Adaptive Communication Networks for Heterogeneous Teams of Robots

by

Stephanie Gil

Submitted to the Department of Aeronautical and Astronautical Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2014

© Stephanie Gil, MMXIV. All rights reserved.

The author hereby grants to MIT permission to reproduce and to distribute publicly paper and electronic copies of this thesis document in whole or in part in any medium now known or hereafter created.

Author .....
Department of Aeronautical and Astronautical Engineering
January, 2014

Certified by .....
Daniela Rus
Professor, MIT, CSAIL Director
Thesis Supervisor

Accepted by .....
Paulo C. Lozano
Chair, Graduate Program Committee
Associate Professor of Aeronautics and Astronautics, MIT

Adaptive Communication Networks for Heterogeneous Teams of Robots

by

Stephanie Gil

Submitted to the Department of Aeronautical and Astronautical Engineering
on January, 2014, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Seemingly, the era of ubiquitous connectivity has arrived, with smart phones, tablets, and small computing devices bringing internet straight to our fingertips – or has it? Two thirds of the world still does not have access to the internet, and a lack of realistic communication guarantees for multi-agent robotic networks are standing in the way of taking these systems from research labs into the real world. In this thesis we consider the problem of satisfying communication demands in a multi-agent system where several robots cooperate on a task and a fixed subset of the agents act as mobile routers.

Our goal is to position the team of robotic routers to provide communication coverage to the remaining client robots, while allowing clients maximum freedom to achieve their primary coordination task. We develop algorithms and performance guarantees for maintaining a desired communication quality over the entire heterogeneous team of controlled mobile routers and non-cooperative clients.

In the first part of the thesis we consider the problem of router placement while explicitly accounting for client motion over *a priori* unknown trajectories. We formulate this problem as a novel optimization called the connected reachable k-connected center problem that extends the classical k-center problem. We propose an algorithm to compute a small representative set of clients where this set is of size $(klog(n)/\varepsilon)^O(1)$, can be constructed in $O(nk)$ time and updated in $(klog(n)/\varepsilon)^{O(1)}$ time as clients move along their trajectories. Here k is the number of routers, n is the number of clients, and ε is a user-defined acceptable error tolerance. Our router placement algorithm applied to this sparse set provides a configuration of router positions that is bounded by a multiplicative factor, $(1+\varepsilon)$ from optimal.

Secondly, we incorporate a realistic communication model into our router placement optimization problem. We do this by developing a novel method of directional signal strength mapping that has sufficient richness of information to capture complex wireless phenomena such as fading and shadowing, and can be used to derive a simple optimization formulation that is based on quadratic link costs and is solved using our router placement algorithm. Using off-the-shelf hardware platforms we

present aggregate results demonstrating that the resulting router placements satisfy communication demands across the network with 4X smaller standard deviation in performance and 3.4X faster convergence time than existing methods, and our solutions assume *no environment map* and *unknown client positions*.

Finally, we derive distributed controllers for the special case where clients are static. We show that by the tuning of a control parameter our routers maintain a connected network using only local information. We support our theoretical claims with experimental results using AscTec hummingbird platforms as well as iRobot Create platforms of small 10 client and large 500 virtual client implementations.

Thesis Supervisor: Daniela Rus
Title: Professor, MIT, CSAIL Director

Acknowledgments

First and foremost I would like to acknowledge my advisor Daniela Rus who over the course of my doctoral studies has provided ample support, guidance, creative freedom, and unrelenting challenge that has directly led to the synthesis this work.

I would also like to thank the Bell Labs Graduate Research Fellowship, NSF Graduate Research Fellowship, the MIT Aeronautical and Astronautical department fellowship, MAST, SMART, and Lincoln Labs for their financial support which has made this work possible. In particular, I would like to acknowledge Brian Sadler for his support through MAST and for his guidance towards seeking communication network solutions that while grounded in theory, are capable of achieving real-world performance. In many ways Dr.Sadler's input has shaped the problems targeted in this thesis and the character of this thesis' contributions.

I would like to acknowledge my committee Nick Roy and Emilio Frazzoli, as well as my collaborators who have been important contributors both to the thesis and to my academic and professional development. In particular I would like to acknowledge Dan Feldman for introducing me to coreset approximations, as well as Swarun Kumar and Dina Katabi for introducing me to new tools and techniques in wireless communication. More broadly I would like to thank Mac Schwager, Brian Julian, Lixin Shi, Sam Prentice, and Lars Blackmore for their collaborations over the course of my graduate studies. I would like to thank Dimitri Bertsekas for research provoking conversations and guidance, as well as Brian Williams for his mentorship throughout the program. To the DRL family, Antonio Molins, Hoda Eydgahi, and Miriam Makhlouf, I thank you for your support in completing this degree.

Finally, I would like to thank my family for their omnipresent encouragement; I dedicate this thesis to you.

This doctoral thesis has been examined by a Committee of the Department of Aeronautics and Astronautics as follows:

Professor Nicholas Roy .....
Chairman, Thesis Committee
Professor of Aeronautics and Astronautics, MIT

Professor Daniela Rus .....
Thesis Supervisor
Professor of Computer Science, MIT

Professor Emilio Frazzoli .....
Member, Thesis Committee
Professor of Aeronautics and Astronautics, MIT

Contents

1 Introduction **23**

1.1 Motivation: Why Multi-Robot Communication? 23

1.2 The Challenges of Providing Communication over a Heterogeneous Network 25

1.3 State of the Art 27

1.4 Our Approach 33

1.5 Contributions, Assumptions, and Scope of the Thesis 37

1.6 Thesis Roadmap 42

2 Problem and Related Work **43**

2.1 Problem Overview 43

2.2 Related Work 44

2.2.1 Works Related to Coverage of Client Vehicles 46

2.2.2 Works Related to Wireless Communications Tasks 51

2.3 Preliminaries 62

3 Proximity-based Communication for Dynamic Clients with Unknown Trajectories **65**

3.1 Introduction 65

3.1.1 Assumptions 66

3.1.2 Summary of Problems Addressed 67

3.1.3 Results Snapshot 68

3.1.4 Notation 70

3.2 Problem Formulation based on the k-Center and Reachability Problems 70
3.2.1 Problem Statement . . . 70
3.3 A Formulation for Optimizing Router Placements . . . 75
3.3.1 Background on the k-Center and Reachability Problems . . . 75
3.3.2 Controller Development . . . 77
3.4 Algorithm for Finding k-Connected Centers with Reachability Constraints . . . 84
3.5 Numerical Studies . . . 86
3.6 Conclusion . . . 87

4 Efficient Computation of Router Placements 89
4.1 Introduction . . . 89
4.1.1 Assumptions . . . 90
4.1.2 Results Snapshot . . . 91
4.1.3 Notation . . . 92
4.2 Problem Statement . . . 93
4.3 Tracking a Representative Set of Clients . . . 94
4.3.1 Empirical Results for Static Approximations . . . 96
4.3.2 An Improved Dynamically Updated Approximate Solution . . . 97
4.3.3 Algorithm for Maintaining a Kinematic Coreset . . . 98
4.3.4 Analysis . . . 101
4.4 A Constant Factor Approximation for the k-Connected Center Problem 105
4.5 Empirical Results . . . 105
4.5.1 Hardware Experiment . . . 107
4.5.2 Numerical Simulation . . . 108
4.6 Conclusion . . . 114

5 Real-world Communication with Unknown Client Locations 119
5.1 Introduction . . . 119
5.1.1 Assumptions . . . 120
5.1.2 Results Snapshot . . . 120

5.1.3 Contributions Summary 122
5.1.4 Notation 123
5.2 Problem Formulation 123
5.2.1 Problem Statement 124
5.3 Derivation of Directional Signal Quality Maps 127
5.3.1 Beamforming 128
5.3.2 Synthetic Aperture Radar (SAR) 130
5.3.3 Challenges in Implementing SAR on Independent Wireless Devices 133
5.4 Capturing Real-World Communication in a Simple Quadratic Controller 134
5.4.1 A Generalized Distance Metric for Incorporation of Channel Feedback 135
5.4.2 Network Trade-offs 141
5.4.3 A Position-Independent Solution 142
5.5 A Reduction to the Router Placement Problem 144
5.6 Empirical Results 145
5.6.1 Computing Direction of Maximum Signal Strength 146
5.6.2 Controlling Router Trajectory to satisfy Client Demands . . . 147
5.6.3 Aggregate System Results 148
5.6.4 Comparison with Existing Schemes 149
5.6.5 Robustness to Dynamic Obstacle Positions 151
5.6.6 Complex Indoor Environments 152
5.6.7 Discussion on SAR in Higher Dimensions 153
5.6.8 Conclusion 154

6 A Distributed Router Placement Formulation for Static Clients 155
6.1 Introduction 155
6.1.1 Notation 156
6.2 Decentralized (local) Cost Development based on Physical Model . . 158
6.2.1 Problem 158

6.3 Non-Smooth Analysis of Controller 162
6.3.1 Generalized Gradient and the Generalized Gradient Vector Field 162
6.3.2 Stability of Controller 163
6.3.3 Maintaining Network Connectivity 166
6.4 Emperical Results 172

7 Conclusions, Lessons Learned, and Future Work 177
7.1 Lessons Learned 178
7.2 Future Work 179
7.2.1 The Effects of Time Delays and Positional Uncertainties on the Network 180
7.2.2 Generalized Router and Client Dynamics 181
7.2.3 A k-Means Formulation 182

List of Figures

1-1 Advantages of theoretical vs. experimental treatment of the problem. 28

1-2 Schematic drawing of a true signal strength map along different directions in the local environment of a robotic agent. Large lobes correspond to high signal strength. 30

1-3 Data from hardware experiments of 3 clients and 1 router compares the performance of current methods, the Euclidean disk and stochastic sampling methods, for incorporating communication quality demands. The blue arrows indicate the traversed path of the router from initial to final converged positions. 31

1-4 An example of a typical measurement (from an actual experiment) of a directional signal strength map, accompanied by a schematic interpretation of the directions over which this data reports. 33

1-5 A simple reduction allows us to use our router placement algorithms to optimize the network for achieving demanded client rates according to realistic communication models. 35

1-6 Thesis content summary and organization. 37

1-7 Examples of our hardware platforms include heterogeneous implementations with iRobot Create vehicles and AscTec hummingbird quadrotors both small (2 routers, 10 clients) and large (3 routers, 500 virtual clients). 39

2-1 A standard model for predicting signal quality (signal strength) along different spatial positions [35,79], and the related assumptions common to approaches utilizing these models. 56

2-2 A directed antenna (left) and the result of beamforming applied to the received signals where the signal received along a particular direction is largest (large lobe indicates high signal strength) [110]. 60

3-1 The point p can communicate with q through the path of centers p, c_1, c_2, c_3, q. The minimum required distance r for this communication is the longest edge (c_2, c_3) in this path. 67

3-2 Schematic drawings showing the differences between the k-center and connected k-center solutions for the n=3, k=2 case (a) and the n=3, k=3 case (b). Note that clients have the same positions for all scenarios depicted above. 69

3-3 Schematic representation of the k-Center, k-Connected Center, and Reachable k-Connected Center Problem. 79

3-4 Schematic of both scenarios of our case study where sensors (red circles) either move randomly through the environment or move radially outward and test the communication limits provided by the centers (blue squares). 86

3-5 Effect of sensor velocity on expiration time studied for a fixed $R = 60$m and communication (router) vehicle speed $v_C = 1.5m/s$. Different curves correspond to different ratios of comm vehicles to sensors k/n and the slope of the curves become shallower as the number of router vehicles k increases per unit area. 87

