MOTION CONTROL STRATEGIES FOR NETWORKED ROBOT TEAMS IN ENVIRONMENTS WITH OBSTACLES

M. Ani Hsieh

A DISSERTATION

in

Mechanical Engineering and Applied Mechanics

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

2007

Vijay Kumar
Supervisor of Dissertation

Pedro Ponte Castañeda
Graduate Group Chairperson

To my mother.

Acknowledgements

First and foremost, I would like to express my most sincere gratitude and admiration to my advisor Vijay Kumar. Without his encouragement, support, and guidance, I would never have made the decision to come to GRASP and would have never completed this work.

My deepest thanks to the remainder of my dissertation committee, Professors C.J. Taylor, Ali Jadbabaie, and Claire Tomlin for their comments and suggestions. I especially want to express my gratitude to C.J. Taylor and George Pappas for supporting me in my various endeavors throughout my time at GRASP.

Special thanks to Luiz Chaimowicz, Adam Halasz, and Savvas Loizou for the numerous discussions, including topics that go beyond the scope of this work. Thanks to Jonathan Fink and Nathan Michael for all their help with the indoor experiments. Also, I would like to acknowledge everyone involved in the MARS2020 program and the Penn Athletics Department for making Chapter 3 possible.

Additionally, I would like to thank all GRASPees whom I have had the privelege to know and work with the last five years, especially Nora Ayanian, Dave Cappelleri, Jim Killer, Babak Shirmohammadi, and Peng Song.

My most sincere gratitute and appreciation to my family, especially my brother and mother, for whom all my accomplishments belong to.

Last, but always first in my heart, my deepest thanks to Anthony, for not only

being my most ardent supporter, but also my most honest critic. I can't wait to see where this journey takes us next.

ABSTRACT

MOTION CONTROL STRATEGIES FOR NETWORKED ROBOT TEAMS IN ENVIRONMENTS WITH OBSTACLES

M. Ani Hsieh

Vijay Kumar

Communication in multi-robot teams has, historically, been a means to improve control and perception. Recent advances in embedded processor technology have made it possible to equip every robot with inexpensive off–the–shelf wireless communication capabilities. These advances have also given robots the ability to monitor and respond to changes in the quality of their communication links. As such, progress in multi-agent robotics and sensor networks, and particularly the convergence of the two, will inevitably engender problems at the intersection of communication, control and perception.

While control is necessary for successful mission execution, reliable communication is essential for coordination and cooperation in multi–robot teams. For example, in applications such as perimeter surveillance or the cordoning off of hazardous regions, robots must be capable of forming complex shapes in the plane while maintaining the quality of the communication network. Thus, motion control strategies that do not require inter–agent communication can often be beneficial since they preserve limited bandwidth for the transmission of critical data. This is especially relevant in teams composed of large a number of small, resource constrained agents where bandwidth often becomes the limiting factor in agents' abilities to communicate.

Towards this end, this thesis considers scalable motion control strategies for networked robot teams that can be implemented with no inter–agent communication. Experimental studies of strategies for maintaining end-to-end communication links

for tasks like surveillance, reconnaissance, and target search and identification are discussed in the first part of the thesis. This then motivates the work presented in the second part: the synthesis of decentralized controllers for robot teams to form complex patterns in two dimensions. These decentralized controllers do not require the explicit communication of robots' state information. Rather, agents are assumed to be equipped with appropriate sensors, enabling them to infer relative position and bearing information of their neighbors. The stability and convergence properties of the controllers are presented, and the feasibility of the proposed methods is verified via computer simulations and experimental results using two multi-robot testbeds.

Contents

List of Figures

Chapter 1

Introduction

Nikola Tesla first proposed and demonstrated the principles of wireless communication to the National Electric and Light Association in 1893. He followed this with the exhibition of two radio remote–controlled boats in the Electrical Exhibition at New City's Old Madison Square Garden in 1898. Despite Tesla's faith in military applications of wireless communication, the technology remained a novelty until the advent of the television remote controller in the 1950s. Nevertheless, the foundation for all present day remote operated vehicles such as the Mars Rovers and Predator drones, and all multi–robot systems, was laid by Tesla almost half a century before Isaac Assimov introduced the word "*robotics*" to the modern day vernacular [94–97].

In addition to the television remote control, the 1950s also saw the births of modern robotics, and of the first multi–robot system, built by a biologist named W. Grey Walter. Walter's system consisted of two electronic, completely analog, autonomous robots with rudimentary navigation and obstacle avoidance capabilities. Much like Tesla's original conception of radio remote-controlled vehicles, these and other robots of the 1960s remained novelties until the late 1970s with the invention of the robotic manipulator. This event transformed the manufacturing industry and heralded the era of the automated assembly line.

Today, with the ever decreasing price to performance ratio of embedded processors and sensors, and increasingly ubiquitous wireless technology, automation has permeated every day life with robots poised to do the same. In 2004, industrial robots alone accounted for a $4 billion dollar market, with the personal service, entertainment, and domestic robot sector estimated at over $3.5 billion [9]. These advances that have made individual robots smaller, more capable, and less expensive have also enabled the development and deployment of teams of robotic agents.

While multi-robot systems may seem like a recent paradigm shift brought on by the latest technological advances, research in the field began almost at the conception of the robotic manipulator. In the early 1980s, the focus was primarily on the control and coordination of multiple robotic arms for cooperative grasping and handling of large objects [8]. Then in 1986, California established the Partners for Advanced Transit and Highways (PATH) program in an attempt to find novel ways to address the traffic congestion problem. The program generated much interest in the areas of automated highway systems and helped advance research in areas like autonomous vehicle platooning and formation maintenance [91]. However, the vision, and the motivation, for multi-robot systems were not clearly set forth until the 1988 publication of "Gnat Robots: A low-intelligence, low-cost approach" by Anita Flynn where she claimed:

> With low cost and small size, we can begin to envision *massive parallelism*, using millions of small simple robots to perform tasks that might otherwise be done with one large, expensive, complex robot. In addition, we also begin to view our robots as *disposable*. They can be cheap enough that they are thrown away when they have finished their mission or are broken, and we do not have to worry about retrieving them from hazardous or hard to reach places. [34]

Today, multi-robot systems have been deployed for tasks like environmental monitoring [3,48,83], surveillance of indoor environments [78] and support for first responders in search and rescue operations [52]. While there are many successful embodiments of multi-robot systems with numerous applications, they are still mostly found in the manufacturing industry where robots operate in structured environments executing fixed tasks with no variations in operating conditions. On the other hand, urban and unstructured envrionments provide unique challenges for the deployment of multi-robot teams. In these environments, buildings and large obstacles pose 3-D constraints on visibility, communication network performance is difficult to predict, and GPS measurements can be unreliable or even unavailable.

Consider the vision of the MARS2020 program funded by the Defense Advanced Research Projects Agency (DARPA). The collaborative effort between the General Robotics, Automation, Sensing & Perception (GRASP) Laboratory at the University of Pennsylvania, Georgia Tech Mobile Robot Laboratory and the University of Southern California's (USC) Robotic Embedded Systems Laboratory required the deployment of a heterogeneous team of aerial and ground autonomous robots with the vision of providing a framework that would enable deployment by a single human operator. Once deployed, the team of autonomous air and ground robots would cooperatively execute tasks such as surveillance, reconnaissance, and target search and localization within an urban environment while providing high-level situational awareness for the remote human operator. The framework would enable autonomous robots to synthesize the desirable features and capabilities of both deliberative and reactive control while incorporating a capability for learning, and include a software composition methodology that incorporates both pre-composed coding and learning-derived or automated coding software to increase the ability of autonomous robots to function in unpredictable environments. A team of heterogeneous robots with such capabilities has the potential to efficiently and accurately characterize the environment and to exceed the performance of human agents. As such, the objectives of the program were the development and demonstration of an architecture along with algorithms and software tools that:

- are independent of team composition;
- are independent of team size, i.e. number of robots;
- are able to execute a wide range of tasks;
- allow a single operator to command and control the team;
- allow for interactive interrogation and/or reassignment of any robot by the operator at the task or team level.

From our MARS2020 experience, such goals of coordinating multiple autonomous units, and making them cooperate inherently create problems at the intersection of communication, control and perception. While control is necessary for sucessful mission execution, reliable communication is essential for coordination and cooperation in multi-robot teams, particularly at the level required by the MARS2020 vision.

The prevalence of wireless technology has made it possible to inexpensively outfit every agent with some off–the–shelf wireless networking solution, however, the level of coordination required by the MARS2020 program would mean robots must also be able to perceive changes in their abilities to communicate, anticipate information needs of other network users, and reposition/self-organize themselves to best acquire and deliver the relevant information, and as such provide seamless situational awareness within various types of environments.

1.1 Problem Statement

With this in mind, the first part of the thesis is concerned with *strategies for maintaining end-to-end communication links in a multi-robot team.* Specifically, the coupling of high-level planning with low-level reactive controllers for communication link maintenance is presented along with experimental results considering the differences between monitoring inter-agent signal strengths versus data throughput. The second part of this thesis is concerned with the synthesis of *provably correct scalable motion control strategies for circular robots of finite size.* Specifically, we are interested in strategies that will enable a team of robots to form complex shapes in the plane while avoiding colllisions using little to no inter-robot communication. This is relevant because bandwidth is often the limiting factor in agents' abilities to transmit critical data in large teams. As such, robots must not only have the ability to form complex shapes, they must do so with as little communication overhead as possible to ensure reliable communication of crucial data with other team members or to a remote base station. Lastly, we extend our proposed motion coordination strategy to second order dynamic systems and incorporate some of the link maintenance strategies presented in the first part to synthesize *decentralized feedback controllers for pattern generation while maintaining team connectivity.*

In summary, the contributions of this thesis are two-fold: 1) experimentally verified strategies for maintaining point-to-point communication links in multi-robot teams and 2) provably correct methods for scalable motion synthesis for large teams of robots with obstacle avoidance. The motivations for these areas and a review of the relevant literature are provided in the following sections.

1.2 Maintaining Network Quality and Performance in Robot Teams

In recent years, the communication network has evolved from being just a medium of information transmission to an actual sensor, where properties like connectivity and signal strength are used to maintain the quality of the medium [7, 37, 73, 85]. Agents can use communication links to infer their individual locations with respect to those of their neighbors and other landmarks. Simultaneously, agents may also control their position and orientation relative to other agents to sustain communication links. We are interested in developing robotic teams that can operate autonomously in urban and/or hazardous areas and perform tasks such as surveillance, target search and identification, and reconnaissance all while maintaining team connectivity. These tasks are relevant in applications such as urban search and rescue, and environmental monitoring for homeland security, to name a few. We note that while the maintenance of network connectivity is required for useful situational awareness and system responsiveness, often the very environments we wish to operate in make this extremely challenging, especially when the mobile robots consist of small, lightweight ground vehicles that operate very close to the ground.

The growing interest in the convergence of the areas of multi-agent robotics and sensor networks has spurred interest in the development of networks of sensors and robots that can perceive their environment and respond to it. Much of the research in the mobile wireless network community has been devoted to the development of novel algorithms to handle packet routing [1, 99], resource allocation [31], and bandwidth management [61] for mobile nodes. However, control of mobile robot teams provides us with the capability to shape the team's communication needs based on continuous evaluation of the demands on the network [7].

One of the earliest works studying the effects of communication on multi-agent systems is the work by Dudek *et al.* where the effects of two-way, one-way, and completely implicit communication and sensing in a leader follower task was considered [30]. This, along with other works like the one by Winfield [98], and Arkin and Diaz [5], often assumed constant communication ranges and/or relied on line-of-sight maintenance for communication. Other examples include the works by Pereira [67] and Sweeney *et al.* [85], where decentralized controllers for concurrently moving toward goal destinations while maintaining relative distance and line-of-sight constraints were respectively presented; and the discussion of the fomration of communication relays between any pair of robots using line-of-sight was discussed by Anderson *et al.* [4]. Although coordination strategies that rely on line-of-sight maintenance may significantly improve each agent's ability to communicate, it has been shown through simulation by Thibodeau *et al.* [90], that line-of-sight maintenance strategies are often not necessary and may potentially be too restrictive. This work showed through simulations that coordination strategies based on line-of-sight maintenance for cooperative mapping are overall less efficient than strategies based on inter-agent wireless signal strengths.

Recent works that consider coordination strategies based on inter-agent signal strength include one by Wagner and Arkin where the combination of planning and reactive behaviors for was used for communication link maintenance in a multi-robot team conducting reconnaissance [93]. In this work, robots are tasked to go to different goal positions while maintaining communication links with the base station and/or a communication relay robot. In the event the robots sense a drop in the quality of their communication link(s), a contingency plan, *i.e.* a plan used to re-establish network connectivity, is triggered. In this case, the contingency plan re-tasked the robots to go to a location within the workspace selected *a priori*.

Simulation results were presented for teams of two to four robots. In general, goal positions are determined and planned based on all available information including radio transmission properties. However, most reasonably ambitious missions run the risk of encountering situations that were not reflected in planning. In the case of radio signal propagation in urban environments, one could rely on simulation validation of a plan, however this would require one to be extremely conservative in mission planning due to the difficulty in accurately predicting radio transmission characteristics.

Navigation based on perceived wireless signal strength between robots for exploration was presented by Sweeney *et al.* [86]. Here a null-space projection approach was used to navigate each robot towards its goal while maintaining point-to-point communication links. This work included simulation results for a team of four planar robots. Powers and Arkin [73] proposed a strategy where individual agents made control decisions based on their actual and predicted signal strength measurements while moving towards a goal. Simulation results for teams of one to four robots with and without the controller were presented. Although coordination strategies based on inter-agent signal strength can significantly improve overall performance, they do not account for the effects of team size on overall network performance. As team size increases, bandwidth becomes a limiting factor since an acceptable level of signal strength no longer guarantees a robot's ability to transmit critical data.

Figure 1.2(a) shows the number of transactions[1] per interval of time between four different robots, positioned at four distinct fixed locations, and a fifth stationary robot which we call the Base. Initially, one robot is transmitting at the maximum data rate supported by the network. As the second, third and fourth robots successively begin their transmissions to the Base, we see not only a drop in the bandwidth

[1]This metric is defined more precisely later in Section 3.2.2.

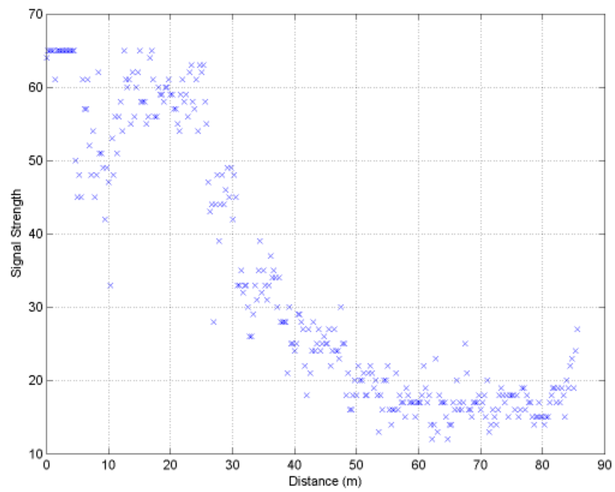
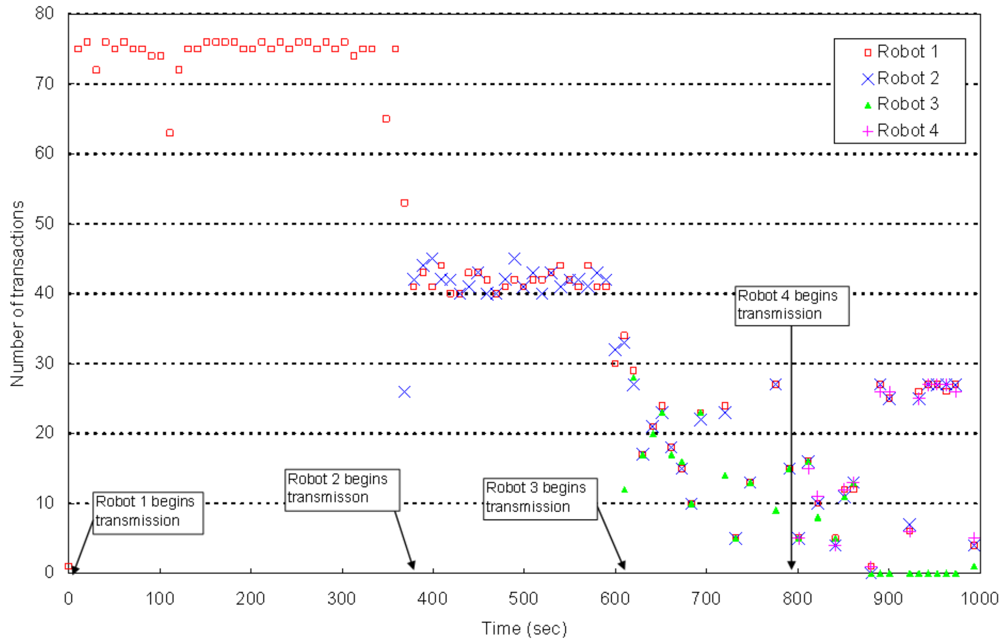


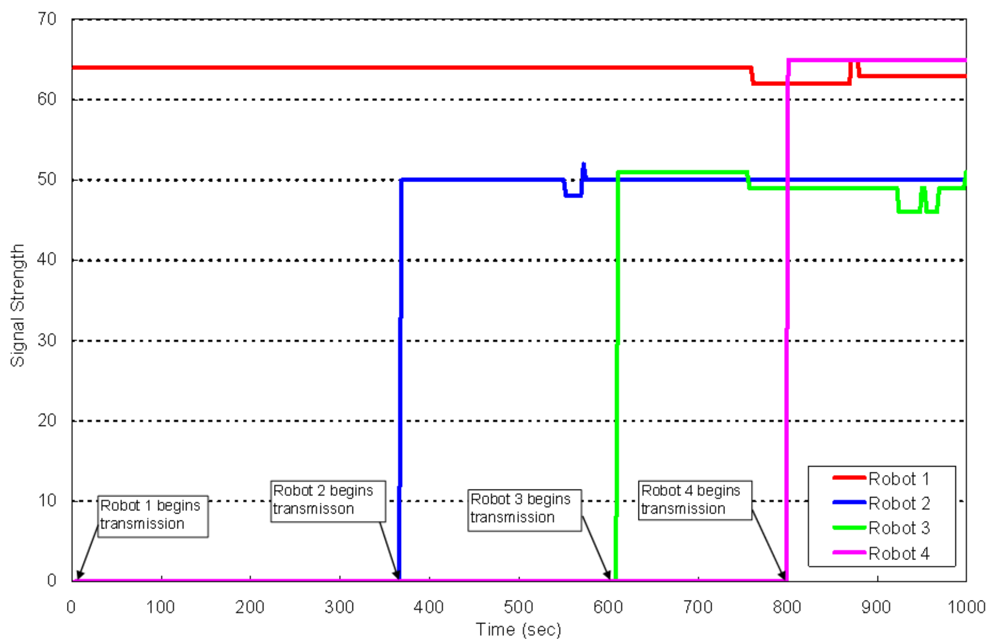
Figure 1.1: Signal strength measurements over various distances obtained in an environment similar to an urban park. Antennae were positioned 40 cm above the ground and the signal strength (y-axis) is normalized to a scale of 0 - 100.

available to each robot, but also a drop in total network throughput as significant network resources are spent coping with low-level packet collisions, retries and contention resolution. Situations such as this often occur in practice because a robot's sensing bandwidth typically exceeds the network bandwidth. It is important to note that during this time, the wireless signal strength measurements between the individual robots and the Base are virtually constant, as shown in Figure 1.2(b), since inter-robot distances were kept constant.

Additional works considering the impact of communication include the distributed optimization approach for cooperative motion planning while maintaining network connectivity proposed by Pimentel and Pereira [72]. Motion control algorithms for achieving biconnectivity in ad-hoc mobile networks was considered by Basu and Redi [7], while deployment strategies for achieving k-connectivity in sensor networks were proposed by Bredin*et al.* [12]. Additionally, Basu and Redi [6] considered flocking strategies for the placement of unmanned aerial vehicles for connectivity maintenance of ground networks. The effects of time-varying communication links



(a)



(b)

Figure 1.2: (a) Number of transactions per interval of time between four stationary robots, at four distinct locations, and the Base. The number of successful transactions between each robot and the Base drops as the number of transmitting robots increases over time. (b) Signal strength measurements from the robots to the Base for the same period of time.

on control performance of a mobile sensor node over a wireless network and in distributed sensing and target tracking was analyzed by Mostofi and Murray [58, 59]. The use of wireless communication for localization was presented by Howard [37] and for localization and navigation was presented by Corke *et al.* [20]. Deployment strategies for a mobile sensor network to control sensor node density were considered by Zhang and Sukhatme [100]. An exploration methodology for a multi-robot team to map the radio propagation characteristics of an urban environment was proposed by Hsieh *et al.* [41]. Lindhe *et al.* studied the effects of multi-path fading and developed a strategy of exploiting the fading for robot communications in [54].

In general, it is difficult to predict radio transmission properties *a priori* due to their sensitivity to a variety of factors including transmission power, terrain characteristics, and interference from other sources. Most existing propagation models assume transmission distances of approximately 100–200 meters with antennae placed high above the ground [62], and are not applicable to small, lightweight mobile nodes operating with low transmission power. This is because, at these small scales, the signal propagation mechanism is often dominated by the effects of reflection and scattering making modeling especially challenging in unexplored and unstructured environments.

In the first part of this work, we present techniques for ground vehicles connected via a wireless network to collaboratively perform surveillance tasks while providing situational awareness to an operator. We first show how nominal models of an urban environment can be used to generate strategies for exploration and present the construction of a radio signal strength map that can be used to plan multi-robot tasks and also serve as useful perceptual information. Additionally, we present reactive controllers for communication link maintenance. These controllers can be used in conjunction with information gleaned from our radio signal strength maps to enable our robots to adapt to changes in actual signal strength or estimated available bandwidth. We describe techniques to aid in planning robotic missions subject to connectivity constraints, and a reactive technology layer that maintains those constraints that may be composed with other controllers.

1.3 Formation Generation and Control for Robot Swarms

The advances in embedded processor and sensor technology that have made individual robots smaller, more capable, and less expensive have also enabled the development and deployment of teams of robotic agents, where capabilities are expressed by populations rather than super-capable individuals. As team sizes increase, it is often difficult, if not impossible, to efficiently manage or control the team through centralized algorithms or tele–operation. Accordingly, it makes sense to develop strategies where robots can be programmed with simple but identical behaviors that can be realized with limited on–board computational, communication and sensing resources.

In nature, the emergence of complex group level behaviors from simple agent level behaviors is often seen in the group dynamics of bee [13] and ant [74] colonies, bird flocks [24], and fish schools [66]. These systems generally consist of large numbers of organisms that individually lack either the communication or computational capabilities required for centralized control. As such when considering the deployment of large robot teams, it makes sense to consider such “swarming paradigms” where agents have the capability to operate asynchronously and can determine their trajectories based on local sensing and/or communication.

One of the earliest works to take inspiration from biological swarms for motion generation was presented by Reynolds in 1987 [76] where he proposed a method for generating visually satisfying computer animations of bird flocks, often referred as *boids*. Almost a decade later, Vicsek *et al.* showed through simulations that a team of autonomous agents moving in the plane with same speed but different headings converge to the same heading using nearest neighbor update rules [92]. The theoretical explanation for this observed phenomena was provided by Jadbabaie *et al.* [46] and Tanner *et al.* [87] extended these results to provide a detailed analysis of the stability and robustness of such flocking behaviors. Olfati-Saber then extended some of the results in these works to address the design and analysis of distributed flocking algorithms for robot teams in free–space and in environments with obstacles [64] . These works show that teams of autonomous agents can stably achieve concensus through local interactions alone, i.e. without centralized coordination, and have attracted much attention in the multi-robot community.

Previous works in group coordination using decentralized controllers to synthesize geometric patterns include the works by Albayrak *et al.* [2] and Suzuki *et al.* [84]. While Albayrak *et al.* only considered line and circle formations [2] , more general geometric patterns were considered by Suzuki *et al.* [84]. These aproaches, however, assume that each robot has full knowledge of the positions of all the other robots. Another approach to team formation is via a leader-follower formulation. One of the earliest synthesis of decentralized leader-follower controllers for robot formations was proposed by Desai *et al.* [25]. Although the controllers were decentralized, the methodology requires the assignment of different controllers and set points to different robots, making scaling to large groups difficult. Another decentralized leader-follower approach to formation control was presented by Fierro *et al.* [32]. Here, the authors established the asymptotic stability of the leader-follower formation control for a group of nonholonomic robots in $SE(2)$. Leader-follower controllers, in general, require the labeling of robots; Ogren *et al.* relaxed this assumption in the development of coordination strategies for a group of unidentified, holonomic robots [63].

Navigation function based approaches for multi-robot coordination include the works by Tanner *et al.* [88], Loizou *et al.* [55], and by Dimarogonas *et al.* [26]. While these works concern the motion coordination of non-point agents, they assume global sensing capabilities. This requirement was relaxed by Dimarogonas in [27] and extended to dynamic systems in [28], however, the resulting methodologies still require knowledge of team size which makes the addition and deletion of agents difficult. Other similar approaches for multi-robot manipulation are presented by Song and Kumar [81] and Pereira and Kumar [68] respectively. Chaimowicz *et al.* extended these approaches to arbitrary shapes and established convergence to patterns that approximate the desired shape [17]. With the exception of [55], the controllers in [17, 68, 81] enabled the team to converge to the boundary of some desired two-dimensional shape in the plane. The stability and convergence properties of these potential field based controllers with inter-agent constraints for a class of boundaries were analyzed by Hsieh and Kumar [40, 42]. Kalantar and Zimmer proposed a distributed switched controller to enable a swarm of underwater vehicles to uniformly disperse within a given region, with the outter robots aligning to the desired boundary using Fourier descriptors [49]. Pattern formation is achieved in [103] for a certain class of closed curves by determining each agent's distance with its neighbors and the desired contour. A similar problem was considered in [47], where formation control was formulated as a global energy minimization task over the entire collective.

Other approaches to detecting and tracking specific boundaries include the one by Bertozzi's group where the control laws are determined by solving a partial differential. This, however, requires the communication of each agent's position to its nearest neighbors [11]. A similar strategy is used by Pimenta *et al.* to construct potential functions for navigation in complex environments by large robot teams. In these works, robots are modeled as fluid particles and finite element methods are employed to obtain these navigation-like functions [69–71]. Sepulchre *et al.* derived control laws to stabilize a team of kinematic particles in the plane to isolated relative equilibria using certain types of communication interconnection topologies [79, 80]. Paley *et al.* extended these results to include elliptical and superelliptical orbits and relaxed the communication interconnection topology to undirected circulant graphs [65]. The stabilization of multiple agents to star-shaped orbits with relative arc-length constraints was presented by Zhang *et al.* [101,104] In these works, boundary coverage is achieved by maintaining inter-agent separation distances specified in terms of the arc-length of the boundary of interest rather than inter–agent Euclidean distances. Correll *et al.* experimentally compared three distributed algorithms for boundary coverage for a robotic swarm [23]. In addition, they modeled a robotic swarm as a collection of probabilistic finite state machines and presented a methodology for the system identification of both the linear and non-linear robotic swarm systems for parts inspection applications [22]. Although experimental and simulation results were shown in these works, they do not provide theoretical results for stability and convergence. Surveillance of an environment with obstacles was achieved by Kerr *et al.* by modeling individual robots in a swarm as gas particles [50]. While much of these works may have the capability of handling more complex environments, they model robots as individual point particles with unit mass and Euclidean dynamics, *i.e.* no second order effects, and as such are not realistic.

Belta *et al* presents a different approach to the shape generation/formation control problem [10]. Control abstractions for groups of planar robots were derived along with decentralized controllers such that motion planning for the group can be achieved in a lower dimensional space. They showed how groups of robots can be modeled as deformable ellipses, and presented decentralized controllers that allowed the control of the shape and position of the ellipses. The approach was extended by building a hierarchy of ground and air vehicles which allowed groups to split and merge [16]. Belta's method was also extended for robots in three dimensions [57]. Formations for small teams of robots can also be achieved by modeling the team as controlled Lagrangian systems on Jacobi shape space [102]. More recently, the problem of positioning a team of robots to generate different shapes in two and three dimensions was formulated as a second-order cone program [82]. Lastly, a coordination strategy that stabilizes a group of vehicles to an arbitrary desired group shape derived from spatial networks of interconnected struts and cables, i.e. tensegrity structures, was presented by Nabet *et al.* [60].

The second part of this works builds on the results of Chaimowicz [17] and Hsieh [40,45], and, in the spirit of the works by Belta, Chaimowicz, and Michael [10,16,57], we address the synthesis of decentralized controllers that guarantee the convergence of the team to the boundary of some desired shape as well as the stability of the resulting formation, all while avoiding collisions and/or maintaining inter–agent constraints through local interactions. While our approach is similar to the works of Zhang *et al.* [101,104], we take a slightly different approach to the pattern generation problem and consider inter–agent constraints (collision avoidance and/or proximity maintenance) that are functions of Euclidean distances between agents rather than arc–lengths. Furthermore, our approach enables us to control swarms of circular robots with finite size while ensuring collision avoidance, and can be implemented via sensing alone with no inter–agent communication. This may be relevant in applications like persistent surveillance where limited bandwidth must be preserved to enable robots to communicate with each other in order to integrate and fuse the information acquired by various sensors.

