THE ROLE OF THE DYNAMICS OF RELATIVE MOTION IN INFORMATION PASSING IN NATURAL AND ENGINEERED COLLECTIVE MOTION

DANIEL SWAIN

A DISSERTATION
PRESENTED TO THE FACULTY
OF PRINCETON UNIVERSITY
IN CANDIDACY FOR THE DEGREE
OF DOCTOR OF PHILOSOPHY

RECOMMENDED FOR ACCEPTANCE
BY THE DEPARTMENT OF
MECHANICAL AND AEROSPACE ENGINEERING
ADVISER: NAOMI LEONARD

NOVEMBER 2012

Abstract

The breathtaking motions of natural groups such as bird flocks and fish schools have been a major inspiration for research in the development of control algorithms for multi-robot systems. A main thrust of research in this area is to understand how the interconnections among individual behaviors result in high-performing group behaviors. In this dissertation, we are concerned with how individuals — either natural or engineered — move dynamically relative to one another and how this affects information passing in the group, thereby impacting the performance of the group when completing tasks.

The work presented in this dissertation is inspired by observations of coordinated speed oscillations in schools of banded killifish (*Fundulus diaphanus*). We analyze trajectory data of schools of two and three killifish and show that their speed oscillations produce periodic relative motion that modulates line-of-sight visibility between fish. Using tools from graph theory and dynamical models of consensus decision making, we show that this phenomenon can significantly improve group decision-making performance.

We then turn to engineered systems investigate the ways that relative motion can enhance mobile sensor networks with respect to group connectivity and decision-making performance. We also derive control laws, based on killifish-like coordinated speed oscillations into a rich family of moving formations along circular trajectories. We present design guidelines for these types of formations and algorithms to compute what speed oscillation waveform will produce or approximate a formation with a given shape.

Finally, we present the design of a testbed in which robotic fish interact with a school of live fish in real time. Each robotic fish consists of a wheeled robot and a model fish; the model fish is moved about a shallow tank of water via magnetic coupling to a wheeled robot beneath the tank. A significant part of the design of this testbed is the real-time computer vision tracking of the robotic fish and the school of fish. The technology that we have developed for the robotic fish testbed is relevant to a number of other applications. This testbed provides new opportunities for real-time feedback-controlled behavioral experiments with fish schools.

Acknowledgements

I am deeply grateful to my advisor, Naomi Leonard, for her guidance through this experience. I have been exposed to countless fantastic opportunities thanks to her, and she has always granted me the freedom to pursue the opportunities that I found most compelling. It's been a long journey, and it would not have been possible, nor would it have been nearly as enjoyable, without Naomi's extraordinary wisdom, vision, and patience. There were so many times when a dose of her encouragement and enthusiasm made all the difference in the world. Of course all of the work presented in this dissertation was done in collaboration with her, and has benefited immensely because of her involvement.

I am also deeply indebted to Iain Couzin for introducing me to the killifish data and to the idea of the robotic fish testbed. Without those sparks of inspiration and his willingness to lend me both time and resources, this would be a very different and probably much less interesting dissertation. Iain's energy is infectious, and I am a smarter and more thoughtful person because of all the times he has made me see things from a different perspective. I want to thank Iain for being a reader for this dissertation, and for providing invaluable feedback.

There are many people in the MAE department at Princeton without whom my education and this dissertation would not have been possible. Everyone in the front office has been wonderfully friendly and helpful. Jessica O'Leary and Jill Ray deserve the deepest possible appreciation not only from myself but from every graduate student in the MAE department for all of their hard work behind the scenes that makes our lives livable. The dynamics and controls faculty in our department are unsurpassed in both the depth and the breadth of their knowledge; so thank you to Mike Littman, Phil Holmes, Jeremy Kasdin, and Rob Stengel, who have all made a profound and positive impact on my life. I especially want to thank Clancy Rowley for being a reader for this dissertation and for all that I learned from him through three semesters of being an MAE 433 AI. Mike Littman and David Radcliff were also instrumental in making 433 a great experience.

One thing that makes life as a graduate student significantly more enjoyable is having people to ~~commiserate~~ share the experience with. All of the people who have been graduate students and post-docs in the Leonard lab have been fantastic colleagues, and I've been here long enough that if I tried to name them all I'd surely forget a few, so I'll just try to name a few who were especially significant. Derek Paley was a great office-mate and a huge source of inspiration and wisdom. Kendra Cofield and Tian Shen have also been wonderful friends and office-mates. Ben Nabet, Ioannis Poulakakis, Luca Scardovi, Fumin Zhang, Carlos Caicedo, and Ming Cao are all appreciated for their willingness to share their experiences. I am thankful to Andy Stewart, Darren Pais, George Young, and Paul Reverdy for their contributions to various experimental endeavors as well as their friendship. I'm grateful for having had the privilege of working with some fantastic Princeton undergraduates, including Leslie Lee, Qian Wan, Rebecca Legett, and Aaron Trippe, and one Swarthmore undergraduate, Madeleine Abromowitz. I also want to thank the many members of Iain's lab group that have provided insight and comraderie over the years. I especially want to thank Yael Katz for many great conversations and for inspiring me to become a better programmer.

The trouble with trying to thank your friends is that you might neglect some. At least I know that I would. So I'm not going to mention any names other than the ones that I already have (and there are plenty of other names). To anyone who has been my friend along this journey: Thank you for being there.

I would not be who I am today without my family's presence and their support. My parents raised me to have a responsible work ethic, to follow my dreams, and to never give up. Those are key ingredients for making it through graduate school not once, but twice. To them and all of my close and extended family: I love you.

Finally, I want to (and need to) thank my wife, Dr. Mallory A. Balmer-Swain, D.O. She's been by my side since before I started college. We got married, we moved to Philadelphia, and she started medical school as I started at Princeton all within the span of a few weeks, and it doesn't feel like things have slowed down much since then. Nor have they gotten any less exciting or any less fulfilling. I couldn't have done this without her support, understanding, and encouragement. To my wife: I love you and appreciate you more than I can say.

I am grateful to have received a National Defense Science Engineering Graduate (NDSEG) Fellowship from the Air Force Research Laboratory/Air Force Office of Scientific Research (AFRL/AFOSR). The NDSEG fellowship granted me an even greater degree of freedom to pursue the research ideas that I found interesting.

This dissertation carries T-3249 in the records of the Department of Mechanical and Aerospace Engineering, Princeton University.

For my family.

Contents

Abstract iii

Acknowledgements v

List of Figures xiii

1 Introduction **1**

1.1 Motivation and Research Questions 3

1.2 Background and Related Work 5

1.2.1 Mechanisms of Collective Motion 5

1.2.2 Consensus Dynamical Models and Collective Decision-making 9

1.2.3 Engineering Design of Collective Motion for Mobile Sensor Networks 12

1.2.4 Interactive Robotics in Collective Behavior Research 14

1.3 Outline of Dissertation 15

2 Coordinated Speed Oscillations in *Fundulus Diaphanus* **19**

2.1 Source of the Data 20

2.2 Data Analysis Methods 22

2.2.1 Model of Killifish Speed Oscillations 22

2.2.2 Data Filtering and Velocity Estimation 24

2.2.3 Coordinated Speed Oscillation Detection 26

The Hilbert Transform and Instantaneous Phase 27

Detecting the Speed Oscillations of Single Fish 28
Detecting Coordinated Speed Oscillations 30
2.2.4 Measuring the Degree of Synchrony: Mean Sign Product (MSP) 31
2.2.5 Measuring Common Spatial Configurations 32
2.2.6 Visual Information and Sensitivity 34
2.3 Results 36
2.3.1 Coordinated Oscillation Properties 37
2.3.2 Common Spatial Configurations 43
2.3.3 Relative Bearing Oscillations 45
2.4 Discussion 46

3 Coordinated Relative Motion and Decision-Making Performance 48
3.1 Background 50
3.1.1 Graph Theory 50
3.1.2 Linear Consensus Dynamics Over Graphs 53
3.1.3 Sensing Topologies and Sensing Regions 56
3.2 Group-level Decision-Making Performance With Periodic Relative Motion 60
3.2.1 Rate of Convergence to Consensus Over Periodic and Piecewise Constant Graphs 61
3.2.2 Decision-Making Performance of Killifish Schools 64
3.3 Impact of Relative Motion on Connectivity 70
3.3.1 Effective Sensing Regions 70
3.3.2 Improved Connectivity of Random Graphs 72
3.3.3 Optimal Relative Motion Parameters for Group-level Decision-Making Performance 76

4 Coordinated Speed Oscillations for Engineered Systems 83

4.1 Background 85
    4.1.1 Steered Particle Model 85
    4.1.2 Phase Potentials and Phase Arrangements 88
4.2 Trajectories of Steered Particles with Periodic Speed Profiles 91
4.3 Coordinated Control of Steered Particles with Periodic Speed Profiles 105
4.4 Designing Coordinated Relative Motion 116
    4.4.1 Geometry of Coordinated Motion Formations 117
    4.4.2 Design of Motion Parameters to Obtain Specific Formations . 123

5 Hybrid Biological and Engineering Collective Motion Testbed 141

5.1 Testbed Description 143
    5.1.1 Hardware 143
        The Experimental Arena 143
        Wheeled Robots 145
        Model Fish 147
        Tracking System Hardware 148
    5.1.2 Software 149
5.2 Real-Time Tracking 152
    5.2.1 Segmentation 153
        Image Processing 154
        Blobbing 155
    5.2.2 State Estimation 165
    5.2.3 Estimating the Properties of Fish Schools 170
5.3 Experimental Demonstrations 174
    5.3.1 School Centroid Chasing 174
    5.3.2 Triggered Dart Toward School 177
5.4 Extensions 178
    5.4.1 Other Testbeds and Tracking Applications 179

        off-line Fish Tracking 179
        Dancer Tracking 180
        MiaBot Pro Testbed 181
        Beluga Underwater Vehicle Testbed 183
    5.4.2 The MADTraC C++ Library 189
    5.4.3 Inter-process Communication and Web Interfaces 192

6 Conclusion and Future Work **198**
  6.1 Summary of Contributions 199
  6.2 Ongoing and Future Work 204
    6.2.1 Natural Systems 204
    6.2.2 Engineered Systems 206

A Relationship Between MSP and Phase Separation (Proof of Thm. 2.1) **209**

B Visual Cues and Their Sensitivities **212**
  B.1 Angular velocity 214
  B.2 Loom 215
  B.3 Time-to-collision 218

Bibliography **220**

List of Figures

2.1 Example speed profiles of killifish. 20

2.2 Illustration of the definition of relative bearing (2.7). 33

2.3 Relative position oscillation of a two-fish school. 39

2.4 Relative position oscillation of a three-fish school. 40

2.5 Relative position oscillation of a three-fish school (another example). 41

2.6 Histogram of relative bearing pairs showing the most common configurations of three-fish killifish schools. 44

3.1 A simple graph and its Laplacian matrix L. 53

3.2 Example template sensing regions. 59

3.3 Switching killifish sensing topology. 66

3.4 A sampling of topology pairs and their rate of convergence to consensus as a function of duty cycle. 69

3.5 The effective sensing region for two agents undergoing periodic relative motion resulting from sinusoidal speed profiles. 73

3.6 Simulation results for $N = 100$ agents distributed uniformly randomly over a unit square showing the probability that the resulting sensing topology is connected. 75

3.7 Diagram of the ordered spatial configuration used in Sec. 3.3.3. 77

3.8 The two non-null graphs encountered in the ordered setting described in Sec. 3.3.3 and Fig. 3.7. 79

3.9 Convergence rate, σ, as a function of the oscillation frequency, Ω. . . 81

4.1 Illustration of the steered particle model. 86
4.2 Unbounded spiral trajectory. 95
4.3 Illustration of the trajectory decomposition (4.18) for purely sinusoidal speed (see Lemma 4.4). 99
4.4 Trajectories of steered particles with sinusoidal speed profiles as described by (4.13). 106
4.5 Trajectory construction for a periodic speed profile with $v_k(t)$ as described by (4.21). 107
4.6 Trajectory of a steered particle with the periodic speed profile described in Fig. 4.5. 108
4.7 Swarm-like formation with no heading or speed phase control. 119
4.8 A formation with distributed trajectory offsets. 120
4.9 A pair of formations of $N = 4$ agents with similar motion parameters. 121
4.10 Formation of $N = 4$ agents with the periodic speed profile shown in Fig. 4.5a. 122
4.11 Range of values for which Design Problem 4.1 has a solution. 127
4.12 Formations designed by solving Design Problem 4.1. 129
4.13 Example formation found by solving Design Problem 4.2. 137
4.14 Five snapshots of four agents moving around the formation found by solving Design Problem 4.2 with the formation input shown in Fig. 4.13. 138
4.15 Another example formation found by solving Design Problem 4.2. . . 139
4.16 Five snapshots of four agents moving around the formation found by solving Design Problem 4.2 with the formation input shown in Fig. 4.15. 140

5.1 Diagram of the robotic fish testbed components. 144
5.2 A MiaBot Pro wheeled robot with a pair of cylindrical rare-earth magnets press-fit into holes on a custom-made hood. 146
5.3 Model golden shiner and koi. 148
5.4 Diagram of the segmentation and state estimation steps of tracking. . 153
5.5 Background subtraction and thresholding applied to an image of a fish. 155
5.6 Head orientation estimation and correction using skewness of the distribution of pixels. 161
5.7 Illustration of the EMMG algorithm. 164
5.8 A video frame captured from the robotic fish testbed's tracking and control software, showing a robotic predator fish chasing the centroid of a live school of golden shiners. 173
5.9 A sequence of video frames of a robotic predator fish approaching a school of live golden shiners, with overlayed visualization. 175
5.10 Snapshot of ten golden shiners that were tracked off-line from high-definition overhead video. 180
5.11 Snapshot of output from the tracking of dancers performing human flocking experiments. 181
5.12 Snapshot of a coordinated control experiment using the MiaBot Pro robots. 182
5.13 The Beluga testbed and one of the Beluga vehicles. 184
5.14 Diagram of the Beluga testbed system. 185
5.15 Overview of the real-time tracking algorithm used for the Beluga testbed. 187
5.16 Snapshot of the real-time tracking in the Beluga testbed. 190
5.17 Diagram of the two-tiered server framework used on the Beluga testbed. 194

5.18 User interfaces built into the web server. 196

Chapter 1

Introduction

As technology advances, robots become simultaneously less expensive and more capable. Society therefore progressively looks to the use of robotics to solve a wide range of problems, ranging from health care to factory work to planetary exploration. Furthermore, it is becoming increasingly feasible to use *groups* of robots to achieve these tasks. Because robots must necessarily be designed, constructed, and programmed on an *individual* basis, it has become more and more important to find ways to prescribe individual behaviors for robots that, when deployed as a group, can accomplish their mission as a group.

While robotics researchers have made significant progress in this field in recent years, there is a huge space of problems that remain unsolved and applications still lacking deployable solutions. On the other hand, we can readily observe that natural groups are capable of performing group-level tasks: from slime molds to wolf packs to human society, groups of living organisms seem to be innately capable of turning relatively simple individual behaviors into breathtaking group-level feats. This drives us, as designers of collective motion in engineered systems, to seek inspiration from biological collective motion. However, there is still much that is not well understood about the mechanisms of biological collective motion. Tools and expertise from engineering and the physical sciences have recently proven useful in furthering the pursuit of basic understanding in this field.

Understanding information flow within both biological and engineered groups is highly important because (arguably) no group of individual agents can achieve any task without passing some form of information among themselves. This information passing can be explicit — for example, a team of football players planning their next play from the huddle — but, in many cases, information passing is implicit. That is, each member of the group governs their behavior based on the observed states of other group members and the environment rather than explicitly exchanged messages. This ability of mobile animal groups to quickly and effectively pass information throughout the group is especially apparent when witnessing the dramatic maneuvering of a murmuration of starlings and the split-second direction changes of a school of fish.

Both implicit and explicit information passing are fundamentally linked to the geometry of the group, via relative positions and the spatial configuration of individual agents. Distance plays an important role in the quality and availability of information: the power density of radio signals and sound intensity both obey inverse-square laws, the ability of the eye to distinguish points decays with the inverse of the distance from the eye, and the density of a diffusing olfactant decays exponentially with the square root of distance. Directionality also plays a major role in information passing: directional antennas, binaural auditory systems, and forward-facing vision all depend on directionality. Many of these information passing mechanisms require line-of-sight visibility for information to pass from one agent to another, and when there are more than two individuals within a group there is a possibility for one individual to block line-of-sight visibility between the other two. Thus, the spatial configuration of agents can have a dramatic influence on the ability of information to readily pass through even very small groups.

Because geometry plays such a critical role in information flow within a group (and therefore the group's ability to complete tasks), the dynamic movement of individuals relative to one another must play a critical role in information passing. In this dissertation, we consider that role in detail. We consider both biological and engineered groups; specifically, schools of fish and groups of mobile robots. Our goals are to

- further the fundamental understanding of the mechanisms of collective motion and decision-making in biological groups,
- contribute to the set of mathematical and technological tools that are available to study both biological and engineered groups, and
- introduce design methodologies that enhance the capabilities of engineered groups.

1.1 Motivation and Research Questions

One of our primary motivations to consider relative motion is the observation that schools of banded killifish (*Fundulus diaphanus*) engage in coordinated speed oscillations (see Ch. 2). We observe from previously-recorded overhead video that killifish strongly modulate their speed and that these oscillations are clearly phase-coordinated within groups of fish. We seek to extract precise and meaningful quantitative observations from those videos. That is, we want to know *how often do killifish engage in coordinated speed oscillations, under what conditions does this happen, and what are the average properties of these oscillations?* Answering these questions is the main theme of Ch. 2 of this dissertation.

One reason that killifish coordinated speed oscillations are remarkable to us is that they hint at a deeper role that relative motion could play in the functioning of biological groups. The vast majority of previous research that considers the role of spatial structure in collective motion is focused on attempting to explain evolutionary benefits in terms of average group geometry such as average spacing and configuration (see Sec. 1.2.1 for more detailed background information). Killifish coordinated speed oscillations highlight the fact that mobile animal groups are *not* statically configured; in all but the simplest cases, animals in mobile groups are constantly moving relative to each other. This relative motion affects sensory information, and as individuals move through the group they may encounter and be influenced by different neighbors. We are therefore motivated to explore the potential benefits of relative motion for mobile animal groups as well as engineered groups in terms of its effect on information passing. In other words, we ask *why do killifish engage in coordinated speed oscillations, how does relative motion affect information passing in mobile groups in general, and can we apply these results to improve engineered systems?* Ch. 3 is concerned with addressing these questions.

Killifish coordinated speed oscillations also motivate us to design and therefore model the motion of multiple mobile agents that are engaged in coordinated speed modulations. We are particularly motivated to discover what kinds of multi-robotic formations these speed modulations make possible, and what applications might benefit from these kinds of formations. Our model extends previous work on cooperative control of formations, where individual agents are modeled as steered particles [104]. Our earliest simulations of this model made it clear that coordinated speed oscillations lead to a rich new set of multi-agent formations. In those simulations, we prescribed the speed oscillations and observed the resulting patterns. We therefore seek a method to formalize these results and to address the corresponding design problem; that is, *how can we characterize the trajectories of individual agents undergoing periodic speed modulations, can we design control laws to stabilize such agents* *into cohesive formations, and how can we design the speed modulation of a group of N agents to achieve a desired formation?* We address these questions in Ch. 4.

Finally, we are motivated to design experiments to obtain data that would further illuminate our understanding of the role of relative motion in animal groups. Specifically, we seek to design a system that will allow us introduce a stimulus to a school of live fish. Because we are interested in relative motion, we require an experimental platform that is able to *responsively interact* with the school. We are also motivated by recent experiments in which live fish interacted with a robotic fish that was driven along a pre-planned path [126, 41] (see Sec. 1.2.4 for further background information). Therefore, we asked *can an experimental testbed be developed in which robotic fish interact in real-time with a school of live fish, and in what other settings can this technology be applied?* The design of such a testbed is presented in Ch. 5, along with some additional applications of the underlying technology.

1.2 Background and Related Work

The following is a review of the literature as it relates to the work presented in this dissertation. We touch on research in the fields of biological collective motion, engineering of cooperative robotics, theoretical analysis of group decision-making models, real-time computer vision, and ethological robotics. Because we are drawing from the literature in such diverse fields and because many of these fields have a long and rich history, this review is inevitably incomplete. We strive to cover the previous works that are most relevant to our own.

1.2.1 Mechanisms of Collective Motion

Many species of animals have evolved to move together in groups. The *reasons* that this happens have been relatively well-studied. Scientists have discovered a range evolutionary benefits of group motion, including energetic advantages due to aero- or hydro-dynamic coupling, increased chance of surviving an encounter with a predator, increased chance of locating food sources while foraging, and others. See, for example, Chapter 2 in the text *Living in Groups* by Krause and Ruxton [63] for an extensive review of these benefits.

The *mechanisms* of collective motion are not yet well-understood. Individual animals are thought to follow relatively simple "rules" and act on potentially unreliable sensory information, yet they demonstrate group-level behavior that is both highly robust and highly effective. Therefore, researchers are compelled to study the individual-level mechanisms that lead to collective motion not only from the perspective of furthering our basic scientific understanding, but also with the hope that these mechanisms can be beneficially applied to man-made systems.

Collective motion occurs because of the way that individual animals process and react to sensory information about other animals and their environment. Spooner [107] published one of the earliest academic studies of fish schooling in 1931, and in it he posed the following as one of three fundamental questions about the phenomenon of schooling: "From the special aspect of sense topology, what are the senses concerned? How refined are these senses?". Schooling behavior in fish is driven by a combination of olfactory information, visual information, and information from the lateral line (a sensory organ distributed along the sides of some fish species that enables the fish to sense fluctuations in the water pressure) [97, 95]. One of the main themes of this dissertation is that relative motion can enhance sensory information and therefore improve the performance of collective motion.

A key component of the study of collective motion is the use of models to analyze hypotheses and to make testable predictions. Most models of collective motion can be categorized as one of two types: Eulerian and Lagrangian. Eulerian, or continuum, models describe a group using a set of partial differential equations that model the spatial and temporal dynamics of the group's density. These models are useful for groups formed by large numbers of densely packed individuals; for example large wildebeest herds [48], bacteria [119], and insect swarms [47].

Lagrangian models are individual-based: the dynamics of each animal are modeled as a function of the states of the other animals. Group-level observations are made by evolving the individual dynamics forward in time and then making bulk measurements of the individual solutions. A large variety of individual-based models of collective motion exists (see, for example, the review by Parrish and Viscido [93]). Most of these models are constructed by composing three basic behaviors:

1. attraction towards individuals that are far away (cohesion)
2. repulsion away from individuals that are very close (collision avoidance)
3. alignment with individuals that are at intermediate distances

Okubo was among the first researchers to create models of the dynamic motion of individual fish in a school. Okubo's models were inspired by Newtonian dynamics and included terms for attraction and alignment, but not repulsion [84]. Aoki produced one of the first computer simulations of collective motion, and included all three behaviors [2].

One reason for the success of individual-based models is that they lend themselves well to computer simulation. With the increase in the availability of computing power, large-scale simulation of collective motion has transitioned from a special interest topic in the computing community [101] to a standard tool for many biologists. Early and influential examples of the use of large-scale simulations include work by Viscido, Parrish, and Grünbaum [124] and Couzin et al. [28]. Today, there are hundreds of collective motion simulation studies in the literature. In the last few years, the adoption of the graphical processing unit (GPU) as a tool for highly-parallel numerical processing has enabled yet another jump in simulation capabilities. Simulations of schools consisting of millions of fish are now possible on a single GPU-enabled desktop computer [40]. While GPU-powered simulations are becoming more widely used, they are not yet commonplace because they require specialized hardware and programming techniques.

In this dissertation, we make use of individual-based models primarily for two reasons. First, individual-based dynamical motion models provide a natural mathematical language with which to discuss both biological modeling and mobile robotics applications. For more background on the engineering applications of collective motion, see Sec. 1.2.3. Second, individual-based models allow us to directly model the interaction of individual fish with one another, which is a key component of our analysis of killifish coordinated speed oscillations.

Studying the mechanisms of biological collective motion is challenging. At the individual level, the animals' sensory, nervous, and motor systems are all involved, and their interactions can complicate the problem of deciphering behavior [30]. Even with good models of individual behavior, it can be a challenge to find meaningful ways to compare one model to another, or to compare models with data. Generally, this involves computing some *average* quantity for the group. Average distance between fish was a topic of early research [107, 18, 19]. van Olst and Hunter were among the first to consider the role of angular heading and relative bearing in addition to spacing in fish schools [122].

The average inter-individual geometry of fish schools provides empirical observations that can be compared with predicted "preferred" distances and relative bearings. These comparisons can be used to test evolutionary hypotheses. For example, Breder proposed that fish schools may be spatially organized into regular lattice structures to maximize energetic advantages [20]. Partridge, et al. reported average relative positions in three dimensions for several schooling species of fish and connected these average relative positions with predictions based on visual sensory capabilities [94].

Dill et al. showed that particular values of the relative bearing would optimize the the ability of individual fish to observe changes in the direction and/or speed of their neighbors [37]. Average geometry studies have also contributed to the study of bird flocks (see, for example, [3]). Average quantities have produced useful results, yet they cannot tell the whole story; Parrish and Turchin show, for example, that average relative positions can remain steady in empirical data even as individual fish continuously move relative to one another [92]. In this dissertation, we explicitly consider the role of *relative* motion in animal groups.

1.2.2 Consensus Dynamical Models and Collective Decision-making

In a consensus dynamical model, the group attempts to reach agreement on the value of some quantity. Each agent is modeled as having an internal representation (e.g., an estimate or belief) of the value of a quantity of interest to the group; this value is called the agent's *decision variable*. Over time, each agent dynamically updates their decision variable as a function of the values of the decision variables of some subset of the other agents in the group. The agents that influence the k^{th} agent are called the *neighbors* of the k^{th} agent. The group is said to reach *consensus* when all of the decision variables converge to a common value. See Ch. 3 for a mathematical formulation.

The family of consensus dynamical models is large and has been used in a variety of settings. Indeed, Bertsekas and Tsitsiklis introduced the notion of an "agreement algorithm" in the 1980's as a way to abstract and analyze distributed computational algorithms in general. In that work, the decision variable represented the result of some calculation on a single node of a large computer network, and a variety of algorithms were considered [121, 120, 9].

One of the benefits of using consensus dynamical models is their flexibility to model a variety of settings: linear, nonlinear, time-invariant, time-varying, continuous-time, discrete-time, etc. In this dissertation, we make use of dynamical consensus models in two settings. First, we use a linear consensus dynamical model to study the effects of relative motion in a school of killifish. Second, we use a nonlinear consensus dynamical model to derive control laws for the motion of robotic agents. In both cases, the decision variable represents the direction of motion of each individual in the group.

Consensus dynamical models have been used in many places throughout the literature to model group-level decision-making about the group's direction of motion. Reynolds [101] introduced an algorithm in 1987 that simulated the motion of a flock of birds by having each bird update its own velocity at each time step in response to the average of the velocities of its neighbors. Vicsek et al. [123] used an algorithm similar to Reynolds' to investigate the spontaneous synchronization of self-propelled particles, where each agent dynamically updates the direction of its velocity by averaging the direction of all neighbors within a given distance. Couzin et al. [28] used simulation studies to investigate the emergence of leadership in mobile animal groups, showing that the group could correctly make a decision about group direction when only a small portion of the group was informed about the correct choice, the informed were not identified, and no explicit communication was modeled.

The use of linear consensus dynamical models is appealing because we can rely on the large existing toolset for linear dynamical systems. We use a linear consensus dynamical model to study the role of relative motion for information passing in collective motion. We do this by studying how each agent's neighbors change as a function of time and how this affects group-level performance.

The network describing which agents are neighbors of which other agents is called the *communication topology* or *sensing topology*. The primary theoretical tool that we use to model the sensing topology is a *graph*. Each agent (e.g., animal or robot) is modeled as a node on the graph and there is an edge from the k^{th} node to the j^{th} node if the k^{th} agent is influenced by the j^{th} agent.

In robotic systems, it is straightforward to determine the sensing topology because the communication is either explicit or, if it is implicit, the conditions of communication are due to programmed behavior. In natural systems, however, the degree of influence is not well-understood. One of the simplest models is the one employed by Vicsek et al. [123], in which each individual is influenced by all individuals within a particular distance. Various modifications to this type of model have been employed, for example weighting by distance [31]. The "zone model" used by Couzin et al. [29] is a special case that is similar to the models used by Reynolds [101]. Paley et al. [91, 87] used zone models for analysis of biological collective motion based on ideas from cooperative control. Recent research by Ballerini et al. [6] suggests that interaction in starling flocks may be governed by so-called "topological distance", i.e., each bird has a fixed number of neighbors regardless of distance. A recent simulation study by Kunz and Hemelrijk [65] models interaction based on line-of-sight visibility, which is a notion that we use when considering visual information in Ch. 3.

Theoretical analysis of consensus dynamical models on graphs is well-studied. In the control theory literature, Jadbabaie et al. [53] used algebraic graph theory to analyze the linearization of the model of Vicsek et al. [123] and proved that synchronization occurs if certain assumptions are met. Other important early theoretical studies of linear consensus dynamical systems include [86, 12, 85]. Moreau [80] contributed important work towards generalizing the analysis of consensus dynamical systems by using tools from the analysis of time-varying nonlinear dynamical systems. Graph theory and consensus dynamical models have become popular and useful tools for both modeling and designing collective motion.

Linear models of consensus in direction produce only local results because the angle of direction evolves on the circle, which we denote here as S^1, not on the real line. Global models of direction consensus are therefore fundamentally nonlinear and evolve on a non-Euclidean state space. Consensus on the direction of motion of N agents evolves over the N-torus, which we denote $T^N = S^1 \times \cdots \times S^1$ (N times). Consensus on T^N is equivalent to the synchronization of N coupled oscillators. Kuramoto [66] first studied the synchronization of oscillators in 1984 in the context of "chemical oscillations" (see also the review by Strogatz [110]). Scardovi et al. [103] provided a key result by proving global results for nonlinear consensus dynamics on T^N. Nabet et al. [82] and Leonard et al. [71] performed a theoretical analysis of the simulation studies of Couzin et al. ([28]) where the model was treated with tools from nonlinear dynamical systems theory.

1.2.3 Engineering Design of Collective Motion for Mobile Sensor Networks

One of the principal motivators for the study of collective motion in the engineering field is its application to the design of mobile sensor networks. A motivating example for us is the work by Leonard et al., in which a group of autonomous underwater vehicles was deployed in Monterey Bay, California, to collect measurements that were used to improve oceanographic models [69, 44, 70]. Those experiments were part of the Autonomous Ocean Sampling Network (AOSN) project [33, 32] and the Adaptive Sampling and Prediction (ASAP) project [70].

In the AOSN and ASAP networks and in many other sensor networks, the individual sensor units are mobile robots with on-board sensor suites. It is typical for underwater and aerial mobile robots to have strict constraints on vehicle dynamics: bounded speed, minimum turning radius, nonholonomic constraints, etc. One of the key challenges in designing mobile sensor networks is therefore creating motion control laws that account for these types of constraints but also achieve the design goal.

Various authors have considered the problem of designing motion control laws for mobile sensor networks. One approach is to design control laws to adjust the spatial distribution of the sensors so that they optimally cover some scalar field [27, 67] or minimize estimation error [73, 35]. Recent work by Caicedo-Nùñez and Leonard [24] addresses both of these objectives by distributing robots to minimize the objective analysis (OA) mapping error (see [21]).

Another approach to mobile sensor network motion design is to create moving formations or patterns of sensors. Adaptive formations can be used to enable a network to climb a gradient in a noisy sampled field as shown in [83] and as demonstrated in the AOSN field experiments [44]. Formations and patterns have also been used to track level curves [74, 8, 128]. Patterns were used in the ASAP field experiment by Leonard et al. [70, 69] to minimize OA mapping error. Other proposed uses of sensor network motion pattern design include hurricane sampling [36] and surveillance [61]. Zhang et al. have investigated the use of sensor network formations for cooperative filtering, and show that using moving patterns of vehicles can reduce the total number of vehicles that are required for effective filtering [130, 131].

In this dissertation, we consider formations that result from periodic speed modulations. We extend previous results by Sepulchre, Paley, and Leonard, who developed coordinated control laws for circular formations of constant-speed self-propelled particles (see, for example, [104, 91]). That work was itself inspired by the work of Justh and Krishnaprasad on formations of self-propelled particles in the context of unmanned aerial vehicles [56, 55] and by the work of Kuramoto [66] and Strogatz [110] on coupled oscillator dynamics. Related work includes coordination along general planar closed curves [129], coordination in the presence of flow fields [89], coordination with limited vehicle-to-vehicle communication [105], coordination on the sphere [90], and coordination in three dimensions [102].

1.2.4 Interactive Robotics in Collective Behavior Research

Advances in robotics technology are enabling new methodologies in the study of social animal behavior. A recent survey by Krause et al. [62] outlines these methodologies and their consequences (see also the earlier prospective by Balch et al. [4]). Robots that are used in behavioral research are typically programmed to *interact* with the experimental specimens, usually by having the robot pose as either a conspecific or heterospecific organism. Michelsen et al. [78] studied honey bee communication dances by manipulating a mechanical model of a bee. Ward et al. used a remotely controlled model fish to investigate quorum decision-making [126]. Faria et al. designed a computer-controlled model fish that was able to move around the arena in pre-planned routes so that they could study mechanisms of recruitment and leadership in fish schools [41]. Halloy et al. used robotic cockroaches that were able to responsively interact with live cockroaches to study decision-making in the mixed robot-animal cockroach group [49]. In Ch. 5, we describe the design and development of a testbed that enables a robotic fish to interact *responsively* with a school of live fish. That is, the robotic fish is able to adjust its behavior in real-time in response to the other live fish in the tank. The design of our system was inspired by [49, 126, 41].

One of the primary challenges in designing our robotic fish testbed was development of the real-time tracking and control software that enables the responsiveness of the robotic fish to a school of live fish. Off-Line tracking of fish has been done successfully in the past; see, for example, [124, 79, 23, 57].

Tracking multiple animals presents a significant challenge because the subjects can be very similar in appearance to one another and because they are often in very close proximity to one another. Both of these effects make it very difficult to solve the *data association* problem: the problem of maintaining the identity of individuals over time. Recent work by Feldmen et al. uses multiple laser range finders to track multiple humans [43]. Butail et al. have recently developed algorithms to track mosquitos off-line from video captured in the wild [22]. The authors of [6] point out the potential benefit in reconstructing the trajectories of individual starlings off-line using high-resolution stereo photography sequences of very large schools of starlings. Straw et al. have used multiple cameras and multiple high-powered workstations to track flies in three dimensions in real-time [109]. The approach of [109] is similar to ours, though we use a single camera to track fish in a planar arena using commodity hardware. The use of multiple cameras by [109] also reduces the difficulty of the data association problem.

Others have succesfully used particle filter methods for off-line tracking of multiple objects[79, 52, 58]. Particle filtering methods generally perform well if the number of sampling particles is large. Increasing the number of sampling particles used in a particle filter incurs additional computational cost. Consequently, we were unable to achieve satisfactory performance in real-time with particle filtering methods.

The method that we use to solve this problem is similar to one used for off-line tracking of flies in a planar arena by Branson et al. [17]. We use one unscented Kalman filter (UKF) to dynamically estimate the state of each fish. The UKF allows us to model the dynamics of the fish and to apply nonlinear constraints to both the dynamical model and the measurement model [54, 125].

1.3 Outline of Dissertation

We begin in Ch. 2 by analyzing the trajectories of schools of two and three killifish. We describe the source of the data in Sec. 2.1 and our data analysis methods in Sec. 2.2. Sec. 2.2.3 describes our Hilbert transform-based method for detecting coordinated speed oscillations. In Sec. 2.3, we present quantitative results on our analysis of killifish coordinated speed oscillations. In particular, we describe the properties of these oscillations (Sec. 2.3.1) and the typical oscillating spatial configurations of the three-fish schools (Sec. 2.3.2). We also compare our results to those of Dill et al. [37] (Sec. 2.3.3), who investigate the relationship between relative position and visual information (see Sec. 2.2.6 for an overview and App. B for details). We provide a brief discussion of our killifish analysis results in Sec. 2.4 in order to provide context for subsequent chapters.

In Ch. 3, we investigate the decision-making performance benefits of killifish relative motion due to coordinated speed oscillations. We provide technical background on graph theory, linear consensus dynamics, and sensing topologies in Sec. 3.1. In Sec. 3.2.1, we provide some theoretical results regarding the rate of convergence to consensus over periodically time-varying graphs. Using the results from Ch. 2, we derive a model of the visual sensing topology of the three-fish killifish school and the way that the topology changes periodically with time due to the motion of the fish relative to one another. We use this model and the theoretical results of Sec. 3.2.1 to show in Sec. 3.2.2 that, under certain assumptions, killifish can experience a performance gain in speed of consensus dynamics as a result of the oscillations. We investigate other benefits of relative motion in Sec. 3.3 with respect to the connectivity of the group's sensing topology. In particular, we show in Sec. 3.3.2 that coordinated relative motion can reduce the density of agents required to maintain connectivity when the sensing radius of each agent is fixed. In Sec. 3.3.3, we compute relative motion parameters that maximize the rate of convergence to consensus. The results in Sec. 3.3.2 and Sec. 3.3.3 use effective sensing regions, a theoretical tool that we develop in Sec. 3.3.1.

In Ch. 4, we explore the uses of coordinated speed oscillations for engineered systems. We employ the theoretical framework developed by the work of Sepulchre et al. [104], which makes use of a steered particle model and notions from the coupled oscillator literature (see Sec. 4.1 for technical background). We describe the trajectory solutions in Sec. 4.2 for our model of steered particles with constant turning rates and time-periodic speed profiles inspired by those we observe in killifish schools. The trajectories can be described in terms of a circular trajectory (i.e., what we would observe for a constant-speed particle) and a component that results from the speed modulation. The component due to speed modulation can be described directly in terms of the Fourier decomposition of the periodic speed profile. In Sec. 4.3, we derive coordinated control laws for circular formations of steered particles with periodic speed profiles. The control laws allow coordination of the trajectory centers (i.e., nominal locations), the headings of the agents, and the phase relationship of their speed oscillations. In Sec. 4.4.1, we catalog some of the formation geometries that this coordination enables. In Sec. 4.4.2, we present methods to determine what speed profile is necessary to obtain a desired formation of steered particles with periodic speed profiles.

We describe the robotic fish testbed in Ch. 5. An overview of the testbed, its components, and its physical design is given in Sec. 5.1. In Sec. 5.2, we describe the tracking algorithms that enable the robotic fish to responsively interact with a live school of fish in real time. That discussion is broken down into two components: segmentation (Sec. 5.2.1), which is the process by which we extract measurements of the positions and orientations of individual fish from an image, and state estimation (Sec. 5.2.2), where we dynamically estimate the trajectories and velocities of fish over time. We also describe methods for estimating properties of the fish school (e.g., centroid, polarization, and boundaries) in Sec. 5.2.3. Descriptions of two experimental demonstrations of the testbed are given in Sec. 5.3: school centroid chasing, in which the robotic fish continuously pursues the fish school (Sec. 5.3.1), and a triggered dart towards the school, in which the robotic fish waits for the fish school to satisfy some pre-determined condition and then accelerates toward the school (Sec. 5.3.2). The technology that we have developed for this testbed is being used in other projects in the lab; we describe some of these extensions in Sec. 5.4.

Ch. 6 is the conclusion of this dissertation, in which we re-address the research questions posed above in Sec. 1.1. We also suggest some directions for future research.

Chapter 2

Coordinated Speed Oscillations in *Fundulus Diaphanus*

In this chapter, we explore coordinated speed oscillations exhibited by *Fundulus diaphanus*, banded killifish (hereafter: killifish). Examples of these speed oscillations are shown in Fig. 2.1.

We argue in Ch. 3 that coordinated speed oscillations among individual killifish can dramatically improve group-level decision-making performance. The focus of this chapter is to establish the presence of coordinated speed oscillations and to describe the nature of these oscillations in terms of frequency, amplitude, and the phase relationship between oscillations of neighboring fish. Further, we analyze the most common spatial configurations of three-fish schools. It is from changes between these configurations due to speed oscillation, that we derive the changing sensing topologies that form the basis of our theoretical analysis of group-level decision-making performance in Ch. 3.

Another potential benefit of coordinated speed oscillations is their effect on the ability of the fish to see one another and to perceive changes in behavior from this visual information. Dill et al. [37] derive a set of relative bearings between fish that

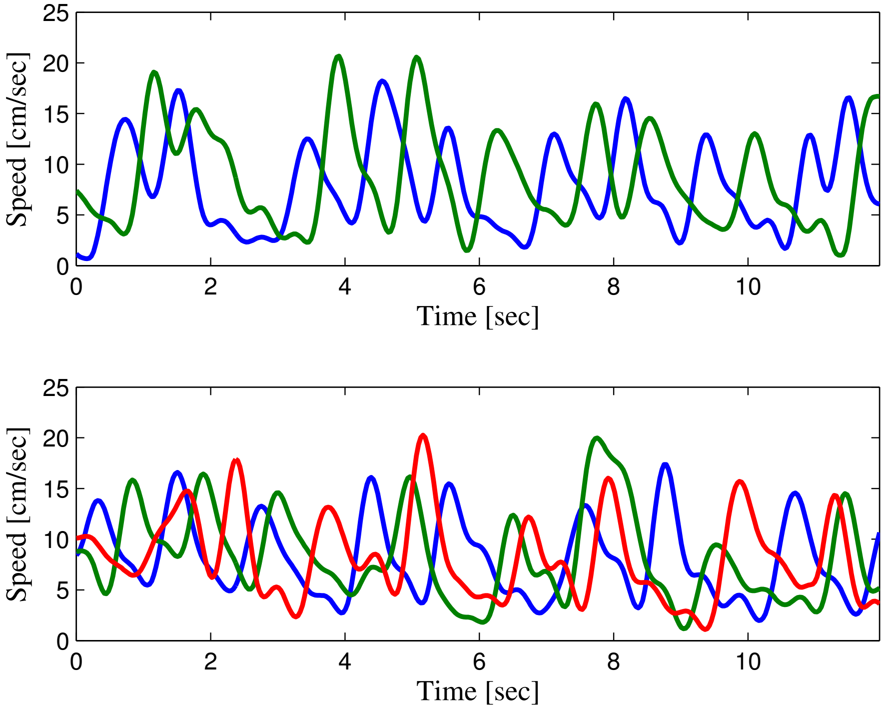


Figure 2.1: Example speed profiles of killifish in two (top) and three (bottom) fish schools.

optimize sensitivities to various visual signals; those results are rederived in App. B. We discuss here how killifish coordinated relative motion allows fish to visit multiple optimal relative bearings predicted by [37] as well as generate optical flow as they move relative to one another.

The majority of the work in this chapter is an alternate presentation of work in preparation for publication [114].

Sec. 2.1 describes the source of the data. Sec. 2.2 describes the methods used to analyze the data. The results of analysis are presented in Sec. 2.3. Sec. 2.4 includes some concluding remarks.

2.1 Source of the Data

The data presented here were produced by analyzing video of schools of two and three killifish. The experiments were conducted and the videos were recorded by Iain

Couzin around September 2000. The experimental arena was a 190 cm square tank filled with water to a depth of a few cm (shallow compared to the length of the fish). The fish used were 4 ± 0.3 cm in length. 15 groups of two fish and 11 groups of three fish were filmed. The experiments were recorded on digital video tape at 31.2 frames per second (FPS). In October 2010, the relevant sections of tape were converted to dv-format video files and stored on a lab computer hard drive[*].

Custom tracking software based on the real-time tracking software developed as part of this dissertation and described in Ch. 5 was used to estimate the trajectories of fish from the video. The camera's field of view did not completely cover the tank and therefore there were time segments during which one or more fish were not visible in the video. These segments were discarded.

A simple scaling factor was used to convert positions of the fish in pixels relative to the bottom left corner of the video frame to cm relative to the intersection of the bottom left corner of the video frame with the plane defined by the bottom of the tank. The camera was mounted in such a way that the image plane was nearly parallel to the bottom of the tank, and the camera lens produced a very small amount of radial distortion. Therefore, a scaling factor provides satisfactory accuracy. The scaling factor, 0.239 cm/pixel, was determined by dividing the known width of the tank in cm by the measured width of the tank in pixels (measured at several locations in the image). No camera calibration data was available other than the dimensions of the tank, therefore a more accurate coordinate conversion was unavailable. The location and orientation of the camera remained fixed across all experiments, thus the scaling factor is consistent for all data.

It should be noted that one artifact of the transfer process from tape to file was time scaling. That is, the frame rate of the video files is incorrectly reported by

---

[*] `poincare.princeton.edu/data3/Videos/Killis`

most software as 25 FPS. The original video was recorded at 31.2 FPS[†] and the correct time in seconds can be determined by dividing the frame index by 31.2 FPS. Correspondingly, the sampling period used below is $T_s = 32.1$ msec.

2.2 Data Analysis Methods

This section describes the methods used for data analysis of killifish speed oscillations. The model that we use for killifish speed oscillations is presented in Sec. 2.2.1, which serves also to establish notation that we will use throughout the remainder of this chapter. Sec. 2.2.2 describes the methods used to filter trajectory data and to obtain velocity (and therefore speed and orientation) estimates. Sec. 2.2.3 describes a Hilbert-transform based algorithm for detecting time segments during which one or more fish are engaged in (potentially coordinated) speed oscillations. Sec. 2.2.4 describes a metric that we use to measure the degree of synchrony between fish in killifish speed oscillations. Sec. 2.2.5 describes the method used to measure common configurations of three-fish schools. Sec. 2.2.6 reviews the results of Dill et al. [37] in the terminology and notation of this dissertation.

2.2.1 Model of Killifish Speed Oscillations

In this section, a model for killifish speed oscillations is presented. This model is the basis for our analysis and for the coordinated oscillation detection algorithm presented in Section 2.2.3.

Let

$$\mathbf{r}_k(t) = \begin{pmatrix} x_k(t) \\ y_k(t) \end{pmatrix}$$

[†] Confirmed by comparing frame numbers to Iain Couzin's notes about the lengths of the original experiments.

be the position of the k^{th} fish at time t. Its velocity is

$$\mathbf{v}_k(t) = \begin{pmatrix} \dot{x}_k(t) \\ \dot{y}_k(t) \end{pmatrix},$$

its speed is

$$s_k(t) = \|\mathbf{v}_k(t)\| = \sqrt{\dot{x}_k^2(t) + \dot{y}_k^2(t)},$$

and we will refer to its orientation with respect to the x axis as $\theta_k(t)$. Let

$$\mathbf{e}_k^{\theta}(t) = \begin{pmatrix} \cos\theta_k(t) \\ \sin\theta_k(t) \end{pmatrix}$$

be the unit vector pointing in the direction of the fish's orientation. We make the simplifying assumption that the fish are swimming in the direction of their orientation, i.e., $\mathbf{v}_k(t) \parallel \mathbf{e}_k^{\theta}(t)$. There was no externally-generated fluid flow inside the experimental arena and it is clear from the videos that this assumption holds true for the vast majority of the data.

For brevity, we may drop the subscript and/or time notation when they are not relevant, e.g., we may write $s(t)$ or simply s to refer generally to a fish's speed when index and time are not important.

The speed of a killifish may be modeled as

$$s(t) = \nu(t) + \mu(t)\cos\phi(t). \tag{2.1}$$

We refer to $\nu(t)$ as the *nominal speed*, $\mu(t)$ as the *speed amplitude*, and $\phi(t)$ as the *speed phase*. Note that (2.1) is a very general model; indeed, we may set $\mu(t) = 0$ and model any speed profile with $\nu(t)$. However, we are primarily concerned with cases for which $\nu(t)$ and $\mu(t)$ are nearly constant and $\cos\phi(t)$ is approximately sinusoidal in t.

For *perfectly* sinusoidal speed, $\nu(t)$ and $\mu(t)$ are constant and the speed phase grows linearly with time as

$$\phi(t) = \Omega t - \bar{\phi}, \tag{2.2}$$

where Ω is the *natural frequency* of speed oscillation and $\bar{\phi}$ is a constant *speed phase offset*.

The *speed variation* is defined as

$$\delta s(t) \triangleq s(t) - \nu(t) \tag{2.3}$$

and is an important quantity below. For perfectly sinusoidal speed, the speed variation fits the model

$$\delta s(t) = \mu \cos\left(\Omega t - \bar{\phi}\right).$$

2.2.2 Data Filtering and Velocity Estimation

In this section, we describe how the trajectory data is filtered to reduce the impact of noise in our analysis and how velocities of the fish are estimated from the filtered data. Our approach uses differencing of the position data and low-pass filtering to estimate the velocity of each fish. The speed and heading are then calculated directly from the velocity vector. We feel that this approach introduces fewer assumptions about the behavior of the fish than may be introduced by using other dynamic estimates (for example, nonlinear Kalman filtering).

A first estimate of the velocity, $\bar{\mathbf{v}}$, is determined by forward differences in position:

$$\bar{\mathbf{v}}[t] = \frac{1}{T_s}\left(\mathbf{r}[t+1] - \mathbf{r}[t]\right),$$

where here we use square bracket notation to indicate discrete time indexing of the time sequence[‡]. The first estimate of the velocity is filtered by convolving it with a truncated sinc kernel. A truncated sinc kernel approximates an ideal low-pass filter in the frequency domain (see, for example, [98] or any introductory signal processing text). The truncated sinc kernel, $K(\tau)$, is the sequence

$$K[\tau] = \bar{K}\frac{\sin 2\pi\tau T_s f_c}{2\pi\tau T_s f_c},\ -\frac{L_K-1}{2} \leq \tau \leq \frac{L_K-1}{2},$$

where f_c is the filter cutoff frequency in hertz, L_K is the length of the kernel and an odd integer, and

$$\bar{K} = \left(\sum_{\tau=-\frac{L_K-1}{2}}^{\frac{L_K-1}{2}} \frac{\sin 2\pi\tau T_s f_c}{2\pi\tau T_s f_c}\right)^{-1}$$

is a constant that normalizes the kernel. The cut-off frequency was empirically chosen to be $f_c = 2.0$ Hz in order to reduce as much noise as possible without over-smoothing the velocity. A kernel length of $L_K = 201$ samples was found to produce good results.

The convolution operation is undefined near the beginning and end of a sequence. To avoid losing samples in these regions, the initial velocity estimate is extended beyond its original boundaries. The resulting extended velocity sequence, $\bar{\mathbf{v}}_e$, is calculated as

$$\bar{\mathbf{v}}_e[t] = \begin{cases} \bar{\mathbf{v}}[-t], & -\frac{L_K-1}{2} \leq t < 0 \\ \bar{\mathbf{v}}[t], & 0 \leq t < N_t \\ \bar{\mathbf{v}}[2N_t - t], & N_t \leq t < N_t + \frac{L_K-1}{2} \end{cases}.$$

That is, the values are mirrored about the beginning and end of the sequence. This process preserves the slow variations in the signal value while having minimal effect on short time-scale behaviors. The final velocity estimate, $\mathbf{v}[t]$, is the convolution of

---

[‡] I.e., $x[t] = x(tT_s)$ for a signal x, time index t, and assuming that the sampling is such that $x[0] = x(0)$.

the extended velocity sequence $\bar{\mathbf{v}}_e$ and the truncated sinc kernel K:

$$\mathbf{v}[t] = \sum_{\tau=-\frac{L_K-1}{2}}^{\frac{L_K-1}{2}} \bar{\mathbf{v}}_e[\tau - t]K[\tau],\ 0 \leq t \leq N_t.$$

The speed is estimated as the magnitude of the velocity estimate:

$$s[t] = \sqrt{v_x^2[t] + v_y^2[t]},$$

where $v_x[t]$ and $v_y[t]$ are the x and y components of $\mathbf{v}[t]$. The direction of motion (equivalently, the orientation of the fish) is the angle that the velocity estimate forms with the $x-$axis:

$$\theta[t] = \tan^{-1}\frac{v_y[t]}{v_x[t]}.$$

In the rare occurrence that $v_y[t] = v_x[t] = 0$, the last valid value of θ is used, i.e., $\theta[t] = \theta[t - t^*]$ where $t^* > 0$ is the smallest value for which either $v_x[t - t^*] \neq 0$ or $v_y[t - t^*] \neq 0$. We also refer to $\theta[t]$ as the heading angle.

The nominal speed, μ, is estimated in exactly the same way as the speed, s, is estimated, except that the filter cutoff frequency is $f_c = 0.375$ Hz and the kernel length is $L_K = 1067$ samples. The low cutoff frequency ensures that only the very slowly time-varying component of the speed is captured.

2.2.3 Coordinated Speed Oscillation Detection

When a fish's speed is perfectly sinusoidal, its nominal speed, ν, and speed amplitude, μ, are both constant and its speed phase grows linearly with a slope Ω as in (2.2). This is never exactly true for real data; the fish modulate their speed *approximately* sinusoidally with a slowly time-varying nominal speed and amplitude, and the modulation changes over time in accordance with the transient nature of the fish's behavior. Therefore, the goal of the speed oscillation detection algorithm is to isolate segments of time during which the fish's speed is approximately sinusoidal.

This section describes a Hilbert transform-based algorithm for detecting both individual and coordinated speed oscillations. The goal of the algorithm is to detect time segments during which the speed phase of a fish fits well to the phase model (2.2). Coordinated speed oscillations are detected by finding segments for which two fish simultaneously fit the phase model with similar frequencies.

We begin by describing the Hilbert transform and how it can be used to estimate the instantaneous phase of a signal. We then describe how this can be used to detect speed oscillations for a single fish. It is straightforward to then extend the single-fish algorithm to detection of coordinated speed oscillations.

The Hilbert Transform and Instantaneous Phase

The Hilbert transform and its application to estimating the instantaneous phase of a signal is described here to motivate the oscillation detection algorithm below. For a more thorough discussion of the estimation of instantaneous phase, see [14].

The Hilbert transform of a signal $u(t)$ is denoted $H(u)(t)$ and is defined as the Cauchy principal value of the integral

$$\frac{1}{\pi}\int_{-\infty}^{\infty}\frac{u(\tau)}{(t-\tau)}d\tau. \tag{2.4}$$

The complex signal

$$u_*(t) \triangleq u(t) + iH(u)(t)$$

is called the *analytic signal* of $u(t)$.

An important property of the Hilbert transform is its effect on pure sinusoids. Namely, if

$$u(t) = A\cos\left(\Omega t - \bar{\phi}\right),$$

where A, Ω, and $\bar{\phi}$ are all constant, then

$$H(u)(t) = A \sin\left(\Omega t - \bar{\phi}\right).$$

That is, the Hilbert transform phase shifts a sinusoid by $\frac{\pi}{2}$ radians. Furthermore, the analytic signal of $u(t)$ is

$$u_*(t) = Ae^{i\left(\Omega t - \bar{\phi}\right)}.$$

If a signal

$$u(t) = A(t) \cos \phi(t)$$

is *approximately* sinusoidal, i.e., if the amplitude, $A(t)$, is sufficiently band limited and the phase, $\phi(t)$, grows approximately linearly with time, then its analytic signal is approximately

$$u_*(t) \approx A(t)e^{i\phi(t)}.$$

It is for this reason that, given a signal $u(t)$, the quantity $|u_*(t)|$ is referred to as the *instantaneous amplitude* of $u(t)$ and the quantity $\angle u_*(t)$ is referred to as the *instantaneous phase* of $u(t)$.

In practice, the integral (2.4) is not calculated directly. Instead, the *discrete Hilbert transform* is used. Matlab implements this via the `hilbert` command, which calculates the analytic signal for a given input.

Detecting the Speed Oscillations of Single Fish

The first step of the algorithm is to isolate time segments of interest. We begin by discarding any time samples for which the fish's nominal speed, ν, is less than 4.0 cm/sec. Any of the remaining time segments that are less than 3.0 sec long are discarded. Next, we calculate the analytic signal δs_* of the speed variation δs (as defined in (2.3)). Estimates of the instantaneous speed phase and amplitude are obtained from the analytic signal, i.e., $\angle \delta s_*[t] \approx \phi[t]$ and $|\delta s_*[t]| \approx \mu[t]$.

A piecewise linear fit is then performed on the instantaneous phase to determine potential oscillatory segments. The piecewise linear fit algorithm has two steps and operates as follows. In the first step, the least squares best-fit line is calculated. The mean of the squared errors between the fit and data (i.e., the variance of the residuals) is compared to a threshold parameter, T_R. If the error is less than T_R, the first step of the piecewise linear fit is finished. Otherwise, the data segment is split into halves (e.g., the interval $[0, N_t)$ becomes $[0, \lfloor \frac{N_t}{2} \rfloor)$ and $[\lfloor \frac{N_t}{2} \rfloor + 1, N_t))$ and new fits are calculated for each of the new segments. This process is iterated for each subsegment until either the resulting subsegments a single time step long or the error threshold is met. The output of the first step of the piecewise linear fit algorithm is a set of endpoints defining segments that are either one sample long or meet the error threshold. In the second step of the piecewise linear fit algorithm, we compare the sum of errors for adjacent segments with the error obtained if the two segments are joined into one and re-fit to a common line. If the new fit produces an error less than T_R, then the segments are joined. Otherwise we move on to the next pair of adjacent segments. This process is repeated until no adjacent segments can be joined without exceeding the error threshold T_R.

The output of the piecewise linear fit algorithm is a sequence of endpoints for segments during which the instantaneous phase fits well to a line, as well as the slopes and intercepts of those lines. Equivalently, the output is a collection of segments during which a fish may be oscillating its speed approximately sinusoidally and the frequency at which this oscillation is occurring. We further filter these results by rejecting any segments that are either shorter than a length threshold, T_L, or for which the average value of the oscillation amplitude is less than 25% of the nominal speed.

Detecting Coordinated Speed Oscillations

The single-fish speed oscillation detection algorithm described above is the basis of the coordinated speed oscillation detection algorithm.

We begin by performing the single-fish algorithm for each fish that is being considered. The union of the endpoints of the oscillatory segments for both fish creates a new set of intervals, for which each interval represents a time segment during which both fish are (potentially) exhibiting speed oscillations. The frequencies of the individual fish speed oscillations (from the individual fits) are then compared to determine if the fish are oscillating at approximately the same frequency. The difference in frequency as a fraction of the average frequency is used as a difference metric, i.e., we compare the frequencies f_1 and f_2 for fish 1 and 2 respectively to a threshold T_f and identify the speed oscillations to be coordinated if

$$\frac{2\,|f_1 - f_2|}{f_1 + f_2} < T_f.$$

Only intervals that meet this frequency similarity condition and are longer than a length threshold, T_L, are kept; the rest are discarded. For each interval that is kept, we compute a common frequency estimate for the two fish by simultaneously fitting the instantaneous phases of both fish. We do this by subtracting from each instantaneous phase sequence its mean value and then fitting both sequences simultaneously to a single line. The slope of the simultaneous line fit is taken as the common frequency.