3-6 Plot of t_e vs. radial sensor distance from center of environment. Computation time (horizontal dashed lines) is compared against expiration time (solid blue line) and switch times (solid vertical lines) between exact and approximate methods is demonstrated. 88

4-1 Schematic interpretation of the property of a (k, ε)-coreset. 94

4-2 This plot shows aggregate results over 1000 runs for the error induced by using a representative set of size $|S|$ for the input set P vs. increasing representative set size. The plot shows that a *coreset* (solid line) provides better performance with approximation error $\varepsilon \leq 0.34$ (vs $\varepsilon = 3.8$ for a uniform random sample (dashed line)) for only $|P|/2$ sample points when $|P| = 30$, and $\varepsilon \leq 0.05$ (vs. $\varepsilon = 2.7$ for a uniform random sample) with less than $|P|/3$ sample points when $|P| = 300$. . . 97

4-3 Overhead view of our hardware setup of two Kuka Youbot robots and white helmets with vicon tracking markers that were worn by five adults, or "clients", moving around the motion capture room at a walking pace as shown in figure 4-5. This figure shows the constraint that the first Youbot was tethered to a power source at the top left corner of the environment and the second Youbot was tethered to the first Youbot for power. The Matlab plot in the top right corner demonstrates the two kinematic coreset points (red) and the optimal configuration of the Youbots (blue) computed for the client positions under the given tethering constraints. 98

4-4 Typical test scenario where clients move randomly between clusters forming large and small clusters. The red squares are sampled input points from a (k, ε)-coreset demonstrating that most clusters (both large and small) are fairly sampled in contrast to uniform random sampling where several small clusters are often missed thus adversely affecting approximation quality. On the left, key properties of the construction of a (k, ε)-coreset are explained. 101

4-5 Side and overhead views of hardware experiments for heterogeneous Kuka Youbot (server) and human (client) systems. Arbitrary intial positions with coreset (left), servers tracking moving client (middle), overhead view of clients divided into two clusters and resulting coreset (right). Matlab plots show computed kinematic coreset points (red), commanded Youbot positions (blue), and power tether constraints (blue line). 106

4-6 These plots show calculation time to compute updated representative set after each position update for points in the input set P for up to $n = 2000$ input points averaged over 500 runs, and the cost over each representative set after each position update for points in the input set P for up to $n = 20$ input points averaged over 500 runs. Results show that kinematic updates perform comparably to using a static coreset in terms of accuracy and random sampling performs up to $5\times$ worse than both of these methods. The last plot shows the case of server velocity constraints where a kinematically updated coreset outperforms a statically updated coreset since in the former case there is consistency between iterations. 109

4-7 A hardware implementation of a kinematic coreset over real San Fransisco taxi data for 500 taxis. Taxi trajectories were reflected in real time over a window of 10 minutes in San Fransisco. These data points were taken from an online source [69] and our router placement algorithms were not given information regarding these trajectories *a priori*. . . . 111

4-8 A plot of the cost $\max_{p \in P} \min_{s \in S} dist(p, s)$, ie. how well the input set of clients P is represented by the sparse set S. The sparse set S is computed as a uniform random sample of points from P (black), a static coreset (blue), or a kinematic coreset (green). 111

4-9 This plot compares the size of the representative set S for every iteration where a new router placement is computed. The data shows that a constant sized representative set of 50 taxi points are used for the uniform random sample whereas both the static and kinematic coresets < 50 points but represent the entire client set with a much smaller cost. 112

4-10 This plot shows the resulting k-connected center cost for each router placement over 180 iterations. For the case where taxis roam to the outskirts of the city (iterations 140-180) the uniform random sample set cannot maintain a bounded error on the solution and thus grows arbitrarily large due to misrepresentation of these outlier points in the client set. 112

4-11 This plot shows the accumulated distance traveled by the routers and compares the stability of the resulting router trajectories for each choice of representative set. The advantage of using a kinematic coreset over a static coreset for dynamic points is clear from this plot which shows that the consistency provided by a kinematic coreset across iterations allows for the same error bound in cost as a static coreset but with much less travel required of the routers. 113

4-12 Snapshot of hardware experiment involving a heterogeneous robot platform with 10 mobile sensor clients (iRobot Create vehicles) and two mobile routers (AscTec Hummingbird quadrotors). 113

5-1 A problem sketch demonstrating our goal of positioning routers to satisfy heterogeneous communication demands of different clients over the network. 124

5-2 A schematic showing a sensor antenna array and a corresponding direction of arrival (DOA) plot for a beamformer steered to 18 degrees. On the right a schematic shows how the received signals are weighted and processed to produce the output from J sensors at time k [110]. . 129

5-3 (a)/(c) LOS and NLOS topologies annotated with signal paths. (b)/(d) $f(\theta)$ of the signal in LOS and NLOS. (e) Shows how θ is defined in SAR. (f) Shows $h(t_i)$, the forward channel from transmitter to receiver and $h^r(t_i)$, the reverse channel from receiver to transmitter at time t_i. 131

5-4 This figure shows an image of our off-the-shelf platform, as well as a schematic of our technique for gathering wireless channel information for constructing our directional signal strength maps (this supplements the sketch in Figure 5-3). 132

5-5 A directional signal strength map computed during our experiments. The accompanying schematic shows an interpretation of the information contained in this signal strength map. 132

5-6 These plots show directional signal strength profiles from actual experiments. They demonstrate how the confidence metric identifies cases of high confidence, low confidence, and multipath automatically from the f_{ij} signal strength profile. The dotted red line is the variance,σ_N, of a uniform signal strength profile $f_{Nij}(\theta) = 1/L$ centered around $\theta_{\max}$. Comparing the variance (bottom row) σ_f to σ_N indicates which of the three cases are occuring in the f_{ij} plot (top row): high confidence $\theta_{\max}$ ($\sigma_f < \sigma_N$), low confidence (noise) $\theta_{\max}$ ($\sigma_f \approx \sigma_N$), or multipath around $\theta_{\max}$ ($\sigma_f > \sigma_N$). 138

5-7 Data collected for a one-link system of one router and one client where the client is stationary at the top right corner of a basement environment and a mobile router is driven in a lawn mower pattern throughout the environment through line-of-sight and non-line-of-sight regions. Each colored data point represents an acquired directional signal profile (two example profiles are shown) and the color of the data point is the result of automatic mode detection from the data using the confidence metric from Eq. (5.15) where red=noise, yellow=multipath, and green=high confidence peak. 139

5-8 These plots show the level sets of a Euclidean distance function and a Mahalanobis distance function. 141

5-9 Trade-offs between Clients: (a)-(b) show the $f_{ij}(\theta)$ map for the high demand client; (c)-(d) show the optimized router direction; 143

5-11 (a) Depicts testbed with robot router servicing three clients in a candidate non-line-of-sight setting. The blue line depicts the trajectory, and colored arrows indicate instantaneous θ_{max} for the corresponding clients. (b) Plots the ESNR across time (as dotted lines) for each client through the experiment. Solid lines denote client demands. 147

5-10 Gradient field of θ_{max} and power profile for (a) Line-of-sight and (b) Non-Line-of-Sight. 147

5-12 Statistical results obtained over 5 runs. Our plots show that demands are consistently met even in the presence of obstacles as demonstrated by the candidate converged solutions. 148

5-13 Plots comparing our method against the Euclidean disk model and a stochastic gradient descent method based on ESNR. Our method both converges to a position that meets communication demands, and converges quickly along an efficient path. 150

5-14 These plots show the result of disturbing the wireless channels via movement of a line-of-sight obstructing obstacle. Actual testbed snapshots are shown on the right. 151

5-15 Trajectories using measured $\vec{v}_{\theta_{\max}}$ directions satisfy a client's demand in line-of-sight and non-line-of-sight settings in complex indoor environments. 153

6-1 These figures show the initial and converged configurations for two routers and three ground clients. Figure 6-1(c) demonstrates the new equilibrium achieved when one flier is re-assigned to a ground station. 157

6-2 Plot of f_{ij}. 161

6-3 This plot shows the force felt by a communication vehicle in the presence of two clients, S1 and S2. It demonstrates the effect of the design parameter ρ on the communication vehicle gradient field where connectivity is maintained for $\rho \geq \rho_{\text{Crit}}$. 169

6-4 Position trajectories and aggregate cost function for three routers (shown as blue solid line in Fig 6-6(a)) with router equilibrium positions marked as blue squares and ground client positions marked as red squares. After reaching equilibrium one of the routers is deactivated and moved to the side while the remaining routers find a new equilibrium position (post-deactivation trajectories shown in dotted magenta line). 174

6-5 Position data and cost function for hardware-in-the-loop simulation where router trajectories are shown as blue lines and converged positions as blue dots. The ground clients are plotted as red squares in this figure. 175

6-6 Matlab simulation results of converged positions and position trajectories for 8 routers and 8 ground vehicles with non-smooth cost function H. Initial router positions are shown as blue circles, converged positions are shown as filled blue circles, and trajectories are shown as a blue line in Figure 6-6(a). Communication radius of clients 1 and 6 shown in green demonstrate that not all nodes are in communication initially. Trajectories as well as cost function show non-smooth transitions at the points where nodes enter each others communication neighborhood. 176

List of Tables

3.1 Common Notation for Chapter 3 71

4.1 Common Notation for Chapter 4 92

4.2 This tables summarizes the result of our hardware experiment employing two servers and five clients. Computing new server positions over a kinematic coreset (KC) of two points is 97× faster with approximation cost of $\varepsilon = 0.14$ compared to performing computation over entire input set of five points. In contrast, naïvely sampling two input points at random (U) produces an approximation cost of $\varepsilon = 0.5$ at comparable computational speed. Calculations of server positions using three points shows similar trends. 108

5.1 Common Notation for Chapter 5 123

6.1 Common Notation for Chapter 6 156

Chapter 1

Introduction

1.1 Motivation: Why Multi-Robot Communication?

We are moving towards a future where all vehicles on the road have increasing levels of autonomy and will interact with each other to ensure safety and efficiency, where teams of robots will be deployed in hazardous environments to perform search and rescue tasks, and where maneuverable devices will be incorporated to help with everyday tasks such as global connectivity. In the case of search and rescue tasks, disaster relief efforts are inhibited by lack of information of where these efforts should be directed (i.e. where survivors may be trapped), and the absence of a communication infrastructure where valuable information can be disseminated quickly. If a such a communication network could be deployed in these cases, it could configure itself to allow robots exploring the wreckage to send video streams back to human relief workers so that they can coordinate a safe and targeted rescue effort. In the case of global connectivity, large ad-hoc networks could bring wireless access to remote areas in the world where just a small number of mobile base-stations can position themselves to provide best bandwidth service to users dependent on their needs (i.e. provide highest bandwidth to schools during the day and relocate to provide highest bandwidth to neighborhoods in the evenings). Ideally, we would be able to mount wireless routers

onto several mobile base-stations that can autonomously position themselves to form a communication network for supporting the rates needed to complete these tasks. Along these lines, Google's Project Loon is investigating the use of maneuverable hot air balloons to provide internet service across the two-thirds of the world's population who do not currently have wireless infrastructure. The next natural question for these efforts is, how can we enable such networks to satisfy variable communication needs by leveraging the mobility inherent to these communicating platforms?