1.4 Organization of this work

The remainder of this thesis is organized as follows. Chapter 2 is a discussion of the modeling approaches taken in this work for the communication medium, the robotic agents, and the controllers. Chapter 3 begins with the first major contribution of this work: experimentally verified strategies for maintaining communication links in a multi-robot team. Here, an exploration strategy for a team of mobile robots to collect information to populate a radio signal strength map of an urban environment is discussed, along with low-level reactive controllers for communication link maintenance. Then, Chapter 4 presents motion synthesis for large teams of robots with obstacle avoidance, the second contribution of this work. We present the synthesis of decentralized controllers that can drive a swarm of robots to some desired boundary curve while achieving collision and obstacle avoidance. The stability and convergence properties of these controllers are also analyzed and discussed. Simulation and experimental results using our indoor multi–robot testbed are also presented. Finally, Chapter 5 extends some of the methodology presented in Chapters 3 and 4 to second order dynamic systems. Specifically, this chapter covers the synthesis and analysis of decentralized controllers for pattern formation with inter–agent constraints. This work concludes with a brief summary of the major contributions of this work and a discussion on directions for future work in Chapter 6.

Chapter 2

Modeling

In this chapter, we provide a brief overview of the various communication, robot, and controller models that are employed and discussed throughout this work.

2.1 Robots

2.1.1 Theoretical Models

This work is concerned with the coordination of large teams of robots, specifically for applications such as perimeter surveillance, environmental boundary monitoring, and/or surrounding hazardous regions while satisfying inter-robot constraints such as collision avoidance and/or maintaining communication links. In general, we assume a team of N planar, fully actuated robots that can be modeled by the following kinematics,

$$\dot{q}_i = u_i \tag{2.1}$$

where $q_i = (x_i, y_i)^T$ and u_i denote the i^{th} agent's position and control input respectively. The robot state is a 2×1 state vector given by q_i and the state of the team of robots is given by $\mathbf{q} = \left[q_1^T \ldots q_N^T\right]^T \in \mathbf{Q} \subset \mathbb{R}^{2N}$. In this situation, the state space is equivalent to the configuration space.

In situations where the velocity response of a robot is significantly slower than its position response, we model each robot with the following dynamics,

$$\dot{q}_i = v_i \tag{2.2a}$$

$$\dot{v}_i = u_i. \tag{2.2b}$$

Similarly, q_i, v_i and u_i respectively denote the i^{th} agent's position, velocity, and control input. Here, the configuration space, Q, is defined as $Q \subset \mathbb{R}^{2N}$, and the configuration of the team of robots is given by $\mathbf{q} = \left[q_1^T \ldots q_N^T\right]^T \in Q$. The robot state is a 4×1 state vector given by $\mathbf{x}_i = \left[q_i^T v_i^T\right]^T$, with the state vector of the team given by $\mathbf{x} = \left[\mathbf{x}_1^T \ldots \mathbf{x}_N^T\right]^T \in \mathbf{X} \subset \mathbb{R}^{4N}$. In general, we consider all systems of N agents whose states can be diffeomorphically transformed into (2.1) or (2.2).

2.1.2 Hardware

The experiments presented in this work were conducted using two experimental testbeds. The first experimental testbed consists of five outdoor ground vehicles modified from commercially available, radio controlled scale model trucks used in the MARS2020 program. Each vehicle's chassis is approximately 480 mm long and 350 mm high. Mounted in the center of the chassis is a Pentium III laptop computer. Each vehicle contains a specially designed Universal Serial Bus (USB) device which controls drive motors, odometry, steering servos and a camera pan mount with input from the PC. A GPS receiver is mounted on the top of an antenna tower, and an inertial measurement unit (IMU) is mounted between the rear wheels. A

Figure 2.1: The University of Pennsylvania's MARS2020 multi-robot testbed. [14]

forward-looking stereo camera pair is mounted on a pan mount which can pivot 180 degrees to look left and right. A small embedded computer with 802.11b wireless Ethernet, called the Junction Box (JBox), and an omnidirectional antenna are used to handle wireless communication. The JBox, jointly developed by the Space and Naval Warfare Systems Center, BBN Technologies, and the GRASP Laboratory is used to handle multi-hop routing in an ad-hoc wireless network and provide signal strength measurements for all nodes on the network. The omnidirectional antenna is mounted approximately 40 cm off the ground. Lastly, these robots are designed to travel at a fixed speed of $1m/s$. A picture of the multi-robot team is shown in Figure 2.1.

These vehicles cannot be described by the simple kinematic and dynamic models given by (2.1) and (2.2). However, since each robot is equipped with GPS, the control input for each agent is specified in terms of absolute position rather than velocities or accelerations. Thus, for every goal position, we compute a reference heading which is then used to generate a look-ahead waypoint. Then, fixing the vehicle's speed, a proportional and integral (PI) controller is used to control the robot's heading to steer the robot towards the look-ahead waypoint. This is repeated until the goal

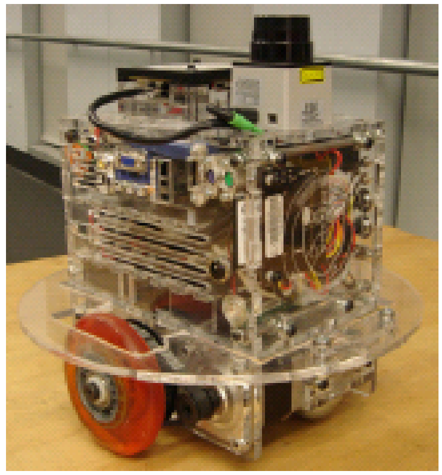


Figure 2.2: The SCARAB multi-robot testbed.

position is within a given error tolerance.

The second experimental testbed consists of four indoor, ground platforms developed by at the GRASP Laboratory and referred to as the SCARABs [33, 56]. These robots are approximately $20 \times 13.5 \times 22.2$ cm^3 in volume with a diameter of 30 cm. These robots are differential drive robots with a wheel base of 21 cm and each robot possesses two stepper motors that drive 10 cm rubber wheels and have a nominal holding torque of 28.2 kg-cm at the axle. Each SCARAB is equipped with an on-board embedded computer that has a 1 GHz processor and 1 GB of RAM and and supports IEEE 1394 firewire and 802.11a/b/g wireless communication. Each unit weighs approximately 8 kg and is equipped with a Hokuyo URG laser range finder, a motor controller that provides odometry information from the stepper motors, and a power management board. Lastly, each robot is able to support up to two firewire cameras. A picture of a SCARAB is shown in Figure 2.2.

While these robots are non-holonomic, it is possible to model them as holonomic robots with kinematics of the form given by (2.1). This is achieved by considering the coordinates of a reference point, p, on the robot which is offset from the axle by a distance of l and by translating the velocities of the reference point to commanded linear and angular velocities, v and ω respectively, for the robot through the following equations:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \cos\theta & -l\sin\theta \\ \sin\theta & l\cos\theta \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}.$$

Given a reference point p and a radius for a circle circumscribing the robot, which we denote by r, this methodology ensures that all points on the robot lie within a circle of radius $l + r$ centered at p, enabling us to treat these vehicles as holonomic circular robots with finite radius $l + r$.

2.2 Communications

Radio connectivity is often difficult to predict since it relies on complex propagation mechanisms as well as other factors like the distance between the transmitter and receiver, the three-dimensional geometry of the environment, and tranmission power. As such, the received signal is often the sum of the various components resulting from diffraction, scattering, reflection, transmission, and refraction of the transmitted signal. Therefore, accurate prediction models must take into account the diverse attenuation effects stemming from all these physical phenomena.

In the last few decades, the increase in demand of good quality and cost effective radio systems has further emphasized the need of accurate models for predicting radio propagation losses [62]. There are numerous prediction methods and models in the literature today. In general, these propagation models can be categorized as empirical, theoretical, or a combination of both. Empirical models implicitly take into account all the environmental influences regardless of whether these can be easily distinguished from one another. With enough data and tuning of the various parameters, empirical models can often be quite accurate for a given environment. The downside of these models, however, is that their quality is often dependent on the accuracy of the measurements as well as the similarities between the environment of interest and the environment where the measurements were made. Theoretical models, on the other hand, are concerned with the physics of radio wave propagation and are thus applicable to various different environments. However, implementations of these models often require large databases of environmental characteristics which can be very difficult to obtain. Additionally, given the large amount of data that must be kept handy at all times, these models are often computationally inefficient due to their complexity, which limits their use to mostly small outdoor environments.

While numerous models exist, they are often inadequate when applied to mobile robot teams and/or sensor networks. This is because most existing models are either a *macrocell* or a *microcell* propagation model. Macrocell propagation models deal with systems whose transmission powers are in the tens of Watts and propagation distances in the tens of kilometers. Under these circumstances, it is nearly impossible to ensure line-of-sight between transmitters and receivers, and as such, the signal propagates primarily by diffraction and reflection. Additionally, at these scales, refraction from the atmosphere results in a significant contribution to propagation loss. Microcell propagation models, on the other hand, deal with much smaller outdoor regions, *e.g.* the area around a street corner. These models are generally concerned with tranmission powers that are mostly in the milliWatt range, transmission distances ranging between 200 m to 1000 m, and base station antennae placed approximately 3 m to 10 m above the ground. In these models, the surrounding geometry plays a significant role when determining propagation losses.

Our work is primarily concerned with the deployment of teams of autonomous robots that operate in urban environments. In these situations, radio paths are relatively short in distance, roughly on the order of 10s or 100s of meters, with low transmission powers. While microcell models provide an approximation, they are often inadequate since these models generally assume fixed repeaters with antennae approximately one to two stories above the ground, similar to mobile cell phone networks and/or wireless data networks. On the other hand, the autonomous ground vehicles under consideration are generally small, light-weight vehicles, often with antennae no more than one to two feet off the ground. Under these circumstances, ground effects such as reflection and scattering become the dominant propagation mechanisms. As such, few, if any, existing propagation models actually match the actual operating conditions of the robots in the field.

Despite these issues, the most common model used in the mobile robot literature [54, 86] to determine the *signal strength*, i.e. the measure of the strength of a transmitted signal as it is received, is based on the following *path loss* or *path attenuation* model:

$$PL = -20 \log d^a \cdot (1 + d/g)^b + c$$

where PL denotes the path loss in $dB\mu V/m$, d denotes the distance to the transmitting antenna, a is the basic attenuation rate for short distances often referred to as the *path loss exponent*, b is the additional attenuation rate coefficient beyond the *breakpoint*, g is the distance to the breakpoint, and c is a scaleable factor [37, 85, 90]. The path loss exponent typically ranges between 2, e.g. free space propagation, to 4, e.g. tunnel-like evironments. The breakpoint is the maximum distance before the effects of diffraction becomes dominant with $b \approx 4$.

There are many variations of the above model depending on the operating conditions. For example, at distances less than the breakpoint, the model can be simplified to

$$PL = -20 \log d^a + c,$$

while at distances greater than the breakpoint, the model becomes

$$PL = -20 \log d^{(a+b)} + c.$$

In this work, outdoor experiments were conducted using a team of modified radio controlled trucks equipped with omni-directional antennae that are no more than two feet off the ground. Under these circumstances, the path loss exponent is closer to 4. The interested reader is referred to [62] for further details.

2.3 Controllers

This work addresses the synthesis of controllers that can position a team of robots along a desired boundary curve of a given shape while maintaining constraints with other agents. In general, we assume each robot has the ability to localize itself within some desired precision. Additionally, we assume each robot possesses a map of the environment in which they are operating in, and, as such, potential function based navigation strategies can be employed for navigation.

We rely on potential function based strategies to address the problem of coordinating large numbers of robots. The main advantage of potential based methods is that they allow us to simultaneously solve the path planning and control problems simultaneously. In practice, however, these approaches require robots to have precise localization capabilities to ensure continuity of the feedback policy. In this work, we assume robots have either well-defined safe zones or repulsion zones to ensure collision avoidance. As such, given the limitations of the hardware, it is possible for us to limit computational errors in our potential function strategies resulting from inaccurate estimation of robots' positions by appropriately selecting these safety zones.

Since our focus is on the coordination of large teams of robots, as such, we are concerned with feedback controllers that are decentralized, simple to implement, and identical across all agents, thus enabling scalability as the size of the team increases. While each robot is equipped with a wireless communication module, we assume no state information is exchanged among the agents. Rather, the positions of each robot's neighbors are inferred using range and bearing sensors. In the event a robot must infer a neighbor's velocity, we assume the robot is capable of doing so from its existing sensor suite, *i.e.* filtering position history information for each neighbor, rather than assume the presence of communication. Lastly, in certain special cases, we will assume each robot has the ability to detect failed agents within its sensing vicinity. Depending on the severity of the failure, in practice, this can be achieved via minimal communication between the failed robot and the live ones, or through the use of a bump sensor, and/or some high-level reasoning capabilities.

Chapter 3

Maintaining Network Connectivity and Performance in Robot Teams

We consider the problem of a team of robots operating in an urban, potentially hazardous, environment for tasks such as reconnaissance and perimeter surveillance, where maintaining team connectivity is essential for situational awareness. In these tasks, robots must have the ability to align themselves along the boundaries of complex shapes in two dimensions while ensuring the successful transmission of critical data. Importantly, navigation based solely on the geometry of the environment will not always guarantee a connected communication network. In these situations, a rough model of the radio signal propagation encoded in a radio connectivity map, *i.e.* a map that gives the average signal strength measurements from one position in the workspace to any other position, becomes extremely helpful in the planning phase. Furthermore, since real-world environments are often very complex and dynamic, it is important for robots to also have the ability to respond to real-time changes in link quality to ensure network connectivity.

Figure 1.1 shows actual signal strength measurements obtained in an environment representative of an urban park using two nodes at different separation distances. Although, there is a strong correlation between signal strength and distance, there is also a lot of variability due to the various factors mentioned earlier [62]. These kinds of information cannot always be accurately inferred from a radio connectivity map. Thus, successful mission execution will require both proper deliberative planning and suitably designed reactive behaviors to facilitate the operation of the team with little to no direct human supervision.

Most prior works in the area of communication link maintenance leave the burden of performance specification to fixed metrics, typically based on the distance between nodes or on simulated signal strength. However, as mentioned earlier, radio signal propagation depends on a variety of factors that are often difficult to capture in simulation alone. Rather than rely on simulation, our approach entails the use of radio connectivity maps for planning as well as low level reactive controllers that respond to changes in actual signal strength or verified network bandwidth. The goal is to develop strategies that exploit information gathered during an initial exploration phase coupled with well-designed reactive behaviors that remain minimally disruptive to any high level deliberative plans in order to maximize the team's ability to provide effective situational awareness to a base station. In essence, our strategies are based on metrics that do not rely on assumptions that may not be transferable or realistic in the physical workspace that the team is operating within.

This chapter presents experimentally verified strategies for maintenance of point-to-point communication links in multi-robot teams. Section 3.1 begins with the presentation of a methodology to enable a team of robots to obtain signal strength information used to populate a radio signal strength map for an urban environment. Next, Section 3.2 presents low-level reactive controllers that can be used to constrain the motion of individual agents based on two link quality measures: signal strength and perceived network bandwidth. We present two sets of experimental results using these controllers in outdoor environments under different network interconnection topologies. In the first set of experiments, the radio connectivity map was used to determine a deployment strategy for a reconnaissance task. In the second set of experiments, we deployed our multi-robot team to execute a perimeter surveillance task. The reactive controllers are designed to be minimally disruptive to the overall deliberative plan, and provide situational awareness to a base station including notification regarding potential failure points in the communication network.

3.1 Multi-robot Radio Mapping

This section describes a methodology to generate a deployment strategy which enables a homogeneous team of mobile-robots to obtain a *radio connectivity map* for an urban environment. Specifically, the methodology is described for two special cases: i) when the team consists of two robots, and ii) when the team consists of three robots. Experimental results obtained using a three-robot team are presented.

3.1.1 Modeling

For any given environment, denote the configuration space as $\mathcal{C} \subset \mathbb{R}^2$, and the obstacle free portion of $\mathcal{C}$ as $\mathcal{C}_f$, also referred to as the free space. Given any two positions $q_i, q_j \in \mathcal{C}_f$, define the *radio connectivity map* as a function $\varphi : (q_i, q_j) \to \mathbb{R}$ such that φ returns the average radio signal strength between the two positions given by q_i and q_j. In general, it is extremely difficult to obtain a connectivity map for all pairs of positions in $\mathcal{C}_f$. Rather, assume a convex cell decomposition can be performed for any given $\mathcal{C}_f$ and denote the centroid of the convex cell as q_i where i denotes the i^{th} cell. Therefore, the objective is to construct a map for pairs of locations in the set $Q = \{q_1, \ldots, q_{n_1}\}$ where Q is a subset of $\mathcal{C}_f$.

While it is reasonable to assume a convex decomposition exists for any given $\mathcal{C}_f$, it does not necessarily mean that the signal strength for all pairs of positions in any two cells will be the same. However, since each cell is convex, any two positions within the same cell will have line-of-sight, thus making it possible to predict the signal strength for the pair given prior knowledge of the variation of radio signal transmission characteristics with distance. Since the objective is to develop a methodology for the construction of the radio connectivity map rather than determining the appropriate cell decomposition, it will be assumed that the decomposition is given.

Additionally, assume a connected roadmap can be constructed from the given cell decomposition of $\mathcal{C}_f$. The roadmap is represented as an undirected graph, given by $G_1 = (V_1, E_1)$, where each cell is associated with a node in V_1 and every edge in the set E_1 represents the existence of a feasible path between neighboring cells. Given,

$$V_1 = \{v_1^1, \ldots, v_1^{n_1}\} \quad \text{and} \quad E_1 = \{e_1^1, \ldots, e_1^{m_1}\},$$

denote the total number of nodes and edges in G_1 as n_1 and m_1 respectively. Thus, for every $q_i \in Q$, there is a corresponding $v_1^i \in V_1$. The adjacency matrix for G_1, denoted as A_1, is defined as

$$A_1 = [a_{ij}] = \begin{cases} 1 & \text{if path exists between } v_1^i \text{ and } v_1^j \\ 0 & \text{otherwise} \end{cases}$$

The graph G_1 is called the *roadmap graph.* Since the team consists of homogeneous robots, the same G_1 applies to every member of the team.

Next, define the *radiomap graph*, $R = (V_1, L_1)$, such that R is used to encode the signal strength information one would like to gather. The edge set L_1 represents signal strength measurements that must be obtained for pairs of nodes and is selected *a priori* based on the task objectives, the physical environment, prior knowledge of radio signal transmission characteristics, and may include all possible edges in G_1. The adjacency matrix A_R for the radiomap graph, R, is then given by

$$A_R = [a_{R_{ij}}] = \begin{cases} 1 & \text{if signal strength between } v_1^i \\ & \text{and } v_1^j \text{ is to be measured} \\ 0 & \text{otherwise} \end{cases}$$

Given the roadmap and radiomap graphs, G_1 and R, denote the *multi-robot exploration graph* as $G_k = (V_k, E_k)$, where k denotes the number of robots in the team. The graph G_k is then constructed such that obtaining an optimal deployment strategy for measuring the edges in L_1 is equivalent to solving for a shortest path on the graph G_k. The methodology is presented in the following section with special attention paid to the two and three robot cases.

3.1.2 Methodology

Given the roadmap, $G_1 = (V_1, E_1)$, and k robots, a *configuration* on G_1 is an assignment of the k robots to k nodes on the graph. Figure 3.1(b) shows some possible configurations of three robots on the roadmap graph G_1, shown in Figure 3.1(a). In these figures, cells are represented by empty circles and the edges denoting feasible paths between neighboring cells are represented by solid lines. In Figure 3.1(b), robot locations are denoted by solid circles. Since the graph G_1 is connected, a path always exists for k robots to move from one configuration to another. For certain

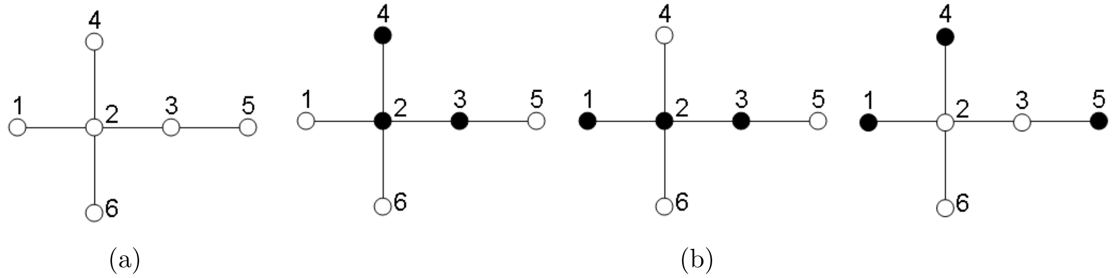


Figure 3.1: (a) Roadmap graph G_1. (b) Three different configurations that three robots can take on the graph G_1.

configurations of k robots on G_1, the complete, *i.e.* fully connected, graph whose vertices are given by the locations of the robots, contains some of the edges in L_1. Figure 3.2(b) shows two three robot configurations on G_1 that can measure edges in L_1 – the edge set of the radiomap graph shown in Figure 3.2(a). In these figures, the edges in L_1 are denoted by the dashed lines. Note the set of edges for the complete graph generated by the robots, denoted by the solid circles and lines, consists of some edges in L_1. And as such, the fully connected graph generated by the k robots may have more edges than the subgraph of R_1 consisting of the same vertices. Thus, a plan or exploration strategy to measure all edges in the set L_1 can be viewed as a sequence of robot configurations such that every edge in L_1 is measured by at least one of these configurations.

In general, given the graphs G_1 and R and k robots, the multi-robot exploration graph, G_k, is constructed such that every node in V_k denotes a k-robot configuration on G_1 that can measure some edges in the set L_1. An edge, $e_k^{ij} \in E_k$, exists between any two nodes $v_k^i, v_k^j \in V_k$ if the configuration associated with v_k^i is reachable from the configuration associated with v_k^j. Since G_1 is always connected, k robots can always move from one configuration to another, therefore, G_k is always a complete graph, *i.e.* a graph where every node is adjacent to every other node. Every edge in E_k is then assigned a weight that represents the total cost to move the robots from

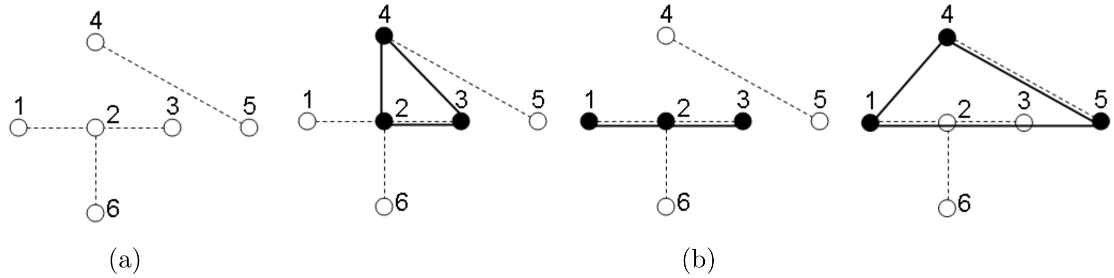


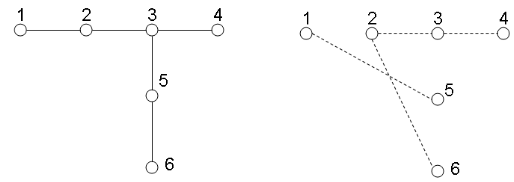
Figure 3.2: (a) Radiomap graph, R, for G_1 shown in Figure 3.1(a). The dashed edges denote links for which signal strength information must be obtained. (b) Three sample configurations of three robots on G_1 that can measure at least one of the edges in R. The solid vertices denote the robots and the solid edges denote the edges that can be measured for the given configuration.

one configuration to another. For the example shown in Figure 3.2, let the cost to move a robot from one node to another be given by the minimum number of edges on the path connecting the two nodes. Thus, for the configuration given by the nodes $\{2, 3, 4\}$, the minimum cost to move to the configuration given by nodes $\{1, 2, 3\}$ is 2.

Given the multi-robot exploration graph, an optimal plan/exploration strategy would simply consist of a sequence of configurations, such that the movement through all configurations in the sequence results in covering all edges in L_1 while minimizing the total cost. Since G_k encodes all necessary information needed to determine an exploration strategy for the k–robot team, finding an optimal plan is equivalent to solving for a minimum cost path on G_k that results in covering every edge in L_1. The methodology for obtaining G_k for $k = 2$ and $k = 3$ is further described in the following sections.

Two Robot Problem

Given the roadmap and radiomap graphs $G_1 = (V_1, E_1)$ and $R = (V_1, L_1)$ and two robots, the maximum number of links that can be measured for any configuration

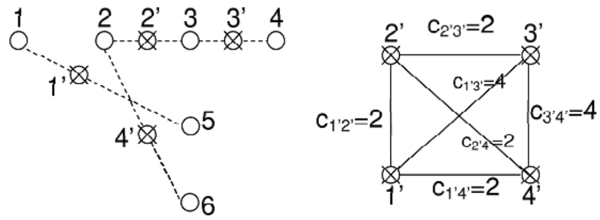


(a) (b)

Figure 3.3: (a) A sample roadmap graph G_1. (b) Corresponding radiomap graph R.

is one. For the two robot case, the radio exploration graph $G_2 = (V_2, E_2)$ can be constructed such that each node in G_2 corresponds to one edge in the set L_1. For example, given the roadmap and radiomap graphs shown in Figure 3.3, Figure 3.4(a) shows the mapping of every edge in L_1 to a node in G_2. For simplicity, let the cost to move from one 2 robot configuration to another be defined by the minimum number of edges traversed by the team and weight every edge in G_2 with the associated cost. The weights shown in Figure 3.4(b) were obtained by first determining the minimum number of edges between every pair of nodes in G_1. These were then used to determine the minimum number of edges required to enable the team to move between every pair of nodes in G_2. For example, the cost to move from the configuration $\{2, 6\}$ to $\{1, 5\}$, denoted by nodes $4'$ and $1'$ respectively in Figure 3.4(b), is equal to 2 and therefore edge $e_2^{4'1'}$ has a weight of 2. From this example, the optimal plan for a start configuration given by node $1'$ is the path $\{1', 4', 2', 3'\}$ with a total cost of 6 edges. In the two robot case, an optimal deployment strategy will necessitate the team to visit every node in G_2 once. This is equivalent to solving the traveling salesman problem on the graph G_2.

Algorithm 1 describes the method used to obtain the optimal strategy for the 2-robot case. To determine the weight of every edge in E_2, we compute the shortest path between every pair of nodes in G_1. The adjacency and cost matrices for G_2 are



(a) (b)

Figure 3.4: (a) Graph R superimposed with G_2 nodes denoted by $\otimes$. (b) G_2 for the G_1 and R shown in Figure 3.3.

obtained by considering the set of allowable moves given by G_1 and the set of edges given by R. Once the adjacency and cost matrices for G_2 have been determined, the optimal strategy is obtained by solving an open path traveling salesman problem on G_2. Although the Traveling Salesman Problem is known to be NP-hard, there are known approximation algorithms that solve for the minimum cost path in polynomial time [19]. For small graphs the problem can be solved using branch and bound techniques [21].

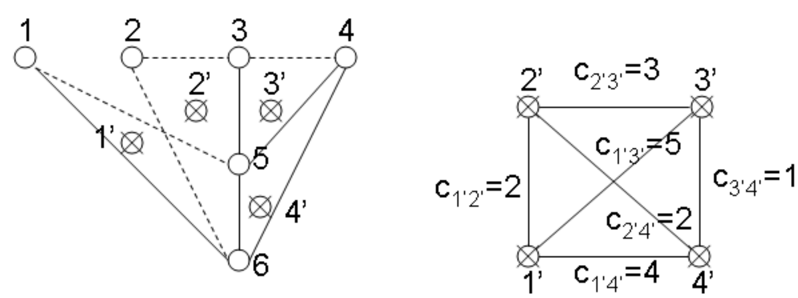
Three Robot Problem

Given the roadmap and radiomap graphs, G_1 and R, the set of nodes in V_3 is obtained by considering all 3-robot configurations on the graph G_1 that contain at least one edge in L_1. The algorithm to obtain the vertex set V_3 is outlined in Algorithm 2. For the roadmap and radiomap graphs given in Figure 3.3, some of these configurations are shown in Figure 3.5(a). Here the configurations given by nodes $\{1,5,6\}, \{2,3,6\}, \{3,4,5\}$, and $\{3,4,6\}$ would correspond to nodes $1'$, $2'$, $3'$, and $4'$ on G_3 respectively. Figure 3.5(b) is a subgraph of G_3 with the nodes associated with the configurations shown in Figure 3.5(a) as its vertices.