Note that both the single fish speed oscillation detection algorithm and the coordinated speed oscillation detection algorithm have important parameters. Both algorithms have an error threshold, T_R, and a minimum-length threshold T_L. The coordinated algorithm adds a frequency similarity threshold T_f. We cite the threshold values that were used in the results below. The results of the analysis do depend on the values of these thresholds to varying degrees for each quantity of interest. The values that were used were chosen by observing the results over a range of reasonable parameter values and comparing the fits with data for visual agreement for a random sampling of data sequences.

2.2.4 Measuring the Degree of Synchrony: Mean Sign Product (MSP)

The degree to which two fish's speed oscillations are synchronized (i.e., they have a phase difference of 0 radians) or anti-synchronized (i.e., they have a phase difference of π radians) can be estimated from the sample covariance of the signs of the two speed variations. We call this quantity the *mean sign product* (MSP) of two speed variations and, for fish i and fish j, it is calculated as

$$\mathrm{MSP}\left(\delta s_i, \delta s_j\right) = \frac{1}{N_t} \sum_{t=0}^{N_t-1} \mathrm{sign}\left\{\left(\delta s_i[t] - \delta \bar{s}_i\right)\left(\delta s_j[t] - \delta \bar{s}_j\right)\right\}, \tag{2.5}$$

where $\delta \bar{s}_i$ ($\delta \bar{s}_j$) is the average value of $\delta s_i[t]$ ($\delta s_j[t]$) over the time interval. Compared to other estimates of phase difference (for example, the $y-$intercept from the linear fit of instantaneous frequency), the MSP provides a direct estimate of synchrony and its value is insensitive to amplitude fluctuations.

It is useful to examine the relationship between the phase separation of two sinusoids at the same freqiency and the value of their MSP.

Theorem 2.1. *The MSP for a pair of sinusoids, δs_1 and δs_2, that share a common frequency and have a phase separation of $|\Delta\phi| \leq \pi$ satisfies*

$$\mathrm{MSP}\left(\delta s_1, \delta s_2\right) \approx 1 - 2\frac{|\Delta\phi|}{\pi}, \tag{2.6}$$

A proof of Thm. 2.1 is given in App. A.

Thm. 2.1 is an approximation that applies in a simplified case, but from it we can infer some general relationships between the value of the MSP for two signals and the degree to which they are synchronized or anti-synchronized. For example, we see that the MSP is 1 for two perfectly synchronized sinusoids ($|\Delta\phi| = 0$) and the MSP is -1 for two perfectly anti-synchronized sinusoids ($|\Delta\phi| = \pi$). This result holds for any pair of signals that consistently have the same sign (synchronization) or opposite signs (antisynchronization). Time-varying amplitudes do not affect the value of the MSP as long as the sign of the amplitude is consistent (e.g., $\mu_i(t) = \bar{\mu}_i + \delta\mu_i(t)$ where $\bar{\mu}_i$ is constant and $|\delta\mu_i(t)| < |\bar{\mu}_i|$).

Because of noise, the relationship (2.6) is only an approximation. Noise causes the value of the MSP to be greater than -1 for anti-synchronized signals and less than 1 for synchronized signals. Calculating the mean and standard deviation of the MSP for multiple intervals of equal length gives an indication of the degree of synchronization between speed oscillations. Negative mean MSP values indicate antisynchronization and positive mean MSP values indicate synchronization. Speed oscillations with uniformly distributed random phase separation over the range $0 \leq |\Delta\phi| \leq \pi$ would generate a uniform distribution of MSP values in the range $-1 \leq MSP \leq 1$. The mean of this distribution would be zero and its standard deviation would be 1/3.

2.2.5 Measuring Common Spatial Configurations

The configurations of three-fish schools can be measured in terms of relative bearings from the *central fish*. The central fish at any sample instant t is the one closest to the centroid of the school, which is defined as

$$\bar{\mathbf{r}}[t] = \frac{1}{N}\sum_{k=1}^{N}\mathbf{r}_k[t]$$

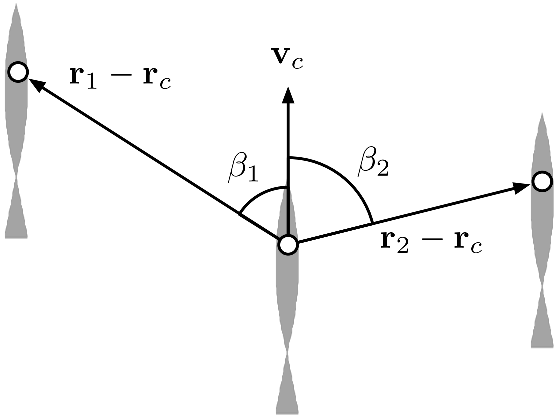


Figure 2.2: Illustration of the definition of relative bearing (2.7). Note that the left-most neighbor of the central fish is labeled with an index of 1 and the right-most neighbor of the central fish is labeled with an index of 2, so that $\beta_1 \geq \beta_2$.

Below, a subscript c will be used to indicate the central fish. That is, if fish k is the central fish, then $\mathbf{r}_c = \mathbf{r}_k$ is its position and $\theta_c = \theta_k$ is its direction of motion and orientation. The *relative bearing* from the central fish to its k^{th} neighbor at sample instant t is denoted $\beta_k[t]$ and is defined as

$$\beta_k[t] = \angle\left\{\mathbf{r}_k[t] - \mathbf{r}_c[t]\right\} - \theta_c[t],\ k = 1,\ 2. \tag{2.7}$$

See Fig. 2.2 for an illustration. Relative bearing is always calculated in the range $-\pi \leq \beta_k[t] \leq \pi$.

The pair of relative bearings of the central fish's neighbors in a three-fish school gives a quantification of the spatial configuration of the school that is insensitive to variations in the distance between fish. To avoid ambiguity with respect to left-right symmetry, the fish are always labeled so that $\beta_1 \geq \beta_2$. That is, the central fish's left-most neighbor is labeled with an index of 1 and its right-most neighbor is labeled with an index of 2.

2.2.6 Visual Information and Sensitivity

Dill et al. [37] predict that a fish that is swimming in parallel with and behind another fish may improve its ability to detect changes in the leading fish's behavior by positioning itself to maximize sensitivity to visual cues. They consider three candidate visual cues: the angular velocity of the image of the leader's eye on the follower's retina, the loom of the leader's eye, and the time-to-collision calculated from the relative loom of the leader's eye. We define these quantities precisely below and (following the derivation in [37]) derive mathematical expressions for them in App. B. The sensitivities and the related optimally-sensitive relative positions are calculated from these expressions.

The relative bearing from the focal fish to the leader fish is considered as a proxy for the angular position of the leader fish on the focal fish's retina. When the fish are thin compared to the separation and are similar in length, the relative bearing is a very good approximation of the retina angle. We label the relative bearing here as β, and define it as

$$\beta \triangleq \tan^{-1}\frac{y}{x}, \tag{2.8}$$

where (x, y) gives the position of the leader fish relative to the focal fish in a coordinate frame whose $x-$axis is aligned with the focal fish's direction of motion. This is the same relative bearing that is illustrated for a three-fish school in Fig. 2.2; the central fish is in this case the focal fish, β_1 is the relative bearing to the fish on the left, and the $x-$axis is aligned with $\mathbf{v}_c$.

The first visual cue that Dill et al. consider is the angular velocity of the retina angle, i.e.,

$$B \triangleq \frac{d}{dt}\beta. \tag{2.9}$$

In Dill et al., the angular velocity is labeled Ω; B is used here to avoid confusion with the speed oscillation natural frequency. The angular velocity represents the rate of change of the retina angle, i.e., how fast the image moves across the eye. The angular velocity, B, is therefore related to optical flow along the image plane. When two fish are swimming in parallel and at the same speed, there is no relative motion and hence the image does not move on the retina. Therefore, when two fish are swimming in parallel and at the same speed, we have $B = 0$.

The second visual cue that Dill et al. consider is the loom, Λ. Loom is defined as the time rate of change of the solid angle, α, subtended by the leader fish's eye. We show in App. B that

$$\alpha = E\frac{\sin(\beta - \theta)}{d^2}, \tag{2.10}$$

where E is the (constant) cross-sectional area of the leader fish's eye, $\theta = \theta_1 - \theta_c$ is the leader fish's direction of motion relative to the focal fish's, and d is the distance between centers of mass of the two fish. The loom is then defined as

$$\Lambda \triangleq \frac{d}{dt}\alpha. \tag{2.11}$$

Loom is a measure of how quickly the size of the image of the leader fish changes and is therefore related to optical flow into or out of the image plane. Similar to the angular velocity, when two fish are swimming in parallel and at the same speed, there is no motion in the image and therefore $\Lambda = 0$.

The third visual cue that Dill et al. consider is time-to-collision, τ. Time-to-collision is the amount of time that it would take the two fish to collide if they were approaching each other at constant velocity (see [118, p. 179] for a mathematical formulation of time-to-collision). Time-to-collision is defined as 2 times the inverse of the relative loom, λ. The relative loom is defined as

$$\lambda = \frac{\Lambda}{\alpha}$$

and the time-to-collision is defined as

$$\tau = \frac{2}{\lambda} = 2\frac{\alpha}{\Lambda}. \tag{2.12}$$

Like loom, time-to-collision is related to optical flow into or out of the image plane. When two fish are swimming in parallel and at the same speed, the time-to-collision is infinite.

Dill et al. consider the sensitivity of B, Λ, and τ to changes in the leader fish's speed s_L and/or heading angle, θ. They fix the offset between fish perpendicular to the direction of motion and calculate values of β that maximize the magnitudes of the partial derivatives $\frac{\partial B}{\partial \theta}$ (sensitivity of angular velocity with respect to heading), $\frac{\partial B}{\partial s_L}$ (sensitivity of angular velocity with respect to speed), $\frac{\partial \Lambda}{\partial \theta}$ (sensitivity of loom with respect to heading), $\frac{\partial \Lambda}{\partial s_L}$ (sensitivity of loom with respect to speed), $\frac{\partial \tau}{\partial \theta}$ (sensitivity of time-to-collision with respect to heading), and $\frac{\partial \tau}{\partial s_L}$ (sensitivity of time-to-collision with respect to speed). The maximizing relative bearings are shown in Table 2.1.

	Relative bearing [degrees] that maximizes sensitivity with respect to	
Signal	Heading	Speed
Angular Velocity	90.0	45.0
Loom	63.4	90.0
Time-to-collision	35.3	35.3

Table 2.1: Values of the relative bearing, β, that maximize sensitivity with respect to changes in the leader's heading (θ) and speed (s_L) as in Table 14.1 of [37].

2.3 Results

This section presents the results of the data analysis described above. Results regarding the existence and properties of coordinated speed oscillations in killifish are presented in Sec. 2.3.1. The common spatial configurations are described in Sec. 2.3.2.

	School Size	% of Time Oscillating	Frequency [Hz]	Amplitude [cm/s]	Nominal [cm/s]	MSP for Pairs
Individuals	2	92.8	0.75 ± 0.24	4.3 ± 1.1	8.0 ± 2.1	N/A
Pairs	2	76.9	0.74 ± 0.16	4.2 ± 0.84	8.1 ± 1.7	-0.15 ± 0.27
Individuals	3	83.3	0.76 ± 0.25	4.1 ± 2.6	8.3 ± 2.7	0.0 ± 0.27

Table 2.2: Results of the speed oscillation analysis for two- and three-fish killifish schools. The analysis of pairs is carried out for time segments during which both fish are engaged in coordinated speed oscillations. The MSP for three-fish schools is calculated from segments during which near neighbors were engaged in coordinated speed oscillations.

The results of Sec. 2.3.1 and Sec. 2.3.2 are discussed in the context of the results of Dill et al. [37] in Sec. 2.3.3.

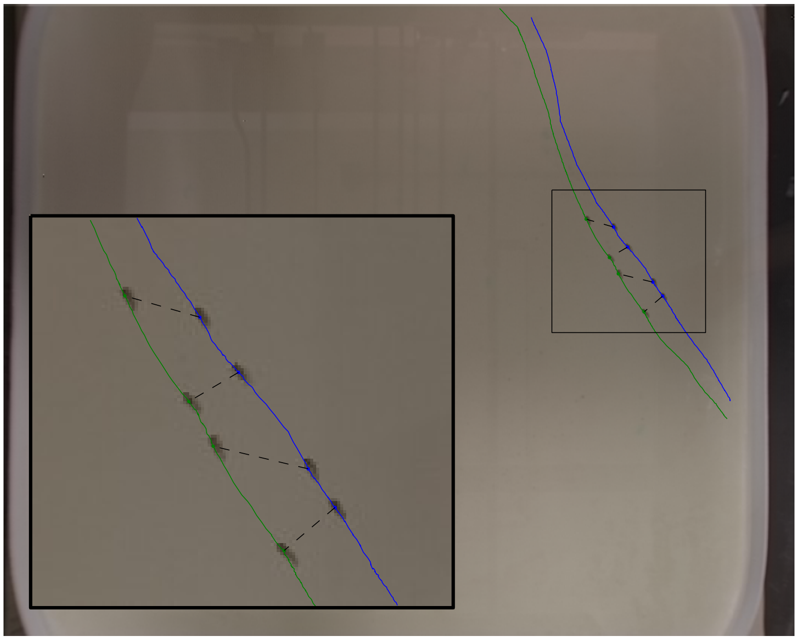
2.3.1 Coordinated Oscillation Properties

The coordinated speed oscillation analysis results that we discuss here are summarized in Table 2.2. These results were produced using the analysis algorithms described in Sec. 2.2. The phase fitting error threshold was $T_R = 1.0$ rad/sec^2, the minimum segment length threshold was $T_L = 2.67$ sec, and the frequency similarity threshold was $T_f = 0.75$.

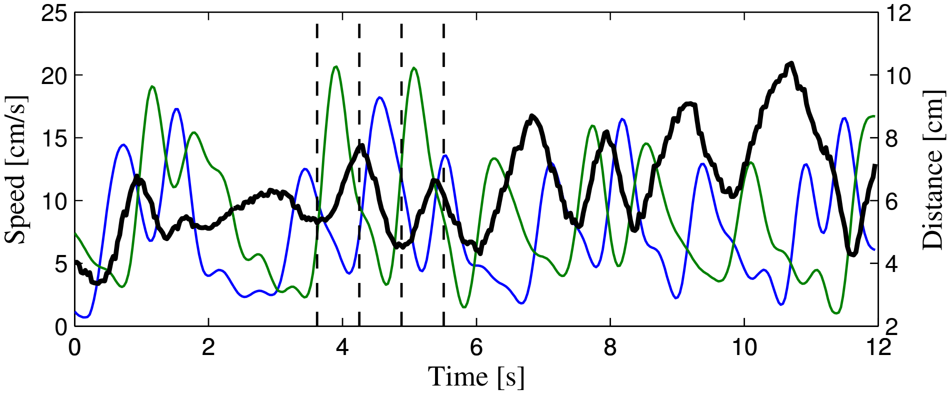
Individual fish in two-fish schools exhibit speed oscillations 92.8% of the time and the average oscillation frequency is 0.75 Hz. Coordinated speed oscillations (i.e., at the same frequency) are observed in two-fish schools 76.9% of the time and the average oscillation frequency is 0.74 Hz. Individual fish in three-fish schools exhibit speed oscillations 83.3% of the time and the average oscillation frequency is 0.76 Hz. The parameters of the oscillations as reported in Table 2.2 are very similar for two- and three-fish schools and regardless of coordination: the average nominal speed is approximately 8.1 cm/s and the speed oscillations occur at a frequency of approximately 0.75 Hz with an amplitude of approximately 4.2 cm/s. The amplitude of oscillations is consistently approximately 50%.

Figs. 2.3, 2.4, and 2.5 illustrate the oscillations in relative position that result from speed oscillations. In each of these figures, the trajectories of the fish are shown in the experimental arena (two fish in Fig. 2.3, three fish in Figs. 2.4 and 2.5) along with overlaid video snapshots at four evenly-spaced time instants. Dashed lines join the positions of near neighbor fish in each figure, and insets show close-ups of the region of the arena where the snapshots occur. The bottom half of each figure is a double $y-$axis plot that shows the speed of the fish (left-hand $y-$axis) and the distance between each pair of fish (right-hand $y-$axis). The distances are plotted in black and the speeds are plotted in colors to match the trajectories (e.g., the fish with the blue trajectory has the blue speed curve). The data segments are the same that were used to produce the example speed plots in Fig. 2.1, and the same colors are used in that figure. For convenience, we will refer below to the fish with the blue trajectory in each figure as the “blue fish”, the fish with green trajectory as the “green fish”, and the fish with the red trajectory as the “red fish”. Figs. 2.4 and 2.5 are created from different time instants in the same data segment. Correspondingly, the red, blue, and green fish Fig. 2.4 are the same as the red, blue, and green fish in Fig. 2.5.

In Fig. 2.3b, we see that the speed oscillations are consistently about 180 degrees apart in phase. That is, when the blue fish is moving fastest, the green fish is moving slowest, and vice versa. The distance between the two fish oscillates because their speeds oscillate and because the speed oscillations are not synchronized. The distance oscillation is apparent in Fig. 2.3b. The effects of the distance oscillation on the spatial configuration of the fish can be seen in Fig. 2.3a. At the time of the first snapshot, the fish are side-by-side and close to one another. By the time of the second snapshot, the green fish has pulled ahead and the fish are farther apart. The blue fish catches up with the green fish by the time of the third snapshot. In the fourth snapshot, we see the cycle beginning to repeat as the green fish pulls forward again.

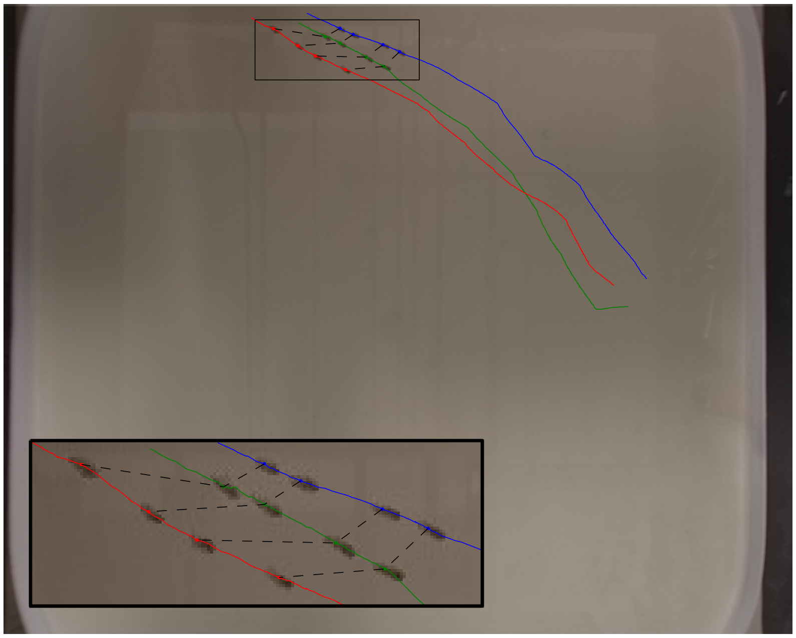


(a)

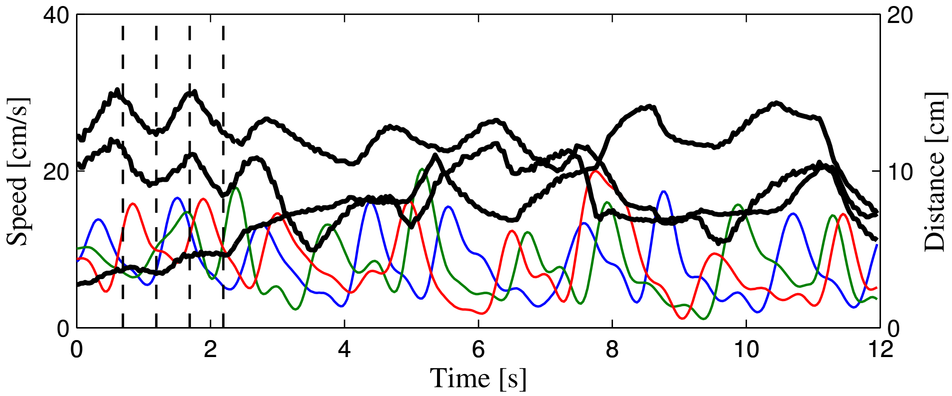


(b)

Figure 2.3: Relative position oscillation of a two-fish school (same data as shown in Fig. 2.1a). (a) The trajectories (blue and green solid curves) of the two fish in the experimental arena. The direction of motion is up and left in the image. Four snapshots of the video are superimposed to show where the fish are at four evenly-spaced time instants. The dashed lines join the positions of the two fish at each time instant. The four times are indicated by the vertical dashed lines in (b). The inset is a close-up of the boxed region. Note that the fish are side-by-side in the first and third snapshots. In the second and fourth snapshots, the fish on the left (green trajectory) is ahead. (b) The black curve shows the distance between the two fish. The blue and green curves show the speeds of the fish with the blue and green trajectories, respectively.

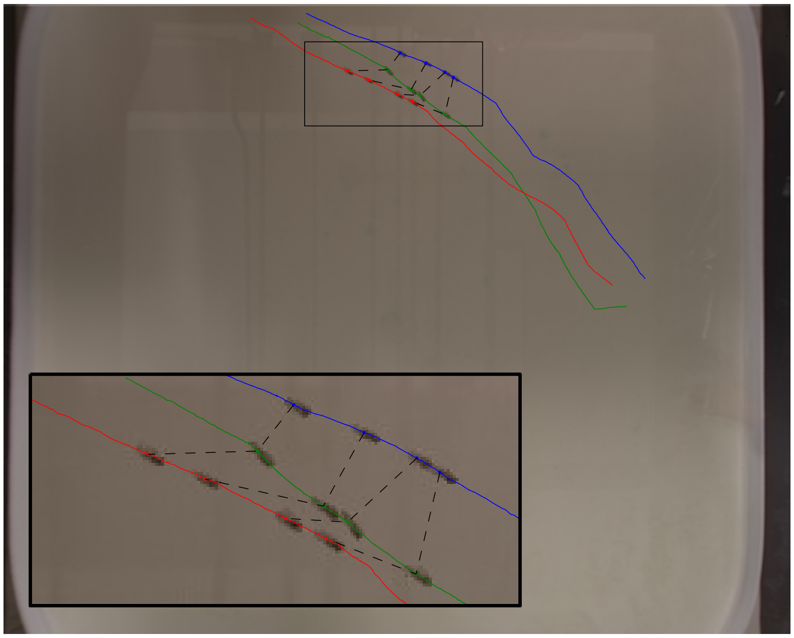


(a)

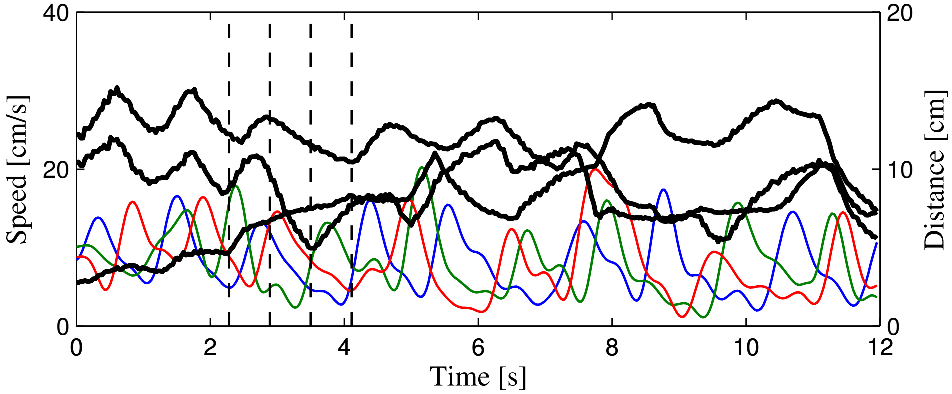


(b)

Figure 2.4: Relative position oscillation of a three-fish school (same data as shown in Fig. 2.1b). (a) The trajectories (blue, green, and red solid curves) of the three fish in the experimental arena. The direction of motion is down and right in the image. Four snapshots of the video are superimposed to show where the fish are at four evenly-spaced time instants. The dashed lines join the positions of nearest neighbors at each time instant. The four times are indicated by the vertical dashed lines in (b). The inset is a close-up of the boxed region. In all four snapshots, the rightmost two fish (blue and green trajectories) are side-by-side because their speeds are nearly synchronized. The leftmost fish (red trajectory) alternates between being closer to (second and fourth snapshots) and farther from (first and third snapshots) the other two fish. (b) Each black curve is the distance between a pair of fish. The blue, green, and red curves show the speeds of the fish with the blue, green, and red trajectories, respectively.



(a)



(b)

Figure 2.5: Relative position oscillation of a three-fish school (same data as shown in Fig. 2.1b). The presentation is the same as in Fig. 2.4 except that the four snapshots are taken later in time. The direction of motion is down and right. In the first and third snapshots, the middle fish (green trajectory) creates an occlusion between the two outer fish because it is between them. In the second and third snapshots, the middle fish pulls ahead and breaks the occlusion.

During the time period of the four marked time instants in Fig. 2.4b, the speed oscillations of the blue and green fish are approximately synchronized with one another, while both are approximately anti-synchronized with the red fish. Correspondingly, we see in Fig. 2.4a that the blue and green fish are consistently about the same distance from one another. This also corresponds to the lowest distance curve in Fig. 2.4b, which is relatively consistent in value when compared to the two upper distances curves. The two upper distance curves exhibit clear oscillation. In Fig. 2.4a, we see that the blue fish and the green fish are swimming roughly side-by-side. The red fish, on the other hand, moves regularly with respect to the blue fish and the green fish. The red fish begins far behind the green fish, gets closer to the other two by the time of the second snapshot, and falls behind again by the time of the third snapshot.

Fig. 2.5a shows another set of snapshots from the same trajectory data sequence that was used to create Fig. 2.4a, but at a different set of time instants. The speed oscillation phase relationships have changed from what was observed in Fig. 2.4. The blue fish is no longer synchronized with the green fish. The formation is V-shaped, with the green fish in the lead. In the first time instant, the green fish is between the red fish and the blue fish, causing an occlusion between the red fish and the blue fish. By the the second time instant, the green fish has pulled ahead of the red fish and blue fish, breaking the occlusion. The blue fish and red fish catch up with the green fish by the third time instant, and we see the pattern begin again by the time of the fourth time instant.

We calculated the MSP using (2.5) in order to obtain a measure of the phase relationship between pairs of killifish. For the two-fish schools, we calculated the MSP for all time segments during which the pair of fish were engaged in coordinated speed oscillations (i.e., at about the same frequency). The average value of the MSP for two-fish schools was -0.15 (see Table 2.2), indicating that the speed oscillations tend towards anti-synchronization. This is consistent with our observations, as seen for example in Fig. 2.1a. The average MSP for near neighbors in three-fish schools is 0.0 (see Table 2.2). This is consistent with near neighbors being sometimes synchronized and sometimes anti-synchronized. In Fig. 2.4, for example, the red fish and green fish are anti-synchronized near neighbors while the green fish and blue fish are synchronized near neighbors. In Fig. 2.5b, the phase relationships between the speed oscillations are less clear, but the variations in the formation shape in Fig. 2.5a are very clear.

2.3.2 Common Spatial Configurations

The results presented in this section were produced using the analysis algorithms described in Sec. 2.2. The same threshold parameter values were used here that were used to produce the results in Sec. 2.3.1.

A 2D histogram of the relative bearings β_1 and β_2 for the two neighbors of the central fish in three-fish schools is shown in Fig. 2.6. The values of β_1 and β_2 were calculated at each time instant for which the slowest fish had a nominal speed of at least 4 cm/sec and all fish were at least 12 cm (3 body lengths) away from the edge of the tank. Intervals less than 2 sec were not considered. Time instances during which either fish was more than 20 cm from the central fish were discarded. The results are shown in a histogram using 15 by 15 degree bins. The histogram is plotted as a heat map; low histogram values appear as blue tiles and high histogram values appear as red tiles. The most common configurations are therefore represented by clusters of red and/or orange tiles.

The clusters corresponding to the two most common configurations are circled in Fig. 2.6 with arrows connecting the clusters to illustrations of the corresponding configurations. Each illustration has three fish shown in three slightly different configurations. The middle configuration represents the nominal (average) configuration.

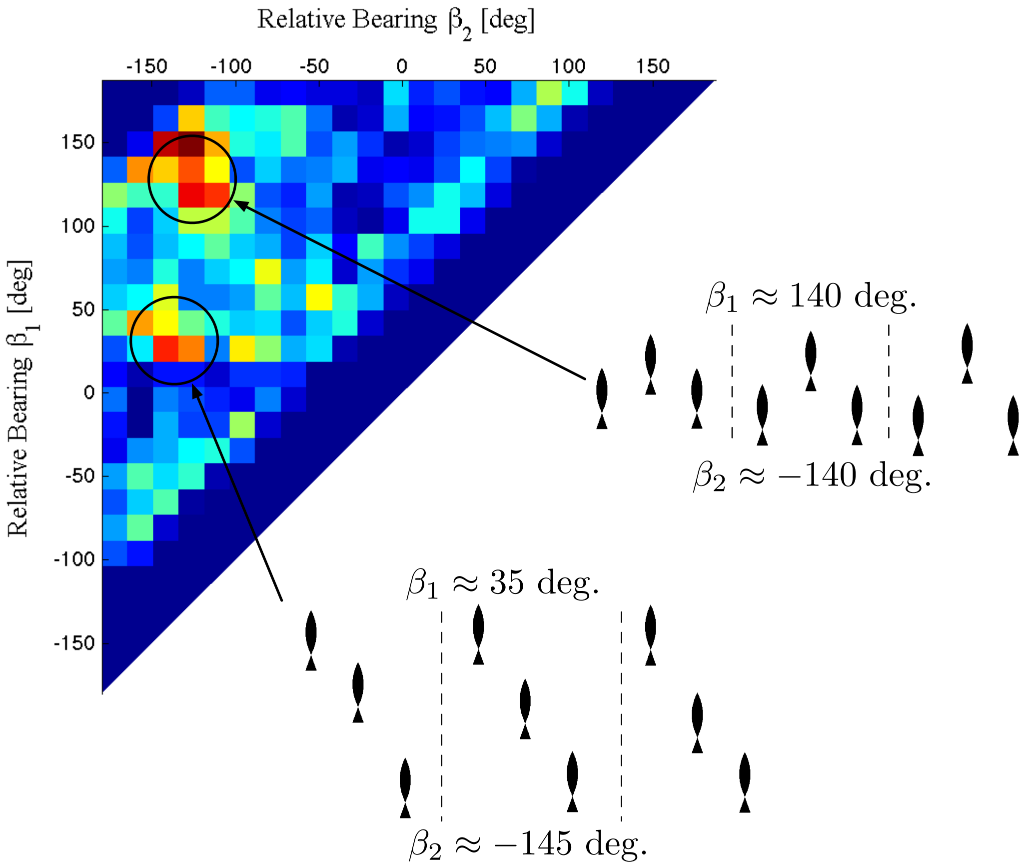


Figure 2.6: Histogram of relative bearing pairs showing the most common configurations (circled) of three-fish killifish schools. The two most common configurations are illustrated in the bottom right, along with changes caused by coordinated speed oscillations.

When the fish oscillate their speeds, they move relative to one other and therefore the configuration changes. The two outer configurations show the two extrema of relative motion about the nominal configuration, assuming that the two outer fish are engaged in anti-synchronized coordinated speed oscillations with the central fish.

The upper circle in Fig. 2.6 shows the most common configuration of the three-fish schools. The nominal relative bearings for this configuration are $(\beta_1, \beta_2) = (140 \text{ degrees}, -140 \text{ degrees})$, which corresponds to a v-shaped formation with the central fish in front (relative to the common direction of motion). Fig. 2.5a shows a clear example of the v-shaped formation. The second-most common configuration corresponds to the relative bearing pair $(\beta_1, \beta_2) = (35 \text{ degrees}, -145 \text{ degrees})$, which corresponds to a diagonal line formation with the left-most fish in front (relative to the common direction of motion). The average distance between the central and outside fish in the three-fish schools (averaged over all of the data represented by Fig. 2.6) is 7.1 ± 2.7 cm.

2.3.3 Relative Bearing Oscillations

In Sec. 2.3.1, we have shown that three-fish killifish schools engage in speed oscillations with average nominal speed $\nu = 8.3$ cm/sec, average amplitude $\mu = 4.1$ cm/sec, and average frequency $\Omega = 4.8$ rad/sec. From the videos and from two-fish school analysis, we know that nearest neighbors tend to antisynchronize their speed oscillations. From Sec. 2.3.2, the average separation between the central and outside fish is 7.1 cm and the relative bearing from a fish to its forward neighbor is approximately 35 or 40 degrees in the most common configurations of three-fish schools.

Consider a coordinate frame located at the eye of the central fish with its $x-$axis oriented along the direction of motion of the central fish (the direction of $\mathbf{v}_c$ in Fig. 2.2) and its $y-$axis perpendicular to the direction of motion and in the plane of motion. For a pair of fish engaged in coordinated speed oscillations with a phase offset $\Delta\phi$, their $x-$axis distance oscillates with an amplitude of

$$2\frac{\mu}{\Omega}\sin\left|\frac{\Delta\phi}{2}\right|.$$

For an average relative bearing of $\beta = 37.5$ degrees, the average $x-$axis distance between the fish is 5.6 cm and the average $y-$axis distance is 4.3 cm. For anti-synchronized oscillations, the amplitude of $x-$axis oscillations is 1.7 cm and therefore the relative bearing oscillates between $\tan^{-1}\frac{4.3}{5.6} \approx 34.9$ degrees and $\tan^{-1}\frac{7.3}{5.6} \approx 52.5$ degrees. Therefore, during an average oscillation period, the relative bearing between killifish cycles from the value that maximizes sensitivity of time-to-collision with respect to both heading and speed (35.3 degrees), through the value that maximizes sensitivity of angular velocity with respect to speed (45.0 degrees), and comes close to the value that maximizes sensitivity of loom with respect to heading (63.4 degrees).

Furthermore, optical flow is generated both into and across the retina because the image of the neighbor fish is moving and causing both angular velocity and loom to oscillate with nonzero amplitude. The amplitudes of these oscillations is strongest for anti-synchronized speed oscillations and nonzero unless speed oscillations are synchronized.

2.4 Discussion

The results for two-fish schools show that killifish engage in coordinated speed oscillations a majority of the time. The "typical" speed oscillation has a frequency of 0.74 Hz (well below the tail beat frequency, which is above 1 Hz) and an amplitude that is approximately 50% of the nominal speed. Pairs of fish engaged in coordinated speed oscillations tend to anti-synchronize their oscillations (i.e., their phase separation tends to be close to ±180 degrees).

The results also show that killifish in three-fish schools engage in speed oscillations a majority of the time, and that the typical speed oscillation of a fish in a three-fish school is very similar to the typical speed oscillation of a fish in a two-fish school. The phase coordination tends to change over time, with near neighbors being sometimes synchronized and sometimes anti-synchronized. In Fig. 2.4, for example, we see an example of a fish having speed oscillations nearly synchronized with one neighbor and nearly anti-synchronized with another neighbor. Regardless of speed phase arrangements, the spatial formation of the fish is continuously changing. This is especially apparent in Fig. 2.5.

The two most common spatial configurations for three-fish schools are the v-shaped formation and the diagonal line formation shown in Fig. 2.6. The relative bearing between any fish and its nearest neighbors to the front (left or right) tends to be between 30 and 40 degrees, though this (and the configurations themselves) vary slightly as the fish move relative to each other due to coordinated speed oscillations.

We show in Sec. 2.3.3 that killifish speed oscillations cause the relative bearing between fish to oscillate through a range of values that include those predicted by Dill et al. [37] to maximize sensitivity of various visual cues with respect to heading or speed changes by the other fish. The results of Dill et al. are derived for a static case; that is, the fish are assumed to be swimming with the same constant speed. Speed oscillations violate this assumption. New optimal values of relative bearing could be calculated using the dynamic rederivation of the results of Dill et al. that we present in App. B. However, the qualitative result will not change: coordinated speed oscillations cause relative motion that could result in a benefit in terms of increased overall sensitivity to various visual cues and therefore improved ability of the fish to detect and react to changes in the behavior of neighboring fish.

Chapter 3

Coordinated Relative Motion and Decision-Making Performance

In this chapter, we consider the effects of coordinated relative motion on the decision-making performance of multi-agent systems. This work is mainly motivated by the killifish observations in Chap. 2, although our goal is to understand underlying mechanisms and concepts that are relevant both for natural and engineered systems. Therefore, we refer to "agents" rather than specifically to fish, with the intention that an agent could be a model for a fish, bird, robot, mobile sensor, etc.

In Chap. 2, we presented evidence that schools of killifish engage in coordinated speed oscillations. We investigated potential benefits that the killifish may experience due to the resulting coordinated periodic relative motion, mainly with respect to enhanced quality of visual information (see Sec. 2.3.3). Here, we consider what effects relative motion may have with respect to the *availability* of visual information. That is, as killifish move relative to one another, line-of-sight visibility between distanced fish is modulated due to the making and breaking of occlusions by intermediate fish. We show below that this can, in fact, lead to dramatically improved group-level decision making performance. We also show that periodic relative motion can benefit engineered systems by reducing the communication range necessary for a group to maintain connectivity and accomplish group-level tasks.

This chapter begins by framing the relevant background material within the current discussion. In Sec. 3.1.1, we review some necessary concepts from graph theory so that we can describe and analyze the communication network by which agents influence one another as a graph. In Sec. 3.1.2, we provide some background on consensus dynamics; in particular, we are interested in linear consensus dynamics over graphs as a model of decision-making in multi-agent systems. In Sec. 3.1.3, we introduce the notion of a *sensing topology* as a means to describe how the spatial arrangement of a group of agents relates to the (time-varying) graph describing which agents can communicate with (or influence) which other agents.

The results of this chapter are organized into two groups. Results regarding the rate of convergence to consensus, which is used here as a way to measure decision making performance, are presented in Sec. 3.2. We provide a means to calculate the rate of convergence to consensus for piecewise constant and periodically time-varying systems in Sec. 3.2.1. In Sec. 3.2.2, this result is used along with the analysis of three-fish schools in Chap. 2 and some assumptions to show that killifish can be experiencing up to a 50% improvement in decision-making performance.

In Sec. 3.3, results are presented regarding the communication connectivity of multi-agent systems engaged in coordinated relative motion. In Sec. 3.3.1, we introduce the *effective sensing region*, which is a tool that we use to analyze connectivity. In Sec. 3.3.2, we use effective sensing regions to show that coordinated relative motion reduces the sensor footprint necessary for multi-agent systems to establish and maintain connectivity. In Sec. 3.3.3, we use both effective sensing regions and the convergence rate results of Sec. 3.2.1 to investigate the design of motion parameters for multi-agent systems with respect to decision-making performance.

Sec. 3.2 and Sec. 3.3 present two separate discussions about the benefits of periodic relative motion for decision-making performance. The results of Sec. 3.2 are mainly concerned with the convergence rates of consensus dynamics over periodically switching graphs. We focus on one pair of graphs that results from applying some assumptions to the spatial configurations of killifish that we discuss in Chap. 2, though the results can be leveraged to analyze decision-making performance over arbitrary periodic graph sequences. The results in Sec. 3.3 are concerned mainly with the connectivity of geometric graphs in which the individual nodes are moving periodically relative to one another. Both sections are motivated by our observations of killifish.

The results here on effective sensing regions and the rate of convergence to consensus for periodically time-varying graphs was previously published in [113] (which was, in turn, partly inspired by work on convergence rates by Cao et al. [25]). The results that relate to killifish decision-making performance are also part of a manuscript that is in preparation [114].

3.1 Background

3.1.1 Graph Theory

Here we review some graph theoretical notions that are necessary for our discussion of group-level decision making. The viewpoint and terminology used here is largely inspired by the work of Sepulchre et al. [105] and Moreau [80], who were concerned with the behavior of multi-agent systems whose dynamics evolve over time-varying graphs. For a thorough treatment of graph theory, see [46].

A *graph* in this context refers specifically to a $\delta-$*digraph*. A $\delta-$digraph $\mathcal{G}$ with N nodes is defined as the set

$$\mathcal{G} = \{\mathcal{V},\ \mathcal{E},\ A\}\,,$$

where $\mathcal{V}$ is the *node* set, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the *edge* set, and $A \in \mathbb{R}^{N \times N}$ is the *adjacency matrix*. The $(i, j)^{th}$ element of the adjacency matrix specifies the weight of the edge, and must satisfy the conditions

$$A_{ij} > \delta \text{ for some } \delta > 0 \iff (i, j) \in \mathcal{E}$$

and

$$A_{ij} = 0 \iff (i, j) \notin \mathcal{E}.$$

We consider only graphs with no self-edges; that is, $(i, i) \notin \mathcal{E}$ and $A_{ii} = 0$ for each $i = 1, \ldots, N$. A graph is *undirected* if $A_{ij} = A_{ji}$ for all i and j, and *directed* otherwise. We generally consider directed graphs here.

A graph with fixed number of nodes, N, is *time-varying* if its edges and/or edge weights are time-varying, i.e., $\mathcal{E} = \mathcal{E}(t)$ and/or $A = A(t)$. We write $\mathcal{G}(t)$ to refer to a time-varying graph at time t or to clarify that a graph is time-varying. If $\mathcal{E}$ and A are constant, we say that the graph is time-invariant. If $\mathcal{E}$ and A are piecewise constant, we say that the graph is piecewise constant. The *neighbor set* of the k^{th} node at time t is $\mathcal{N}_k(t) = \{j \in \mathcal{V} : (k, j) \in \mathcal{E}(t)\}$. $|\mathcal{N}_k(t)|$ is the cardinality of $\mathcal{N}_k(t)$ and therefore equal to the number of neighbors that node k has at time t.

The *connectivity* of graphs is important in many contexts. A graph $\mathcal{G}$ is *strongly connected* if and only if any two distinct nodes can be connected by a path that respects the edges and edge directions of the graph. That is, given any pair of nodes i and j, either $(i, j) \in \mathcal{E}$ or there is a sequence $i_1, \ldots, i_n$ for some $n \geq 1$ such that (i, i_1), (i_1, i_2), ... , and (i_n, j) are all in $\mathcal{E}$. A graph is *weakly connected* if and only if any two distinct nodes can be connected by a path that respects the edges of the graph but may not necessarily respect edge directions. Node i is *connected* to node j if there is a path from i to j that respects the edges and edge directions of the graph.

Consider a time-varying graph $\mathcal{G}(t) = \{\mathcal{V},\ \mathcal{E}(t),\ A(t)\}$ and a non-empty time interval I. We define the time-invariant graph

$$\bar{\mathcal{G}}_I = \{\mathcal{V},\ \bar{\mathcal{E}}_I,\ \bar{A}_I\} \tag{3.1}$$

where

$$\bar{\mathcal{E}}_I = \bigcup_{t \in I} \mathcal{E}(t)$$

and

$$\bar{A}_I = \frac{1}{|I|} \int_{t \in I} A(t)dt,$$

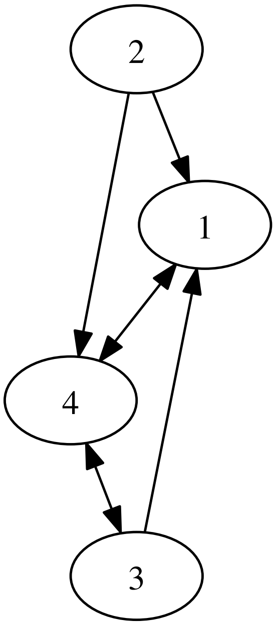
where $|I|$ is the (non-zero) length of I. We call $\bar{\mathcal{G}}_I$ the *interval graph* of $\mathcal{G}(t)$ over the interval I. Node i is *connected over the interval* I to node j in $\mathcal{G}(t)$ if node i is connected to node j in $\bar{\mathcal{G}}_I$. $\mathcal{G}(t)$ is *uniformly connected* (a notion introduced by Moreau [80]) if there exists a time horizon $T > 0$ such that, for all t, the interval graph $\bar{\mathcal{G}}_I$ is strongly connected over the interval $I = [t, t+T]$. A graph is *periodic* with a period T if $\mathcal{G}(t+T) = \mathcal{G}(t)$ for all t. A graph that is periodic with period T and strongly connected over any interval of length T is uniformly connected.

The *Laplacian matrix*, $L(t)$, of a graph $\mathcal{G}(t)$ is an important mathematical object. The Laplacian matrix is defined, in terms of its elements, as

$$L_{ij}(t) = \begin{cases} -A_{ij}(t), & i \neq j \\ \sum_{k=1}^{N} A_{ik}(t), & i = j \end{cases}.$$

The quantity $\sum_{k=1}^{N} A_{ik}(t)$ is called the *out degree* of node i at time t. Fig. 3.1 shows an example of a graph and its corresponding Laplacian matrix. Note that we draw edge (i, j) from node i to node j with the arrowhead at the j end of the connection when $L_{ij} < 0$ (equivalently, $(i, j) \in \mathcal{E}$ and $A_{ij} > 0$). In this thesis an edge (i, j)

with $A_{ij} > 0$ represents that agent i senses agent j and applies the weight A_{ij} to the information it gets from agent j.



$$L = \begin{bmatrix} 1 & 0 & 0 & -1 \\ -1 & 2 & 0 & -1 \\ -1 & 0 & 2 & -1 \\ -1 & 0 & -1 & 2 \end{bmatrix}$$

Figure 3.1: A simple graph (left) and its Laplacian matrix L (right). The edge weights (not drawn) are all equal to 1. An arrowhead on both ends of an edge between node i and node j indicate that both (i, j) and (j, i) are in the edge set $\mathcal{E}$.

The spectrum of the Laplacian matrix, L, of a time-invariant graph, $\mathcal{G}$, has important connections to the topological properties of the graph. By definition, L has at least one eigenvalue at 0 with a corresponding eigenvector $\mathbf{1}_N$ (the vector of all ones). The eigenvalues $\lambda_i,\ i = 1, \ldots, N$ all have $\operatorname{Re}\{\lambda_i\} \geq 0$ and only one eigenvalue has $\operatorname{Re}\{\lambda_i\} = 0$ if the graph is strongly connected [85].

3.1.2 Linear Consensus Dynamics Over Graphs

We define *linear consensus dynamics* here and review relevant results.

We consider consensus dynamics here as a model of the process by which a group of N agents attempt to reach agreement on the value of some decision variable. The k^{th} agent is modeled as having a *consensus variable*, x_k. Each agent updates the value of its consensus variable dynamically as it interacts with (possibly a subset of) the other agents, and the goal is to come to consensus:

$$\lim_{t \longrightarrow \infty} x_k(t) = \bar{x}, \ k = 1, \ldots, N.$$

$\bar{x} \in \mathbb{R}$ is called the *consensus value.*

We are focused here on linear consensus dynamics of the form

$$\dot{\mathbf{x}}(t) = -L(t)\mathbf{x}(t), \tag{3.2}$$

where $\mathbf{x} = \begin{pmatrix} x_1 & \ldots & x_N \end{pmatrix}^T \in \mathbb{R}^N$ and $L(t)$ is the Laplacian matrix for the graph $\mathcal{G}(t)$ that describes the interactions among the N agents. We say that these are consensus dynamics *over* the graph because the graph determines which agents influence which others. When there is no edge from agent k to agent j in the graph, $A_{kj} = L_{kj} = 0$ and hence $\dot{x}_k$ does not depend directly on x_j.

A special case of (3.2) is *averaging consensus*, wherein

$$\dot{x}_k = \begin{cases} \dfrac{1}{|\mathcal{N}_k(t)|} \displaystyle\sum_{j \in \mathcal{N}_k(t)} x_j(t) - x_k(t), & |\mathcal{N}_k(t)| \neq 0 \\ 0, & |\mathcal{N}_k(t)| = 0 \end{cases} . \tag{3.3}$$

That is, the k^{th} agent updates its own consensus variable in the direction of the average of the differences between its own consensus variable and its neighbors' variables (recall that $\mathcal{N}_k(t)$ is the neighbor set of the k^{th} agent at time t). Averaging consensus is equivalent to the graph Laplacian matrix being defined as

$$L_{kj}(t) = \begin{cases} 0, & j \neq k, j \notin \mathcal{N}_k(t) \\ -\frac{1}{|\mathcal{N}_k(t)|}, & j \neq k, j \in \mathcal{N}_k(t) \\ 1, & j = k, |\mathcal{N}_k(t)| \neq 0 \\ 0, & j = k, |\mathcal{N}_k(t)| = 0 \end{cases}.$$

It can be shown that, when $\mathcal{G}(t)$ is constant and undirected and strongly connected, the unique consensus value of the averaging consensus dynamics is the average of the initial values, i.e., $\bar{x} = \frac{1}{N}\sum_{k=1}^{N} x_k(0)$ [86]. In general, the consensus value depends on the initial conditions and the graph.

When the graph $\mathcal{G}(t)$ is time-invariant, the graph Laplacian matrix is time-invariant as well and the dynamics (3.2) become a linear time-invariant (LTI) system:

$$\dot{\mathbf{x}} = -L\mathbf{x}. \tag{3.4}$$

The LTI consensus dynamics (3.4) are always stable (the eigenvalues of L have non-negative real part). The number of zero eigenvalues of L determines the number of stable equilibria of the system; therefore, the system has a single stable equilibrium manifold $\mathbf{x}^* = \bar{x}\mathbf{1}_N$ for $\bar{x} \in \mathbb{R}$ if the graph $\mathcal{G}$ is strongly connected [85]. That is, the LTI consensus dynamics reach the consensus state, $x_1 = x_2 = \ldots = x_N = \bar{x}$, for all initial conditions if the graph is strongly connected. Moreau [80] extended this result to the time-varying case: the linear consensus dynamics (3.2) reach the consensus state if the graph $\mathcal{G}(t)$ is uniformly connected.

We use the rate of convergence to consensus, σ, (following [25, 113]) as a measure of the performance of a consensus dynamical system.

Definition 3.1. *The* rate of convergence to consensus, *σ, of the linear consensus dynamics (3.2) is defined as the infimum over all $\bar{\sigma} > 0$ for which there exists a $\beta > 0$*

for any initial conditions $\mathbf{x}(0) \in \mathbb{R}^N$ *such that*

$$\|\mathbf{x}(t) - \bar{x}\mathbf{1}_N\| \leq \beta e^{-\bar{\sigma}t}, \tag{3.5}$$

where $\bar{x}$ *is the consensus value corresponding to the initial conditions,* $\mathbf{x}(0)$*. If convergence is not reached (i.e., if the graph is not strongly connected), we say that* $\sigma = 0$*. When it is not ambiguous, we will refer to* σ *simply as the* convergence rate*.*

The convergence rate is determined by L and therefore by the properties of the graph $\mathcal{G}$. For time-invariant graphs, it is straightforward to show that the convergence rate is equal to the real part of the second smallest eigenvalue, λ_2, of L. For symmetric graphs, L is symmetric and $\lambda_2 \in \mathbb{R}^+$ is called the algebraic connectivity or Fiedler constant (see, e.g., [46, 59]). When the graph is time-varying, the convergence rate can be difficult to compute. Lower bounds on the convergence rate can be established using graph theoretic ideas such as the joint spectral radius and scrambling constants [12, 25]. The main result in Sec. 3.2.1 below is a method to calculate the convergence rate for periodic and piecewise-constant graphs in terms of the Laplacians of the individual graphs and their time durations during a cycle.

3.1.3 Sensing Topologies and Sensing Regions

Limited communication in multi-agent systems is modeled here using a graph, $\mathcal{G}$, for which the k^{th} node represents the k^{th} agent, and an edge from node j to node k (i.e., $(j, k) \in \mathcal{E}$) means that agent k is *influencing* agent j. Equivalently, we can say that agent j can *sense* node k when there is an edge from node j to node k. Therefore, the graph $\mathcal{G}$ is referred to as the *sensing topology* of the group. We say that agent k is a *neighbor* of agent j (in the context of the sensing topology) if $(j, k) \in \mathcal{E}$, or equivalently $k \in \mathcal{N}_j$.

We consider here only settings in which there is a single sensing topology for a group, although it is possible to make use of multiple topology models for a single group. For example, Paley et al. [87] adapt the zone model of Couzin et al. [29] to study spatial bistability in collective motion. They explicitly model the effects of long-range attraction, short-range repulsion, and mid-range alignment using a separate topology for each behavior.

Limited communication often derives from sensing limitations that can be modeled geometrically. For example, radio and visual signals degrade with increasing distance. The way that this phenomenon affects a multi-agent system can be modeled using a *sensing radius*; a fixed distance, ρ, beyond which communication is considered impossible. Another example is forward-facing vision, which results in a *blind angle*, a region to the rear of an agent in which it cannot see other agents. In general, we may consider a spatial region, Γ_k, that defines the area about agent k in which it may have neighbors. That is, if agent j is not in Γ_k, then agent j cannot be a neighbor of agent k. We call Γ_k the *sensing region* of agent k ([87] uses a similar concept with the name "perceptual zone").

There are many situations in which agent j being in agent k's sensing region is a necessary and sufficient condition for $(j, k) \in \mathcal{E}$ (i.e., every agent in Γ_k is a neighbor), although this is not always the case. Consider, for example, an agent that relies on visual information from its neighbors and whose sensing region is defined by a sensing radius that corresponds to the maximum distance at which visual information is reliably available. In this case, having another agent within the sensing region is a necessary condition for that agent to be a neighbor because visual information is unavailable from agents outside the sensing region. However, being inside the sensing region is not a sufficient condition for being a neighbor because the view may be occluded (for example, by a third agent or by a part of the environment). Another example of when being in the sensing region represents a necessary but not sufficient condition for being a neighbor is when there is an upper limit on the number of neighbors that the agent can have (for example, due to perceptual limitations). There is evidence that this is the case for birds within starling flocks [6].

The sensing region Γ_k for agent k is, in general, a time-varying region of the fixed physical space because it moves with the agent. We consider here the two-dimensional case, although there is no reason that this concept cannot be extended to three dimensions. For convenience, we identify the real and complex planes, i.e., $r_k \in \mathbb{C} \sim \mathbb{R}^2$ is the position of the k^{th} agent. The heading direction of the k^{th} agent is $\theta_k \in S^1$, where S^1 is the circle. The sensing region for agent k is therefore

$$\begin{aligned}\Gamma_k(t) &= e^{i\theta_k(t)}\tilde{\Gamma}_k + r_k(t) \\ &\triangleq \left\{x \in \mathbb{C} : (x - r_k(t))\, e^{-i\theta_k(t)} \in \tilde{\Gamma}_k\right\},\end{aligned} \tag{3.6}$$

where $\tilde{\Gamma}_k$ is the *template sensing region* for agent k. The template sensing region is the sensing region for agent k defined relative to the position and heading of agent k; we define the template sensing region for the agent located at the origin with heading in the zero direction. Then, the sensing region for agent k at time t is found by translating the template region to $r_k(t)$ and rotating it by $\theta_k(t)$. In many cases, the template sensing regions of all agents in a group are identical. For example, a group of robots may all have the same sensing radius, ρ, and blind angle, β. In these cases, we simply write Γ as the common template sensing region and

$$\Gamma_k(t) = e^{i\theta_k(t)}\Gamma + r_k(t).$$

We assume that Γ, and consequently Γ_k, is an open set so that connectivity over intervals of zero length is not possible. Note that the sensing region, Γ, cannot be a point.

Fig. 3.2 shows some common template sensing regions. When a sensing region is defined by a radius, ρ, we call that radius the *sensing radius*. When a sensing region excludes a cone defined by an angle 2β to the rear, we call one half of that angle, β, the *blind angle* (i.e., a blind angle of 10 degrees means that neighbors within ±10 degrees directly to the rear cannot be seen; the total angle defining the excluded cone is 20 degrees).

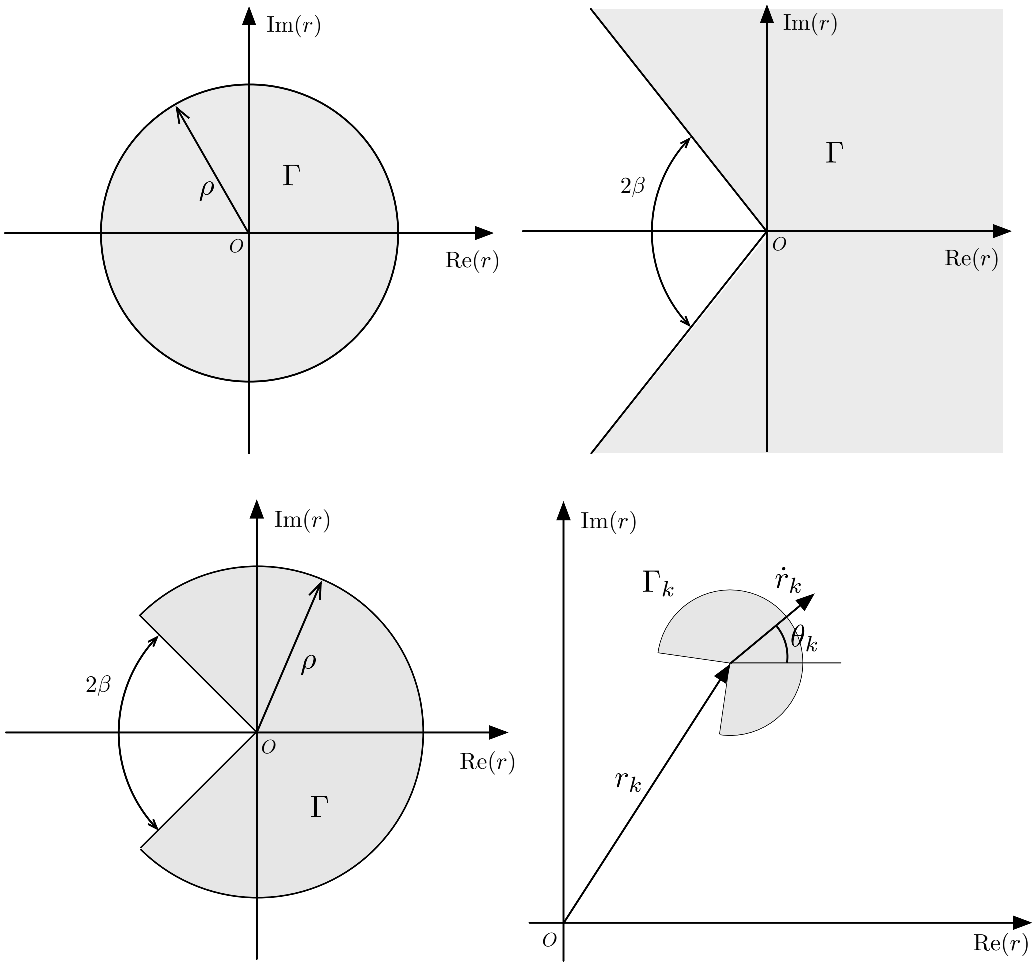


Figure 3.2: Example template sensing regions. (Top left) Sensing radius, ρ. (Top Right) Blind angle, β. (Bottom left) Combination of a sensing radius and blind angle. (Bottom right) The sensing region for agent k at time t, obtained by translating the template sensing region to the agent's position at $r_k(t)$ and aligning it with its velocity at $\theta_k(t) = \angle \dot{r}_k(t)$.