In order for these heterogeneous networks (composed of mobile base-stations that create the necessary infrastructure, and clients who utilize this infrastructure) to be applied to solve tasks in real-world environments, we must couple the control and communication of the robot routers. In this thesis we focus on developing theoretical and experimental foundations for communication amongst multi-agent teams in real-world environments. The class of problems that we investigate are those where mobile base-stations are positioned to provide communication coverage to robots, sensors, and/or people on the move, in environments without an existing communication infrastructure. We will generally refer to the users of the network as "client-agents." We are interested in the case where these mobile base-stations can be positioned autonomously in a way that is adaptive, i.e. where the communication network can re-position as needed due to dynamic environments, or changes in the demanded communication rates over different links in the network. This capability facilitates many multi-agent tasks such as coverage, exploration, search and rescue, rendezvous and flocking amongst many others. In particular, most of these tasks require reliable communication in order to exchange information and converge to a desired behavior [25, 85, 101, 106]. Establishing a communication network that is general to many multi-agent tasks requires simultaneous development along two fronts: i) investigation of realistic communication models that relate spatial positioning to signal quality between any two communicating agents and ii) design of controllers that use these communication models to best position the mobile base-stations to satisfy (possibly variable) communication requirements over the network.

1.2 The Challenges of Providing Communication over a Heterogeneous Network

Incorporating Wireless Signal Quality via Channel Feedback

The problem of providing wireless communication coverage amongst multi-robot systems is very challenging. At the crux of this problem is the following realization: the majority of robotic tasks leverage mobility in Euclidean three-dimensional space and thus require knowledge of how positions in $\mathbb{R}^3$ effect communication, however, wireless signals are notoriously difficult to predict via analytical models [52, 75, 79]. The quality of a wireless link are often related in complex ways to the structure of the environment, such as the presence of obstacles (and their material composition [104]), as well as any dynamics of the surroundings (i.e. moving people, cars, robots, etc). This makes the predictability of signal quality very challenging and often specific to different environments. Thus, it is not clear how to develop mathematical solutions that both allow for guaranteeing the communication quality necessary for the performance of our systems, and sufficiently capture complex wireless phenomena such that our guarantees hold for real world implementations.

Keeping a Network Connected and Balanced as Clients Move

Assuming that we do have sufficient models of communication quality, how do we trade-off client coordination goals such that they may have the freedom to choose trajectories that achieve their primary task (exploration, coverage, rendezvous, etc) while remaining connected? For example, even under the idealistic assumption that communication quality is perfectly predictable and given by a Euclidean distance metric, how should our mobile base-stations divide themselves amongst the network of clients such that no link in the graph suffers from unacceptable communication quality (i.e. does not become unbalanced)? Here we refer to clients as the agents in the network that have a primary coordination task such as exploration or coverage and must use the communication network to complete this primary task. A careless

approach may result in pockets of the network that receive poor communication quality, or suffers from loss of connectivity altogether, thus compromising the coordination task. And finally, the dynamics of the clients could easily lead to a disconnected system over time if our mobile agents do not account for these dynamics and position themselves accordingly. However, finding a balance between restricting the paths of the mission agents in order to achieve communication goals, and giving these agents enough freedom to complete their primary task is a challenge in itself.

Updating the Network in Real-time even for Large Numbers of Clients

As we move towards the incorporation of these systems into everyday life, such as for the case of global coverage, we must be able to cater our solutions for larger and larger client bases. In order to find globally optimal solutions such that our mobile base-stations are best allocated according to network demand, these solutions become more expensive as the number of clients increase; and so we must move towards approximate solutions. Heuristic solutions however, are undesirable for then we do not have a sense of how far below optimal we perform given a fixed number k of mobile base-stations. Ideally, we favor solutions that can provide computable bounds on the approximation costs so that we may understand how far from optimal we are performing for a given ratio of mobile base-stations to clients. Such information can additionally provide guidance as to how many mobile base-stations should be deployed or what the maximum coverage area we can hope to service may be.

Connecting Clients with Unknown Positions and in Unknown Environments

Finally, for maximum flexibility of these communication networks and applicability to general environments, we cannot assume a known environment map or even known client positions. For example, in a disaster zone immediate response is often required and there is no time to survey the area or compose a map, although it is likely that structural damage has led to many obstacles and obstructions that will affect the ability to communicate reliably. By the definition of "exploration" in coverage or

exploration scenarios, it is implied that a map of the environment is not known prior to deployment and thus catering to such coordination tasks also precludes using information about obstacles that may be present or their material composition. Further challenges are posed by the fact that these environments are also often GPS-denied, and we cannot assume that client positions are known.

Challenges Addressed in the Thesis

This thesis focuses on making progress along each of the following directions: i) deriving methods for measuring the relationship between spatial directions and signal strength, ii) formation of a wireless network infrastructure through the positioning of robot routers, where these positions are adaptive to client mobility and variable communication needs and iii) maintaining generality of our methods to unknown and gps-denied environments. The results in this thesis serve to mitigate or remove altogether the challenges stated in this section; which we believe bring us much closer to making these systems feasible for real-world application.

1.3 State of the Art

There are many projects on today's frontier that are pushing the capabilities of multi-agent systems. Google's *Project Loon* [73] (or similar projects [21, 38, 84]) that envisions using a network of hot-air balloons to provide wireless infrastructure for communication coverage in remote areas of the world, the DARPA Urban challenge [14] that advances technologies for a future of driver-less vehicles in a networked city, and robotic swarms [23, 66, 89] that can be deployed for autonomous exploration or search and rescue tasks all constitute just a snapshot of the potential of coordinated robotics applications. But there are many questions that must be answered before we can utilize this full potential. For example, if we could control the positions of Google's routers, how could we best allocate them across users such that with a limited number of resources we could best satisfy client demands and moreover, adapt to variable demands? The winning team of the DARPA Urban challenge describe

one of the largest challenges in bringing the autonomous vehicle vision to fruition in actual urban environments as lack of sensory information [14]. What if we could solve this problem in multi-vehicle networks, not by adding advanced sensing capabilities, but rather by leveraging all the information across the network via communication between vehicles [71]? For robotic swarms, many theorized capabilities rely on communication and the question of how to provide the necessary communication guarantees has stood in the way of bringing these systems from the laboratory into the real world.

The current state of the art in combining position control of networks of mobile base-stations with explicit treatment of the communication problem largely centers around either geometric models (such as Euclidean disk or visibility graph approaches) where signal quality is believed to depend deterministically on the environment, or stochastic models where the quality of the wireless signals at different points in the environment is measured and incorporated into the controller. Thus there is a natural division in the contributions made towards solving this problem; those using a theoretical treatment based on idealized (deterministic) models of communication, and those pursuing an experimental treatment based on realistic models of communication that use feedback on the wireless signals (see Figure 1-1).

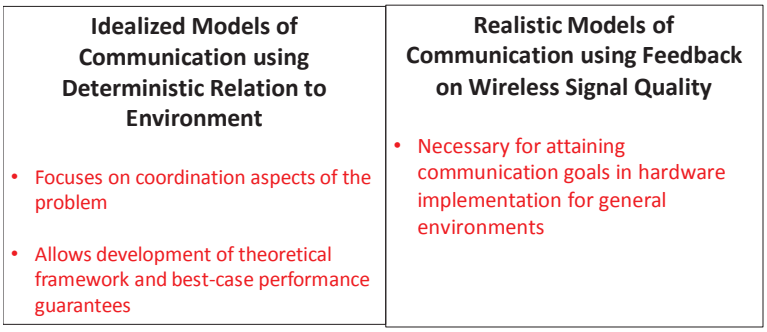


Figure 1-1: Advantages of theoretical vs. experimental treatment of the problem.

Two of the most ubiquitous approaches for modeling the dependence of signal quality on spatial positioning for every pair of communicating vehicles (also referred to as nodes for generality in this section) are:

(i) **Euclidean Disk Approach** In this case the signal quality is assumed to be deterministic and mapped perfectly to the Euclidean distance between the communication nodes. Controllers developed using this assumption are open-loop and oblivious to the wireless channels. In reality, signal strength suffers from large variations even over small displacements [75]. Euclidean disk models simply do not capture this reality. However, these controllers are often very simple in nature due to the fact that a Euclidean metric allows for quadratic cost structures and is amenable to graph-theoretic treatments of the resulting communication systems. Thus this approach has a natural appeal due to this simplicity that allows for both a geometric treatment of an otherwise complex problem, and the derivation of theoretical performance guarantees for the communication network.

(ii) **Stochastic Sampling Approaches** These approaches attempt to incorporate a more realistic treatment of the signal quality by sampling the signal strength at discrete locations in space and inferring a closed-loop model of the signal strength in the local environment, see Figure 1-2. Acquiring these samples are energy expensive however, because they require physically moving the vehicle to random exploratory points in the local environment in order to sample the signal quality (the blue arrow trajectory in Figure 1-3 demonstrates the added exploratory paths typical of stochastic sampling approaches). Therefore these samples are often supplemented with certain models that attempt to capture the remaining part of the signal strength curve. Specifically, these samples are assumed to come from particular distributions whose parameters must be found such that the samples fit the distribution. Unfortunately, these approaches are often prohibitive for real-world application due to the following reasons: i) these models are often complex, ii) it is unclear which model to choose for the

environment since the fitting of these parameters often depend on the structure of the environment (how many obstacles are present) and the materials in the environment (concrete vs. dry wall for example), iii) they often require static environments (otherwise the samples are invalidated) and iv) they often assume that node positions are known. See Figure 2-1 for an example of a standard signal quality prediction model and accompanying assumptions.

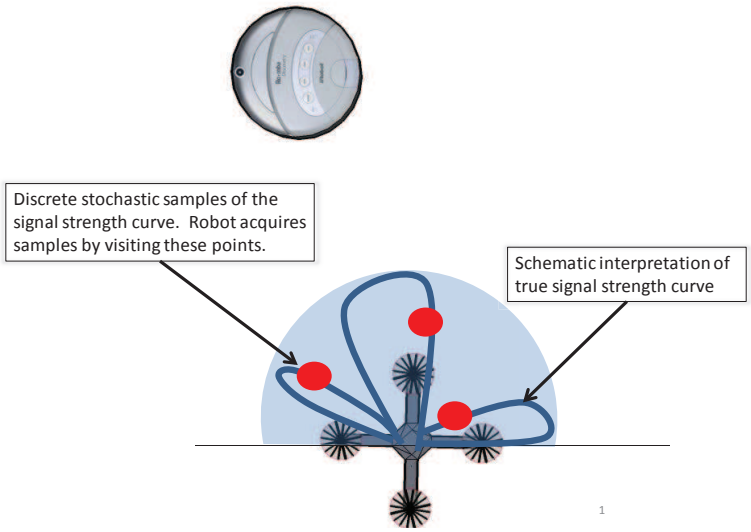
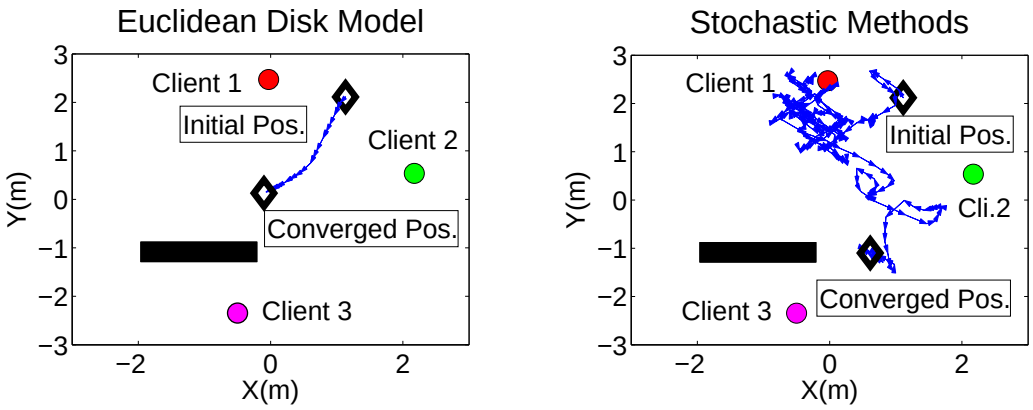


Figure 1-2: Schematic drawing of a true signal strength map along different directions in the local environment of a robotic agent. Large lobes correspond to high signal strength.