Similar to the two robot case, shortest path computation between every node in

Algorithm 1 Computation of the optimal plan for 2-robots

Construction of the vertex set V_2
Given G_1, A_1 and R, A_R
$V_2 = 0$
for each node $v_1^1, \ldots, v_1^{n_1}$ **do**
  for each node $v_1^1, \ldots, v_1^{n_1}$ **do**
    if $A_R(i,j) = 1$ **then**
      $V_2 = V_2 \bigcup v_2^z$, where v_2^z denotes the vertex associated with v_1^i and v_1^j
    end if
  end for
end for
Computing the cost, C_2, and adjacency, A_2, matrices for G_2
for each node $(v_2^1 \ldots v_2^{n_2})$ **do**
  for each node $(v_2^1 \ldots v_2^{n_2})$ **do**
    if $v_2^i \neq v_2^j$ **then**
      determine number of moves required to move from v_2^i to v_2^j using A_1
      $A_2(i,j) = 1$
      $C_2(i,j) =$ number of moves
    end if
  end for
end for
Compute minimum cost open path on G_2 such that each node in V_2 is traversed only once



(a) (b)

Figure 3.5: (a) Graph R overlayed with some G_3 nodes, denoted by $\otimes$. (b) Subgraph G_3 for the G_1 and R in Figures 3.3(a) and 3.3(b).

Algorithm 2 Construction of the vertex set of $G_3 = (V_3, E_3)$

Given G_1, A_1 and R, A_R
$V_3 = 0$
for each node $(v_1^1 \dots v_1^{n_1})$ **do**
  for each node $(v_1^1 \dots v_1^{n_1})$ **do**
    for each node $(v_1^1 \dots v_1^{m_1})$ **do**
      if $v_1^i \neq v_1^j \neq v_1^k$ **then**
        if $(l_{ij}, l_{jk}$ or $l_{ik} \in L_1)$ **then**
          $V_3 = V_3 \bigcup v_3^x$ where v_3^x denotes the vertex associated with v_1^i, v_1^j, v_1^k
        end if
      end if
    end for
  end for
end for

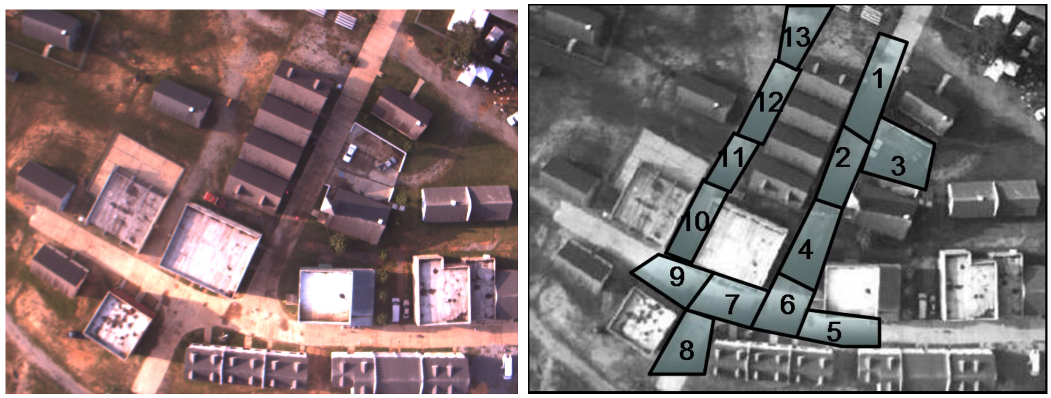
Algorithm 3 Computation of the adjacency and cost matrices, A_3 and C_3, for $G_3 = (V_3, E_3)$

Initialize A_3, C_3
for each node $(v_3^1, \dots v_3^{n_3})$ **do**
  for each node $(v_3^1, \dots v_3^{n_3})$ **do**
    if $v_3^i \neq v_3^j$ **then**
      Calculate minimum number of moves from v_3^i to v_3^j
      $A_3(i, j) = 1$
      $C_3(i, j)$ =minimum number of moves
    end if
  end for
end for

G_1 is required to determine the weight of every edge in E_3. For the three robot case, every edge in the set L_1 may potentially be associated with more than one node in V_3. Thus, the optimal plan for the three robot case would result in a path that contains a subset of the nodes in V_3. For this example, an optimal plan starting at the configuration given by node $1'$ is the path $\{1', 2', 4'\}$ with a total cost of 4, and does not contain node $3'$. In general, given a starting node on G_3, a greedy algorithm is used to compute a path on G_3 such that traversal of each node on the path increases the number of measured edges in L_1. Thus, at any configuration, the next configuration chosen is the one that increases the number of edges measured in L_1 and requires the least amount of moves to reach. During execution, robots are allowed to cross each other as they switch from one configuration to another since inter-robot collisions can be prevented via each robot's local obstacle avoidance routines.

3.1.3 Experimental Setup and Results

The objective of this experiment was to deploy a team of three robots to obtain a radio signal strength map for the Military Operations on Urban Terrain (MOUT) training site, located in Ft. Benning, Georgia, where radio signal strength data is important for operations such as surveillance, reconnaissance, and search and rescue. Figure 3.6(a) is an aerial surveillance picture of the MOUT site. The image was obtained using a fixed wing UAV taken at an altitude of 150 meters. The area shown is approximately 90×120 squared meters. More information on the experiments conducted at the MOUT site can be found in [15, 35, 36].



(a) (b)

Figure 3.6: (a) An overhead view of the MOUT site taken from a fixed wing UAV at an altitude of 150 m. The area shown is approximately 90m × 120 m. (b) A manual cell decomposition of the free configuration space for the MOUT site.

Results

The multi-robot team consists of five autonomous ground vehicles described in Chapter 2, Section 2.1.2. Since the objective was to obtain the desired radio signal strength map for a given decomposition of the free space rather than to determine the appropriate decomposition, we assumed a cell decomposition of the free space shown in Figure 3.6(b), which was obtained by hand. The corresponding roadmap and radiomap graphs for this particular decomposition are shown in Figure 3.7. The edges for the radiomap graph were selected such that they cover the main North-South and East-West roadways on the MOUT site where other planar multi-robot experiments were conducted ([15, 35, 36]).

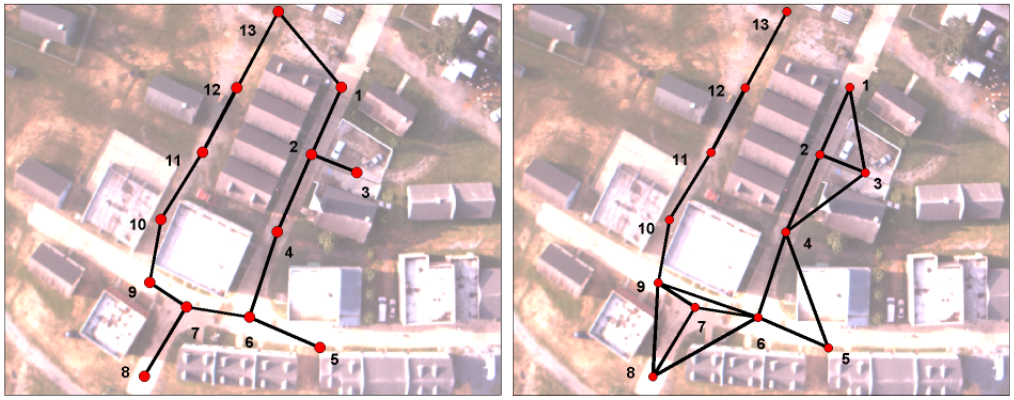
Following the procedure outlined in the previous section, we obtained the three robot exploration graph, G_3, which contained 188 nodes[1]. Rather than weight the edges of G_3 with the total moves to go from one configuration to another, we weighted each edge by the total Euclidean distance the team would have to travel to get from

[1]The roadmap and radiomap graphs each contained 13 nodes.

one configuration to another to more accurately capture the cost associated with each configuration change. The starting configuration for the 3-robot team was selected to be at nodes $\{1, 2, 3\}$ in Figure 3.7. The exploration strategy that was obtained would deploy the robot team in the following sequence: $\{1, 2, 3\}$, $\{1, 2, 4\}$, $\{2, 3, 4\}$, $\{4, 5, 6\}$, $\{6, 7, 9\}$, $\{6, 8, 9\}$, $\{7, 8, 9\}$, $\{9, 10, 11\}$, $\{10, 11, 12\}$, and $\{11, 12, 13\}$.

Once this strategy was obtained, a centrally located waypoint was selected for each cell and each robot was then assigned a set of waypoints to be traversed based on the exploration strategy. Using GPS, the robots navigated to each of their assigned locations. Upon arrival at each waypoint, the robots would synchronize and measure their signal strengths to other team members. If the synchronization failed, each robot would move on towards the next waypoint on its assigned list. To enable each robot to return to its starting position after completion, we assigned each robot its starting position as its last waypoint. The waypoints and cell decompositions were manually chosen to minimize the number of failed synchronizations during execution based on prior knowledge of the variation of signal strength with distance (See Figure 1.1). In addition, each robot was continuously logging both signal strength and position data such that in the event of a failed synchronization the information could be retrieved. There were no synchronization failures during the experiment.

Figure 3.8 shows the radio signal strength map constructed for the MOUT site. The numbers by each edge are the averaged normalized signal strength measurements obtained by the robots located at each pair of positions. On average the GPS errors ranged from $2-3$ meters to as much as 5 meters. However, the robots were generally able to stay within the boundaries of the convex cells.



(a) (b)

Figure 3.7: (a) Roadmap graph used for the site shown in Figure 3.6(a). (b) Radiomap graph for the site shown in Figure 3.6(a).

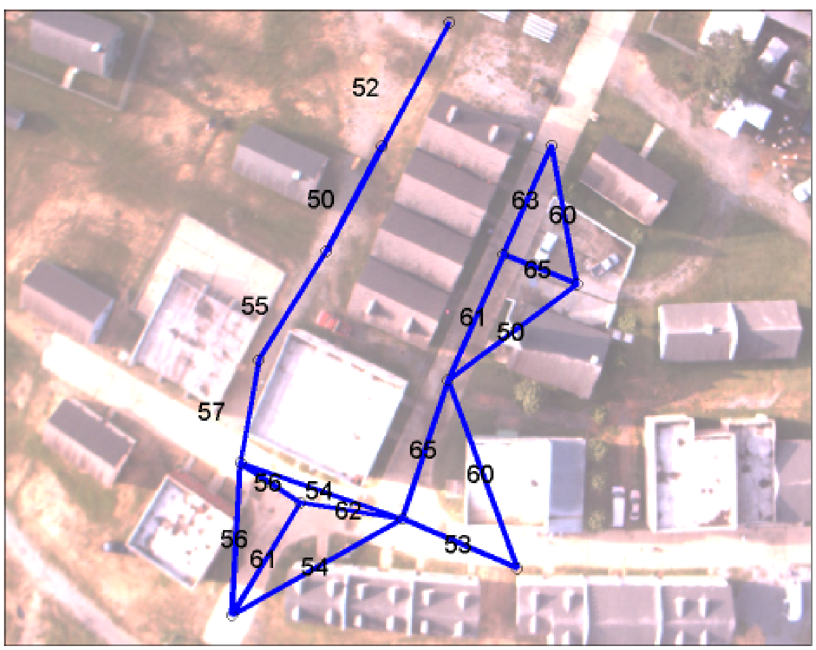


Figure 3.8: Radio signal strength map obtained for the MOUT site. The number on each edge is the average normalized signal strength for each position pair.

3.2 Reactive Controllers for Communication Link Maintenance

In this section, we consider the problem of guiding a group of N robots to a set of goals, or simply a desired boundary (curve), while maintaining point-to-point communication links. We discuss the synthesis of reactive controllers that allow each robot to respond to changes in its perceived communication link quality with respect to other team members within its sensing range. We present experimental results with our multi-robot testbed in two separate outdoor environments. In the first experiment, reactive controllers were used in conjunction with the radio connectivity map shown in Figure 3.8 to determine a deployment strategy for a reconnaissance task on the MOUT site using a four-robot team. In the remaining experiments, reactive controllers capable of responding to changes in signal strength or data throughput were used to maintain point-to-point communication links in a perimeter surveillance task conducted in a separate outdoor environment.

3.2.1 Controllers

In general, for a team of N robots each with kinematics $\dot{q}_i = u_i$, where i denotes the i^{th} robot, $q_i = (x_i, y_i)^T$ denotes the i^{th} robot's position, and u_i denotes the i^{th} robot's control input, consider the following controller

$$u_i = -k\nabla_i \phi_i(q_i) - \sum_{j \in \Gamma_i} \nabla_i g_{ij}(q_i, q_j) \tag{3.1}$$

where k is a positive constant scalar, ϕ is some artificial potential function, and Γ_i denotes the set of neighbors for agent i. The first term of the control law (3.1) guides each robot to its goal position and the second term maintains the constraints that need to be satisfied between robot i and a pre-specified set of neighbors. The functions, $g_{ij} : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$, are artificial potential functions used to model inter-robot constraints. We are interested in maintaining radio connectivity, thus g_{ij} should model the radio propagation characteristics among agents such that $-\nabla_i g_{ij}$ results in a policy that increases the quality of the communication link between robots i and j, where ∇_i denotes the gradient with respect to the coordinates of the i^{th} robot.

As described in Section 2.1.2, our multi-robot team consists of five modified radio controlled scale model trucks, and therefore cannot be described by the simple kinematic model $\dot{q}_i = u_i$. However, taking inspiration from Equation (3.1), our reactive controller is composed of two components: one for navigation to specific goal positions and one that modifies the navigation based on variations in a robot's link quality. These controller elements correspond directly to the first and second terms of Equation (3.1). For each goal position, a reference heading, similar to the descent direction of a potential field controller, is computed by the navigation component. Given this "descent direction", a look-ahead waypoint is generated based on the vehicle's speed and position. Then a simple proportional, integral, and derivative (PID) controller is used to steer the robot towards the look ahead waypoint. The process is repeated until the goal position is reached.

To maintain constraints, each robot continuously monitors the quality of the communication link(s) to its specified set of neighbors. In our experiments, our robots have the capability to continuously monitor either their signal strength to its neighbors, as in Figure 1.2(b), or the number of successful transactions[2] per unit time as in Figure 1.2(a). When the link quality drops below a minimum threshold, the constraint maintenance component will either stop the robot or move it closer

[2]This metric is defined more precisely later in Section 3.2.2.

to its neighbor until the quality returns to an acceptable level. When stopped, a robot can wait for a specified time interval before attempting to move towards its goal again. If the stopped robot perceives an increase in its link quality above the acceptable level, it can once again attempt to move towards its goal. Such recovery measures may be used to lessen the times a robot is caught in a spatio-temporal dip in link quality due to dynamic changes in the environment. In other words, local minima situations, in which a robot stopped before reaching its goal, caused by some temporary interference in the environment. Additionally, these measures also ensure that a robot is constantly minimizing its distance to the goal as long as all constraints are satisfied. We have purposefully incorporated these recovery measures in some of our experiments to highlight the reactive nature of our controller. The algorithm is summarized in Algorithm 4. In this algorithm, the "Recover" behavior drives the robot closer to the neighbor it is attempting to maintain its link quality with. While we chose a binary signal derived from the link quality to control the robots since they are designed to travel at a constant fixed speed, there is an effective deadband in the controller given by the "Minimum" and "Acceptable" quality thresholds in Algorithm 4.

By design, our reactive controller favors constraint satisfaction above reaching the goal position. This is to ensure that the quality of the communication link is always maintained, thus providing a human operator at a remote base station the ability to monitor the various communication links in the network. At the base station, display panels with each robot's imagery data and/or the signal strength measured by the various team members can be displayed. In the event the Base did not receive new data from a particular robot, or should some particular link exhibit low signal strength, an indication of the detected failure is relayed to the human operator. This capability, possible because the team always remains connected, enables the operator

Algorithm 4 Link Quality Constrained Navigation

```
if LinkQuality < Minimum then
  Recover;
  recover_flag = true;
end if
if Minimum < LinkQuality ≤ Acceptable then
  if recover_flag then
    Stop and wait;
    recover_flag = false;
    stopped = true;
    waitTime = current time;
  end if
  if (current time - waitTime) > MaxWaitTime then
    Retry going to goal;
    stopped = false;
  end if
end if
if LinkQuality > Acceptable then
  Go to goal;
end if
```

to decide whether or not to deploy additional robots, or to re-organize the team.

3.2.2 Link Quality Estimation

Signal strength between a sender and a receiver is a function of the transmission power, antenna gains, and signal attenuation. Our robots are equipped with JBoxes that, among other things, provide signal strength measurements to every node on the network. We refer the interested reader to [89] and [75] for operational details on the JBox.

In contrast, it is difficult for an individual robot to estimate the available bandwidth at any given point in time since bandwidth is a function of the number of nodes, the amount of traffic on the network, as well as the signal strength. In multi-robot applications, it is often relevant to talk about bandwidth in terms of units of application level data that can be transmitted, therefore we define a *successful transaction* to be the transmission of one unit of application level data sent by a sender with an acknowledgment of receipt sent by the receiver. A robot's conservative estimate of available bandwidth is determined based on the number of successful transactions it achieves over some interval of time.

For our experimental setup, we set one unit of application level data equal to a JPEG image of approximately 10 KB in size. We define a successful transaction to be the transmission of such an image by the sender followed by the receipt of acknowledgment sent by the receiver. Then, based on the desired transaction rate, *i.e.* number of successful transactions per time interval, the robot, *i.e.* the sender, will periodically evaluate its connection with the receiver. The available bandwidth controller comprises two stages of response: network usage throttling, and robot repositioning corresponding to the "Recover" behavior in Algorithm 4. A robot that is streaming data over the network is capable of detecting when a network connection is not keeping up with the load being put on it, *i.e.* when some messages are dropped due to a full send buffer. When this type of application–level packet loss is detected, the system will automatically throttle communication over this particular network link until a prescribed threshold is hit. This threshold is determined based on mission specifications; we typically specified video at a rate of three frames-per-second as a requirement because this was a realistic target for the 802.11b hardware in use, and provided sufficient video coverage to convey situational awareness. When the network throttling mechanism bumps into the lower threshold, the controller subsumes position control to move the robot closer to the peer it is attempting to communicate with. This action serves to increase signal strength, which reduces the number of low-level transmission retries caused by noise or attenuation.

All throughput estimation in this framework is conservative; we do not attempt to measure maximum data rates available on the network, but rather we verify that some prescribed minimum data throughput rate is available. This approach minimizes the amount of network traffic related solely to throughput measurement, and instead leverages throughput assessment on normal data traffic when such traffic satisfies the constraint. When normal traffic is not of sufficient volume to verify that the minimum available bandwidth constraint is met, a connection monitor will periodically verify available throughput. This latter mechanism, which simply sends data at a rate that verifies constraint satisfaction for a short period of time, allows us to deploy robots that do not maintain consistent data flow back to the base station, *e.g.* robots that only send event data, yet still be confident that the available throughput would likely be available if it should be needed.

3.2.3 Experimental Results

In this section we present three sets of experimental results. The first experiment was conducted at the MOUT site, shown in Figure 3.6(a). This experiment was modeled after a reconnaissance application where the objective was to deploy a team of four robots to obtain surveillance imagery at a designated location out of line-of-sight and single-hop radio communication range with the base station. In this experiment we used information gleaned from our radio connectivity map, shown in Figure 3.8, to determine the deployment strategy, and coupled this with low level reactive controllers to enable the team to respond to unforeseen changes in signal strength. The second and third sets of experiments were conducted at one of the University of Pennsylvania's soccer fields. A satellite image and a schematic of its surroundings are shown in Figure 3.9. These experiments were based on a perimeter surveillance application in which each robot was required to send imagery

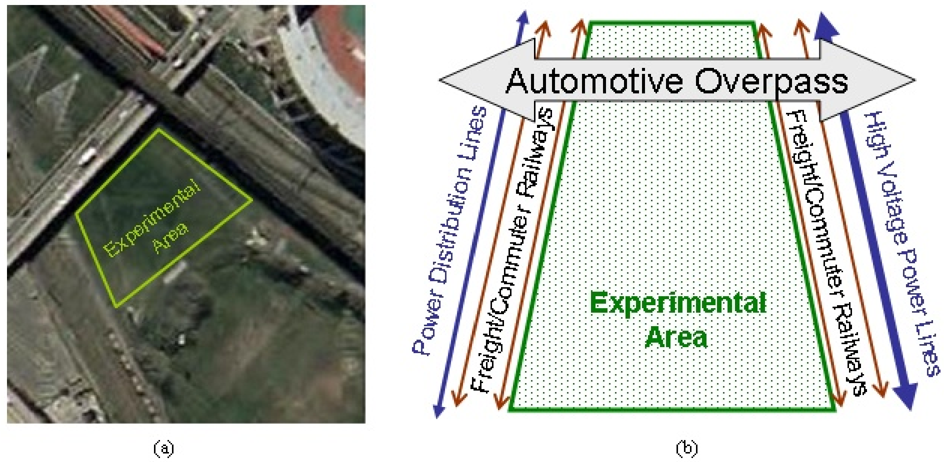


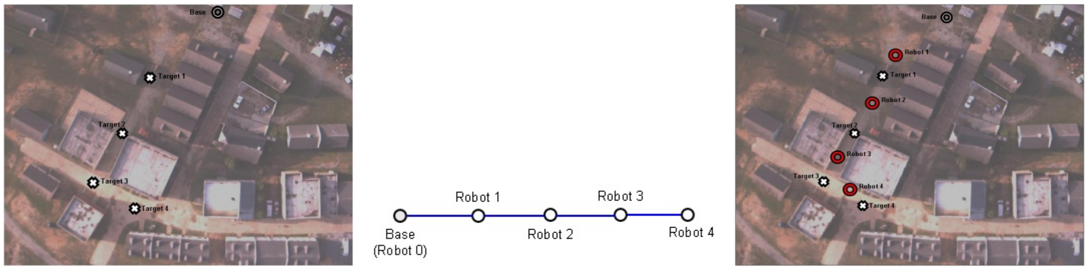
Figure 3.9: (a) A satellite image of the soccer field and its surrounding. (b) Detailed schematic of the experimental area and its surroundings.

data back to a base station. In these experiments, we focused on the individual robot's capability to respond to changes in signal strength or perceived available bandwidth rather than the pairing of high level planning with reactive controllers to ensure communication link maintenance. Once again, our multi-robot team for this set of experiments consisted of the five UGVs described in Section 2.1.2, one of which was chosen to be the base station (Base). We remind the reader that our robots were configured to travel at a fixed speed of 1 m/s and refer the interested reader to Chapter 2 for a picture of the robots.

Reconnaissance at a MOUT Site

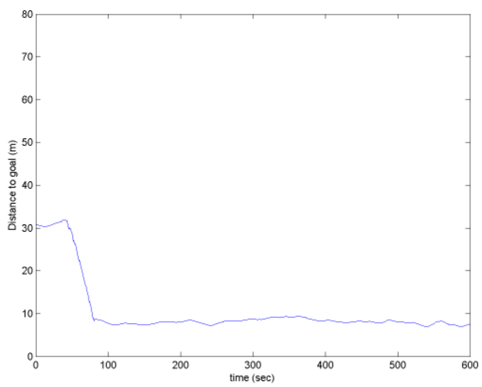
In this experiment, we deployed a team of four robots to obtain surveillance imagery at a designated location out of direct radio communication range. The objective was to deploy four robots to four separate goal positions such that the team formed a linear multi-hop network. Each robot's goal position was determined based on signal strength information given by our radio connectivity map, shown in Figure 3.8. The goal positions were chosen to be slightly beyond nodes 9, 10, 11, and 12 in the roadmap graph shown in Figure 3.7(a) with respect to the Base location (see Figure 3.10(a)). To account for unforeseen variations in signal strength during mission execution, we tasked the i^{th} robot to monitor its signal strength to the $(i-1)^{th}$ robot, and stop when the signal strength dropped below the acceptable threshold, *i.e.* $\mathsf{MaxWaitTime} = \infty$ in Algorithm 4. The Base was considered the 0^{th} robot. The radio connectivity map was then used to determine the minimum acceptable signal strength for each of the robots. The minimum acceptable signal strength was set to 55, 65, 65, and 60 for robots 1, 2, 3, and 4 respectively. Once the team stopped, the human operator at the Base requested images from the robot that was closest to the location of interest, *i.e.* the 4^{th} robot. In this experiment, data was only transmitted between the 4^{th} robot and the Base via the linear multi-hop network.

As shown in Figure 3.10(c), although the targeted locations were chosen to ensure team connectivity, these locations were not reached since each robot was also responding to changes in the real-time signal strength measurements to its designated neighbor, ensuring its signal strength was above the required threshold. Figure 3.11 shows the distance of each robot to its respective goal position over time. These measurements were obtained using each robot's raw GPS data with approximately 2–3 meters in accuracy, and, thus, the slight variations are due to GPS errors rather than robot movements. The final signal strength measurements for robots 1, 2, 3 and 4 were 52, 63, 64, and 57 respectively. While there exists some unpredictability in terms of each robot's ultimate destination, operations at the limits of hardware capabilities, such as demonstrated here, fall outside typical confidence intervals of reliable simulation. If dynamic responses are not allowed, then mission specification must be performed with such a level of conservatism as to severely limit system capabilities.

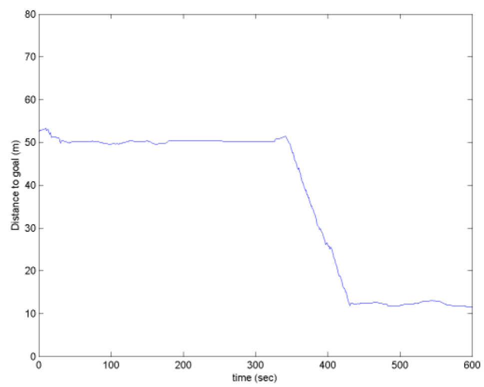


(a) (b) (c)

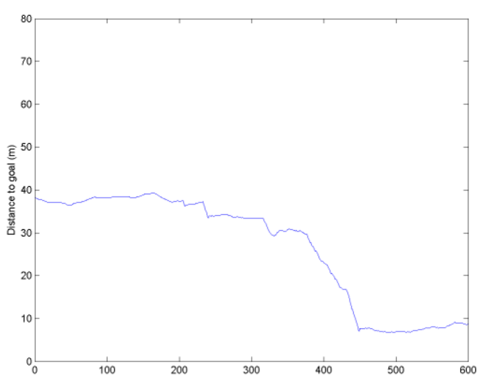
Figure 3.10: (a) An overhead view of the MOUT site. The location of the Base is denoted by ◯ and the target locations for the team are denoted by ×. (b) The underlying communication graph for the reconnaissance application. (c) The final positions attained by each robot and their designated target locations, denoted by ◯ and × respectively.



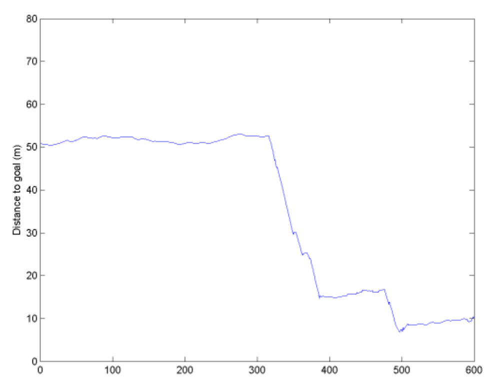
(a) Robot 1



(b) Robot 2



(c) Robot 3



(d) Robot 4

Figure 3.11: Robots' distances to their respective goals over time. The data is obtained using each robot's raw GPS data with approximately 2–3 meters in accuracy.

Perimeter Surveillance Application

These next experiments were based on a perimeter surveillance application where robots would navigate to positions on the desired perimenter and send imagery data back to a base station. In our experiments, we tasked a team of four robots to go to four separate goal positions where the positions were chosen to represent distinct locations on the perimeter of interest. In addition, each robot was also tasked to monitor its signal strength or estimated available bandwidth to the Base while continuously sending imagery data to it. Thus, every robot was sending approximately 10 KB JPEG images back to the Base, as compared to the previous experiment, where only one robot was sending data to the Base.

At the Base, a display panel with each robot's imagery data was provided to the operator. In the event the display panel did not receive new data from a particular robot over a specified interval of time, the panel would highlight the display box for that particular robot. The objective of these experiments was to focus on an individual robot's ability to respond to changes in signal strength or perceived available bandwidth. Thus, we only considered single hop network connections between each robot and the Base. Additionally, to avoid robots being caught in a "local minimum" due to dynamic changes in the environment that may affect signal strength measurements, and to better emphasize the reactive nature of our controllers, we set the MaxWaitTime in Algorithm 4 to a finite time.

The first such experiment conducted at this location demonstrates the reactive controller in the presence of dynamic network disturbances. In this experiment, the network disturbance was caused by the addition of a second robot to a network originally used by a single robot transmitting a video stream to the Base. As shown earlier in Figure 1.2, as new members are introduced into the team, the maximum bandwidth available to each robot drops.