In Sections 3.2 and 3.3, we use the *effective sensing regions* for a group of agents to evaluate the role of periodic relative motion on decision making (Sec. 3.2) and connectivity (Sec. 3.3). In Sec. 3.3, we consider agents that are randomly placed within a unit square and have a fixed sensing radius ρ, and investigate the connectivity of the sensing topology as a function of ρ and the parameters of relative motion. In the static case (i.e., no relative motion), this defines a random geometric graph [96]. The static case has been studied in the context of sensor networks [5, 100] using results from graph percolation theory [15, 75]. The focus of these studies was the phase transition in graph connectivity as the sensing radius increases. For a fixed number of agents, N, the sensing topology transitions from disconnected to connected when the sensing radius, ρ, is varied from small values (disconnected) to large values (connected). This work is done in a probabilistic framework because the positions of the agents/sensors are random; therefore the goal is to find a critical sensing radius, ρ_c, for which the probability of connectedness transitions from low to high. Balister et al. [5] combined theoretical analysis with Monte Carlo simulations to obtain the estimated bounds

$$\sqrt{\frac{1.43}{N}} < \rho_c < \sqrt{\frac{1.48}{N}} \tag{3.7}$$

with probability 0.9999 when agents are placed indepently and uniformly randomly in a unit square. We use this result in Sec. 3.3 as a baseline for comparison with the results that we obtain when relative motion is introduced.

3.2 Group-level Decision-Making Performance With Periodic Relative Motion

In this section, we present results on the rate of convergence to consensus for periodic and piecewise constant graphs. In Sec. 3.2.1, we present analytical results that facilitate the calculation of convergence rate in terms of the component Laplacian matrices and their time durations. In Sec. 3.2.2, we use the results from Sec. 3.2.1 along with some assumptions about the sensing topologies in the three-fish killifish schools to show that coordinated speed oscillations can result in a significant improvement to group-level decision making performance with respect to the rate of convergence to consensus.

3.2.1 Rate of Convergence to Consensus Over Periodic and Piecewise Constant Graphs

Consider a graph, $\mathcal{G}(t)$, that is piecewise constant and periodic with period T. During each period, the graph cycles through a set of n component graphs. We label the i^{th} component graph $\mathcal{G}_i$ and its duration $\Delta t_i > 0$ ($\sum_{i=1}^{n} \Delta t_i = T$). Define $t_i \triangleq \sum_{j=1}^{i} \Delta t_i,\ i = 1, \ldots, n$. The graph $\mathcal{G}(t)$ can then be described as

$$\mathcal{G}(t) = \mathcal{G}\left(\tau = t - \left\lfloor \frac{t}{T} \right\rfloor T\right) = \begin{cases} \mathcal{G}_1, & 0 \leq \tau < t_1 \\ \mathcal{G}_2, & t_1 \leq \tau < t_2 \\ & \vdots \\ \mathcal{G}_n, & t_{n-1} \leq \tau < T \end{cases}. \tag{3.8}$$

For notational convenience, we define $L_0 = I_{N \times N}$, $\Delta t_0 = 0$, $t_0 = 0$, $L_{n+1} = L_1$, $\Delta t_{n+1} = \Delta t_1$, and $t_{n+1} = t_1$. The Laplacian matrix of the i^{th} component graph is labeled L_i. We say that the i^{th} graph is *active* when $t_{i-1} \leq t - \left\lfloor \frac{t}{T} \right\rfloor T < t_i$.

The following result allows us to calculate the rate of convergence to consensus for a periodic and piecewise-constant graph in terms of the Laplacian matrices, L_i, of the component graphs, $\mathcal{G}_i$, and their time durations, Δt_i.

Theorem 3.1. *Consider the linear consensus dynamics (3.2) where $L(t)$ is the Laplacian matrix of a periodic and piecewise constant graph as described by (3.8). Assume* *that the interval graph over one period, $\bar{\mathcal{G}}_{I=[0,T)}$ as defined by (3.1), is strongly connected. The rate of convergence to consensus, σ, for this system as defined by Def. 3.1 is given by*

$$\sigma = -\frac{1}{T}\log|m_2|, \tag{3.9}$$

where m_2 is the second largest (by magnitude) eigenvalue of the matrix

$$M = e^{-L_n \Delta t_N} \cdots e^{-L_1 \Delta t_1}. \tag{3.10}$$

When $\bar{\mathcal{G}}_{I=[0,T)}$ is not strongly connected, we say that $\sigma = 0$.

Proof. Consider a time, t, where $qT \leq t < (q+1)T$ and $t_{\bar{n}-1} \leq t - qT < t_{\bar{n}}$ for integers $q \gg 1$ and $1 \leq \bar{n} \leq n$. That is, q periods of length T have elapsed and the $\bar{n}^{th}$ graph is active. The solution to the linear consensus dynamics (3.2) at time t is given by

$$\mathbf{x}(t) = M_1 M^q \mathbf{x}(0), \tag{3.11}$$

where M is defined by (3.10) and

$$M_1 = e^{-L_{\bar{n}}(t - t_{\bar{n}-1})} e^{-L_{\bar{n}-1}\Delta t_{\bar{n}-1}} \cdots e^{-L_1 \Delta t_1}$$

accounts for the interval for which $t > qT$. Because $\bar{\mathcal{G}}_{I=[0,T)}$ is strongly connected, $\mathcal{G}(t)$ is uniformly connected, and therefore we have $\lim_{q\to\infty} M^q \mathbf{x}(0) = \bar{x}\mathbf{1}_N$ for some $\bar{x} \in \mathbb{R}$.

M^q dominates the asymptotic behavior of (3.11) as q increases and therefore the magnitude of its second largest eigenvalue, m_2, of M (ordered by magnitude) determines the rate at which $\|M^q \mathbf{x}(0) - \bar{x}\mathbf{1}_N\| \longrightarrow 0$ as a function of q [25]. Let $\bar{L} = -\frac{1}{T}\log M$ so that we have $\lim_{q\to\infty} e^{-\bar{L}qT} = \bar{x}\mathbf{1}_N$. Therefore, the real part of the second smallest eigenvalue, λ_2, of $\bar{L}$ (ordered by real parts) determines the rate at which $\left\|e^{-\bar{L}t}\mathbf{x}(0) - \bar{x}\mathbf{1}_N\right\| \longrightarrow 0$ as a function of t, and $\mathrm{Re}\,\{\lambda_2\} = -\frac{1}{T}\log|m_2|$. $\square$

The following corollary states that the value of the rate of convergence to consensus is unaffected by time shifts or, equivalently, what time we mark as the beginning of a period.

Corollary 3.1. *Consider the conditions of Thm. 3.1. The value of the convergence rate,* σ*, is invariant to a time-shift of the graph. That is, the convergence rate for* $\mathcal{G}(t-t^*)$*,* $t^* \in \mathbb{R}$*, is equal to the convergence rate for* $\mathcal{G}(t)$*. Equivalently, the convergence rate is unaffected by a circular shifting of the graph sequence - i.e., the convergence rate is the same for the graph sequences* $(\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_n)$*,* $(\mathcal{G}_2, \ldots, \mathcal{G}_n, \mathcal{G}_1)$*,* $(\mathcal{G}_n, \mathcal{G}_1, \ldots, \mathcal{G}_{n-1})$*, etc., so long as the order and the time durations are the same.*

Proof. Assume that $0 < t^* < T$ and that $t^* = t_{\bar{n}-1} + \delta t^*$, where $0 \leq \delta t^* < \Delta t_{\bar{n}}$ (i.e., the $\bar{n}^{th}$ graph of $\mathcal{G}(t)$ is active at $t = t^*$). Because $\mathcal{E}(t)$ is periodic, $\bar{\mathcal{E}}_{I^*=[t^*,t^*+T)} = \bar{\mathcal{E}}_{I=[0,T)}$. Therefore, if $\mathcal{G}(t)$ is uniformly connected then so is $\mathcal{G}(t-t^*)$. Hence, Thm. 3.1 applies with $\sigma^* = -\frac{1}{T}\log|m_2^*|$ where m_2^* is the second largest eigenvalue (by magnitude) of

$$\begin{aligned} M^* &= e^{-L_{\bar{n}}\delta t^*} e^{-L_{\bar{n}-1}\Delta t_{\bar{n}-1}} \cdots e^{-L_1 \Delta t_1} e^{-L_n \Delta t_n} \cdots e^{-L_{\bar{n}+1}\Delta t_{\bar{n}+1}} e^{-L_{\bar{n}}(\Delta t_{\bar{n}} - \delta t^*)} \\ &= \bar{M} M \bar{M}^{-1} \end{aligned}$$

where M is from the original ordering via (3.10) and

$$\bar{M} = e^{L_{\bar{n}}(\Delta t_{\bar{n}} - \delta t^*)} e^{L_{\bar{n}+1}\Delta t_{\bar{n}+1}} \cdots e^{L_{n-1}\Delta t_{n-1}} e^{L_n \Delta t_n}.$$

M and M^* have the same eigenvalues by similarity, and therefore $m_2^* = m_2$ and $\sigma^* = \sigma$.

If $\mathcal{G}(t)$ is not uniformly connected, then neither is $\mathcal{G}(t-t^*)$ and saying that $\sigma = 0$ is congruent with Thm. 3.1. □

3.2.2 Decision-Making Performance of Killifish Schools

In this section, we consider schools of $N = 3$ killifish and their periodic sensing topologies that result from coordinated speed oscillations. The sensing topologies are derived from the common spatial configurations described in Sec. 2.3.2 and some assumptions (described below) about how the killifish are influenced by visual information. We describe performance in terms of the rate of convergence to consensus over these periodic sensing topologies. The linear consensus dynamics (3.2) can be viewed as a linearization of consensus about the direction of motion.

It is impossible to know precisely the sensing topology of a school of fish. We can measure line-of-sight visibility based on spatial configurations, but there is no easy way to know to what degree one fish may influence another fish nor how this influence varies with time. We therefore apply a set of assumptions to the common spatial configurations described in Sec. 2.3.2 in order to obtain a set of proposed sensing topologies. Our assumptions are motivated by our interest in the way that visual information is modulated by relative motion. The assumptions are as follows:

1. Fish k cannot be a neighbor of fish j unless fish j can *see* fish k.

2. Each fish has at most one neighbor to the left and one neighbor to the right. When more than one fish is visible to the left (respectively, right), then the most recently unoccluded fish to the left (respectively, right) is taken as the neighbor.

3. At any time and for any fish, the influence due to its neighbors is evenly distributed across all neighbors. Furthermore, the total influence on a given fish stays constant over time, unless there are no neighbors, in which case the total influence drops to zero.

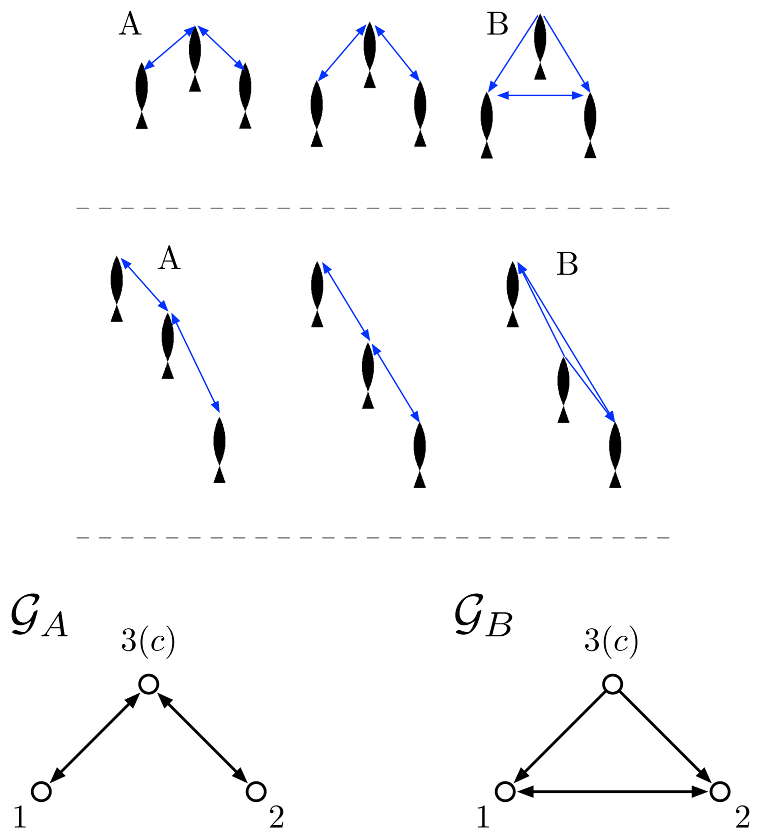
The first assumption follows from the assumption that killifish are reacting to visual information without memory. The second assumption is reasonable if we propose that fish have a limited amount of visual "attention", and that the newest-available information warrants the most attention. The third assumption implies that the row sums of the adjacency matrix, A, are either equal to 1 (if there are any visible neighbors) or 0 (if there are no visible neighbors). Equal weights correspond to the non-zero values in each row of A being equal to each other. Therefore, the third assumption is equivalent to assuming that the fish are engaged in *averaging* consensus, as in (3.3).

Consider the common killifish spatial configurations for three-fish schools shown in Fig. 2.6. When the fish move relative to one another in an oscillatory manner resulting from coordinated speed oscillations, line-of-sight visibility between the two outer fish is alternately available and occluded. Applying the above assumptions to this regular occlusion making and breaking yields switching between the two topologies shown in Fig. 3.3. The resulting graphs and their corresponding Laplacian matrices are also shown in Fig. 3.3. Note that both common configurations (the v-shaped formation and the diagonal-line formation) from Fig. 2.6 yield the same two component graphs and therefore the same sensing topology. The spatial configurations shown in the snapshots in Fig. 2.4 also result in the sensing topology generated by switching between the two graphs shown in Fig.3.3. We will refer to the two component graphs below as $\mathcal{G}_A$ and $\mathcal{G}_B$, as labeled in Fig. 3.3.

We now compute the rate of convergence to consensus for the graph $\mathcal{G}(t)$ resulting from the switching sensing topology shown in Fig. 3.3. The convergence rate is calculated here as a function of a *duty cycle* parameter, p, $0 \leq p \leq 1$, that represents the fraction of a period for which $\mathcal{G}_B$ is active. Hence, the graph $\mathcal{G}(t)$ is described by

$$\mathcal{G}(t) = \mathcal{G}\left(\tau = t - \left\lfloor \frac{t}{T} \right\rfloor\right) = \begin{cases} \mathcal{G}_A, & 0 \leq \tau < (1-p)T \\ \mathcal{G}_B, & (1-p)T \leq \tau < T \end{cases}, \tag{3.12}$$

Note that when $p = 0$, $\mathcal{G}(t) = \mathcal{G}_A$ for all t, and when $p = 1$, $\mathcal{G}(t) = \mathcal{G}_B$ for all t.



$$L_A = \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ -\frac{1}{2} & -\frac{1}{2} & 1 \end{pmatrix} \qquad L_B = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ -\frac{1}{2} & -\frac{1}{2} & 1 \end{pmatrix}$$

Figure 3.3: Switching killifish sensing topology resulting form the common configurations described in Sec. 2.3.2 (See Fig. 2.6) and the assumptions described in the text. (Top and middle) The v-shaped and diagonal-line formations (middle) and their extrema of relative motion due to coordinated speed oscillations (left and right). The blue lines indicate neighbor relationships due to the assumptions described above. (Bottom) The formations above result in the same two component graphs, $\mathcal{G}_A$ and $\mathcal{G}_B$, shown with their corresponding graph Laplacian matrices L_A and L_B. The node labels give their indices in the Laplacian matrices. (c) indicates the central (middle) fish. The periodically time-varying sensing topology of the killifish school consists of these two component graphs, with the switches occuring when the spatial configuration of the fish changes due to relative motion thus making and breaking occlusions between fish.

Both component graphs $\mathcal{G}_A$ and $\mathcal{G}_B$ are strongly connected and therefore $\mathcal{G}(t)$ is uniformly connected. Thus, by Thm. 3.1, the rate of convergence to consensus for

$\mathcal{G}(t)$ is determined by the second largest eigenvalue of

$$M = e^{-pTL_B} e^{-(1-p)TL_A}. \tag{3.13}$$

From the killifish data above, we have that $T \approx 1.3$ sec. The value of p depends on the killifish' parameters of motion and could vary from experiment to experiment, or even over the course of a single experiment. Nominally, the value should be around $p \approx 0.5$. Regardless, the results below are calculated as a function of p and we show that a performance benefit is present for any value of $p \neq 0$ and $p \neq 1$ (i.e., when any switching is present).

The second largest eigenvalue of M in (3.13) with L_A and L_B from Fig. 3.3 is

$$m_2 = \begin{cases} e^{-T(p+1)}, & 0 \leq p < \frac{1}{2} \\ e^{T(p-2)}, & \frac{1}{2} \leq p \leq 1 \end{cases}.$$

Therefore, the convergence rate is

$$\sigma = \begin{cases} p+1, & 0 \leq p < \frac{1}{2} \\ 2-p, & \frac{1}{2} \leq p \leq 1 \end{cases},$$

and we see that the convergence rate is maximized at a value of $\sigma = \frac{3}{2}$ when $p = \frac{1}{2}$ and minimized at $p = 0$ and $p = 1$ with a value of $\sigma = 1$. That is, the value of the convergence rate is 50% larger when the graphs are switched with a duty cycle of $p = \frac{1}{2}$ than it is for either component graph alone. In fact, $\sigma > 1$ for *all* values of $p \in (0, 1)$.

Note that the set of assumptions presented above is just one of many possibilities. Other possibilities that we have considered do not result in a decision-making performance benefit due to periodic relative motion. The second assumption above is the most critical of the three to obtaining a decision-making performance benefit because it strongly modulates the sensing topology. One alternative assumption is for fish to have a preference for influence from the nearest visible fish on either side, or all fish within a fixed sensing radius. Under this assumption, the sensing topology for the common killifish spatial configurations that we have studied is constant and represented by the graph $\mathcal{G}_A$ in Fig. 3.3. Therefore, the sensing topology does not change under this assumption and there is no decision-making performance benefit due to periodic relative motion. Similarly, if fish are influenced by all visible neighbors, then the sensing topology becomes an all-to-all graph for both the "A" and "B" configurations in the v-shaped formations in the top of Fig. 3.3 and for the "B" configuration in the diagonal-line formations in the middle of Fig. 3.3. The second-largest eigenvalue of the Laplacian matrix corresponding to the all-to-all topology is $\frac{3}{2}$, and hence the convergence rate is not improved by switching.

To investigate whether the decision-making performance improvement due to topology switching is an isolated phenomenon or not, we calculated the convergence rates for a variety of other graph pairs. We found that for many graph pairs, switching produced a higher convergence rate than either graph alone. Fig. 3.4 shows two examples for which switching is beneficial and one for which switching is not beneficial. The topmost pair of graphs is over $N = 3$ nodes and corresponds to a situation in which the second node periodically switches between being connected to the first and being connected to the third node. This could correspond to a physical situation in which a fish (represented by the second node) moves between two other fish (represented by the first and third nodes) that are too far apart to be influenced by one another directly. In this case, it is straightforward to see that switching should improve decision-making performance because neither component graph is strongly connected but the interval graph is strongly connected. The maximum value of σ occurs when $p = 0.5$. This kind of phenomenon may occur in larger groups where the sensing topologies are more complex. For example, the middle pair of graphs is over $N = 4$ nodes and shows an example for which $\mathcal{G}_A$ is strongly connected and $\mathcal{G}_B$ is not. Switching improves the convergence rate in this case, though it is not obvious from the two component graphs. The bottom pair of graphs is an example situation where switching *reduces* the convergence rate. The period is $T = 1$ in all three cases.

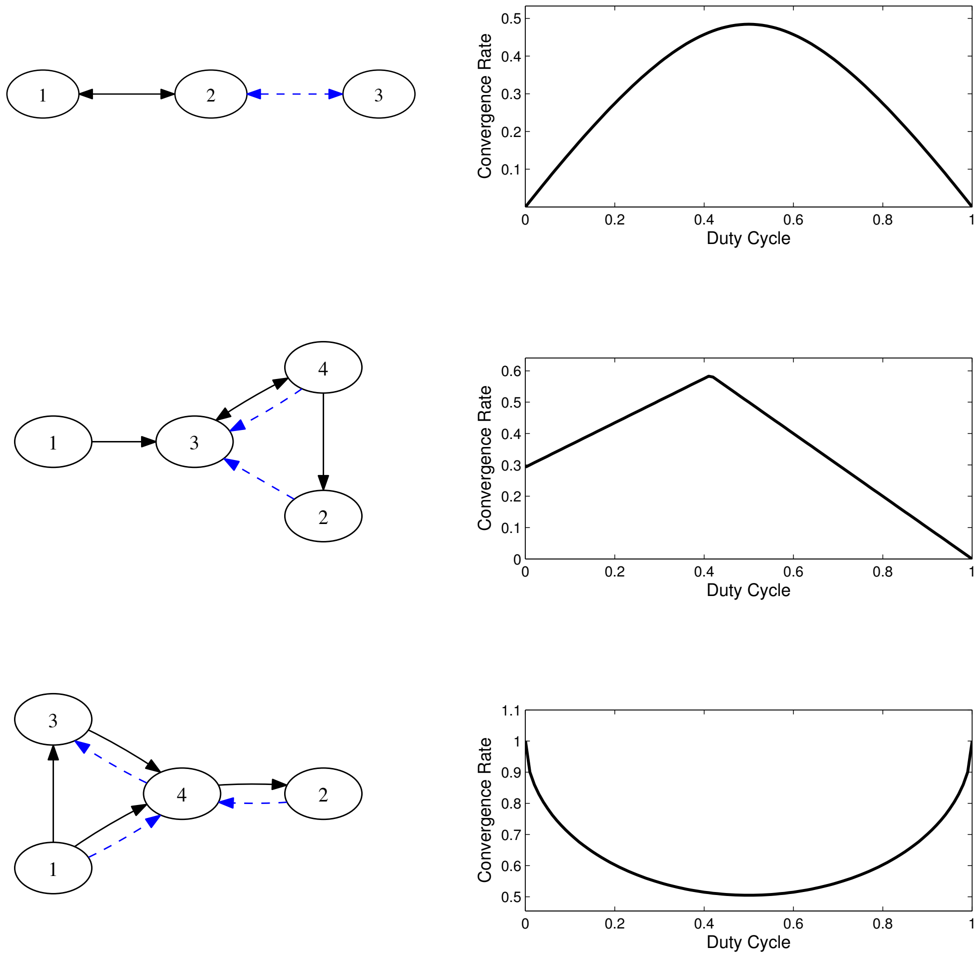


Figure 3.4: A sampling of topology pairs and their rate of convergence to consensus as a function of duty cycle. (Left) The topologies. The A topology is represented by the solid black edges, the B topology is represented by the dashed blue edges. (Right) The corresponding convergence rate as functions of the duty cycle p, calculated using the results of Thm. 3.1 with M as in (3.13). The period is $T = 1$ in all three cases.

3.3 Impact of Relative Motion on Connectivity

This section considers the conditions of connectedness for multi-agent systems engaged in coordinated relative motion, whereas the results of the previous section focused mainly on the performance of group-level decision making assuming that the underlying graph is connected. In contrast to the results of Sec. 3.2.2, where we construct a theoretical sensing topology for a school of killifish based on assumptions involving line-of-sight visibility, this section considers sensing topologies for multi-agent systems based on pairwise geometric relationships (for example, a fixed sensing radius). In both this section and Sec. 3.2, the sensing topologies that we consider can can be time-varying due to relative motion.

We introduce the effective sensing region in Sec. 3.3.1 and use it as a tool to analyze connectivity in Sec. 3.3.2. Sec. 3.3.3 uses effective sensing regions as well as the convergence rate results from above to investigate optimal motion parameters for decision-making performance.

3.3.1 Effective Sensing Regions

Consider a group of N agents that are engaged in periodic relative motion with period T. The relative position between agents j and k is $r_{jk}(t) \triangleq r_j(t) - r_k(t)$, and, because the relative motion is periodic, $r_{jk}(t+T) = r_{jk}(t)$ for all t. The average relative position of agent j with respect to agent k is

$$\bar{r}_{jk} \triangleq \frac{1}{T}\int_0^T r_{jk}(\tau)d\tau.$$

In this section we assume that any agent, j, that is in agent k's sensing region, $\Gamma_k(t)$, at time t is a neighbor of agent k at time t. That is,

$$(k,j) \in \mathcal{E}(t) \Longleftrightarrow r_j \in \Gamma_k(t), \tag{3.14}$$

where $\mathcal{E}(t)$ is the edge set of the graph $\mathcal{G}(t)$ representing the group's sensing topology. Furthermore, we assume that all agents share a common sensing region template, Γ. By (3.6), the right-hand side of (3.14) is equivalent to

$$r_{jk}(t)e^{-i\theta_k(t)} \in \Gamma. \tag{3.15}$$

Because $r_{jk} = \bar{r}_{jk} + (r_{jk} - \bar{r}_{jk})$, (3.15) is equivalent to

$$\bar{r}_{jk} \in \Gamma e^{i\theta_k(t)} - \left(r_{jk}(t) - \bar{r}_{jk}\right). \tag{3.16}$$

Strong connectivity of the interval graph $\bar{\mathcal{G}}_{I=[0,T)} = \left\{\mathcal{N}, \bar{\mathcal{E}}_I, \bar{A}_I\right\}$ is equivalent to uniform connectivity of $\mathcal{G}(t)$. Therefore, with respect to constructing a graph from which we can investigate uniform connectivity of the sensing topology, we need only determine if an edge is present at some time during the first period. That is, to determine if $(k, j) \in \bar{E}_I$, we may determine if there is any time $t^* \in [0, T)$ for which the condition (3.16) is true. We may consider a single time instant because Γ is an open set and therefore, if (3.16) is true for $t = t^*$, it is also true for $t \in (t^* - \delta t, t^* + \delta t)$ for some $\delta t > 0$. Therefore, $(k, j) \in \bar{\mathcal{E}}_I$ if and only if

$$\bar{r}_{jk} \in \bigcup_{t \in I=[0,T)} \left(\Gamma e^{i\theta_k(t)} - \left(r_{jk}(t) - \bar{r}_{jk}\right)\right) \triangleq \bar{\Gamma}_{jk}.$$

We call the set $\bar{\Gamma}_{jk}$ the *effective sensing region* for the pair (k, j).

We have proven the following.

Theorem 3.2. *Consider N agents that are engaged in periodic relative motion with period T and share a common sensing region template Γ. Their periodically time-varying sensing topology is uniformly connected if and only if the interval graph $\bar{\mathcal{G}}_{I=[0,T)} = \left\{\mathcal{N}, \bar{\mathcal{E}}_I, \bar{A}_I\right\}$ is strongly connected. Furthermore, the $N^2 - N$ ($\frac{N^2-N}{2}$ if*

Γ *is isotropic) edges of* $\bar{\mathcal{E}}_I$ *can be determined by the effective sensing regions as*

$$(k, j) \in \bar{\mathcal{E}}_I \Longleftrightarrow \bar{r}_{jk} \in \bar{\Gamma}_{jk}.$$

3.3.2 Improved Connectivity of Random Graphs

In this section and the next, we consider agents with the same finite sensing radius, ρ, that move along straight lines with periodic speed profiles and the same average speed and direction of motion. Because only relative positions are relevant, we may assume that the average speed and heading are both zero. The speed of the k^{th} agent is modeled as

$$\dot{r}_k = \mu \cos \phi_k(t)$$

where μ is a constant amplitude and

$$\phi_k(t) = \Omega t + \phi_k(0)$$

is the speed phase with Ω the speed oscillation frequency. The position of the k^{th} agent is therefore

$$r_k(t) = \bar{r}_k + \frac{\mu}{\Omega} \sin \phi_k,$$

where $\bar{r}_k = r_k(0) - \frac{\mu}{\Omega} \sin \phi_k(0)$ is the average position.

The position of agent j relative to agent k is

$$r_{jk} \triangleq r_j - r_k = \bar{r}_{jk} + \frac{\mu}{\Omega} (\sin \phi_j - \sin \phi_k)$$

where $\bar{r}_{jk}$ is the average position of j relative to the average position of k. Let $\phi_{jk} = \phi_j - \phi_k = \phi_j(0) - \phi_k(0)$ and $\varphi_{jk} = \phi_j + \phi_k = 2\Omega t + \phi_j(0) + \phi_k(0)$. Applying

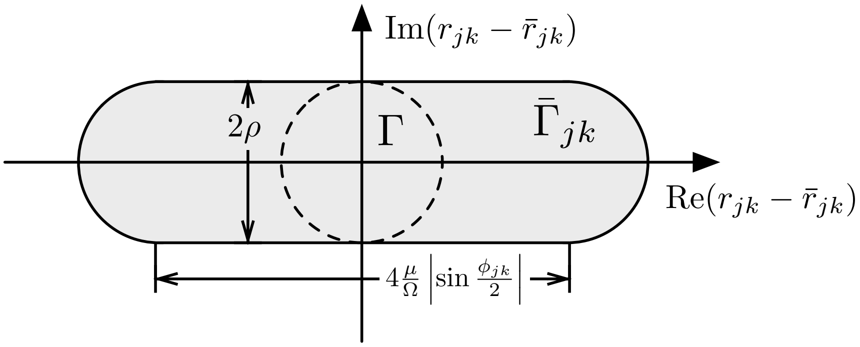


Figure 3.5: The effective sensing region for two agents undergoing periodic relative motion resulting from sinusoidal speed profiles with amplitude μ and frequency Ω. The oscillation phase difference between agents is ϕ_{jk} and they share a common sensing region template that is specified by a disc of radius ρ (i.e., a fixed sensing radius).

the trigonometric identity $\sin a - \sin b = 2\sin\frac{a-b}{2}\cos\frac{a+b}{2}$ yields

$$r_{jk} - \bar{r}_{jk} = 2\frac{\mu}{\Omega}\sin\frac{\phi_{jk}}{2}\cos\frac{\varphi_{jk}}{2}. \tag{3.17}$$

Therefore, the relative positions oscillate about the average relative position $\bar{r}_{jk}$ with a frequency Ω and amplitude $2\frac{\mu}{\Omega}\left|\sin\frac{\phi_{jk}}{2}\right|$.

The effective sensing region $\bar{\Gamma}_{jk}$ between two agents is computed by translating the sensing region template — a circle of radius ρ — along a line whose length is equal to the peak-to-peak amplitude of the relative position oscillations. No rotation is required because the headings are identical. The resulting shape is shown in Fig. 3.5. Note that only the length of the shape changes from pair to pair, and that this length is a function of the phase difference ϕ_{jk}.

We used effective sensing regions to perform numerical simulations that evaluate the connectivity of the sensing topology of a group of $N = 100$ agents undergoing periodic relative motion. We use these simulations to compare the connectivity of the sensing topology with the sensing topology that would be obtained without relative motion, which is equivalent to a random geometric graph as described in Sec. 3.1.3.

The upper bound in (3.7) is used as a baseline for comparison to the random geometric graph case.

Each agent in our simulation was placed at a uniformly random initial position within a unit square and the connectivity of the sensing topology was calculated for a range of conditions. The speed oscillation frequency was $\Omega = 1$ and the amplitude was $\mu = 0.3$. The sensing radius was varied from $\rho = 0.05$ to $\rho = 0.25$ over 50 values. The initial phases $\phi_k(0)$ were drawn independently from a uniform random distribution over the range $[\frac{-\alpha}{2}, \frac{\alpha}{2}]$, where the phase spread, α, varied from 0 to 2π radians over 50 values. At each of the 2500 grid points (i.e., each combination of the 50 values of ρ and α each), the simulation was carried out 100 times. For each simulation, we calculated the graph Laplacian matrix, $\bar{L}_I$, for $\bar{\mathcal{G}}_{I=[0,\frac{2\pi}{\Omega})}$ using the effective sensing regions for each pair of agents. The probability that the sensing topology is connected for each set of conditions was estimated as the fraction of simulations for which the second smallest eigenvalue of $\bar{L}_I$ was greater than 10^{-4} in magnitude. The results of this simulation are shown in Fig. 3.6.

Fig. 3.6 shows a clear increase in the probability that the sensing topology is connected when periodic relative motion is introduced as compared to the static case (random geometric graphs). The $\alpha = 0$ case corresponds to random geometric graphs (i.e., no relative motion). Zero probability of connectivity is represented by black tiles in the figure and a probability of one is represented by white tiles. The solid vertical line represents the critical value $\rho_c = \sqrt{\frac{1.48}{N}} \approx 0.12$ estimated in [5] for random geometric graphs (see (3.7)). For values of ρ above about 0.06, there is a value of α above which the connectivity improves dramatically. The phase spread required for connectedness decreases as the sensing radius increases; the probability of connectedness does not increase significantly for values of ρ much greater than ρ_c. For small values of the sensing radius, there is little increase in connectivity for

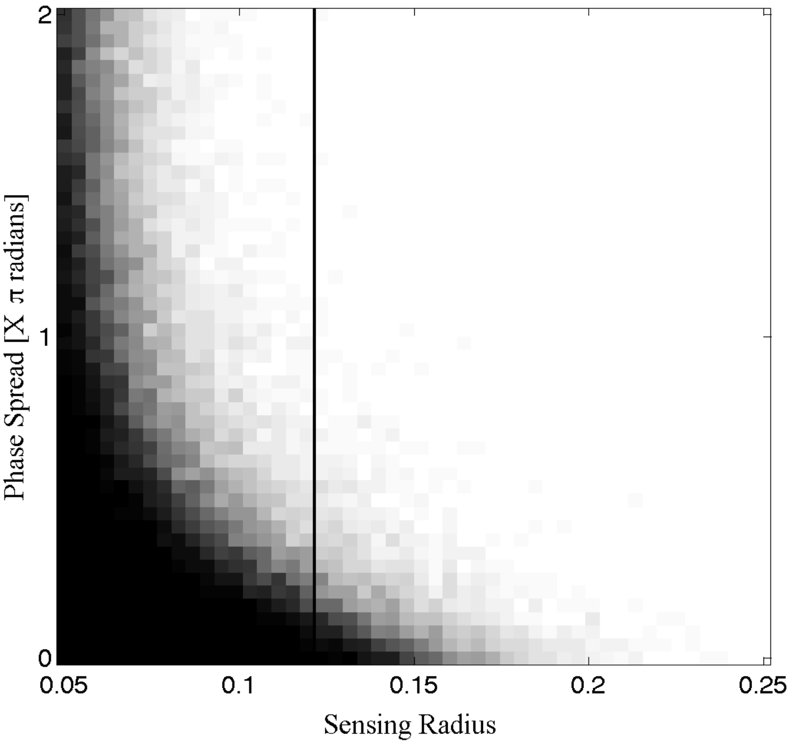


Figure 3.6: Simulation results for $N = 100$ agents distributed uniformly randomly over a unit square showing the probability that the resulting sensing topology is connected. Each agent was engaged in periodic speed oscillations with amplitude $\mu = 0.3$ and frequency $\Omega = 1$, and had a sensing radius ρ. The sensing radius was varied from $\rho = 0.05$ to $\rho = 0.25$ over 50 values. The initial phase of each speed oscillation was drawn independently from a uniform random distribution over the range $[-\frac{\alpha}{2}, \frac{\alpha}{2}]$, such that the total phase spread was α. α was varied from 0 to 2π radians over 50 values. At each grid point, the probability that the resulting graph was connected was calculated. The values range from 0 (never connected, black) to 1 (always connected, white). The solid vertical line represents the critical sensing radius $\rho_c = \sqrt{\frac{1.48}{N}} \approx 0.12$ from the graph percolation literature [5].

values of α above π radians because a phase difference of π radians maximizes the magnitude of relative motion oscillations.

3.3.3 Optimal Relative Motion Parameters for Group-level Decision-Making Performance

In this section, we consider a highly-ordered spatial configuration of $N = 4$ agents undergoing periodic relative motion. The small number and spatial regularity allow us to analytically investigate the optimal parameters of relative motion for group-level decision making. We utilize both the effective sensing regions described above and the convergence rate result of Thm. 3.1 to derive these results.

As in the previous section, we assume that the agents are undergoing coordinated speed oscillations so that their relative positions are given by (3.17). Each agent's average position is evenly distributed along the real axis with the first agent at the origin and each subsequent agent's average position separated from the previous one by a distance a, i.e., $\bar{r}_k = (k-1)a$. See Fig. 3.7. The position of agent k is therefore

$$r_k(t) = (k-1)a + \frac{\mu}{\Omega}\sin\phi_k(t), \tag{3.18}$$

where, as above, $\phi_k(t) = \Omega t + \phi_k(0)$. We assume that

$$\max\left\{\frac{\mu}{\Omega} + \frac{\rho}{2}, \rho\right\} < a < 2\frac{\mu}{\Omega} + \rho, \tag{3.19}$$

which guarantees that

1. constant connectivity of the graph is not possible ($a > \rho$),
2. each agent can become a neighbor of only one of its neighbors to the left and to the right ($a > \frac{\mu}{\Omega} + \frac{\rho}{2}$), and

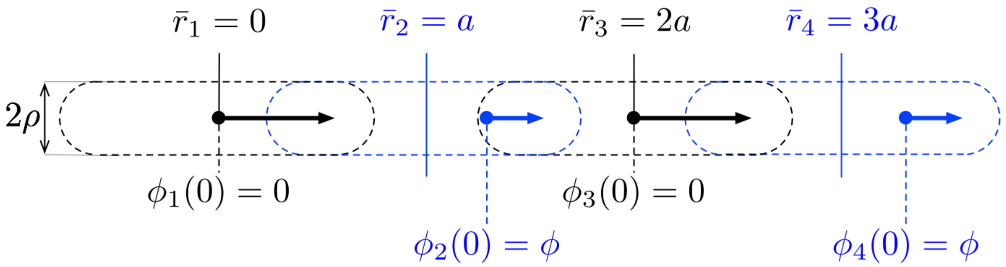


Figure 3.7: Diagram of the ordered spatial configuration used in Sec. 3.3.3. The positions of four agents that are engaged in coordinated speed oscillations are distributed along a line so that their average positions (indicated by the solid vertical lines) are separated by a distance of a and the first one has average position $\bar{r}_1 = 0$. The position of the k^{th} agent (indicated by the solid circle at the arrow base) is described by (3.18). The speed oscillation phases of the first and third agents are synchronized and the speed oscillation phases of the second and fourth agents are synchronized. The speed oscillation phase difference between the first and second agents is ϕ. The sensing radius of each agent is ρ. The effective sensing region of each agent is shown with a dashed outline.

3. connectivity is possible ($a < 2\frac{\mu}{\Omega} + \rho$).

We are motivated by the killifish results above in Sections 2.3.1 and 2.3.2, where we see that killifish tend to coordinate their speed oscillations so that near neighbors are not synchronized. Therefore, we assume that every other agent is synchronized, i.e., $\phi_1(0) = \phi_3(0)$ and $\phi_2(0) = \phi_4(0)$. Without loss of generality, we set $\phi_1(0) = \phi_3(0) = 0$ and $\phi_2(0) = \phi_4(0) = \phi$ for some $0 \leq \phi \leq \pi$. We do not assume that $\phi = \pi$ as in the killifish results; we instead allow ϕ to vary and show below that $\phi = \pi$ optimizes the rate of convergence to consensus.

In order to compute the rate of convergence to consensus over the sensing topology that results from this set up, we need to first determine the associated graphs, their Laplacians, and the time durations of each graph during a period of oscillation. Using the effective sensing region (Fig. 3.5), as in the previous section, we see that there is no connectivity when

$$\left|\sin\frac{\phi}{2}\right| \leq \frac{\Omega}{2\mu}(a-\rho) \tag{3.20}$$

because the amplitude of relative position oscillations is too small. In this case, the sensing topology graph, $\mathcal{G}(t)$, is always the null graph, its Laplacian matrix is the zero matrix, and the convergence rate is 0. When (3.20) is not true, the graph cycles through three component graphs: the graph shown in Fig. 3.8a, which has Laplacian matrix

$$L_1 = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}, \tag{3.21}$$

the null graph, which has Laplacian matrix $L_2 = \mathbf{0}_{N \times N}$, and the graph shown in Fig. 3.8b, which has Laplacian matrix

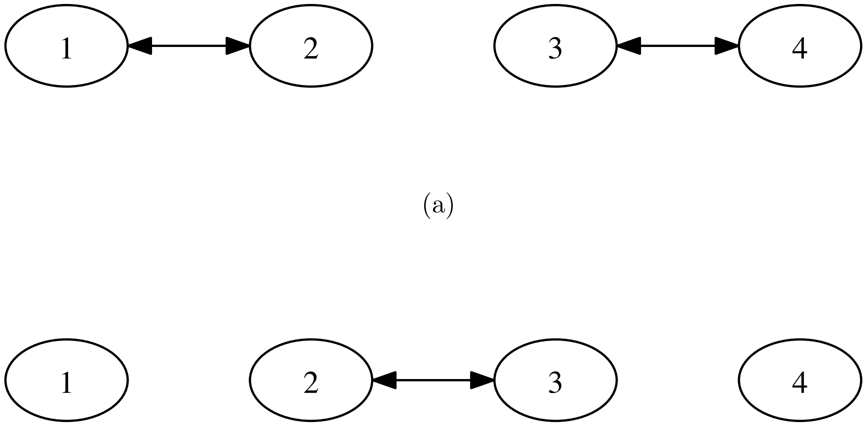
$$L_3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{3.22}$$

All four agents are undergoing periodic relative motion with equal frequency and at different phases. By construction, the phase difference, ϕ, between the first and second agents equals the phase difference between the third and fourth agents. Furthermore, the phase difference between the second and third agents is $2\pi - \phi$. Thus, by (3.18), the relative positions are given by

$$r_{21} = r_{43} = a + 2\frac{\mu}{\Omega} \sin\frac{\phi}{2} \cos\left(\Omega t + \frac{\phi}{2}\right) \tag{3.23}$$

and

$$r_{32} = a + 2\frac{\mu}{\Omega} \sin\frac{2\pi - \phi}{2} \cos\left(\Omega t + \frac{2\pi - \phi}{2}\right) = a - 2\frac{\mu}{\Omega} \sin\frac{\phi}{2} \cos\left(\Omega t - \frac{\phi}{2}\right). \tag{3.24}$$



(a)

(b)

Figure 3.8: The two non-null graphs encountered in the ordered setting described above (see also Fig. 3.7). (a) The graph with Laplacian matrix L_1 in (3.21). (b) The graph with Laplacian matrix L_2 in (3.22).

By symmetry, this implies that $\Delta t_1 = \Delta t_3 = \frac{\Delta\phi}{\Omega}$, where $\Delta\phi$ is the measure of the set

$$\left\{ \varphi \in S^1 : a + 2\frac{\mu}{\Omega}\sin\frac{\phi}{2}\cos\varphi < \rho \right\}. \tag{3.25}$$

Therefore, the durations of the first and third graphs are

$$\Delta t_1 = \Delta t_3 = \frac{2}{\Omega}\cos^{-1}\left(\frac{\Omega\,(a-\rho)}{2\mu}\sin\frac{\phi}{2} \right). \tag{3.26}$$

The remaining portion of the oscillation period is taken up by the null graph, hence $\Delta t_2 = T - 2\Delta t_1$. Note that the null graph occurs twice during each oscillation period, but that the null graph does not explicitly contribute to the computation of the convergence rate because $e^{-L_2\Delta t_2} = I_{N\times N}$.

Computing the eigenvalues of the matrix $M = e^{-L_1\Delta t_1} e^{-L_3\Delta t_3}$ yields

$$m_2 = e^{-2\Delta t_1}\left(\cosh^2\Delta t_1 + \sqrt{\cosh^4\Delta t_1 - 1}\right)$$

as the second-smallest. Therefore, by Thm. 3.1 the convergence rate is

$$\sigma = -\frac{\Omega}{2\pi} \log |m_2| = \frac{\Omega \Delta t_1}{\pi} \log \left(\cosh^2 \Delta t_1 + \sqrt{\cosh^4 \Delta t_1 - 1} \right). \tag{3.27}$$

From this, we arrive at the following result.

Theorem 3.3. *Consider $N = 4$ agents undergoing relative motion as per (3.23) and (3.24) and performing linear consensus dynamics over the group's periodically changing sensing topology. For a given frequency Ω, amplitude μ, separation a, and sensing radius ρ, the rate of convergence to consensus is maximized when the phase separation is $\phi = \pi$ assuming that the parameters satisfy (3.19). That is, the convergence rate is maximized when every other agent is antisynchronized with the agent(s) immediately to its left and/or right.*

Proof. We have shown above that the convergence rate, σ, satisfies (3.27) and is therefore a monotonically increasing function of Δt_1. Therefore, σ is maximized when Δt_1 is maximized. For fixed Ω, μ, a, and ρ, Δt_1 given by (3.26) is maximized when $\phi = \pi$. □

Furthermore, note from (3.27) and (3.26) that the convergence rate depends upon the oscillation frequency in a nontrivial manner. In fact, we have the following.

Theorem 3.4. *Under the same assumptions as Thm. 3.3, the rate of convergence to consensus is maximized for some nontrivial value of the oscillation frequency Ω in the range $\frac{a-\rho}{2\mu} \leq \Omega < 2\frac{\mu}{a-\rho} \sin \frac{\rho}{2}$.*

Proof. We have $\sigma \geq 0$ and

$$\lim_{\Omega \longrightarrow 0^+} \sigma = 0$$

from (3.27). From (3.26), $\Delta t_1 = 0$ when

$$\Omega = 2\frac{\mu}{a-\rho} \sin \frac{\phi}{2},$$

and thus $\sigma = 0$ by (3.27). Therefore, by continuity, $\sigma > 0$ for some value of $0 < \Omega < 2\frac{\mu}{a-\rho}\sin\frac{\phi}{2}$. The lower bound $\Omega \geq \frac{a-\rho}{2\mu}$ gives the value of Ω below which the assumption (3.19) fails. □

Fig. 3.9 shows the relationship between σ and Ω for $a = 1$, $\mu = 0.5$, $\rho = 0.5$, and $\phi = \pi$. The vertical dashed red line indicates the value $\Omega = \frac{a-\rho}{2\mu}$, below which the assumption (3.19) fails. Failure of the assumption (3.19) implies that the amplitude of relative motion is large enough that agent 1 and agent 4 are occasionally connected and therefore the graph Laplacians L_1, L_2, and L_3 no longer accurately describe the sensing topology. For small frequencies, the amount of time between topology changes is very large because the period is very long, and hence convergence is slow. Likewise, for large frequencies, the amplitude of relative motion is small and therefore the portion of a period during which the agents can exchange information (i.e., the portion of a period during which either the L_1 or L_3 graph is active) is very short, and hence the convergence rate is slow. Intermediate values balance these trends and lead to higher convergence rates.

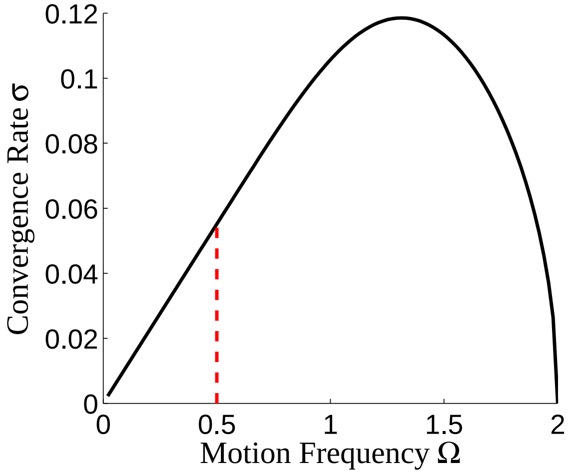


Figure 3.9: Convergence rate, σ, as a function of the oscillation frequency, Ω, illustrating the result of Thm. 3.4 that the convergence rate is maximized for some non-trivial value of Ω. See the text for other parameter values, which were fixed. The vertical dashed red line indicates the value $\Omega = \frac{a-\rho}{2\mu}$, below which the assumption (3.19) fails.

The setup shown in Fig. 3.7 and the plot in Fig. 3.9 illustrate the need to design motion parameters for mobile sensor networks that involve agents with limited sensing radius and periodic relative motion. Patterns like the one shown in Fig. 3.7 can be constructed to represent a variety of situations, and similar analyses to those presented above can be performed. Fig. 3.9 illustrates the benefits of carefully designed motion parameters for situations in which the amplitude of relative motion increases with the period of motion, which is the case for fixed-amplitude speed or acceleration oscillations. When the period is long, the amplitude of motion is high and agents interact for longer periods of time. When the period is shorter, interactions occur more frequently and thus information spreads across the group more quickly. The rate of convergence to consensus is hence maximized for some nontrivial value of the period (equivalently, the frequency) of relative motion.

Chapter 4

Coordinated Speed Oscillations for Engineered Systems

In this chapter, we consider circular formations formed by multiple agents undergoing coordinated speed oscillations. These formations are motivated by our observations of killifish motion, which led us to investigate how coordinated speed oscillations would affect circular motion. The formation motion exhibits regular exchanges of roles among the individuals, which is highly reminiscent of a variety of animal groups, including fish schools and bird flocks. However, we focus our discussion here on the use of this formation motion to meet the requirements of engineered systems. Formations of agents with periodic speed profiles exhibit properties that could be very useful for mobile sensor networks. See, for example, Fig. 4.9a, which shows a formation of four agents with periodic speed profiles. Because the agents maintain their formation as they move around the larger circle, they can provide redundant measurements in a small region and they can combine measurements to produce estimates of local gradients of spatially-varying quantities. In this chapter, we develop control laws that stabilize a group of autonomous agents that are engaged in coordinated speed oscillations into formations and then explore how the parameters of motion and the control laws can be designed to achieve specific desired formations.

We begin in Sec. 4.1 by providing some necessary background. In Sec. 4.1.1, we describe a steered particle model that we use to model the motion of agents moving on the plane with time-varying speed. In Sec. 4.1.2, we discuss some notions relating to the control and description of groups of phase angles (i.e., coupled oscillators). Specifically, we discuss *phase arrangements*, which provide a means to precisely describe patterns of phases, and *phase potentials*, which are a class of functions that are useful when designing control laws to stabilize particular phase arrangements.

In Sec. 4.2, we present several results that allow us to systematically describe the open-loop trajectories of steered particles with time-varying speed and constant turning rates. We construct the solution by first considering constant speed motion with a constant turning rate, which results in circular motion. The special case of sinusoidal speed is considered in detail because it provides a natural means to expand to the general periodic speed case via Fourier decomposition. We also provide results on the boundedness and periodicity of trajectories of steered particles with time-varying speeds and constant turning rates.

In Sec. 4.3, we prove stability of a feedback control law that can be used to stabilize a wide variety of formations of steered particles with time-varying speeds. In steady-state, the trajectories are described by those that we discuss in Sec. 4.2.

We discuss in Sec. 4.4 how the parameters of motion can be designed to obtain specific formations. First, in Sec. 4.4.1, we consider what kinds of formations are achievable by choosing various combinations of control parameters to obtain specific phase arrangements and spatial distributions. We give focus to formations in which the headings are synchronized and the speed phases are evenly distributed, because this results in all agents moving synchronously around a circle while maintaining a formation whose shape is tied directly to the speed profile. Then, in Sec. 4.4.2, we provide methods to determine what speed profile is necessary to obtain a given formation shape.

The case of purely sinusoidal speed oscillations first appeared in [117], which was itself strongly motivated by the work of Sepulchre, Paley, and Leonard [104] on stabilizing planar circular formations of constant-speed steered particles. In [116], we extended the results for sinusoidal speed profiles to more general periodic speed profiles and investigated boundedness of the trajectories. We expand on those results here and provide means to design specific formations in a systematic way.

4.1 Background

4.1.1 Steered Particle Model

For convenience, we identify the real and complex planes; i.e., $\mathbb{R}^2 \sim \mathbb{C}$. We make use of the inner product

$$\langle z_1, z_2 \rangle \triangleq \operatorname{Re}\{z_1^* z_2\}$$

for complex scalars $z_1,\ z_2 \in \mathbb{C}$ and where z_1^* is the complex conjugate of z_1. For complex vectors $\mathbf{z}_1$ and $\mathbf{z}_2 \in \mathbb{C}^N$, we use the inner product

$$\langle \mathbf{z}_1, \mathbf{z}_2 \rangle \triangleq \operatorname{Re}\left\{\mathbf{z}_1^H \mathbf{z}_2\right\}$$

where $\mathbf{z}_1^H$ is the complex conjugate transpose of $\mathbf{z}_1$.

Consider N unit-mass particles moving in the plane. The position of the k^{th} particle is $r_k \in \mathbb{C}$. Each particle's speed is assumed to be nonzero for all time and hence there is no ambiguity in expressing its velocity in polar coordinates as

$$\dot{r}_k = \alpha_k e^{i\theta_k}, \tag{4.1}$$

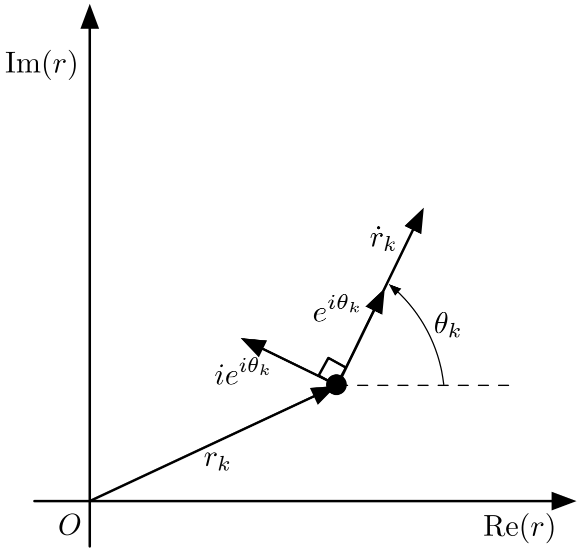


Figure 4.1: Illustration of the steered particle model. The position of the k^{th} particle is $r_k \in \mathbb{C} \sim \mathbb{R}^2$ and its velocity is $\dot{r}_k$. The velocity forms an angle θ_k with the real axis, and $\left\{e^{i\theta_k},\ ie^{i\theta_k}\right\}$ forms an orthonormal basis for a coordinate frame that is oriented with the velocity.

where $\alpha_k = |\dot{r}_k| > 0$ is the speed and $\theta_k \in S^1$ is the heading with respect to the real axis. The pair $\left\{e^{i\theta_k},\ ie^{i\theta_k}\right\}$ forms a basis for a coordinate frame oriented with particle k's velocity. Fig. 4.1 illustrates the geometry of the steered particle model.

We consider here the case in which the speed of the k^{th} particle is time-varying. We are primarily concerned with speeds that are 2π-periodic functions of some variable $\phi_k \in S^1$, i.e., $\alpha_k = \alpha_k(\phi_k)$ and $\alpha_k(\phi_k + 2\pi) = \alpha_k(\phi_k),\ \forall\ \phi_k \in S^1$. We refer to the mapping $\alpha_k : S^1 \longrightarrow \mathbb{R}^+$ as the *speed profile* for particle k, and refer to ϕ_k as the *speed phase* of particle k (see also Sec. 2.2.1). The speed phase is, in general, a function of time: $\phi_k = \phi_k(t)$.

We assume that each particle has steering control

$$\dot{\theta}_k = \bar{u}_k$$

and speed phase control

$$\dot{\phi}_k = \bar{g}_k.$$

We also assume that the speed profile is a smooth function of the speed phase and therefore the speed control, $\dot{\alpha}_k$, can be computed from the speed phase control via the mapping

$$\dot{\alpha}_k = \frac{\partial \alpha_k}{\partial \phi_k} \bar{g}_k.$$

The speed phase and heading dynamics are assumed to contain constant components Ω and ω, respectively. We therefore write

$$\dot{\phi}_k = \bar{g}_k = \Omega + g_k, \tag{4.2}$$

$$\dot{\theta}_k = \bar{u}_k = \omega + u_k, \tag{4.3}$$

and hereafter refer to g_k and u_k as the speed phase control and heading control, respectively. Because θ_k and ϕ_k both evolve on the circle, we refer to ω and Ω as their respective *natural frequencies*. We also refer to ω as the *steering rate* and to Ω as the *speed oscillation frequency*. Note that in steady-state, $g_k = 0$ and we have $\dot{\phi}_k = \Omega$ and there is no ambiguity in referring to α_k as the speed profile.

Rewriting (4.1) to emphasize the functional dependence on the speed phase gives

$$\dot{r}_k = \alpha_k(\phi_k) e^{i\theta_k}. \tag{4.4}$$

The velocity (4.4) along with the speed phase dynamics (4.2) and heading dynamics (4.3) make up the steered particle dynamics that we will use throughout this chapter.

By taking the derivative of (4.4) with respect to time we get

$$\ddot{r}_k = \frac{\partial \alpha_k}{\partial \phi_k} e^{i\theta_k} \bar{g}_k + \alpha_k(\phi_k) i e^{i\theta_k} \bar{u}_k.$$

Thus, we can see that $\bar{g}_k$ controls the component of acceleration of particle k in the direction of motion and $\bar{u}_k$ controls the component of acceleration of particle k orthogonal to the direction of motion.

4.1.2 Phase Potentials and Phase Arrangements

We review here some notions from the study of coupled oscillators. Both the headings and speed phases evolve over the $N-$torus, $T^N = \{S^1 \times S^1 \times \cdots \times S^1\}$ (N times). Specifically, we discuss *phase potentials* and *phase arrangements*. In the following, we will denote the set of headings as $\boldsymbol{\theta} = \begin{pmatrix} \theta_1 & \cdots \theta_N \end{pmatrix}^T \in T^N$ and use them for illustration. These concepts apply equally, however, to the speed phases, $\boldsymbol{\phi} = \begin{pmatrix} \phi_1 & \cdots & \phi_N \end{pmatrix}^T \in T^N$. All operations on phases below are considered on the topology of the circle, S^1. That is, we identify 0 with 2π so that, for example, $\theta_k = \theta_j$ means that the phasors are equal, i.e., $e^{i\theta_k} = e^{i\theta_j}$.

We refer to a description of the relationship amongst phases on T^N as a *phase arrangement*. The following are three important phase arrangements that we will refer to below.

- *Synchronization:* All phases are equal, i.e., $\theta_1 = \theta_2 = \ldots = \theta_N$.
- *Balanced:* The phasor sum is zero, i.e, $\sum_{k=1}^{N} e^{i\theta_k} = 0$.
- *Splay:* The phases are evenly distributed around the circle, separated by arcs of $\frac{2\pi}{N}$ radians.

Note that the splay phase arrangement is a special case of the balanced phase arrangement, and that all three phase arrangements are invariant to a re-indexing of the phases.

The following class of functions is useful for designing control laws that stabilize coupled oscillators to particular phase arrangements.

Definition 4.1. *(Phase Potentials) Consider the class $\mathcal{P}$ of C^2 functions on T^N defined such that, for any function $U(\boldsymbol{\theta}) \in \mathcal{P}$,*

1. $U : T^N \to [0, U_{max}]$ *for some scalar* $U_{max} > 0$.
2. $\text{div } U(\boldsymbol{\theta}) = \langle \nabla_{\boldsymbol{\theta}} U, \mathbf{1}_N \rangle = \sum_{k=1}^{N} \frac{\partial U}{\partial \theta_k} = 0$.