Using either the Euclidean disk or stochastic sampling approach, higher level controllers are then developed to carry out the primary coordination task (e.g. coverage, consensus, flocking, etc) in a communication-aware setting. In general, geometric approaches (such as work based on the Euclidean disk model) focus on the coordination aspects of the problem such as: 1) how to best allocate the robot routers, 2) how to allow for topology changes without losing connectivity of the network, and 3) what the best performance guarantees are that we can hope to achieve. On the other hand, for sampling based approaches the focus is largely on experimental validation of algorithms, i.e., showing that algorithms are capable of achieving desired rates or other performance goals in actual implementations. Both approaches, one



(a) (b)

Figure 1-3: Data from hardware experiments of 3 clients and 1 router compares the performance of current methods, the Euclidean disk and stochastic sampling methods, for incorporating communication quality demands. The blue arrows indicate the traversed path of the router from initial to final converged positions.

that makes idealized assumptions on communication but provides theoretical insight and guarantees on agent coordination, and one that takes a more realistic treatment of wireless communications but targets experimental results, address very challenging aspects of the communication coverage problem. We provide a detailed account of related work both for different communication models and for the resulting control approaches in Chapter 2.2 of the thesis.

Relation of the Thesis to the State of the Art

This thesis focuses on providing communication support to general multi-agent tasks where our controllers use simple quadratic costs to represent signal strength over every link in the network. Under this framework we make progress on both i) deriving theoretical performance guarantees for the coordination aspects of the problem under the simplifying assumption of a Euclidean disk model for communication and ii) generalizing the theoretical framework to incorporate a more realistic treatment of the wireless link quality by using channel feedback, resulting in the attainment of performance goals in actual hardware experiments.

(i) **Contributions in Multi-agent Coordination under the Euclidean Disk Model:** The multi-agent coordination aspects of the communication coverage problem include router placement (i.e. how to best allocate routers such that communication service to clients is maximized), and questions of connectivity (i.e. how to maintain a connected network for a certain duration of time). This thesis extends the state of the art in this domain in several ways. Firstly, we focus on heterogeneous problems of clients and routers where clients are allowed to be non-cooperative, i.e. we allow for an entire subset of the network to be outside of our control authority. This assumption allows for a much greater level of freedom for the client agents who may then complete their independent tasks without needing to alter or report their trajectories to support communication. The objective here is to place routers such that connectivity is maintained over a computable time window and is robust to future client trajectories. Secondly, we target scalability of our results to large numbers of clients. In particular, we provide approximate solutions to the router placement problem that can be computed efficiently even as the number of clients increases, while bounding the approximation error within a known tolerance factor.

(ii) **Contributions in Network Optimization using Wireless Channel Feedback:** We provide a realistic treatment of the communication quality by developing a novel method for directly measuring a mapping that relates signal strength to relative heading directions for each router. By "mapping" we are referring to a function that assigns a signal strength value to different angular heading directions relative to the current heading of the router in the range of $[-\frac{\pi}{2}, \frac{\pi}{2}]$. This range represents all the directions from moving directly towards the signal source ($-\pi/2$ deg), moving directly away from the signal source ($\pi/2$ deg), and moving horizontally w.r.t. the signal source (0 deg). We do not assume knowledge of the analytical form of this mapping. Instead, our method allows for this mapping to be directly measured (see Figure 1-4) for each link in the communication graph. Additionally, we are able to measure the full

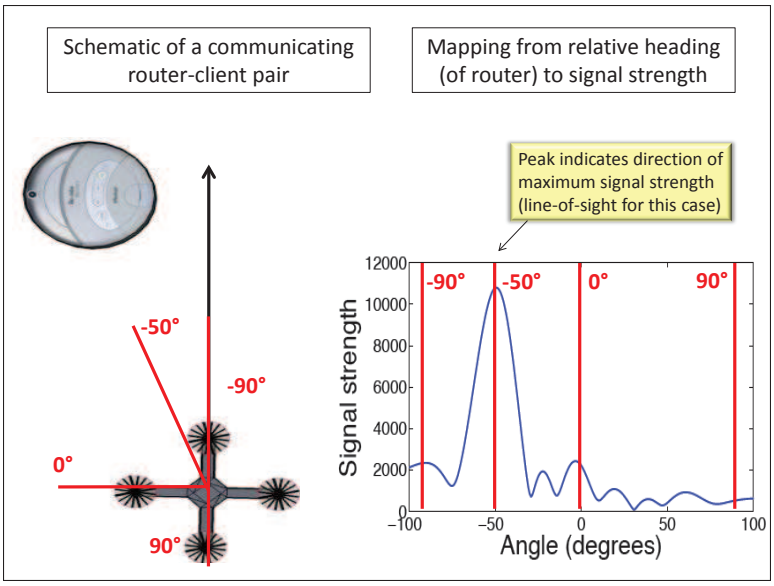


Figure 1-4: An example of a typical measurement (from an actual experiment) of a directional signal strength map, accompanied by a schematic interpretation of the directions over which this data reports.

signal strength map with no extra effort, i.e. by moving along any arbitrary path (the path that is conducive to the coordination goal in contrast to random exploratory paths). One key difference between this method and previous stochastic sampling methods is that we use the full channel information, signal strength plus *phase*, whereas previous methods use only signal strength (see Chapter 2 for an in-depth comparison). The interested reader is referred to Chapter 5 of the thesis for derivation of the method and resulting controller.

1.4 Our Approach

In this thesis we consider the problem of maintaining an ad-hoc communication network over a heterogeneous group of robotic agents, under realistic communication models, and with large numbers of mobile client-agents moving over unknown trajectories. We assume two types of vehicles, k mobile base-stations whose motion we control, and n client-agents whose motion we cannot control (e.g.. robots, mobile sensors, people, etc). The objective is to develop a controller guaranteed to provide a communication network to all clients according to their communication needs. The

controlled vehicles must adapt to the movement of the clients and the communication quality provided by their network must adapt to task needs and dynamic environments. Furthermore, we aim to provide control algorithms that meet performance goals in actual hardware implementations, where communication service is measured by instantaneous, achieved rates on each link in the network.

Our approach to this problem is to break it up into two main sub-problems each of which is challenging in its own right: **P1:** assuming an idealized model for communication quality versus position, solve the problem for placing routers to achieve balanced communication quality for mobile client-agents as well as large systems of clients and **P2:** derive a generalized problem formulation that accounts for a realistic communication model, by incorporating real-time channel feedback, while retaining simplicity of the resulting controller. We show that the general optimization formulation of problem 2 (P2) can be reduced to that of problem 1 (P1) so that the router placement algorithm derived in the first half of the thesis can be applied to the most general realistic communications settings. This results in powerful, yet simple, controllers for placing routers to form adaptive networks in general environments. Figure 1-5 demonstrates the interconnectivity between these two sub-problems of the thesis.

In the first half of the thesis we focus on the first sub-problem and make the following simplifying assumptions: i) signal quality is given by the Euclidean distance between two communicating nodes, ii) communication demands are equal for all clients, and iii) client positions are known. Under these assumptions we derive a new problem formulation and a corresponding algorithm for finding optimal placements for mobile base-stations. This solution is optimal in the sense that for a fixed number of base-stations the best allocation of resources is attained such that the worst-case communication link in the network is maximized. Our problem formulation also allows for client-agents to move along *a priori* unknown trajectories. This is necessary for allowing maximum flexibility for the clients who may need to change their plans on the fly to complete their primary task. For this case we can handle changing network topologies such that links in the graph can be broken and formed as

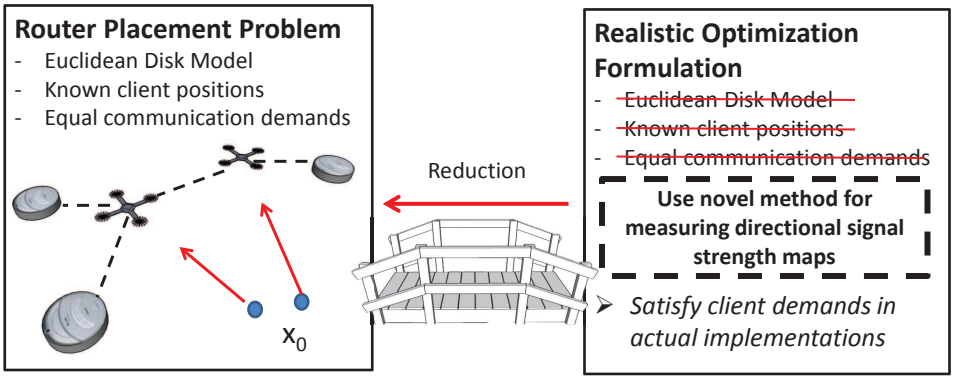


Figure 1-5: A simple reduction allows us to use our router placement algorithms to optimize the network for achieving demanded client rates according to realistic communication models.

necessary to maintain communication, and the amount of time for which the network connectivity is guaranteed can be directly related to client velocities. We scale our results to apply to large numbers of clients, where an approximate solution can be computed efficiently while remaining within a bounded error from optimality.

For the second sub-problem we focus on solving mobile base-station placements for 1) realistic communication scenarios, 2) where client positions are not known, and 3) client communication demands are heterogeneous and possibly variable over time. We replace the idealistic assumption from our previous problem formulation, that maps signal strength to Euclidean distance between nodes, with a generalized Mahalanobis distance that is carefully constructed to incorporate channel feedback. We show that by making this modest generalization to our previous problem formulation, we are able to incorporate realistic communication phenomena into our controllers for mobile base-station placement. We are able to achieve this by deriving a mapping between a mobile base-station's current heading and the signal strength that

it receives along *each spatial direction* for its wireless links to every other node in the system. The left-hand side of Figure 1-4 shows a schematic diagram of a router (quadrotor vehicle) and a client (iRobot Create vehicle) representing a single edge in the communication network. The plot on the right-hand side of this figure is an actual measured directional signal strength mapping that shows the full profile of how the link's signal strength varies for different heading angles relative to the router. The maximum value on this plot gives a recommendation for a router heading that would maximize the signal quality along this specific communication link for this specific client. By simply moving along any arbitrary straight line path (not a random path) a router will simultaneously obtain a similar directional signal plot for *every* node in the network that it is currently communicating to.

We show that these directional signal strength mappings can be used to design a simple position controller that retains a quadratic structure, by using this information to carefully construct a generalized distance metric. This generalized distance metric then replaces the Euclidean costs for each link in the network that we had used for deriving our k-center-based router placement algorithms. This allows us to use our previously derived algorithm for finding mobile base-station placements; resulting in simple quadratic optimizations for the formation of ad-hoc networks that are adaptive to dynamic environments, can satisfy heterogeneous client demands, and do not rely on a known environment map nor exact client positions.

Finally, for the special case of slowly moving or static client-agents we investigate gradient-based, distributed optimizations for positioning our mobile base-stations that have the following characteristics: i) base-station positions are computed using only local information, ii) communication is optimized according to the well-known physical model based on Signal to Interference Ratio (SIR) developed by Gupta and Kumar [57], and iii) connectivity is provably maintained. For this special case we can simplify the solution and in particular, derive a distributed solution. This case is prevalent for cases such as failure scenarios where routers may fail and remaining routers must reconfigure to maintain connectivity, and/or clients exit or enter the communication network.