Figure 3.13 shows how our controller responded to the addition of the second robot. We first deployed a single robot, Robot 1, to a goal position and required that it continuously send imagery data to the Base while maintaining a minimum transaction rate of 7 transactions per second. This can be seen in the top graph of Figure 3.13 where each marker denotes the number of transactions received in between the time of the current marker and the one before it, approximately 10 seconds. The MaxWaitTime variable in Algorithm 4 was set to 30 seconds. A schematic of the deployment strategy is shown in Figure 3.12. At around $t = 60s$, Robot 1 settled to a location about halfway to the goal as shown in the bottom graph of Figure 3.13. Between $t = 90s$ and $t = 125s$, the robot attempted to reach its goal a second time and settled to a similar location shown in bottom two graphs in Figure 3.13. A second robot, Robot 2, transmitting to the Base was introduced to the network at approximately $t = 130s$, as shown in the first graph of Figure 3.13. Immediately, Robot 1 was no longer able to maintain the required transaction rate and therefore began moving back towards the Base in an effort to boost its transaction rate. This can be seen in the last graph of Figure 3.13 where the robot's distance to its goal starts increasing. Put simply, a robot was tasked with sending video back to a base station as it monitored a perimeter. In the early stages of the mission, the robot could transmit video at a high rate, but as the robot moved further away from the base station, the rate at which it could transmit video dropped. The robot continued moving until the transmission rate hit a pre-defined threshold. At this point, the robot stabilized its distance from the base station in order to maintain the minimum required transmission rate. Once another network user was added, the first robot had to move yet closer to the base station. This response is made because the transmission rate for a single robot is a function both of total network usage

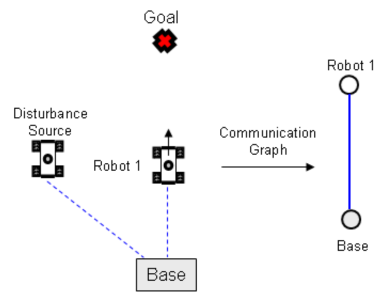


Figure 3.12: Schematic of experimental setup and underlying communication graph for the results shown in Figure 3.13. On the left, the dashed line denotes the communication link monitored by the robot. In this experiment, Robot 2 was used to cause a network disturbance by transmitting to the Base.

and signal quality. Once the robots are transmitting at the lowest acceptable rate, the other variable they can individually have an effect on is radio signal strength, which is related to the distance between transmitters. Utilizing this last control, the first robot will settle on the maximum distance at which it can maintain the required transmission rate given the new networking situation. This re-positioning is automatic, and does not require any changes in calibration or thresholds to reflect the new state of the network.

The behavior demonstrated by the two-robot experiment allowed us to successfully deploy a team of robots capable of maximizing network utilization while providing effective situational awareness. Subsequent experiments involved deploying a team of four robots to separate locations from a starting position by the Base. Each robot was tasked to continuously send imagery data from its camera to the Base at a rate above a pre-determined minimum transaction rate. Goal positions for each team member were chosen to provide a wide net of surveillance coverage. This type of goal specification is flexible in that it establishes a vector for the robots to move along, as opposed to specific waypoints to achieve. Thus, success is a matter of degree, rather than a binary distinction: we wish to effectively cover as wide an area

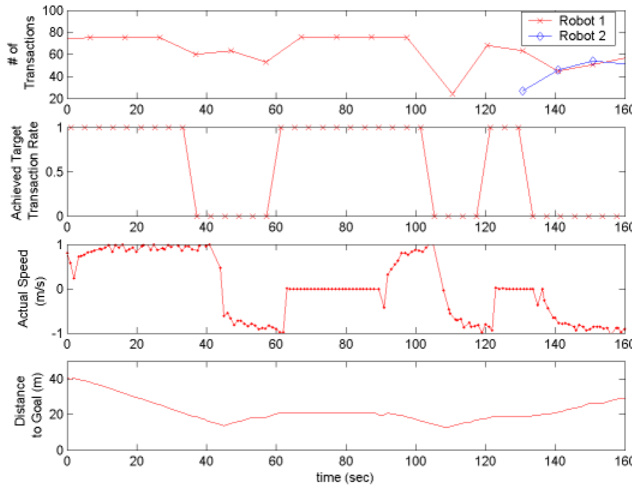


Figure 3.13: Top: Number of transactions received by the Base from Robot 1 and Robot 2 over time. Robot 2 began its transmission at around $t = 130s$. Center Top: 1 denotes Robot 1 achieved the target transaction rate and 0 otherwise. Center Bottom: Actual speed achieved by Robot 1. Positive speed denotes the robot is moving towards the goal and negative speed denotes the robot is moving towards the Base. Bottom: Robot 1's distance from the goal.

as possible. Using the control algorithm described in Section 3.2.1, each robot would move towards its goal until the link quality dropped below the minimum threshold, at which point it would move back towards the Base and stop when the link quality rose back above the chosen minimum level. Once stopped, each robot would wait for a fixed time interval before attempting to go to its goal again. Two sets of experiments were conducted in which each robot's controller reacted based on changes in: (i) signal strength measurements and (ii) estimated transaction rate. A schematic of the deployment strategy is shown in Figure 3.14. Four trials for each experiment were conducted. Since the results are similar for all four robots in all four trials, we have selected one representative result for each set of experiments as shown in Figures 3.15 and 3.16.

Figure 3.15 shows signal strength measured by the robot to the Base along with

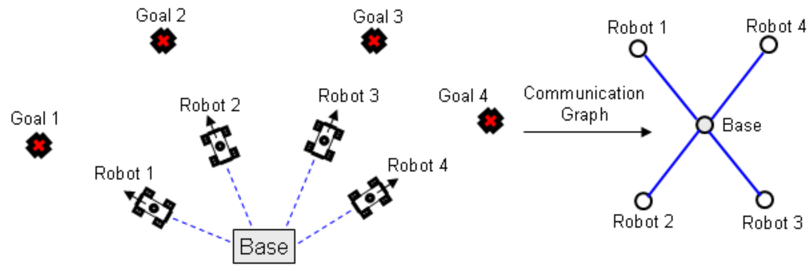


Figure 3.14: Schematic of experimental setup and underlying communication graph for the results shown in Figure 3.15 and 3.16. Similarly, the dashed lines in the figure on the left denote communication links monitored by each robot.

the corresponding commanded speed and actual speed. The MaxWaitTime in Algorithm 4 was set to 60 seconds. Initially, when the robot was close to the Base, the signal strength measurements were high. As the robot moved toward its goal, we saw these measurements drop. The first time the signal strength dropped below the minimum threshold, around $t = 45s$, the robot attempted to move closer to the Base. Once the signal strength rose above the threshold, the robot stopped. Subsequently, the robot made additional attempts to move towards the goal but had to stop and move closer to the Base each time.

Similarly, Figure 3.16 shows the results for one of the four robots whose controller was reacting to changes in its estimated transaction rate. In these experiments, MaxWaitTime in Algorithm 4 was set to 120 seconds and the minimum rate was set to 3 transactions per second. Similar to the results shown in Figure 3.15, the robot's transaction rate dropped as it moved further away from the Base as shown in Figure 3.16(a) and the top graph in Figure 3.16(b). We note that it is possible for the robot to reach its goal location and achieve its target transaction rate. This can be seen in the bottom graph in Figure 3.16(b) where at approximately $t = 50s$ the robot is within 2.5 meters of the goal location. Around the same time, we see a change in the robot's speed from positive to zero as shown in the second and third graphs in the

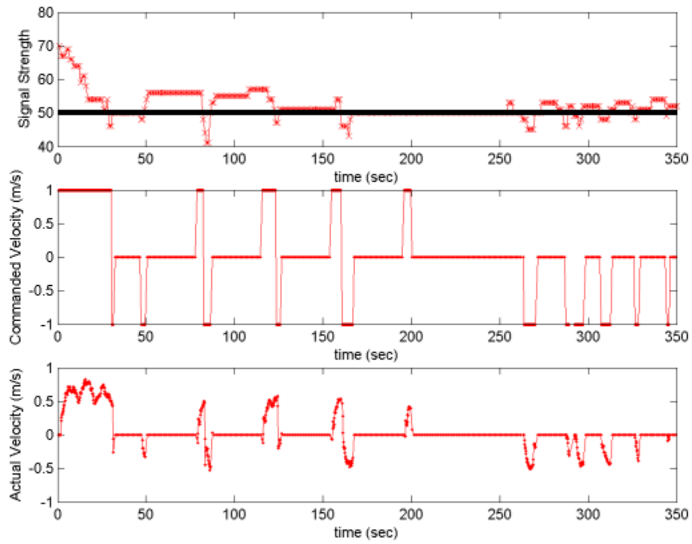
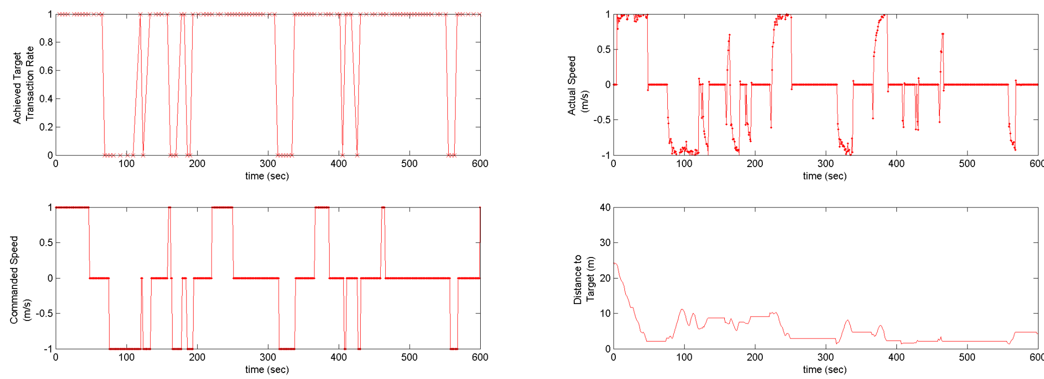


Figure 3.15: Top: Signal strength measured by Robot 1 to the Base. The solid black line denotes the minimum acceptable level. Center: Commanded speed based on the signal strength measurements. Positive speed denotes Robot 1 is moving towards the goal and negative speed denotes it is moving towards the Base. Bottom: Estimated speed achieved by Robot 1 based on the commanded speed. Data for Robots 2, 3 and 4 are similar and thus not shown.

same figure. When the transaction rate dropped, around $t = 75s$, the robot began to move back towards the Base, leaving its goal location.

3.2.4 Discussion

By design, our reactive controller favors constraint satisfaction above reaching the goal position. This was seen in both the reconnaissance and perimeter surveillance experiments. By designing our controller in this fashion, we ensure the human operator will always be able to get real-time status updates from the team. Should the operator notice certain robots not getting close enough to their goal positions, the operator could deploy additional robots to provide a multi-hop link to the Base or request a reconfiguration of the whole team should the original target connectivity prove unachievable. Similarly, should an intermediate robot fail in a configuration



(a) (b)

Figure 3.16: (a) Top: 1 denotes the target transaction rate was achieved and 0 otherwise. Bottom: Commanded speed based on whether the target transaction rate was achieved. Positive speed denotes Robot 1 is moving towards the goal and negative speed denotes it is moving towards the Base. (b) Top: Actual speed achieved by Robot 1 based on the commanded speed. Bottom: Robot 1's distance from the goal. Data for Robots 2, 3 and 4 are similar and thus not shown.

as shown in Figure 3.10(b), this information would be immediately reflected at the Base. Under these circumstances, the robot furthest away from the Base would surely lose connectivity, however this could be mitigated through the implementation of communication recovery measures, such as return to the last known location with good connectivity, the dispatch of additional robots, or the reconfiguration of the remaining robots still within communication range of the Base.

When considering multi-hop scenarios, it would be important to set the MaxWaitTime variable in Algorithm 4 to infinity. In our perimeter surveillance experiments, we were only considering single-hop communication links. One of the objectives in these experiments was to show the reactive nature of our controller along with minimizing the amount of time a robot was caught in a spatio-temporal "local minimum" due to dynamic interference in the environment. Therefore, in these experiments MaxWaitTime was set to a finite time. In contrast, our MOUT site reconnaissance experiment, where the objective was to deploy a multi-hop network, did not have these recovery responses because we did not want the constant back and forth motion to affect the robots ability to send data through the multi-hop network. Had the reactive response been present, it is very likely the constant back and forth motion would affect the team's ability to reliably relay information to the Base.

In all our experiments, the minimum thresholds were chosen based on a combination of previously collected data and/or specific mission requirements. In the reconnaissance experiment, signal strength thresholds were determined based on information gleaned from a radio connectivity map. On the other hand, when considering perceived network bandwidth, the minimum acceptable threshold was determined based on hardware limitations in conjunction with acceptable transmission rates based on mission requirements specified by the human operator.

Lastly, our reactive controllers can be easily decentralized based on the methodology proposed in [40], and thus scaled to large number of robots. Rather than specifying specific goal positions for every robot in the team, [40] specifies a one-dimensional boundary curve for the team. This, however, does not necessarily mean the existing network would be able to handle the increase in traffic brought on by the increase in team size.

3.3 Conclusion

In this chapter, we have presented a paradigm and algorithms for deploying a mobile robot network with specifications on end-to-end performance. Our approach entails the automated construction of a radio map for a partially known urban environment which can then be used to deploy a team of robots, and control algorithms that drive the team to designated targets on some desired boundary (curve) while maintaining communication link quality.

There are two main contributions. First, we developed a method for obtaining radio signal strength maps that can be used to plan multi-robot tasks and also serve as useful perceptual information. Second, since a radio signal strength map only serves to create a nominal model, we have shown the importance for individual robots to have the ability to monitor communication links, in particular signal strength measurements as well as available data throughput. This method of link quality control provides scalability in the number of robots that may be added to the network, and an abstraction of the underlying network architecture. Since the robots constantly strive to maximize network usage efficiency, robots may be added to, or removed from the network without changes to any thresholds or calibration numbers. This type of deployment characteristic is extremely important as robot team sizes scale, as we want teams to take advantage of bandwidth when it is available, and automatically scale back individual usage as available resources are stretched thin. Moreover, we have also shown that channel contention between multiple nodes can have a severe adverse effect on total network throughput. By monitoring successful transactions, we give our robots the ability to throttle their own network usage such that the transmission rates of each robot stabilize to levels that make efficient use of the network.

Additionally, as shown in our perimeter surveillance experiments, in dynamic environments where radio propagation characteristics may exhibit significant changes over time, it is good practice for agents to always attempt to move closer to the goal regardless of where they first come to a stop. The forward movement is the only way to confirm that positions closer to the goal violate communication constraints, and to ensure the agents always minimize their distance to the goal location while remaining connected. Ideally, robotic agents should be deployed with the capability of monitoring inter-agent signal strengths as well as data throughput. In general, signal strength is a good indicator of potential connectivity while data throughput can efficiently be used to ensure minimum actual data throughput rates. Combining the two, good signal strength paired with unacceptably low throughput may indicate a need for human attention to the network architecture and the demands being placed on it. As such, the communication medium becomes a useful sensor that can be used to monitor the effectiveness of any given multi-robot deployment.

Reactive navigation controllers such as the one presented provide a reliable foundation on which to build scalable, portable, high-level tasks. The reactive controller acts as a scenario-independent support that allows for the deployment of a robot team to any location, regardless of prior reconnaissance. As shown in our reconnaissance experiment, behaviors built on such a controller inherit respect for network constraints, thereby allowing both flexible goal specification and more deliberative trajectory planning done with environmental models that do not necessarily capture all static and dynamic aspects of an environment's radio propagation characteristics. Since the team always remain connected during mission execution, potential failure points in the communication network, as perceived by individual robots, can be relayed back to the base station to trigger contingency management routines, *e.g.* deployment of additional robots or a reallocation of resources. While the strategies presented in this work assign the highest priority to maintaining the network, applications that may benefit from a relaxation of this constraint provide a direction for future work.

Chapter 4

Scalable Motion Control Strategies for Robotic Swarms with Obstacle Avoidance

As the size of robot teams increases, there has been a shift towards a swarming paradigm where robots are programmed with simple but identical behaviors that can be realized with limited on-board computational, communication and sensing resources. Such swarming behaviors are often seen in nature, specifically in biological systems composed of large number of organisms which individually lack either the communication or computational capabilities required for centralized control. Examples of such behaviors can be seen in the group dynamics of beehives [13], ant colonies [74], bird flocks, steer herds and fish schools [66].

We are interested in deploying robotic teams for applications such as perimeter surveillance, monitoring of specified areas, and cooperative manipulation where robots may need to surround an object to transport it from one location to another. In these situations, robots may have to communicate with each other in order to integrate and fuse the information acquired by various sensors and as such, limited bandwidth must be preserved to enable the communication of crucial data between team members and/or to a base station. This is especially critical when we consider very large teams where bandwidth often becomes the limiting factor in agents' abilities to transmit data. In these situations, robots must have the ability to accomplish the necessary tasks while minimizing communication overhead and operate with little to no human supervision.

This work addresses the synthesis of decentralized controllers that guarantee the stability and convergence of all the robots to the boundary of a specified shape such that each robot has non-zero velocity tangential to the desired boundary. We present a communicationless strategy where collision avoidance is achieved via a prioritization schemed based on each robot's relation to its immediate neighbors. While part of our approach is similar in spirit to the gyroscopic forces proposed by Chang and Marsden [18], we achieve collision and obstacle avoidance by modulating each robot's speeds independently using smooth positive scalar functions. Thus, unlike Chang and Marsden, neighbors with higher priorities result in slowing down a robot's travel speed rather than exerting a gyroscopic force on the agent. While we assume that agents are holonomic, it is possible to extend our methodology to include non–holonomic robots. This can be achieved because we consider disk–shaped robots, thus enabling the use of feedback linearization techniques to linearize the non–holonomic model away from each robot's center of rotation.

We begin with the problem formulation and provide some background to our approach in Section 4.1. The controller synthesis is described in Section 4.2. The safety, stability and convergence properties of the controller are discussed in Section 4.3 with simulation results presented in Section 4.4. Lastly, we provide an extension of our proposed controller to handle obstacles in close proximity to the desired boundary in Section 4.5 and conclude with a discussion on directions for future work in Section 4.6.

4.1 Problem Formulation

We consider a group of N planar, fully actuated robots each with kinematics given by

$$\dot{q}_i = u_i \tag{4.1}$$

where $q_i = (x_i, y_i)^T$ and u_i denote the i^{th} agent's position and control input. Thus, the robot state is a 2×1 state vector and the state of the team of robots is given by $\mathbf{q} = \left[q_1^T \ldots q_N^T\right]^T \in \mathbf{Q} \subset \mathbb{R}^{2N}$. We assume disk-shaped agents, each with finite radius r_i.

We would like to design control inputs that will stabilize the group of N robots to the boundary (curve) of a desired smooth, compact set, e.g. shape. We would like to achieve this such that agents have non-zero velocities tangent to the boundary curve, enabling them to travel along the boundary curve in a counter-clockwise direction, all the while avoiding collisions with one another. This is relevant for applications such as perimeter surveillance, surrounding objects for capture, or cordoning off and containing hazardous regions after chemical spills or biological attacks.

We assume the workspace, $\mathcal{W}$, is described by the set,

$$\mathcal{W} = \{q | \, \|q\| \leq R_0\},$$

and let $\partial\mathcal{W}$ denote the boundary of the workspace. For a desired smooth star shape set, $\mathcal{S}$, we denote $\partial\mathcal{S}$ to be the boundary of $\mathcal{S}$ and, similar to [40], we assume that $\partial\mathcal{S}$ is described by a smooth, regular, simple, closed curve of the form $s(x, y) - s_0 = 0$.

Furthermore, we assume the agents have the capability to localize themselves in some global coordinate frame and the ability to sense the proximity of their teammates and/or obstacles within the environment. As such, we define the neighborhood of q_i by the range and field of view of the sensing hardware and denote the set of neighbors in this region by Γ_i. For collision avoidance purposes, we assume a circular influence range, R_i, such that collision/obstacle avoidance maneuvers are active when agents are within each other's influence range.

The objective is to construct artificial potential functions, φ, for the obstacle free environment and augment these with additional behaviors that can locally modify the feedback policy derived from φ to enable the team of N agents to: 1) stabilize onto the desired closed curve (orbit) with non-zero velocities in the orbit's tangent space; 2) avoid collisions with other agents; and 3) avoid collision with unmodeled robot–sized obstacles in the environment. We outline our methodology in the following section.

4.2 Methodology

4.2.1 Assumptions

Given a smooth star shape $\mathcal{S}$, a team of N robots each with radius $r_i > 0$ and influence range $R_i > 0$, we define $r = \max_{r_i \forall i} r_i$ and $R = \max_{R_i \forall i} R_i$. Our goal is to synthesize decentralized controllers that will allow a team to converge to the desired orbit with non-zero velocities in the orbit's tanget space while avoiding collisions. Therefore, the length of $\partial\mathcal{S}$, L, naturally imposes an upper bound on the number of robots, e.g. $N_{max} > 0$, that can travel along the boundary with non-zero velocity. Thus, we make the following assumptions:

1. $N < N_{max}$;

2. $|\rho_{min}| > R$;

3. $\min_{\bar{s}\in[\frac{\pi\rho_0}{3}, L-\frac{\pi\rho_0}{3}]} \|q_0(\bar{s}) - q(\bar{s})\| > R$ for any $q_0(\bar{s}) \in \partial\mathcal{S}$, where $\bar{s} \in [0, L]$ denotes the arclength and ρ_0 denotes the radius of curvature at q_0.

Assumption 1 ensures agents will have non-zero velocities in the orbit's tangent space, *i.e.* the agents will be able to circulate around and along the boundary. Assumptions 2 and 3 ensure convergence by excluding boundaries with sharp turns and star shaped patterns with narrow corridors that may result in robots repelling each other away from the boundary while avoiding collisions.

In situations where robots must avoid neighbors who have been rendered immobile due to damage or failure, let $d(q_i, \partial\mathcal{S})$ be the shortest distance between q_i and $\partial\mathcal{S}$ and Π be the set of broken robots with $|\Pi| = M \ll N$. We make the following additional assumptions:

4. for any $i \in \Pi$, $d(q_i, \partial\mathcal{S}) > (R + r)$;

5. for any $i \in \Pi$, $d(q_i, \partial\mathcal{W}) > 3r$;

6. $\min d(\partial\mathcal{S}, \partial\mathcal{W}) > 2Nr$;

7. for any $j, k \in \Pi$, $\|q_j - q_k\| - (r_j + r_k) > 2R$; and

8. robots have the ability to estimate the velocities of their neighbors.

Assumptions 4–6 are necessary to ensure liveness of the system, *i.e.* the assumptions are necessary to exclude stable configurations where $q_i \notin \partial\mathcal{S}$. The seventh assumption ensures that the "spheres of influence" of the obstacles do not overlap. Lastly, the eighth assumption gives robots the ability to determine whether a robot is broken without requiring communication. This is reasonable since robots already have the ability to infer neighbor positions.

These above assumptions are necessary for our stability and convergence analysis, however, they can be relaxed in practice. In the event that $N > N_{max}$, our proposed controller has the ability to stably drive N_{max} agents towards the boundary while keeping extraneous agents orbiting a distance $2R$ away from the boundary. Similarly, should the curvature constraints of the desired boundary violate the above assumptions, one can easily approximate the existing boundary with another curve that satisfy the above constraints since the bounds are based on hardware specifications. Finally, while system deadlock can in fact occur if the liveness assumptions are not met, these situations can be easily detected and resolved by pre-programming a deadlock prioritization scheme or by allowing minimal inter-agent communications.

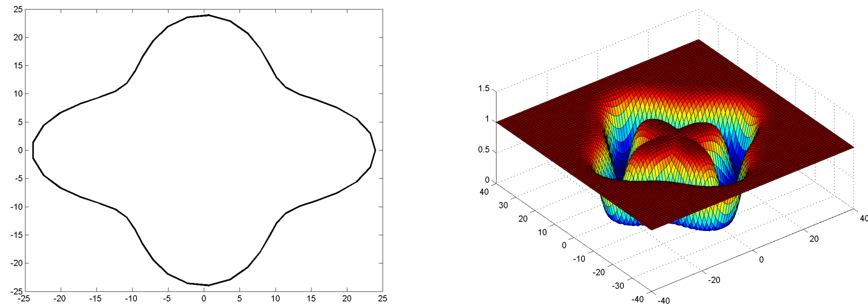
4.2.2 Controller Synthesis

Given a smooth star shape $\mathcal{S}$, we assume that $\partial\mathcal{S}$ is described by a smooth, regular, simple, closed curve $s(x,y)-s_0 = 0$, with $(s(x,y)-s_0) < 0$ for all (x,y) in the interior of $\partial\mathcal{S}$ and $(s(x,y)-s_0) > 0$ for all (x,y) in the exterior of $\partial\mathcal{S}$. Let $\gamma = s(x,y) - s_0$ and $\beta_0 = R_0^2 - \|q\|^2$, we define the *shape navigation function*, φ, as

$$\varphi(q) = \frac{\gamma^2}{[\gamma^2 + \beta_0]}. \tag{4.2}$$

The function φ has the following properties:

- φ is positive semi-definite;
- $\varphi = 0$ if and only if $s(x,y) - s_0 = 0$;
- φ is uniformly maximal, e.g. $\varphi(\partial\mathcal{W}) = 1$;



(a) (b)

Figure 4.1: (a) A star shape whose boundary is given by $r-(a_0+b_0\sin(c_0\theta+d_0))=0$ with $a_0=20$, $b_0=1$, $c_0=5$, and $d_0=\pi/2$. (b) The shape navigation function for the boundary given in (a).

- φ is real analytic.

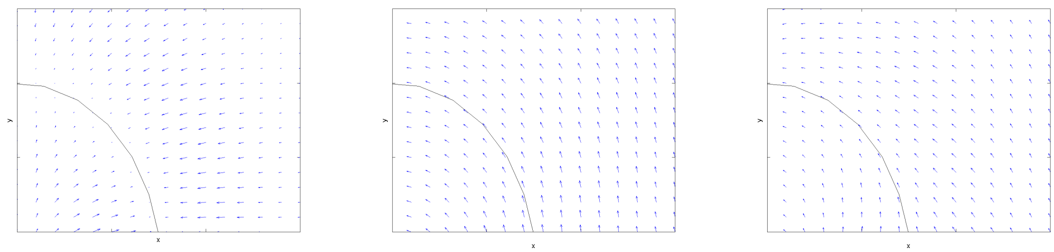
Figure 4.1 shows a star shaped boundary and its shape navigation function. The shape navigation function will generate an input that will drive each agent towards the desired boundary, $\partial\mathcal{S}$.

To enable the agents to travel along $\partial\mathcal{S}$ in a counter-clockwise direction, let

$$\psi = \begin{bmatrix} 0 \\ 0 \\ \frac{\gamma}{\sqrt{\gamma^2+\beta_0}} \end{bmatrix}$$

and we impose an additional input given by $-\nabla\times\psi$, where $\nabla\times\psi$ is a vector tangent to the level set curves of φ. Furthermore, $\nabla\times\psi$ is chosen such that on the boundary, $\partial\mathcal{S}$, each agent has a non-zero tangent velocity. This input enables each agent to travel along the boundary in a counter-clockwise direction[1]. Thus, the proposed

[1]To enable each agent to travel along $\partial\mathcal{S}$ in a clockwise direction consider adding $(\nabla_i\times\psi)g(T_i)$ in (4.3) instead of subtracting.



(a) (b) (c)

Figure 4.2: Vector fields for the star shaped boundary given by $r - (a_0 + b_0 \sin(c_0\theta + d_0)) = 0$ with $a_0 = 20$, $b_0 = 1$, $c_0 = 5$, and $d_0 = \pi/2$. (a) Vector field given by $-\nabla\varphi$. (b) Vector field given by $-\nabla \times \psi$. (c) Vector field given by equation (4.3) with $f(N_i) = 1$ and $g(T_i) = 1$.

decentralized controller in an environment with no obstacles is given by:

$$u_i = -\nabla_i \varphi_i \cdot f(N_i) - \nabla_i \times \psi_i \cdot g(T_i) \tag{4.3}$$

where ∇_i denotes differentiation with respect to agent i's coordinates and $\varphi_i = \varphi(q_i)$ and $\psi_i = \psi(q_i)$. The functions $f(N_i)$ and $g(T_i)$ are positive scalar functions used to modulate each agent's velocity for collision avoidance and their construction is described in the following section. The first term of (4.3) drives the agents towards $\partial\mathcal{S}$ and the second term drives the agents along the level set curves of φ in a counter-clockwise direction. Figures 4.2(a) and 4.2(b) show the vector fields generated by the first and second terms of Equation (4.3) for a star-shaped boundary. Figure 4.2(c) shows the combined vector fields for the same boundary with $f(N_i) = g(T_i) = 1$. In the remainder of this paper we will refer to the first and second term of (4.3) as the descent and tangential velocities of agent i.

4.2.3 Collision Avoidance

For collision avoidance, we would like a prioritization scheme that will enable individual robots to modulate their descent and tangential speeds based on its position in a local prioritization queue. Given φ is common among all agents and robots have the ability to estimate their neighbors' relative positions, every agent i can determine the value of φ for each one of their neighbors. Additionally, by construction, φ increases monotonically as one moves farther away from $\partial\mathcal{S}$ towards the workspace boundary. Therefore, within the domain $\mathcal{W}$, φ serves as a measure of how close an agent is to the boundary. As such, agent i can assign a higher priority to any neighbor $j \in \Gamma_i$ that is closer to $\partial\mathcal{S}$ and lower priority to those who are further away. Similarly, for a star shaped set $\mathcal{S}$, if we denote the center of the star as q_c, then the angle made by the vector $(q_i - q_c)$ with respect to the horizontal axis, which we denote by θ_i, such that $\theta_i \in [0, 2\pi)$, is a proxy for the distance traveled by each agent in the tangential direction. Therefore, agent i can assign a priority to every $j \in \Gamma_i$ depending on whether $\theta_j > \theta_i$ or $\theta_j < \theta_i$.