An element of $\mathcal{P}$ *is called a* phase potential.

A group of coupled oscillators can be driven to a particular phase arrangement by designing a phase potential for which the critical sets correspond to the desired phase arrangement, and then using the gradient of the potential as a control law that stabilizes the closed-loop dynamics about those critical sets. An important example is the phase potential defined by the squared magnitude of the *complex order parameter* (introduced by Kuramoto [66], see also the review by Strogatz [110]). The complex order parameter for $\boldsymbol{\theta}$, $p_{\boldsymbol{\theta}}$, is defined as

$$p_{\boldsymbol{\theta}} \triangleq \frac{1}{N} \sum_{k=1}^{N} e^{i\theta_k}, \tag{4.5}$$

and the corresponding phase potential is

$$U_p(\boldsymbol{\theta}) \triangleq \frac{N}{2} \left| p_{\boldsymbol{\theta}} \right|^2 . \tag{4.6}$$

It can be shown that the critical sets of U_p correspond to synchronization, for which $U_p = \frac{N}{2}$, and balancing, for which $U_p = 0$ [104]. Since $0 < |p_{\boldsymbol{\theta}}| < 1$, these also define the maximum and minimum values of U_p, thereby satisfying condition 1 of Def. 4.1 with $U_{max} = \frac{N}{2}$. The k^{th} element of the gradient is given by

$$\begin{aligned}
(\nabla_{\boldsymbol{\theta}} U_p)_k = \frac{\partial U_p}{\partial \theta_k} &= N \left\langle p_{\boldsymbol{\theta}}, \frac{\partial}{\partial \theta_k} p_{\boldsymbol{\theta}} \right\rangle \\
&= \mathrm{Re} \left\{ i \sum_{j=1}^{N} e^{i(\theta_k - \theta_j)} \right\} = \sum_{j=1}^{N} \sin \theta_{jk},
\end{aligned}$$

where $\theta_{jk} = \theta_j - \theta_k$. Therefore, U_p is a valid phase potential because

$$\langle \nabla_{\boldsymbol{\theta}} U_p, \mathbf{1}_N \rangle = \sum_{k=1}^{N} \sum_{j=1}^{N} \sin \theta_{jk} = 0$$

satisfies condition 2 of Def. 4.1. Applying the gradient control

$$\dot{\theta}_k = \kappa_\theta \frac{\partial U_p}{\partial \theta_k} = \kappa_\theta \sum_{j=1}^{N} \sin\theta_{jk},$$

where κ_θ is a non-zero scalar constant, has been shown to stabilize the closed-loop dynamics to the critical points of U_p (see, for example, [104]). When $\kappa_\theta > 0$, balancing is stabilized. When $\kappa_\theta < 0$, synchronization is stabilized. These dynamics are an instance of the well-known Kuramoto model [66, 110].

We also use the phase potential

$$V(\boldsymbol{\theta}) = \sum_{m=1}^{\lfloor \frac{N}{2} \rfloor} \left| \frac{1}{mN} \sum_{j=1}^{N} e^{im\theta_j} \right|^2 \tag{4.7}$$

below. This phase potential was shown by Sepulchre et al. [104] to have the splay phase arrangement as its minimizing critical set.

The following results guarantee that a phase arrangement corresponding to the critical points of a phase potential is rotationally invariant. These results are corollaries of Def. 4.1.

Corollary 4.1. *Consider a phase potential $U(\boldsymbol{\theta}) \in \mathcal{P}$. If $\dot{\theta}_k = \omega$ for some ω and for each $k = 1, \ldots, N$, then*

$$\frac{d}{dt} U(\boldsymbol{\theta}) = 0 \tag{4.8}$$

and

$$\frac{d}{dt} \frac{\partial U}{\partial \theta_k} = 0. \tag{4.9}$$

Proof. From the definition of a phase potential, when $\dot{\theta}_k = \omega$ for each k,

$$\frac{d}{dt} U(\boldsymbol{\theta}) = \sum_{k=1}^{N} \frac{\partial U}{\partial \theta_k} \dot{\theta}_k = \langle \nabla_{\boldsymbol{\theta}} U, \mathbf{1}_N \rangle \, \omega = 0.$$

Similarly,

$$\frac{d}{dt}\frac{\partial U}{\partial \theta_k} = \sum_{j=1}^{N} \frac{\partial}{\partial \theta_j}\frac{\partial U}{\partial \theta_k}\dot{\theta}_j = \frac{\partial}{\partial \theta_k}\sum_{j=1}^{N}\frac{\partial U}{\partial \theta_j}\omega = 0,$$

where we are able to swap the order of partial derivatives because U is C^2 by definition.

□

Corollary 4.2. *(Rotational invariance of phase potentials) Consider a phase potential* $U(\boldsymbol{\theta})$. *U is rotationally invariant. That is,* $U(\boldsymbol{\theta}^* + \bar{\theta}\mathbf{1}_N) = U(\boldsymbol{\theta}^*)$ *for any* $\boldsymbol{\theta}^* \in T^N$ *and* $\bar{\theta} \in S^1$.

Proof. Consider the value of U along a trajectory of $\boldsymbol{\theta}$ defined by $\boldsymbol{\theta} = \boldsymbol{\theta}^* + \omega t\mathbf{1}_N$ for some $\omega \in \mathbb{R}$ and $t \geq 0$. By Corr. 4.1, we have

$$\left.\frac{d}{dt}U(\boldsymbol{\theta})\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*+\omega t\mathbf{1}_N} = \sum_{k=1}^{N}\frac{\partial U}{\partial \theta_k}\omega = 0.$$

Therefore, the value of U is constant along such trajectory. In particular this is true when $\omega t = \bar{\theta}$. □

4.2 Trajectories of Steered Particles with Periodic Speed Profiles

We describe here the trajectories of particles evolving under the steered particle model described in Sec. 4.1.1 with no control inputs, i.e. $u_k = g_k = 0$. $\omega = 0$ corresponds to straight-line motion and is important for translational formations. We focus, however, on $\omega \neq 0$, which produces circular motion under most conditions (although we will show that this is not always the case). The subscript notation is maintained so that the results here may be easily referenced below when multiple particles are involved.

The trajectory of a particle moving at constant speed, $\alpha_k = \nu_k$, where $\nu_k > 0$ is constant, and with a constant turning rate, $\omega \neq 0$, traces a circle of radius $\nu_k \left|\omega\right|^{-1}$. The following theorem makes this precise.

Theorem 4.1. *(Trajectory of a constant-speed steered particle) Consider a steered particle moving with velocity (4.1) and heading dynamics (4.3), with no steering input ($u_k = 0$), constant speed $\alpha_k = \nu_k$, and constant turning rate $\omega \neq 0$. The trajectory, $r_k(t)$, of this particle is described by motion around a circle of radius $\nu_k \left|\omega\right|^{-1}$ centered at $\bar{c}_k = r_k(0) + i\nu_k \left|\omega\right|^{-1} e^{i\theta_k(0)}$, where $r_k(0)$ and $\theta_k(0)$ are the initial position and heading, respectively. If $\omega > 0$, the motion is counter-clockwise. If $\omega < 0$, the motion is clockwise. The trajectory is written as*

$$r_k = \bar{c}_k + R_k(\theta_k), \tag{4.10}$$

where $R_k(\theta_k)$ is defined as

$$R_k(\theta_k) \triangleq -i\frac{\nu_k}{\omega}e^{i\theta_k(0)} + \int_0^t \nu_k e^{i\theta_k(\tau)}d\tau = -i\frac{\nu_k}{\omega}e^{i\theta_k} \tag{4.11}$$

Proof. Because $\dot{\theta}_k = \omega$ is constant, $\theta_k(t) = \omega t + \theta_k(0)$. Integrating the velocity gives the position:

$$\begin{aligned} r_k(t) &= r_k(0) + \int_0^t \nu_k e^{i\theta_k(\tau)}d\tau \\ &= r_k(0) + R_k(\theta_k(t)) - R_k(\theta_k(0)) \\ &= r_k(0) + i\frac{\nu_k}{\omega}e^{i\theta_k(0)} + R_k(\theta_k). \end{aligned}$$

This describes the geometry of a particle moving around a circle of radius $\nu_k \left|\omega\right|^{-1}$ with center $\bar{c}_k = r_k(0) - R_k(\theta_k(0))$. When $\omega > 0$, θ_k is increasing and the motion is counter-clockwise. Likewise, when $\omega < 0$, the motion is clockwise. □

When $\Omega \neq 0$ and $g_k = 0$ (i.e., there is no speed phase control), the speed profile, $\alpha_k(\phi_k(t))$, is $\frac{2\pi}{\Omega}-$periodic in time. We decompose the speed profile as

$$\alpha_k(\phi_k(t)) = \nu_k + v_k(\phi_k(t)),$$

where

$$\nu_k \triangleq \frac{1}{2\pi}\int_0^{2\pi} \alpha_k(\phi_k) d\phi_k$$

is the *nominal* or *average* speed and

$$v_k(\phi_k(t)) \triangleq \alpha_k(\phi_k(t)) - \nu_k \tag{4.12}$$

is the *speed variation.* Note that, by definition, the average value of $v_k(\phi_k(t))$ is zero and $v_k(\phi_k(t)) > -\nu_k, \ \forall t$. Purely sinusoidal speed is an important example:

$$\alpha_k(\phi_k(t)) = \nu_k + \mu_k \cos\phi_k(t) \tag{4.13}$$

where μ_k is the amplitude and $\phi_k(t) = \Omega t + \phi_k(0)$.

The following result gives boundedness of the trajectory of a steered particle with speed profile $\nu_k + v_k(\phi_k(t))$ in terms of the spectrum of $v_k(\phi_k(t))$, and is useful in deriving general trajectory solutions. This result is specialized below for periodic trajectories.

Theorem 4.2. *(Boundedness of trajectories of variable-speed particles) Consider a steered particle with constant turning rate* $\omega \neq 0$ *and speed* $\alpha_k(\phi_k(t)) = \nu_k + v_k(t)$ *where* ν_k *is a constant and* $v_k(\phi_k(t)) > -\nu_k$*. The trajectory,* $r_k(t)$*, of such a particle is bounded if and only if its speed variation,* $v_k(\phi_k(t))$*, is bounded and contains no time-periodic components with frequency equal to the turning rate,* ω*. That is,* $|v_k| < \infty$

and

$$\left|\int_0^\infty v_k(\phi_k(t))e^{-st}dt\right|_{s=i\omega} < \infty. \tag{4.14}$$

Proof. Consider the trajectory of a steered particle with constant speed equal to ν_k and with the same heading, turning rate, and initial position as the variable-speed particle. By Thm. 4.1, the trajectory of the constant-speed particle is circular and bounded. Therefore, the trajectory of the variable-speed particle is bounded if its distance from the constant-speed trajectory is bounded. Let $\bar{r}_k$ be the position of the constant-speed particle. The distance between the two particles at time t is

$$|r_k(t) - \bar{r}_k| = \left|\int_0^t \left(\dot{r}_k - \nu_k e^{i\theta_k}\right) dt\right| = \left|\int_0^t v_k e^{i\theta_k} dt\right|.$$

If v_k is unbounded, then it is unbounded periodically, and therefore $|r_k(t) - r_k|$ is unbounded for any $t > \frac{2\pi}{\Omega}$. If v_k is bounded, then the integral (and hence the trajectory) can only grow unbounded in the limit $t \to \infty$. Because the turning rate is constant, $\theta_k = \omega t + \theta_k(0)$, and we have

$$|r_k(t) - \bar{r}_k| = \left|\int_0^t v_k e^{i(\omega t + \theta_k(0))} dt + r_k(0) - \bar{r}_k(0)\right| = |V_k(s)|_{s=-i\omega}$$

where $V_k(s)$ is the Laplace transform of $v_k(\phi_k(t))$. Because $v_k(t)$ is real, $|V_k(s)|_{s=-i\omega} = |V_k(s)|_{s=i\omega}$. Without loss of generality, we assume that $v_k(\phi_k(t)) = 0$ for $t < 0$, and therefore $|V_k(s)|_{s=i\omega}$ is the magnitude of the Fourier spectrum of v_k at the frequency ω. The magnitude of the Fourier spectrum is finite at ω if and only if $v_k(\phi_k(t))$ has no periodic components at frequency ω. □

Fig. 4.2 shows an example of an unbounded trajectory obtained when the speed is purely sinusoidal as in (4.13) and the speed oscillation frequency, Ω, is equal to the turning rate, ω. The result is a spiral trajectory.

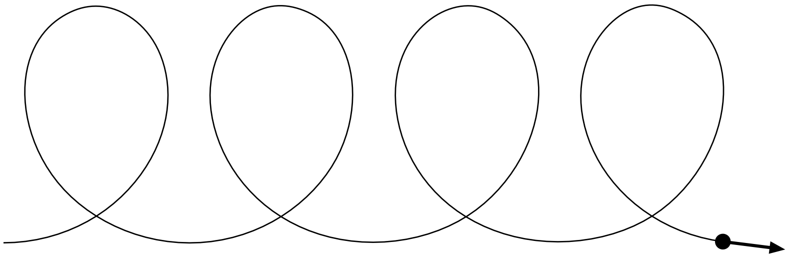


Figure 4.2: Unbounded spiral trajectory obtained when the speed is purely sinusoidal as per (4.13) and the speed oscillation frequency, Ω, is equal to the turning rate, ω. The parameter values are $\Omega = \omega = 1$, $\nu_k = 1$, $\mu_k = 0.75$. The beginning of the trajectory is at the bottom left and the end is at the bottom right.

The following results establish general expressions for the trajectory of a steered particle with time-varying speed in terms of a decomposition similar to (4.10). The first result characterizes an important quantity, q_k, that we use to describe trajectories.

Lemma 4.1. *Consider a steered particle with constant turning rate $\omega \neq 0$ and speed profile $\alpha_k(\phi_k(t)) = \nu_k + v_k(\phi_k(t))$ with $\dot{\phi}_k = \Omega$ and satisfying the conditions of Thm. 4.2 for boundedness of the trajectory. Define $q_k : S^1 \to \mathbb{C}$ as the solution to*

$$\Omega \frac{\partial q_k}{\partial \phi_k} + i\omega q_k = v_k, \tag{4.15}$$

with initial condition

$$q_k(\phi_k(0)) = - \left. V_k(s)\right|_{s=-i\omega}, \tag{4.16}$$

where $V_k(s)$ is the Laplace transform of $v_k(\phi_k(t))$. Then q_k satisfies

$$q_k(\phi_k(t)) = e^{-i\theta_k(t)} \left(q_k(\phi_k(0)) e^{i\theta_k(0)} + \int_0^t v_k(\phi_k(\tau)) e^{i\theta_k(\tau)} d\tau \right), \tag{4.17}$$

when $\dot{\theta}_k = \omega$. Additionally, $q_k(\phi_k(t))$ contains no time-periodic components at the frequency ω; that is,

$$\lim_{s \to -i\omega} (s + i\omega) Q_k(s) = 0,$$

where $Q_k(s)$ is the Laplace transform of $q_k(\phi_k(t))$.

Proof. Differentiating (4.17) with respect to t when $\dot{\theta}_k = \omega$ gives

$$\begin{aligned}\frac{d}{dt}q_k(\phi_k(t)) &= -i\omega e^{-i\theta_k}\left(q_k(\phi_k(0))e^{i\theta_k(0)} + \int_0^t v_k(\phi_k(\tau))e^{i\theta_k(\tau)}d\tau\right) + e^{-i\theta_k}v_k e^{i\theta_k} \\ &= -i\omega q_k(\phi_k(t)) + v_k(\phi_k(t)).\end{aligned}$$

Because $\dot{\phi}_k = \Omega$, we have

$$\frac{d}{dt}q_k(\phi_k(t)) = \Omega\frac{\partial q_k}{\partial \phi_k} = -i\omega q_k + v_k.$$

Taking the Laplace transform of both sides of (4.15) and solving for $Q_k(s)$ yields

$$Q_k(s) = \frac{V_k(s) + q_k(\phi_k(0))}{s + i\omega}.$$

By Thm. 4.2, $V_k(-i\omega)$ is finite. Therefore,

$$\lim_{s\to -i\omega}(s + i\omega)Q_k(s) = V_k(-i\omega) + q_k(\phi_k(0)) = 0,$$

by choice of initial conditions (4.16). If $q_k(\phi_k(t))$ contained a periodic component at the frequency ω due to the natural response $e^{-i\omega t}$, then we would have $\lim_{s\to -i\omega}(s + i\omega)Q_k(s) \neq 0$. □

The following result shows how we use Thm. 4.1 and Lemma 4.1 to describe the geometric properties of the trajectory of a steered particle with a time-varying speed profile.

Theorem 4.3. *(Trajectory of a steered particle with time-varying speed) Consider a steered particle moving with the dynamics (4.1) and (4.3), with initial position $r_k(0)$, initial heading $\theta_k(0)$, $q_k(\phi_k(0))$ determined from v_k from (4.16), no steering input*

($u_k = 0$) or speed phase input ($g_k = 0$), speed $\alpha_k(\phi_k(t)) = \nu_k + \upsilon_k(\phi_k(t)) > 0$ where ν_k is constant and υ_k is bounded, and turning rate $\omega \neq 0$. The trajectory of the particle is

$$r_k = c_k + R_k(\theta_k) + q_k e^{i\theta_k}, \tag{4.18}$$

where $R_k(\theta_k)$ is defined by (4.11), q_k is defined by Lemma 4.1, and the constant $c_k \triangleq r_k(0) - R_k(\theta_k(0)) - q_k(\phi_k(0))e^{i\theta_k(0)} \in \mathbb{C}$ is defined as the center *of the trajectory.*

Proof. Consider a second particle with initial position $r_k(0)$ and initial heading $\theta_k(0)$. The second particle has constant speed equal to ν_k. Let $\bar{r}_k(t)$ be the trajectory of the constant-speed particle, which is given by (4.10). The velocity of the variable-speed particle relative to the constant-speed particle is

$$\frac{d}{dt}(r_k - \bar{r}_k) = \upsilon_k e^{i\theta_k},$$

and hence

$$r_k - \bar{r}_k = q_k e^{i\theta_k} - q_k(\phi_k(0))e^{i\theta_k(0)}$$

by the definition of q_k (4.17). The full trajectory is therefore

$$r_k = \bar{c}_k + R_k(\theta_k) + q_k e^{i\theta_k} - q_k(\phi_k(0))e^{i\theta_k(0)}$$

and is equivalent to (4.18) by setting $c_k = \bar{c}_k - q_k(\phi_k(0))e^{i\theta_k(0)}$. □

Writing

$$r_k = c_k + R_k(\theta_k) + \mathrm{Re}\{q_k\} e^{i\theta_k} + \mathrm{Im}\{q_k\} ie^{i\theta_k} \tag{4.19}$$

motivates the interpretation that q_k expresses the motion of the variable-speed particle relative to a body-centered and velocity-oriented coordinate frame of a constant-speed particle sharing the same center, turning rate, and nominal speed. Fig. 4.3 shows this geometry. Determining the properties of q_k in terms of υ_k is therefore an important step towards describing the dynamic geometry of the trajectories of time-varying speed particles with a constant turning rate.

We focus here on periodic speed profiles, for which we show that there is a direct correspondence between the frequency content of the speed profile and the geometric properties of the resulting trajectories. When v_k is periodic, q_k must also be periodic and hence the trajectory that q_k traces out in the complex plane is a closed curve.

A periodic speed profile can be expressed in terms of its Fourier expansion, hence we establish the following notation. When the speed profile is time-periodic, then v_k is time-periodic and we may write v_k as a cosine Fourier series,

$$v_k(\phi_k) = \sum_{\ell=1}^{\infty} \mu_{k,\ell} \cos\left(\ell\phi_k - \varphi_{k,\ell}\right), \ t \geq 0, \tag{4.20}$$

where $\mu_{k,\ell} \geq 0$ and $\varphi_{k,\ell} \in S^1$ are constants. It is also sometimes convenient to write v_k as a sine and cosine Fourier series,

$$v_k(\phi_k) = \sum_{\ell=1}^{\infty} \left(a_{k,\ell} \cos \ell\phi_k + b_{k,\ell} \sin \ell\phi_k\right), \ t \geq 0, \tag{4.21}$$

where $a_{k,\ell}$ and $b_{k,\ell}$ are real constants. When all particles share a common speed profile, we drop the k subscript and write simply μ_ℓ, φ_ℓ, a_ℓ, and b_ℓ. By trigonometry, the constants in (4.21) are related to those in (4.20) as $a_{k,\ell} = \mu_{k,\ell}\cos\left(\varphi_{k,\ell}\right)$ and $b_{k,\ell} = \mu_{k,\ell}\sin\left(\varphi_{k,\ell}\right)$. The $\ell = 0$ term is omitted because v_k has zero mean by (4.12).

Thm. 4.2 implies the following result for boundedness of trajectories resulting from periodic speed profiles and constant turning rate.

Corollary 4.3. *(Boundedness of trajectories of periodic-speed particles) The trajectory of a steered particle satisfying the conditions of Thm. 4.2 and with time-periodic speed profile* $\alpha_k(\phi_k(t))$ *with period* $\frac{2\pi}{\Omega}$ *is bounded if and only if the speed variation* $v_k(\phi_k(t))$ *contains no harmonics at the frequency* $|\omega|$*, equivalently,* $|V_k(s)|_{s=i\omega}$ *is fi-*

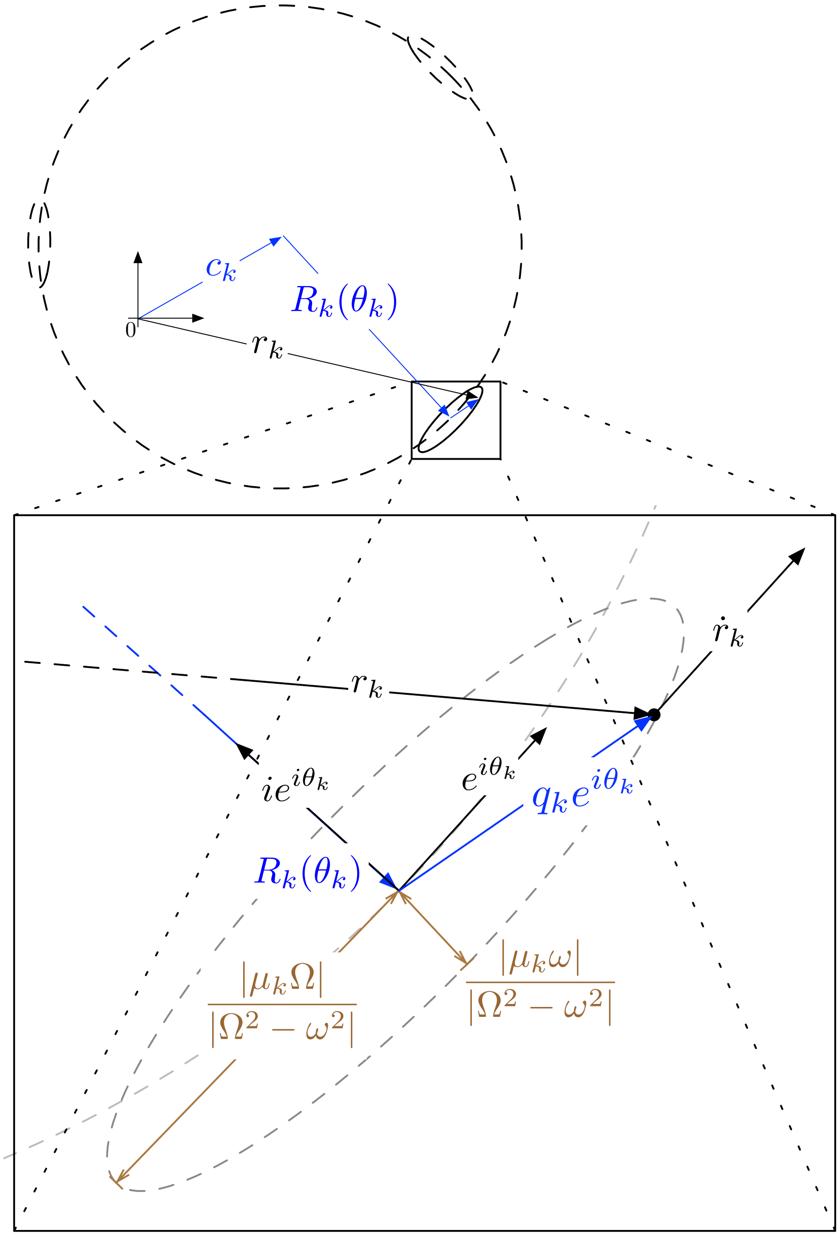


Figure 4.3: Illustration of the trajectory decomposition (4.18) for purely sinusoidal speed (see Lemma 4.4). c_k gives the center of the trajectory, $R_k(\theta_k)$ gives the circular component, and q_k gives the component due to speed modulation, where $\mathrm{Re}\{q_k\}$ is along the direction of motion ($e^{i\theta_k}$) and $\mathrm{Im}\{q_k\}$ is perpendicular to it. For sinusoidal speed, the locus of q_k is an ellipse whose bounds depend on the amplitude μ_k and the frequencies ω and Ω.

nite, where $V_k(s)$ is the Laplace transform of $v_k(\phi_k(t))$. A sufficient condition for boundedness is that either $\Omega > |\omega|$ or there is no integer ℓ such that $\ell\Omega = \omega$.

Proof. Because v_k is periodic, its Fourier spectrum magnitude is finite everywhere except possibly at integer multiples of the base frequency, Ω. If there is no integer ℓ for which $\ell\Omega = \omega$ (note here that ℓ could be negative if Ω and ω have opposite signs), the Fourier spectrum magnitude of v_k must be bounded at $\pm\omega$ and by Thm. 4.2 the trajectory must be bounded. As a consequence, if $\Omega > |\omega|$ the trajectory must be bounded because there can be no integer ℓ such that $\ell\Omega = \omega$. If $\Omega < |\omega|$ and there is an integer ℓ for which $\ell\Omega = \omega$, the trajectory may still be bounded if v_k has no harmonic component at $|\ell|\Omega$, i.e. $\mu_{k,|\ell|}$ in (4.20) is zero, in which case $|V_k(s)|_{s=\pm i\omega}$ is finite. □

See Fig. 4.2 for an example of an unbounded trajectory for which $\Omega = \omega$.

We now establish conditions under which a periodic trajectory exists.

Theorem 4.4. *(Existence of periodic trajectories) The trajectory of a steered particle with constant turning rate $\omega \neq 0$ and time-periodic speed profile with period $\frac{2\pi}{\Omega}$ is periodic if and only if there exist integers ℓ and m for which $\ell\Omega = m|\omega|$ and v_k has no harmonic components at the frequency $|\omega|$.*

Proof. The trajectory is periodic if and only if there is a $T > 0$ such that $r_k(t+T) - r_k(t) = 0$ for all $t \geq 0$. Substituting from (4.18) and (4.11), we have

$$r_k(t+T) - r_k(t) = \left(q_k(\phi_k(t+T)) - i\frac{\nu_k}{\omega}\right) e^{i\theta_k(t+T)} - \left(q_k(\phi_k(t)) - i\frac{\nu_k}{\omega}\right) e^{i\theta_k(t)}.$$

Therefore the trajectory is periodic if and only if both q_k is time-periodic with some period T_q, $e^{i\theta_k}$ is periodic with some period T_R, and there are integers ℓ and m such that $\ell T_R = mT_q$ (i.e., the periods share a common multiple). Because q_k is the solution of the linear ordinary differential equation (4.15) driven by v_k, which is by assumption time-periodic with a frequency Ω, q_k must also be time-periodic with a frequency Ω. $e^{i\theta_k}$ is time-periodic with a frequency of $|\omega|$. Hence, $\ell T_R = \ell \frac{2\pi}{|\omega|}$ and $mT_q = m\frac{2\pi}{\Omega}$ and therefore the condition for periodicity is equivalent to the existence of integers ℓ and m such that $\ell\Omega = m|\omega|$. The boundedness condition follows from Corr. 4.3. $\square$

It follows from Thm. 4.4 that, if $\frac{\Omega}{|\omega|}$ is rational, then the resulting trajectory is periodic. Furthermore, the period is given by $T = m\frac{2\pi}{\Omega} = \ell\frac{2\pi}{|\omega|}$ where $\frac{\Omega}{|\omega|} = \frac{m}{\ell}$ is irreducible.

We now give a solution to (4.18) for periodic speed profiles where q_k is expressed in terms of the Fourier series coefficients of v_k.

Theorem 4.5. *(Trajectory solution for periodic speed profiles) Consider a steered particle with constant turning rate $\omega \neq 0$ and speed oscillation frequency $\Omega > 0$ such that the speed profile is time-periodic with period $\frac{2\pi}{\Omega}$ and its Fourier series is given by (4.20). Assume that there is no integer ℓ such that both $\ell\Omega = |\omega|$ and $\mu_{k,\ell} \neq 0$, so that the trajectory is bounded as per Corr. 4.3. The trajectory of such a particle is given by (4.18) with*

$$q_k(\phi_k(t)) = \sum_{\ell=1}^{\infty} \mu_{k,\ell} \frac{\ell\Omega \sin\left(\ell\phi_k(t) - \varphi_{k,\ell}\right) + i\omega \cos\left(\ell\phi_k(t) - \varphi_{k,\ell}\right)}{\left(\ell\Omega\right)^2 - \omega^2} \tag{4.22}$$

Proof. Let $v_{k,\ell}$ be the ℓ^{th} term of the Fourier series (4.21) and $q_{k,\ell}$ the solution to

$$\ell\Omega \frac{\partial q_{k,\ell}}{\partial \phi_k} + i\omega q_{k,\ell} = v_{k,\ell} \tag{4.23}$$

with $q_{k,\ell}(\phi_k(0)) = -V_{k,\ell}(-i\omega)$, where $V_{k,\ell}(s)$ is the Laplace transform of $v_{k,\ell}$. By linearity, $q_k = \sum_{\ell=1}^{\infty} q_{k,\ell}$. From Lemma 4.1, $q_{k,\ell}$ contains only the solution of (4.23) due to $v_{k,\ell}$ and hence there exist complex constants $A_{k,\ell}$ and $B_{k,\ell}$ such that

$$q_{k,\ell}(\phi_k(t)) = A_{k,\ell}\cos \ell\phi_k(t) + B_{k,\ell}\sin \ell\phi_k(t),\ t \geq 0.$$

Plugging into (4.23) and solving for $A_{k,\ell}$ and $B_{k,\ell}$ in terms of the coefficients $a_{k,\ell}$ and $b_{k,\ell}$ yields

$$\begin{aligned} q_{k,\ell}(\phi_k(t)) = & \frac{\ell\Omega\left(a_{k,\ell}\sin \ell\phi_k(t) - b_{k,\ell}\cos \ell\phi_k(t)\right)}{(\ell\Omega)^2 - \omega^2} \\ & + i\frac{\omega\left(a_{k,\ell}\cos \ell\phi_k(t) + b_{k,\ell}\sin \ell\phi_k(t)\right)}{(\ell\Omega)^2 - \omega^2}, \end{aligned} \tag{4.24}$$

which is by trigonometry equivalent to the ℓ^{th} term of (4.22). □

Next, we examine the trajectory of a steered particle with purely sinusoidal speed as a step towards providing a geometric description of the trajectory of a particle with a general periodic speed profile. We give a description of the sinusoidal-speed particle's trajectory as it relates to the corresponding constant-speed particle's trajectory. The constant-speed particle moves around a circle as described in Thm. 4.1 and the sinusoidal-speed particle moves relative to the constant-speed particle in an elliptical orbit.

Corollary 4.4. *(Trajectory of a steered particle with sinusoidal speed profile) Consider a steered particle with constant turning rate* $\omega \neq 0$ *and speed profile* $\alpha_k = \nu_k + \mu_k \cos\phi_k$ *where* ν_k *and* μ_k *are constants and* $\dot{\phi}_k = \Omega > 0$ *with* $\Omega \neq |\omega|$. *The particle's trajectory is given by*

$$r_k = c_k + R_k(\theta_k) + q_k(\phi_k)e^{i\theta_k} \tag{4.25}$$

where $R_k(\theta_k)$ *is defined by (4.11),*

$$q_k(\phi_k) = \mu_k \frac{\Omega \sin \phi_k + i\omega \cos \phi_k}{\Omega^2 - \omega^2}, \tag{4.26}$$

and $c_k = r_k(0) - R_k(\theta_k(0)) - q_k(\phi_k(0))e^{i\theta_k(0)}$.

Proof. Follows from Thm. 4.5 by taking a single term of the series (4.22). ☐

From (4.26) we have

$$\left(\frac{\mathrm{Re}\left\{q_k\right\}}{\Omega}\right)^2 + \left(\frac{\mathrm{Im}\left\{q_k\right\}}{\omega}\right)^2 = \left(\frac{\mu_k}{\Omega^2 - \omega^2}\right)^2 \tag{4.27}$$

and therefore the locus of q_k in the complex plane is an ellipse with eccentricity

$$e_k = \begin{cases} \sqrt{1 - \frac{\Omega^2}{\omega^2}}, & \Omega < \omega \\ \sqrt{1 - \frac{\omega^2}{\Omega^2}}, & \Omega > \omega \end{cases}$$

and distance from the center to the focus

$$f_k = \frac{\mu_k}{\sqrt{|\Omega^2 - \omega^2|}}.$$

As ϕ_k increases with time, the trajectory moves in a counter-clockwise direction if $\omega > 0$ and clockwise if $\omega < 0$. The semi-major and semi-minor axes are $\mu_k \left|\frac{\Omega}{\Omega^2 - \omega^2}\right|$ and $\mu_k \left|\frac{\omega}{\Omega^2 - \omega^2}\right|$, with the larger of the two being the semi-major axis. q_k is bounded along the real and imaginary axes as

$$|\mathrm{Re}\left\{q_k\right\}| \leq \mu_k \left|\frac{\Omega}{\Omega^2 - \omega^2}\right|, \tag{4.28}$$

$$|\mathrm{Im}\left\{q_k\right\}| \leq \mu_k \left|\frac{\omega}{\Omega^2 - \omega^2}\right|. \tag{4.29}$$

Recall, as in (4.19), that q_k expresses the position of the particle relative to a constant-speed equivalent particle whose trajectory is a circle of radius $\nu_k|\omega|^{-1}$. $\mathrm{Re}\{q_k\}$ is the component in the direction of motion and $\mathrm{Im}\{q_k\}$ is the component perpendicular to the direction of motion. Because the velocity is tangential to the constant-speed circle, $\mathrm{Re}\{q_k\}$ is also tangential and $\mathrm{Im}\{q_k\}$ points either toward the center of the circle if $\omega > 0$ or away from it if $\omega < 0$.

The ratio $\left|\frac{\Omega}{\omega}\right|$ plays a key role in determining the shape of the q_k trajectory and therefore the trajectory of the sinusoidal-speed particle. When $\left|\frac{\Omega}{\omega}\right| > 1$, the semi-major axis of the q_k ellipse is aligned with the real axis and therefore with the direction of motion of the particle. When $\left|\frac{\Omega}{\omega}\right| < 1$, the semi-major axis is aligned with the imaginary axis and therefore perpendicular to the direction of motion. Furthermore, if $\left|\frac{\Omega}{\omega}\right| \in \mathbb{Q}$, then it follows from Thm. 4.4 that the trajectory is periodic. If $\left|\frac{\Omega}{\omega}\right| = m \in \mathbb{Z}^+$, there are m cycles of the speed phase during each trip around the constant-speed circle and, particularly for μ_k near 1, the trajectory of the sinusoidal-speed particle takes a shape similar to a “rounded out” polygon with m equal-length sides.

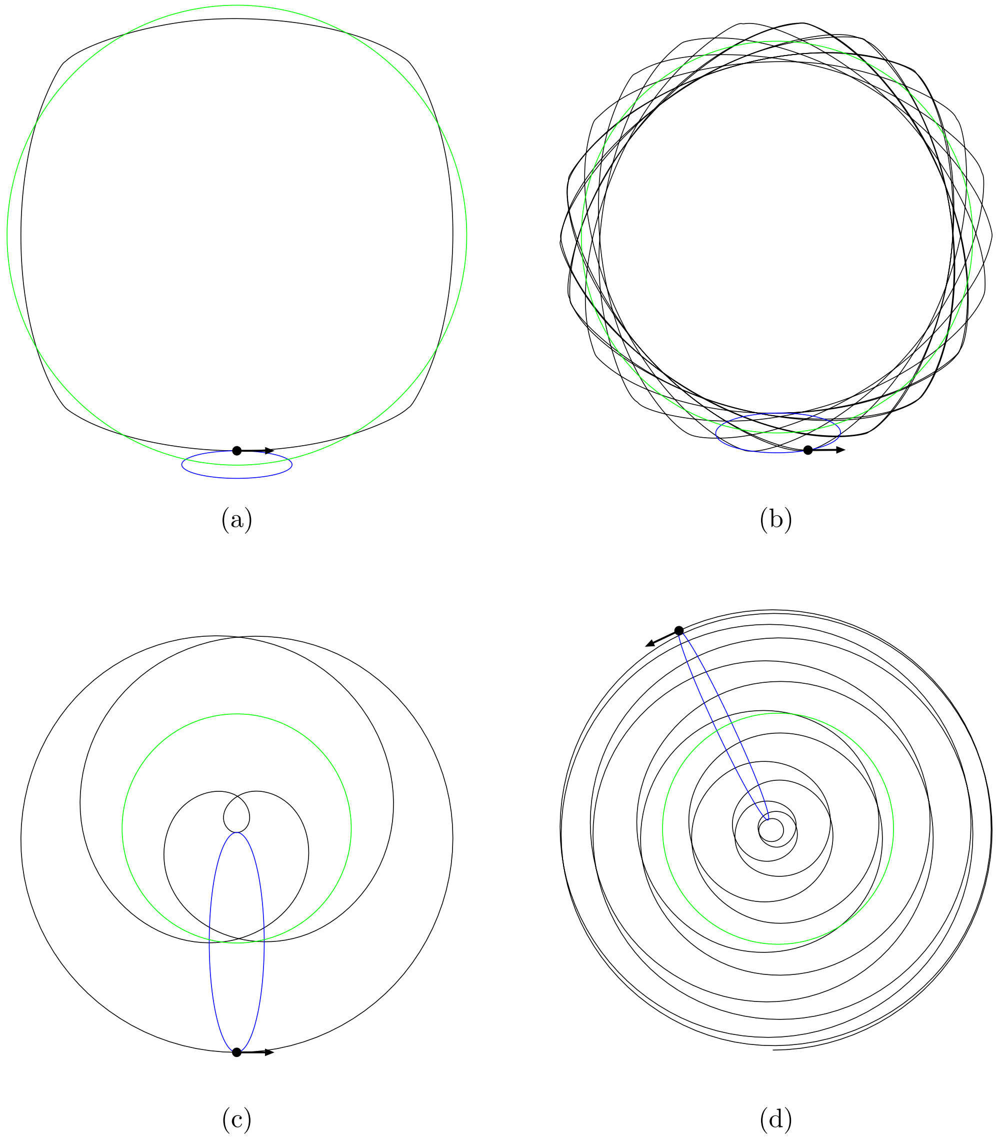
Fig. 4.4 shows four example trajectories of steered particles with periodic speed profiles. In all four trajectories, the nominal speed of the steered particle is $\nu_k = 1$ and the speed oscillation amplitude is $\mu_k = 0.9$. In Fig. 4.4a, the turning rate is $\omega = 1$ and the speed oscillation frequency is $\Omega = 4$. Because $\left|\frac{\Omega}{\omega}\right| > 1$, the q_k ellipse is elongated in the direction of motion. Because $\left|\frac{\Omega}{\omega}\right| = 4 \in \mathbb{Q}$, the trajectory is periodic and resembles a “rounded-out” square. In Fig. 4.4b, the turning rate is $\omega = 1$ and the speed oscillation frequency is $\Omega = \pi$. Here we have $\left|\frac{\Omega}{\omega}\right| = \pi \notin \mathbb{Q}$ and the trajectory is aperiodic. In Fig. 4.4c, the turning rate is $\omega = 4$ and the speed oscillation frequency is $\Omega = 1$. Because $\left|\frac{\Omega}{\omega}\right| < 1$, the q_k ellipse (shown in blue) is elongated perpendicular to the direction of motion (i.e., towards the center of the trajectory) and the trajectory has a cartioid-like shape. Because $\left|\frac{\Omega}{\omega}\right| = \frac{1}{4} \in \mathbb{Q}$, the trajectory is periodic. In Fig. 4.4d, the turning rate is $\omega = \pi$ and the speed oscillation frequency is $\Omega = 1$. As in Fig. 4.4c, $\left|\frac{\Omega}{\omega}\right| < 1$ and therefore the trajectory has a cartioid-like shape $\left|\frac{\Omega}{\omega}\right| = \frac{1}{\pi} \notin \mathbb{Q}$.

The shape of the trajectory of a steered particle with a periodic speed is determined by the R_k circle and the q_k locus. As described in Thm. 4.5, $q_k = \sum_{\ell=1}^{\infty} q_{k,\ell}$, and each of the $q_{k,\ell}$ in is identical in form to (4.26). Therefore, q_k is the superposition of ellipses; one ellipse for each term in the Fourier series of v_k. The $q_{k,1}$ ellipse is centered at the point $c_k + R_k(\theta_k)$, which would be the position of a second particle with the same motion parameters and initial conditions as the original particle but with constant speed. The $q_{k,2}$ ellipse is centered at a point on the $q_{k,1}$ ellipse determined by the value of $\phi_k(t)$, and so on. Fig. 4.5 shows the construction of the q_k locus for a steered particle with periodic speed profile shown in Fig. 4.5a. The full trajectory is shown in Fig. 4.6. The turning rate is $\omega = 0.3\pi$, the speed oscillation frequency is $\Omega = \pi$, and the nominal speed is $\nu = 1$. Note that because $\left|\frac{\Omega}{\omega}\right| = \frac{10}{3} \in \mathbb{Q}$, the trajectory shown in Fig. 4.6 is periodic.

4.3 Coordinated Control of Steered Particles with Periodic Speed Profiles

We now prove stability of a control law for heading (θ_k) and speed phase (ϕ_k) that enable the coordination of a group of N steered particles with constant turning rate and periodically time-varying speed at steady state. Coordination is in terms of the relative position of trajectory centers (c_k) and phase arrangements of the heading and speed phase. The control law is given first. The proof follows a series of supporting lemmas.

The control law that we present in Thm. 4.6 allows for almost arbitrary speed profiles and potentially heterogenous particles if the locations of the trajectory centers can be communicated between agents. When there is homogeneity of the particles, we



(a) (b)

(c) (d)

Figure 4.4: Trajectories (black curves) of steered particles with sinusoidal speed profiles as described by (4.13). The green curve is the locus of the $R_k(\theta_k)$ circle, the arrow indicates the final position and orientation of the particle, and the blue curve is the locus of the $q_k(\phi_k)$ ellipse. In all four cases, $\nu_k = 1$ and $\mu_k = 0.9$. (a) A "rounded-out" periodic polygon resulting from $\omega = 1$ and $\Omega = 4$. (b) An aperiodic trajectory resulting from $\omega = 1$ and $\Omega = \pi$. (c) A periodic cartioid-type trajectory resulting from $\Omega = 1$ and $\omega = 4$. (d) An aperiodic cartioid-type trajectory resulting from $\Omega = 1$ and $\omega = \pi$.

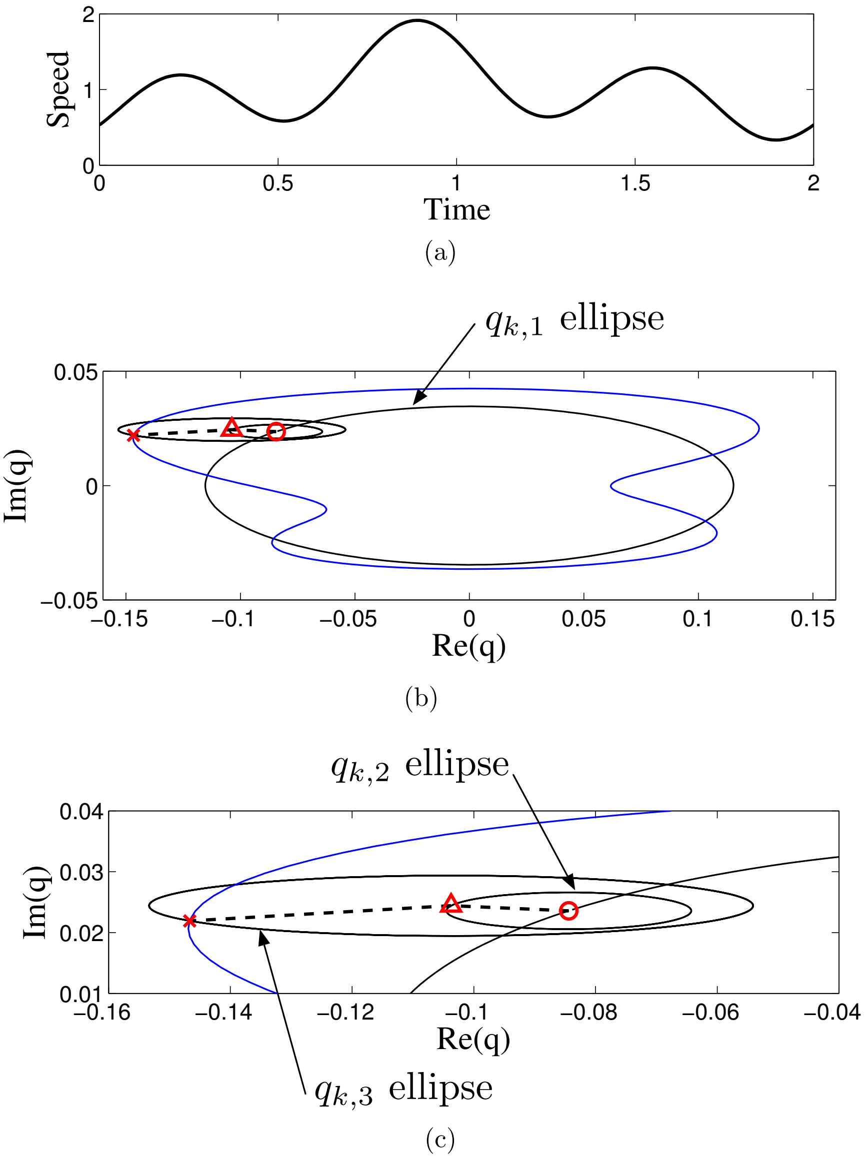


Figure 4.5: Trajectory construction for a periodic speed profile with $v_k(t)$ as described by (4.21). (a) The speed profile, generated from 3 randomly chosen Fourier components: $a_1 = -0.3213$, $a_2 = 0.0858$, $a_3 = -0.2322$, $b_1 = 0.0740$, $b_2 = -0.0890$, $b_3 = 0.4008$. The speed oscillation frequency is $\Omega = \pi$ and the nominal speed is $\nu_k = 1$. (b) The q_k locus (blue curve), which consists of the superposition of the $q_{k,\ell}$, $\ell = 1, 2, 3$, ellipses (black curves). The turning rate is $\omega = 0.3\pi$. For illustration purposes, the value $q_k(t^*)$ for some time $t = t^*$ is indicated by the red 'x'. The blue curve is the locus of $q_k(t)$ for all t. (c) Close-up of (b). At time $t = t^*$, the center of the $q_{k,2}$ ellipse is located at $q_{k,1}(t^*)$ (indicated by the red circle). Similarly, the center of the $q_{k,3}$ ellipse (indicated by the red triangle) is at $q_{k,1}(t^*) + q_{k,2}(t^*)$. The point $q_k(t^*)$ (red 'x') is located at $q_{k,1}(t^*) + q_{k,2}(t^*) + q_{k,3}(t^*)$. The dashed lines connect the centers of the $q_{k,2}$ and $q_{k,3}$ ellipses.

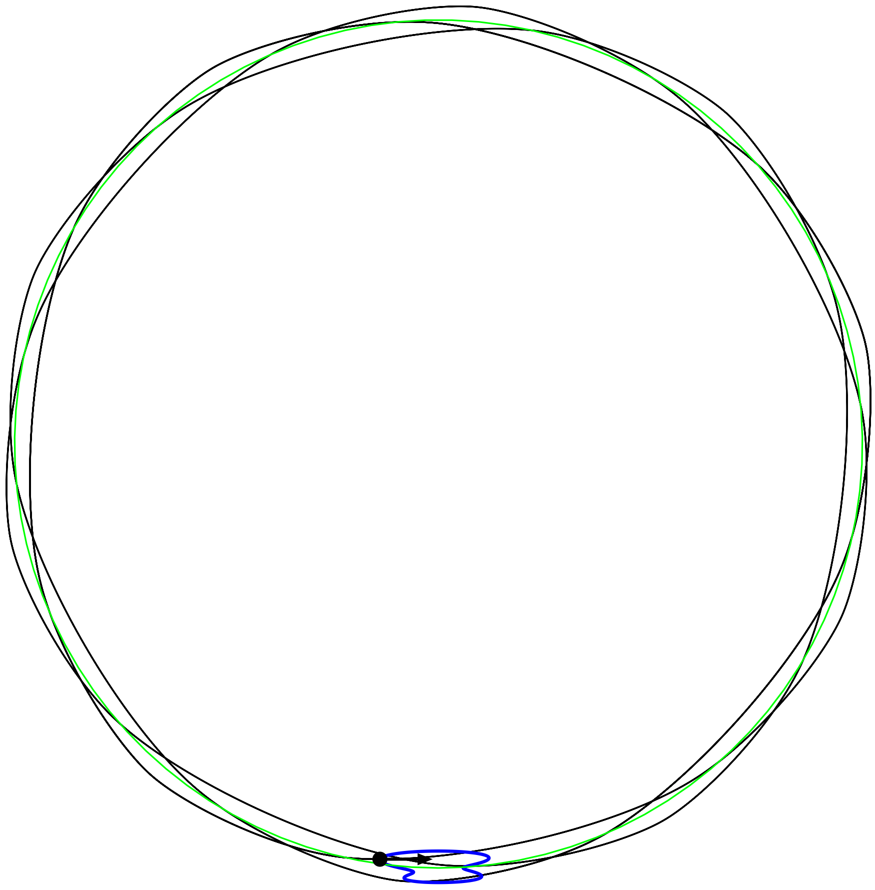


Figure 4.6: Trajectory of a steered particle with the periodic speed profile described in Fig. 4.5. The particle's trajectory is shown in black, the q_k locus at the final time instant is shown in blue, and the R_k circle is shown in green. The particle's position at the final time instant is indicated by the black dot, and the direction of its velocity is indicated by the black arrow.

show that the control laws can be calculated in terms of measured relative quantities and hence communication requirements may be relaxed or eliminated.

The control terms here vanish as the formation approaches steady state (i.e., $u_k \to 0$ and $g_k \to 0$ as $t \to \infty$), and hence the steady state trajectories of individual agents are as described in Sec. 4.2. Therefore, we use the same notation in this section as we have above, with three important exceptions that account for the transient part of the trajectory while keeping the trajectory descriptions consistent with Sec. 4.2 in steady-state. First, we define $R(\theta_k(t))$ independent of the value of $r_k(t)$ as in (4.11):

$$R(\theta_k(t)) \triangleq -i\frac{\nu_k}{\omega}e^{i\theta_k(t)}.$$

Second, we take (4.22) as the definition of q_k. That is, q_k maps the speed phase to a point in the complex plane, and the shape of the locus of q_k is determined by the Fourier coefficients of the speed variation (4.20). Note that q_k still satisfies the ODE (4.15). Finally, we define the *instantanous center* of the trajectory, $c_k(t)$, as

$$c_k(t) \triangleq r_k(t) - R_k(\theta_k(t)) - q_k(\phi_k(t))e^{i\theta_k(t)}. \tag{4.30}$$

In steady state, c_k is a constant and can be interpreted as the center of the steady state trajectory as described above. From here on, when we mention c_k, we mean the instantaneous center of the trajectory as defined by (4.30).

As discussed in Sec. 4.1.2, gradient controls on phase potentials can be used to stabilize particular phase arrangements. Following [104, 105], to stabilize relative positions of the trajectory centers we utilize a consensus-like potential, $C(\mathbf{c})$, which is defined as

$$C(\mathbf{c}) = \frac{1}{2}\left\|P\mathbf{c} - \mathbf{d}\right\|^2 \geq 0, \tag{4.31}$$

where P is the projection matrix defined by $P = I_{N\times N} - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T$ and $\mathbf{d} = \begin{bmatrix} d_1 & \cdots & d_N \end{bmatrix}^T \in \mathbb{C}^N$ is a constant vector of trajectory offsets that satisfies $\sum_{k=1}^N d_k = 0$. Denoting P_k as the k^{th} row of P and $\bar{c} = \frac{1}{N}\sum_{k=1}^N c_k$, we have

$$P_k\mathbf{c} = c_k - \bar{c}$$

and we can write (4.31) as

$$C(\mathbf{c}) = \frac{1}{2}\sum_{k=1}^N \|P_k\mathbf{c} - d_k\|^2 . \tag{4.32}$$

Because each term of (4.32) is non-negative, the set $\mathcal{C} = \left\{\mathbf{c} \in \mathbb{C}^N : C(\mathbf{c}) = 0\right\}$ is equivalent to the set $\left\{\mathbf{c} \in \mathbb{C}^N : c_k = \bar{c} + d_k,\ k = 1, \ldots, N\right\}$. That is, when $C(\mathbf{c}) = 0$, d_k is the offset of c_k from the center of mass of all trajectory centers. If $d_k = 0$ for each k, then $\mathcal{C}$ corresponds to all trajectories sharing the same center, $\bar{c}$.

We can now state the control law.

Theorem 4.6. *Consider a group of N steered particles, each with dynamics (4.1), (4.2), and (4.3), and speed profile*

$$\alpha_k = \nu_k + v_k(\phi_k) \tag{4.33}$$

where $\nu_k > 0$ is constant and $v_k > -\nu_k$ has Fourier coefficients as defined by (4.20). Let $\mathbf{d} \in \mathbb{C}^N$ be a constant vector such that $\sum_{k=1}^N d_k = 0$ and $U(\boldsymbol{\theta})$ and $V(\boldsymbol{\phi})$ be two phase potentials (see Def. 4.1). The heading control

$$u_k = \kappa_c\omega^{-1}\left\langle P_k\mathbf{c} - d_k, (\nu_k + i\omega q_k)\, e^{i\theta_k}\right\rangle - \kappa_\theta\frac{\partial U}{\partial\theta_k} \tag{4.34}$$

and speed phase control

$$g_k = \kappa_c \Omega^{-1} \left\langle P_k \mathbf{c} - d_k, (v_k - i\omega q_k)\, e^{i\theta_k} \right\rangle - \kappa_\phi \frac{\partial V}{\partial \phi_k}, \tag{4.35}$$

where q_k *is defined by (4.22),* $\kappa_c > 0$*,* κ_θ*, and* κ_ϕ *are real constants, stabilize the group of* N *particles into formations defined by the following criteria.*

1. *The headings belong to the phase arrangement corresponding to a critical set of* $U(\boldsymbol{\theta})$ *when* $\kappa_\theta \neq 0$*.*
2. *The speed phases belong to the phase arrangement corresponding to a critical set of* $V(\boldsymbol{\phi})$ *when* $\kappa_\phi \neq 0$*.*
3. *Each trajectory center is constant and given by* $c_k = \bar{c} + d_k$*, where* $\bar{c} = \frac{1}{N}\sum_{k=1}^{N} c_k$*.*

Furthermore, the phase arrangement for $\boldsymbol{\theta}$ *(resp.* $\boldsymbol{\phi}$*) corresponds to a local maximum of* U *(resp.* V*) when* $\kappa_\theta > 0$ *(resp.* $\kappa_\phi > 0$*) and a local minimum when* $\kappa_\theta < 0$ *(resp.* $\kappa_\phi < 0$*).*

Proof of Thm. 4.6 requires several supporting lemmas. The first lemma defines a candidate Lyapunov function.

Lemma 4.2. *Define the function* $S(\mathbf{r}, \boldsymbol{\theta}, \boldsymbol{\phi})$ *as*

$$S(\mathbf{r}, \boldsymbol{\theta}, \boldsymbol{\phi}) \triangleq \kappa_c C(\mathbf{c}) + \kappa_\theta U(\boldsymbol{\theta}) + \kappa_\phi V(\boldsymbol{\phi}) + S_0 \geq 0, \tag{4.36}$$

where $C(\mathbf{c})$ *is defined by (4.31),* $U(\boldsymbol{\theta})$ *and* $V(\boldsymbol{\phi})$ *are both phase potentials satisfying Def. 4.1,* $\kappa_c > 0$*,* κ_θ *and* κ_ϕ *are real constants, and*

$$S_0 = \min_{(\boldsymbol{\theta}, \boldsymbol{\phi}) \in T^N \times T^N} \left\{ \kappa_\theta U(\boldsymbol{\theta}) + \kappa_\phi V(\boldsymbol{\phi}) \right\}$$

ensures that $S \geq 0$*. Along solutions of the steered particle dynamics (4.1)-(4.3) with controls (4.34) and (4.35), we have*

$$\dot{S} = -\sum_{k=1}^{N} \left(u_k^2 + g_k^2\right) \leq 0$$

and therefore S *is non-increasing.*

Proof. The time-derivative of S is

$$\dot{S} = \kappa_c \frac{d}{dt} C(\mathbf{c}) + \kappa_\theta \sum_{k=1}^{N} \frac{\partial U}{\partial \theta_k} \dot{\theta}_k + \kappa_\phi \sum_{k=1}^{N} \frac{\partial V}{\partial \phi_k} \dot{\phi}_k \tag{4.37}$$

We have

$$\frac{d}{dt} C(\mathbf{c}) = \langle P\mathbf{c} - \mathbf{d}, P\dot{\mathbf{c}} \rangle = \left\langle P^T \left(P\mathbf{c} - \mathbf{d}\right), \dot{\mathbf{c}} \right\rangle = \langle P\mathbf{c} - \mathbf{d}, \dot{\mathbf{c}} \rangle = \sum_{k=1}^{N} \langle P_k \mathbf{c} - d_k, \dot{c}_k \rangle \tag{4.38}$$

because $\mathbf{d}$ is constant, $P^T\mathbf{d} = P\mathbf{d} = 0$, and $P^T P = P^2 = P$. The k^{th} term of the sum depends on

$$\dot{c}_k = \dot{r}_k - \frac{\partial R_k(\theta_k)}{\partial \theta_k} \left(\omega + u_k\right) - \frac{\partial q_k(\phi_k)}{\partial \phi_k} \left(\Omega + g_k\right) e^{i\theta_k} - i q_k e^{i\theta_k} \left(\omega + u_k\right).$$

Substituting (4.11) and (4.15) gives

$$\dot{c}_k = -\frac{u_k}{\omega} \left(\nu_k + i\omega q_k\right) e^{i\theta_k} - \frac{g_k}{\Omega} \left(v_k - i\omega q_k\right) e^{i\theta_k} \tag{4.39}$$

and hence the k^{th} term of (4.38) is

$$\langle P_k \mathbf{c} - d_k, \dot{c}_k \rangle = \\ - \left\langle P_k \mathbf{c} - d_k, \frac{u_k}{\omega} \left(\nu_k + i\omega q_k\right) e^{i\theta_k} \right\rangle - \left\langle P_k \mathbf{c} - d_k, \frac{g_k}{\Omega} \left(v_k - i\omega q_k\right) e^{i\theta_k} \right\rangle.$$

From (4.34) and (4.35) we have

$$\left\langle P_k \mathbf{c} - d_k, (\nu_k + i\omega q_k)\, e^{i\theta_k} \right\rangle = \frac{1}{\kappa_c}\left(\omega u_k + \kappa_\theta \omega \frac{\partial U}{\partial \theta_k}\right)$$

and

$$\left\langle P_k \mathbf{c} - d_k, (v_k - i\omega q_k)\, e^{i\theta_k} \right\rangle = \frac{1}{\kappa_c}\left(\Omega g_k + \kappa_\phi \Omega \frac{\partial V}{\partial \phi_k}\right),$$

and thus

$$\frac{d}{dt} C(\mathbf{c}) = \frac{1}{\kappa_c} \sum_{k=1}^{N} \left(-u_k^2 - g_k^2 - u_k \kappa_\theta \frac{\partial U}{\partial \theta_k} - g_k \kappa_\phi \frac{\partial V}{\partial \phi_k}\right). \tag{4.40}$$

Therefore, substituting (4.40), (4.2), and (4.3) into (4.37), we have

$$\begin{aligned} \dot{S} = & \sum_{k=1}^{N} \left(-u_k^2 - g_k^2 - u_k \kappa_\theta \frac{\partial U}{\partial \theta_k} - g_k \kappa_\phi \frac{\partial V}{\partial \phi_k} \right. \\ & \left. + \kappa_\theta \frac{\partial U}{\partial \theta_k}(\omega + u_k) + \kappa_\phi \frac{\partial V}{\partial \phi_k}(\Omega + g_k)\right) \\ = & \sum_{k=1}^{N} \left(-u_k^2 - g_k^2\right) + \kappa_\theta \sum_{k=1}^{N} \frac{\partial U}{\partial \theta_k}\omega + \kappa_\phi \sum_{k=1}^{N} \frac{\partial V}{\partial \phi_k}\Omega. \end{aligned}$$

Because U and V are phase potentials, $\sum_{k=1}^{N} \frac{\partial U}{\partial \theta_k}\omega = \sum_{k=1}^{N} \frac{\partial V}{\partial \phi_k}\Omega = 0$, and therefore

$$\dot{S} = -\sum_{k=1}^{N} \left(u_k^2 + g_k^2\right) \leq 0. \tag{4.41}$$

Hence S is non-increasing along the dynamics (4.1)-(4.3) with controls (4.34) and (4.35). □

We also require the existence and compactness of positively invariant sets for S along solutions of the dynamics (4.1)-(4.3) with controls (4.34) and (4.35).