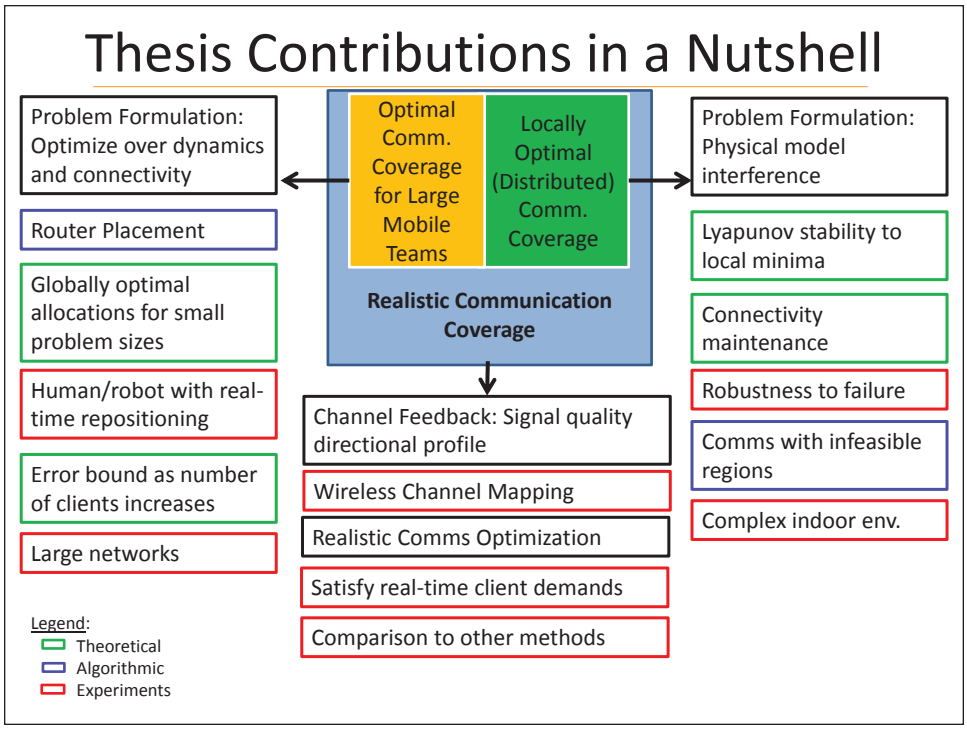


Figure 1-6: Thesis content summary and organization.

1.5 Contributions, Assumptions, and Scope of the Thesis

Scope of the Thesis

We investigate the class of problems where the communication task is formulated as the optimization of a cost that depends on base-station positions. Our objective is to navigate our mobile base-stations to a goal configuration that satisfies a specified communication quality metric, and remain fixed at this configuration unless there is a change in the environment, client-agents move, and/or client-agents change their communication demands. We target globally optimal solutions in the sense that for a fixed number of k mobile base-stations, these base-stations are optimally assigned to clients based on the current state of the environment; or in the case of an approximate large-scale solution, the approximation error can be bounded from optimal. We do not optimize over the entire trajectories of our vehicles to find globally optimal paths as in motion planning problems. This is because we do not assume known

communication maps over the entire environment, nor do we assume known environment/obstacle maps, and further, we wish to maintain adaptability to dynamic environments. Throughout the thesis we do not consider routing protocols. Our objective is to establish links that are capable of attaining the requested rates to client vehicles without regards to how clients may wish to utilize the network to route messages to other points in the network. We are mostly considering the case where our mobile base-stations have sufficient bandwidth to relay data throughout the network, whereas client communication hardware is assumed to be the larger bottleneck. This is often the case since the primary goal of the mobile base-stations is to provide communication and thus it is a natural assumption that their equipment would be better suited for high-bandwidth communication.

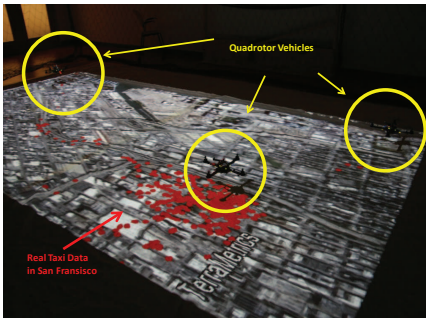
Contributions and Basic Assumptions of the Thesis

For the remainder of the thesis we will refer to client-agents as those that utilize the communication network, and our "mobile base-stations" as *router robots* whose positions are controlled in order to form the network. We assume that client and router dynamics are linear and subject to maximum velocity bounds. Our initial treatment of the communication coverage problem will make the following simplifying assumptions: 1) signal quality for any link is given by the Euclidean distance between the two communicating nodes (*Euclidean disk model*), 2) client positions are known to all mobile routers in the system, and 3) the requested communication rates are the same for all clients in the system. Under these assumptions we present the following contributions:

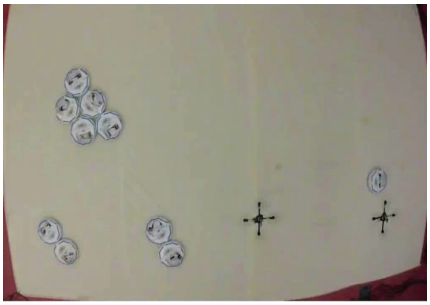
(i) **Solution of the router placement problem:** Algorithms for optimal placement of router robots for servicing dynamic clients such that the maximum distance edge on the communication graph is minimized.

(ii) **Accounting for unknown client trajectories:** Guarantees on connectivity of the graph over a computable time window, where client-agents are allowed to move over *a priori* unknown trajectories.

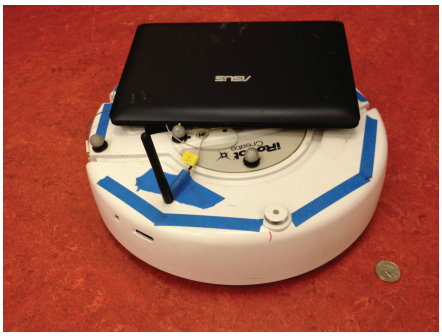
(iii) **Efficient real-time solutions for the router placement problem:** We present algorithms for computing a small representative set of client-agents (of size $O(klog(n)/\varepsilon^d)$ where d is the dimension) that can be constructed in $O(nk)$ time and updated in $(klog(n)/\varepsilon)^{O(1)}$ time as clients move. The router placement algorithm applied to this sparse representative set provides a configuration of router positions that is bounded by a multiplicative factor $(1+\varepsilon)$ from optimal where ε is a user-specified error tolerance.



(a) Large-scale quadrotor + 500 virtual taxi client vehicles (from real San Francisco taxi data)



(b) Small scale heterogeneous robot team



(c) Our off-the-shelf wireless mapping platform

Figure 1-7: Examples of our hardware platforms include heterogeneous implementations with iRobot Create vehicles and AscTec hummingbird quadrotors both small (2 routers, 10 clients) and large (3 routers, 500 virtual clients).

We then focus on replacing the Euclidean disk model with a realistic treatment of the communication quality over each link. We allow for heterogeneous communication demands over the network so that clients may request the required rate for their specific task (i.e. 50 Mb/s for streaming a video transmission or 5 Mb/s for requesting

a status update). We also relax the assumption on known client positions, although we still assume that robot router positions are known. Under these relaxed assumptions we provide the following contributions:

(i) **A new method for directly measuring directional signal strength maps** We derive a novel method for finding directional signal strength maps (see Figure 1-4) that directly measure the signal strength of a particular link along every spatial direction relative to the heading of each robot router using small, single-antenna, off-the-shelf platforms and commodity wifi cards.

(ii) **A problem formulation for router placements with realistic communication** We derive a new formulation for the router placement problem, that is based on simple quadratic costs, and whose optimization results in the formation of a communication network that can satisfy heterogeneous client demands in actual hardware implementations.

(iii) **Experimental Results:** We provide various empirical results showing the capability of the system to converge to configurations that satisfy all client demands on the network, while being adaptive to dynamic environment, with no known environment map, and no knowledge of the environment.

(iv) **Comparison to existing methods in hardware implementations** We provide aggregate empirical data to show that our method outperforms existing Euclidean disk or Stochastic sampling methods both in variability of performance (4X times smaller variance) and convergence time (3.4X faster).

An example of an actual directional signal strength map can be found in Figure 1-4 where the schematic on the right-hand side demonstrates the signal strength along each spatial direction relative to the robot router (quadrotor in this figure) heading. Although the principles of our approach are extensible to $\mathbb{R}^3$, our current development is in $\mathbb{R}^2$. Furthermore, by careful construction of our new optimization problem, we maintain the ability to reduce this most general problem to the previous router placement problem where now the Euclidean disk assumption is replaced with

a generalized distance metric based on wireless channel feedback. The result is that we are able to use our previous algorithms for finding router placements, culminating in simple controllers that capture realistic wireless communication and can perform well in real-world environments.

Finally, for the special case of static clients we derive distributed controllers for connectivity maintenance. In particular,

(i) **A distributed controller for communication optimization** We derive a physically motivated communication cost that encodes the effects of ambient noise and interference from neighboring agents competing for the same router.

(ii) **Topology changes in a connected network** We present a non-smooth stability analysis for convergence to local minima, where the non-differentiability is necessary due to agents moving in or out of communication with one another.

(iii) **Maintaining connectivity using local information** We prove that connectivity can be maintained in a distributed fashion either by i) satisfying certain computable conditions or ii)by setting a tunable design parameter of the controller to be greater than a critical value that we derive.

Summary of Contributions

In summary, the broad contributions of this thesis are:

(i) a communication coverage control algorithm for a network of mobile base-stations providing coverage to agents with unknown trajectories

(ii) a scalable version of the mobile base-station control algorithm that can accommodate hundreds of mobile agents with unknown trajectories

(iii) development of a virtual sensor for the local quality and directional gradient of the signal strength

(iv) an adaptive algorithm for controlling a network of mobile base-stations using the signal strength virtual sensor; the algorithm supports varying communication needs and demonstrates empirical gains over existing methods

(v) extensive simulation and hardware experiments

1.6 Thesis Roadmap

This thesis is organized in the following way (see Figure 1-6). In Chapter 2 we present the overall problem of communication coverage for heterogeneous networks and define nomenclature that is general to the body of the thesis. We also present a thorough literature review and a comparison to the work contained in the thesis. In Chapter 3 we present a formulation of the router placement problem for achieving communication coverage for clients moving over unknown trajectories under the assumptions of 1) the Euclidean disk model, 2) current client and router positions are known and 3) communication demands are homogeneous over the network. We present an algorithm for finding a configuration of routers that *balances* the network, meaning that the weakest link is maximized for a fair network. In Chapter 4 we present scalability results for the router placement problem. In particular, we present algorithms for finding a sparse subset of client vehicles such that the router placement algorithm can be computed efficiently over this representative subset and the resulting approximation error is bounded. In Chapter 5 we replace the Euclidean disk model assumption of signal quality and derive a new method for measuring rich directional information from wireless channel feedback. We pose a new router placement formulation that takes into account real-time feedback on the signal quality. We show that this most general case can be reduced to the router placement problem of Chapter 2 and thus can use this algorithm for achieving router placements that satisfy variable client communication demands in experimental implementations, where neither an environment map, nor client positions need to be known. In Chapter 6 we present a distributed, gradient-based controller for the special case where client-agents are static. In the final chapter we present conclusions and directions of future work.