Since local priorities can be established in both the descent and tangential directions based on the values of φ and θ, it is possible for agent i to decrease its descent speed if it detects any $j \in \Gamma_i$ such that $\varphi_j < \varphi_i$, *i.e.* a neighbor j with higher priority. Similarly, agent i can decrease its tangential speed for any $j \in \Gamma_i$ such that $\theta_j > \theta_i$. Such a scheme would result in slowing down a robot's convergence towards the boundary if it has neighbors with lower values of φ or slow down a robot's tangential speeds if it has neighbors with larger values of θ.

To accomplish this, we would to synthesize $f(N_i)$ and $g(T_i)$ in (4.3) to appropriately decrease the descent and tagential velocities of q_i. Consider the following scalar functions:

$$N_i = \sum_{j\in\Gamma_i,\varphi_j<\varphi_i} \frac{-1}{d_{ij}^{p_1}} \tag{4.4}$$

$$T_i = \sum_{j\in\Gamma_i,\theta_j>\theta_i} \frac{-1}{d_{ij}^{p_1}}, \tag{4.5}$$

where $d_{ij} = \|q_i - q_j\| - (r_i + r_j)$ and p_1 is a positive even number. As such, we define the functions $f(N_i)$ and $g(T_i)$ as

$$f(N_i) = \sigma_-(N_i), \tag{4.6}$$

$$g(T_i) = \sigma_-(T_i), \tag{4.7}$$

where σ_- is defined as the following analytic switching function:

$$\sigma_-(w) = \frac{1}{1 + e^{-t_1(w-t_2)}}.$$

Note as $\sigma_- \to 0$ as $w \to -\infty$ and $\sigma_- \to 1$ as $w \to +\infty$ with $t_1, t_2 > 0$ defined as constant scalars. Figure 4.3 shows the graph for σ_-. By design, as agent i approaches a neighbor with higher priority in either the descent or tangential direction, $f(N_i)$ or $g(T_i)$ would slow down agent i's descent or tangential speeds. Figure 4.4 provides a schematic of the prioritization scheme.

We note since φ and ψ is common among all agents, robots do not have to exchange information. Instead, the positions of the neighbors can be obtained via sensing alone. Moreover, this prioritization scheme is transitive among non-neighboring agents in both the descent and tangential directions in the domains given by $\mathcal{W}$ and $[0, 2\pi)$ respectively. In other words, given $\varphi_i < \varphi_j$ and $\varphi_j < \varphi_k$ such that $i \in \Gamma_j$ and $k \in \Gamma_j$ but $k \notin \Gamma_i$, we can conclude that $\varphi_i < \varphi_k$ and similarly for θ_i.

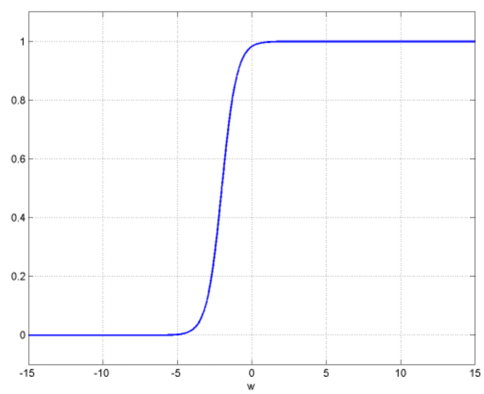


Figure 4.3: Graph of $\sigma_-(w)$ with $t_1 = t_2 = 2$.

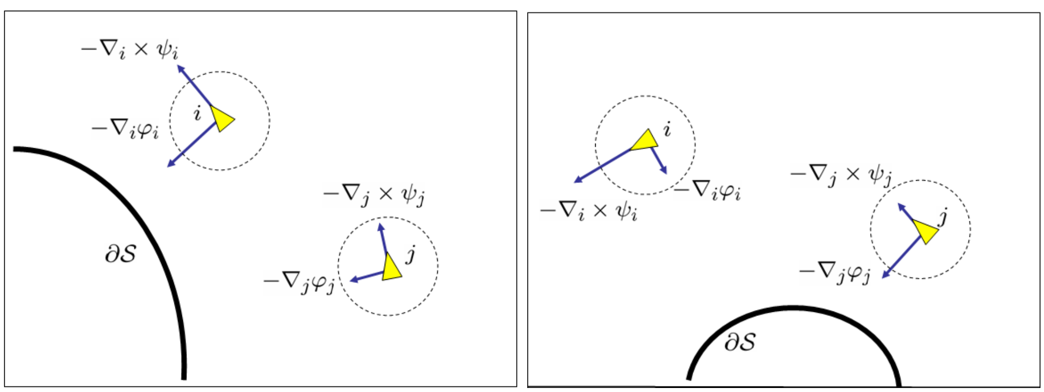


Figure 4.4: Resulting descent and tangential velocity vectors for two neighboring agents after imposing the proposed prioritization scheme. The dark solid line denotes the boundary, the dotted circles denote the size of the agents, and the arrows denote the descent and tangential velocities. (a) Agent i is assigned higher priority in both the descent and tangential directions. (b) Agent i is assigned higher priority in the tangential direction and lower priority in the descent direction.

4.2.4 Obstacle Avoidance

While the proposed collision avoidance scheme ensures inter-agent collisions do not occur, this is only guaranteed while all agents have non-zero velocities. Should an agent fail and become immobilized, the existing methodology may result in preventing the remainder of the team to converge to the desired boundary and resulting in a potential deadlock situation.

In this section, we extend the decentralized controller described in the previous section to give active robots the ability to avoid circular obstacles that are comparable, in size, to other robots, thus avoiding collisions with robots that have been rendered *immobile* for one reason or another. This will ensure graceful degradation of our system in the event of robot failures and gives the agents the ability to avoid unmodeled obstacles.

Consider the following decentralized controller:

$$u_i = -\nabla_i \varphi_i \cdot f(N_i) - \nabla_i \times \psi_i \cdot g(T_i) - \sum_{j \in \Gamma_i, \|v_j\|=0} \frac{\nabla_i \times \beta_j}{d_{ij}^{p_2}} h(q_i, q_j) \qquad (4.8)$$

with $0 < p_2 < p_1$, and β_j and $h(q_i, q_j)$ respectively defined as:

$$\beta_j = \begin{bmatrix} 0 \\ 0 \\ \frac{d_{ij}}{\sqrt{d_{ij}^2 + \beta_0}} \end{bmatrix},$$

$$h(q_i, q_j) = \left(f(\bar{N}_i)\sigma_-(\varphi_{ij}) - f(N_i)\sigma_+(\theta_{ij})\sigma_+(\varphi_{ij}) \right).$$

Here $\varphi_{ij} = \varphi_i - \varphi_j$ and $\theta_{ij} = \theta_i - \theta_j$ with σ_+ and $\bar{N}_i$ defined as

$$\begin{aligned}
\sigma_+(w) &= \frac{1}{1 + e^{t_1(w+t_2)}}, \\
\bar{N}_i &= \sum_{k \in \Gamma_i \setminus j, \varphi_k > \varphi_i} \frac{-1}{d_{ij}^{p_1}}.
\end{aligned}$$

Similar to σ_-, $\sigma_+ \to 0$ as $w \to +\infty$ and $\sigma_+ \to 1$ as $w \to -\infty$ with $t_1, t_2 > 0$ defined as constant scalars. Figure 4.5 shows the graph of σ_+. The first and second terms of equation (4.8) are identical to (4.3). The third term of (4.8) generates a vorticity field around the obstacle such that the strength of the field increases as one approaches the obstacle. Furthermore, in the vicinity of the obstacle where $\varphi_i > \varphi_j$, $\sigma_+(\varphi_{ij}) = 0$, which results in creating a counter-clockwise vorticity field around the obstacle that becomes zero when $\varphi_{ij} < 0$. Similarly, in the vicinity of the obstacle where $\varphi_i < \varphi_j$ and $\theta_i < \theta_j$, $\sigma_- = 0$, which results in creating a clockwise vorticity field around the obstacle that becomes zero when $\theta_{ij} > 0$. Figure 4.6(a) provides a schematic of the region around an obstacle and the respective relations between φ_i, φ_j and θ_i, θ_j. Lastly, the third term is modulated by the scalar function $f(\bar{N}_i) \in [0, 1]$, which is used to scale agent i's obstacle avoidance velocity by only takes into account active neighbors with lower priorities in the descent direction. This approach assigns a higher priority to agents who are not avoiding obstacles and thus ensures collisions do not occur between agents executing obstacle avoidance maneuvers and those who are not. Lastly, we synthesize $f(\bar{N}_i)$ such that $f(\bar{N}_i) = 0$ if $\varphi_{ij} < 0$ and $\theta_{ij} < 0$.

Figure 4.6(b), shows the resulting vector field in the immediate vicinity of a failed robot. The desired boundary, $\partial\mathcal{S}$ is denoted by the solid black line and the failed robot is denoted by the solid circle. The vector fields above and to the lower right of the immobilized robot is generated by the third term of equation (4.8). We note that while the first and second terms in equation (4.8) are always present. However,

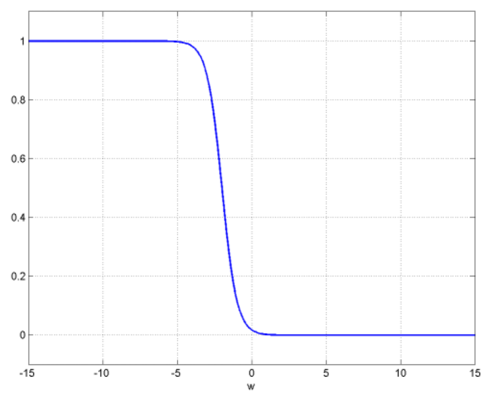


Figure 4.5: Graph of $\sigma+(w)$ with $t_1 = t_2 = 2$.

by construction, $f(\cdot)$ and $g(\cdot)$ tend to zero as a robot approaches the immobilized agent.

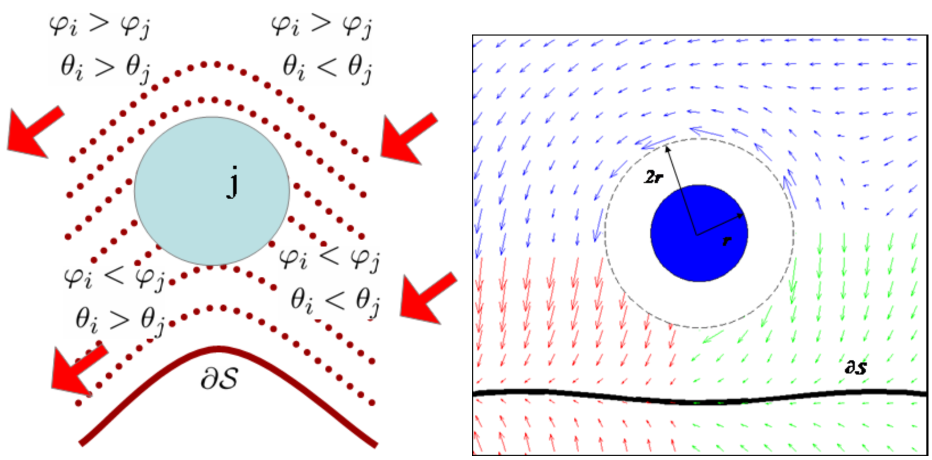
4.3 Safety and Stability Results

In this section, we first consider the stability and covergence properties of the controller given by (4.8) for a group of N robots each with kinematics given by (4.1).

Our first proposition concerns the safety of the system, *e.g.* no collisions can occur in finite time, and non-penetration between any two agents.

Proposition 4.3.1 *Given $\partial\mathcal{S}$, the system of N robots with kinematics (4.1), the feedback control law (4.8) satifies $(q_i - q_j)^T(v_i - v_j) > 0$ for all i,j pairs as $d_{ij} \to 0$.*

Proof: For any i and j, we begin with the scenario where $i \in \Gamma_j$ and $j \in \Gamma_i$ and there are no immobiled neighbors in Γ_i and Γ_j. Thus, the third term of (4.8) is



(a) (b)

Figure 4.6: (a) A sketch of the vicinity of an obstacle denoted by j. The dotted lines denote the level set curves, and the arrows denote the general direction of the vector field given by the first two terms of (4.8). (b) The desired boundary is denoted by the solid black line. The vector field generated by the first two terms of (4.8) is shown in red. The vector field generated by the obstacle avoidance term in equation (4.8) is shown in blue.

zero for both i and j. Then,

$$
\begin{aligned}
(q_i - q_j)^T(v_i - v_j) \\
&= (q_i - q_j)^T\left[(-\nabla_i\varphi_i f(N_i) - (\nabla_i \times \psi_i)g(T_i)) - (-\nabla_j\varphi_j f(N_j) - (\nabla_j \times \psi_j)g(T_j))\right] \\
&= (q_j - q_i)^T\left(\nabla_i\varphi_i f(N_i) + (\nabla_i \times \psi_i)g(T_i)\right) + (q_i - q_j)^T\left(\nabla_j\varphi_j f(N_j)\right. \\
&\quad \left. + (\nabla_j \times \psi_j)g(T_j)\right)
\end{aligned}
$$

Consider the following two cases:

Case I: $\boldsymbol{\varphi_{ij} < 0,\ \theta_{ij} > 0}$ Then, $(q_j - q_i)^T(\nabla_i\varphi_i) > 0$, $(q_j - q_i)^T(\nabla_i \times \psi_i) > 0$, $(q_i - q_j)^T(\nabla_j \times \psi_j) < 0$, and $(q_i - q_j)^T(\nabla_j\varphi_j) < 0$. However, by construction,

$f(N_i) > f(N_j)$ and $g(T_i) > g(T_j)$. Furthermore, as $d_{ij} \to 0$, $f(N_j), g(T_i) \to 0$. Thus,

$$(q_i - q_j)^T (v_i - v_j) = (q_j - q_i)^T \left(\nabla_i \varphi_i f(N_i) + (\nabla_i \times \psi_i) g(T_i)\right) > 0.$$

The same analysis holds when $\varphi_{ij} > 0$ and $\theta_{ij} < 0$ by simply interchanging i and j.

Case II: $\boldsymbol{\varphi_{ij} < 0, \theta_{ij} < 0}$ This results in $(q_j - q_i)^T(\nabla_i \varphi_i) > 0$, $(q_j - q_i)^T(\nabla_i \times \psi_i) < 0$, $(q_i - q_j)^T(\nabla_j \varphi_j) < 0$, $(q_i - q_j)^T(\nabla_j \times \psi_j) > 0$. By construction, $f(N_j) > f(N_i)$, $g(T_i) < g(T_j)$, thus as $d_{ij} \to 0$, $f(N_j), g(T_i) \to 0$, with

$$(q_i - q_j)^T (v_i - v_j) = (q_j - q_i)^T \left(\nabla_i \varphi_i f(N_i)\right) + (q_i - q_j)^T \left(\nabla_j \times \psi_j g(T_j)\right) > 0.$$

The same analysis holds when $\varphi_{ij} > 0$ and $\theta_{ij} > 0$ by simply interchanging i and j.

Next, consider the scenario where $i \in \Gamma_j$ and $j \in \Gamma_i$, but j is immobilized, *i.e.* $\|v_j\| = 0$. Then, as $d_{ij} \to 0$, $f(N_i), g(T_i) \to 0$, resulting in

$$(q_i - q_j)^T (v_i - v_j) = (q_i - q_j)^T \frac{-\nabla_i \times \beta_j}{d_{ij}^{p_2}} h(q_i, q_j)$$

Regardless of whether $\varphi_i > \varphi_j$ or vice versa, by construction $(q_i - q_j)$ is always perpendicular to $\nabla_i \times \beta_j$. Thus, $(q_i - q_j)^T (v_i - v_j) = 0$ when i is actively avoiding j. Once i has successfully avoided j, then $(q_i - q_j)^T (v_i - v_j) > 0$ since i will be moving away from j. This can be seen by setting v_j and following the analysis for Case II if $\varphi_{ij} > 0$ or the analysis for Case I if $\varphi_{ij} < 0$.

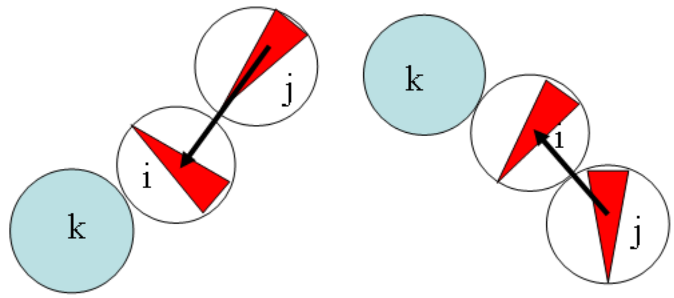
Our next scenario is when $i \in \Gamma_j$ and $j \in \Gamma_i$ and there exist a $k \in \Gamma_i$ but $k \notin \Gamma_j$

with $\|v_k\| = 0$. We begin by consider the case when $\varphi_i > \varphi_k$.

$$
\begin{aligned}
(q_i-q_j)^T&(v_i - v_j) \\
&= (q_j - q_i)^T \left(\nabla_i \varphi_i f(N_i) + \nabla_i \times \psi_i g(T_i) + \frac{\nabla_i \times \beta_k}{d_{ik}^{p_2}} h(q_i, q_j)\right) \\
&- (q_i - q_j)^T \left(\nabla_j \varphi_j f(N_j) + \nabla_j \times \psi_j g(T_j)\right)
\end{aligned}
$$

As i approaches k, $f(N_i), g(T_i) \to 0$ and as $d_{ij} \to 0$, $f(N_j), g(T_j) \to 0$. We note that by construction the magnitude of the obstacle avoidance term increases at a slower rate than the rate at which $f(\cdot), g(\cdot) \to 0$ since $0 \leq p_2 \leq p_1$. If $\varphi_{ik} > 0$ there exists configurations where $\varphi_{ij} < 0$ and $\theta_{ij} < 0$ such that $(q_i - q_j)^T(v_i - v_j) < 0$ as shown in Figure 4.7(a). However, as $d_{ij} \to 0$, $f(\bar{N}_i) \to 0$ and $f(N_j), g(T_j) \to 0$. Under these circumstances, i becomes immobilized with $(q_i - q_j)^T(v_i - v_j) = 0$, and, as such, j will realize $\|v_i\| = 0$ and take the appropriate obstacle avoidance action. This then becomes the scenario discussed previously which results in $(q_i - q_j)^T(v_i - v_j) > 0$. This anomaly, however, does not occur when $\varphi_{ik} < 0$. When $\varphi_{ik} < 0$ and $\varphi_{ij} < 0$, i would move away from j towards the direction of the boundary as it is avoiding the obstacle k. Similarly, if $\varphi_{ij} > 0$ when $\varphi_{ik} < 0$ as shown in Figure 4.7(b), j would move away from i as it heads towards the boundary. As such, the condition $(q_i - q_j)^T(v_i - v_j) > 0$ is always satisfied when $\varphi_{ik} < 0$.

Lastly, we consider the scenario where $i \notin \Gamma_j$ and $j \notin \Gamma_i$, however there exist a k such that $k \in \Gamma_i, \Gamma_j$. If we assume $\varphi_i > \varphi_k$, then $\varphi_k > \varphi_j$, otherwise, $i \in \Gamma_j$ and vice versa. Thus, the only possible configurations would be $\varphi_i > \varphi_k > \varphi_j$ or $\varphi_i < \varphi_k < \varphi_j$ and similarly in the tangential direction. Then, to show that $(q_i - q_j)^T(v_i - v_j) \geq 0$ is equivalent to the proof for Case I. By induction, we can conclude that $(q_i - q_j)^T(v_i - v_j) > 0$ for all i,j pairs. ■



(a) (b)

Figure 4.7: The solid circle denotes the obstacle. The robots are denoted by the circles and the triangles denote the robots headings. The arrow denotes the direction of the vector $(q_i - q_j)$. (a) Configuration where $\varphi_{ik} > 0$, $\varphi_{ij} > 0$, and $\theta_{ij} < 0$ resulting in $(q_i - q_j)^T(v_i - v_j) < 0$. (b) Configuration where $\varphi_{ik} < 0$, $\varphi_{ij} > 0$, and $\theta_{ij} < 0$, but this results in $(q_i - q_j)^T(v_i - v_j) > 0$.

Remark 4.3.2 *This result guarantees collision avoidance between the robots since we can either consider the disk-shaped robot sets to be open or we can assume an implied non-zero safety margin between the actual and modeled boundaries of the robots.*

Our next proposition concerns the liveness of the system. We denote the initial positions of the agents as q_i^0.

Proposition 4.3.3 *For any smooth star shape, $\mathcal{S}$, the system of N robots each with kinematics (4.1), control input (4.3), and initial conditions $s(q_i^0) > s_0$ with $M = 1$, there does not exist a configuration where $q_i \notin \partial\mathcal{S}$ with $\|v_i\| = 0$ for all $i \notin \Pi$.*

Proof: Recall the assumptions from Section 4.2:

- $N < N_{max}$;
- $|\rho_{min}| > R$;
- $\min_{\bar{s} \in [\frac{\pi\rho_0}{3}, L - \frac{\pi\rho_0}{3}]} \|q_0(\bar{s}) - q(\bar{s})\| > R$ for any $q_0(\bar{s}) \in \partial\mathcal{S}$;

- for any $i \in \Pi$, $d(q_i, \partial\mathcal{S}) > (R + r)$;
- for any $i \in \Pi$, $d(q_i, \partial\mathcal{W}) > 3r$; and
- $\min d(\partial\mathcal{S}, \partial\mathcal{W}) > 2Nr$.

Assume there exist is a set of agents $\{k, \ldots, l\}$ such that $\|v_i\| = 0$ for all $i \in \{k, \ldots, l\}$. Let $q_{min} = \arg\min_i \varphi(q_i)$. If $\varphi(q_{min}) \neq 0$, then there must exist a $j \in \Gamma_{min}$ and $j \in \{k, \ldots, l\}$ such that $\varphi_j < \varphi_i$. Since φ_i is the minimum in the set $\{k, \ldots, l\}$, this means q_{min} was stopped because $\varphi_j > \varphi_i$, however this would violate the conditions $|\rho_{min}| > R$ and $\min_{\bar{s} \in [\frac{\pi\rho_0}{3}, L - \frac{\pi\rho_0}{3}]} \|q_0(\bar{s}) - q(\bar{s})\| > R$. Similarly, in the tangential direction, let $q_{max} = \arg\max_i \theta_i$. Since q_{max} has the maximum θ in the set $\{k, \ldots, l\}$, q_{max} was stopped because there exists a $j \in \Gamma_{max}$ such that $\theta_j < \theta_{max}$. However, this would violate the condition $N < N_{max}$. If $q_{min} \in \Pi$, for the set of agents to be immobilized, this would require either $d(q_{min}, \partial\mathcal{S}) <= (R + r)$, $d(q_{min}, \partial\mathcal{W}) <= 3r$, $\min d(\partial\mathcal{S}, \partial\mathcal{W}) < 2Nr$, which violates our assumptions. Thus, there always exists at least one agent whose velocity is non–zero. ■

Our next proposition concerns the convergence of the system to the desired boundary.

Proposition 4.3.4 *For any smooth star shape, $\mathcal{S}$, the system of N robots each with kinematics (4.1), control input (4.3), and initial conditions $s(q_i^0) > s_0$ for all i with no obstacles, the system converges asymptotically to $\partial\mathcal{S}$.*

Proof: To show that the system converges asymptotically to $\partial\mathcal{S}$, we first consider the stability of the proposed controller. Consider the following positive semi-definite function:

$$V(\mathbf{q}) = \sum_i \varphi(q_i). \tag{4.9}$$

Since $\mathcal{D}$ is compact, then by continuity of V, the level sets of φ are compact subsets of $\mathcal{W}$. Then the time derivative of V is given by

$$\begin{aligned} \dot{V} &= \sum_i \nabla\varphi(q_i)^T \dot{q}_i \\ &= \sum_i \nabla_i\varphi(q_i)^T \left(-\nabla_i\varphi_i f(N_i) - (\nabla_i \times \psi) g(T_i)\right) \end{aligned}$$

By construction, $\nabla\varphi$ is orthogonal to $\nabla \times \psi$, therefore the above equation simplifies to

$$\dot{V} = \sum_i -\|\nabla_i\varphi_i\|^2 f_i(N_i). \tag{4.10}$$

Since $f(\cdot) \in [0,1]$, $\dot{V} \leq 0$. Thus, the system N agents with kinematics (4.1) and feedback control (4.8) is stable and approaches the largest invariant set inside $\Omega_I = \{\mathbf{q} \in Q | \dot{V}(\mathbf{q}) = 0\}$.

By Proposition 4.3.3, the closed loop system does not admit stationary points, i.e. points where $\dot{\mathbf{q}} = 0$ as long as $s(q_i) > s_0$ for all $i = 1, \ldots, N$. Since there can be no stationary points, we conclude that $\dot{V} = 0$ in (4.10) if and only if $\nabla_i\varphi_i = 0$ for all i. For a star shape, with φ given by (4.2), $\nabla\varphi = 0$ if and only if $q \in \partial\mathcal{S} \in \mathcal{W}$. Thus, the system of N robots converges asymptotically to $\partial\mathcal{S}$. ■

The above proposition can be extended to include the obstacle avoidance case, *i.e.* the controller given by (4.8) if we assume $|\Pi| = M = 1$.

Proposition 4.3.5 *For any smooth star shape, $\mathcal{S}$, the system of N robots each with kinematics (4.1), control input (4.8), $M = 1$, and initial conditions $s(q_i^0) > s_0$ for all i, the system converges asymptotically to $\partial\mathcal{S}$.*

Proof: We begin by first showing stability. Consider the function given by

(4.9), then for the controller given by (4.8),

$$\dot{V} \;=\; \sum_i \left(-\|\nabla_i \varphi_i\|^2 f_i(N_i) - \nabla_i \varphi_i^T \sum_{j \in \Gamma_i, \|v_j\|=0} \frac{\nabla_i \times \beta_j}{d_{ij}^{p_2}} h(q_i, q_j) \right).$$

Since $f(\cdot) \in [0,1]$, the first term is always less than or equal to zero. The synthesis of $h(q_i, q_j)$ is such that every agent exits the obstacle avoidance mode at a lower value of φ than when it entered the mode. Once out of obstacle avoidance mode, $\dot{V} < 0$ since the avoidance term becomes zero. While a temporary increase in V may occur when agents are in obstacle avoidance mode, by design of $h(\cdot)$ and the fact that $\dot{V} < 0$ when all agents are not in obstacle avoidance mode, this ensures the agents will never encounter the same obstacle at the same value of φ as when they first encountered it. Thus, the system N agents with kinematics (4.1) and feedback control (4.8) is stable and approaches the largest invariant set.

Then by Proposition 4.3.3, there cannot be stationary points such that $\varphi_i \neq 0$ for all i. Thus, given initial conditions $s(q_i^0) > s_0$ for all i and $d(q_j, \partial\mathcal{S}) > (R + r)$, after some finite time, $\dot{V}$ for the system with obstacle is the same as (4.10). As such, for a star shape, with φ given by (4.2), $\nabla\varphi = 0$ if and only if $q \in \partial\mathcal{S}$ and the system converges to $\partial\mathcal{S}$. ∎

4.4 Simulation and Experimental Results

We illustrate the proposed controllers with some simulation and experimental results.

