzi Jr, A., Tsai, A., and Willsky, A., "A statistical approach to snakes for bimodal and trimodal imagery," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, pp. 898–903, IEEE, 1999.