Chapter 2

Problem and Related Work

2.1 Problem Overview

Throughout the thesis we focus on the following problem: deploy a group of robot routers to provide a communication service for client agents that are performing a collaborative task. Over the entire network we assume positional control only over a subset of the network, as the client vehicles are viewed as non-cooperative. In the sequel we will sometimes refer to robot routers as "routers" or "robots" to emphasize that we control their positions, and we will sometimes refer to the client agents as "clients" or simply "agents." Viewing the clients as non-cooperative is an important assumption as it allows the client vehicles maximum freedom to complete their independent collaboration task, without altering their positions or trajectories to accommodate communication, but while still benefiting from guaranteed communication quality necessary to complete their task.

Specifically, given a set of positions for n agents at time t, $P_t = \{p_{1t}, \ldots, p_{nt}\}$, and a set $C_t = \{c_{1t}, \ldots, c_{kt}\}$ of k router robots, where p_j, c_i are d-dimensional column vectors for all $j \in [n], i \in [k]$ and $t \geq 0$, we aim to find a configuration C_{t+1} of robot routers such that a communication cost g is minimized:

$$C_{t+1} = \arg\min_{C \in \Omega} g(P_t, C_t, C) \tag{2.1}$$

Where Ω is an optional constraint set; for example, it can be used to encode certain dynamical constraints on the communication vehicles. In this thesis we consider many aspects of this problem. We derive different formulations of the cost g such that the resulting optimized positions C_{t+1} provide minimum interference communication, maximize the worst-case service over the network in a proximity-based sense, or obtain some target level of service (i.e. specified communication rates). We also consider different methods for solving the optimization problem over mobile router positions. We develop solutions that may be distributed, where each router will only rely on local information for finding its updated position, or are globally optimal where we maximize the level of communication service over the entire network, and mobile router positions are assigned by a central server. Finally, we design the constraint set Ω, in combination with generalized formulations of the cost g, for solving the communication optimization problem where communication must be maintained for clients moving over unknown trajectories. In the following sections we provide a detailed review of work related to solving this problem of communication coverage, and we define a few preliminary concepts that will be used throughout the thesis.

2.2 Related Work

The question of how best to provide the required level of communication quality over a network with existing infrastructure has a rich history in the literature. The wireless and networking communities have extensively studied how to satisfy competing client demands by optimizing parameters such as scheduling, routing protocols, transmit power, spatial separation, and channel or frequency selection. The paper [77] discusses the optimization of a transmission schedule for multiple users over multiple channels that maximizes the total network throughput (attained data rate). A second dynamic

approach to scheduling transmissions is discussed in [42–44, 77] and [41] where "white space" or empty slots of time between bursty transmissions of data are utilized. This allows for orthogonality in the time domain (as opposed to the frequency domain) for maximum utilization of the network resources. The survey paper [120] provides an excellent motivation and overview of such methods. The paper [51] discusses the design of a routing protocol to transmit data over a static network using paths that are spatially separated in order to combat the malignant effects of interference and multipath on the communication quality. The survey paper [88] reviews many similar aspects of what is referred to as Cooperative Communication for improving communication quality over a network with multiple clients. Although these techniques have had a large impact in improving performance such that multiple users may enjoy high-quality communication, they applying to existing, often static infrastructure. But what if this wireless infrastructure does not exist?

In this thesis we study problems of the formation of wireless infrastructure by controlling robot routers to position themselves in configurations that provide demanded communication quality amongst multiple client agents. Here, we do not use the most common aforementioned parameters of transmit power, routing, channel selection or scheduling to optimize an existing network; rather, we leverage robot mobility to position robot routers to provide a network infrastructure with the capacity for providing the requested rates. Unlike static infrastructures, this allows for 1) temporary deployments of communication networks in remote areas or for disaster relief efforts where existing infrastructure may not exist or may have been damaged, 2) optimal usage of a limited number of routers that can adapt to accommodate the needs of the clients (i.e. some parts of the network may need to stream video to a base station while others may only need intermittent status update capability) and 3) mobility of the entire network (re-deployments of the routers in different environments as needed).

We divide the problem of communication coverage into two sub-parts, both of which are difficult problems in their own right: 1) understand the relationship between channel quality and spatial positioning, which is necessary for any location-based optimization of the communication network and 2) using a communication model as

a black box, control the positions of robot routers to provide the best communication coverage over multiple clients with competing demands. In order to tackle these problems, this thesis builds on a large body of related work in the areas of multi-agent coverage, facility location, control of dynamic systems under unknown but bounded disturbances, computational geometry, non-smooth analysis, and wireless communication. Similar to the way that we divide our approach to the problem into two parts, we divide the body of related literature into the following areas: 1) those bodies of work concerned with coverage where resources (static or otherwise) are placed to minimize Euclidean distance to (dynamic) client nodes or areas of interest in the environment and 2) approaches that give explicit treatment to the wireless communication aspect of the problem.

2.2.1 Works Related to Coverage of Client Vehicles

The common thread amongst these works is that they are primarily concerned with minimizing physical proximity to clients, points of interest in the environment, or to a desired target state.

Facility Location and its Coreset Approximations

The facility location literature solves the question of how best to place a fixed number of k facilities (such as ATM teller machines) such that for a given number of n users (such as clients wanting to use the ATM machines) the maximum distance that any user is from its closest facility is minimized. A common name for this class of problems is the k-center problem. Our objective of communication coverage over client agents thus has a natural connection to this body of literature. Although, as we explain in the body of the thesis, the dynamics of our client vehicles, and the connected nature of the mobile base stations who must provide a communication network over the entire system, require us to substantially generalize the problem statement beyond the k-center problem.

Many works have studied the k-center problem. The well-known book [111] defines

the k-center problem for different distance metrics as well as some common approximations to this problem. The paper [64] provides some general applications of the k-center problem to applications on random graphs. It is well known that the k-center problem is NP-hard, meaning that the complexity of its solution grows exponentially in the number of users n. For example, [81] investigates the NP completeness of this problem for the case of k line segments and argues for the extension of the results to the case where these facilities are points. Because of the well-known difficulty of solving the k-center problem for large problem sizes, much effort has been dedicated to finding faster, approximate solutions. Notably, Gonzalez [53] has provided a well-known linear time algorithm for solving the k-center problem up to a constant error factor of 2, but states that for any factor smaller than 2, constant factor solutions do not exist. Therefore tighter error bounds require solutions that are dependent on the input.

In particular, a tool from computational geometry called *coresets* has been extensively studied for approximate solution of the k-center problem. These coresets can be constructed in time that is linear in both n in k, and returns a small set of size roughly $O(k/\varepsilon^d)$, i.e., independent of n. We then run exhaustive search algorithms, approximations, or heuristics on these small representative sets. Approximation algorithms in computational geometry often make use of random sampling [16], feature extraction, and ϵ-samples [63]. Coresets can be viewed as a general concept that includes all of the above, and more. A comprehensive survey on this topic by Agarwal, Har-Peled, and Varadarajan can be found in [4]. The work by Badoiu, Har-Peled, and Indyk [7] presents a comprehensive derivation of coresets for the k-center and k-median clustering problems in Euclidean space. The work on coresets that is most relevant in our context are for k-center [33, 62]. These approaches provide a much tighter $(1 + \varepsilon)$ error bounds than constant factor solutions.

The majority of the references cited above center around static problems where the users in the system are viewed as stationary points. For the problems treated in this thesis we are interested in the case of dynamic users, or clients. Therefore any approximations that we employ should ideally take this mobility into account. The

focus of part of the thesis is to improve the static coreset approximation to explicitly account for the dynamic nature of the client vehicles under constraints. For the original k-center without any constraint, such coresets were suggested in [15, 32, 61]. In particular, the work by Timothy Chan in [15] and Grahling et.al in [55] develops what is referred to as a *dynamic* coresets where input points (for which the objective is to find a sparse representative set) are being added and deleted from the set. Similar to the problem in this thesis, the coresets developed in [15] are for application to the k-center problem that can be updated in time $\log(n)^{O(1)}$ for constant k and ε. The *streaming* problem is similar in that points may be added to the set and only a limited amount of processor memory is allowed to be used. While one can imagine using these tools to account for input points that move in a continuous fashion (not as discrete insert and delete operations) by deleting "old" coreset points corresponding to previous client positions and adding "new" coreset points corresponding to updated client positions, this approach leads to coresets that may change dramatically (in terms of represented client positions) between iterations where router placements are computed.

An important departure in the current thesis from the work in [15] is that our coreset can be used for approximating the distances of the clients to *any* k servers, rather a *specific* configuration of the k-center of the clients or their coresets. This makes our coreset useful for solving the k-center problem with additional constraints, as needed to accommodate client dynamics, or when maximum client-server distance is only part of the optimization function as in our case where a connectivity constraint over the entire network of clients and routers must also be optimized.

Keeping up with Dynamic Clients

The mobility of our client agents must be explicitly handled when optimizing for updated router positions. We take inspiration from the problem of reachability in the field of control for systems under unknown but bounded disturbances. A thorough development of reachability for dynamic systems, where the goal is to use control authority to drive the system into a desired state (which can be position, a desired

velocity, etc) in the presence of unknown but bounded disturbances can be found in the seminal work of [8]. In our formulation we consider the system to be the entire network of routers and clients, where our desired state is a connected state for the network, and the objective is to use control authority over the routers to maintain connectivity over clients. For our problem, the movement of the client vehicles is assumed to be modelled by a Linear Time Invariant discrete-time dynamical model and client motion is treated as a bounded but unknown disturbance on the network. We find inspiration for our treatment of the problem from applications of reachability to differential games as described in [10]. As a topic of future work we are interested in expanding to general nonlinear dynamic models for our clients, and the recent paper [68] provides a few key results regarding the computation of attainable positions for general dynamics system. However, we currently restrict our attention to LTI systems.

More generally, reachability techniques have been used for control of complex dynamic systems [30, 50], and in achieving of feasibility of a target state over long or infinite horizons [9]. Connectivity for an adversarial agent is investigated in [107] for a single agent that must be tethered to a base station. The paper by [94] investigates the feasibility of connectivity maintenance of a network of vehicles with second-order dynamics using tools from reachability. However, the guarantees derived in [94] focus on fixed, known *a priori* network topologies. The element of uncontrolled motion in the current thesis precludes fixed communication assignments between agents that can be maintained throughout.

A distinguishing characteristic of the current work is that it targets connectivity over multiple clients whose motion is uncontrolled, and where the amount of control assumed for our communication routers is limited. This problem formulation most closely approximates differential game type approaches to reachability. We decide to take a worst-case approach where we attempt to provide the best service even in the case that client vehicles are moving at their maximum velocities and thus imparting the largest possible disturbances on the system as in [10]. The approach described in the recent paper [86] discusses mitigation of the conservatism that comes along with

optimizing the worst-case scenario while preserving performance guarantees on the system. For the general case scenarios that we target, where no additional information beyond knowledge of client dynamics and max velocity is assumed, we are confined to worst-case scenarios in order to maintain our guarantees. However, an intermediate approach that trades off increased availability of information over the clients for less conservatism is a fruitful avenue of future research.

The unpredicted motion of the clients introduces many challenges and necessitates the ability of routers to handle frequent changes in the topology of the network. For this reason we decide to use a centralized approach over a decentralized approach, that provides complete flexibility for the mobile clients, while maintaining maximum utility of the fleet of routers.