Lemma 4.3. *Consider the candidate Lyapunov function S as defined in Lemma 4.2. For any $p > 0$, define the set*

$$W_p = \left\{(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\phi}) \in (\text{Image } P) \times T^N \times T^N \mid \mathbf{x} = P\mathbf{c} - \mathbf{d},\ S(\mathbf{r}', \boldsymbol{\theta}, \boldsymbol{\phi}) \leq p, \right\},$$

where $\mathbf{r}'$ is the position vector of a set of N particles with headings $\boldsymbol{\theta}$, speed phases $\boldsymbol{\phi}$, and trajectory center vector $\mathbf{c}$. Then W_p is positively invariant along solutions of the dynamics (4.1)-(4.3) with controls (4.34) and (4.35). Furthermore, W_p is a compact subset of

$$D = \left\{(\text{Image } P) \times T^N \times T^N\right\}.$$

Proof. Positive invariance follows from Lemma 4.2. On W_p, we have $\frac{\kappa_c}{2}\|\mathbf{x}\|^2 \leq S \leq p$ and $\mathbf{x} = P\mathbf{c} - \mathbf{d} = P(\mathbf{c} - \mathbf{d}) \in \text{Image } P$; hence $\mathbf{x}$ is in the compact set $\left\{\mathbf{z} \in (\text{Image } P) \mid \|\mathbf{z}\|^2 \leq \frac{2p}{\kappa_c}\right\}$. $T^N \times T^N$ is itself a compact set and therefore W_p is a compact subset of D. □

Finally, we establish the following result for invariant sets on $\dot{S} = 0$.

Lemma 4.4. *Consider the candidate Lyapunov function S as defined in Lemma 4.2 and the set*

$$\Lambda = \left\{(\mathbf{r}, \boldsymbol{\theta}, \boldsymbol{\phi}) \in \left(\mathbb{C}^N \times T^N \times T^N\right) \mid \dot{S}(\mathbf{r}, \boldsymbol{\theta}, \boldsymbol{\phi}) \equiv 0\right\}.$$

Invariant sets on Λ are subsets of

$$M = \left\{(\mathbf{r}, \boldsymbol{\theta}, \boldsymbol{\phi}) \in \Lambda \mid P\mathbf{c} - \mathbf{d} = 0 \text{ and } \frac{\partial U}{\partial \theta_k} = \frac{\partial V}{\partial \phi_k} = 0,\ \forall k\right\}.$$

Furthermore, $\dot{c}_k = 0$, $\dot{\theta}_k = \omega$, $\dot{\phi}_k = \Omega$, and $P_k\mathbf{c} = d_k$ on Λ for each k.

Proof. From (4.41), $\dot{S} \equiv 0$ if and only if $u_k \equiv 0$ and $g_k \equiv 0$ for each k and therefore $\dot{\theta}_k = \omega$ and $\dot{\phi}_k = \Omega$. The time derivative of the heading control is

$$\frac{d}{dt}u_k = \kappa_c \omega^{-1} \left\langle P_k\mathbf{c} - d_k, \left(\nu_k + i\omega q_k\right) e^{i\theta_k} i\dot{\theta}_k + i\omega \dot{q}_k e^{i\theta_k} \right\rangle \\ + \kappa\omega^{-1} \left\langle P_k\dot{\mathbf{c}}, \left(\nu_k + i\omega q_k\right) e^{i\theta_k} \right\rangle - \kappa_\theta \frac{d}{dt}\frac{\partial U}{\partial \theta_k}.$$

On Λ we have $\dot{\mathbf{c}} = 0$ because $\dot{c}_k = 0$ for each k by (4.39), $\dot{\theta}_k = \omega$ because $u_k = 0$, $\dot{q}_k = \Omega \frac{\partial q}{\partial \phi_k}$ because $g_k = 0$, and from Corr. 4.1 we have

$$\frac{d}{dt}\frac{\partial U}{\partial \theta_k} = \frac{\partial}{\partial \theta_k} \sum_{j=1}^{N} \frac{\partial U}{\partial \theta_j}\omega = 0.$$

Therefore, on Λ

$$\frac{d}{dt}u_k = \kappa_c \left\langle P_k\mathbf{c} - d_k, i\left(\nu_k + i\omega q_k + \Omega\frac{\partial q_k}{\partial \phi_k}\right) e^{i\theta_k} \right\rangle.$$

Substituting (4.15) yields

$$\frac{d}{dt}u_k = \kappa_c \left\langle P_k\mathbf{c} - d_k, i\left(\nu_k + v_k(\phi_k)\right) e^{i\theta_k} \right\rangle = \kappa_c \left\langle P_k\mathbf{c} - d_k, i\dot{r}_k \right\rangle.$$

Since on Λ we have $u_k \equiv 0$, $\frac{d}{dt}u_k \equiv 0$ and

$$\left\langle P_k\mathbf{c} - d_k, i\dot{r}_k \right\rangle \equiv 0.$$

Because $\mathbf{c}$ is constant and the velocity $\dot{r}_k$ is nonzero in magnitude and has a direction that is evolving with a constant nonzero natural frequency, we must have that $P_k\mathbf{c} = d_k$ on Λ. Furthermore, on Λ, because $P_k\mathbf{c} - d_k = u_k = g_k = 0$, by (4.34) and (4.35) we must have

$$\frac{\partial U}{\partial \theta_k} = \frac{\partial V}{\partial \phi_k} = 0.$$

□

We can now complete the proof of Thm. 4.6.

Proof. (Of Thm. 4.6) Lemma 4.2 establishes a positive semidefinite Lyapunov function $S(\mathbf{r}, \boldsymbol{\theta}, \boldsymbol{\phi})$ with $\dot{S} \leq 0$ along solutions of the dynamics. From Lemma 4.3, we have the existence of positively invariant sets W_p that are compact subsets of $\left\{(\text{Image } P) \times T^N \times T^N\right\}$. The dynamics therefore converge to the largest invariant set Λ on which $\dot{S} \equiv 0$. On Λ and for each k, we have $P_k\mathbf{c} = d_k$, $\dot{\theta}_k = \omega$, $\dot{\phi}_k = \Omega$, and $\frac{\partial U}{\partial \theta_k} = \frac{\partial V}{\partial \phi_k} = 0$ by Lemma 4.4. Therefore, the steered particles described by the conditions of the theorem converge to a formation in which the trajectory centers are constant and related to each other by the condition $c_k = \sum_{j=1}^{N} c_j + d_k$. Furthermore, the headings converge to a critical set of $U(\boldsymbol{\theta})$. Proof that the minima of $\kappa_\theta U$ and $\kappa_\phi V$ are stable follows the same argument employed by Sepulchre et al. for constant-speed particles on circular trajectories; roughly, the dynamics of the headings (resp. speed phases) on Λ can be seen to be stable and we have that $\dot{S}$ is non-increasing in all neighborhoods of Λ (see [104] Theorem 3, as well as [88] Lemma A.2 for a generalization). When the sign of κ_θ (resp. κ_ϕ) is flipped, the maxima of U (resp. V) are stabilized. □

4.4 Designing Coordinated Relative Motion

Sec. 4.3 provides a control law that stabilize a group of N steered particles with time-varying speed profiles to a large family of formations. We describe some of the possibilities in Sec. 4.4.1. Two of the terms that we can design to determine the formation are the phase potential for the headings, $U(\boldsymbol{\theta})$, and the phase potential for the speed phases, $V(\boldsymbol{\phi})$. We consider several combinations of phase potentials in Sec. 4.4.1, including synchronization of the headings and splaying of the speed phases. We show that this leads to a formation shape that is defined by the locus of

q_k, which is determined by the speed profile. In Sec. 4.4.2, we then provide methods to systematically design the speed profiles in order to obtain desired formation shapes.

4.4.1 Geometry of Coordinated Motion Formations

In this section, we discuss how the parameters of motion and the control laws provided by Thm. 4.6 determine the geometry of a formation of N agents, and we provide a survey of the types of formations that are available. We are concerned here only with periodic speed profiles and with bounded trajectories, i.e., $\omega \neq 0$ and the conditions of Corr. 4.3 are satisfied for each particle. Sec. 4.2 describes the trajectories of single steered particles with periodic speed profiles. Our goal here is to describe how individual trajectories can be coordinated to obtain various formations.

In Sec. 4.2, we showed that the shapes of individual trajectories can be described as follows. The radius of the circular component of each trajectory, i.e., the radius of $R_k(\theta_k)$, is determined by the steering rate, ω, and the nominal speed, ν_k. The trajectory center offset, d_k, describes the relative location of the center of the $R_k(\theta_k)$ circle for each particle k. The locus of q_k describes the trajectory's path relative to the $R_k(\theta_k)$ circle. The speed oscillation frequency Ω and the Fourier coefficients of the speed profile v_k determines the shape of the q_k locus as described by Thm. 4.5.

The shapes of *formations* are determined not only by the shapes of the trajectories of the individual particles, but also by the positions and orientations of trajectories relative to one another. As described in Thm. 4.6, we have three main mechanisms to control the coordination of trajectories. The relative positions of trajectory centers can be controlled by setting the individual d_k. The relative headings of the particles — and therefore their relative positions around the circular components of their trajectories — can be controlled via the heading phase potential, $U(\boldsymbol{\theta})$, and the sign of the gain κ_θ. The relative speed phases of the particles — and therefore their relative positions around their respective q_k — can be controlled via the speed phase potential, $V(\boldsymbol{\phi})$, and the sign of the gain κ_ϕ.

Coordination of the centers of the trajectories can be achieved without the phase potentials U and V. Fig. 4.7 shows a formation of $N = 25$ steered particles generated without heading or speed phase control (i.e., $U = V = 0$) and with $\mathbf{d} = 0$. The resulting formation is swarm-like; the only coordination is that all of the agents share the same trajectory center. Each particle alternates between being towards the center of the formation when its speed is small and being towards the edge of the formation when its speed is large. Adding speed phase balancing would balance the distribution of agents with respect to their distances from the center of the formation.

Fig. 4.8 shows a formation in which the speed phases are synchronized and the headings are in the splay phase arrangement (i.e., U is given by (4.6) and $\kappa_\theta > 0$ and V is given by (4.7) with $\kappa_\phi < 0$) and the trajectory offsets are evenly distributed in an array: $d_k = \sqrt{2}e^{i\frac{\pi}{4}(1+2k)}$. The individual trajectory shapes are square-like because the speed oscillation frequency is four times the turning rate, i.e., $\Omega = 4\omega$. The trajectory shapes are oriented identically because the speed phases are synchronized. At one point during each heading rotation period (i.e., $\frac{2\pi}{\omega}$), the agents are close to each other in the center of the formation. One half of a period after that, the agents are at the four corners of the formation (as shown).

Fig. 4.9 shows two formations of $N = 4$ agents with sinusoidal speed profiles. The headings are synchronized and the speed phases are in the splay phase arrangement, therefore they are evenly distributed around the same q_k ellipse. That is, the shape of the q_k ellipse — which is the same for all four agents because they share the same speed profile as a function of the speed phase — determines the shape of the formation. The agents stay in this formation over time because their headings stay the same and therefore, from (4.18), their trajectories vary only by their location along the locus of their (common) q_k locus. The speed phase determines the location

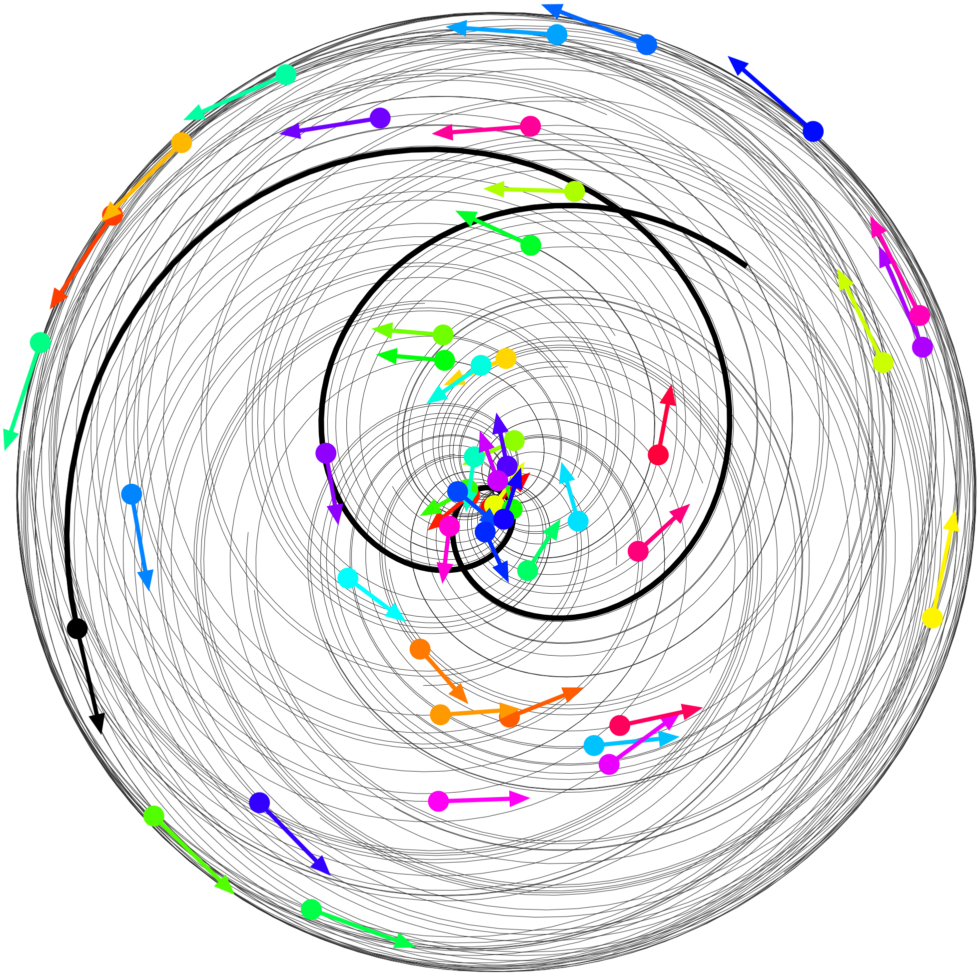


Figure 4.7: Swarm-like formation with no heading or speed phase control. The thick black line is the trajectory of the black particle. The initial headings and speed phases are random. The arrow lengths are related to speed: longer arrows correspond to higher speed. Parameters: $N = 25$, $\omega = 1$, $\Omega = 0.3$, $\nu = 1$. Purely sinusoidal speed with $a_{k,1} = 0.9$ and $b_{k,1} = 0$ for all k.

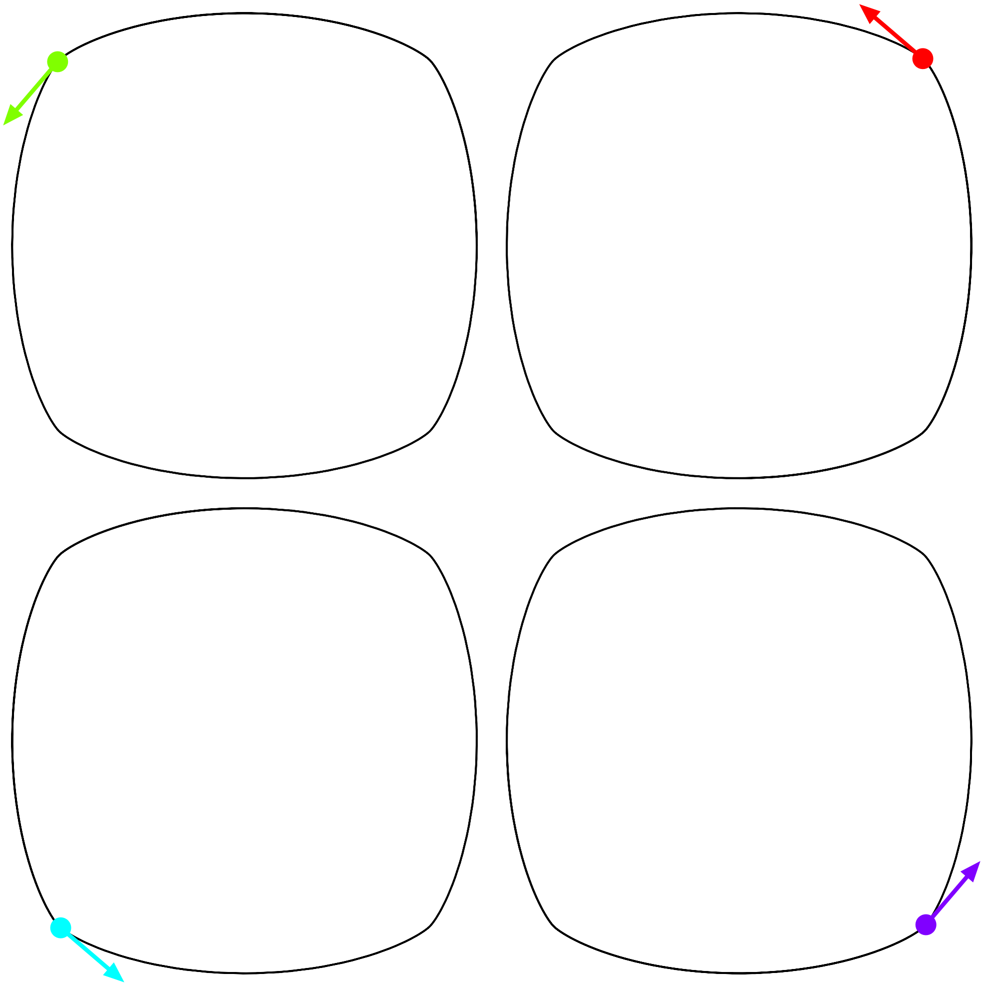
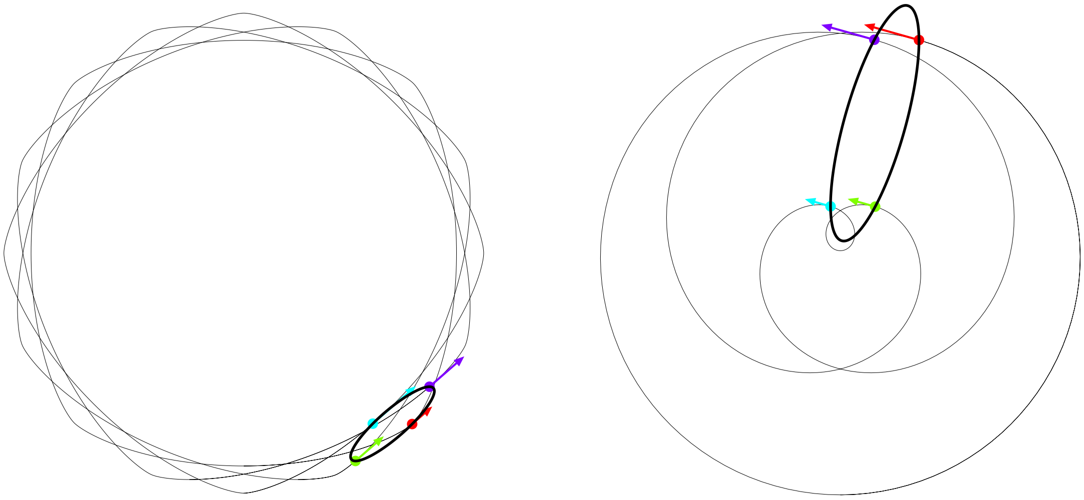


Figure 4.8: A formation with distributed trajectory offsets: $d_k = \sqrt{2}e^{i\frac{\pi}{4}(1+2k)}$. The speed phases are synchronized and the headings are in the splay phase arrangement. Parameters: $N = 4$, $\omega = 1$, $\Omega = 4$, $\nu = 1$. Purely sinusoidal speed with $a_{k,1} = 0.9$ and $b_{k,1} = 0$ for all k.



(a) (b)

Figure 4.9: A pair of formations of $N = 4$ agents with the same purely sinusoidal motion parameters except that in the left formation (a), $\omega = 1$ and $\Omega = 4$, and in the right formation (b), $\Omega = 1$ and $\omega = 4\Omega$. In both cases, the headings are synchronized and the speed phases are in the splay phase arrangement. Note that in both formations, all four agents are on the same q_k curve (thicker black curve) and therefore the shape of q_k (which is the same for all four agents) determines the shape of the formation. $\nu_k = 1$, $a_{k,1} = 0.9$, $b_{k,1} = 0$ for all k.

along the q_k. In Fig. 4.9a, $\omega = 1$ and $\Omega = 4\omega$ and the q_k ellipse has its semi-major axis in the direction of motion. In Fig. 4.9b, $\Omega = 1$ and $\omega = 4\Omega$ and the q_k ellipse has its semi-major axis perpendicular to the direction of motion. Aside from the natural frequencies, the motion parameters are identical in both formations: $\nu_k = 1$, $a_{k,1} = 0.9$, $b_{k,1} = 0$ for all k.

Fig. 4.10 shows a formation of $N = 4$ agents with periodic speed profiles, where the speed profile is the same one shown in Fig. 4.5a (generated with 3 random Fourier components). The speed oscillation frequency and turning rate are the same ones that were used for Fig. 4.5b ($\Omega = \pi$, $\omega = 0.3\pi$), and therefore the q_k locus is the one shown there. As in Fig. 4.9 above, the headings are synchronized and the speed phases are in the splay phase arrangement, and therefore the formation shape is defined by the

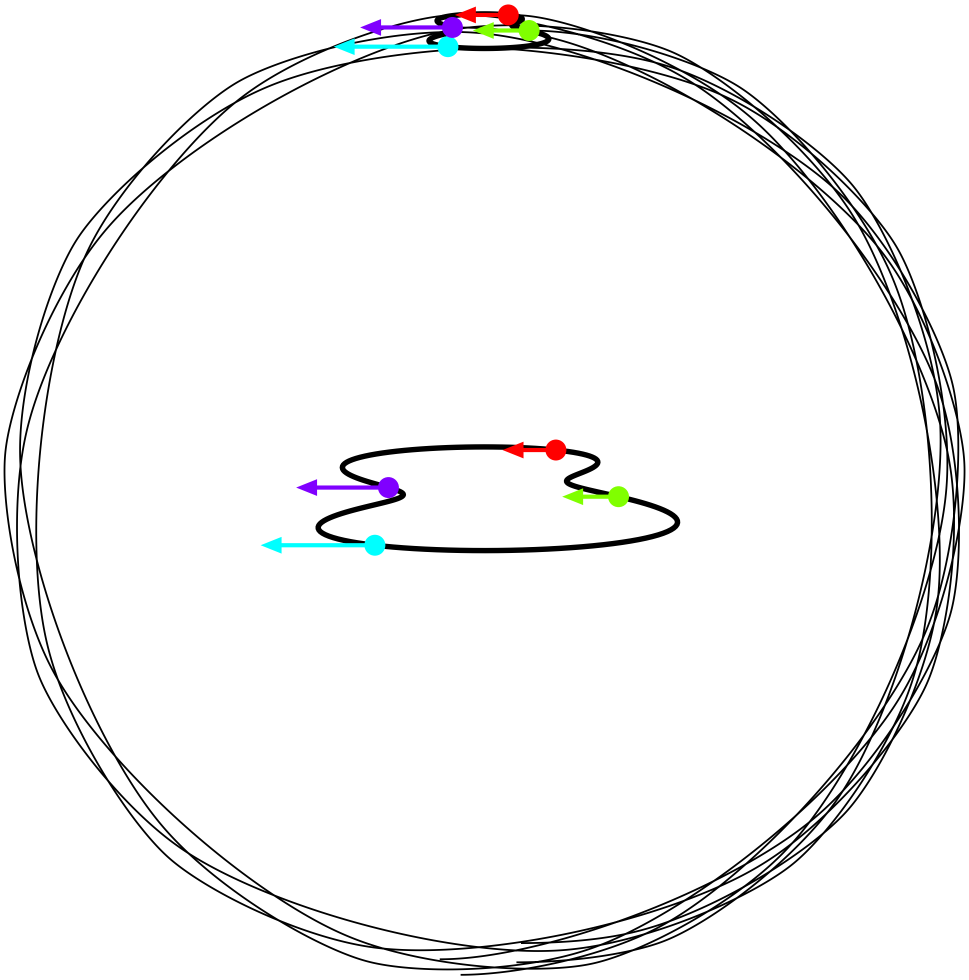


Figure 4.10: Formation of $N = 4$ agents with the periodic speed profile shown in Fig. 4.5a. As in Fig. 4.9, the headings are synchronized and the speed phases are in the splay phase arrangement. Therefore, the q_k shape — which is the same as in Fig. 4.5b — determines the shape of the formation. The shape of the formation is shown magnified in the center of the figure. $\Omega = \pi$, $\omega = 0.3\pi$.

q_k shape, with the agents being at different locations according to their ϕ_k. The formation shape is shown magnified in the center of the figure.

4.4.2 Design of Motion Parameters to Obtain Specific Formations

In this section, we describe a method for systematically designing motion parameters for obtaining specific formations. In the previous section, we describe how synchronizing the headings of a group of agents that share common motion parameters leads to a formation whose shape is defined by the shape of the (common) q_k locus moving around a circle with radius determined by the nominal speed, ν_k, and turning rate, ω. Here, we consider the inverse problem: given a formation in terms of the q_k locus and the radius of the circular component of the trajectory, how can we determine the necessary turning rate, nominal speed, speed oscillation frequency, and speed profile? Consequently, we consider the nominal speeds and speed profiles to all be the same and hence drop the k subscript, i.e., the nominal speed is ν and the speed profile is $\alpha(\phi) = \nu + v(\phi)$ (with the speed of the k^{th} agent being $\alpha(\phi_k) = \nu + v(\phi_k)$. We also drop the subscript from q_k and write $q(\phi_k)$ to denote the position of agent k on the (common) q locus. We will also generally consider the nominal speed to be fixed. This is reasonable for most physical applications, where the nominal speed is likely to be dictated by operational constraints.

We first consider the sinusoidal speed case because it can be solved analytically and gives us a method to estimate initial parameters for the general periodic speed profile case. We will also be able to establish some bounds on the types of formations that are achievable. The general periodic speed profile case is solved numerically.

For purely sinusoidal speed profiles, the elliptical component of the trajectory has bounds given by (4.28) and (4.29). These bounds provide a natural way to specify the elliptical shape of trajectories, and therefore we define the maximum dimensions of the q locus as

$$A \triangleq \max_{\phi \in S^1} \left| \mathrm{Re}\left\{ q(\phi) \right\} \right| \tag{4.42}$$

$$B \triangleq \max_{\phi \in S^1} \left| \mathrm{Im}\left\{ q(\phi) \right\} \right| . \tag{4.43}$$

We will also define the *aspect ratio** of the formation as

$$e \triangleq \frac{A}{B}. \tag{4.44}$$

We state the design problem for the case of a purely sinusoidal speed profile as follows.

Design Problem 4.1 (Formation Design - Sinusoidal Speed Profile)**.** *Consider a group of N agents under the assumptions of Thm. 4.6, with the added assumption that each agent shares a common purely sinusoidal speed profile*

$$\alpha(\phi) = \nu + v(\phi) = \nu + a_1 \cos\phi + b_1 \sin\phi. \tag{4.45}$$

Given the nominal speed, ν, the desired direction of motion, the desired radius of the circular component of the trajectory, C, and the formation dimensions, A and B defined by (4.42) and (4.43), determine

1. *if the formation is physically achievable, and, if so*
2. *what combination of turning rate, ω, speed oscillation frequency, Ω, and coefficients a_1 and b_1 will achieve it.*

---

*Aspect ratios are typically defined as the ratio of the longer of two dimensions to the shorter one and therefore are always greater than or equal to 1. Here we define it strictly as the ratio of A to B and therefore it can take any positive value.

Given the nominal speed, ν, and the radius of the circular component, C, the magnitude of the turning rate can be solved for from (4.11) as

$$|\omega| = \frac{\nu}{C}. \tag{4.46}$$

The rotation direction determines the sign of the turning rate as in Thm. 4.1; counter-clockwise rotation requires $\omega > 0$ and clockwise rotation requires $\omega < 0$.

We are free to choose $b_1 = 0$ and assume $a_1 > 0$ because the size of the elliptical component depends on $\mu = a_1^1 + b_1^2$ as in (4.27). Therefore, from (4.28) and (4.29), we have

$$\begin{aligned} A &= a_1 \left| \frac{\Omega}{\Omega^2 - \omega^2} \right| \\ B &= a_1 \left| \frac{\omega}{\Omega^2 - \omega^2} \right| \end{aligned}$$

and hence

$$\Omega = e\,|\omega| = \frac{A}{B}\,|\omega|\,. \tag{4.47}$$

We now have

$$\left|A^2 - B^2\right| = a_1^2 \left| \frac{\Omega^2}{\left(\Omega^2 - \omega^2\right)^2} - \frac{\omega^2}{\left(\Omega^2 - \omega^2\right)^2} \right| = \frac{a_1^2}{\left|\Omega^2 - \omega^2\right|}$$

and hence we can solve for a_1 as

$$a_1 = \sqrt{\left|\left(A^2 - B^2\right)\left(\Omega^2 - \omega^2\right)\right|} = \frac{\nu}{C}\sqrt{\left|\left(A^2 - B^2\right)\left(\frac{A^2}{B^2} - 1\right)\right|}. \tag{4.48}$$

We have answered the second part of Design Problem 4.1. Given A, B, C, and ν, we can solve for ω using (4.46), Ω using (4.47), and a_1 using (4.48).

We must now determine under what conditions a solution exists. First, note that we cannot have $A = B$ (i.e., q cannot be a circle), because this would imply that

$\omega = \Omega$ and by Corr. 4.3 this would lead to unbounded motion. Consider, however, the case $A = B(1+\epsilon)$, $0 < |\epsilon| \ll 1$, for which (4.48) yields

$$\begin{aligned} a_1 &= \frac{\nu}{C}\sqrt{\left|\left(B^2(1+\epsilon)^2 - B^2\right)\left(\frac{B^2(1+\epsilon)^2}{B^2} - 1\right)\right|} \\ &= \frac{\nu}{C}B\left|2\epsilon + \epsilon^2\right| \\ &\approx 2|\epsilon|\frac{\nu}{C}B. \end{aligned}$$

That is, to achieve $A \approx B$, the amplitude of speed oscillations must be very small. In particular, when B and C are of similar magnitudes, we have $a_1 \ll \nu$, i.e., the speed oscillation amplitude may be below the noise floor of the hardware. In theory, however, the formation shape, q, may be arbitrarily close to circular.

Our design needs to satisfy the constraint that the speed, α, must always be positive and therefore we must have $a_1 < \nu$. From (4.48), we have

$$a_1^2 = \frac{\nu^2}{C^2}\left|\left(A^2 - B^2\right)\left(\frac{A^2}{B^2} - 1\right)\right| < \nu^2, \tag{4.49}$$

or, solving in terms of A, B, and C,

$$\frac{|A^2 - B^2|}{B} < C. \tag{4.50}$$

Fig. 4.11 shows the range of values of A and B for which (4.50) is true when $C = 10$. As the parameters get closer to violating the bound (4.50), the speed oscillation amplitude becomes closer to 1. This may be problematic for vehicles that have a lower bound on their speed (flying vehicles, for example). Note also that the size of the formation can be very large. However, from (4.49) we have

$$\left|\left(\frac{A^2}{C^2} - \frac{B^2}{C^2}\right)\left(\frac{A^2}{B^2} - 1\right)\right| < 1$$

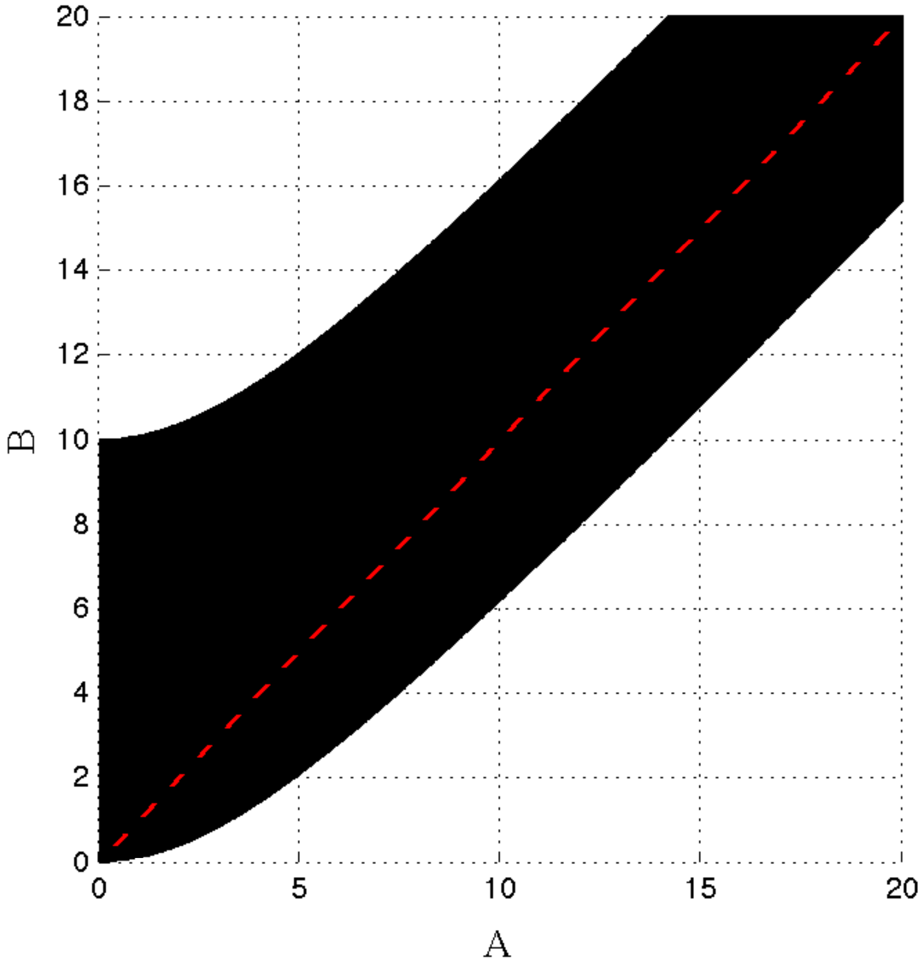


Figure 4.11: Range of values (shaded black) for which (4.50) is true, and therefore Design Problem 4.1 has a solution, when $C = 10$ over the range $0 < A < 20$ and $0 < B < 20$. Note that $A = B$ (dashed red line) is also not possible because this would lead to unbounded motion as per Corr. 4.3.

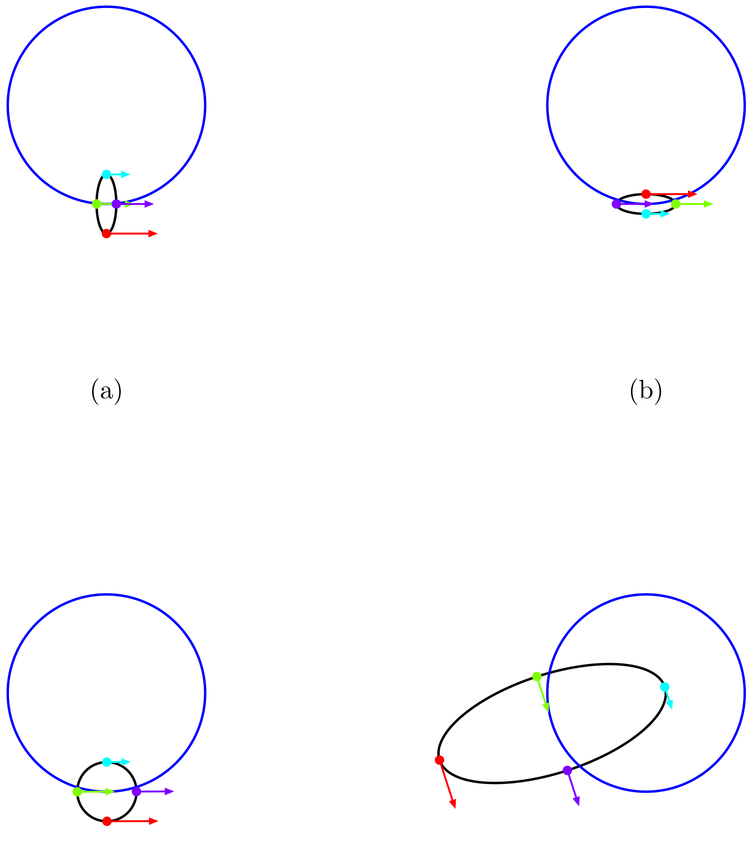
and we see that for large formations (i.e., $A \gg C$ and/or $B \gg C$), the aspect ratio must be close to 1 for feasibility. As discussed above, this is theoretically feasible but it may be difficult to stabilize and maintain large formations in realistic situations because it requires vanishingly small speed oscillations.

Fig. 4.12 shows four example solutions to Design Problem 4.1. In all four cases, the nominal speed and radius of the circular component of the trajectory are identical: $\nu = 1$ and $C = 10$, respectively. Therefore, the turning rate for all four cases is

$\omega = 0.1$ (for counter-clockwise motion). Fig. 4.12a and Fig. 4.12b show solutions for ellipses with aspect ratios that are inverses of each other. Fig. 4.12c shows an example where the aspect ratio is close to 1. Note that the resulting value of $a_1 = 0.0002$ is very small compared to $\nu = 1.0$ and the numerical value of $\Omega = 0.09967$[†] is very close to the numerical value of $\omega = 0.1$. The values of A and B in Fig. 4.12d are close to the feasible limit (4.50) (i.e., the pair (A, B) is close to the edge of the dark region in Fig. 4.11).

We now consider the general periodic speed case. The solution to this problem depends on the way that we choose to specify the formation shape (i.e., the "input"). Here, we will consider the formation shape to be provided, and we would like to minimize in some sense (which we will describe below) the error between the obtained formation shape q and the desired formation shape. We allow the desired formation shape to be prescribed as a sequence of M points $q'[m]$, $m = 1, \ldots, M$, that lie on or close to (allowing for error) the desired formation shape. The goal of the processes we describe below is to find speed oscillation parameters so that the obtained formation shape, q, approximates the set of points q'.

[†] Extra digits of precision are added here to emphasize the small numerical differences.



(a) (b) (c) (d)

Figure 4.12: Formations designed by solving Design Problem 4.1. In each case, the nominal speed is $\nu = 1$ and the radius of the circular component is $C = 10$ which results in $\omega = 0.1$. Only the circular component (blue circle) and the q shape (black curve) are shown; no individual trajectories are shown. (a) $A = 1$, $B = 3$, yielding $a_1 = 0.267$ and $\Omega = 0.1$. (b) $A = 3$, $B = 1$, yielding $a_1 = 0.8$ and $\Omega = 0.3$. (c) $A = 3$, $B = 3.001$, yielding $a_1 = 0.0002$ and $\Omega = 0.09967$ (see discussion above). (d) $A = 5$, $B = 12$, yielding $a_1 = 0.992$ and $\Omega = 0.042$.

One way to input the set of points q' that describes the desired formation shape is to provide a binary image and map the "on" pixels to points in the set q'. This allows us to draw the desired shape of the formation using a computer program. The mapping from the image to the point set q' can be accomplished as follows. Define the binary image $B : \{1, \dots, H_B\} \times \{1, \dots, W_B\} \to \{0, 1\}$ where H_B and W_B are the height and width of the image in pixels, and invertible mappings $Y : \{1, \dots, H_B\} \to \mathbb{R}$ and $X : \{1, \dots, W_B\} \to \mathbb{R}$ that map pixel locations to real world coordinates. Then the points q' can be formed using the following algorithm.

```
$q' \leftarrow \{\emptyset\}$                                ▷ Initialize $q'$ as empty
for $a = 1 \to H_B$ do
    for $b = 1 \to W_B$ do
        if $B(a, b) = 1$ then
            $q' \leftarrow q' \bigcup (X(b) + iY(a))$       ▷ Add this point to $q'$
        end if
    end for
end for
```

The number of points M is equal to the number of "on" pixels in the binary image, B. The ordering of the pixels does not affect the algorithm described below.

The coordinate mapping used to obtain the results below is a simple shifting and scaling of the image coordinates to formation coordinates. The centroid, $(\bar{m}, \bar{n})$ of the input image is calculated as

$$\bar{m} = \frac{1}{M} \sum_{(n,m)} mB(n, m)$$

and

$$\bar{n} = \frac{1}{M} \sum_{(n,m)} nB(n, m).$$

The coordinate mapping is then

$$X(m) \triangleq \gamma\,(m - \bar{m})$$

and

$$Y(n) \triangleq \gamma\,(n - \bar{n})\,,$$

where $\gamma > 0$ is a scaling factor. Using this coordinate mapping ensures that the centroid of the image maps to the center of the designed formation.

Given the points q' and the parametric form of q as it relates to the motion parameters, the problem of finding those parameters reduces to a curve fitting problem. We provide a simple iterative least-squares solution here that seeks to minimize the sum of squared distances between each point in q' and its closest point in the curve q. Let

$$\psi \triangleq \begin{pmatrix} a_1 \\ b_1 \\ \vdots \\ a_{N_\ell} \\ b_{N_\ell} \end{pmatrix},$$

where N_ℓ is the number of Fourier components in our solution, and let $q_\psi(\phi)$ be the curve calculated from Thm. 4.5 with the Fourier coefficients ψ. Our goal is to minimize the cost function

$$J(\psi) = \frac{1}{2} \sum_{m=1}^{M} \left\| q'[m] - q_\psi(\phi_m) \right\|^2 \tag{4.51}$$

where the m^{th} fitting phase, ϕ_m, is the value of $\phi \in S^1$ that minimizes $\|q'[m] - q_\psi(\phi)\|$.

We can now precisely state the design problem for the periodic speed profile case. As above, we assume that the nominal speed, ν, is fixed by operational constraints.

Similarly, we assume that the number of harmonic components, N_ℓ, in the speed profile is fixed.

Design Problem 4.2 (Formation Design - Periodic Speed Profile)**.** *Consider a group of N agents under the assumptions of Thm. 4.6. All agents are assumed to have the same periodic speed profile, which can be specified in terms of its Fourier coefficients as*

$$\alpha(\phi) = \nu + v(\phi) = \nu + \sum_{\ell=1}^{N_\ell} \left(a_\ell \cos \ell\phi + b_\ell \sin \ell\phi \right), \tag{4.52}$$

where N_ℓ is the number of harmonic components in the speed profile.

Given the nominal speed ν, the desired direction of motion, the desired radius of the circular component of the trajectory, C, the number of Fourier components in the speed profile, N_ℓ, and the desired formation shape as a set of points, q', determine

- *if the formation is physically achievable, and, if so*
- *what combination of turning rate, ω, speed oscillation frequency, Ω, and Fourier coefficients, ψ, minimizes the sum of squared distances cost function (4.51).*

Note that the cost function, $J(\psi)$, depends recursively on the parameters ψ via the fitting phases ϕ_m. That is, the distances are computed at values of ϕ that minimize the distance to the fitted curve. Therefore, we seek an iterative solution where the cost at the n^{th} step is calculated based on the phases that minimize the distance to the $(n-1)^{th}$ fit; i.e., we define the cost at the n^{th} iteration as

$$J(\psi[n]) = \frac{1}{2} \sum_{m=1}^{M} \left\| q'[m] - q_{\psi[n]}(\phi_m) \right\|^2$$

where each ϕ_m is chosen to minimize $\left\| q'[m] - q_{\psi[n-1]}(\phi_m) \right\|$. For convenience, denote the Fourier coefficients in the parameter set $\psi[n]$ as $a_\ell[n]$ and $b_\ell[n]$, $\ell = 1, \ldots, N_\ell$.

We use the statistics of the sequence q' to determine the speed oscillation frequency, Ω, and the initial estimate $\psi[1]$. The input sequence, q', is assumed to be oriented with $\theta = 0$ and therefore we can estimate the distribution of the q' points as an ellipse with semi-major and semi-minor axes A and B, respectively, where

$$A \triangleq \left(\frac{2}{M-1} \sum_{m=1}^{M} \left(\operatorname{Re}\{q'[m]\} - \operatorname{Re}\{\bar{q}'\} \right) \right)^{\frac{1}{2}} \tag{4.53}$$

$$B \triangleq \left(\frac{2}{M-1} \sum_{m=1}^{M} \left(\operatorname{Im}\{q'[m]\} - \operatorname{Im}\{\bar{q}'\} \right) \right)^{\frac{1}{2}} \tag{4.54}$$

and

$$\bar{q}' \triangleq \frac{1}{M} \sum_{m=1}^{M} q'[m]. \tag{4.55}$$

As above, the turning rate ω depends only on the radius C and the direction of motion, hence ω is determined by (4.46). We then take $\frac{A}{B}$ as the aspect ratio and set

$$\Omega = \frac{A}{B} |\omega| \, . \tag{4.56}$$

Also as above, we are free to choose $b_1[1] = 0$. The initial estimate $a_1[1]$ is found by (4.48) with the A and B calculated from (4.53) and (4.54), respectively.

For the purposes of solving this optimization problem numerically, it is useful to switch from complex notation for q to vector notation, i.e., $q = x + iy \in \mathbb{C}$ becomes $q = \begin{pmatrix} x & y \end{pmatrix}^T \in \mathbb{R}^2$. We then define

$$\Phi_\ell(\phi) \triangleq \frac{1}{(\ell\Omega)^2 - \omega^2} \begin{bmatrix} \ell\Omega \sin \ell\phi & -\ell\Omega \cos \ell\phi \\ \omega \cos \ell\phi & \omega \sin \ell\phi \end{bmatrix}$$

and

$$\psi_\ell[n] \triangleq \begin{pmatrix} a_\ell[n] \\ b_\ell[n] \end{pmatrix},$$

so that the formation shape for the n^{th} iteration is $q_{\psi[n]}(\phi) = \sum_{\ell=1}^{N_\ell} q_{\psi[n],\ell}(\phi)$, where

$$q_{\psi[n],\ell}(\phi) = \Phi_\ell(\phi)\psi_\ell[n].$$

Then

$$q_{\psi[n]}(\phi) = \sum_{\ell=1}^{N_\ell} \Phi_\ell(\phi)\psi_\ell[n] = \Phi(\phi)\psi[n],$$

where

$$\Phi(\phi) = \begin{bmatrix} \Phi_1(\phi) & \cdots & \Phi_{N_\ell}(\phi) \end{bmatrix} \in \mathbb{R}^{2\times 2N_\ell}.$$

and the cost function becomes

$$J(\psi[n]) = \sum_{m=1}^{N} \left\| q'[m] - \Phi(\phi_m)\psi[n] \right\|^2 = \left\| Q' - \boldsymbol{\Phi}\psi[n] \right\|^2, \tag{4.57}$$

where

$$Q' = \begin{bmatrix} q'[1] \\ \vdots \\ q'[M] \end{bmatrix} \in \mathbb{R}^{2M\times 1} \tag{4.58}$$

and

$$\boldsymbol{\Phi} = \begin{bmatrix} \Phi(\phi_1) \\ \vdots \\ \Phi(\phi_M) \end{bmatrix} \in \mathbb{R}^{2M\times 2N_\ell}. \tag{4.59}$$

The cost-minimizing parameters are found by taking the pseudoinverse:

$$\psi[n] = \left(\boldsymbol{\Phi}^T\boldsymbol{\Phi}\right)^{-1} \boldsymbol{\Phi}^T Q'. \tag{4.60}$$

Using these results we arrive at Algorithm 4.1. Algorithm 4.1 uses the initial parameter estimates $\psi[1]$ determined as described above to find initial fitting phases ϕ_m and solves for new parameters using (4.60). This process is iterated until a maximum number of iterations, n_{max}, is reached or until the parameters do not change in magnitude by more than a threshold, t_ψ.

Algorithm 4.1 Algorithm for solving Design Problem 4.2.

```
Given the initial parameter estimates, $\psi[1]$, as described above, maximum number
of iterations $n_{max}$, and convergence threshold $t_\psi$.
$n \leftarrow 2$
$\delta\psi \leftarrow \infty$
while $n \leq n_{max}$ and $\|\delta\psi\| < t_\psi$ do
    Find the $\phi_m$ that minimize $\left\|q'[m] - q_{\psi[n-1]}(\phi_m)\right\|$, $m = 1, \ldots, M$
    Calculate $\mathbf{\Phi}$ (4.59) and $Q'$ (4.58)
    $\psi[n] \leftarrow \left(\mathbf{\Phi}^T\mathbf{\Phi}\right)^{-1} \mathbf{\Phi}^T Q'$
    $\delta\psi \leftarrow \psi[n] - \psi[n-1]$
    $n \leftarrow n + 1$
end while
return $\psi[n]$
```

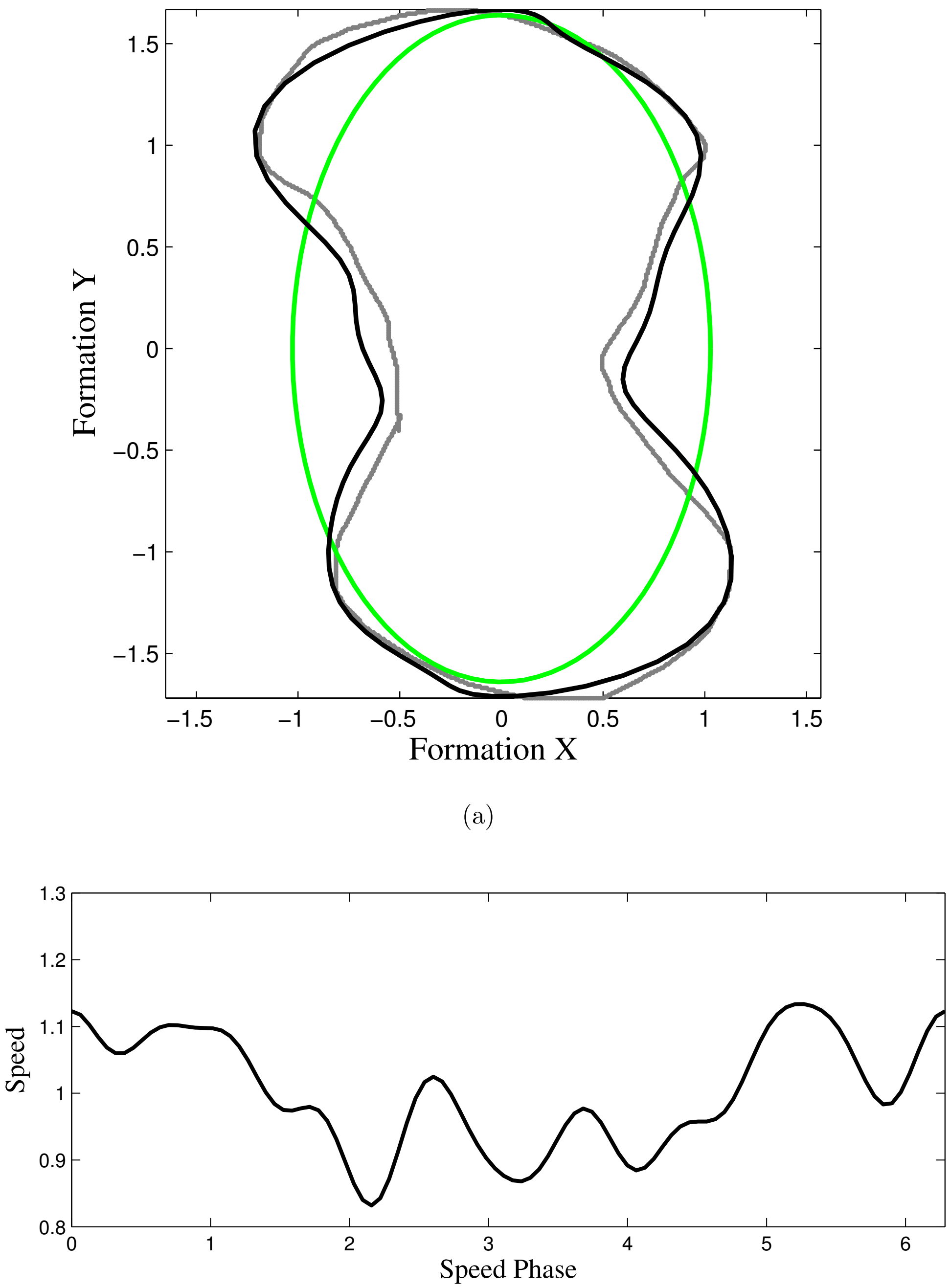
To determine if the formation is feasible, we first check that $A \neq B$ and then calculate the speed profile from (4.45) and check that $v(\phi) > -\nu$ for all $\phi \in S^1$. We may also deem the formation infeasible if a sufficient fit is not achieved within n_{max} iterations.

Figs. 4.13 and 4.15 show example solution to Design Problem 4.2. The input formation shape (gray curve in Fig. 4.13a and Fig. 4.15a) was generated by drawing the shape in a Paint-like program. The resulting bitmap was then loaded into a Matlab script that determines the q' points using the coordinate mappings described above with $\gamma = \frac{1}{120}$[‡], determines the aspect ratio and the speed oscillation frequency, and then determines the Fourier coefficients of the speed profile using Algorithm 4.1. The final speed profiles are shown in Figs. 4.13b and 4.15b. Fig. 4.14 shows five snapshots of four agents moving around the formation shown in Fig. 4.13 with their speed phases in the splay phase configuration. Fig. 4.16 shows the same but for the formation shown in Fig. 4.15. Note that each agent cycles its position around the formation as the formation moves around the common circular component (shown in blue) of

---

[‡] $\gamma = \frac{1}{120}$ scales one half of the height of a 640 × 480 pixel image to 2 formation units.

the trajectories. The formation shape obtained in Fig. 4.13 is a good approximation of the input shape. The fit is not as good in Fig. 4.15, though some features of the shape are preserved; particularly near the front and back of the formation.



(a)

(b)

Figure 4.13: Example formation found by solving Design Problem 4.2. The input formation shape is the gray curve in the top figure. The green curve is the ellipse corresponding to the initial parameter estimates. The black curve is the output formation (see also Fig. 4.14), corresponding to the speed profile shown in the bottom figure. The radius of the circular component of the trajectory is $C = 10$. The nominal speed is $\nu = 1$. The resulting turning rate is $\omega = 0.1$ and the speed oscillation frequency is $\Omega = 0.214$.