4.4.1 Simulation Results

Figure 4.8 shows a team of 20 robots each with radius of 2 converging towards a star shaped boundary using feedback control law (4.3). The desired boundary is denoted

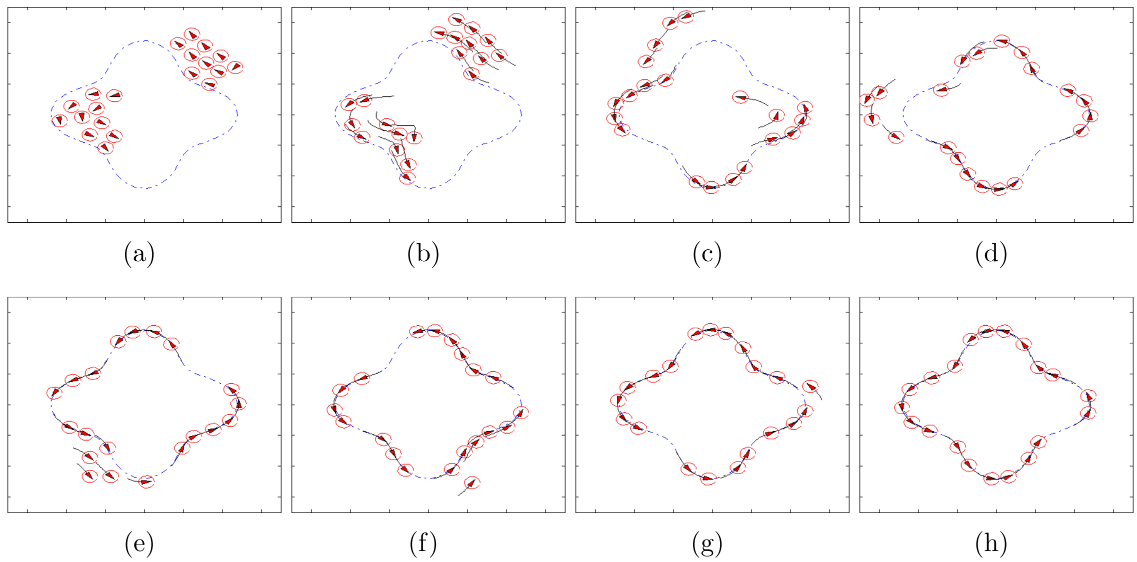


Figure 4.8: A team of 20 robots, each with radius of 2, converging towards a star shaped boundary. The circles denote the size of the robots and the triangles denote their headings. The dotted line denotes the desired boundary and the solid lines denote the agents' trajectories.

by the dotted line while the agents' trajectories are denoted by the solid lines. The circles denote the size of the agents and the triangles denote their headings. Similarly, Figure 4.9 shows a team of 40 robots each with radius of 1 converging to a six lobe star–shaped boundary. Figure 4.10 shows a team of 30 robots converging towards a star–shaped boundary with three immobilized robots denoted by the solid circles. Each robot has a radius of 2 and is using feedback control law (4.8). We have intentionally limited the number of robots in these simulations in order to better display the individual robot trajectories.

4.4.2 Experimental Results

We present two sets of experimental data obtained using the SCARAB indoor ground platform described in Chapter 2, Section 2.1.2. While these robots are non-holonomic, as mentioned in Chapter 2, they can be treated as kinematic robots of

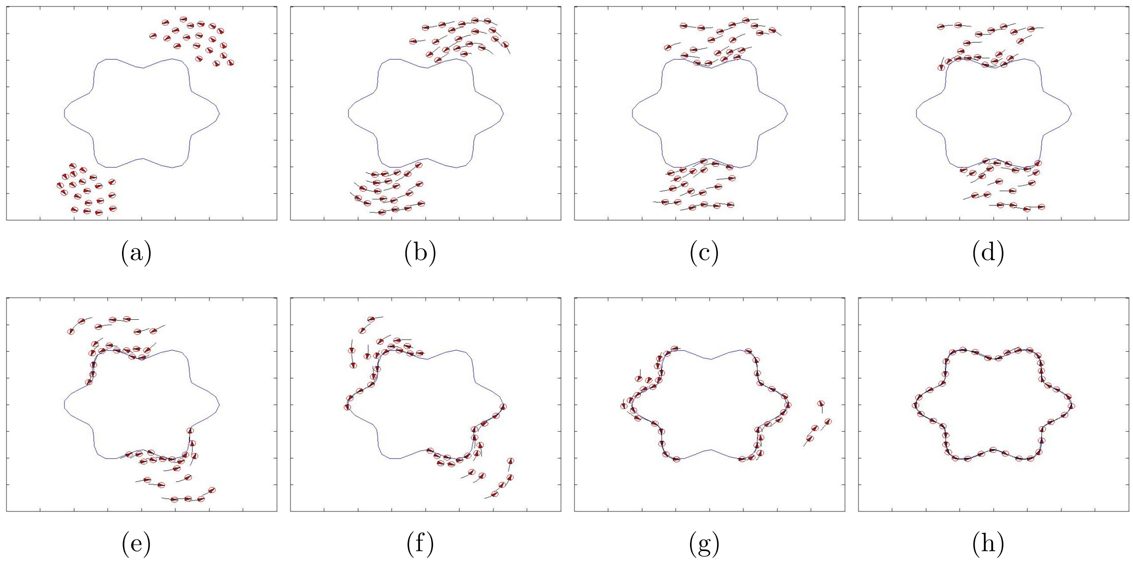


Figure 4.9: A team of 40 robots, each with radius of 1, converging towards a six lobe star-shape boundary. The circles denote the size of the robots and the triangles denotes their headings. The solid line denotes the desired boundary and the solid lines originating from the agents denote their trajectories.

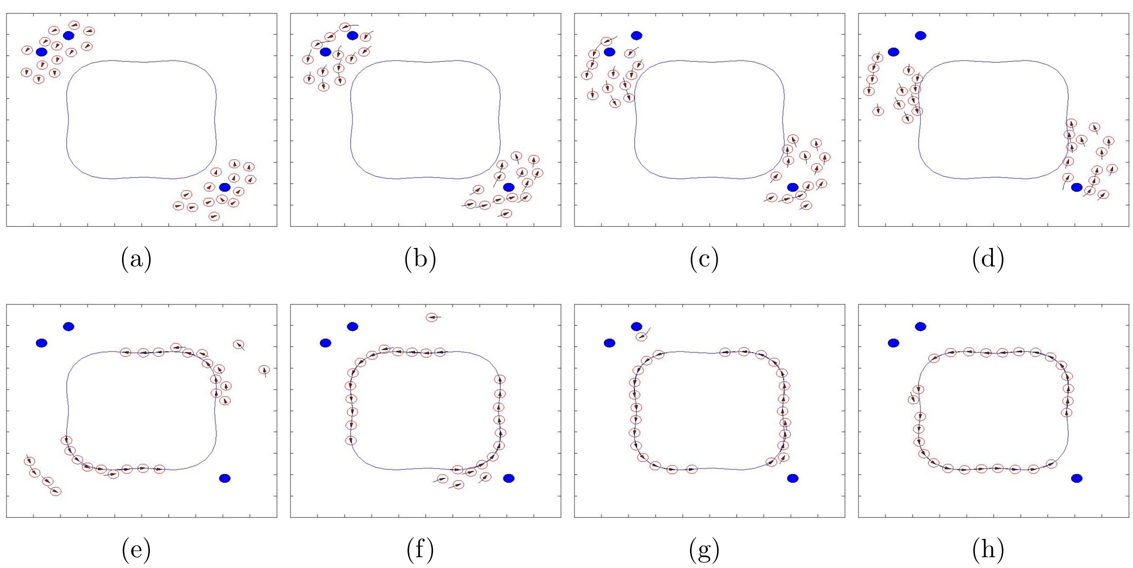
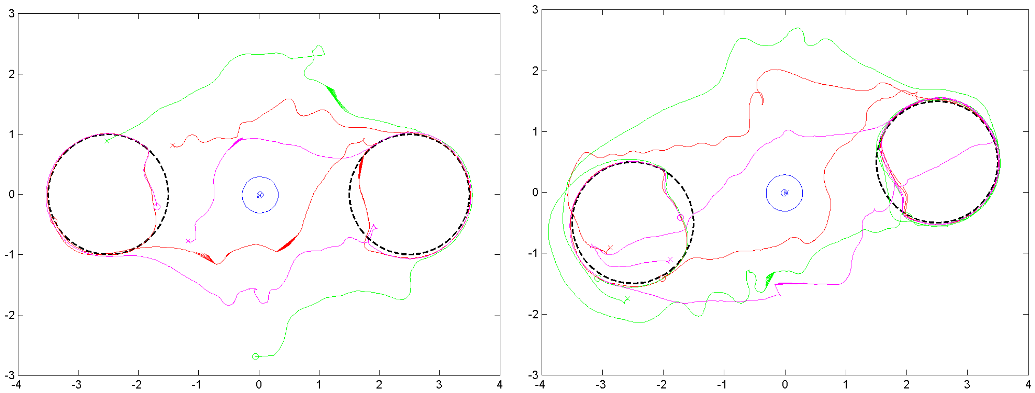


Figure 4.10: A team of 30 robots, each with radius of 2, converging towards a four lobe star-shaped boundary. The circles denote the size of the robots and the triangles denote the heading. The solid circles denote immobilez robots. The boundary is denoted by the solid line and the lines originating from the agents denote their trajectories.



(a) (b)

Figure 4.11: A team of 4 robots, each with radius of 30 cm, converging towards two separate circular boundaries each with a radius of 1 m. The boundaries are denoted by the dotted black lines and the robot trajectories are the solid lines. The robots' initial positions are denoted by $\times$s and their final positions by $\bigcirc$s. The smaller circle in the middle is an obstacle represented by the fourth robot which has been immobilized due to failure.

the form given by (4.1) through feedback linearization. In these experiments, the radius of the circle used to circumscribe our non–holonomic robot in our feedback linearization is 42.5 cm.

Figure 4.11 shows the trajectories of four robots converging onto a circular boundary while avoiding collisions with obstacles in the environment represented by the fourth robot which has been rendered immobile to simulate failure. The fourth robot is denoted by the smaller circle located between the two boundaries in Figure 4.11. Figure 4.11(a) shows the trajectories of the team converging first to a circular boundary centered at $(2.5, 0)$ m and then to a circular boundary centered at $(-2.5, 0)$, each with a radius of 1 m. Similarly, Figure 4.11(b) shows the trajectories of the same team of robots converging first to a circular boundary centered at $(2.5, -0.5)$ m and then to a circular boundary centered at $(-2.5, 0.5)$, each with a radius of 1 m. In these experiments, each agent executed the controller given by (4.8).

Robot positions in these experiments are determined using an overhead camera tracking system with accuracy of less than one centimeter. This information is then relayed to each of the robots via the wireless communication network. Since the tracking system simultaneously tracks the positions of every member in the team, the positions of each robot's neighbors were also communicated to the robot via the wireless network. While this implementation relies on inter–agent communication, our proposed communication–less strategy can be easily achieved since each robot is equipped with a laser range finder.

4.5 Extension

In this section, we describe the synthesis of a decentralized feedback controller that will allow a team of robots to converge and circulate around a desired boundary curve, $\partial\mathcal{S}$, while enabling individual robots to modify its shape navigation function on-the-fly should they encounter an immobile agent in close proximity of $\partial\mathcal{S}$. While (4.8) gives robots the ability to avoid collisions with immobile robots, it may potentially result in undesirable behaviors should a robot fail in close proximity to the boundary. Consider a perimeter surveillance application where $\partial\mathcal{S}$ describes the perimeter of a building of interest. If agent i is immobilized at a location such that $d(q_i, \partial\mathcal{S}) \leq (R+r)$, (4.8) may potentially navigate the agent into the physical building resulting in a collision. Under these circumstances, it makes sense to give robots the ability to treat the immobile robot as part of the object of interest and to redefine the boundary as the convex envelope of $\mathcal{S}$ and the failed agent in real-time.

To achieve this, recall from Section 4.2.2, the boundary of the desired smooth star shape, $\partial\mathcal{S}$, is given by $s(x, y) - s_0 = 0$. Let q_i denote the position of the failed robot such that $d(q_i, \partial\mathcal{S}) \leq R$. Let $\partial\hat{\mathcal{S}}$ denote the new boundary and we will construct the new shape navigation function, $\hat{\varphi}$, such that $\partial\hat{\mathcal{S}}$ is the convex envelope of $\mathcal{S}$ and the failed robot, q_i. Let $\beta_i = \|q - q_i\| - r_i$ and define $\hat{\gamma}$ as

$$\hat{\gamma} = \left(\frac{a}{s(x,y) - b \cdot s_0} + \frac{c}{\beta_i} \right) - d, \tag{4.11}$$

where a, b, c, d are scalar constants and are chosen such that: 1) $d(\partial\mathcal{S}, \partial\hat{\mathcal{S}}) \leq \epsilon_{\mathcal{S}}$ outside a neighborhood of q_i with $\epsilon_{\mathcal{S}}$ being a small positive constant, and 2) $\partial\hat{\mathcal{S}}$ satisfies the assumptions outlined in Section 4.2.2. Then, the new shape navigation function is given by

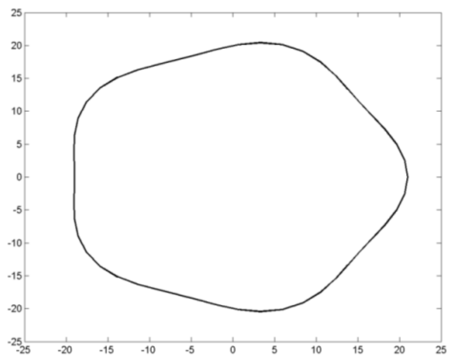
$$\hat{\varphi} = \frac{\hat{\gamma}^2}{[\hat{\gamma}^{2\kappa} + \beta_0]^{\kappa}} \tag{4.12}$$

where $\kappa > 0$ is chosen to be large enough to ensure all minima of $\hat{\varphi}$ lies on $\partial\hat{\mathcal{S}}$ [51]. As mentioned before, to enable the agents to travel along $\partial\hat{\mathcal{S}}$ in a counter-clockwise direction, we define

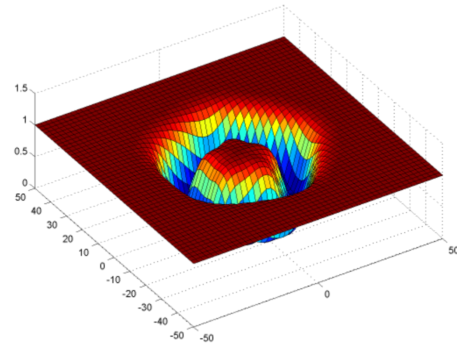
$$\hat{\psi} = \begin{bmatrix} 0 \\ 0 \\ \frac{\hat{\gamma}}{\sqrt{\hat{\gamma}^2 + \beta_0}} \end{bmatrix}.$$

Figure 4.12 shows a star shape boundary with its shape navigation function with and without a robot-sized obstacle in close proximity to $\partial\mathcal{S}$. Figures 4.12(a) and 4.12(b) show the original star shape boundary and its corresponding shape navigation function and Figures 4.12(c) and 4.12(d) show the new star shape boundary and its corresponding shape navigation function generated from the original boundary and the failed robot.

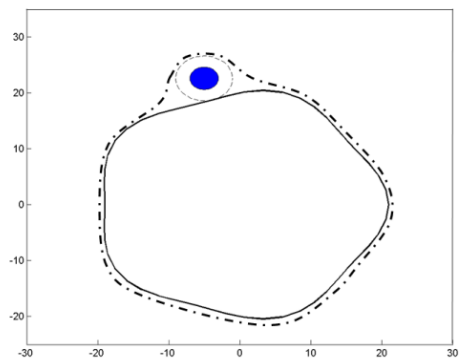
Given, $\partial\mathcal{S}$ and the synthesis methodology for $\partial\hat{\mathcal{S}}$, we proposed the hybrid decentralized controller given in Figure 4.13. In the absence of immobile neighbors, every robot converges to and circulates around $\partial\mathcal{S}$ using decentralized controller I



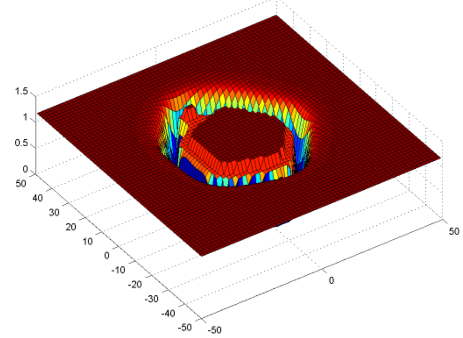
(a)



(b)



(c)



(d)

Figure 4.12: (a) A star shape whose boundary is given by $r-(a_0+b_0\sin(c_0\theta+d_0))=0$ with $a_0=20$, $b_0=1$, $c_0=5$, and $d_0=\pi/2$. (b) The shape navigation function for the boundary given in (a). (c) The new boundary obtained using equation (4.11) with $a=50$, $b=0.8$, $c=21$, and $d=12$. (d) The shape navigation function for the boundary given in (c).

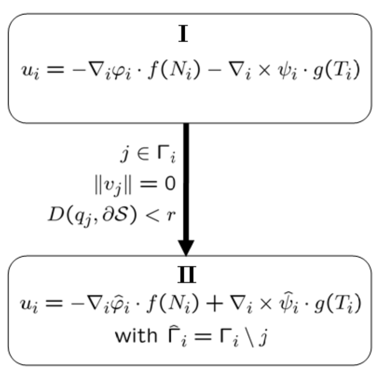


Figure 4.13: Hybrid decentralized controller.

in Figure 4.13, *i.e.* control input (4.3). Each robot then updates its control input to controller II in Figure 4.13 based on the first time it detects the immobile neighbor. Once a failed robot is detected in close proximity of $\partial\mathcal{S}$, the new shape navigation function, $\hat{\varphi}$, is activated and the old shape navigation function, φ, discarded. We note this controller only switches once and the switch occurs the first time the failed robot is detected. Lastly, to ensure avoidance of failed robots away from the desired boundary, the third term in equation (4.8) can be included.

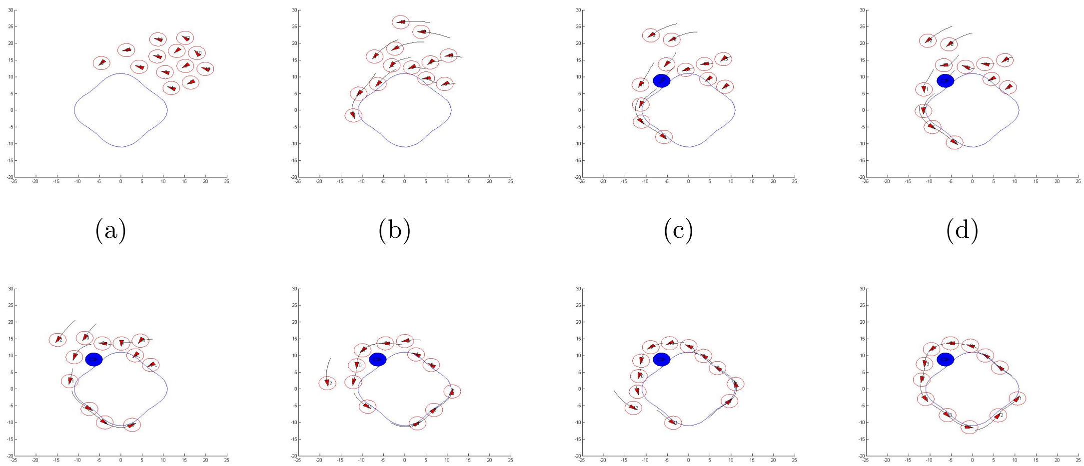
The stability and convergence results in Section 4.3 can be extended for the control input described in Figure 4.13. Since the particular controller switches from one shape navigation function to another and the switch only occurs once, the time derivative for the positive semi–definite function V given by (4.9) is simply given by (4.10), where φ denotes the appropriate shape function. Furthermore, in the event more and more robots fail in close proximity to the boundary, $\partial\hat{\mathcal{S}}$ can be updated incrementally.

4.5.1 Simulations

We illustrate this controller in this simulation with 13 robots converging to the same star–shaped boundary as the one shown in Figure 4.10. As the team begins to approach the boundary, one robot is immobilized in close proximity to the boundary. In this figure the broken robot is denoted by the solid circle.

4.6 Conclusions and Future Work

In this chapter, we have presented an efficient decentralized approach for a team of robots to converge and track the boundary of a desired two-dimensional shape while avoiding collisions. The algorithm can be used to deploy multiple robots to do



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 4.14: A team of 13 robots each with radius of 2, converging towards a star-shaped boundary. Eventually, a robot, denoted by the solid circle, is immobilized in close proximity to the boundary. The circles denote the size of the robots and the triangles denotes the heading. The boundary is denoted by the solid black line, and the solid lines originating from the agents denote the agents' trajectories.

perimeter surveillance or to cordon off hazardous areas. The algorithm is scalable to large number of robots since control inputs solely rely on information obtained from each robot's sensors thus preserving bandwidth for critical data transfers. Additionally, the computational complexity of the decentralized controller for each agent is linear in the number of neighboring agents. The controller was shown to be stable and convergence to the boundary of star shaped sets was established. Moreover, the methodology ensures that collision avoidance is achieved between the robots.

There are many directions for future work. We would like to extend our stability and convergence results in the presence of more than one obstacle. Additionally, we would like to further investigate additional topological requirements on the level set curves of our shape navigation functions to enable the extension of our results to more general two-dimensional patterns. Lastly, we would also like to extend our methodologies to enable tracking of time-varying boundaries.

Chapter 5

Decentralized Controllers for Shape Generation with Robotic Swarms

Similar to the previous chapter, this chapter considers motion control strategies for robotic swarms in applications such as surveillance monitoring of specified areas, surrounding large objects for capture or for manipulation, and cordoning off hazardous regions after chemical spills or biological terrorist attacks. The work presented here is an initial attempt to extend the results presented in the previous chapter to second order systems.

This chapter builds on the results of Chaimowicz *et al.* [17] and Hsieh *et al.* [40] and considers not only the pattern generation problem, but the pattern generation problem with inter–agent constraints. This work is similar in spirit to the works by Belta, Chaimowicz, and Michael [10, 16, 57], where we address the synthesis of decentralized controllers that guarantees the convergence of the team to the boundary of some desired shape as well as the stability of the resulting formation, all the while maintaining inter-agent constraints through local interactions.

We formulate the problem in Section 5.1 and present our methodology in 5.2. The stability and convergence properties of the controller are discussed in Section 5.3 with simulation results presented in Section 5.4. This chapter concludes with Section 5.5.

5.1 Problem Formulation

Assume a swarm of N planar fully actuated robots each with the following dynamics,

$$\dot{q}_i = v_i \tag{5.1a}$$

$$\dot{v}_i = u_i \tag{5.1b}$$

where $q_i = (x_i, y_i)^T$, v_i, and u_i respectively denote the i^{th} agent's position, velocity, and control input. Thus, the robot state is a 4×1 state vector $\mathbf{x}_i = \left[q_i^T v_i^T\right]^T$. We define the configuration space as $Q \subset \mathbb{R}^{2N}$, and the configuration of the swarm of robots as given by $\mathbf{q} = \left[q_1^T \dots q_N^T\right]^T \in Q$. Similarly, the state space vector is given by $\mathbf{x} = \left[\mathbf{x}_1^T \dots \mathbf{x}_N^T\right]^T \in \mathbf{X} \subset \mathbb{R}^{4N}$. In general, we consider all systems of N agents whose individual dynamics can be transformed into (5.1) via some diffeomorphic state transformation.

Our goal is to design control inputs that will drive the group of N robots to the boundary (curve) of a desired smooth, compact set, *i.e.* *shape*, while maintaining inter-robot constraints. This is relevant for applications such as perimeter surveillance or surrounding an object for capture and/or transportation. Thus, the controller synthesis problem for *pattern generation* is to find a controller that can drive the team to the desired boundary while:

[P1] avoiding collisions with other agents, and/or

[P2] maintaining specified proximity constraints with other agents, such as for communication maintenance.

We outline our methodology in the following section.

5.2 Methodology

5.2.1 Assumptions and Definitions

For a given desired shape, $\mathcal{S}$, whose boundary is denoted by $\partial\mathcal{S}$, assume $\partial\mathcal{S}$ is a two dimensional planar curve in an obstacle–free workspace that can be described by an implicit function, $s(x, y) = 0$. In general, we will assume that the boundary curves of interest are regular closed, simple, smooth planar curves enclosing star shaped sets. The regular and simple assumptions are necessary to ensure that the closed curves do not self intersect [29].

In situations where it is important that the team maintains a connected communication network or specific team members remain within a prescribed range of one another to enable grasping of objects for transportation/manipulation, we will further require the team to maintain a desired interconnection topology. We use a *proximity graph*, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, to model the inter-robot constraints, where $\mathcal{V}$ and $\mathcal{E}$ denote the set of vertices and edges of $\mathcal{G}$. Each robot is then represented by a vertex in $\mathcal{V}$ and proximity relations between pairs of robots are represented by the edges in $\mathcal{E}$. As such, for any two robots represented by $a, b \in \mathcal{V}$, we say a and b are adjacent or neighbors, denoted by $a \sim b$, if a is in the neighborhood of b and b is in the neighborhood of a, and as such the edge $(a, b) \in \mathcal{E}$. In our analysis, robot i's neighborhood is defined as the ball given by $\mathcal{B}_i = \{q | \, \|q_i - q\| \leq d\}$, where $d > 0$ denotes the interaction range. For the constraints under consideration, we choose $d = \delta$ for collision avoidance and $d = \Delta$ for proximity maintenance such that $\delta < \Delta$. In practice, whether for collision avoidance or proximity maintenance, this prescribed range can be determined based on the communication and/or sensing hardware, performance requirements within a given environment, and/or experimental results. Thus, for any proximity graph $\mathcal{G}$, the the $N \times N$ adjacency matrix is defined as:

$$A_{ij} = \begin{cases} 1 & \text{if } j \in \Gamma_i \\ 0 & \text{otherwise} \end{cases} .$$

Given a set of inter-agent constraints, we encode the information in a *desired proximity graph*, $\mathcal{G}^d$, such that every inter-agent constraint is represented by an edge. Thus, the graph $\mathcal{G}^d$ represents the desired interconnection topology and we denote its associated adjacency matrix as A^d. We will assume that the desired proximity graph is always a subgraph of the initial proximity graph to ensure the team initializes in a feasible configuration.

Lastly, since our goal is to synthesize decentralized controllers that will allow a team of robots to converge to the boundary while satisfying inter-agent constraints, we note that given $d > 0$, the length of $\partial\mathcal{S}$ naturally imposes an upper bound on the number of robots, denoted by *e.g.* $N_{max} > 0$, that can fit on the boundary. Similarly, the length of the boundary will naturally impose a lower bound, N_{min}, on the number of robots that can effectively cover the desired boundary given a fixed sensing range.[1]

[1]Given an interaction range or a fixed sensing radius $d > 0$, N_{min} and N_{max} can be determined purely geometrically. To accomplish this, begin by selecting an initial point on $\partial\mathcal{S}$ and placing a circle of radius d centered at this point. Next, select another point on $\partial\mathcal{S}$ by going along the boundary in a counter-clockwise direction. Select this next point such that a circle of radius d centered at this second point is tangent to the first circle. By repeating this process, one will be able to generate a sequence of circles centered at points on the boundary that are tangent to

In this work, we are primarily concerned with convergence to the desired boundary and as such we will make the following additional assumptions:

1. $N < N_{max}$;
2. $|\rho_{min}| > \delta$, where ρ_{min} denotes the smallest radius of curvature of $\partial\mathcal{S}$;
3. $\min_{s\in[\frac{\pi\rho_0}{2}, L-\frac{\pi\rho_0}{2}]}\|q_0(s) - q(s)\| > \delta$ for any $q_0(s) \in \partial\mathcal{S}$, where $s \in [0, L]$ denotes the arclength and ρ_0 denotes the radius of curvature at q_0.

Assumption 1 ensures all agents with finite interaction range will be able to converge to the desired boundary while satisfying all constraints. Assumptions 2 and 3 ensure convergence by excluding boundaries with sharp turns and star shapes with narrow passages, *e.g.* hourglass shapes, that may result in robots repelling each other away from the boundary when avoiding collisions. These are similar to those presented in Chapter 4 and as such, we refer the reader to Chapter 4 for the more detailed discussion on them.

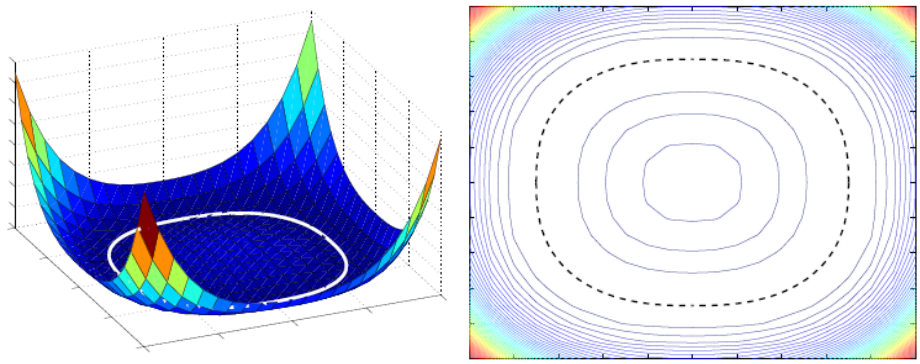
5.2.2 Controller Synthesis

Shape Functions

For a desired shape, $\mathcal{S}$, we define the *shape function*, $f : \mathbb{R}^2 \to \mathbb{R}$, such that f is positive semi-definite in Q and for all $(x, y) \in \partial\mathcal{S}$, i.e. points on the curve $s(x, y) = 0$, $f(x, y) = 0$. In general, for any parameterization $s(x, y) = 0$, $f = (s(x, y))^2$ is a candidate shape function. For star shapes, we choose $f = s^2$ such that

1. $s(x, y)$ is at least twice differentiable on Q; and

one another. Since the boundary and the d are both finite, one can continue this process until one can no longer fit a circle centered at a point on $\partial\mathcal{S}$ without intersecting any of the previous circles. Thus, the number of tangent circles drawn at the end of the process gives N_{max}. A similar procedure can be used to determine N_{min}.



(a) (b)

Figure 5.1: (a) The shape function for $\mathcal{S}$ enclosed by Piet Hein's superellipse, $s(x, y) = |\frac{x}{a}|^r + |\frac{y}{b}|^r - 1$ with $a = b = 7$ and $r = 2.5$. The boundary is shown in white. (b) The level curves for the corresponding shape function with the boundary shown by the dashed line.