The requirements of communication connectivity mixed with the need to accommodate uncertainty and dynamics limitations in our problem requires tools from both fields which results in the reachable k-connected centers problem (presented in Chapter 3), which to the authors' knowledge is a novel formulation, that incorporates uncertainty, and dynamic and communication limitations for heterogeneous systems of controlled routers and uncontrolled clients.

Distributed Approaches to Client Coverage

In the final part of this thesis we consider providing coverage to a set of client vehicles that are either static or moving slowly relative to the router vehicles. The development of distributed control of groups of robots working collaboratively to achieve a task has been a research focus in broad ranging fields including dynamic routing problems [39, 90], collaborative construction tasks [118], modelling of biological systems, and coverage [23, 98]. In many of these applications communication across the network is an important and challenging problem. The paper [5] concerns formation control of agents under communication constraints. Other work concerns using a communication tether to link a ground, or base station, to an exploring agent [13, 105]. The paper [40] addresses the communication problem by integrating information theoretic measures into the objective function and demonstrates this approach on a

chain configuration of mobile robots. A previous paper by the current author and colleagues [48], investigates the use of gradient methods for improving a proximity-based communication metric in environments with infeasible, or "no-fly" zones.

A second challenge we address in this part of the thesis is to ensure that aerial vehicles will never move to disconnect the communication graph. This is a difficult problem in a distributed system because each agent's controller only accounts for local information and the connectivity status is a global property of that graph. Other research efforts have focused attention solely on the problem of maintaining connectivity for distributed systems [22,83,105]. Many of these works use distributed algorithmic methods of checking the connectivity of the graph via gossip algorithms, local minimum spanning trees, or other iterative approaches. Our approach allows for a continuous method of connectivity maintenance using local information at the expense of a more conservative controller. Less conservative approaches to this problem could involve a combination of our distributed controller for communication optimization and an algorithmic check for graph connectivity such as the work in [22].

2.2.2 Works Related to Wireless Communications Tasks

Typically, approaches to solving the problem of maintaining communication over a multi-robot team, rely on modelling these systems as graphs, where every robot is a node and edges correspond to communication links between pairs of robots defined according to a pre-specified communication model. The "pre-specified" models used in the current literature largely come in two flavors: i) the Euclidean disk model and ii) models derived using stochastic sampling of the wireless channels. Each of these methods has its strengths and deficiencies and the purpose of this thesis is to provide new solutions to the multi-agent communication coverage problem that circumvents key deficiencies of existing approaches and that provides measurable communication quality improvement in actual implementations. We now compare and contrast recent approaches in the literature to solving the communication coverage problem.

Euclidean Disk Model Approaches

The Euclidean disk model approach assumes an ideal model of communication where the quality of a link in the network (an edge on the communication graph) is given by the Euclidean distance of the link. In other words, the farther away two communicating agents are from one another, the worse the communication quality of their link and vice versa. While this correlation can be shown in practice for open-space environments, it does not capture reality for general environments [52]. In particular it does not capture environments where obstacles may cause shadowing, or attenuation, of the signals, and reflections of the same signal may cause interference and multipath fading. These phenomena are generally very difficult to predict and are not captured by the Euclidean disk model. Controllers based on this model are often open-loop and are thus oblivious to the actual quality of the communication across link between two agents.

While ignorance to the actual quality of a communication link between communicating agents may be an obvious pitfall of modelling signal quality using a Euclidean disk model approach, there are also many advantages. In particular, such a model allows for simple, often quadratic costs due to the Euclidean distance metric, and is amenable to graph-theoretic approaches to network connectivity. The works [25,80,97] investigate many important multi-agent coordination problems such as coverage where agents must deploy to survey an environment, while using Euclidean metrics or visibility graphs to ensure acceptable communication quality. The comprehensive book on multi-agent networks [82] studies graph-theoretic guarantees for systems where links in the graph are often based on Euclidean distance or disk assumptions (where the distance between the two nodes forming the edge is less than a prescribed value). Using similar theoretical tools to guarantee convergence rates, the works [66,89,106] study the problems of consensus and flocking where all agents in the network must agree on some state. The important question of connectivity of a communication graph, meaning that desirable communication rates can be supported over all links in the graph, has implications for the performance guarantees of many

multi-agent tasks. In particular, one approach to assessing network connectivity is based on graph-theoretic measures such as the Fiedler value, which is the second largest eigenvalue of the graph Laplacian [28, 83]. The question of maintaining connectivity also requires that the communication graph is robust to switching network topologies where the communication neighbors of any node may change over time. This is a question that we handle in the current thesis by explicitly considering client mobility over time.

Additionally, it may be important to maximize the *throughput*, or number of packets successfully received, over the network. The objectives of the paper [26] are very relevant in that a subset of capable routers are assigned to a group of less capable "regular" nodes (similar to our client nodes) and the objective is to find both *assignments* of routers to regular nodes (resource allocation) and *placements* of routers in order to maximize the network throughput. One of the main differences between the problem targeted by [26] and that addressed in this thesis is that we additionally assume that client agents (or "regular" nodes in [26]) are mobile and their trajectories are not known *a priori*, nor do we assume any control authority over their motion during the optimization. Other distinctions include the fact that in the current work, connectivity amongst the backbone of routers themselves must also be maintained and optimized.

As apparent from the plethora of advancements made in literature related to multi-agent communication-aware coordination under idealized assumptions on signal quality, the advantage of Euclidean disk model approaches is in its simplicity. It is this simplicity that allows for the development of multi-agent controllers that achieve many important tasks from coverage, to flocking, to rendezvous, and permits tractable cost formulations that can be provably optimized for maintaining network connectivity over all agents. However, in addition to the Euclidean disk model assumption, the aforementioned works often require fixed network topologies, or assume that all nodes in the graph are controllable, and do not mention the applicability of their methods to large systems. In this thesis we develop methods for maintaining connectivity over the entire communication graph by leveraging positional control

over only a subset of the agents in the network. Under this premise we also allow for changing network topologies, we provide time windows over which connectivity can be maintained, and we develop tools to generalize our algorithms to large-scale systems where the number of uncontrolled agents in the graph grows larger.

Although the assumption that communication quality can be predicted deterministically based on Euclidean distance alone allows for the development of provably optimal router placement algorithms and related performance guarantees, this assumption unfortunately does not accurately capture real wireless phenomena. This fact has been largely studied and verified experimentally. We discuss these related works in the next subsection. In this thesis we develop algorithms based on the Euclidean disk assumption that are later generalized to capture realistic communication models.

Real Wireless Channels Don't Follow the Euclidean Disk Model

Unfortunately, the problem of mapping signal quality between two communicating agents to their spatial positioning is a very difficult problem [52, 79]. In reality, variation in signal quality over tens of centimeters of displacement can be larger than the variation over meter-scale displacements which is in stark contrast to that predicted by the Euclidean disk model. It is well known that the environment also has a significant affect on the propagation of signals and the resulting received power of these transmissions [104]. In fact, any prediction of signal quality is often attempted via the use of different models such as Rayleigh or Ricean fading models, and/or Gaussian process models, whose parameters must be fitted using knowledge of the environment and the obstacles within it [35, 75, 79]. Given the complexity of the true behavior of signal quality over different spatial positions, many recent efforts have adopted techniques to sample the wireless channel stochastically. This is the focus of the stochastic sampling methods discussed in the next subsection.

Stochastic Sampling Methods

Many recent efforts have focused on giving the communication over each link in the network a more realistic treatment. In particular these efforts adopt models of signal strength that incorporate *pathloss* which is the attenuation of the signals due to propagation over long distances, *shadowing* which is attenuation due to obstacles and certain materials that the signals traverse through, and *fading* which is a complex phenomena based on signal interference due to reflections in the environment. Often times the errors in these models are accounted for by adding stochasticity, where the variations in signal strength are assumed to come from different distributions such as the Rayleigh or Ricean fading models [75, 76] or Gaussian processes [79], and where samples of the signal strength at different locations are used to fit parameters for these distributions. The paper by Johansson [75] derives a sampling pattern and a theoretical bound on the number of samples that must be taken in order to fit the model parameters. His assumption is that these samples come from a Rayleigh or Ricean fading model, which he specifies is based on the environment (indoor/outdoor) and can change from room to room. The authors of the paper [117] investigate the use of instantaneous signal strength and stochastic models of communication quality in order to "co-optimize" communication goals and higher-level task goals as a vehicle traverses a pre-specified path. Using these predictions on the signal quality, Fink et. al. [34–36] derives routing protocols to maintain specified rate demands between one node and any other node in the network. In his paper [74], LeNy combines general coordination objectives such as coverage, with communication goals, by using a Simultaneous Perturbation Stochastic (SPSA) algorithm for sampling the channel quality. He makes the assumption that the channel capacity is directly observable and also assumes that fading and shadowing effects on the channel can be modelled by a zero-mean random variable.

While the aforementioned approaches are a large step forward towards adapting robotic multi-agent tasks to incorporate realistic communication, they are not quite amenable to implementation in general, real-world environments. Taking a closer

look at stochastic sampling methods, the main objective of sampling the wireless channels is to infer what the signal quality will be in the local environment of the robot whose position must be optimized. Figure 1-2 shows a schematic drawing of what the true directional signal strength map might look like for a link between two agents. Stochastic approaches sample the signal strength at discrete points (shown as red dots in Fig 1-2) by physically moving the agent to these locations. This is an expensive procedure as these directions are often counter-productive to the primary coordination task.

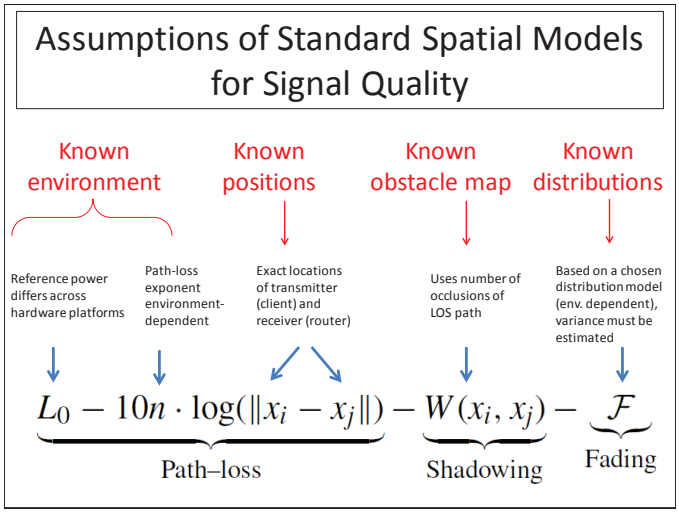


Figure 2-1: A standard model for predicting signal quality (signal strength) along different spatial positions [35,79], and the related assumptions common to approaches utilizing these models.

Therefore the works mentioned in this section attempt to *predict* the remaining portions of the signal strength curve by relying on models for the fading and shadowing effects, where the parameters describing these models must be fitted using the gathered samples. Therefore these models necessitate assumptions that the environment is known (necessary for choosing an appropriate model and for fitting the model parameters), the environment is static (otherwise the samples are no longer reflective of the channels), and that the positions of both communicating agents is known (necessary for modelling pathloss effects). Figure 2-1 demonstrates a standard model used for predicting the signal strength curve as a function of node positions, the environment map, transmit powers, and collected RSS samples. These assumptions are

often prohibitive for real-world robotics applications that must be energy efficient, adaptive to dynamic and/or unknown environments, and deployable in GPS-denied areas where localization may not be available. As an example, the Association for Unmanned Vehicle Systems recently identified autonomous flight in GPS-denied environments as a primary challenge due to its application to disaster relief scenarios. Therefore the assumption of a known environment is particularly prohibitive. The papers [1, 6, 12] provide a response to this challenge in the area of agile robot deployment for exploration of unknown environments. We aim to maintain this level of applicability to general environments in communication-aware domains as well.