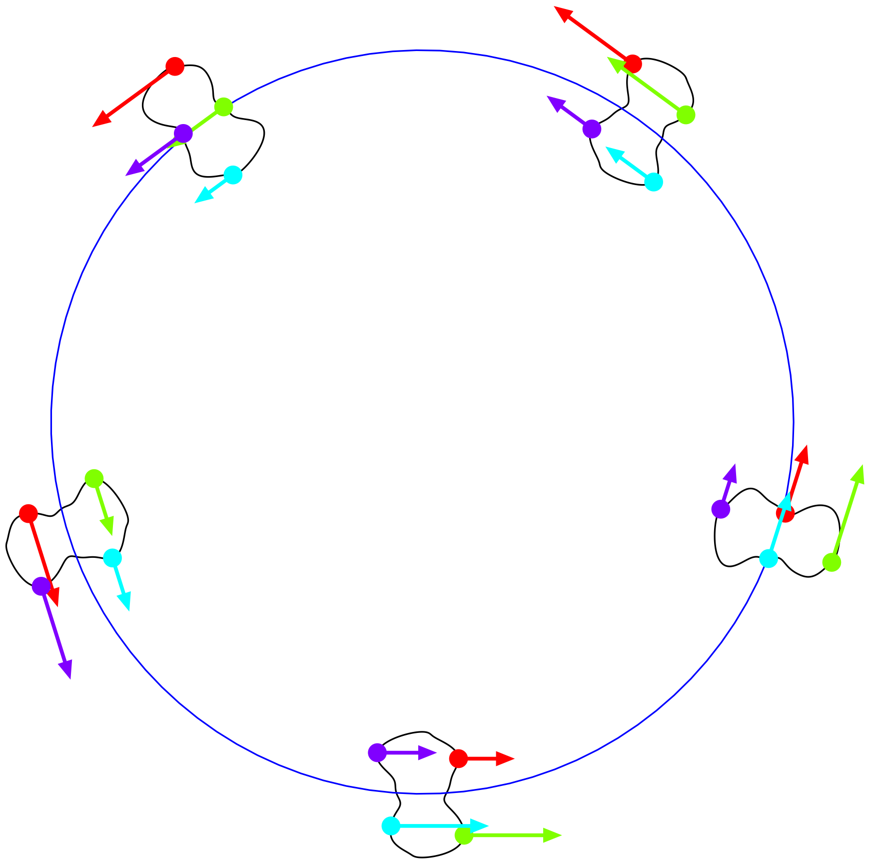
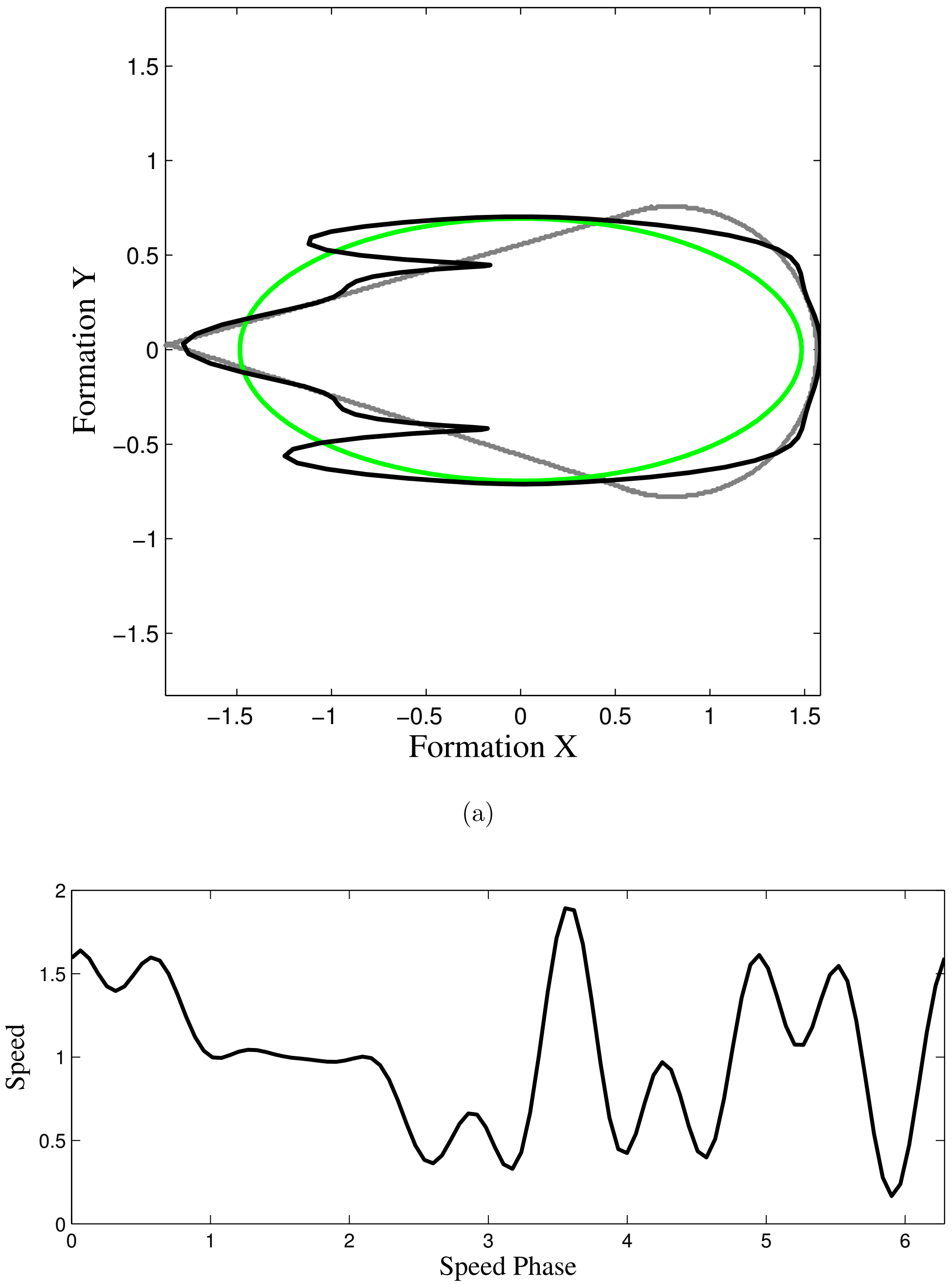


Figure 4.14: Five snapshots of four agents moving around the formation found by solving Design Problem 4.2 with the formation input shown in Fig. 4.13. The snapshots correspond to equally distributed times. The speed phases of the four agents are in the splay phase arrangement. The blue circle is the radius of the circular component of the trajectory of all of the particles and has a radius of $C = 10$.



(a)

(b)

Figure 4.15: Another example formation found by solving Design Problem 4.2. The input formation shape is the gray curve in the top figure. The green curve is the ellipse corresponding to the initial parameter estimates. The black curve is the output formation (see also Fig. 4.16), corresponding to the speed profile shown in the bottom figure. The radius of the circular component of the trajectory is $C = 10$. The nominal speed is $\nu = 1$. The resulting turning rate is $\omega = 0.1$ and the speed oscillation frequency is $\Omega = 0.214$.

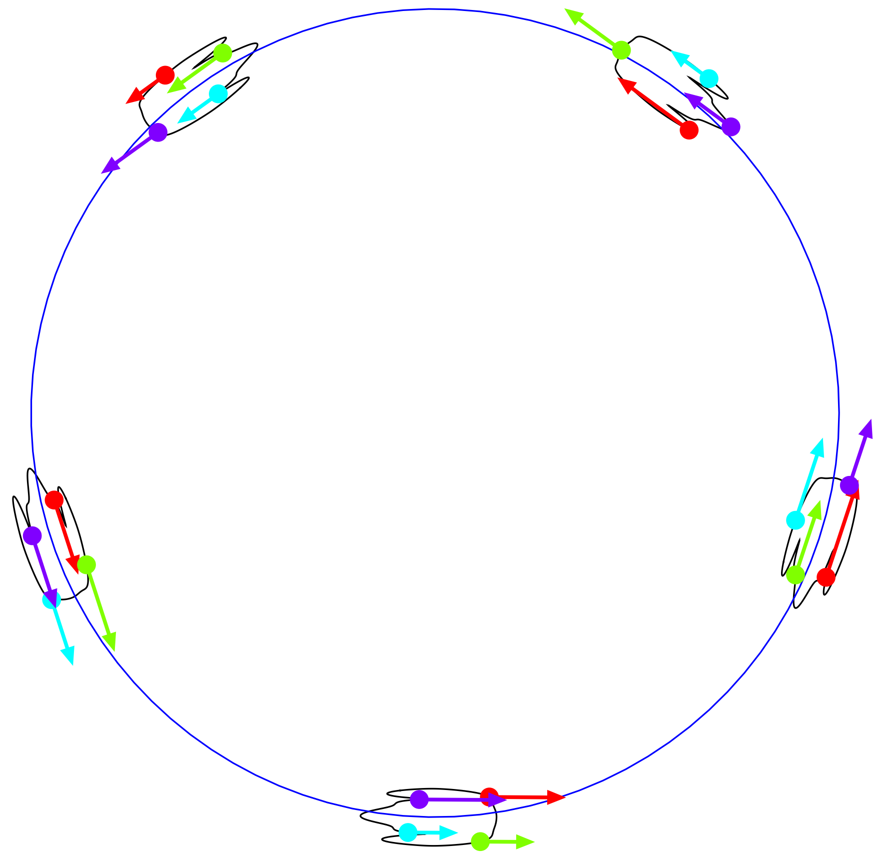


Figure 4.16: Five snapshots of four agents moving around the formation found by solving Design Problem 4.2 with the formation input shown in Fig. 4.15. The snapshots correspond to equally distributed times. The speed phases of the four agents are in the splay phase arrangement. The blue circle is the radius of the circular component of the trajectory of all of the particles and has a radius of $C = 10$.

Chapter 5

Hybrid Biological and Engineering Collective Motion Testbed

In this chapter, we describe a testbed that was designed to facilitate experiments in which one or more robotic fish is controlled in real-time to introduce a stimulus to a live fish school. Each robotic fish consists of a model fish that is magnetically coupled to a wheeled robot beneath the tank. Real-time computer vision is used to estimate the states and to estimate properties of the fish school. Control commands are calculated in response to these estimations and sent wirelessly to the wheeled robots, whose motion causes the model fish to move in tandem because of the magnetic coupling. We refer to the wheeled robot and model fish combination as a "robotic fish". Depending on experimental needs, the robotic fish can act as a predator fish, a conspecific, or even a reactive piece of the environment. This flexibility is afforded by the ability to easily change a) the shape of the model that is magnetically coupled to the wheeled robot and b) the algorithms that drive the wheeled robots in response to the state of the fish school. The design and description of the robotic fish testbed were first published in [115]. Figures in this chapter that first appeared in [115] are noted in the corresponding captions.

We describe the components of the testbed in Sec. 5.1. The hardware components are detailed in Sec. 5.1.1, including the arena, wheeled robots, model fish, and tracking system hardware. We provide an overview of the real-time tracking software architecture in Sec. 5.1.2.

In Sec. 5.2, we describe the real-time tracking methods that we use to track the robots and fish from the live video feed. We consider the tracking algorithm as a two-step iterative process. In the first step, we isolate regions of the image that are likely to correspond to fish (live or robotic) and extract from these regions measurements of the positions and orientations of the fish. We refer to this as the "segmentation" step and described it in Sec. 5.2.1. We refer to the second step as the "state estimation" step and describe it in Sec. 5.2.2. The state estimation step involves associating the measurements obtained in the segmentation step with the correct fish and then applying an unscented Kalman filter (UKF) to refine estimates of position, orientation, and speed. We also describe in Sec. 5.2.3 methods that are used to estimate properties of the fish school: for example, centroid location, size, boundaries, etc.

We discuss two experimental demonstrations of this testbed in Sec. 5.3. First, in Sec. 5.3.1, we describe an experiment in which the robotic fish continually follows the centroid of the school. In Sec. 5.3.2, we describe an experiment in which the robotic fish darts toward the school when triggered by the occurrence of some preprogrammed condition.

The technology that was developed for this testbed is directly applicable to a variety of other applications. These other applications have, in turn, motivated improvements to the technology used for the robotic fish testbed. We describe some of the extensions to other applications in Sec. 5.4. Sec. 5.4.1 enumerates some of the other testbeds that use our real-time tracking technology. In Sec. 5.4.2, we describe the MADTraC C++ library [112], which was designed to facilitate rapid development of these and other testbeds. In Sec. 5.4.3, we describe software components that were designed to enable interaction of the tracking and robot control software with other software platforms, for example, MATLAB-based control algorithms and web-based multi-user interfaces.

5.1 Testbed Description

We provide here a basic description of the testbed components and provide details of the hardware in Sec. 5.1.1 and an overview of the software in Sec. 5.1.2.

Fig. 5.1 shows the basic components of the testbed. The experimental arena consists of a shallow tank mounted on a base. The wheeled robots move about a platform that is hanging beneath the tank and is part of the base. Each robot is magnetically coupled to a model fish above it inside the tank. A FireWire camera is mounted above the tank and captures real-time video. Another camera is optionally mounted above the tank to record high definition video that can be used for off-line analysis. Real-time tracking and control calculations are performed on a computer workstation. Commands for each robot are computed on the workstation and sent to the robot using a BlueTooth radio, thus closing the control loop.

5.1.1 Hardware

We break the testbed hardware down into four main components here: the experimental arena, the wheeled robots, the model fish, and the tracking system hardware.

The Experimental Arena

The experimental arena consists of the tank and its base. The tank measures 4-foot by 5-foot by 12-inches high. The water in the tank is typically 2.5 to 3 inches deep. The high sides of the tank are designed to keep fish from leaping out of the tank.

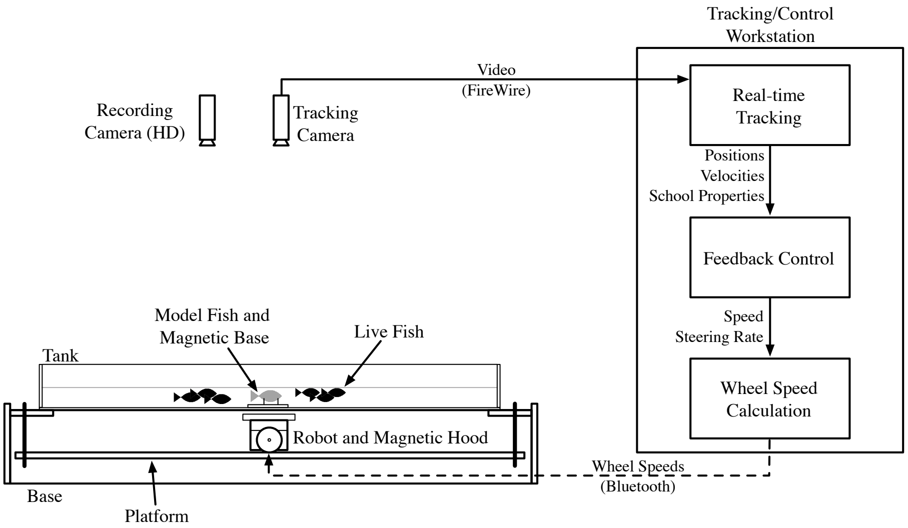


Figure 5.1: Diagram of the robotic fish testbed components. This figure first appeared in [115].

There are 12-inch diagonal pieces across each corner of the tank that add structural integrity and prevent fish from congregating in the corners of the tank (see Fig. 5.8). A small hole is drilled through the base of one diagonal piece to allow water into that corner, where a drain hole is drilled through the bottom of the tank and a valve and hose attached below.

The tank is made from Garolite G10 fiberglass composite. This material provides superior stiffness compared to plexiglass and allows the bottom piece of the tank to be only 1/4-inch thick. An earlier prototype used 1/4-inch plexiglass and exhibited severe bowing under the weight of the water in the tank (the tank bowed by several inches in the middle). Bowing is a significant problem for the testbed because it introduces variability in the distance — and therefore the coupling strength — between the wheeled robots and the model fish. With a G10 tank, there is minimal bowing under water weight and therefore consistent coupling between the wheeled robots and the model fish. The sides of the tank and the diagonal pieces are all made from 1/8-inch thick pieces of G10. The pieces of G10 are secured to one another using epoxy, resulting in a watertight seal.

The G10 tank is secured to a wooden base by bolts along the tank's periphery. The main purpose of the wooden base is to elevate the tank so that the robots can move beneath it. A veneered platform is suspended by adjusting bolts within the wooden base. The veneered platform provides a consistently flat surface for the robots to move on. Because it is mounted on adjusting bolts, the standoff distance between the wheeled robots and the model fish can be adjusted to account for various conditions such as bowing, robot height, and magnet strength. The nominal distance from the veneered platform to the bottom of the G10 tank is approximately 4 inches.

Wheeled Robots

We use Merlin Systems Corp. MiaBot Pro wheeled robots [77] to move the model fish. These wheeled robots are designed for robotic soccer. We chose them because they are wirelessly controlled via Bluetooth and have a high speed and turning rate (up to 3 m/s and 85 radians/s, respectively). Each robot has two wheels, mounted on opposite sides of a 7.5-cm cube. Internally, the robot consists of one motor for each wheel, a battery pack, and electronics. Plastic nubs on the bottom of the robot keep it from rocking forward or backward on the wheel axis. Fig. 5.2 shows a MiaBot Pro robot.

The wheels are parallel to the sides of the robot and its spin axis is through the middle of the cube. The speed of each wheel is controlled independently; commanding both wheels with the same positive (negative) speed commands causes the robot to move forward (backward), and commanding different speeds to each wheel causes it to turn. Each motor has an encoder that the robot uses for on-board proportional-integral-derivative (PID) feedback control of the wheel speeds. The on-board wheel speed control is sufficiently fast that we neglect the transient effects associated with commanding a change in wheel speeds.

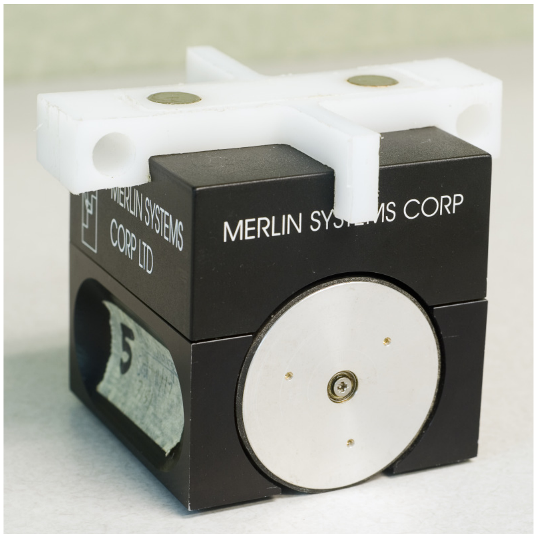


Figure 5.2: A MiaBot Pro wheeled robot with a pair of cylindrical rare-earth magnets press-fit into holes on a custom-made hood. This figure first appeared in [115].

The two-wheeled differential drive and high speed and turning rates of the MiaBot Pro make it well-suited to emulate the translation and turning movements of a fish, although it cannot reproduce body contortions or side-drifting. The MiaBot Pro is capable of accelerating rapidly to high speed, although wheel slip, latency in the closed-loop system, and drag forces acting on the model fish place practical limits on maximum acceleration and speed.

We mount a custom-machined plastic hood to the top of each robot, as shown in Fig. 5.2. Each hood has a rectangular slot milled into its top. The milled slot has a matching rectangular insert. Two holes are drilled through the insert and a pair of cylindrical rare-earth magnets are press-fit into the holes. The magnets are 1/2-inch in diameter, 1/4-inch thick, separated by 2 inches, and mounted such that they are centered on the robot and aligned front-to-back with the direction of motion. Other magnet sizes and configurations can be accommodated by creating a new insert, and the height of the magnets can be adjusted slightly by inserting shims beneath the insert. The circular faces of each magnet correspond to its two poles. The magnets are inserted into the rectangular insert with opposite pole orientations, which facilitates torque transfer to the model fish when the robot turns. This also ensures that the correct mating orientation is maintained. The wheeled robot can sometimes lose traction with the veneered surface due to the strong attraction between the magnets in the hood and the magnets in the tank above. Weights can be added to the hood to counteract this effect.

Model Fish

The appearance of the model fish can be an important factor in the success of an experiment. This can be true for predator models and is especially true for conspecific models. The reaction of the fish to the model is species-dependent. For example, three-spined sticklebacks (*Gasterosteus aculeatus L.*) are observed to respond well to models that have realistically painted eyes [126, 41]. We have thus far had the opportunity to experiment only with golden shiners (*Notemigonus crysoleucas*), and they have proved mostly unresponsive to models. A series of golden shiner models were developed by casting hard plastic into models created from deceased golden shiners. One of the golden shiner models is shown in Fig. 5.3a. Live golden shiners were mostly indifferent to these model golden shiners.

Live golden shiners are slightly more reactive to a model koi (*Cyprinus carpio haematopterus*) than they are to a model golden shiner. We created a model koi, shown in Fig. 5.3b, by casting tin-cured silicone rubber into a mold created from a deceased koi. The model koi is larger than the live golden shiners we used. When we used the model koi, we used it in the role of a predator fish. The experimental demonstrations described below in Sec. 5.3 were performed using the model koi. The model koi is shown in use in Figs. 5.8 and 5.9.

The molding material we used to create the golden shiner and koi molds is tin-cured silicone rubber (MoldMax 30T). The model koi was created by casting a different formula of silicone rubber (MoldMax 15T) in the mold. The molding and model

(a) (b)

Figure 5.3: Model fish. (a) Model golden shiner (top) produced by casting hard plastic in a mold created from a deceased golden shiner (bottom). The model has been painted to match the appearance of the deceased fish. (b) Model koi produced by creating casting silicone in a mold created from a deceased koi (not shown). The model has been painted black, and is shown mounted on its magnetic base. The photo shown here was taken using an underwater camera in the tank. This figure first appeared in [115].

materials are distributed by Smooth-On, Inc. [106]. A limited variety of specialty paints are available for painting silicone rubber, and therefore the model koi was painted all in black. Silicone rubber is flexible and is well-suited to recreating fine details in the model. The model golden shiners were created by casting a standard hard rubber in a tin-cured silicone rubber mold. Hard plastic does not reproduce fine details in the model, but does allow for more realistic painting.

Tracking System Hardware

The tracking system consists of a tracking workstation and a FireWire camera mounted above the tank. The camera is an Allied Vision Technologies Guppy F-080 grayscale FireWire camera [1] with 1032 × 778 resolution and a 4.5 mm focal length lens. The tracking workstation consists of a single computer with modern commodity hardware. The only requirement for the computer is that it must have a FireWire port for the camera. If the computer does not have built-in Bluetooth, an off-the-shelf USB Bluetooth adapter can be used to communicate with the robotic fish.

The camera is mounted 3 m above the tank and aligned so that the image plane is close to parallel with the plane of the tank bottom. The image plane is oriented so that its x-axis is parallel to the long edge of the tank. As a result, we achieve sufficient performance from calculating feedback control for the robotic fish using image coordinates (as opposed to first converting to real-world coordinates). We do not correct for refraction due to Snell's law because the depth of the water is small compared to the distance between the tank and the camera. Calibration of the HD camera is necessary for off-line analysis so that experimental results can be reported accurately in real-world units.

5.1.2 Software

We describe here the software that we have developed for this testbed. The software is built using the MADTraC (Multi-Agent Dynamic Tracking and Control) C++ library, which is described in further detail in Sec. 5.4.2. MADTraC provides a GUI framework and several capabilities that are useful for developing software to drive testbeds like this one. The real-time tracking algorithm is an important component of the software driving this testbed and is described in detail in Sec. 5.2. In the rest of the current section, we provide a description of the communications and control components of the software.

The workstation communicates with the MiaBot Pro wheeled robot through a virtual serial port over a Bluetooth channel. Communication with multiple robots is possible by opening a separate port for each robot. The Bluetooth specification sets a limit of seven devices that can be connected to a single host [13], and therefore up to seven robots can be used simultaneously. This limit can be overcome with specialty hardware.

The MiaBot Pro platform provides a small grammar of commands that can be sent to the robot using an ASCII protocol over the virtual serial port [76]. When controlling the robots for use in this testbed, we use the wheel speed command. The wheel speed command has the form `[=LL,RR]`, where `LL` and `RR` are the left and right wheel speeds (respectively) in robot-specific units that can vary in value from -2000 to 2000. Positive values correspond to motion in the forward direction of the robot and negative values correspond to motion in the reverse direction.

We apply feedback laws to the robots that calculate wheel speed commands in response to the states of the robotic fish and the live fish school. These feedback laws can be used to model conspecific or heterospecific fish behavior, and generally involve calculating a desired robot speed and turning rate which is then converted into wheel speed commands. For example, to apply feedback laws like those discussed in Ch. 4 that can be viewed as models of schooling behavior, we calculate the speed and turning rate of the robotic fish as a function of the relative positions and speeds of the other fish.

To illustrate the control techniques used for this testbed, we consider here a model of predator behavior in which the robotic fish moves continuously toward the estimated centroid location of the fish school. Let $(x(t), y(t))$ be the position of the robot in the plane at time t, and let $s(t)$ and $\theta(t)$ denote its speed in cm/s and heading relative to the $x-$axis in radians, respectively. Let $\omega(t) = \frac{d\theta}{dt}$ be the robot's steering rate in radians/s. Then, the wheel speed commands `LL` and `RR` are calculated as

$$\begin{bmatrix} \texttt{LL} \\ \texttt{RR} \end{bmatrix} = \frac{k_e}{50} \begin{bmatrix} 1 & -\frac{L}{2} \\ 1 & \frac{L}{2} \end{bmatrix} \begin{bmatrix} s(t) \\ \omega(t) \end{bmatrix},$$

where $L = 0.07$ m is the robot wheelbase, $k_e = 4.0 \times 10^{-5}$ is a scaling factor to convert meters to wheel encoder counts, and the factor of 50 accounts for the robot's internal software's scaling of encoder units.

Now, let $(x'(t), y'(t))$ be the estimated location of the centroid of a fish school in the plane. To "attack" the school, we want to drive the robot from $(x(t), y(t))$ to $(x'(t), y'(t))$ while continually updating our estimates of both locations. This can be accomplished by prescribing the steering control law

$$\omega(t) = -k \sin\left(\theta(t) - \psi(t)\right)$$

where $k > 0$ is a constant gain and

$$\psi(t) = \tan^{-1} \frac{y(t) - y'(t)}{x(t) - x'(t)}$$

is the bearing to the target. The controlled heading dynamics have a stable solution for which the heading angle and target bearing are synchronized, i.e., $\theta(t) = \psi(t)$, and an unstable solution for which the two angles are antisynchronized, i.e., $\theta(t) = \psi(t) + (2n+1)\,\pi$ for some integer n. We use a saturating speed control that is calculated as

$$s(t) = \begin{cases} s_{max}, & d(t) > d^* \\ s_{max}\frac{d(t)}{d^*}, & d(t) \leq d^* \end{cases},$$

where $s_{max} < 3$ m/s is the maximum desired approach speed,

$$d(t) = \sqrt{\left(x(t) - x'(t)\right)^2 + \left(y(t) - y'(t)\right)^2}$$

is the distance between the robot and the centroid of the school, and d^* is a threshold distance. Using this speed control, the robot moves at a fixed constant speed, s_{max}, at sufficiently large distances from the school. Once it is within the threshold distance, d^*, the robot slows linearly with distance as it approaches the school.

The above example is a very simple control law for driving the robot from one location to another when the angle of approach is unspecified, and has worked well for us — for example, in the experimental demonstrations described in Sec. 5.3 below. More sophisticated motion planning and control algorithms for nonholonomic vehicles can be substituted. The software architecture allows for easy implementation of alternate control schemes.

5.2 Real-Time Tracking

In this section, we describe the algorithms we use to extract estimates of the states of the robotic fish and individual live fish from live video sequences. We also describe methods that we use to estimate group-level properties of the fish school. In order to to be able to control each robotic fish, we must be able to estimate its position and velocity in real time. This requires a tracking solution that is both computationally efficient and able to estimate dynamic states from static measurements. Furthermore, because multiple robotic fish may share similar appearances and because there may be a large number of real fish moving in the tank, our tracking solution must be able to handle a great deal of occlusion without sacrificing too much in computational cost. It is for these reasons that we have implemented our solution in C++ using the OpenCV computer vision library [16] to perform individual steps whenever possible. We further reduce latency by integrating tracking and robot communication and control into the same piece of compiled software.

We break down our discussion of tracking methods into two conceptual steps: segmentation and state estimation. Segmentation involves locating features of interest in the image and extracting measurements from them. State estimation involves incorporating these measurements into a dynamical and statistical model in order to produce filtered estimates of the dynamic states of individual fish. The process of assigning measurements to individual tracked fish is a non-trivial problem called data

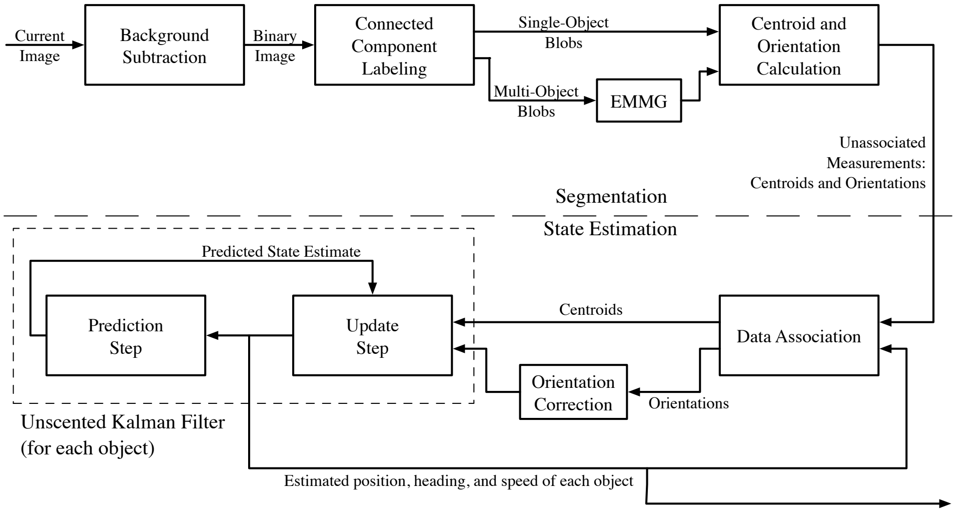


Figure 5.4: Diagram of the segmentation and state estimation steps of tracking. This figure first appeared in [115].

association. We treat data association here as a part of the state estimation step. See Fig. 5.4 for an overview of the tracking steps.

5.2.1 Segmentation

In this section, we describe the segmentation step of our real-time tracking algorithm. The input for the segmentation step at time instant t is the current video frame, which we label $I(t)$. The output of the segmentation step at time instant t is an array of measurements consisting of centroids and orientations of fish that were found in the video frame. The segmentation step breaks down into two sub-steps: image processing and blobbing. First, we describe the image processing sub-step, in which we locate pixels in the current video frame that are likely to belong to the image of a fish. Second, we describe the blobbing sub-step, in which we group pixels together that belong to the image of a single fish and then extract measurements of the location and orientation of that fish.

Image Processing

We will refer to the $(i, j)^{th}$ pixel of an image with subscripts, i.e., the value of the pixel in row j and column i of $I(t)$ is $I_{j,i}(t)$[*]. We will also refer to the $(i, j)^{th}$ pixel as the pixel at location (i, j). The location $(0, 0)$ is at the top-left corner of the image. We are concerned here only with grayscale images for which the pixel value increases with brightness, i.e., if $I_{j,i}(t) > I_{j',i'}(t)$ then the $(i, j)^{th}$ pixel of $I(t)$ is brighter than the $(i', j')^{th}$ pixel of $I(t)$.

There is a high degree of contrast between the appearance of the fish (including the robotic fish) and the appearance of the tank. Namely, the fish are significantly darker than the tank. Additionally, the appearance of the tank and its surroundings remains constant over the course of an experiment. Therefore, we are able use a simple background subtraction and thresholding algorithm to successfully isolate pixels that have a high likelihood of belonging to a fish from those that are more likely to belong to the background (the tank, its fixtures, and its exterior). The first step in the background subtraction process is to calculate the difference image, $D(t)$, by subtracting $I(t)$ from the background image, B. That is, we calculate $D(t) = B - I(t)$, where B is the background image and is typically computed by averaging several images of the empty arena prior to an experiment. Pixels in $I(t)$ that are darker than the background, B, have smaller values and therefore $D(t) > 0$ for those pixels. We then create a threshold image, $T(t)$, by setting any pixels of $D(t)$ that are less than a threshold value, T_B, to zero and the rest to 255:

$$T_{j,i}(t) = \begin{cases} 0, & D_{j,i}(t) < T_B \\ 255, & D_{j,i}(t) \geq T_B \end{cases} = \begin{cases} 0, & B_{j,i} - I_{j,i}(t) < T_B \\ 255, & B_{j,i} - I_{j,i}(t) \geq T_B \end{cases}.$$

---

[*] This ordering of the subscripts is consistent with our notational convention for matrices, which is itself intended to be consistent with MATLAB's syntax. That is, $M(j, i)$ is the value of the element of M in the j^{th} row and i^{th} column.

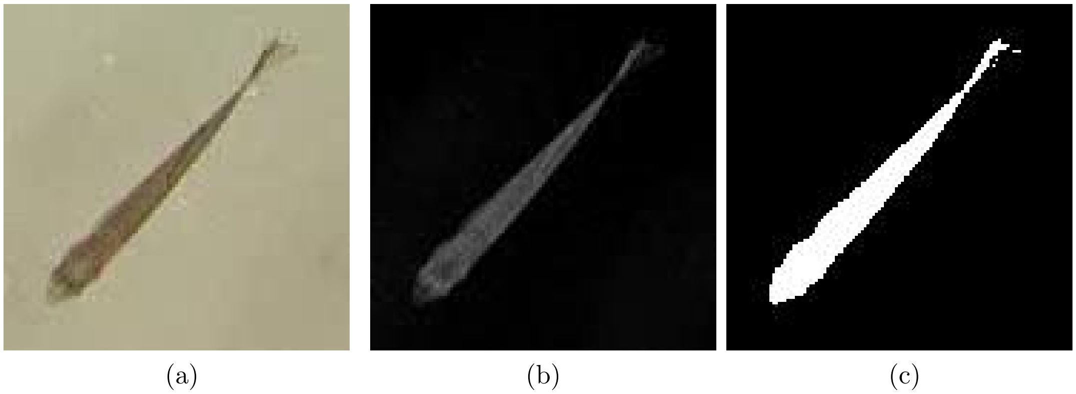

(a) (b) (c)

Figure 5.5: Background subtraction and thresholding applied to an image of a fish. (a) The original image, I. (b) The difference image, D. The background image, B, is not shown. (c) The threshold image, T. The threshold was $T_B = 25$.

The threshold value is determined experimentally. For an image that varies over the range $[0, 255]$, a value of $T_B = 25$ is reasonable. The background image and threshold should be recalculated if lighting conditions change significantly in the laboratory. Using the values 0 and 255 in the threshold image allows us to display it directly as a standard grayscale image, where white pixels are the ones that are likely to belong to the image of a fish. Note that most image processing libraries (including OpenCV) provide an image subtraction routine that truncates negative values to zero in the result. This is acceptable because the threshold, T_B, is positive and therefore the resulting threshold image, $T(t)$, is unaffected by the truncation.

Fig. 5.5 shows an example of these image processing steps applied to an image of a single fish.

Blobbing

The second sub-step of our segmentation algorithm is to group pixels from the threshold image, $T(t)$, into *blobs*. A blob is a set of pixels that represents the image of one fish. This process is complicated by the fact that the images of multiple fish often overlap because they are swimming close to one another. We solve this problem using a combination of two methods. First, we label *connected components* (a notion that we make precise below) in the threshold image, $T(t)$, using an efficient connected component labeling algorithm. We then use area and perimeter thresholds to determine which connected components are likely to contain a single fish and which connected components are likely to contain more than one fish. We extract measurements directly from single-fish connected components using a moment-based method that models the fish's appearance as an elongated ellipse. We resolve a multi-fish connected component using an expectation-maximization mixture-of-gaussian (EMMG) algorithm to fit the connected component to n ellipses, where n is the number of fish whose images are contained in that connected component. This process is described in detail below. We begin by precisely defining an image connected component. We then describe how the image of a fish can be modeled as an elongated ellipse using *image moments*, and how we use this method to extract measurements of position and orientation. Finally, we describe the EMMG algorithm for resolving multi-fish connected components.

Distinct pixels (i, j) and (i', j') are said to be *8-connected* at time instant t if $|j' - j| \leq 1$, $|i' - i| \leq 1$, and $T_{j,i}(t) = T_{j',i'}(t) > 0$. Distinct pixels (i, j) and (i', j') are said to be *4-connected* at time instant t if either $|j' - j| = 1$ or $|i' - i| = 1$ but not both, and $T_{j,k}(t) = T_{j',i'}(t) > 0$. That is, 8-connected pixels are adjacent either on a side or diagonally, and 4-connected pixels are adjacent on a side but not diagonally. A set of pixels that are mutually 8-connected (4-connected) is called an 8-connected component (4-connected component). We are concerned here with 8-connected components and will refer to them simply as *connected components*.

We extract connected components from the threshold image, $T(t)$, using an efficient (linear time in the number of pixels scanned) connected component labeling algorithm based on the one implemented in OpenCV [16] and first presented by Chang, Chen, and Lu [26]. Our implementation provides calculations of the area (number of pixels) and perimeter (number of pixels on the boundary) of each connected component. We reject connected components whose areas are less than a predetermined threshold. The areas and perimeters of the remaining connected components are compared to predetermined thresholds to determine how many fish' images are represented by each connected component.

We use image moments to approximate the shape of a fish in a single-fish connected component as an elongated ellipse. Image moments were introduced by Hu in 1962 [51] and have been widely used in computer vision applications since; see the review by Prokop and Reeves [99]. Consider a set of n pixels, $O = \{(x_1, y_1), \ldots, (x_n, y_n)\}$. When O is a connected component that corresponds to the image of a single fish, we say that O is that fish's *blob*.

The $(j, l)^{th}$ image moment, $M_{j,l}$, of O is defined as

$$M_{j,l} \triangleq \sum_{(x,y)\in O} x^j y^l = \sum_{k=1}^{n} x_k^j y_k^l.$$

The $(0, 0)^{th}$ moment is the number of pixels in O and therefore is equal to its area. The centroid, (x_{cm}, y_{cm}), of O is

$$(x_{cm}, y_{cm}) \triangleq \left(\frac{M_{1,0}}{M_{0,0}}, \frac{M_{0,1}}{M_{0,0}} \right). \tag{5.1}$$

The centroid of a fish is typically closer to the fish's head than it is to the tail, because the head is wider than the tail.

The *central moments* of O are important quantities because they provide a useful set of shape-invariant quantities [99]. The $(j, l)^{th}$ central moment, $\mu_{j,l}$, is defined as

$$\mu_{j,l} \triangleq \sum_{x,y\in O} \left(x - x_{cm}\right)^j \left(y - y_{cm}\right)^l.$$

The shape of O can be approximated as an ellipse centered at (x_{cm}, y_{cm}) and with semi-major axis

$$\left[\frac{2\left(\mu_{2,0}+\mu_{0,2}+\sqrt{\left(\mu_{2,0}-\mu_{0,2}\right)^2+4\mu_{11}^2}\right)}{\mu_{0,0}}\right]^{\frac{1}{2}} \tag{5.2}$$

and semi-minor axis

$$\left[\frac{2\left(\mu_{2,0}+\mu_{0,2}-\sqrt{\left(\mu_{2,0}-\mu_{0,2}\right)^2+4\mu_{11}^2}\right)}{\mu_{0,0}}\right]^{\frac{1}{2}}. \tag{5.3}$$

The orientation angle of the semi-major axis of the ellipse is

$$\theta_m = \frac{1}{2}\tan^{-1}\frac{2\mu_{1,1}}{\mu_{2,0}-\mu_{0,2}}, \tag{5.4}$$

where a four-quadrant arctangent must be used (e.g., the function called `atan2` in both Matlab and C). We call this ellipse the *moment ellipse* of O.

When the set O is the blob of a single fish, we take (x_{cm}, y_{cm}) as a measurement of the location of that fish. Using θ_m to measure the orientation of the fish leads to an ambiguity: does θ_m point toward the head of the fish or does θ_m point toward the tail of the fish (in which case $\theta_m + \pi$ points toward the head)? This ambiguity results from the shape-invariance of $\mu_{1,1}$, $\mu_{2,0}$, and $\mu_{0,2}$ with respect to mirroring O about the ellipse's axes.

To discriminate between the head and tail directions, we consider the *skewness* of the distribution of pixels along the θ_m direction. The skewness of the probability distribution of a random variable, X, gives a quantitative measure of its asymmetry and is defined as

$$\gamma \triangleq E\left[\frac{\left(X-\mu_X\right)^3}{\sigma_X^3}\right],$$

where μ_X is the mean value of X and σ_X is the variance of X. When a distribution has a positive (negative) skewness, it tails towards positive (negative) values. For example, the distribution histogram shown in Fig. 5.6b has positive skewness because the distribution tail is longer on the right than on the left. The method we describe here for using the skewness of the pixel distribution to estimate orientation is discussed in general by Prokop and Reeves [99]. DeFroment [34] suggested using it for discriminating the head/tail direction of fish. Let $\mathbf{q}_{\theta_m}$ be the unit vector in the direction θ_m, i.e.,

$$\mathbf{q}_{\theta_m} = \begin{pmatrix} \cos\theta_m \\ \sin\theta_m \end{pmatrix}.$$

Now consider a pixel at location (x, y) and project its position relative to the centroid, i.e., $(x - x_{cm}, y - y_{cm})$, into the direction $\mathbf{q}_{\theta_m}$. Let $\rho(x, y)$ be the coordinate of this projection for the pixel at location (x, y):

$$\rho(x,y) \triangleq \begin{pmatrix} x - x_{cm} \\ y - y_{cm} \end{pmatrix}^T \mathbf{q}_{\theta_m} = (x - x_{cm})\cos\theta_m + (y - y_{cm})\sin\theta_m. \tag{5.5}$$

By construction, the mean value of $\rho(x, y)$ over all $(x, y) \in O$ is zero. The skewness of the values of $\rho(x, y)$ over all $(x, y) \in O$ is therefore

$$\gamma = \frac{1}{n} \sum_{(x,y)\in O} \left(\frac{\rho(x,y)}{\sigma(\rho)} \right)^3, \tag{5.6}$$

where $\sigma(\rho)$ is the standard deviation of $\rho(x, y)$ over all $(x, y) \in O$. Because $\sigma(\rho) > 0$, the sign of γ is the same as the sign of

$$\bar{\gamma} = \sum_{(x,y)\in O} (\rho(x,y))^3 .$$

By substituting (5.5) into this expression, we obtain the following expression for $\bar{\gamma}$ in terms of central moments up to the third order.

$$\bar{\gamma} = \mu_{3,0} \cos^3 \theta_m + 3\mu_{2,1} \cos^2 \theta_m \sin \theta_m + 3\mu_{1,2} \cos \theta_m \sin^2 \theta_m \mu_{0,3} \sin^3 \theta_m. \tag{5.7}$$

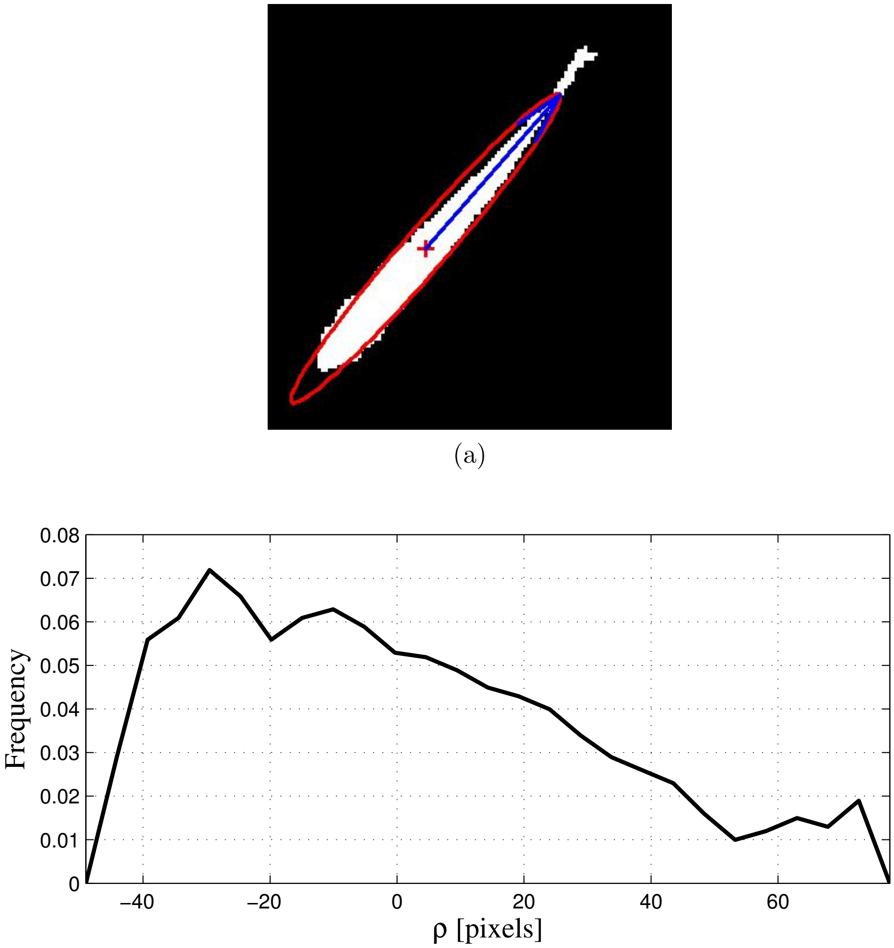
Because the centroid of the fish is closer to the head than it is to the tail, the sign of the skewness — which is equal to the sign of $\bar{\gamma}$ — determines whether θ_m points toward the head or the tail as follows. Negative skewness implies that the majority of the pixels in O have $\rho^3(x, y) < 0$ and therefore θ_m points toward the head. Likewise, positive skewness implies that θ_m points toward the tail. We therefore arrive at the following adjustment rule for determining whether θ_m or $\theta_m + \pi$ is the orientation of the fish.

$$\theta_m \longleftarrow \begin{cases} \theta_m + \pi, & \bar{\gamma} > 0 \\ \theta_m, & \bar{\gamma} \leq 0 \end{cases}, \tag{5.8}$$

where $\bar{\gamma}$ is calculated using (5.7).

Fig. 5.6 shows the moment ellipse for the blob of the fish shown in Fig. 5.5. The blob for this fish is the largest connected component in the threshold image in Fig. 5.5c (there are three small connected components around the tail of the fish). The θ_m direction for this blob points toward the tail of the fish. Fig. 5.6b shows the histogram of the values of $\rho(x, y)$ for the blob; note that the distribution has positive skewness because it tails to the right more than to the left. The value of $\bar{\gamma}$ for this blob is $\bar{\gamma} = 1.5 \times 10^7$, therefore applying the rule (5.8) correctly results in the head direction being adjusted to $\theta_m + \pi$.

We use an expectation-maximization mixture-of-gaussians (EMMG) algorithm to resolve connected components that contain multiple fish. EMMG uses an expectation-maximization technique to fit a sample of points to a mixture-of-gaussians distribution model. The expectation-maximization (EM) technique for estimating the parameters



(a)

(b)

Figure 5.6: (a) The largest connected component extracted from the threshold image in Fig. 5.5c. Its moment ellipse is drawn in red, the centroid is marked with a red "+", and the θ_m direction is indicated by the blue arrow. Note that θ_m points toward the tail in this case. (b) Histogram of the values of ρ for the connected component in (a). The distribution is skewed toward positive values of ρ (i.e., the distribution tails to the right more than to the left), indicating that the tail is in that direction and θ_m should be adjusted as per (5.8).

of statistical distributions was formalized by Baum et al. [7] (see also [11] for a description of EM algorithms and mixture models applied to pattern recognition). A mixture-of-gaussians model is a natural choice to model the distribution of pixels in an image of multiple fish because the level sets of normal distributions are ellipses and, as discussed above, the shape of a fish can be modeled as an elongated ellipse. The EM technique extends naturally to mixture-of-gaussian models. See [10] and [38] for general derivations of the EMMG model and discussions of their implementation. Our implementation follows Example 2 in Section 4 of [38].

The i^{th} distribution in the mixture model has five parameters: its mean, (x'_i, y'_i), and the three independent values of its symmetric and positive definite covariance matrix, Σ_i. The *likelihood* that the pixel at location (x, y) belongs to the i^{th} of n distributions is

$$p_i(x,y) \triangleq \frac{1}{\sum_{j=1}^{n} p(x,y|j)} p(x,y|i) \tag{5.9}$$

where

$$p(x,y|i) \triangleq \frac{1}{2\pi|\Sigma_i|} e^{-\frac{1}{2}\mathbf{d_i}^T \Sigma_i^{-1} \mathbf{d_i}}, \tag{5.10}$$

$$\mathbf{d_i} = \begin{pmatrix} x - x'_i \\ y - y'_i \end{pmatrix},$$

is the probability of observing a pixel at location (x, y) due to the i^{th} distribution. The pixel at location (x, y) has a maximum likelihood of belonging to the i^{th} distribution if $p_i(x, y) > p_j(x, y),\ j \neq i$. As described below, we also use the dimensionless quantity

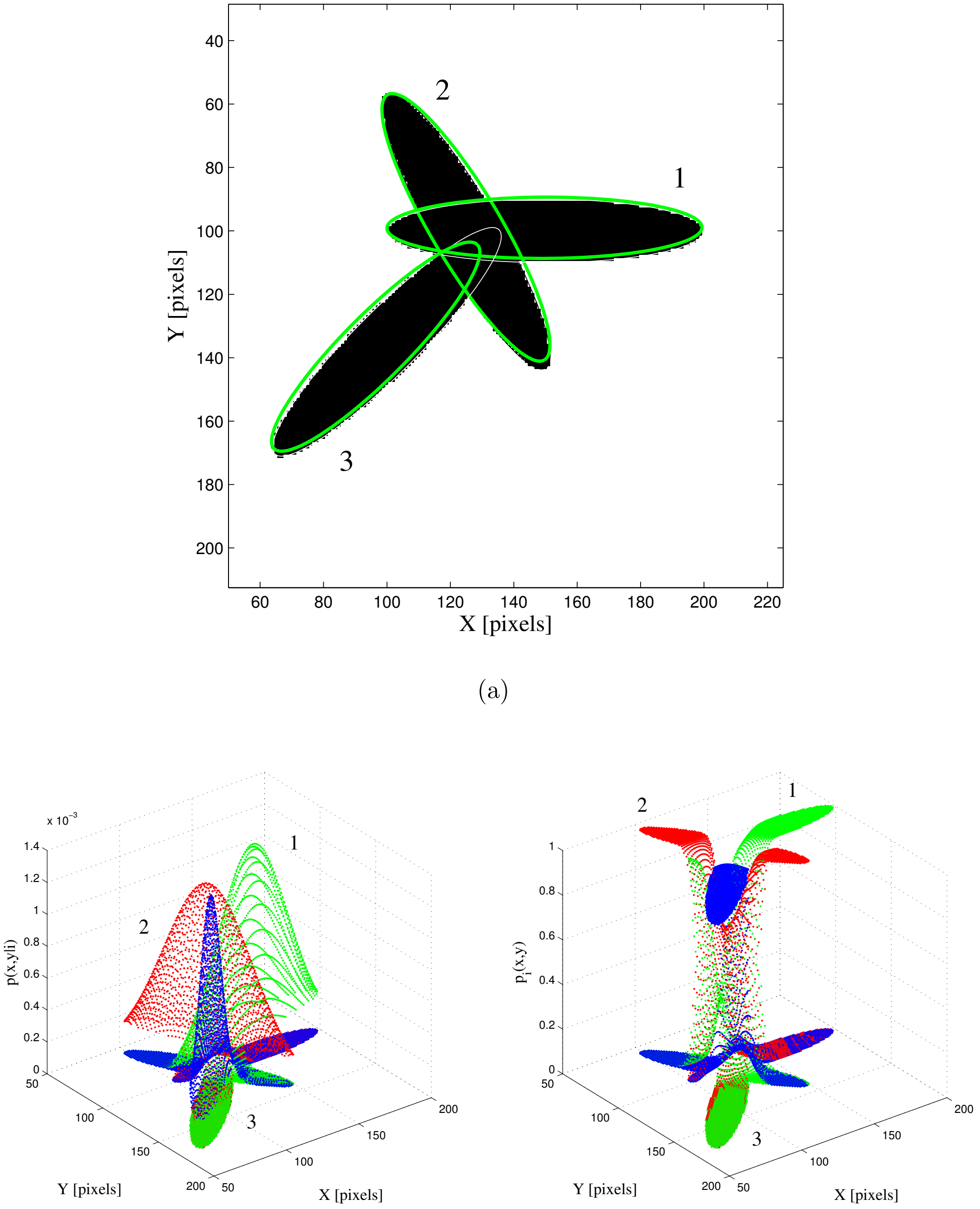
$$s_i(x,y) \triangleq \mathbf{d}_i^T \Sigma_i^{-1} \mathbf{d}_i, \tag{5.11}$$

which describes how close the point (x, y) is to the i^{th} distribution. Note that the level sets of the i^{th} normal distribution are equivalent to the level sets of $s_i(x, y)$.

We implemented an EMMG algorithm in C++ that takes as input an image of a connected component and a count, n, of how many fishes' images are represented in that connected component, and produces as output a label for each pixel that describes to which of the n normal distributions that pixel belongs. The algorithm is initialized by uniformly arranging normal distributions to completely cover the area of the connected component. We iterate the EMMG algorithm until either numerical convergence of the parameters is obtained or a preset maximum number of iterations is carried out. Typically less than 10 iterations are required. Then, we label each pixel according to the following rules. The pixel is labeled as belonging to the distribution for which it has the maximum likelihood. The pixel is also labeled as belonging to any distribution for which $s_i(x, y) \leq 4$. Note that a pixel can be labeled as belonging to more than one distribution. That is, a pixel may belong to the image of more than one fish.

We construct n fish blobs from the n distributions estimated by the EMMG algorithm by assigning each pixel to the i^{th} blob if it is labeled as belonging to the i^{th} distribution. The centroid and orientation of each fish is then calculated from the blobs using the moment-based methods described above.

Fig. 5.7 shows an example of the EMMG algorithm applied to an image of a connected component that was generated by superimposing three overlapping ellipses (outlined in white). The ellipses generated from the EMMG algorithm are outlined in green, and closely match the input ellipses. The largest error is in the bottom left ellipse and this error is most apparent in the region where all three ellipses overlap. The values of $p_i(x, y)$ (as in (5.9)) and $p(x, y|i)$ (as in (5.10)) are shown in Fig. 5.7b and Fig. 5.7c, respectively. Note that the values of $p_i(x, y)$ are high where the ellipses do not overlap and lower where the ellipses do overlap.



(a)

(b)

(c)

Figure 5.7: Illustration of the EMMG algorithm. (a) Binary image input into the algorithm and the ellipses generated by the algorithm. The image is shown in inverse (i.e., black pixels are in the connected component). The input image was generated by superimposing three ellipses. The input ellipses are outlined in white. The output ellipses are outlined in green. (b) The values of $p(x, y|i)$ (as in (5.10)) for each distribution. (c) The values of $p_i(x, y)$ (as in (5.9)) for each distribution. In (b) and (c) one color is used per distribution. In (a), (b), and (c), the distribution numbers are labeled.

5.2.2 State Estimation

We describe here how we use the results of the segmentation process above — a list of fish centroids and orientations — to estimate the dynamic states of the fish. The state of the k^{th} tracked fish at time instant t is

$$\hat{\mathbf{x}}_k(t) = \begin{pmatrix} \hat{x}_k(t) \\ \hat{y}_k(t) \\ \hat{\theta}_k(t) \\ \hat{s}_k(t) \end{pmatrix},$$

where $(\hat{x}_k(t), \hat{y}_k(t))$ is its estimated position, $\hat{\theta}_k$ is its estimated orientation, and $\hat{s}_k(t)$ is its estimated speed. We use an unscented Kalman filter (UKF) to estimate these quantities using the blob measurements and a dynamical model of the motion of the fish. We assume that the dynamics of each fish are independent and therefore we use a separate UKF for each fish. In reality, the dynamics of each fish are affected by all of the other fish and the robotic fish; the assumption of independence serves to both simplify the implementation and to avoid biasing the state estimates by including additional assumptions about the interactions between fish.

We begin by establishing notation for the measurement variables and the inputs to each UKF. Using this notation, we then discuss the problem of *data association*; that is, determining which measurement should be assigned to which tracked fish. Then, we describe the operation of the UKF and the dynamical model that underlies its operation. Finally, we describe some additional processing steps that must be undertaken to ensure that the heading angle measurements are correct and in the right format for the UKF.

The input to the state estimation step is the collection of centroids and orientations that are calculated during the blobbing step, as described in Sec. 5.2.1. The j^{th} fish's blob measurements at time step t are labeled as

$$\tilde{\mathbf{z}}_j(t) = \begin{pmatrix} \tilde{x}_j(t) \\ \tilde{y}_j(t) \\ \tilde{\theta}_j(t) \end{pmatrix},$$

where $(\tilde{x}_j(t), \tilde{y}_j(t))$ is the blob centroid as calculated in (5.1) and $\tilde{\theta}_j(t)$ is the blob orientation as calculated by (5.4) and corrected according to (5.8). The input to the k^{th} UKF at time step t is the measurement vector,

$$\bar{\mathbf{z}}_k(t) = \begin{pmatrix} \bar{x}_k(t) \\ \bar{y}_k(t) \\ \bar{\theta}_k(t) \end{pmatrix},$$

and $\bar{\mathbf{z}}_k(t) = \tilde{\mathbf{z}}_j(t)$ for some j. The problem of data association is to determine the mapping between the blob measurement j and the tracked fish k. The ordering of the blob measurements has no guaranteed ordering and therefore the data association mapping must be re-evaluated at every time step.

To solve the data association problem, we use the efficient Hungarian assignment algorithm developed by Kuhn [64] and expanded by Munkres [81]. We leverage an implementation of this algorithm written in C by Gerkey [45]. For each pair (j, k), we calculate the squared distance,

$$d_{j,k}^2(t) = \left(\hat{x}_k(t-1) - \tilde{x}_j(t)\right)^2 + \left(\hat{y}_k(t-1) - \tilde{y}_j(t)\right)^2,$$

between the previously estimated position of the k^{th} fish[†] and centroid of the j^{th} blob. The Hungarian algorithm finds an assignment $j'(k)$ for each k such that $\sum_k d^2_{j'(k),k}(t)$ is minimized. The UKF measurement for the k^{th} fish is then $\bar{\mathbf{z}}_k(t) = \tilde{\mathbf{z}}_{j'(k)}(t)$.

The UKF iteration at time instant t has two steps: prediction and update. The predicted state, $\mathbf{x}_k(t)$, is based on the previously-estimated state and uses the dynamical model

$$\mathbf{x}_k(t) = \hat{\mathbf{x}}_k(t-1) + \Delta t \begin{pmatrix} \hat{s}_k(t-1)\cos\hat{\theta}_k(t-1) \\ \hat{s}_k(t-1)\sin\hat{\theta}_k(t-1) \\ 0 \\ 0 \end{pmatrix} + \boldsymbol{\zeta}_k(t), \tag{5.12}$$

where $\Delta t = 0.1$ s is the time-step length and $\boldsymbol{\zeta}_k(t)$ is a disturbance vector. The disturbance vector, $\boldsymbol{\zeta}_k(t)$, is assumed to be drawn from a normal distribution with zero mean and covariance matrix

$$Q = \begin{pmatrix} \sigma_d^2 & 0 & 0 & 0 \\ 0 & \sigma_d^2 & 0 & 0 \\ 0 & 0 & \sigma_\theta^2 & 0 \\ 0 & 0 & 0 & \sigma_s^2 \end{pmatrix},$$

where σ_d, σ_θ, and σ_s are disturbance variances for position, heading, and speed, respectively. The disturbance vectors are assumed to be independent at each time step and for each fish. The values of σ_d, σ_θ, and σ_s can be modified on-line to tune tracking performance as needed. In the UKF update step, the state estimate, $\hat{\mathbf{x}}_k(t)$,

---

[†] We have also tried using the UKF's predicted position of the k^{th} fish, taken from $\mathbf{x}_k(t)$ as calculated in (5.12), to calculate the squared distance. The previous position, $(\hat{x}_k(t-1), \hat{y}_k(t-1))$ gives better results in practice.

is updated by comparing the measurement $\bar{\mathbf{z}}_k(t)$ with the predicted measurement,

$$\mathbf{z}_k(t) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \hat{\mathbf{x}}_k(t).$$

The calculations that produce the updated state estimate are described in [54] and [125]. These calculations are based on the state propagation model (5.12) and a noisy measurement model, $\tilde{\mathbf{z}}_k(t) = \mathbf{z}_k(t) + \boldsymbol{\eta}_k(t)$, where $\boldsymbol{\eta}_k(t)$ is the measurement noise. We assume that the measurement noise is drawn from a normal distribution with zero mean and covariance matrix

$$R = \begin{pmatrix} \sigma_{pm}^2 & 0 & 0 \\ 0 & \sigma_{pm}^2 & 0 \\ 0 & 0 & \sigma_{\theta m}^2 \end{pmatrix},$$

where σ_{pm} and $\sigma_{\theta m}$ are the variance of the position and heading measurement errors, respectively. The values of σ_{pm} and $\sigma_{\theta m}$ can be modified on-line to tune tracking performance as needed. The measurement errors are assumed to be independent at each time step and for each fish.