2. $s(x, y)$ is polar at some $\hat{q}$, where $\hat{q}$ exists in the interior of $\mathcal{S}$, i.e. $s(x, y)$ has a unique minimum at $\hat{q}$.

Shape functions of this form have level set curves that are consistent with the desired boundary curve, i.e. if the desired boundary curve is convex then so are the level sets of f. This is relevant for stability and convergence analysis. Figures 5.1(a) and 5.1(b) shows the shape function for a compact set enclosed by Piet Hein's superellipse and its corresponding level set curves.

Shape Discrepancy Functions

To determine the peformance of our controller, we define the *shape discrepancy function*, $\phi_{\mathcal{S}} : Q \to \mathbb{R}$, such that $\phi_{\mathcal{S}}$ is real analytic and positive semi-definite, whose zero isocontour is identically the boundary of the desired shape $\mathcal{S}$. While there are many choices for this measure, we use the definition [53, 64, 87, 104]:

$$\phi_{\mathcal{S}}(\mathbf{q}) = \sum_i f(q_i). \tag{5.2}$$

Thus, the shape discrepancy function provides a measure of how close the team is to $\partial\mathcal{S}$.

Controller

For the system of N robots with dynamics given by equation (5.1), consider the following feedback policy for each robot

$$u_i = -\nabla_i f(q_i) - c\, v_i - \sum_{j\; s.t.\; A^d_{ij}=1} \nabla_i g_{ij}(\|q_{ij}\|) \tag{5.3}$$

where $c > 0$ is a constant scalar and $q_{ij} = q_i - q_j$. The first term of (5.3) drives the agent towards the desired curve while the second term adds damping to the system. The function $g_{ij}(\|q_{ij}\|)$ in the third term is an artificial potential function whose gradient models the interactions between each robot and its neighbors in the neighborhood given by $\mathcal{B}_i$. In the remainder of the paper, we will denote $g_{ij}(\|q_{ij}\|)$ as simply g_{ij}. Lastly, ∇_i denotes the partial derivatives with respect to the coordinates of the i^{th} robot.

For collision avoidance, *i.e.* **[P1]** in Section 5.1, consider the following candidate function for g_{ij}:

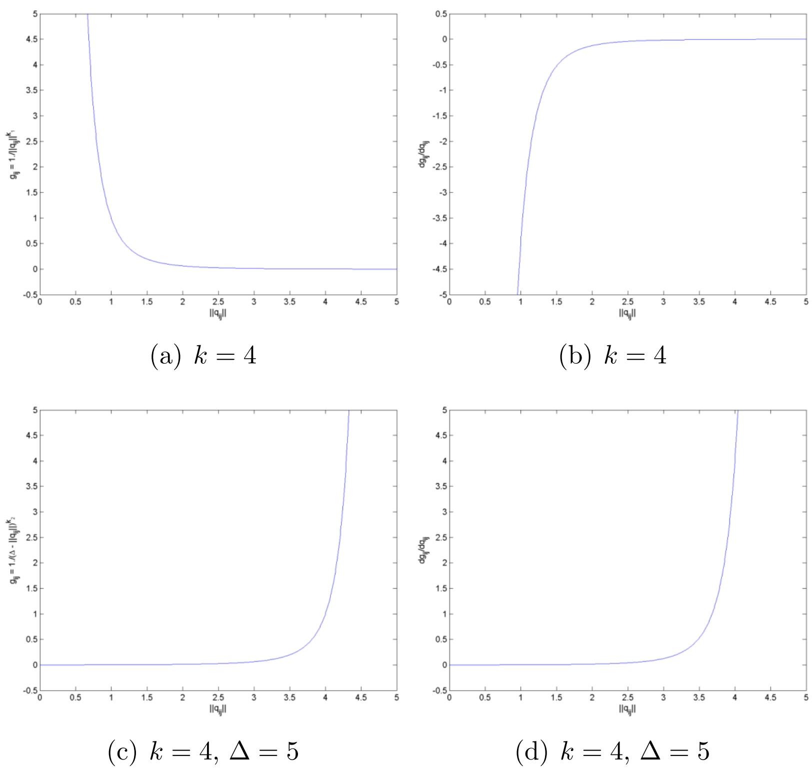
$$g_{ij} = \frac{1}{(\|q_{ij}\|)^{k_1}} \tag{5.4}$$

where the postive even scalar k_1 is chosen such that the interaction forces are negligible when $\|q_{ij}\| > \delta$ and repulsive when $0 < \|q_{ij}\| \leq \delta$, thus ensuring collisions do not occur. Similarly, when the maintenance of a specific proximity graph is required, *i.e.* **[P2]**, consider the following candidate function:

$$g_{ij} = \frac{1}{(\Delta - \|q_{ij}\|)^{k_2}} \tag{5.5}$$

where the positive even scalar k_2 is chosen such that the interaction forces are negligible when $\|q_{ij}\| < \Delta - \epsilon$ and attractive when $\Delta - \epsilon \leq \|q_{ij}\| \leq \Delta$, with $0 < \epsilon < \Delta$.

Figure 5.2 shows candidate functions for g_{ij} with the corresponding $\nabla_i g_{ij}$. It is important to note that both (5.4) and (5.5) result in gradients of the form $\nabla_i g_{ij} = -\frac{dg_{ij}}{d\|q_{ij}\|} q_{ij}$ and thus $\nabla_i g_{ij} = -\nabla_j g_{ij}$. Finally, we note that the desired interconnection



(a) $k = 4$ (b) $k = 4$

(c) $k = 4$, $\Delta = 5$ (d) $k = 4$, $\Delta = 5$

Figure 5.2: (a) A potential function of the form given by equation (5.4) with $k = 4$. (b) Gradient of the potential function shown in (a) with respect to $\|q_{ij}\|$. (c) A potential function of the form given by equation (5.5) with $k = 4$ and $\Delta = 5$. (b) Gradient with respect to $\|q_{ij}\|$ of the potential function shown in (c).

topology, $\mathcal{G}^d$, for problem **[P1]** is a position dependent graph while $\mathcal{G}^d$ for problem **[P2]** is a static graph whose edges are determined by the inter-agent constraints specified *a priori*.

5.3 Analysis

In this section, we study the stability and convergence properties of the controller given by Equation (5.3) for the system of N robots each with dynamics given by Equation (5.1). The system is in equilibrium when $\dot{\mathbf{q}} = \mathbf{0}$ and $\dot{\mathbf{v}} = \mathbf{0}$ or equivalently

$$\begin{aligned} \dot{q}_i &= v_i = 0 \\ \dot{v}_i &= -\nabla f(q_i) - c\, v_i - \sum_{j\ s.t.\ A_{ij}^d = 1} \nabla_i g_{ij} = 0 \end{aligned} \tag{5.6}$$

where $c > 0$ for all $i = 1, \ldots, N$.

Our first lemma shows that the equilibrium points of the N robot system are extremal points of the shape discrepancy function.

Lemma 5.3.1 *For a system of N robots each with dynamics (5.1), shape function, f, and control (5.3), the set of equilibrium points satisfy the necessary condition for the shape discrepancy function to be at an extremum.*

Proof: When the system is in equilibrium, (5.6) simplifies to

$$u_i = -\nabla f(q_i) - \sum_{j\ s.t.\ A_{ij}^d = 1} \nabla_i g_{ij} = 0$$

Recall $\nabla_i f = 2s\nabla_i s$ and as such, when summing over all agents, we obtain

$$\sum_i u_i = \sum_i \left(2s(q_i)\nabla s(q_i) + \sum_{j\ s.t.\ A_{ij}^d = 1} \nabla_i g_{ij} \right) = 0.$$

Since $\nabla_i g_{ij} = -\nabla_j g_{ij}$, the second term sums to zero resulting in

$$\sum_i u_i = \sum_i s(q_i)\nabla s(q_i) = 0.$$

This is identically the necessary condition for the shape discrepancy function to be at an extremum, *i.e.*

$$\nabla\phi_{\mathcal{S}}(\mathbf{q}) = \nabla\sum_i f(q_i) = \sum_i 2s(q_i)\nabla s(q_i) = 0. \tag{5.7}$$

■

The next proposition concerns the stability of the system. To show that our proposed controller is stable, consider the following positive semi-definite function:

$$E(\mathbf{q},\mathbf{v}) = \phi_{\mathcal{S}}(\mathbf{q}) + \sum_i \sum_{j\ s.t.\ A^d_{ij}=1} g_{ij} + \frac{1}{2}\mathbf{v}^T\mathbf{v}. \tag{5.8}$$

One can interpret E as an artificial energy function for the system.

Proposition 5.3.2 *Given the set $\mathcal{S}$ whose boundary is given by the closed, smooth curve $s(x,y) = 0$, consider the system of N robots each with dynamics (5.1) and feedback control law (5.3). For any initial condition given by $\mathbf{x}_0 \in \Omega_0$, where $\Omega_0 = \{\mathbf{x} \in \mathbf{X} | E(\mathbf{q},\mathbf{v}) \leq e_0\}$ with $e_0 > 0$, the system converges to some invariant set, $\Omega_I \subset \Omega_0$, such that the points in Ω_I minimize the shape discrepancy function.*

Proof: We begin by showing the set Ω_0 is compact. Given e_0, the set Ω_0 is closed by continuity of E. To show boundedness, given $E \leq e_0$, then $\left(\phi_{\mathcal{S}} + \sum_i\sum_j g_{ij}\right) \leq e_0$ and $\mathbf{v}^T\mathbf{v} \leq e_0$. Moreover, $\phi_{\mathcal{S}} \leq e_0$, which implies $f(q_i) \leq e_0$ for all $i = 1,\ldots,N$. Since the shape function f is a radially unbounded function, $f(q_i) \leq e_0$ implies bounded $\|q_i\|$ and consequently bounded $\|q_i\|$ when $\sum_i\sum_j g_{ij} \leq e_0$. We note this is not always true in the general case where bounded g_{ij} only implies bounded $\|q_{ij}\|$. Lastly, given $c\mathbf{v}^T\mathbf{v} \leq e_0$, then $\|\mathbf{v}\|$ is bounded by $\sqrt{e_0}/c$. Thus, Ω_0 is compact.

The time derivative of E is given by

$$\dot{E} \;=\; \sum_i \left(\nabla f(q_i)^T \dot{q}_i + v_i^T \dot{v}_i\right) + \sum_i \sum_{j\, s.t.\, A_{ij}^d=1} \nabla_i g_{ij}^T \dot{q}_i$$

Recall $\dot{q}_i = v_i$ and $\dot{v}_i = u_i$ which is given by (5.3). Substituting these into the above equation results in

$$\begin{aligned}
\dot{E} &= \sum_i \nabla f(q_i)^T v_i + \sum_i v_i^T \left(-\nabla f(q_i) - c\, v_i - \sum_{j\, s.t.\, A_{ij}^d=1} \nabla_i g_{ij}\right) + \\
&\quad \sum_i \sum_{j\, s.t.\, A_{ij}^d=1} \nabla_i g_{ij}^T v_i \\
&= \sum_i \nabla f(q_i)^T v_i - \sum_i v_i^T \nabla f(q_i) - c \sum_i v_i^T v_i - \sum_i \sum_{j\, s.t.\, A_{ij}^d=1} v_i^T \nabla_i g_{ij} + \\
&\quad \sum_i \sum_{j\, s.t.\, A_{ij}^d=1} \nabla_i g_{ij}^T v_i \\
&= -c\,\mathbf{v}^T\mathbf{v}.
\end{aligned}$$

Recall from Section 5.2.2, c is a positive scalar constant used to add damping to the system. Then, $\dot{E} = 0$ if and only if $\mathbf{v} = \mathbf{0}$. By LaSalle's Invariance Principle, for any initial condition in Ω_0, the system of N agents with dynamics (5.1), converges asymptotically to the largest invariant set Ω_I, where $\Omega_I = \{\mathbf{x} \in \mathbf{X} | \dot{E}(\mathbf{q}, \mathbf{v}) = 0\}$ and $\Omega_I \subset \Omega_0$.

Furthermore, Ω_I contains all equilibrium points in Ω_0 and as such, by Lemma 5.3.1, satisfy the necessary condition for $\phi_{\mathcal{S}}$ to be at an extremum. ∎

The above proposition states the convergence of the system of N agents to equilibrium points that satisfy the necessary condition for $\phi_{\mathcal{S}}$ to be at an extremum.

However, it does not guarantee the final positions of the robots to be on $\partial\mathcal{S}$. To show this, we begin with the following proposition which shows the set $\Omega_{\mathcal{S}}$, defined as

$$\Omega_{\mathcal{S}} = \{\mathbf{x} | s(q_i) = 0\,,\; v_i = 0, \nabla_i g_{ij} = 0 \quad \forall i\},$$

is a stable subset of Ω_I.

Proposition 5.3.3 *Consider the system of N robots each with dynamics (5.1) and feedback control (5.3). For convex g_{ij}, the set $\Omega_{\mathcal{S}}$ is a stable submanifold and $\Omega_{\mathcal{S}} \subset \Omega_I$.*

Proof: From (5.8) the system's artificial potential energy, $\mathcal{P}$, is given by

$$\mathcal{P} = \phi_{\mathcal{S}} + \sum_i \sum_{j\, s.t.\, A^d_{ij}=1} g_{ij}$$

To show that $\Omega_{\mathcal{S}}$ is a stable submanifold, it suffices to show the Hessian of $\mathcal{P}$, $\mathcal{H}_{\mathcal{P}}$, is positive semi-definite on ∂S. $\mathcal{H}_{\mathcal{P}}$ is given by

$$\mathcal{H}_{\mathcal{P}} = \left(\mathbf{H}^I_\phi + \mathbf{H}^{II}_\phi\right) + \sum_i \sum_{j \in \mathcal{N}_i} \mathbf{H}_{g_{ij}}. \tag{5.9}$$

$\mathbf{H}^I_\phi$ and $\mathbf{H}^{II}_\phi$ are $2N \times 2N$ block diagonal matrices with $2\nabla_i s(q_i) \nabla_i s(q_i)^T$ and $s(q_i)H(q_i)$ respectively on the i^{th} diagonal block. $H(q_i)$ refers to the 2×2 matrix of partial derivatives of $s(q)$ evaluated at q_i. $\mathbf{H}_{g_{ij}}$ is a $2N \times 2N$ matrix with $\frac{\partial^2 g_{ij}}{\partial q_i^2}$ and $\frac{\partial^2 g_{ij}}{\partial q_j^2}$ in the i,i and j,j entries, $\frac{\partial^2 g_{ij}}{\partial q_i \partial q_j}$ in the i,j and j,i entries, and zero everywhere else. We note that for convex g_{ij}, all $\mathbf{H}_{g_{ij}}$ are symmetric positive semi-definite matrices. Since $s(q_i) = 0$, $\mathcal{H}_{\mathcal{P}}$ is the sum of positive semi-definite matrices and therefore positive semi-definite. Thus, the set $\partial\mathcal{S}$ is a stable submanifold. Furthermore, when $\mathbf{q} \in \partial\mathcal{S}$ and $\dot{q}_i = \dot{v}_i = 0$ with $\nabla_i g_{ij} = 0$ for all i, then $\mathbf{x} \in \Omega_{\mathcal{S}} \subset \Omega_I$ by Proposition

5.3.2. ■

We note that Lemma 5.3.1 and Propositions 5.3.2 and 5.3.3 can be extended to include boundaries defined by a collection of disjoint convex regions.

The following propositions concerns the convergence of the team to the desired boundary for different desired interconnection topologies. We begin with the case when no interconnection topology is imposed, *i.e.* no inter-agent constraints.

Proposition 5.3.4 *For any smooth star shape, $\mathcal{S}$, the system of N robots each with dynamics (5.1), control law (5.3) with $g_{ij} = 0$ for all i, j, the system converges to $\Omega_{\mathcal{S}} \equiv \partial\mathcal{S}$ for any initial condition in Ω_0.*

Proof: For any desired circular boundary centered around $\hat{q}$, the boundary can be parameterized by the following implicit function

$$s(q) = \|q - \hat{q}\| - R = 0$$

with $f(q) = s^2(q)$ as the corresponding shape function. The system equilibrium condition is given by (5.6) which simplifies to $\nabla_i f(q_i) = 0 \ \forall i$ when all $g_{ij} = 0$. Furthermore, for any initial condition in Ω_0, the system converges to the invariant set Ω_I. For the circular boundary, $\Omega_I \equiv \Omega_{\mathcal{S}} \equiv \partial\mathcal{S}$. By Proposition 5.3.3, $\partial\mathcal{S}$ is stable and by Proposition 5.3.2, the system converges asymptotically towards the circular boundary.

For any smooth star shape, $\bar{\mathcal{S}}$, there exists a diffeomorphic transformation that maps the boundary of the star shape onto the boundary of the circle given by $s(q)$ and the interior and exterior points of the star boundary to interior and exterior points of the circular boundary [77]. Since such a diffeomorphic map exists, stationary points are diffeomorphically mapped between the circular and the star boundaries. Thus, from Lemma 5.3.1, the system is in equilibrium when $q_i \in \partial\bar{\mathcal{S}}$ for all i. And from Proposition 5.3.2, the system converges asymptotically towards $\partial\bar{\mathcal{S}}$. ■

In the remainder of the section, we prove the convergence of N robots to the desired boundary curve for two desired interconnection topologies. These desired interconnection topologies can be either static proximity graphs for maintaing team cohesion or dynamic position dependent proximity graphs for collision avoidance. In both scenarios, the edges of the proximity graphs will represent the constraints that need to be maintained. We begin with the convergence of the team to $\partial\mathcal{S}$ with g_{ij} given by equation (5.4) and remind the reader of the key assumptions outlined in Section 5.2.1.

1. $N < N_{max}$;
2. $|\rho_{min}| > \delta$;
3. $\min_{s\in[\frac{\pi\rho_0}{2}, L-\frac{\pi\rho_0}{2}]} \|q_0(s) - q(s)\| > \delta$ for any $q_0(s) \in \partial\mathcal{S}$, where $s \in [0, L]$ denotes the arclength and ρ_0 denotes the radius of curvature at q_0.

Proposition 5.3.5 *Given a smooth star-shaped set, $\mathcal{S}$, the system of N robots, each with dynamics (5.1) and control law (5.3), such that N and $\partial\mathcal{S}$ satisfy the above assumptions, then the system with only repulsive interaction forces under arbitrary interconnection topologies and initial conditions in Ω_0, can only be in stable equilibrium if $q_i \in \partial\mathcal{S}$ for all i.*

Proof: Recall that the system equilibrium condition is given by

$$\nabla_i f = -\sum_{j\in\Gamma_i} \nabla_i g_{ij} \qquad \forall i = 1, \ldots, N.$$

Assume N and $\partial\mathcal{S}$ satisfy the above assumptions and the system of N robots is in stable equilibrium such that not all $q_i \in \partial\mathcal{S}$, *i.e.* $s(q_i) \neq 0$ for some i's. Without loss of generality, assume $\mathcal{S}$ is centered about $\hat{q}$ and let $\theta_i \in [0, 2\pi)$ denote the angle between the vector $(q_i - \hat{q})$ and the horizontal axis. Let q_N denote the agent with the maximum value of θ_i in the team. Then $\theta_j < \theta_N$ for all $q_j \in \mathcal{B}_N$.

For every q_i, we choose a body–fixed coordinate frame such that the basis is given by unit vectors in the directions of $(q_i - \hat{q})$ and $(q_i - \hat{q})^{\perp}$. Then for every $\nabla_N g_{Nj}$ for q_N, we denote the component of $\nabla_N g_{Nj}$ in the direction of $(q_N - \hat{q})$ as $(\nabla_N g_{Nj})_{\|}$ and the component of $\nabla_N g_{Nj}$ in the direction of $(q_N - \hat{q})$ as $(\nabla_N g_{Nj})_{\perp}$. Since θ_N is the maximum θ for all N agents, $\sum_{j\ s.t.\ A^d_{Nj}=1} \nabla_N g_{Nj}$ would result in a net force on q_N that would push q_N away from its neighbors in $\mathcal{B}_N$. In other words, the net force from the neighbors of q_N would result in pushing q_N in the general direction of $(q_N - \hat{q})^{\perp}$ such that θ_N would increase. Thus, for q_N to be in stable equilibrium, $\nabla_N f$ must have a component that is equal and opposite in the $(q_N - \hat{q})^{\perp}$ direction. For this to happen, either $\rho_{min} < \delta$ or $\min_{s \in [\frac{\pi\rho_0}{2}, L - \frac{\pi\rho_0}{2}]} \|q_0(s) - q(s)\| < \delta$ which violates our assumption on $\partial\mathcal{S}$ and the level sets of $\partial\mathcal{S}$ since the radius of curvature increases monotonically as one moves away from the boundary. As such, the system can only be in stable equilibrium when $q_i \in \partial\mathcal{S}$ for all i. ∎

Our final proposition proves the convergence of the team to convex boundaries while maintaining a desired proximity graph, where the edges of $\mathcal{G}^d$ are specified *a priori* and g_{ij} are of the form given by equation(5.5).

Proposition 5.3.6 *For any smooth convex shape, $\mathcal{S}$, the system of $N < N_{max}$ robots with dynamics (5.1), control law (5.3), a tree $\mathcal{G}^d$ where the edges represent attractive forces, and initial conditions in Ω_0, can only be in stable equilibrium if $q_i \in \partial\mathcal{S}$ for all i.*

Proof: Assume the system is in equilibrium such that $\mathbf{q} \notin \partial\mathcal{S}$. The equilibrium condition is given by:

$$\nabla_i f = -\sum_{j \in N_i} \nabla_i g_{ij} \qquad \forall i = 1, \dots, N. \tag{5.10}$$

Assume the system of N robots is in stable equilibrium such that not all $q_i \in \partial\mathcal{S}$, *i.e.* $s(q_i) \neq 0$ for some i's. Denote the level set of f evaluated at q_i as $\hat{s}_{q_i}$. Since $\partial\mathcal{S}$ is convex, the level sets of f are also convex. Furthermore, for any $q_i \in \mathbb{R}^2$, $\hat{s}_{q_i}$ lies entirely to one side of the tangent line defined by $\nabla_i f^{\perp}$. Additionally, since the level sets do not intersect, given q_i and q_j such that $s(q_i) > 0$ and $s(q_j) > 0$, $s(q_j) > s(q_i)$ implies q_j lies outside the level set $\hat{s}_{q_i}$. Similarly, given q_i and q_j such that $s(q_j) > s(q_i)$ implies q_i lies outside the level set $\hat{s}_{q_j}$. Since the system is in stable equilibrium, for every q_i there exists a q_j such that $A_{ij} = 1$, $f(q_j) > f(q_i)$, and q_j lies in the halfplane, defined by $\nabla_i f^{\perp}$, that does not contain s_{q_i}. Define $q_N = \arg\max_i f(q_i)$.

If $s(q_N) > 0$, then for q_N to be in stable equilibrium, there must exist a q_k such that $A_{Nk} = 1$, $f(q_k) > f(q_N)$, and q_k lies in the halfplane that does not contain s_{q_N}. This contradicts the definition of q_N and thus the system cannot be in equilibrium.

If $s(q_N) < 0$, we must show that the equilibrium configuration cannot be a stable one if the desired interconnection topology is a tree. Recall a tree is a graph with no loops, *i.e.* paths whose start and end nodes are identical. Thus, for any tree, there exists at least one node or one edge whose removal results in creating two or more disconnected graphs from the original tree. Let q_d denote the node with the maximum connectivity in the tree. Then, any perturbation of q_d and its associated neighbors would result in breaking the equality condition for q_d given by (5.10). Since $\nabla_i g_{ij} = -\nabla_j g_{ij}$, this would result in breaking the equilibrium condition for the team and therefore, the only stable configuration is one where all $q_i \in \partial\mathcal{S}$.

If $s(q_N) = 0$, this implies $s(q_i) = 0$ for all $i \in \{1, \ldots, N-1\}$. Thus, equilibrium can only be reached when $q_i \in \partial\mathcal{S}$ for all i.

To show that $\nabla_i g_{ij} = 0$ for all i, j pairs, again consider the equilibrium condition of any leaf vertex q_u in $\mathcal{G}^d$ given by

$$\nabla_u f = -\nabla_u g_{uv} \tag{5.11}$$

where v denotes a neighbor of q_u such that $A_{uv} = 1$. Since the only equilibrium is when all q_i are on $\partial\mathcal{S}$, $\nabla_u f = 0$ and thus $\nabla_u g_{uv} = 0$ for all leaf vertices. Further, since $\nabla_i g_{ij} = -\nabla_j g_{ij}$, then $\nabla_v g_{uv} = 0$ for every neighbor q_v of each leaf vertex q_u.

Denote $\mathcal{V}_1$ as the set of leaf vertices of $\mathcal{G}^d$ and consider the subgraph $\mathcal{G}_1^d = \mathcal{G}^d \setminus \mathcal{V}_1$. Since $\mathcal{G}^d$ is a tree, $\mathcal{G}_1^d$ is also a tree. Then $\nabla_v g_{uv} = 0$ for each neighbor q_v of every $q_u \in \mathcal{V}_1$ which implies that every leaf vertex q_m of $\mathcal{G}_1^d$, must also have equilibrium conditions of the form given by equation (5.11). Thus, on the boundary, for each neighbor q_n of q_m, $\nabla_m g_{mn} = -\nabla_n g_{mn} = 0$. By induction, we can conclude that $\nabla_i g_{ij} = 0$ for all i, j pairs. ■

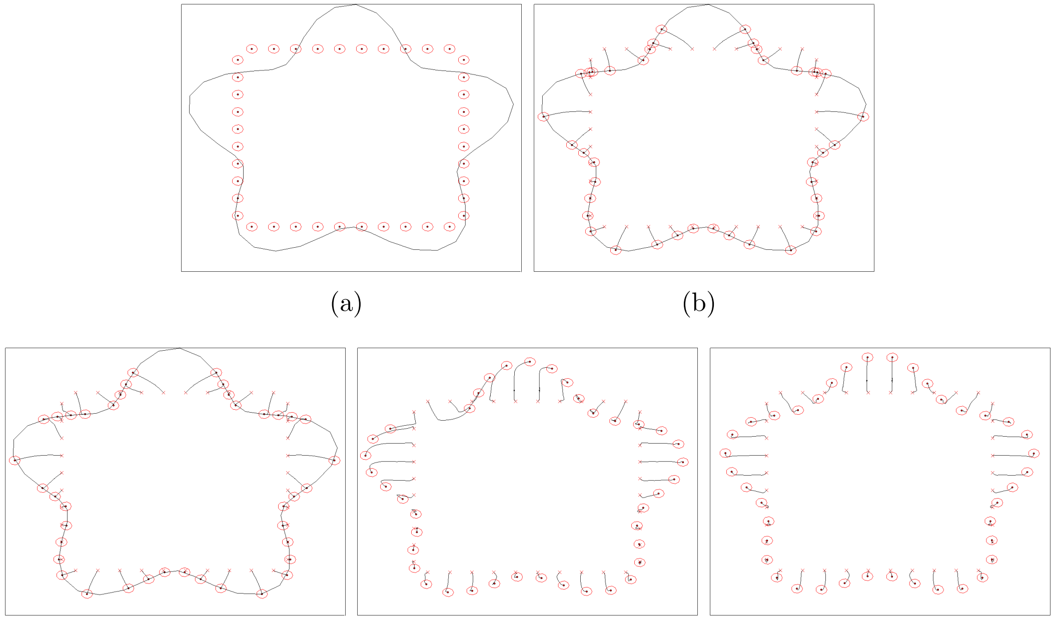
The above proof can be extended to show convergence to star shaped boundaries if we require the largest radius of curvature of $\partial\mathcal{S}$, which we denote by ρ_{max}, to satisfy the condition $\rho_{max} < \Delta$. Furthermore, the above proof can be extended for arbitrary $\mathcal{G}^d$, however the inter-agent constraints are not guaranteed to be satisfied, i.e. $\nabla_i g_{ij} \neq 0$.

5.4 Simulations

We illustrate the algorithm presented in the previous sections with some simulation results. We begin with the case when $\mathcal{S}$ is star and consider teams consisting of approximately 40 robots to demonstrate the scalability of the algorithm. Figure 5.3 shows the initial and final positions and the trajectories of the team for: i) no interactions, $g_{ij} = 0$; ii) collision avoidance only and g_{ij} given by equation (5.4); iii) proximity maintenance only with a path graph as the desired proximity graph, *i.e.* robot i maintains constraints with robots $i-1$ and $i+1$ and no constraints between robots 1 and N, and g_{ij} given by equation (5.5); and iv) collision avoidance and proximity maintenance with a path graph as the the desired proximity graph and g_{ij} given by the sum of equations (5.4) and (5.5). In these results, we chose $\delta = 2$, $\Delta = 10$, and $k_1 = k_2 = 4$. Note the difference in the final positions resulting from the different g_{ij} choices.