The UKF works well with the nonlinear dynamics (5.12) because it uses several copies of the previous state estimate to calculate the new state estimate using the full nonlinear dynamics and allows the application of nonlinear constraints (for example, the speed may not be negative). The state copies are produced by offsetting the previous state estimate in symmetric pairs that are distributed according to a dynamic estimate of the covariance of the state estimation error. Our implementation uses the original state estimate, two state copies for each element of the state, two copies for each diagonal element of Q, and two copies for each diagonal element of R, for a total of 23 copies of the state. The source code of our implementation is available online at `http://github.com/leonard-lab/MADTraC/blob/master/MT/MT_Core/support/UKF.cpp`.

We perform two additional processing steps on each measurement before it is used in a UKF. Despite the correction rule (5.8), there are occasional errors in the blob orientation measurements with respect to the head/tail direction (i.e., the measured orientation could be off by π rad). The first processing step addresses these possible errors. The second processing step addresses the fact that, for continuity, we require the orientation measurement to evolve over $\mathbb{R}$, whereas the blob orientation measurement is always in the range $[-\pi, \pi]$.

For the first processing step, we keep a history of the estimated position of each fish for five time steps. If the fish's displacement over those time steps is at least one half of a body length, then we use the angle of the displacement, $\alpha \triangleq \tan^{-1} \frac{\hat{y}_k(t)-\hat{y}_k(t-5)}{\hat{x}_k(t)-\hat{x}_k(t)}$, to align the orientation measurement $\bar{\theta}_k(t)$. The alignment rule is

$$\bar{\theta}_k(t) \longleftarrow \begin{cases} \bar{\theta}_k(t) + \pi, & \left|\cos \frac{\bar{\theta}_k(t)-\alpha}{2}\right| \leq \rho \\ \bar{\theta}_k(t), & \left|\cos \frac{\bar{\theta}_k(t)-\alpha}{2}\right| > \rho \end{cases},$$

where ρ parameterizes the degree of alignment required. We use $\rho = \frac{\sqrt{2}}{2}$ because it corresponds to flipping $\bar{\theta}_k(t)$ if it disagrees with the displacement direction by more than $\frac{\pi}{2}$ radians.

For the second processing step, we modify the value of the orientation measurement, $\bar{\theta}_k(t)$, so that its value is within π radians of the previous orientation estimate, $\hat{\theta}_k(t-1)$, without changing its phasor direction. This is accomplished using the update rule

$$\bar{\theta}_k(t) \longleftarrow \hat{\theta}_k(t-1) + \tan^{-1} \frac{\sin\left(\bar{\theta}_k(t) - \hat{\theta}_k(t-1)\right)}{\cos\left(\bar{\theta}_k(t) - \hat{\theta}_k(t-1)\right)}.$$

As an example, suppose that the previous estimate is $\hat{\theta}_k(t-1) = \pi$ rad and the measurement is $\bar{\theta}_k(t) = -3.13$ radians. In this example, the heading and measurement correspond to roughly the same angle but with very different numerical values, which would cause the UKF to incorrectly estimate the measurement error and produce a bad update. Following the above update rule, the measurement is correctly updated to $\bar{\theta}_k(t) = -3.13 + 2\pi \approx 3.15$ radians. For both processing steps, it is important to calculate the arctangent using a four-quadrant method. We use the C function `atan2` in our implementation.

5.2.3 Estimating the Properties of Fish Schools

We use the estimated states of individual live fish to estimate properties of the fish school in real-time. We can then use these estimates to calculate commands for the robotic fish in an experiment. For example, we can estimate the location of the centroid of the fish school by averaging the positions of the live fish. Similarly, we can estimate the linear momentum of the school by summing their estimated velocities (assuming the fish have equal masses).

We have been able to track tens of fish with an overall bandwidth of about 10 Hz on commodity hardware. The majority of the processing occurs during the segmentation step and the data association portion of the state estimation step. The UKF calculations consume a relatively small portion of each time step. To track larger numbers of fish, the overall bandwidth of the system would need to be reduced because the segmentation would be more difficult and therefore more computationally expensive, and the number of unscented Kalman filters would need to be increased. Reducing bandwidth is unacceptable because our ability to stably control the robotic fish is reduced and because we lose the ability to react to the quickly changing behavior of the live fish. We overcome this limitation by estimating the properties of the fish school from fish that we segment from the image at each time step. That is, we extract blob estimates of a large portion of the fish in the school using the methods described in Sec. 5.2.1, but do not necessarily track all of the fish from time step to time step. This reduces the overall computational load because we do not need to solve the association problem for those fish, nor do we estimate their states dynamically using one UKF for each fish. We further reduce the computational load by rejecting multi-fish connected components from this process, thus eliminating the need to perform the computationally expensive EMMG algorithm for the fish that we are not tracking. The remaining segmentation steps — background subtraction and connected component labeling — are relatively computationally inexpensive.

We carry out the segmentation step twice for each video frame; once to measure the positions and orientations of the fish that we are fully tracking (e.g., the robotic fish) and once to measure the positions and orientations of a subset of fish that we will use to estimate properties of the school. Performance of the two segmentations can be tuned by changing the thresholds on background subtraction (T_B above) and on blob size independently for both segmentations. The size of the subset of fish that is used to estimate properties of the school is dependent on the values of the thresholds used in the segmentation algorithm. Fish that are omitted from this subset are typically either too small or are clustered with other fish. Small fish are difficult to distinguish from image noise and produce errors in orientation estimates with a higher frequency, and clustered fish produce large multi-fish connected components that are omitted. Therefore, the error properties of estimating the school's centroid location, linear momentum, or any of the quantities discussed below are dependent on the distribution properties of small fish and clusters within the school. These properties are difficult to predict and may vary over the course of an experiment, but we may optimistically assume that the errors are uniformly distributed across the school and that therefore the school property estimation errors are small.

To estimate the polarity of the school, we compute the magnitude of the complex order parameter, $|p_{\boldsymbol{\theta}}|$, as defined in (4.5). In this case, the phase set $\boldsymbol{\theta}$ consists of the measured orientations of all available fish. Recall that the value of $|p_{\boldsymbol{\theta}}|$ gives a measure of synchrony, and that $|p_{\boldsymbol{\theta}}| \in [0, 1]$. When $|p_{\boldsymbol{\theta}}| = 1$, the school is maximally polarized and all of the fish are facing the same direction. When $|p_{\boldsymbol{\theta}}| = 0$, the school is minimally polarized. This can occur, for example, if the fish are uniformly distributed while moving around a circle, or if the fish are milling around and moving in random directions that are uniformly distributed. See [91] for further examples of using the complex order parameter to classify the motion patterns of schooling fish.

We can estimate the boundary of the fish school by computing a bounding ellipse based on spatial moments. These calculations are identical to the ones described in Sec. 5.2.1 for estimating the orientation of a fish from its blob, except that here we use the measured positions of all of the fish to compute spatial moments rather than using pixel locations to compute image moments. That is, $(j, l)^{th}$ spatial central moment, $\mu_{j,l}$ is

$$\mu_{j,l} = \sum_{k=1}^{N_m} (x_k - x_c)^j \, (y_k - y_c)^l$$

where (x_k, y_k) is the measured position of the k^{th} fish,

$$(x_c, y_c) = \left(\frac{1}{N_m} \sum_{k=1}^{N_m} x_k, \frac{1}{N_m} \sum_{k=1}^{N_m} y_k \right) \tag{5.13}$$

is the location of the centroid of the school, and N_m is the number of fish whose position and orientation measurements are available. Note that we use the same $\mu_{j,l}$ notation here for spatial central moments of the school that we have used above for image central moments in Sec. 5.2.1; the two concepts are analogous except that here we consider the positions of fish rather than the positions of pixels. Note also that $\mu_{0,0} = N_m$. Using the spatial central moments, the semi-major and semi-minor axes of the bounding ellipse can then be calculated using (5.2) and (5.3), respectively, and its orientation can be calculated using (5.4). Fig. 5.8 shows an example of estimating the bounding ellipse of a school of fish. In this case, the bounding ellipse poorly represents the shape of the school because the fish are spread far apart and a small

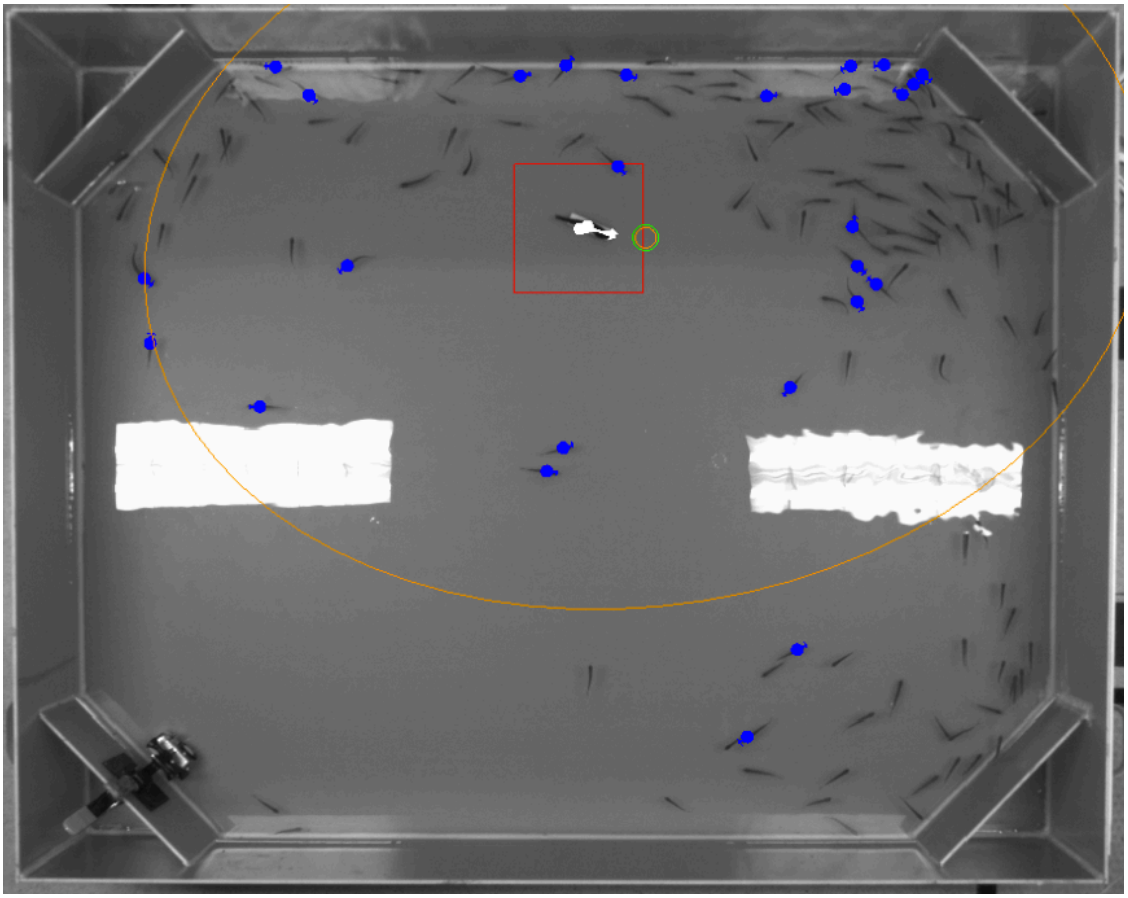


Figure 5.8: A video frame captured from the robotic fish testbed's tracking and control software, showing a robotic predator fish chasing the centroid of a live school of golden shiners. The robotic predator fish's position and orientation are represented by a white dot and arrow, respectively. The red square is the tracking search window for the robotic predator fish. The blue dots and small blue arrows indicate the positions and directions of segmented fish within the school. The background subtraction threshold, τ, was set relatively high so that a relatively small number of the fish are segmented, thus increasing the closed-loop speed of the system. The green circle indicates the estimated centroid location of the school, and the orange ellipse indicates the moment-based estimation of the bounding ellipse. The white regions in the image are caused by reflections of overhead lights in the laboratory. An underwater camera is mounted in the lower-left diagonal of the tank. This figure first appeared in [115].

subset (the ones marked with blue dots) of the fish are used to estimate it. Much more accurate bounding ellipse calculations are shown in Fig. 5.9.

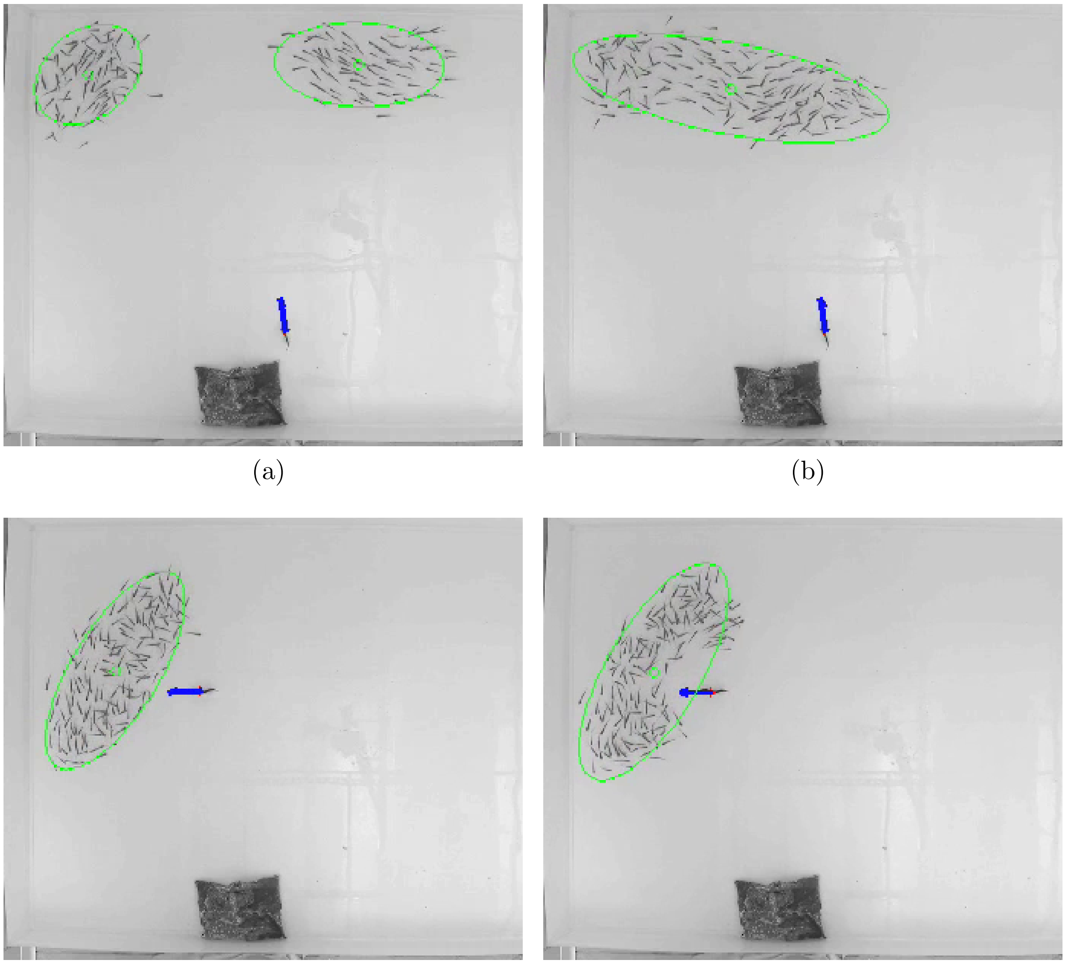
Clustering techniques can be used to detect splits in the group. One computationally inexpensive method to find clusters is to subsample the image and then perform connected-component labeling on the subsampled image. Fig. 5.9 shows four video snapshots showing the application of this technique used to identify two separate groups of fish (Fig. 5.9a) as they merge into a single, larger, group (Figs. 5.9b-5.9d). The bounding ellipses shown in Figs. 5.9a-5.9c provide good approximations of the school shapes because the fish are relatively densely spaced and the boundary of the school is roughly ellipsoidal. Note, however, that in Fig. 5.9d, the robotic fish (labeled with a thick blue line in the figure) has impacted the shape of the group and the bounding ellipse no longer accurately represents the boundary of the school, particularly near the robotic fish. To handle schools with more complex boundaries, as in Fig. 5.8 and Fig. 5.9d, we could employ more sophisticated computational geometry methods that are available for estimating the boundaries of shapes based on a sampling of points. Two viable alternatives include convex hull computation (for which there are many algorithms available) and alpha shape algorithms (see, for example, Edelsbrunner [39]).

5.3 Experimental Demonstrations

In this section, we describe two experimental demonstrations that use our testbed to enable a robotic fish to interact with a live fish school. Both of these experiments use the robotic fish to mimic a predator fish using the koi model shown in 5.3b. The same group of about 100 golden shiners was used for both experiments. These are proof-of-concept demonstrations of what is possible with this testbed.

5.3.1 School Centroid Chasing

In this demonstration, one robotic predator fish uses real-time feedback to continually follow the centroid of a live fish school. The school centroid is estimated as in (5.13), using the subset of the school that is segmented at each time step. We use feedback



(a) (b)

(c) (d)

Figure 5.9: A sequence of video frames of a robotic predator fish approaching a school of live golden shiners, with overlayed visualization. The frames are ordered in time with frame (a) being the earliest. In all four frames, the position and orientation of the robotic predator fish is indicated by the thick blue line. A subsampling and clustering algorithm (see text) is being used to detect splits in the group, as shown in frame (a). The subgroups merge in frame (b) and the robotic predator fish approaches the group in frame (c). In frame (d), the robotic predator fish has moved into the school and caused the shape of the school to deform as nearby fish swim away from it. This figure first appeared in [115].

control law described in Sec. 5.1.2 with the target location, $(x'(t), y'(t))$, set equal to the school centroid estimate at each time step. That is, the robotic fish is continually steering to head toward the centroid of the school.

Real-time feedback control of the robotic fish is required for this type of experiment because the location of the centroid of the school cannot be known ahead of time. As the robot moves toward the school, the fish react to the robot and the location of the centroid moves. Using real-time feedback allows the robot to respond quickly to the fish school during this interaction. A human operator (for example, with a remote control) could perform this type of experiment, but automation introduces superior repeatability and responsiveness. Furthermore, the programmed interaction rules can be accounted for in off-line analysis of the fish school.

Fig. 5.8 shows one video frame that was captured from the tracking and control software while it was running a demonstration of school centroid chasing. The location of the robotic fish is marked with a white dot, and its orientation is indicated with a white arrow. Segmentation of the robotic fish was limited at each time step to a small window around its last known position; this window is indicated in Fig. 5.8 with a red rectangle. If the robotic fish was not found in this window, then the segmentation algorithm was repeated for the entire image. Individual fish were segmented from the image but their velocities were not estimated (i.e., they were not tracked from frame to frame); the blue dots and arrows in Fig. 5.8 indicate the positions and orientations of all of the fish that were found in that image. The background subtraction threshold, τ, was set relatively high for performance reasons. Consequently, a small portion of the fish are segmented from the image. The green circle in Fig. 5.8 indicates the estimated location of the fish school's centroid, and the orange ellipse is the bounding ellipse that was estimated using the methods described above. Note that the robotic fish's orientation is pointed directly toward the estimated centroid location.

5.3.2 Triggered Dart Toward School

In this demonstration, the robotic predator fish begins in a random position within the tank and waits for a triggering event. When the triggering event occurs, the robotic fish accelerates quickly toward the estimated location of the school centroid. For this demonstration, the triggering event was the school's polarity dropping below a threshold value. We estimate the polarity by calculating the complex order parameter, $|p_{\boldsymbol{\theta}}|(t)$, as defined in (4.5) and discussed in Sec. 5.2.3 above. The value of $|p_{\boldsymbol{\theta}}|$ is calculated from the orientations of all fish segmented from the current overhead camera image. The value of the polarity threshold was 0.5. As discussed above, low values of polarity correspond to milling behavior of the school. Similar to the school centroid chasing demonstration, we use a relatively low value of the background subtraction threshold, τ.

Real-time feedback is crucial for repeatability and responsiveness in this type of experiment. The closed-loop system can wait an indefinite amount of time for the fish school to be in the right state before triggering an attack, and an attack occurs immediately after the trigger condition occurs. The triggering event can correspond to any number of conditions that we can calculate based on the state of the fish school. For example, an attack could be triggered when the school is in a particular position and/or when the school has a particular shape. The triggering event can also depend on dynamic or time-integrated quantities. For example, we could trigger an attack when the school is moving in a particular direction or when the school has stayed in a particular location for more than a few seconds. The attack need not be directed at the centroid of the school; the robotic fish could dart toward the nearest fish or toward a fish that is separated from the group. Facilitating these various conditions and behaviors is achieved mainly by changing the control software.

5.4 Extensions

In this section, we describe some extensions of the methods described above in this chapter. Many of these have emerged as natural extensions because they address a common need to track multiple objects from overhead video. Furthermore, in most of these extensions the objects to be tracked share a very similar appearance to one another and can be potentially very close together, leading to many of the same challenges that were addressed when designing the real-time fish tracking algorithms for the robotic fish testbed.

We begin in Sec. 5.4.1 by giving a brief description of some of the other applications using our tracking techniques and a synopsis of the unique challenges associated with each one. In Sec. 5.4.2, we describe the MADTraC C++ library that we have created in an effort to facilitate rapid development of these applications by using shared code.

The projects that we describe here have highly symbiotic relationships with one another. While the robotic fish testbed was the initial inspiration for this work and the focus of the earliest development efforts, all of these applications have simultaneously been in active development (to some extent) for the past few years. As a result, improvements made or bugs discovered while working on one application often lead to improvements in the other applications — either directly or via improvements to MADTraC. In Sec. 5.4.3, we describe a system that was designed to enable human-in-the-loop experiments on the Beluga testbed (described below); in the future, this system can be directly applied to the robotic fish testbed to enable human-in-the-loop experiments with fish schools as well as high-level control of the system through Matlab scripting.

5.4.1 Other Testbeds and Tracking Applications

In this section, we briefly describe four other applications of the tracking techniques described above in this chapter. The first two applications are off-line (i.e., not real-time) uses of the tracking software to reconstruct trajectory data from overhead video. The second two applications are real-time robotics testbeds that use overhead cameras for localization of the robots in a manner that is very similar to the robotic fish testbed. For each application, we give a brief overview and a synopsis of the design challenges unique to that application.

off-line Fish Tracking

It is straightforward to disable the robot control and communication modules in the software that was developed for the robotic fish testbed and use it to reconstruct the trajectories of fish by tracking them in pre-recorded video. This is the method that was used to obtain the killifish trajectory data used in Chap. 2. Figs. 2.3-2.5 show example trajectories for schools of two and three killifish. Fig. 5.10 shows an example of tracking 10 golden shiners. Note that the two left-most fish are overlapping and correctly segmented.

Because the tracking is done off-line, the bandwidth of the system (in this case, the tracking frame rate) is less crucial and therefore we can adjust the tracking parameters to favor tracking performance at the cost of throughput speed. For example, the background subtraction threshold, τ, and the minimum connected component size can both be reduced, which leads to better segmentation but also admits more noise and therefore requires more computational effort to filter out the noise. Furthermore, high resolution and high frame-rate video can be used because the images do not need to be manipulated in real-time. The example tracking output in Fig. 5.10 was created from high definition video; the image is zoomed in to the immediate area around the fish.

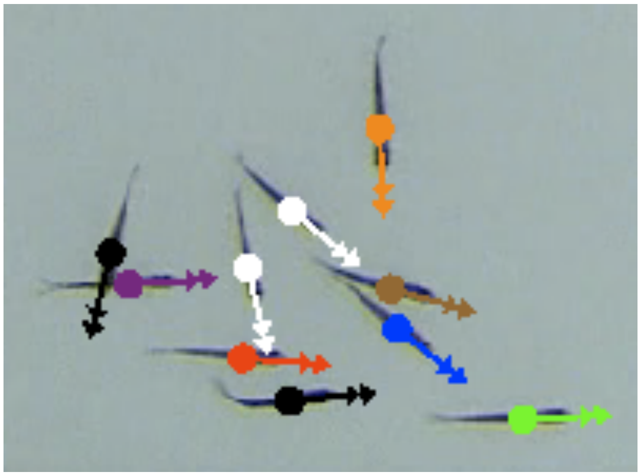


Figure 5.10: Snapshot of ten golden shiners that were tracked off-line from high-definition overhead video. This is a zoomed-in view of just the portion of the arena around the fish. Each colored dot represents the estimated position of a fish. Each arrow represents the heading angle of that fish: the shorter arrow for the measured orientation and the longer arrow for the heading estimated by the UKF. This figure first appeared in [115].

Dancer Tracking

A modified version of the fish tracking software was used to track the locations of dancers from an overhead video. The dancers were participating in a series of "human flocking" experiments, in which each dancer was instructed to follow a set of motion rules that are inspired by common models of biological collective motion. The reconstructed trajectories were used to analyze the communication topology of the dancers with respect to group leadership. See [72] for further description of the human flocking experiments and their analysis.

Fig. 5.11 shows a snapshot from the tracking of 13 dancers performing a human flocking experiment. Each dancer wears a yellow hat in order to improve contrast from the background. Instead of background subtraction, a color-based segmentation algorithm was used. A different dynamical model than the one described in Sec. 5.2.2 for fish tracking was used for each dancer's UKF; in particular, there is no orientation measurement. The orientation of each dancer was therefore estimated from their

Figure 5.11: Snapshot of output from the tracking of dancers performing human flocking experiments. The estimated position of each dancer is indicated with an colored dot. The estimated heading of each dancer is indicated with an arrow. The dancers are deliberately blurred to protect their privacy. This figure first appeared in [72].

velocity. Thus, the orientations must be taken as approximations because each dancer is free to turn their head while they move about. More sophisticated techniques would be required to estimate the true orientations of the dancers.

MiaBot Pro Testbed

The robotic fish testbed can be used as a cooperative mobile robotics testbed with a small amount of additional effort. By removing the tank (see Fig. 5.1) and modifying the tracking software to track the MiaBot Pro robots directly, we can perform a variety of coordinated control experiments.

Fig. 5.12 shows a snapshot of a coordinated control experiment using the MiaBot Pro robots. This experiment was performed using coordinated control laws that were inspired by tensegrity structures (see, for example, [68]). The blue lines in Fig. 5.12

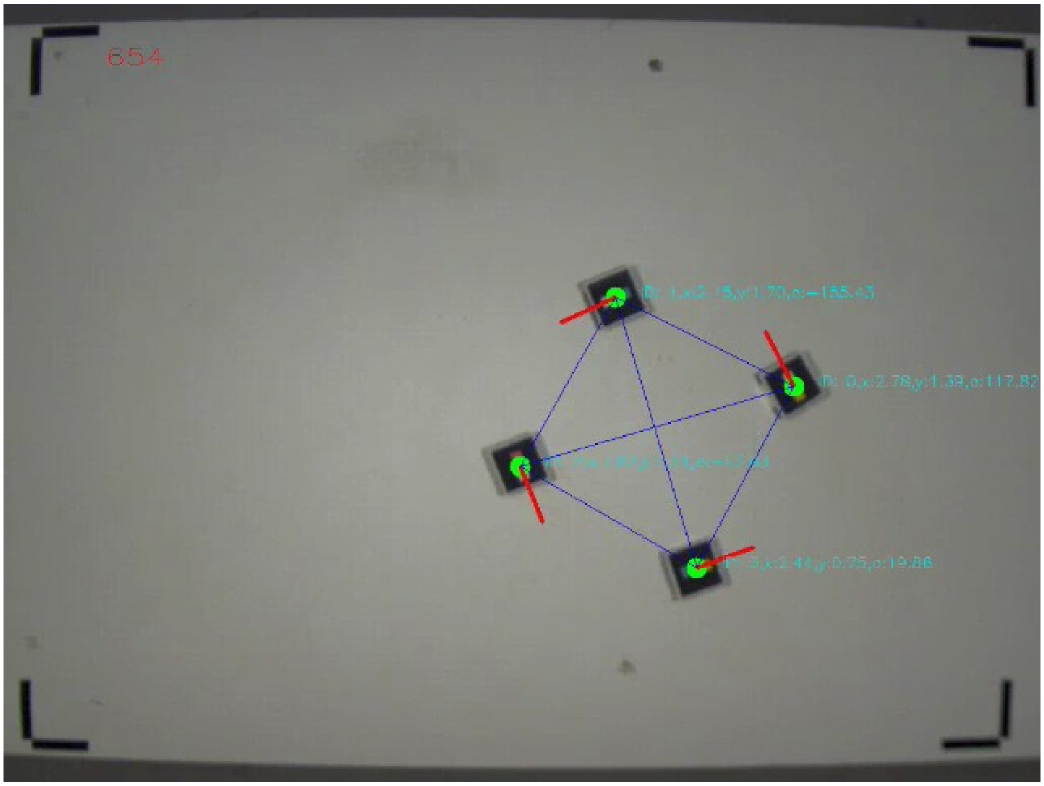


Figure 5.12: Snapshot of a coordinated control experiment using the MiaBot Pro robots. The position of each robot is marked by a green circle. The orientation of each robot is indicated by a red line.

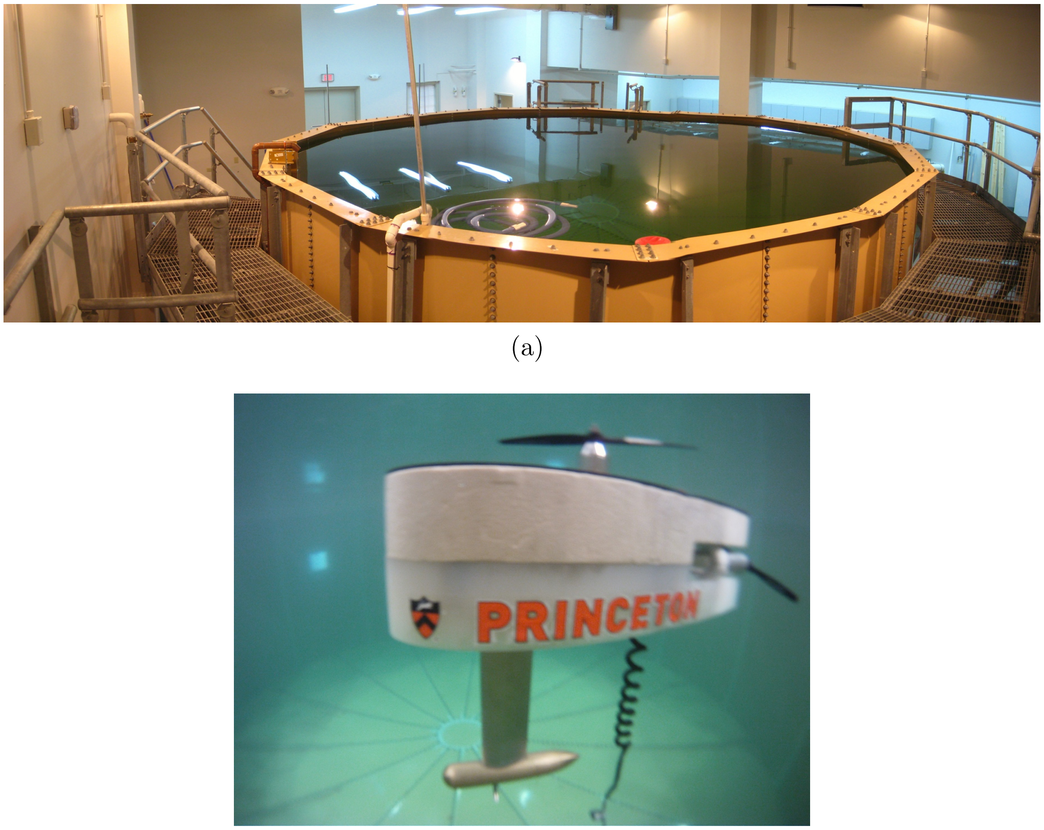
represent the components of the virtual tensegrity structure underlying this control algorithm. The light blue text near each robot is visual feedback of the positions and orientations of each robot (the text has become illegible through video conversion processes). The red number in the upper left is the number of frames that have elapsed since the beginning of the experiment.

The MiaBot Pro robots have a square profile when viewed from above. Therefore, their orientations cannot be estimated directly. To overcome this, we place markers on the tops of the robots. In Fig. 5.12, colored markers are used. Some of the markers can be seen beneath the green dots that mark the robots' locations. The practice of applying a marker to the top of a robot for identification and tracking purposes is widely used. For example, this practice is used extensively in the RoboCup Small Size League [42] (this is the robotic soccer league for which the MiaBot Pro robots were originally designed). In the future, we may use black arrow-shaped markers printed on white paper and glued to the robots. Arrow-shaped markers will work well because the same moment-based algorithms can be used to estimate orientation that we used to accurately estimate the orientation of fish. Note that our tracking algorithms can be used to track individual robots without colored markers or other identifying marks. Once the identities of the robots are established initially (via initial positions or by selectively moving each robot in turn), our tracking algorithms are able to accurately maintain the identities of the robots over time.

Beluga Underwater Vehicle Testbed

The Beluga underwater vehicle testbed was developed at the Dynamical Control Systems Laboratory (DCSL) at Princeton University to provide a facility for exploring three-dimensional cooperative control laws in a variety of contexts, including human-in-the-loop control and decision making. The following is an abridged description of the system, focusing on how we track and control the Beluga robots using software that was adapted from what was originally developed for the robotic fish testbed. For a thorough description of the Beluga testbed, including design and model documentation for the underwater vehicles, see the Ph.D. thesis of Andrew Stewart [108].

The Beluga testbed is built around a water tank that is 20 feet in diameter, 8 feet deep, and holds approximately 20,000 gallons of water. A tank-side computer sends low-level motor controls to up to four underwater vehicles via cable tethers. The vehicle platform is named "Beluga" and was designed to have four degrees of freedom controlled by three actuators: a forward/reverse propeller, a vectoring servo for the forward/reverse propeller, and a vertical propeller. Each vehicle's pitch and yaw are passively stabilized by a keel-hung ballast. Fig. 5.13a shows a view of the water tank, and Fig. 5.13b shows one of the Beluga vehicles.



(a)

(b)

Figure 5.13: (a) A view of the Beluga testbed's 20,000 gallon water tank. (b) A view (taken with an underwater camera) of one of the Beluga vehicles. Note the fixed vertical thruster on the top of the vehicle, the vectored thruster at the rear of the vehicle, the ballast on the bottom of the vehicle, and the tether cable coming out of the bottom of the vehicle. Photo credit for (a) and (b): Andrew Stewart.

The tank-side computer that controls the Beluga vehicles computes closed-loop control commands using estimated positions and orientations of the vehicles. The positions and orientations are estimated using four overhead cameras and a depth sensor on each vehicle. Four cameras are required because the wide diameter of the tank (20 feet) and height of the ceiling (about 9 feet above the water surface) prohibit a single camera from having a sufficiently wide field of view. Fig. 5.14 shows a a diagram of the system.

We modified the robotic fish testbed's software to track the Beluga vehicles across all four cameras and to compute and send low-level control signals to each vehicle.

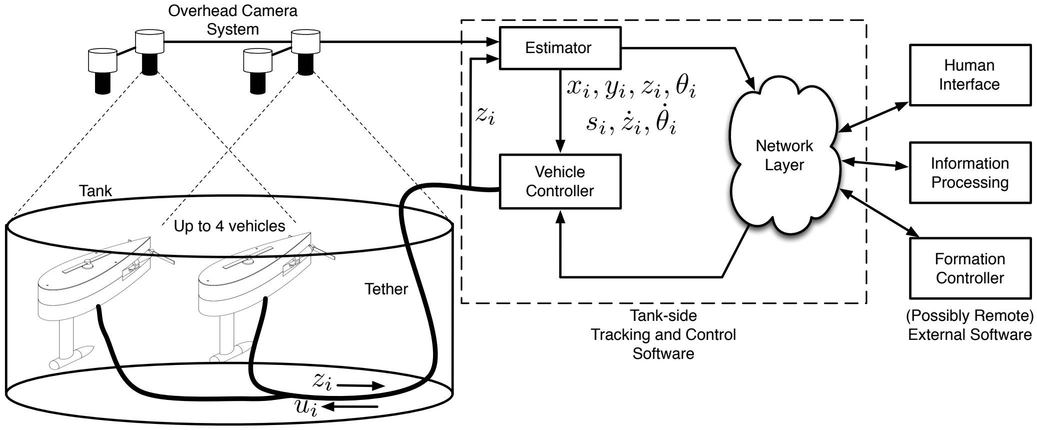


Figure 5.14: Diagram of the Beluga testbed system. (x_i, y_i, z_i) is the position of the i^{th} vehicle, θ_i is its orientation in the $x - y$ plane, s_i is its speed in the $x - y$ plane, and u_i is its control input. The network layer and interface to external software is described in further detail in Sec. 5.4.3. Note: Vehicles and tank are not to scale. This diagram was modified from one created by Andrew Stewart and Paul Reverdy.

Using camera calibration data and the depth measurements from the vehicles, we are able to measure the positions and orientations of the vehicles in real-world coordinates. Camera calibration is difficult for this system due to refraction at the water's surface. We developed an iterative procedure that uses the known locations of several tank features and the known water depth to calculate calibration parameters for each camera. Minor modifications to the communications module were required to be able to send commands to the Beluga vehicles and receive back depth measurements.

The real-time tracking algorithm described above in Sec. 5.2 was modified to create the real-time tracking algorithm used for the Beluga testbed. Fig. 5.15 gives an overview of the algorithm. The basic structure of the algorithm is similar to the one used for the robotic fish testbed (see Fig. 5.4). We begin by isolating pixels in each of the four camera frames that are likely to belong to the image of a robot using thresholding in the hue, saturation, and value (HSV) color space. We then use a connected component labeling algorithm (similar to the one described above in Sec. 5.2.1) to group pixels together. The positions in world coordinates are converted into image coordinates using the camera calibration data. From this, we can determine which robots should be visible in which camera frames and use that information for EMMG resolution (by setting the initial conditions of the algorithm using the previous positions and orientations), for data association, and to determine the measurement size for the UKF (see below). The orientation of each robot is estimated using the same moment-based method describe above for estimating the orientation of fish; the head/tail discrimination algorithm can be applied directly because the overhead view of the Beluga is very similar to that of a fish (it is elongated and wider toward the front of the vehicle).

The state of the UKF used for the i^{th} vehicle is

$$\mathbf{x}_i(t) = \begin{pmatrix} x_i(t) \\ y_i(t) \\ z_i(t) \\ \theta_i(t) \\ \dot{z}_i(t) \\ s_i(t) \\ \dot{\theta}_i(t) \end{pmatrix},$$

where (x_i, y_i, z_i) is the position, θ_i is the orientation in the $x-y$ plane, s_i is the speed in the $x-y$ plane, $\dot{z}_i$ is the vertical speed, and $\dot{\theta}_i$ is the turning rate. The dynamical model used in the UKF was determined through a series of system identification

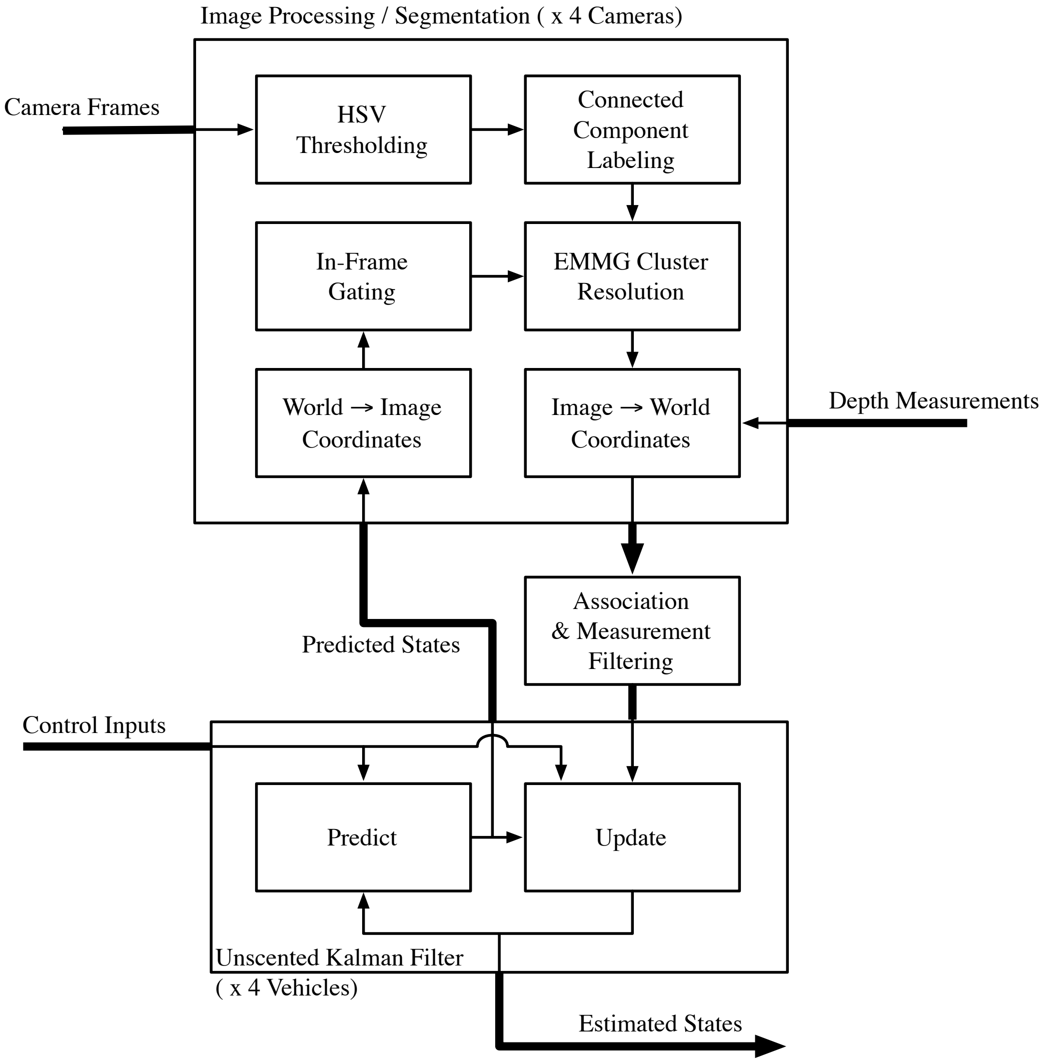


Figure 5.15: Overview of the real-time tracking algorithm used for the Beluga testbed. Thick lines represent collected measurements (i.e., multiple images, multiple vehicles). The upper box contains steps that are carried out for each of the four cameras. The lower box contains the UKF steps that are carried out for each vehicle.

experiments (see [108]). The UKF measurement input for the i^{th} vehicle is

$$\bar{\mathbf{z}}_i(t) = \begin{pmatrix} x_{i,1} \\ y_{i,1} \\ \theta_{i,1} \\ \vdots \\ x_{i,n_c} \\ y_{i,n_c} \\ \theta_{i,n_c} \\ d_0 - d_i \end{pmatrix},$$

where the subscript pair (i, k) indicates the measurement for the i^{th} vehicle from the k^{th} camera in which the vehicle appears, $1 \leq n_c \leq 4$ is the total number of cameras in which the vehicle appears, d_0 is the water depth, and d_i is the vehicle's depth measurement ($d_0 - d_i$ gives a measurement of the vehicle's height above the tank floor). The UKF assumes that the measurements are corrupted by zero-mean normally-distributed noise (very similar to what is described in Sec. 5.2.2) and that the noise is uncorrelated (i.e., its covariance matrix is a diagonal matrix). The measurement model is modified for each vehicle at each time step depending on how many cameras have a view of the vehicle.

Fig. 5.16 shows a screenshot of the software tracking two “dummy” Beluga vehicles in all four camera frames. Each dummy vehicle consists of a spare Beluga keel tied to a string. Tracking of the dummy vehicles is identical to tracking of the Beluga vehicles, except that the depth measurement is set to zero. Each image in Fig. 5.16 is the view from a single camera with tracking results overlayed. We do not calculate a merged view of the tank in real-time. The green/white arrows show the positions and orientations of the two dummy vehicles that are measured from the segmentation step (i.e., the input to the UKF). The red arrows show the filtered estimates of position and orientation of the two dummy vehicles (i.e., the output of the UKF). Note that the upper-most dummy vehicle appears to the right of the center of the tank (the white circle) in the bottom-left image and it appears to the left of the center of the tank in the bottom-right image. This is due to the geometry of the different cameras relative to the tank center and the fact that the white circle is on the bottom of the tank approximately 8 feet below the dummy vehicle, which is on the water's surface.

There are two dummy vehicles being tracked in Fig. 5.16. Both dummy vehicles appear in both the bottom-left and bottom-right camera views. The red arrow on each image of a dummy vehicle is calculated by projecting the estimated position and orientation of that vehicle back into image coordinates for that camera view using the camera's calibration data. There is only one UKF and one estimated position and orientation for each vehicle at each time instant, though the number of measurement inputs to the UKF (i.e., the size of $\bar{\mathbf{z}}_i(t)$) can change depending on the number of cameras that have a view of the vehicle. There are small differences between the filtered estimates (red arrows) and the measurements (green/white arrows). The differences are due to errors in the measurement output of the segmentation step, which are contributed to by image noise and errors in the camera calibration data.

5.4.2 The MADTraC C++ Library

The MADTraC (Multi-Agent Dynamic Tracking and Control) C++ library was developed to aid in the development and management of the large shared code base that we use in the robotic fish testbed, all of the projects listed above in Sec. 5.4.1, and a few other projects. MADTraC is designed to provide basic functionality that is needed for these projects. This functionality includes, for example

- video acquisition from file(s) and/or camera(s),
- visualization of the tracking results,

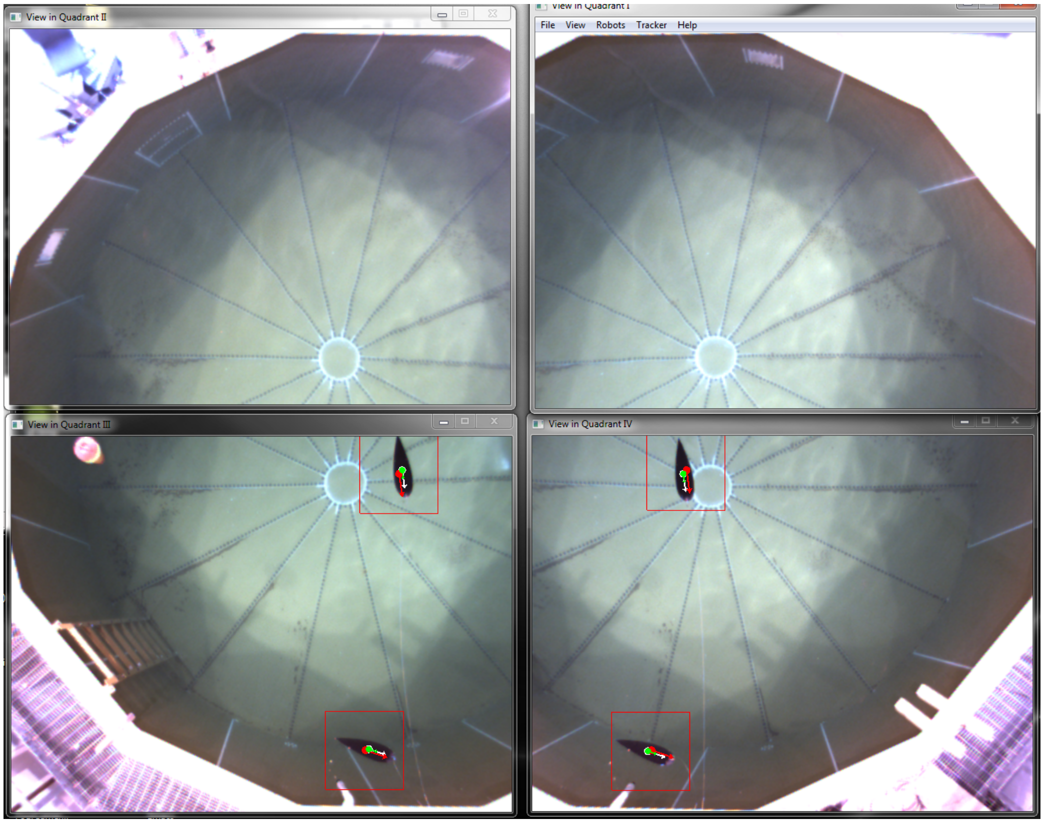


Figure 5.16: Snapshot of the real-time tracking in the Beluga testbed. Four desktop windows are shown: one window for each of the four cameras, arranged by real-world quadrants. The two vehicles appear in both the third and fourth (bottom left and right, respectively) cameras. The arrows indicate the segmentation measurements (green/white) and filtered estimates (red) of the positions and orientations of the vehicles. The red rectangle around each vehicle shows the tracking search window for that vehicle. One filtered estimate of the position and orientation is calculated for each vehicle using measurements from each camera that has a view of that vehicle. The locations and orientations of the red arrows are calculated by projecting the filtered estimates back into image coordinates using the camera calibration data.

- serial communications with robots,
- common graphical user interface (GUI) elements,
- data output formatting,
- the ability for the user to modify parameters in real-time,
- persistence of parameters across application sessions,
- the ability to save screen captures (both single frames and movies),

and more. Some of the subprocesses of the tracking algorithms described above are provided as modules in MADTraC: EMMG, connected component labeling, moment-based shape estimation, and a very flexible UKF implementation. New real-time tracking and control applications or off-line tracking applications can be developed very quickly using these components and the common GUI framework supplied by MADTraC. Because of this, MADTraC allows developers of new applications to focus on developing solutions to their own unique tracking and control problems. MADTraC is available to the public via the MIT license and can be found online at `http://github.com/leonard-lab/MADTraC/`.

MADTraC is written in C++ and makes heavy use of object-oriented programming concepts, particularly inheritance and polymorphism (see any introductory text on object oriented programming). For example, the basic GUI behavior can be easily extended by creating a new class that inherits from one of the base MADTraC GUI classes. Similarly, tracking and control classes that inherit from the MADTraC base classes are automatically managed by the framework using polymorphism.

MADTraC was designed to be platform-independent to facilitate the wide variety of platforms present in the research computing ecosystem. One of the ways that this is accomplished is by utilizing platform-independent third-party libraries. MADTraC depends heavily on two such third-party libraries: OpenCV [16] and wxWidgets [127]. OpenCV is used for image processing and matrix operations. wxWidgets is a cross-platform GUI library that provides native look-and-feel on most systems. Build configuration is accomplished using CMake [60], a tool that is used to generate platform-specific build files (e.g., Makefiles, Visual Studio projects, XCode projects, etc.) from a single configuration script. We have also written wrapper classes that provide a common interface to some of the low-level functionality such as serial communications and FireWire and USB camera interfaces. MADTraC applications can be compiled and run entirely with software development tools that are freely available and free of charge.

Development of MADTraC has stabilized in the last two years while we focus on applications such as the robotic fish testbed and the Beluga testbed. However, small changes continue to be made as we discover them while working on new applications. Furthermore, new features are occasionally “promoted” from applications to MADTraC. For example, the Beluga testbed required a flexible controller infrastructure; the solution that was developed for that testbed was then abstracted so that we can provide its functionality as part of MADTraC and therefore use it in other testbeds like the MiaBot Pro testbed. The inter-process communication module described in the following section is an example of a module that was developed for the Beluga testbed but is easily applicable to other MADTraC testbeds.

5.4.3 Inter-process Communication and Web Interfaces

In this section, we briefly describe a network layer that can be used with MADTraC-based software. The network layer consists of an inter-process communication (IPC) module that allows connections between the tracking and control software and external processes. This module was developed primarily to support the Beluga testbed, but it is very flexible and can be used in any MADTraC-based software. For example, it can be used with the robotic fish testbed or the MiaBot Pro testbed.

We were motivated to create this module to enable experiments in which multiple human subjects are able to control multiple Beluga vehicles to collaboratively accomplish a given task. Therefore we sought a solution that would allow us to exchange information between the tracking and control application, which runs on the Beluga testbed's tank-side workstation, and potentially several user clients, which could be located off-site. We accomplish this using the two-tiered server framework shown in Fig. 5.17. The first tier is an IPC server written in the Ruby programming language using the rhubarb [111] IPC server library. The second tier is a web server written in Ruby using the Ruby on Rails web framework. This two-tier design leverages the flexibility and low computational overhead of direct TCP/IP communication on the tank-side workstation where low latency is important, and benefits from the increased security and highly standardized design patterns of HTTP for external clients.

The IPC server essentially acts as a TCP/IP networked buffer with a simple string-based interface[‡]. For example, we may open a network connection from the tracking and control software to the IPC server and, at each time step, send a string of the form

```
set position i x y z
```

to inform the IPC server that the i^{th} robot is at position (x, y, z). Simultaneously, we establish a network connection from another client[§] and then can request the most recent position of the i^{th} robot from the IPC server by sending a string of the form

```
get position i
```

[‡] The operation of the IPC server is similar, but not identical, to a database management system known as a "key-value store."

[§] From the IPC server's perspective, the web server and the tracking/control software are both clients.

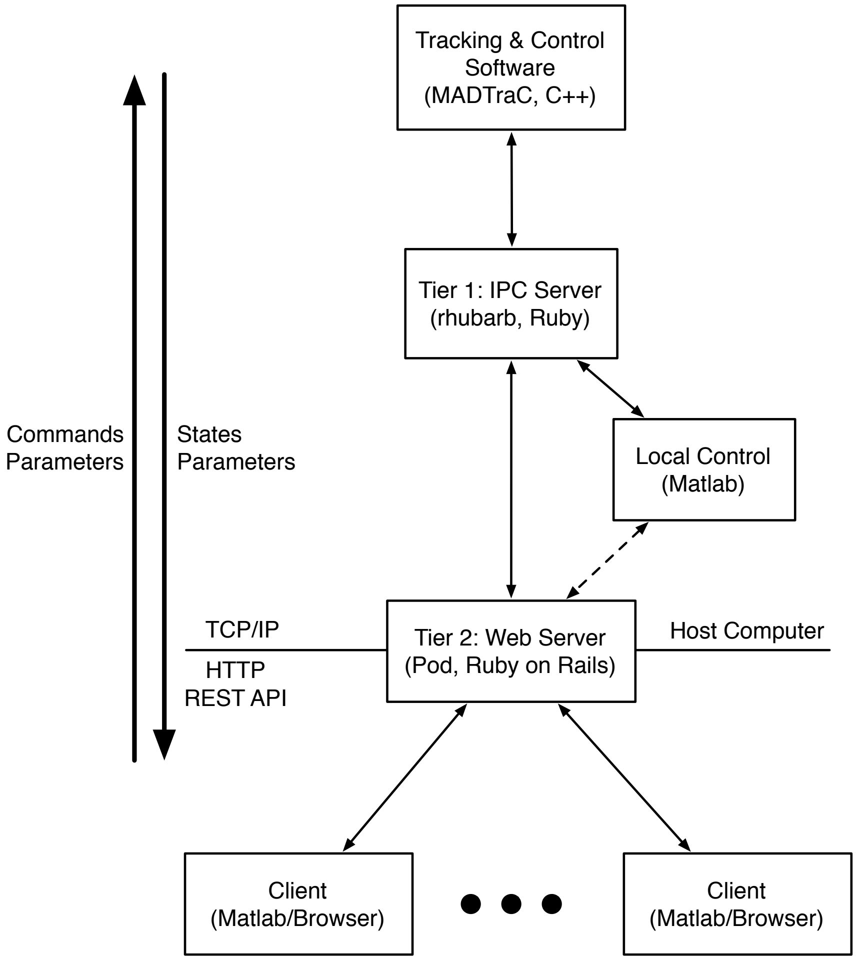


Figure 5.17: Diagram of the two-tiered server framework used on the Beluga testbed. The IPC server acts as a networked buffer with a simple string command protocol. The web server facilitates external clients, particularly browser-based clients developed around the server's REST API.

to which the IPC server sends a reply of the form

```
x y z
```

indicating that the position is (x, y, z). Similarly, the client software can send commands to the tracking and control software via the IPC server. For example, to set a waypoint for the i^{th} robot at position (u, v, w), the client sends the string

```
set control waypoint i u v w
```

and the tracking and control software can request the most recent control for the i^{th} robot by sending the string

`get control i`

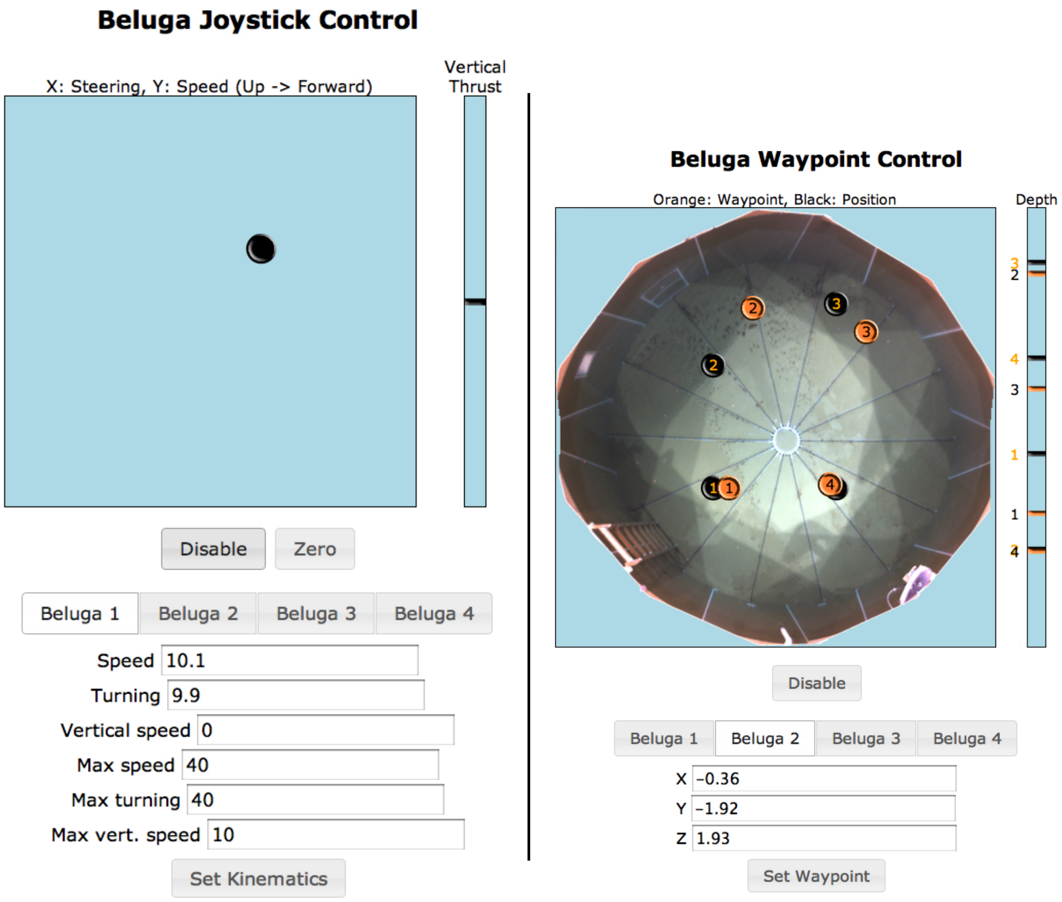
to which the IPC server responds

`waypoint u v w.`

By substituting `kinematics` for `waypoint` in the `set control` string, we can command the tracking and control software to set the speed, turning rate, and vertical speed of the i^{th} robot to u, v, and w, respectively. The Beluga project includes a C++ module that can parse from the IPC server's string protocol to numeric values and vice versa. A set of Matlab functions for communicating with the IPC server are also available.