In this next simulation result, $\partial\mathcal{S}$ is given by the boundaries of a collection of disjoint convex regions. We limit the number of robots in these results to better display the individual robot trajectories. Figures 5.4(a) and 5.4(b) show the individual trajectories for a group of 15 robots. Figure 5.4(a) shows the individual trajectories when robots only consider collision avoidance. Figure 5.4(b) shows the trajectories when a path proximity graph was imposed on the team for the same initial positions. The final positions for the team were solely dependent on the initial positions and robots were not pre-assigned specific boundaries.

We note that when synthesizing patterns composed of disjoint unions of complex boundaries, the ability of the team to align themselves along the desired boundaries often depends on the initial positions of the robots since the convergence results obtained in Section 5.3 do not extend to these patterns.



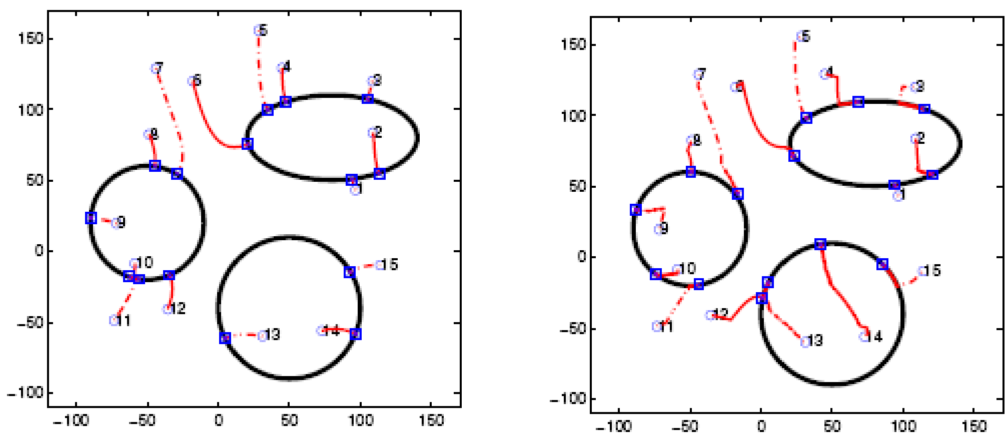
(a) (b)

(c) (d) (e)

Figure 5.3: A 40-robot team converging to a star boundary using the control law (5.3) with $\delta = 2$, $\Delta = 10$, $k_1 = k_4 = 4$. The boundary is denoted by the black solid line in figure (a). The solid circles represent the robots and the empty circles denote the circular region of radius δ around the robot. Robot trajectories are the solid lines connecting the solid circles and the $\times$'s are used to denote the initial positions. (a) Initial position of the team with respect to the desired boundary. (b) Trajectories of the robots when $g_{ij} = 0$ for all i and j. (c) Trajectories of the robots with g_{ij} given by (5.4), *i.e.* collision avoidance only. (d) Trajectories of the robots with g_{ij} given by (5.5), *i.e.* proximity maintenance. The desired proximity graph is a path graph. (e) Trajectories of the robots when g_{ij} is the sum of (5.4) and (5.5), *i.e.* collision avoidance and proximity maintenance.

5.5 Conclusions

We have presented decentralized controllers for generating formations that conform to specified two dimensional patterns with constraints on proximity. These controllers can be used to deploy multiple robots to surround buildings or fenced off areas, or to self-assemble robots to build a two dimensional structure. The algorithm was shown to be stable and convergence to the boundary was established for star shapes in both the absence and the presence of inter-agent interactions.



(a) (b)

Figure 5.4: (a) Trajectories for a group of 15 robots converging to multiple boundaries with g_{ij} given by (5.4). The boundaries are shown in black, and the initial positions are denoted by $\bigcirc$ with final positions denoted by $\boxtimes$. (b) Trajectories for the same group of robots, with the same initial conditions, converging to the same boundaries with g_{ij} given by(5.5).

Chapter 6

Concluding Remarks

Communication in multi–robot research has historically been used to achieve finer fidelity control and/or perception. While the ubiquity of wireless technology has enabled us to outfit every robot with the necessary networking tools, it also introduces additional challenges in terms of how best to coordinate the team to maintain the quality of the communication medium while executing various tasks. This is mostly due to the difficulty in properly predicting and modeling the radio propagation mechanisms in the various environments we wish to operate in. This suggests rather than treating communication solely as a means to improve control and perception, it is necessary to endow robots with the capability to perceive changes in their abilities to communicate and respond to them accordingly. As such, the problem of coordinating multiple autonomous agents to cooperatively achieve a common goal is inherently a problem at the intersection of communication, control, and perception.

6.1 Contributions

The contributions of this thesis are two fold:

Experimentally validated strategies for maintaining communication links

We presented a paradigm and algorithms for the deployment of a team of mobile robots with strategies for maintaining point-to-point communication links based on signal strength and/or estimated available bandwidth. First, a methodology is proposed for the automated construction of a radio signal strength map for a partially known urban environment. We showed how information gleaned from a radio signal strength map can be used to plan future multi-robot tasks. Coupling this with low-level reactive controllers that give robots the ability to monitor their communication links, in particular signal strength measreuments as well as available data throughput, we can ensure the maintenance of overall network quality during mission deployment. Experimental results in three representative urban environments were presented [38, 39, 41, 44].

This work highlights the limitations of the proposed strategies. By design, our strategies favor constraint satisfaction above reaching goal positions. While such a strategy allows for real–time status updates from the team and enable the human operator to deploy additional robots or request a reconfiguration of the whole team should the original task prove unachievable, the reliance on such contingency strategies may result in serious limitations when deployed in the field. Such considerations motivated the second contribution of this work: communication–less motion control strategies.

Provaby correct methods for scalable motion synthesis for teams of robots

Here, we presented two provably correct methods to synthesis communication-less motion coordination strategies to enable a team of robots to converge onto a desired boundary curve. First, potential functions, *i.e.* shape navigation functions, were synthesized such that all minima within the workspace lie exactly on the desired boundary curve. Decentralized feedback controllers were then derived from these shape functions to drive the team towards the boundary. These controllers are decentralized in the sense that agents do not need to communicate each other's state information. Rather, we assume each robot is equipped with range and bearing sensors to enable it to infer neighbors' relative distance and bearing information.

For our first method, collision avoidance was achieved by modulating each agent's descent and orbiting speeds following a local prioritization scheme. Obstacle avoidance was then achieved by introducing a rotational component around the obstacle. When an obstacle was in close proximity to the boundary, a methodology was presented to synthesize a new shape navigation function to achieve obstacle avoidance. In our second method, we extended the shape convergence controller to a system of particles with second order dynamics. Here, collision avoidance and proximity maintenance among neighbors were achieved through the use of inter–agent artificial potential functions.

Both these approaches are efficient and can be used to deploy large robotic teams to perform perimeter surveillance tasks or to cordon off hazardous areas. Our algorithms are scalable to large number of robots since control inputs rely solely on local information obtained from each robot's sensors, thus preserving bandwidth for critical data transfers. The controllers were shown to be stable and convergence to the boundary of star shaped sets was established. Lastly, the methodology ensures that collision and obstacle avoidance are achieved between kinematic robots of finite size [40, 42, 43].

6.2 Future Work

There are a few directions for future work that is of particular interest. First we would like to incorporate some machine learning techniques to the mapping and exploration of radio signal strength data for a given environment. By incorporating machine learning with the low-level reactive controllers capable of responding to changes in signal strength and/or estimated available bandwidth, it may be possible to concurrently map the propagation characteristics while executing other activities. Additionally, this may enable the team to find the most optimal configuration for any given task at all times.

Next, we would like to extend our convergence proofs in the obstacle avoidance case presented in Chapter 4 to a finite number of obstacles. Furthermore, we would like to investigate additional topological properties to ensure the proper synthesis of shape navigation functions when robots fail in close proximity to the desired boundary. This can in turn provide some ideas for real–time estimation the shape navigation functions.

Another direction for future work is the extension of the pattern generation controller discussed in Chapter 5 to two dimensional surfaces. This is relevant in applications such as self-assembly or surface inspection, such as airplanes or ships. A similar direction is to extend the results in Chapters 4 and 5 to time-varying boundaries which is relevant in applications like containing oil-spills and monitoring changing environmental boundaries. Finally, we would like to extend the convergence results discussed in Chapters 4 and 5 to boundaries given by the union of disjoint convex sets.

In conclusion, we acknowledge the need to implement sensing on individual robots to obtain local state information. It may not be reasonable to expect small, resource-constrained robots to be able to sense their individual states. However, it is difficult to get robots to perform tasks like pattern generation in a fixed coordinate frame without a hardware (or software) solution to the localization problem. The important point is that the robots need only local state information and the algorithm is completely decentralized.

Bibliography

[1] Carlos Aguero, Pedro de-las Heras-Quiros, Jose M. Canas, and Vicente Matallan. Pera: Ad-hoc routing protocol for mobile robots. In *Proceedings of the 11th International Conference on Advanced Robotics*, pages 694–702, Coimbra, Portugal, July 2003.

[2] O. Albayrak. *Line and Circle Formation of Distributed Autonomous Mobile Robots with Limited Sensor Range*. PhD thesis, Naval Postgraduate School, Monterey, CA, June 1996.

[3] Amarss. Amarss - networked minirhizotron planning and initial deployment. http://research.cens.ucla.edu/, 2006.

[4] S. O. Anderson, R. Simmons, and D. Goldberg. Maintaining line of sight communications networks between planetray rovers. In *Proceedings of the 2003 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, pages 2266–2272, Las Vegas, NV, October 2003.

[5] R. Arkin and J. Diaz. Line-of-sight constrained eploration for reactive multiagent robotic teams. In *AMC 7th International Workshop on Advanced Motion Control*, 2002.

[6] P. Basu and J. Redi. Coordinated flocking of uavs for improved connectivity of mobile ground nodes. In *Proceedings of IEEE MILCOM 2004*, 2004.

[7] P. Basu and J. Redi. Movement control algorithms for realization of fault-tolerant ad-hoc robot networks. *IEEE Network*, 18(4), July-August 2004.

[8] George Bekey. *Autonomous Robots: From Biological Inspiration to Implementation and Control*. MIT Press, 2005.

[9] George Bekey, Robert Ambrose, Vijay Kumar, Art Sanderson, Brian Wilcox, and Yuan Zheng. Wtec panel report on international assessment of research and development in robotics. `http://www.wtec.org/robotics/`, January 2006.

[10] Calin Belta, Guilherme A. S. Pereira, and Vijay Kumar. Abstraction and control of swarms of robots. In *International Symposium of Robotics Research'03*, 2003.

[11] A. Bertozzi, Mathieu Kemp, and Daniel Marthaler. Determining environmental boundaries: Asynchronous communication and physical scales. In *Cooperative Control*, 2004.

[12] Jon Bredin, Erik Demaine, Mohammad Hajiaghayi, and Daniela Rus. Deploying sensor nets with guaranteed capacity and fault tolerance. In *Proceedings of Mobihoc 2005*, Urbana-Champaign, Illinois, May 2005.

[13] N. F. Britton, N. R. Franks, S. C. Pratt, and T. D. Seeley. Deciding on a new home: how do honeybees agree? *Proc R Soc Lond B*, 269, 2002.

[14] L. Chaimowiczz, A. Cowley, D. Gomez-Ibanez, B. Grocholsky, M. A. Hsieh, H. Hsu, J. F. Keller, V. Kumar, R. Swaminathan, and C. J. Taylor. Deploying

air-ground multi-robot teams in urban environments. In *Proceedings of the 2005 International Workshop on Multi-Robot Systems*, 2005.

[15] L. Chaimowicz, B. Grocholsky, J.F. Keller, V. Kumar, and C.J. Taylor. Experiments in multirobot air-ground coordination. In *Proc. IEEE International Conference on Robotics and Automation*, New Orleans, LA, April 2004.

[16] L. Chaimowicz and V. Kumar. Aerial sheperds: Coordination among uavs and swarms of robots. In *Proceedings of the 7th International Symposium on Distributed Autonomous Robotic Systems (DARS2004)*, pages 231 – 240, 2004.

[17] L. Chaimowicz, N. Michael, and V. Kumar. Controlling swarms of robots using interpolated implicit functions. In *Proceedings of the 2005 International Conference on Robotics and Automation (ICRA05)*, Barcelona, Spain, 2005.

[18] D. E. Chang, S. Shadden, J. E. Marsden, and R. Olfati-Saber. Collision avoidance for multiple agent systems. In *Proc. IEEE Conference on Decision and Control*, Maui, Hawaii, 2003.

[19] William Cook. Traveling salesman homepage. http://www.tsp.gatech.edu/.

[20] Peter Corke, Ronald Peterson, and Daniela Rus. Networked robots: Flying robot navigation using a sensor net. In *Proceedings of the Eleventh International Symposium of Robotics Research (ISRR)*, Springer Tracts on Advanced Robotics (STAR). Springer-Verlag, October 2003.

[21] T.H. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction to Algorithms*. MIT Press, 1990.

[22] Nikolaus Correll and Alcherio Martinoli. System identification of self-organizing robotic swarms. In *Proc. 8th Int. Symp. on Distributed Autonomous Robotic Systems (DARS) 2006*, pages 31–40, Rio de Janeiro, Brazil, 2006.

[23] Nikolaus Correll, Samuel Rutishauser, and Alcherio Martinoli. Comparing coordination schemes for miniature robotic swarms: A case study in boundary coverage of regular structures. In *Proc. 10th International Symposium on Experimental Robotics (ISER) 2006*, Rio de Janeiro, Brazil, 2006.

[24] I. D. Couzin, J. Krause, R. James, G. D. Ruxton, and N. R. Franks. Collective memory and spatial sorting in animal groups. *Journal of Theoretical Biology*, 218, 2002.

[25] Jaydev P. Desai, James P. Ostrowski, and Vijay Kumar. Modeling and control of formations of nonholonomic mobile robots. *IEEE Transactions on Robotics and Automation*, 17(6):905–908, 2001.

[26] D. V. Dimarogonas, S. G. Loizou, K. J. Kyriakopoulos, and M. M. Zavlanos. Decentralized feedback stabilization and collision avoidance of multiple agents. Technical Report 01-04, NTUA, Athens, Greece, 2004.

[27] Dimos V. Dimarogonas and Kostas J. Kyriakopoulos. A feedback control scheme for multiple independent dynamic non-point agents. *International Journal of Control*, 79(12):1613–1623, December 2006.

[28] Dimos V. Dimarogonas and Kostas J. Kyriakopoulos. Decentralized navigation functions for multiple robotic agents with limited sensing capabilities. *Journal of Intelligent and Robotic Systems*, 48(3):411–433, March 2007.

[29] M. P. do Carmo. *Differential Geometry of Curves and Surfaces.* Prentic-Hall Inc., 1976.

[30] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes. Experiments in sensing and communication for robot convoy navigation. In *Proceedings of the 2003 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, pages 268–273, 1995.

[31] Angela Feistel and Slawomir Stanczak. Dynamic resource allocation in wireless ad-hoc networks based on QS-CDMA . In *The 16th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Berlin, September 2005. accepted.

[32] R. Fierro, P. Song, A. Das, and V. Kumar. Cooperative control of robot formations. In R. Murphey and P. Paradalos, editors, *Cooperative Control and Optimization: Series on Applied Optimization*, pages 79–93. Kluwer Academic Press, 2002.

[33] J. Fink, N. Michael, and V. Kumar. Composition of vector fields for multi-robot manipulation via caging. In *Proc. 2007 Robotics: Science and Systems III*, Atlanta, GA, June 2007.

[34] Anita M. Flynn. Gnat robots: a low-intelligence low-cost approach. In *Technical Digest. IEEE Solid-State Sensor and Actuator Workshop*, pages 63 – 66, 6-9 June 1988.

[35] B. Grocholsky, S. Bayraktar, V. Kumar, C.J. Taylor, and G. Pappas. Synergies in feature localization by air-ground robot teams. In *9th International Symposium on Experimental Robotics (ISER'04)*, 2004.

[36] B. Grocholsky, J. F. Keller, V. Kumar, and G. Pappas. Cooperative air and ground surveillance: A scalable approach to the detection and localization of targets by a network of uavs and ugvs. *IEEE Robotics & Automation Magazine*, Sept. 2006.

[37] A. Howard, S. Siddiqi, and G. S. Sukhatme. An experimental study of localization using wireless ethernet. In *4th International Conference on Field and Service Robotics*, July 2003.

[38] M. A. Hsieh, A. Cowley, J. F. Keller, L. Chaimowicz, B. Grocholsky, V. Kumar, C. J. Talyor, Y. Endo, R. Arkin, B. Jung, D. Wolf, G. Sukhatme, and D. C. MacKenzie. Adaptive teams of autonomous aerial and ground robots for situational awareness. *Accepted to the Journal of Field Robotics*, To Appear.

[39] M. A. Hsieh, A. Cowley, V. Kumar, and C. J. Taylor. Towards the deployment of a mobile robot network with end-to-end performance guarantees. In *International Conference on Robotics and Automation (ICRA) 2006*, Orlando, FL, April 2006.

[40] M. A. Hsieh and V. Kumar. Pattern generation with multiple robots. In *International Conference on Robotics and Automation (ICRA) 2006*, Orlando, FL, April 2006.

[41] M. A. Hsieh, V. Kumar, and C. J. Taylor. Constructing radio signal strength maps with multiple robots. In *Proc. IEEE International Conference on Robotics and Automation*, New Orleans, LA, April 2004.

[42] M. A. Hsieh, S. Loizou, and V. Kumar. Stabilization of multiple robots on stable orbits via local sensing. In *Proceedings of the International Conference on Robotics and Automation (ICRA) 2007*, Rome, Italy, April 2007.

[43] M. Ani Hsieh, L. Chaimowicz, and V. Kumar. Decentralized controllers for shape generation with robotic swarms. *Submitted to Robotica*, Conditionally Accepted.

[44] M. Ani Hsieh, A. Cowley, V. Kumar, and C. J. Taylor. Maintaining network connectivity and performance in robot teams. *Accepted to the Journal of Field Robotics*, To Appear.

[45] M. Ani Hsieh, S. Loizou, and V. Kumar. Stabilization of multiple robots on stable orbits with obstacle avoidance via local sensing. *Submitted to Transactions on Robotics*, Under Review.

[46] Ali Jadbabaie, J. Lin, and A.S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, July 2003.

[47] M. Jenkin and G. Dudek. The paparazzi problem. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2000*, pages 2042–2047, Takamatsu, Japan, October 2000.

[48] William Kaiser, Gregory Pottie, Mani Srivastava, Gaurav S. Sukhatme, John Villasenor, and Deborah Estrin. Networked infomechanical systems (nims) for ambient intelligence. In *Ambient Intelligence*, pages 83–114. Springer, 2005.

[49] Shahab Kalantar and Uwe R. Zimmer. Distributed shape control of homogeneous swarms of autonomous underwater vehicles. *to appear in Autonomous Robots (intl. journal)*, 2006.

[50] Wesley Kerr and Diana Spears. Robotic simulation of gases for a surveillance task. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2005*, Edmonton, Alberta, Canada, August 2005.

[51] D.E. Koditschek and E. Rimon. Robot navigation functions on manifolds with boundary. *Advances in Applied Math*, 11, 1990.

[52] Keith Kotay, Ron Peterson, and Daniela Rus. Experiments with robots and sensor networks for mapping and navigation. In *International Conference on Field and Service Robotics*, Port Douglas, Australia, July 2005.

[53] Naomi Ehrich Leonard and Edward Fiorelli. Virtual leaders, artificial potentials and coordinated control of groups. In *Proceedings of the IEEE International Conference on Decision and Control*, pages 2968–2973, Orlando, FL, December 2001.

[54] Magnus Lindhe, Karl Henrik Johansson, and Antonio Bicchi. An experimental study of exploiting multipath fading for robot communications. In *Robotics: Systems and Science III*, Atlanta, GA, June 2007.

[55] S. G. Loizou and K. J. Kyriakopoulos. A feedback based multiagent navigation framework. *International Journal of Systems Science*, 2006.

[56] N. Michael, J. Fink, and V. Kumar. Experimental testbed for large teams of cooperating robots and sensors. *IEEE Robotics and Automation Magazine*, 2007. Submitted.

[57] Nathan Michael, Calin Belta, and Vijay Kumar. Controlling three dimensional swarms of robots. In *IEEE International Conference on Robotics and Automation (ICRA) 2006*, pages 964–969, Orlando, FL, April 2006.

[58] Y. Mostofi, T. Chung, R. Murray, and J. Burdick. Communication and sensing trade offs in decentralized mobile sensor networks: A cross-layer design approach. In *4th International Conference on Information Processing in Sensor Networks (IPSN)*, Los Angeles, CA, April 2005.

[59] Y. Mostofi and R. Murray. Effect of time-varying fading channels on the control performance of a mobile sensor node. In *Proc. of IEEE 1st International Conference on Sensor and Ad Hoc Communications and Networks (Secon)*, Santa Clara, CA, October 2004.

[60] Benjamin Nabet and Naomi E. Leonard. Shape control of a multi-agent system using tensegrity structures. In *IFAC Workshop on Lagrangian and Hamiltonian Methods for Nonlinear Control*, 2006.

[61] Klara Nahrstedt, Samarth H. Shah, and Kai Chen. Cross-layer architectures for bandwidth management in wireless networks. In *Resource Management in Wireless Networking*, 2004.

[62] A. Neskovic, N. Neskovic, and G. Paunovic. Modern approaches in modeling of mobile radio systems propagation environment. *IEEE Communications Surveys*, Third Quarter, 2000.

[63] P. Ogren, E. Fiorelli, and N. E. Leonard. Formations with a mission: Stable coordination of vehicle group maneuvers. In *Proc. 15th International Symposium on Mathematical Theory of Networks and Systems*, pages 267–278, August 2002.

[64] Reza Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Trans. on Automatic Control*, 51(3):401–420, March 2006.

[65] D. Paley, N. E. Leonard, and R. J. Sepulchre. Collective motion of self-propelled particles: Stabilizing symmetric formations on closed curves. In *Submitted to 45th IEEE Conference on Decision and Control*, San Diego, CA, 2006.

[66] J. K. Parrish, S. V. Viscido, and D. Grunbaum. Self-organized fish schools: An examination of emergent properties. *Biol. Bull.*, 202, June 2002.

[67] G. A. S. Pereira, A. K. Das, V. Kumar, and M. F. M. Campos. Decentralized motion planning for multiple robots subject to sensing and communication constraints. In *Proceedings of the Second Multi-Robot Systems Workshop*, pages 267–278, Washington, DC, March 2003.

[68] G. A. S. Pereira, V. Kumar, and M. F. M. Campos. Decentralized algorithms for multi-robot manipulation via caging. *The International Journal of Robotics Research (IJRR)*, 23(7), July-August 2004.

[69] L. C. A. Pimenta, A. R. Fonseca, G. A. S. Pereira, R. C. Mesquita, E. J. Silva, W. M. Caminhas, and M. F. M. Campos. On computing complex navigation functions. In *Proc. 2005 Intl. Conf. on Robotics and Automation (ICRA05)*, pages 3463–3468, Barcelona, Spain, April 2005.

[70] L. C. A. Pimenta, A. R. Fonseca, G. A. S. Pereira, R. C. Mesquita, E. J. Silva, W. M. Caminhas, and M. F. M. Campos. Robot navigation based on electrostatic field computation. *IEEE Transactions on Magnetics*, 42(4), April 2006.

[71] L. C. A. Pimenta, M. L. Mendes, R. C. Mesquita, and G. A. S. Pereira. Fluids in electrostatic fields: An analogy for multirobot control. *IEEE Transactions on Magnetics*, 43(4), April 2007.

[72] Bruno S. Pimentel and Mario F. M. Campos. Cooperative communication in ad hoc networked mobile robots. In *Proceedings of the 2003 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, pages 2876–1881, Las Vegas, NV, October 2003.

[73] M. Powers and T. Balch. Value-based communication preservation for mobile robots. In *7th International Symposium on Distributed Autonomous Robotic Systems*, 2004.

[74] S. C. Pratt. Quorum sensing by encounter rates in the ant temnothorax albipennis. *Behavioral Ecology*, 16(2), 2005.

[75] J. Redi, W. Watson, and V. Shurbanov. Energy conserving reception protocols for ad hoc networks. In *Proceedings of MILCOM 2002*, October 2002.

[76] C. W. Reynolds. Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on Computer Graphics (SIGGRAPH'87)*, pages 25–34. ACM Press, 1987.

[77] E. Rimon and D. E. Koditschek. Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation*, 8(5):501–518, October 1992.

[78] Paul E. Rybski, Sascha A. Stoeter, Michael D. Erickson, Maria Gini, Dean F. Hougen, and Nikolaos Papanikolopoulos. A team of robotic agents for surveillance. In *Agents'2000*, Barcelona, Spain, June 2000.

[79] R. Sepulchre, D. Paley, and N. E. Leonard. Stabilization of planar collective motion, part i: All-to-all communication. *Accepted for publication in IEEE Transactions on Automatic Control*, 2005.

[80] R. Sepulchre, D. Paley, and N. E. Leonard. Stabilization of planar collective motion with limited communication. *Submitted to IEEE Transactions on Automatic Control*, 2006.

[81] P. Song and V. Kumar. A potential field based approach to multi-robot manipulation. In *Proc. IEEE Intl. Conf. On Robotics and Automation*, Washington, DC, May 2002.

[82] J. Spletzer and R. Fierro. Optimal positioning strategies for shape changes in robot teams. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 754–759, Barcelona, Spain, April 2005.

[83] Gaurav S. Sukhatme, Amit Dhariwal, Bin Zhang, Carl Oberg, Beth Stauffer, and David A. Caron. The design and development of a wireless robotic networked aquatic microbial observing system. In *Environmental Engineering Science*, 2006.

[84] I. Suzuki and M. Yamashita. Distributed anonymous mobile robots: Formation of geometric patterns. *SIAM Journal on Computing*, 1999.

[85] J. Sweeney, T. J. Brunette, Y. Yang, and R. A. Grupen. Coordinated teams of reactive mobile platforms. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pages 99–304, Washington, DC, 2002.

[86] John D. Sweeney, Roderic A. Grupen, and Prashant Shenoy. Active qos flow maintenance in controlled mobile networks. In *Proceedings of the Fourth International Symposium on Robotics and Automation (ISRA)*, Queretaro, Mexico, August 2004. IEEE.

[87] Herbert G. Tanner, Ali Jadbabaie, and George J. Pappas. Flocking in fixed and switching networks. In *Transactions on Automatic Control*, 2007.

[88] Herbert G. Tanner and Amit Kumar. *Formation Stabilization of Multiple Agents Using Decentralized Navigation Functions*. MIT Press, 2005.

[89] BBN Technologies. *BBN JBox Informal Manual*. BBN Technologies, 2003.

[90] B. J. Thibodeau, A. H. Fagg, and B. N. Levine. Signal strength coordination for cooperative mapping. Technical report, Department of Computer Science, University of Massachusetts, Amherst, 2004.

[91] Massayoshi Tomizuka. Advanced vehicle control systems (avcs) research for automated highway systems in california path. In *Proceedings of the IEEE Vehicle Navigation and Information Systems Conference*, pages PLEN41 – 45, 31 Aug.-2 Sept. 1994.

[92] T. Vicsek, A. Czirok, E. Ben-Jacob, I. Cohen, and O. Shochet. Novel type of phase transition in a system of self-driven particles. *Physical Review Letters*, 75(6):1226–9, 1995.

[93] A. R. Wagner and R. C. Arkin. Communication-sensitive multi-robot reconnaissance. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2480–2487, 2004.

[94] Wikipedia. Asimov. `en.wikipedia.org/wiki/Asimov`.

[95] Wikipedia. Nikola tesla. `en.wikipedia.org/wiki/Nikola_Tesla`.

[96] Wikipedia. Remote control. `en.wikipedia.org/wiki/Remote_control`.

[97] Wikipedia. Robot. `en.wikipedia.org/wiki/Robot`.

[98] A. F. T. Winfield. Distributed sensing and data collection via broken ad hoc wireless connected network of mobile robots. In G. Bekey L. E. Parker and J. Barhen, editors, *Distributed Autonomous Robotic Systems 4*, pages 273–282. Springer-Verlag, 2000.

[99] Yuan Xue and Klara Nahrstedt. Providing fault-tolerant ad-hoc routing service in adversarial environments. *Wireless Personal Communications, Special Issue on Security for Next Generation Communications*, 2004.

[100] Bin Zhang and Gaurav S. Sukhatme. Controlling sensor density using mobility. In *The Second IEEE Workshop on Embedded Networked Sensors*, pages 141 – 149, May 2005.

[101] F. Zhang, D. M. Fratantoni, D. Paley, J. Lund, and N. E. Leonard. Control of coordinated patterns for ocean sampling. *International Journal of Control*, 80(7):1186–1199, 2007.

[102] F. Zhang, M. Goldgeier, and P. S. Krishnaprasad. Control of small formations using shape coordinates. In *Proc. of 2003 International Conf. of Robotics and Automation*, pages 2510–2515, Taipei, Taiwan, 2003. IEEE.

[103] F. Zhang and N. E. Leonard. Coordinated patterns of unit speed particles on a closed curve. *Systems and Control Letters (submitted)*, 2006.

[104] F. Zhang and N. E. Leonard. Coordinated patterns of unit speed particles on a closed curve. *Systems and Control Letters*, 56(6):397–407, 2007.