The second-tier web server was designed to facilitate the development of web browser-based client software. The web server exchanges positions, commands, and other parameters with the tracking and control software by establishing a TCP/IP client connection to the IPC server. Typically, the web server and IPC server are both running simultaneously on the tank-side workstation. There are two user interfaces built into the web server: a joystick-like interface for driving vehicles directly, and a point-and-click interface for setting vehicle waypoints. Fig. 5.18 shows screen shots of the two user interfaces. These user interfaces were created using HTML to lay out the graphical elements and JavaScript to program the interactive components. The user interfaces can be run in any modern web browser that has JavaScript enabled, including both desktop and mobile (e.g., iPhone, iPad, etc.) browsers.

The web server is not limited to the joystick and waypoint user interfaces shown in Fig. 5.18. Almost any user interface can be created for the Beluga platform using HTML and JavaScript, which are standard tools for browser-



Speed 10.1
Turning 9.9
Vertical speed 0
Max speed 40
Max turning 40
Max vert. speed 10

X -0.36
Y -1.92
Z 1.93

(a) (b)

Figure 5.18: User interfaces built into the web server. (a) Joystick-like interface: the user sets the speed and turning rate by moving the black puck around the blue square and sets the vertical thrust by moving the black slider up and down. (b) Waypoint interface: the user can set waypoints by dragging the black pucks around the overhead image of the tank and can set the vertical waypoint by moving the black sliders up and down. The actual vehicle positions are indicated by the orange pucks/sliders, and are updated in real-time. The user can directly enter numerical values in either interface.

based applications and therefore have a large amount of learning materials available. We have implemented a RESTful API[¶] on the Beluga web server, which makes it relatively easy to write client code that accesses the positions and control parameters of the Beluga vehicles. For example, to query the position of the first vehicle, we need only make an HTTP GET request to the URL `http://pod.princeton.edu/positions/0` (the vehicle index is zero-based). To set the waypoint of the third vehicle to the location $(1, 2, 0)$, we make an HTTP POST request to the url `http://pod.princeton.edu/waypoints/2&x=1&y=2&z=0`. Typically, in a browser-based user interface these requests are made in the background using JavaScript. Then, the JavaScript parses the response from the server and updates the user interface in some way (for example, in the waypoint user interface, the orange pucks are moved to the correct location). Using JavaScript in this manner to make HTTP requests in the background and then update the user interface is called AJAX (asynchronous JavaScript and XML), and there are many JavaScript libraries that provide wrapper functions to make this relatively easy to do. We use the jQuery JavaScript library for this purpose.

Non-browser user interfaces are also possible using this framework. For example, it is possible to communicate directly with the IPC server from an instance of Matlab that is running on the tank-side workstation. This makes it possible to implement high-level control algorithms using Matlab. Furthermore, Matlab provides a set of functions for making HTTP requests and parsing the responses, therefore it is possible to write Matlab clients that run on off-site computers.

[¶] REST stands for representational state transfer and is a common design pattern for web applications. API stands for application programming interface, and an API that adheres to the REST design pattern is called a RESTful API.

Chapter 6

Conclusion and Future Work

In this dissertation we consider the role of the dynamics of relative motion on information flow within multi-agent systems. This work is inspired by observations of coordinated speed oscillations in schools of banded killifish (*F. diaphanus*). We present quantitative analysis of these observations in Ch. 2. In Ch. 3, we introduce analysis tools that enable us to explore the benefits of the type of coordinated relative motion exhibited by the killifish. The results in Ch. 3 are relevant both to the analysis of biological collective motion and to the design of engineered systems, particularly mobile sensor networks. In Ch. 4, we provide systematic methods for the design of formations of mobile agents in which the agents are engaged in coordinated speed oscillations. We introduce a testbed in Ch. 5 that facilitates further exploration of the role of dynamic relative motion by enabling robotic fish to interact in real time with a school of live fish.

Sec. 6.1 provides a more detailed presentation of the contributions of this dissertation by addressing the research questions posed in Ch. 1. In Sec. 6.2, we describe related work that is ongoing and propose some topics for future research in this area.

6.1 Summary of Contributions

We present here a summary of the contributions of this dissertation by way of addressing the research questions that were posed in Ch. 1.

How often do killifish engage in coordinated speed oscillations, under what conditions does this happen, and what are the average properties of these oscillations?

In Ch. 2, we introduce an algorithm that uses the Hilbert transform to detect time segments during which killifish are engaging in speed oscillations by calculating the instantaneous phase of the speed of each fish. By determining when multiple fish are oscillating their speeds simultaneously at similar frequencies, we are able to isolate sequences during which two or more killifish are engaged in coordinated speed oscillations. We find that the killifish are engaged in coordinated speed oscillations a majority of the time, and that the oscillations have an average frequency of approximately 0.75 Hz and an average amplitude of approximately one half of the average speed (see Table 2.2). Coordinated speed oscillations in two-fish schools tend to be anti-synchronized. The phase relationship between speed oscillations in three-fish schools is complex: there is always periodic relative motion present and typically at least one pair of fish exhibits speed oscillations that are anti-synchronized.

We also find that killifish tend to move relative to one another so that the relative bearing from a fish to its neighbor in the front is approximately 35-40 degrees (see Fig. 2.6). Furthermore, because the fish are engaged in coordinated speed oscillations, they continuously move relative to one another and their spatial formations are correspondingly modulated periodically in time. See Figs. 2.3- 2.5 for examples of two- and three-fish spatial formations that result from coordinated speed oscillations.

Why do killifish engage in coordinated speed oscillations, how does relative motion affect information passing in mobile groups in general, and can we apply these results to improve engineered systems?

We show in Ch. 3 that killifish coordinated speed oscillations produce periodically time-varying sensing topologies. Under certain assumptions about how the sensing topologies are determined by the positions of the fish relative to one another, we show that these time-varying sensing topologies result in a significant increase in the speed at which the group can share information. We show this by computing the rate of convergence to consensus over the periodically switching graphs representing the sensing topologies. The rate of convergence to consensus is up to 50% greater with this switching than it would be without switching. This suggests that speed oscillations, and periodic relative motion in general, may serve to improve information passing in mobile animal groups.

We also leverage the tools that we develop in Ch. 3 to study information-passing benefits resulting from relative motion patterns and sensing conditions that are different from those that we have assumed for the killifish. These tools are especially useful for engineered systems, such as multi-agent robotic systems, where the communication and feedback mechanisms are known by design. For example, we show that the sensing radius of an agent that is required to maintain connectivity can be significantly reduced by introducing periodic relative motion among the agents (see, for example, Fig. 3.6). We also show how to design the parameters of periodic relative motion for systems with periodic relative motion in order to maximize decision making-performance (see Sec. 3.3.3).

In Ch. 4, we show that groups of mobile agents with constant turning rates that are engaged in coordinated speed oscillations can yield motion patterns in which the agents regularly cycle around a formation shape as the center of the formation shape moves around a circle at a constant speed. Each individual agent in this type of motion pattern periodically cycles its position relative to the center of the circle and the other agents. That is, as a given individual agent moves around the formation shape, it cycles through positions closer to the center of the circle than the other agents, at the rear of the group, towards the outside of the circle, and at the front of the group. By cycling around the formation shape in this way, each agent balances risk (being at the outside of the circle) with safety (being at the inside of the circle) and leadership (being at the front of the group) with the ability to infer visual information from the group (when at the rear of the group). Furthermore, these kinds of formations can improve mobile sensor network performance by enhancing the spatial distribution of measurements of each agent and by introducing redundancy. That is, if one or more agents are removed from the formation, the remaining agents can be redistributed by dynamically adjusting the distribution of their speed phase arrangement.

How can we characterize the trajectories of individual agents undergoing periodic speed modulations, can we design control laws to stabilize such agents into cohesive formations, and how can we design the speed modulation of a group of N agents to achieve a desired formation?

We develop methods to characterize the trajectories of individual agents undergoing planar motion with constant turning rates and periodic speed modulations in Sec. 4.2. The trajectory of a speed-modulated particle is described in terms of its motion relative to a constant-speed virtual particle. The virtual particle has the same constant turning rate as the speed-modulated particle and its speed is equal to the average speed of the speed-modulated particle. The two particles have the same initial heading and the initial position of the virtual particle is chosen so that the centers of both trajectories are the same. We first show that when the speed modulation is sinusoidal, the locus of the motion of the speed-modulated particle in a frame that is body-fixed and oriented with the constant-speed particle defines an ellipse. The turning rate and the amplitude and frequency of the speed oscillation determine the semi-major and semi-minor axis lengths of the ellipse. Because the virtual particle has constant speed, its trajectory is along a circle whose radius is determined by its speed and turning rate. Therefore, the trajectory of the particle with sinusoidal speed oscillation is a composition of a circular component (due to the average speed) and an elliptical component (due to the speed oscillation): the particle moves around an ellipse that is centered about a point along the constant-speed circle and this ellipse rotates as the particle moves around the circle (see Fig. 4.3). The position of the particle along the ellipse is determined by its speed phase; as the speed cycles through 2π radians, the particle makes a complete cycle around the ellipse. For more complex speed profiles, we can construct the trajectory with a similar composition: a circular component due to the average speed and a series of elliptical components due to the oscillation's Fourier components. The superposition of the elliptical components defines a closed curve that the particle moves around once as its speed phase completes a 2π cycle. We show that there is a direct mapping between the Fourier coefficients of the speed profile and the shape of this closed curve. We also present results on the boundedness of trajectories and the existence of periodic trajectories.

We develop control laws in Sec. 4.3 to systematically stabilize a large variety of planar formations of N individual agents with periodic speed oscillations where all agents share a common constant turning rate and speed oscillation frequency. In Sec. 4.4.1 we explore the geometries of the family of formations that can be stabilized using the control laws in Sec. 4.3 assuming that the speed profile is given. In Sec. 4.4.2, we explore the design of speed profiles to obtain desired formations. In particular, we consider the case where it is desired that the agents move around a common circular trajectory with synchronized headings and splayed speed phases (see, for example, Fig. 4.14). That is, the agents are distributed at different locations around the same elliptical trajectory components and therefore the locus of those components describes the shape of the formation. Following the approach used above for trajectory analysis, we begin by solving the purely sinusoidal case and then expand to the general periodic speed profile case. For the sinusoidal speed case, we derive expressions for the turning rate, speed oscillation frequency, and speed oscillation amplitude that are required to produce a formation ellipse with given dimensions (i.e., the semi-major and semi-minor axis lengths of the ellipse) when the nominal speed of the agents and the radius of the circular component of the trajectory is fixed. We also provide bounds on what formations are achievable. We use the sinusoidal speed case as a basis to construct an algorithm for determining the motion parameters (including the Fourier coefficients of the speed profile) to achieve more complex designed trajectories (see, for example, Figs. 4.14 and 4.16). Our solution to this design problem allows for the desired formation shape to be drawn by hand using a computer program.

Can an experimental testbed be developed in which robotic fish interact in real-time with a school of live fish, and in what other settings can this technology be applied?

We describe the design of such a testbed in Ch. 5. In our testbed, one or more mobile robots moves beneath a tank of shallow water. Each robot moves one model fish about the tank via magnetic coupling between the robot and the fish. See Fig. 5.1 for a diagram of the testbed. We call each mobile robot/model fish combination a "robotic fish." Closed-loop control of the robotic fish is accomplished by sending wheel speed commands over a Bluetooth radio channel, where the wheel speed commands are calculated in response to real-time estimates of the dynamic states of the robotic fish and the live fish school. We have developed and implemented computer vision algorithms (see Sec. 5.2) to calculate these estimates in real-time, and have demonstrated their use in proof-of-concept experiments (see Sec. 5.3).

We developed this technology in such a way that it can be adapted easily to other applications. We have used the technology developed for the robotic fish testbed for off-line fish tracking for experiment analysis, dancer tracking for analysis of human flocking experiments, a multiple wheeled robot testbed, and a multi-AUV testbed (see Sec. 5.4). All of these applications share a large amount of C++ source code, which we have extracted to a library that we call MADTraC and describe in Sec. 5.4.2. We have also created a framework that allows other pieces of software to communicate with

MADTraC applications (see Sec. 5.4.3). This enables, for example, high-level control of the tracked robots from Matlab and remote user interfaces that communicate through a web server.

6.2 Ongoing and Future Work

In this section we describe some of the open topics of research related to the work presented in this dissertation. The discussion is broken down roughly into topics related to natural systems in Sec. 6.2.1 and engineered systems in Sec. 6.2.2.

6.2.1 Natural Systems

The analyses and results regarding coordinated speed oscillations that we present in Ch. 2 and Ch. 3 are based on data for schools of two and three fish. More data could lead to stronger results and further illuminate the mechanisms and benefits of this behavior. We therefore propose the collection of additional trajectory data for schools of killifish of varying sizes. We possess a limited amount of additional video of killifish schools that has not yet been tracked. New experiments could also be performed. It may also be possible to use a larger arena or a flow channel to collect segments of longer duration during which the fish are swimming in a straight line. The robotic fish testbed described in Ch. 5 could be used to enrich the data set. The robotic fish could be programmed to vary the frequency of its speed oscillations to test for oscillation cohesion (i.e., monitor whether the other fish change their oscillation frequencies to match the robotic fish). By changing the speed oscillation phase of the robotic fish, we could query the preference for anti-synchronization with near neighbors.

Our results regarding decision-making benefits for killifish in Sec. 3.2.2 are based on specific assumptions about the way that fish influence one another depending on their relative positions and motion. The assumption that a fish has only one neighbor on either side and that the most recently unoccluded neighbor takes preference (assumption 2 in Sec. 3.2.2) is critical to our results. If, on the other hand, the nearest fish is always preferred or all visible fish are considered neighbors, then decision-making performance is not improved by relative motion. The fact that the conditions of our assumption *do* lead to a decision-making improvement could suggest that our assumption (or part of it) is correct. This notion could be pursued further, either to show that fish are most influenced by their most recently unoccluded neighbor, or to lead to improved assumptions about the communication topology.

The tracking methods that we describe in Ch. 5 and the analysis tools that we describe in Ch. 2 can be used to obtain and analyze biological trajectory data for other settings in addition to fish trajectory analysis. For example, we have already proven its use for obtaining and analyzing trajectory data for experiments involving human subjects. As discussed in Ch. 5, our tracking methods allow for a high throughput rate under severe conditions of occlusion. The state-of-the-art is advancing quickly, however, and new technologies and computational paradigms appear frequently. The MADTraC library is designed for flexibility and should be maintained and updated to incorporate new ideas that can improve tracking performance. Additionally, we believe that MADTraC is an ideal platform to share with the community because it focuses on providing an abstracted software layer that is geared towards allowing researchers to create customized high-performance tracking and control solutions. MADTraC can also be integrated with existing tracking software in both the biological and engineered settings. MADTraC is available online at `http://github.com/leonard-lab/MADTraC/` under the MIT license.

In this dissertation we focus on the information passing benefits of coordinated relative motion. Other potential benefits should be investigated. For example, position within animal aggregations has been linked to evolutionary benefits (see, for example, [50]). Animals on the inside of the group are safe from predators, while animals on the outside of the group may have more opportunities to obtain food. Speed oscillations could provide a mechanism for animals in groups to modulate their position within the group, hence balancing these benefits.

6.2.2 Engineered Systems

Our observations and analysis of killifish coordinated speed oscillations have directly inspired methodologies for the design of multi-agent systems. We believe that further study of dynamic relative motion will lead to more bio-inspired results for engineered systems.

We show in Ch. 3 that periodic relative motion can significantly improve decision-making performance in the context of mobile sensor networks. The results presented there are focused on small groups and the concepts should scale to larger groups. The context of that discussion is communication that requires line-of-sight visibility and hence all-to-all communication is impossible in larger groups. Many engineered systems use radio communication which does not have this limitation. Nonetheless, radio communication systems can experience problems when scaling to large mobile sensor networks and hence designers can benefit from methodologies that reduce communication demands without sacrificing group performance. We show in Sec. 3.3 that periodic relative motion can reduce the required sensing radius to maintain group-level connectivity, and we explore design optimization of group-level decision-making performance. It could be valuable to study under what more general graph-theoretic conditions does periodic switching of communication topologies improve the rate of convergence to consensus. It would also be useful to examine the role of periodically time-varying topologies in terms of other metrics of performance and robustness, such as accuracy and noise rejection.

The planar motion patterns that we study in Ch. 4 have a circular component that corresponds to the constant turning rate and average speed of the agents. One useful extension to these motion patterns is to modify the control laws so that the formation is maintained while the circular component is replaced with a more generic shape. In this way, mobile sensors in formations such as the ones studied in Ch. 4 could be controlled to move around, for example, boundaries of oil spills or ocean isotherms. The spatial distribution of the sensors can enhance measurement performance (see, for example, [130, 131]). The regular role exchanges that we discuss above in the context of biological systems may have analogs for engineered systems. Different agents may be equipped with different sensors and cycling positions along the formation allows all sensors to visit locations both inside and outside the circular component of the trajectory (e.g., the boundary, isotherm, etc.). A level of redundancy is also added: losing a single agent does not correspond to a loss of sensing in a specific region of the formation. In fact, the control laws can be adjusted to redistribute the agents around the formation if one agent is lost.

The technology that we have developed for the robotic fish testbed has already proven useful in other testbeds, as described in Sec. 5.4.1. Our tracking algorithms work well, although there is still room for improvement. Furthermore, new techniques and technological advances can be incorporated to improve the performance and robustness of the system. For example, graphical processing units (GPUs) represent a significant opportunity to accelerate the speed of image processing algorithms. Improvements to the inter-process communication layer described in Sec. 5.4.3 have the potential to enable a wider variety of experiments using this technology. The architecture of our solution is influenced by the technologies that power web services such as Twitter and Facebook that are required to scale to handle thousands or even millions of simultaneous users with very low latency. This technological arena is constantly evolving with new and disruptive technologies emerging often. Technologies such as Redis (a highly scalable and fast data storage system) and Node.js (a very low latency and scalable server framework) could be leveraged to improve the performance of our system.

Appendix A

Relationship Between MSP and Phase Separation (Proof of Thm. 2.1)

The following is a proof of Thm. 2.1, which states that the MSP for a pair of sinusoids δs_1 and δs_2 with a phase separation $|\Delta\phi| \leq \pi$ is approximately

$$\mathrm{MSP}\left(\delta s_1, \delta s_2\right) \approx 1 - 2\frac{|\Delta\phi|}{\pi}.$$

Note that $|\Delta\phi| \leq \pi$ is unambiguous. For any values not satisfying this condition, we may equivalently substitute $\Delta\phi \longleftarrow (\Delta\phi)_{\mathrm{mod}\ 2\pi}$ and/or exchange the labels so that $\Delta\phi \longleftarrow 2\pi - \Delta\phi$. That is, the phase separation $|\Delta\phi|$ is the angular length of the shortest arc joining the phasor angles of the two sinusoids on the unit circle.

Consider two sinusoidal signals δs_1 and δs_2 with a common frequency Ω. The sinusoids are sampled with a sampling period of T_s. For simplicity, assume that the period of oscillation is exactly equal to m_1 samples, where m_1 is an integer. That is, $m_1 T_s = \frac{2\pi}{\Omega}$. Furthermore, assume that the time interval in question consists of exactly $N_t = m_1 m_2$ samples for some integer m_2. That is, the data represents an integer number of periods of oscillation. Therefore, the average values of the oscillations are $\delta\bar{s}_1 = \delta\bar{s}_2 = 0$.

Let $\mu_1 > 0$ be the amplitude of δs_1 and $\mu_2 > 0$ be the amplitude of δs_2. For simplicity, we assume that the start of sampling is synchronized with the start of oscillation so that

$$\delta s_1[t] = \mu_1 \cos \Omega T_s t$$

and, without loss of generality,

$$\delta s_2[t] = \mu_2 \cos\left(\Omega T_s t - |\Delta\phi|\right).$$

This assumption introduces error into our approximation, although this error is minimal when T_s is sufficiently large. Therefore, the sinusoids are described by where $\mu_1 > 0$ and $\mu_2 > 0$ are constant amplitudes and $\Delta\phi$ is the phase difference between the two oscillations.

The MSP for δs_1 and δs_2, as defined by (2.5), is

$$\begin{aligned}
\mathrm{MSP}(\delta s_1, \delta s_2) &= \frac{1}{N_t}\sum_{t=0}^{N_t-1} \operatorname{sign}\left\{\mu_1\mu_2 \cos\left(\Omega T_s t\right)\cos\left(\Omega T_s t - |\Delta\phi|\right)\right\} \\
&= \frac{1}{2m_1m_2}\operatorname{sign}\left\{\mu_1\mu_2\right\}\sum_{t=0}^{m_1m_2-1} \operatorname{sign}\left\{\cos\left(2\Omega T_s t - |\Delta\phi|\right) + \cos|\Delta\phi|\right\} \\
&= \frac{1}{2m_1}\sum_{t=0}^{m_1-1} \operatorname{sign}\left\{\cos\left(2\Omega T_s t - |\Delta\phi|\right) - \cos\left(\pi - |\Delta\phi|\right)\right\} \\
&= \frac{1}{m_1}\left[(m_1 - \widetilde{m}) - \widetilde{m}\right],
\end{aligned}$$

where $\widetilde{m}$ is the number of samples for which

$$\cos\left(2\Omega T_s t - |\Delta\phi|\right) < \cos\left(\pi - |\Delta\phi|\right),\ 0 \le t \le m_1 - 1.$$

Because m_1 is the number of samples in two periods of $\cos(2\Omega T_s t - |\Delta\phi|)$, we have

$$\widetilde{m} = \left\lfloor m_1 \frac{|\Delta\phi|}{\pi} \right\rfloor,$$

and therefore

$$\begin{aligned}\mathrm{MSP}(\delta s_1, \delta s_2) &= \frac{1}{m_1}\left(m_1 - 2\left\lfloor m_1 \frac{|\Delta\phi|}{\pi}\right\rfloor\right)\\ &\approx 1 - 2\frac{|\Delta\phi|}{\pi},\end{aligned}$$

where we have made the approximation $\left\lfloor m_1 \frac{|\Delta\phi|}{\pi}\right\rfloor \approx m_1 \frac{|\Delta\phi|}{\pi}$. The accuracy of this approximation increases as T_s increases.

Because we have assumed that $m_1 T_s = \frac{2\pi}{\Omega}$ and $N_t = m_1 m_2$, there may be additional errors between the actual value of the MSP and (2.6). These errors are minimal for sufficiently large sample sizes (large N_t compared to the number of samples in a period of oscillation) and for sufficiently small sampling periods (small T_s compared to the period of oscillation).

Appendix B

Visual Cues and Their Sensitivities

The following is a rederivation of some of the results of Dill et al. [37]. The set up is described in Sec. 2.2.6. We are essentially computing expressions for the kinematics of the image of the leader fish formed on the eye of the focal fish, from which we can derive the various optical signals of interest and their sensitivities.

For convenience, we establish two dextral coordinate systems, $\mathcal{I}_X$ and $\mathcal{I}_R$. $\mathcal{I}_X$ is defined by the set of unit vectors $\{\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}}\}$ and $\mathcal{I}_R$ by $\left\{\hat{\mathbf{r}}, \hat{\beta}, \hat{\mathbf{z}}\right\}$. The $\hat{\mathbf{z}}$ unit vector in both frames points perpendicularly out of the plane of motion such that $\hat{\mathbf{z}} = \hat{\mathbf{x}} \times \hat{\mathbf{y}} = \hat{\mathbf{r}} \times \hat{\beta}$. The frame $\mathcal{I}_X$ is aligned with the focal fish's velocity $\mathbf{v}_F$ such that

$$\mathbf{v}_F = s_F \hat{\mathbf{x}}, \tag{B.1}$$

where s_F is the scalar speed of the focal fish. $\mathcal{I}_R$ is defined so that $\hat{\mathbf{r}}$ is always aligned with the vector $\mathbf{r}_{lf}$ joining the eye of the focal fish with the eye of the leader, i.e.

$$\mathbf{r}_{lf} = d\hat{\mathbf{r}} = x\hat{\mathbf{x}} + y\hat{\mathbf{y}},$$

where d is the distance between (line-of-sight visible) eyes, and (x, y) gives the coordinates of $\mathbf{r}_{lf}$ in $\mathcal{I}_X$. The relative bearing of the leader's eye is β and hence the rotation angle in the plane orthogonal to $\hat{\mathbf{z}}$ between $\mathcal{I}_X$ and $\mathcal{I}_R$ is β.

Note that $\mathcal{I}_R$ is a moving frame and therefore we must account for its motion when taking derivatives with respect to time. Because $\mathcal{I}_R$ is a rotation of $\mathcal{I}_X$ about the $\hat{\mathbf{z}}$-axis by β, the dynamics of the $\mathcal{I}_R$ frame are simply given by

$$\begin{aligned} \frac{d}{dt}\hat{\mathbf{r}} &= \dot{\beta}\hat{\beta}, \\ \frac{d}{dt}\hat{\beta} &= -\dot{\beta}\hat{\mathbf{r}}. \end{aligned}$$

Without loss of generality, we assume that the focal fish has a zero heading. The leader is assumed to be moving with a speed s_L with heading θ. Hence, leader's velocity $\mathbf{v}_L$ can be expressed as

$$\mathbf{v}_L = s_L \cos\theta \hat{\mathbf{x}} + s_L \sin\theta \hat{\mathbf{y}} = s_L \cos(\theta - \beta)\,\hat{\mathbf{r}} + s_L \sin(\theta - \beta)\,\hat{\beta}. \tag{B.2}$$

The velocity of the leader's eye relative to the focal fish's eye is equal to the relative velocity

$$\mathbf{v}_{lf} = \mathbf{v}_L - \mathbf{v}_F. \tag{B.3}$$

The polar geometry of the leader's eye relative to the focal fish's eye determines the image of the leader's eye on the retina of the focal fish, hence we are primarily interested in the polar coordinates (d, β) and their velocities and accelerations. Therefore, we derive expressions for $\dot{d}$ and $\dot{\beta}$ in terms of the geometry and motion parameters.

The velocity of the leader relative to the focal fish can be written in polar coordinates as

$$\mathbf{v}_{lf} = \frac{d}{dt}\mathbf{r}_{lf} = \dot{d}\hat{\mathbf{r}} + d\dot{\beta}\hat{\beta}.$$

Therefore, substituting (B.2) and (B.1) into (B.3), we have

$$\begin{aligned}\dot{d} &= \mathbf{v}_{lf} \cdot \hat{\mathbf{r}} \\ &= s_L \cos(\theta - \beta) - s_F \cos\beta \qquad \text{(B.4)}\end{aligned}$$

and

$$\begin{aligned}\dot{\beta} &= \frac{1}{r}\mathbf{v}_{lf} \cdot \hat{\beta} \\ &= \frac{1}{r}\left(s_L \sin(\theta - \beta) + s_F \sin\beta\right). \qquad \text{(B.5)}\end{aligned}$$

B.1 Angular velocity

The angular velocity of the retina angle is

$$B = \dot{\beta}$$

From (B.5), we have

$$B = \frac{s_L \sin(\theta - \beta) + s_F \sin\beta}{d}.$$

For initially parallel motion, the offset y is fixed and we are interested in optimal locations in terms of the bearing β. Using the substitution $y = d \sin\beta$, we have

$$\begin{aligned}B &= \sin\beta \frac{s_L \sin(\beta - \theta) + s_F \sin\beta}{d \sin\beta} \\ &= \frac{s_F \sin^2\beta - s_L \sin\beta \sin(\beta - \theta)}{y},\end{aligned}$$

which matches the expression obtained in [37] (note that our y, s_F, and s_L are equivalent to x_0, v_F, and v_L in [37], respectively). Note that there is no angular velocity for parallel motion and common speeds, i.e., $B = 0$ when $\theta = 0$ and $s_L = s_F$.

The sensitivity of angular velocity to a change in the leader's heading is

$$\frac{\partial B}{\partial \theta} = \frac{s_L \sin\beta \cos(\beta - \theta)}{y}.$$

For initially parallel motion, we have

$$\left.\frac{\partial B}{\partial \theta}\right|_{\theta=0} = \frac{s_L \sin 2\beta}{2y}, \tag{B.6}$$

which matches the result in [37]. The relative bearing $\beta = \frac{\pi}{4}$ maximizes the sensitivity of angular velocity to a change in the leader's heading over the range $0 < \beta \leq \frac{\pi}{2}$.

The sensitivity of angular velocity to a change in the leader's speed is

$$\frac{\partial B}{\partial s_L} = -\frac{\sin\beta \sin(\beta - \theta)}{y}.$$

For initially parallel motion, we have

$$\left.\frac{\partial B}{\partial s_L}\right|_{\theta=0} = -\frac{\sin^2\beta}{y}, \tag{B.7}$$

again matching the result in [37]. The relative bearing $\beta = \frac{\pi}{2}$ maximizes the magnitude of the sensitivity of angular velocity to a change in the leader's speed over the range $0 < \beta \leq \frac{\pi}{2}$.

B.2 Loom

Loom is the time rate of change of the solid angle subtended by some object on the retina. As in [37], the object of interest is eye of the leader fish. The solid angle, α, subtended by the eye surface, $\mathcal{S}$, is

$$\alpha = \iint\limits_{\mathcal{S}} \frac{\mathbf{r}\cdot\hat{\mathbf{n}}}{|\mathbf{r}|^3} dS$$

where dS is a differential element of area located at a position $\mathbf{r}$ with a normal vector $\hat{\mathbf{n}}$. In this case, we have $\mathbf{r} = d\cos\beta\hat{\mathbf{x}} + d\sin\beta\hat{\mathbf{y}}$ and $\hat{\mathbf{n}} = \sin\theta\hat{\mathbf{x}} - \cos\theta\hat{\mathbf{y}}$, thus

$$\alpha = \iint\limits_{\mathcal{S}} \frac{1}{r^2}\sin\left(\beta-\theta\right) dS.$$

Considering the area to be sufficiently small such that d and β may be treated as constants over it, we are left with

$$\alpha = \frac{\sin\left(\beta-\theta\right)}{d^2} \iint\limits_{\mathcal{S}} dS = E\frac{\sin\left(\beta-\theta\right)}{d^2}, \tag{B.8}$$

where E is the area of the eye. For simplicity, we will assume that $E = 1$. This only affects the scaling of the quantities of interest and not the values of relative bearing that maximize them.

By the chain rule, the loom is equal to

$$\Lambda = \dot{\alpha} = \frac{\partial\alpha}{\partial d}\dot{d} + \frac{\partial\alpha}{\partial\beta}\dot{\beta} + \frac{\partial\alpha}{\partial\theta}\dot{\theta}. \tag{B.9}$$

The partial derivatives are

$$\begin{aligned} \frac{\partial\alpha}{\partial d} &= \frac{2\sin\left(\theta-\beta\right)}{d^3} \\ \frac{\partial\alpha}{\partial\beta} &= \frac{\cos\left(\theta-\beta\right)}{d^2} \\ \frac{\partial\alpha}{\partial\theta} &= -\frac{\cos\left(\theta-\beta\right)}{d^2}. \end{aligned}$$

Thus, substituting these expressions into (B.9) along with (B.4) and (B.5), and substituting $y = d\sin\beta$, the loom is equal to

$$\Lambda = \frac{\sin^3\beta}{y^3}\left[3s_L\sin(\theta-\beta)\cos(\theta-\beta) + s_F\cos(\theta-\beta)\sin\beta - 2s_F\cos\beta\sin(\theta-\beta)\right] - \frac{\sin^2\beta\cos(\theta-\beta)}{y^2}\dot{\theta}. \tag{B.10}$$

There is no loom for parallel motion and common speeds, i.e., $\Lambda = 0$ when $\theta = 0$ and $s_L = s_F$.

The sensitivity of loom to a change in the leader's heading is

$$\frac{\partial\Lambda}{\partial\theta} = \frac{\sin^3\beta}{y^3}\Big[3s_L\left(\cos^2(\beta-\theta) - \sin^2(\theta-\beta)\right) - s_F\sin(\theta-\beta)\sin\beta - 2s_F\cos\beta\cos(\theta-\beta)\Big] + \frac{\sin^2\beta\sin(\theta-\beta)}{y^2}\dot{\theta}. \tag{B.11}$$

For initially parallel motion and common speed, we have

$$\left.\frac{\partial\Lambda}{\partial\theta}\right|_{\substack{\dot{\theta}=\theta=0\\ s_L=s_F}} = \frac{s_F}{y^3}\sin^3\beta\left(1-3\sin^2\beta\right), \tag{B.12}$$

which matches the result in [37]. The relative bearing $\beta = \frac{\pi}{2}$ maximizes the magnitude of the sensitivity of loom to a change in the leader's heading over the range $0 < \beta \leq \frac{\pi}{2}$.

The sensitivity of loom to a change in the leader's speed is

$$\frac{\partial\Lambda}{\partial s_L} = \frac{\sin^3\beta}{y^3}\left[3\sin(\theta-\beta)\cos(\theta-\beta)\right].$$

For initially parallel motion, we have

$$\left.\frac{\partial\Lambda}{\partial s_L}\right|_{\theta=0} = \frac{-3\sin^4\beta\cos\beta}{y^3}. \tag{B.13}$$

The result in [37] has y^2 in the denominator. This is assumed to be a typo. The relative bearing $\beta = 2\tan^{-1}\sqrt{\frac{1}{2}\left(3-\sqrt{5}\right)}$ radians or about 63.4 degrees maximizes the magnitude of the sensitivity of loom to a change in the leader's speed over the range $0 < \beta \leq \frac{\pi}{2}$.

B.3 Time-to-collision

Time-to-collision, τ, is defined as two times the inverse of the relative loom, λ. The relative loom is defined as

$$\lambda = \frac{\Lambda}{\alpha}$$

and the time-to-collision as

$$\tau = \frac{2}{\lambda} = 2\frac{\alpha}{\Lambda}.$$

When the two fish are moving in parallel and at the same speed, there is no loom on the focal fish's eye; that is, $\Lambda = 0$. Therefore, under the same conditions, the time-to-collision is infinite. This leads to complications in calculating the sensitivities of the time-to-collision. The authors of [37] introduce a small purturbation to the heading of the leader's velocity, i.e., $\theta = \Delta\theta \ll 1$. Using this, they obtain the following expression for sensitivity of time-to-collision with respect to a change in the leaders heading.

$$\frac{\partial \tau}{\partial \theta} = -\frac{2y}{s_F}\frac{1}{1-3\sin^2\beta}\left(\Delta\theta\right)^{-2},$$

where they have neglected higher order terms in $\left(\Delta\theta\right)^{-1}$. Using the same methods, the authors of [37] obtain

$$\frac{\partial \tau}{\partial s_L} = \frac{6y}{s_F^2}\frac{\sin\beta\cos\beta}{\left(1-3\sin^2\beta\right)^2}\left(\Delta\theta\right)^{-2}$$

as the sensitivity of time-to-collision with respect to a change in the leader's speed. Both sensitivities have in their denominator the term $1 - 3\sin^2\beta$, which has a root at $\beta \approx 35.26$ degrees. Therefore, the sensitivities are maximized at the relative bearing $\beta \approx 35.26$ degrees.

References

[1] Allied Vision Technologies. Guppy F-080: FireWire camera. http://www.alliedvisiontec.com/us/products/cameras/firewire/guppy/f-080bc.html, 2011.

[2] I. Aoki. A simulation study on the schooling mechanism in fish. *Bull. Jpn. Soc. Sci. Fish*, 48:1081–1088, 1982.

[3] I. Bajec and F. Heppner. Organized flight in birds. *Animal Behaviour*, 78:777–789, 2009.

[4] T. Balch, F. Dellaert, A. Feldman, A. Guillory, C. Isbell, Z. Khan, S. Pratt, A. Stein, and H. Wilde. How multi-robot systems research will accelerate our understanding of social animal behavior. *Proc. of the IEEE*, 94:1445–1463, 2006.

[5] P. Balister, B. Bollobas, and M. Walters. Continuum percolation with steps in the square or the disc. *Random Structures and Algorithms*, 26:392–403, 2005.

[6] M. Ballerini, N. Cabibbo, R. Candelier, A. Cavagna, E. Cisbani, I. Giardina, V. Lecomte, A. Orlandi, G. Parisi, A. Procaccini, M. Viale, and V. Zdravkovic. Interaction ruling animal collective behaviour depends on topological rather than metric distance: Evidence from a field study. *Proc. National Academy of Sciences*, 105(4):1232–1237, 2008.

[7] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occuring in the statistical analysis of probabalistic functions of Markov chains. *Annals of Mathematical Statistics*, 41(1):164–171, 1970.

[8] A. L. Bertozzi, M. Kemp, and D. Marthaler. Determining environmental boundaries: Asynchronous communication and phyiscal scales. In V. Kumar, N. E. Leonard, and A. Morse, editors, *Cooperative Control, A Post-Workshop Volume: 2003 Block Island Workshop on Cooperative Control*, pages 35–42. Springer-Verlag, New York, 2005.

[9] D. Bertsekas and J. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, 1989.

[10] J. A. Bilmes. A gentle tutorial of the EM algorithm and its application to parameter estimation for gaussian mixture and hidden Markov models. Technical Report TR-97-021, Department of Electrical Engineering and Computer Science, U.C. Berkeley, Berkeley, CA, April 1998.

[11] C. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford, 1995.

[12] V.D. Blondel, J.M. Hendrickx, A. Olshevsky, and J.N. Tsitsiklis. Convergence in multiagent coordination, consensus, and flocking. In *Proc. 44th IEEE Conf. on Decision and Control*, December 2005.

[13] Bluetooth Special Interest Group. Specification of the Bluetooth system version 4.0. `http://www.bluetooth.org`, June 2010.

[14] B. Boashash. Estimating and interpreting the instantaneous frequency of a signal. I. Fundamentals. *Proceedings of the IEEE*, 80(4):520 –538, apr 1992.

[15] B. Bollobas and O. Riordan. *Percolation*. Cambridge University Press, New York, 2006.

[16] G. Bradski. OpenCV (Open Computer Vision) library. `http://opencv.willowgarage.com/wiki/Welcome`, March 2011.

[17] K. Branson, A. A. Robie, J. Bender, P. Perona, and M. Dickinson. High-throughput ethomics in large groups of *drosophila*. *Nature Methods*, 6:451–457, 2009.

[18] C. M. Breder, Jr. Certain effects in the habits of schooling fishes, as based on the observation of Jenkinsia. *American Museum Novitates*, (382):1–5, 1929.

[19] C. M. Breder, Jr. Equations descriptive of fish schools and other animal aggregations. *Ecology*, 35(3):361–370, 1954.

[20] C. M. Breder, Jr. Fish schools as operational structures. *Fishery Bulletin*, 74(3):471–502, 1976.

[21] F. P. Bretherton, R. E. Davis, and C. B. Fandry. A technique for objective analysis and design of oceanographic experiments applied to mode-73. *Deep Sea Research and Oceanographic Abstracts*, 23(7):559–582, 1976.

[22] S. Butail, N. Manoukis, M. Diallo, J. M. Ribeiro, T. Lehmann, and D. A. Paley. Reconstructing the flight kinematics of swarming and mating in wild mosquitos. *J. R. Soc. Interface*, 2012. In Press.

[23] S. Butail and D. A. Paley. 3D reconstruction of fish schooling kinematics from underwater video. In *Proc. 2010 IEEE Int. Conf. on Robotics and Automation*, 2010.

[24] C. H. Caicedo-Nùñez and N. E. Leonard. Symmetric coverage of dynamic mapping error for mobile sensor networks. In *Proc. 2011 American Control Conference*, 2011.

[25] M. Cao, A.S. Morse, and B.D.O. Anderson. Reaching a consensus in a dynamically changing environment: Convergence rates, measurement delays, and asynchronous events. *SIAM J. Control Optim.*, 47(2):601–623, 2008.

[26] F. Chang, C-J. Chen, and C-J. Lu. A linear-time component-labeling algorithm using contour tracing technique. *Computer Vision and Image Understanding*, 93(2):206–220, 2004.

[27] J. Cortès, S. Martìnez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. In *Proc. 2002 IEEE Conf. on Robotics and Automation*, 2002.

[28] I. D. Couzin, J. Krause, N. R. Franks, and S. A. Levin. Effective leadership and decision-making in animal groups on the move. *Nature*, 433(7025):513–516, 2005.

[29] I.D. Couzin, J. Krause, R. James, G. Ruxton, and N. Franks. Collective memory and spatial sorting in animal groups. *J. Theor. Biology*, (218):1–11, 2002.

[30] N. J. Cowan and E. S. Fortune. The critical role of locomotion mechanics in decoding sensory systems. *J. Neuroscience*, 27(5):1123–1128, January 2007.

[31] F. Cucker and S. Smale. Emergent behavior in flocks. *IEEE Trans. Automatic Control*, 52(5):852–862, May 2007.

[32] T. B. Curtin and J. G. Bellingham. Autonomous ocean-sampling networks. *IEEE J. Oceanic Engineering*, 26(4):421–423, 2001.

[33] T. B. Curtin, J. G. Bellingham, J. Catipovic, and D. Webb. Autonomous oceanographic sampling networks. *Oceanography*, 6(3):86–94, 1993.

[34] Adrian de Froment. Personal communication, 2009.

[35] M. A. Demetriou and I. I. Hussein. Estimation of spatially distributed processes using mobile spatially distributed sensor networks. *SIAM J. Control Optim.*, 48(1):266–291, 2009.

[36] L. DeVries and D. A. Paley. Multivehicle control in a strong flowfield with application to hurricane sampling. *J. Guidance, Control, and Dynamics*, 35(3):794–806, 2012.

[37] L. M. Dill, C. S. Holling, and L. H. Palmer. Predicting the three-dimensional structure of animal aggregations from functional considerations: The role of information. In J. K. Parrish and W. M. Hamner, editors, *Animal Groups in Three Dimensions*, pages 207–224. Cambridge University Press, 1997.

[38] I. D. Dinov. Expectation maximization and mixture modeling tutorial. UC Los Angeles: Statistics online computation resource. `http://www.escholarship.org/uc/item/1rb70972`, 2008.

[39] H. Edelsbrunner, D. G. Kirkpatrick, and R. Seidel. On the shape of a set of points in the plane. *IEEE Trans. Information Theory*, 29(4):551–559, 1983.

[40] U. Erra, B. Frola, V. Scarano, and I. D. Couzin. An efficient GPU implementation for large scale individual-based simualtion of collective behavior. *Int. Workshop on High Performance Computational Systems Biology*, pages 51–58, 2009.

[41] J. J. Faria, J. R. G. Dyer, R. O. Clèment, I. D. Couzin, N. Holt, A. J. W. Ward, D. Waters, and J. Krause. A novel method for investigating the collective behaviour of fish: Introducing 'Robofish'. *Behavioral Ecology and Sociobiology*, 64:1211–1218, 2010.

[42] The Robocup Federation. Robocup small size league. `http://www.robocup.org/robocup-soccer/small-size/`, 2012.

[43] A. Feldman, M. Hybinette, T. Balch, and R. Cavallaro. The multi-ICP tracker: An online algorithm for tracking multiple interacting targets. *Submitted.*, 2011.

[44] E. Fiorelli, N. E. Leonard, P. Bhatta, D. Paley, R. Bachmayer, and D. M. Fratantoni. Multi-AUV control and adaptive sampling in monterey bay. *IEEE J. Oceanic Engineering*, 31(4):935–948, 2006.

[45] B. P. Gerkey. C implementation of the Hungarian method. `http://robotics.stanford.edu/~gerkey/tools/hungarian.html`, 2008.

[46] C. Godsil and G. F. Royle. *Algebraic Graph Theory*. Springer, 2001.

[47] D. Grünbaum and A. Okubo. Modelling social animal aggregations. In S. A. Levin, editor, *Frontiers of Theoretical Biology: Lecture Notes in Biomathematics*, volume 100. Springer-Verlag, 1994.

[48] S. Gueron, S. A. Levin, and D. I. Rubenstein. The dynamics of herds: From individuals to aggregations. *J. Theor. Biology*, 182:85–98, 1996.

[49] J. Halloy, G. Sempo, G. Caprari, C. Rivault, M. Asadpour, F. Tache, I. Said, V. Durier, S. Canonge, J. M. Ame, C. Detrain, N. Correll, A. Martinoli, F. Mondada, R. Siegwart, and J. L. Deneubourg. Social integration of robots into groups of cockroaches to control self-organized choices. *Science*, 318:1155–1158, 2007.

[50] W. D. Hamilton. Geometry for the selfish herd. *J. Theor. Biology*, 31:295–311, 1971.

[51] M-K. Hu. Visual pattern recognition by moment invariants. *IRE Trans. Information Theory*, 8(2):179–187, 1962.

[52] M. Isard and J. MacCormick. BraMBLe: A Bayesian multi-blob tracker. In *Proc. IEEE Int. Conf. on Computer Vision*, pages 34–41, 2001.

[53] A. Jadbabaie, J. Lin, and A.S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans. Automatic Control*, 48(6):988–1001, 2003.

[54] S. J. Julier and J. K. Uhlmann. A new extension of the Kalman filter to nonlinear systems. In *Proc. of AeroSense: The 11th Int. Symp. on Aerospace/Defense Sensing, Simulation and Controls*, 1997.

[55] E. W. Justh and P. S. Krishnaprasad. Equilibria and steering laws for planar formations. *Systems and Control Letters*, 52(1):25–38, 2004.

[56] E.W. Justh and P.S. Krishnaprasad. A simple control law for UAV formation flying. Technical Report 2002-38, ISR, U. Maryland, 2002.

[57] Y. Katz, C.C. Ioannou, K. Tunstrom, C. Huepe, and I. D. Couzin. Inferring the structure and dynamics of interactions in schooling fish. *Proc. of the National Academy of Sciences*, 2011. In Press.

[58] Z. Khan, T. Balch, and F. Dellaert. An MCMC-based particle filter for tracking multiple interacting targets. In *Proc. European Conf. on Computer Vision*, pages 279–290, 2004.

[59] Y. Kim and M. Mesbahi. On maximizing the second smallest eigenvalue of a state-dependent graph Laplacian. *IEEE Trans. Automatic Control*, 51:116–120, 2006.

[60] Kitware, Inc. CMake. `http://cmake.org`.

[61] D. J. Klein and K. A. Morgansen. Controlled collective motion for trajectory tracking. In *Proc. 2006 American Control Conf.*, 2006.

[62] J. Krause, A. F. T. Winfield, and J.-L. Deneubourg. Interactive robotics in experimental biology. *Trends in Ecology and Evolution*, 26(7):369–375, 2011.

[63] Jens Krause and Graeme Ruxton. *Living in Groups*. Oxford University Press, Oxford, 2002.

[64] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.

[65] H. Kunz and C. K. Hemelrijk. Simulations of the social organization of large schools of fish whose perception is obstructed. *Appl. Anim. Behav. Sci.*, 138:142–151, 2012.

[66] Y. Kuramoto. *Chemical oscillations, waves, and turbulence.* Springer-Verlag, 1984.

[67] F. Lekien and N. E. Leonard. Nonuniform coverage and cartograms. *SIAM J. Control and Optimization*, 48(1):351–372, 2009.

[68] N. E. Leonard, B. Nabet, and D. Pais. Tensegrity models and vehicle formation shape control. *Proc. 2009 NSF Engineering Research and Innovation Conf.*, 2009.

[69] N. E. Leonard, D. Paley, F. Lekien, R. Sepulchre, D. M. Fratantoni, and R. Davis. Collective motion, sensor networks, and ocean sampling. *Proc. of the IEEE*, 95(1):48–74, 2007.

[70] N. E. Leonard, D. A. Paley, R. E. Davis, D. M. Fratantoni, F. Lekien, and F. Zhang. Coordinated control of an underwater glider fleet in an adaptive ocean sampling field experiment in monterey bay. *J. Field Robotics*, 27(6):718–740, 2010.

[71] N. E. Leonard, T. Shen, B. Nabet, L. Scardovi, I. D. Couzin, and S. A. Levin. Decision versus compromise for animal groups in motion. *Proceedings of the National Academy of Sciences*, 109(1):227–232, 2012.

[72] N. E. Leonard, G. F. Young, K. Hochgraf, D. Swain, A. Trippe, W. Chen, and S. Marshall. In the dance studio: Analysis of human flocking. In *Proc. 2012 of the American Control Conference*, 2012.

[73] K. Lynch, I. Schwartz, P. Yang, and R. Freeman. Decentralized environmental modeling by mobile sensor networks. *IEEE Trans. Robotics*, 24(3):710–724, 2008.

[74] D. Marthaler and A. L. Bertozzi. Tracking environmental level sets with autonomous vehicles. In S. Butekno, R. Murphey, and P. Pardalos, editors, *Recent Developments in Cooperative Control and Optimization.* Kluwer, Norwell, MA, 2003.

[75] R. Meester and R. Roy. *Continuum percolation.* Cambridge University Press, Cambridge, 1996.

[76] Merlin Systems Corp. MiaBot Pro user manual. Available from Merlin Robotics.

[77] Merlin Systems Corp. Merlin Robotics. `http://www.merlinsystemscorp.co.uk/index.php/e-shop/merlin-robotics.html`, 2011.

[78] A. Michelsen, B. B. Andersen, J. Storm, W. H. Kirchner, and M. Lindauer. How honeybees perceive communication dances, studied by means of a mechanical model. *Behavioral Ecology and Sociobiology*, 30(3–4):143–150, 1992.

[79] E.F. Morais, M. F.M. Campos, F.L.C. Padua, and R.L. Carceroni. Particle filter-based predictive tracking for robust fish counting. In *Proc. XVIII Brazilian Symp. on Computer Graphics and Image Processing*, 2005.

[80] L. Moreau. Stability of multiagent systems with time-dependent communication links. *IEEE Trans. Automatic Control*, 50(2):169–182, 2005.

[81] J. Munkres. Algorithms for the assignment and transportation problems. *J. Soc. Indust. Appl. Math.*, 5(1):32–38, March 1957.

[82] B. Nabet, N. E. Leonard, I. D. Couzin, and S. A. Levin. Dynamics of decision making in animal group motion. *Journal of Nonlinear Science*, 19(4):399–435, 2009.

[83] P. Ogren, E. Fiorelli, and N. E. Leonard. Coordinated control of an underwater glider fleet in an adaptive ocean sampling field experiment in monterey bay. *IEEE Trans. Automatic Control*, 49(8):1292–1302, 2004.

[84] A. Okubo. *Diffusion and Ecological Problems: Mathematical Models*, volume 10 of *Biomathematics.* Springer-Verlag, 1980.

[85] R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

[86] R. Olfati-Saber and R. M. Murray. Consensus protocols for networks of dynamic agents. In *Proc. 2003 American Control Conference*, 2003.

[87] D. Paley, N.E. Leonard, R. Sepulchre, and I.D. Couzin. Spatial models of bistability in biological collectives. In *Proc. 46th IEEE Conf. on Decision and Control*, 2007.

[88] D. A. Paley. *Cooperative Control of Collective Motion for Ocean Sampling with Autonomous Vehicles.* PhD thesis, Dept. of Mechanical and Aerospace Engineering, Princeton University, 2007.

[89] D. A. Paley. Cooperative control of an autonomous sampling network in an external flow field. In *Proc. 47th IEEE Conf. Decision and Control*, 2008.

[90] D. A. Paley. Stabilization of collective motion on a sphere. *Automatica*, 45(1):212–216, 2009.

[91] D. A. Paley, N. E. Leonard, R. Sepulchre, D. Grunbaum, and J. K. Parrish. Oscillator models and collective motion: Spatial patterns in the dynamics of engineered and biological networks. *IEEE Control Systems Magazine*, 27(4):89–105, August 2007.

[92] J. Parrish and P. Turchin. Individual decisions, traffic rules, and emergent pattern in schooling fish. In *Animal Groups in Three Dimensions*, pages 126–141. Cambridge University Press, 1997.

[93] J. K. Parrish and S. V. Viscido. Traffic rules of fish schools: A review of agent-based approaches. In C. K. Hemelrijk, editor, *Self-organisation and Evolution of Biological and Social Systems*, pages 50–80. Cambridge University Press, 2005.

[94] B. L. Partridge, T. Pitcher, J. M. Cullen, and J. Wilson. The three-dimensional structure of fish schools. *Behavioral Ecology and Sociobiology*, 6:277–288, 1980.

[95] B. L. Partridge and T. J. Pitcher. The sensory basis for fish schools: relative role of lateral line and vision. *J. Comp. Physiol.*, 135:315–325, 1980.

[96] M. Penrose. *Random Geometric Graphs.* Oxford University Press, New York, 2003.

[97] T. Pitcher. Sensory information and the organization of behaviour in a shoaling cyprinid fish. *Animal Behaviour*, 27(1):126–149, 1979.

[98] J. G. Proakis and D. G. Manolakis. *Digital Signal Processing.* Prentice Hall, 3^{rd} edition, 1996.

[99] J. Prokop and A. P. Reeves. A survey of moment-based techniques for unoccluded object representation and recognition. In *CVGIP: Graphical Models and Image Processing*, pages 438–460, 1992.

[100] J. Quintanilla, S. Torquato, and R.M. Ziff. Efficient measurement of the percolation threshold for fully penetrable discs. *J. Phys. A: Math. Gen.*, 33:L399–L407, 2000.

[101] C. W. Reynolds. Flocks, herds and schools: A distributed behavioral model. In *SIGGRAPH '87: Proc. of the 14th Ann. Conf. on Computer Graphics and Interactive Techniques*, pages 25–34, 1987.

[102] L. Scardovi, N. Leonard, and R. Sepulchre. Stabilization of three dimensional collective motion. *Communications in Information and Systems*, 8(4):473–500, 2008.

[103] L. Scardovi, A. Sarlette, and R. Sepulchre. Synchronization and balancing on the N-torus. *Systems and Control Letters*, 56(5):335–341, 2007.

[104] R. Sepulchre, D.A. Paley, and N.E. Leonard. Stabilization of planar collective motion: All-to-all communication. *IEEE Trans. Automatic Control*, 52(5):811–824, 2007.

[105] R. Sepulchre, D.A. Paley, and N.E. Leonard. Stabilization of planar collective motion with limited communication. *IEEE Trans. Automatic Control*, 53(3):706–719, 2008.

[106] SmoothOn, Inc. SmoothOn website. `http://www.smooth-on.com`, 2011.

[107] G. M. Spooner. Some observations in schooling in fish. *J. Mar. Biol. Ass. U.K.*, 17:421–448, 1931.

[108] A. R. Stewart. *Analysis and Prediction of Decision Making with Social Feedback*. PhD thesis, Princeton University, 2012.

[109] A. D. Straw, K. Branson, T. R. Neumann, and M. H. Dickinson. Multi-camera real-time three-dimensional tracking of multiple flying animals. *J. R. Soc. Interface*, July 2010.

[110] S. H. Strogatz. From Kuramoto to Crawford: exploring the onset of synchronization in populations of coupled oscillators. *Physica D*, 143:1–20, 2000.

[111] D. T. Swain. rhubarb: A lightweight and extensible IPC server. `http://github.com/dantswain/rhubarb`.

[112] D. T. Swain. MADTraC framework source documentation. `http://poincare.princeton.edu/MADTraC/docs/`, 2011.

[113] D. T. Swain, M. Cao, and N. E. Leonard. Effective sensing regions and connectivity of agents undergoing periodic relative motions. In *Proc. 47th IEEE Conf. on Decision and Control*, 2008.

[114] D. T. Swain, I. D. Couzin, and N. E. Leonard. Coordinated speed oscillations in *fundulus diaphanus* enrich information flow. *In Preparation*, 2012.

[115] D. T. Swain, I. D. Couzin, and N. E. Leonard. Real-time feedback-controlled robotic fish for behavioral experiments with fish schools. *Proc. of the IEEE*, 100(1):150–163, 2012.

[116] D. T. Swain and N. E. Leonard. On the trajectories and coordination of steered particles with time-periodic speed profiles. In *Proc. 2009 American Control Conference*, 2009.

[117] D.T. Swain, N.E. Leonard, I.D. Couzin, A. Kao, and R.J. Sepulchre. Alternating spatial patterns for coordinated group motion. In *Proc. 46th IEEE Conf. on Decision and Control*, December 2007.

[118] E. Trucco and A. Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, 1998.

[119] L. Tsimring, H. Levine, I. Aranson, E. Ben-Jacob, I. Cohen, O. Shochet, and W. N. Reynolds. Aggregation patterns in stressed bacteria. *Phys. Rev. Lett.*, 75:1859–1862, 1995.

[120] J. Tsitsiklis and D. Bertsekas. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Trans. Automatic Control*, 31(9):803–812, 1986.

[121] J. N. Tsitsiklis. *Problems in decentralized decision making and computation*. PhD thesis, Dept. of Electrical Engineering and Computer Science, Massachussetts Institute of Technology, 1984.

[122] J. C. van Olst and J. R. Hunter. Some aspects of the organization of fish schools. *J. Fisheries Research Board of Canada*, 27:1225–1238, 1970.

[123] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, O. Shochet, and A. Tenenbaum. Novel type of phase transition in a system of self-driven particles. *Phys. Rev. Lett.*, 75:1226–1229, 1995.

[124] S. V. Viscido, J. K. Parrish, and D. Grünbaum. Individual behavior and emergent properties of fish schools: A comparison of observation and theory. *Marine Ecology Progress Series*, 273:239–249, 2004.

[125] E. A. Wan and R. Van Der Merwe. The unscented Kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symp.*, pages 153–158, 2000.

[126] A. J. W. Ward, D. J. T. Sumpter, I. D. Couzin, P. J. B. Hart, and J. Krause. Quorum decision-making facilitates information transfer in fish shoals. *Proc. National Academy of Sciences*, 105(19):6948–6953, 2008.

[127] wxWidgets Team. wxWidgets cross-platform GUI library. `http://www.wxwidgets.org/`.

[128] F. Zhang, E. Fiorelli, and N. E. Leonard. Exploring scalar fields using multiple sensor platforms: Tracking level curves. In *Proc. 46th IEEE Conf. on Decision and Control*, 2007.

[129] F. Zhang and N. E. Leonard. Coordinated patterns of unit speed particles on a closed curve. *Systems and Control Letters*, 56:397–407, 2007.

[130] F. Zhang and N. E. Leonard. Cooperative Kalman filters for cooperative exploration. In *Proc. 2008 American Control Conf.*, 2008.

[131] F. Zhang and N. E. Leonard. Cooperative filters and control for cooperative exploration. *IEEE Trans. Automatic Control*, 55(3):650–663, 2010.