Realizing the penalties associated with predicting local signal strength, the work by [112] adopts a fine-scale approach for sequentially sampling the channel along small local trajectories in order to find a configuration of positions that avoid deep fades on the channels. However, this approach necessitates physically moving the robots to sample signal strength along counterproductive directions. The work by Twigg et. al. in [108] presents an actual example of mapping signal strength throughout the environment by conducting an experiment in hardware to do so. This paper shows how to estimate the RSS gradient and angle of arrival in the presence of small scale fading, by taking several samples in 2-D and fitting a plane (slope equal RSS gradient, direction of arrival comes from the angle of the plane). While this provides a distinct advantage over other works in this category that offer simulation results or data driven results (but do not provide results from controlling these systems using real-time wireless channel feedback), this work requires generating a signal strength map of the environment. This makes it difficult to handle dynamic surroundings and precludes quick response deployments where there is no time for mapping the whole environment. This paper relies fully on RSS or signal strength measurements to construct this map. Unfortunately, using signal strength alone can be a misleading metric as many times the variation in signal strength can be within noise levels. In contrast, our methods for handling realistic wireless communication use the full channel information including amplitude (signal strength) and phase of the signal. More information about this distinction and the advantages of using the entire channel

information can be found in Chapter 5.3 of this thesis.

Comparison of Current Methods

The plots in Figure 1-3 demonstrate the results of an actual implementation of both the Euclidean disk model method and Stochastic sampling SPSA method (similar to the method used by LeNy et. al. in [74]). In this scenario three client vehicles demand a specific communication rate. A WiFi equipped roomba robot, or mobile router, was tasked to find a position that would satisfy the demands of the three client nodes by supporting the requested rates on each link in the network. The trajectory of the mobile router (blue arrows) demonstrates key strengths and deficiencies of each of these methods. In the case of the Euclidean disk method, the mobile router converges quickly to a solution without expending energy exploring counter-productive paths, however the converged position is behind an attenuating obstacle with respect to client3 and thus the required communication rate is not satisfied. Because the Euclidean disk approach is oblivious to the actual wireless channel, the mobile router cannot find a solution that satisfies actual client demands. In contrast, an implementation using the stochastic sampling method does find a solution that satisfies all client demands, but not before exploring along many random and counter-productive paths. A key observation is that the aforementioned sampling approaches are all based on signal strength, which can be a misleading metric in areas of low signal-to-noise ratio where changes in the signal strength can be below the noise level (as shown in Figure 1-3). A more ideal approach would allow for energy-efficient paths, that do not necessitate counter-productive exploration of the signal strength map, while allowing for satisfaction of actual client demands over the network. Furthermore, this approach should retain the mathematical simplicity of those based on the Euclidean models, but would be capable of meeting actual communication demands in general, unknown, and dynamic environments.

In particular, as part of this thesis we develop a new approach to finding positions for a team of mobile routers such that, similar in spirit to [35], requested rates to client vehicles are maintained. Unlike [35] however, we use an additional piece of

information from the wireless channel, *the phase*, in order to measure the directional signal strength map *directly* without predicting this map by using models that are tied to a specific known environment. Our approach also solves the problem of deciding which routers to move and along which directions to move them in order to serve competing client demands, similar in spirit to [112]. Although in contrast to the referenced approach, we do not necessitate movement along random sampling directions in order to make these network-wide decisions. We describe works in the wireless literature that form the background for our approach in the following section.

Wireless Signal Strength Mapping

We develop a novel method of directional signal strength mapping for multi-robot communication networks. Our method uses the entire channel information, amplitude (signal strength) and phase, that in combination with signal processing techniques, allows us to acquire rich mappings between relative heading angles (of each robot router) and signal quality directly from measurements of the wireless channel (between any two communicating nodes). Our method builds upon a rich body of literature in the fields of signals and wireless communication and we build upon many of the methods developed for applications in these fields. Methods of signal processing, such as spectral processing of signals [103], have applications in many areas including communication. Notably, the method of beamforming [110] uses signal processing in conjunction with an array of clients (such as wifi antennas) to provide spatial filtering of the received signal. The objective of beamforming is to estimate the signal arriving from a desired direction, while attenuating the effects of noise and other interfering signals. Generalizing this technique to account for receiving multiple signals from distinct sources is the subject of the MUSIC algorithm [96]. MUSIC allows for the ability to distinguish signals from multiple distinct transmitters, or in the case where the environment may affect the signal propagation via reflections or multipath, multiple receptions of the same signal from different directions (for example a reflected path). Understanding how to attain this directional information using a moving platform is the subject of Synthetic Aperture Radar (SAR) [37]. This

is angle-of-arrival (AoA) technique that allows even a single-antenna mounted on a flying aircraft or satellite to emulate a multi-antenna array and has been used in applications such as imaging, by reflecting signals off of an environment (much like radar).

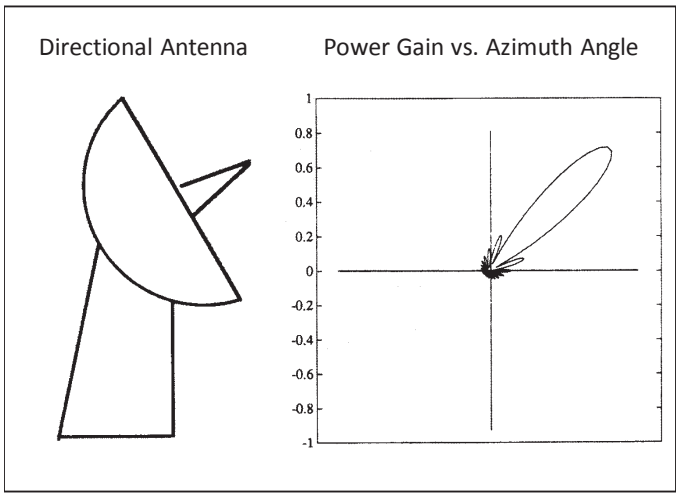


Figure 2-2: A directed antenna (left) and the result of beamforming applied to the received signals where the signal received along a particular direction is largest (large lobe indicates high signal strength) [110].

Past literature has leveraged SAR and/or arrays of antennas for object or transmitter localization [70, 114], terrain imaging [119], seeing through walls [2, 20], and tracking [91]. In addition, techniques such as [17] investigate ways to make the equipment necessary to provide the spatial resolution for the angle-of-arrival techniques for radar imaging applications. Recently, RF-Compass [113] proposed leveraging SAR to navigate robots equipped with RFID tags towards an RFID-tagged object. However, RF-Compass measures angle of arrival from the RFID tags to a centralized custom RFID reader and does not involve direct communication between the RFID tags. For example, recent papers by [67] and [115] demonstrate the benefit of AoA techniques for localization of Wi-Fi transmitters. These use stationary multi-antenna receivers to locate the transmitter with sub-meter accuracy. Unfortunate for the robotics community, many of these techniques require bulky, specialized hardware such as customized software radios, and are thus difficult to place on small, agile, mobile platforms that are ubiquitous for robotics applications. Also, most SAR applications are geared to-

wards radar-type problems where signals are transmitted and processed by the same node, and these works do not make guarantees on signal quality from a single transmitter across spatial directions. For an adaptive communication network composed of small mobile router robots, we need a light-weight, single-antenna system that can produce the same rich directional information as provided by SAR, but where the signals to be processed are actually two-way transmissions (unlike radar) and can be implemented with off-the-shelf components.

Recent work by Dan Halperin [59] provides a method of computing an accurate prediction of the current rate capacity on a wireless channel. In this work he develops a metric called the *Effective Signal to Noise Ratio* (ESNR) which is shown to provide a one-to-one mapping to the maximum rate supported on the link and this information can be transmitted along with the packet similar to RSSI (although RSSI is notoriously unreliable as a measure of instantaneous rate capacity of a link [59]). Moreover his recent tool release [60] allows for the full channel state information (CSI) to be transmitted along with each packet (amplitude and phase information) using an off-the-shelf Intel wifi card. Using the full CSI, we develop a method for generating directional signal strength data that maps the relative heading of a robot router to signal quality, using off-the-shelf components, on small iRobot mobile platforms. We generate these maps as each router moves along an arbitrary direction (which obviates the need to randomly explore the environment) and uses two way communication between clients and routers (unlike radar). To our knowledge this is the first time the capacity of rich directional signal strength maps has been developed and implemented for small mobile robotic platform applications.

Lastly, we must use the feedback from each wireless channel to control our ad-hoc network of robot routers for satisfying client demands. The concept of satisfying competing client demands over a wireless network is not in itself novel. The work by [92] explores the use of beamforming to increase throughput by emulating multiple access points on the same channel, however this work concerns a static infrastructure of routers. The work by [35] uses router mobility to satisfy client rate requests but uses only signal strength (RSSI) measures bolstered by prediction models that are specific

to the environments they operate in. The combination of using rich mappings of signal strength to all relative directions for each router (by using both amplitude and phase components of the signal), along with a simple-to-implement yet powerful optimization framework that leverages router mobility and channel feedback for placing the routers, results in the attainment of adaptive communication networks with a new level of both simplicity and performance in real-world, and general, environments.

2.3 Preliminaries

Throughout the thesis we view the entire system of clients and routers as a graph. Under this treatment, "nodes" are the vertices of the graph and these can be either robot routers or clients. Edges represent the communication link between any two nodes. Formally, the definition of a graph is as follows:

Definition 2.3.1 (A Communication Graph $G(V, E)$)**.** *Given a set $C \subseteq \mathbb{R}^d$ of router positions, and a set $P \subseteq \mathbb{R}^d$ of client positions, a graph is an ordered pair $G = (V, E)$ where $V = C \cup P$ is a set of vertices or nodes of the graph, and E is a set of edges such that any two nodes connected by an edge are communicating. We assign* edge costs*, $h_{ij} > 0$, to be the cost for communication along the edge e_{ij} connecting nodes $v_i \in V$ and $v_j \in V$. We use the names "communication graph" and "communication network" synonymously throughout the thesis. We also sometimes refer to an edge in the communication graph as a "wireless link."*

The mathematical definition of the edge costs between communicating nodes is one of the addressed problems of the thesis. For the Euclidean disk model approach, this cost is directly given by the Euclidean distance between the two nodes such that $h_{ij} = \sqrt{(v_i - v_j)^T (v_i - v_j)}$. In Chapter 5 of the thesis we define a new edge cost that takes into account real-time wireless channel feedback. In the sequel we will adopt the following shorthand notation for the Euclidean distance between two points $p \in \mathbb{R}^d$ and $q \in \mathbb{R}^d$:

$$\text{dist}(p, q) = \sqrt{(p - q)^T (p - q)} \tag{2.2}$$

and the distance between a point $p \in \mathbb{R}^d$ and a set $Q \subseteq \mathbb{R}^d$:

$$\text{Dist}(p, Q) = \min_{q \in Q} dist(p, q) \tag{2.3}$$

We make the following assumptions on the communication graph:

Assumption 2.3.2 (Reliable Communication)**.** *We assume that only router-router and router-client communication is reliable.*

This assumption is important since we view the client agents as non-cooperative in the sense that they are not required to adjust their positions to accommodate communication. We define a spanning tree and a minimum bottleneck spanning tree over the graph:

Definition 2.3.3 (MBST)**.** *Given a set $C \subseteq \mathbb{R}^d$, a* spanning tree *of C is a tree $G(C, E)$ that connects all the centers (points) in C, whose edges are $E \subseteq C^2$. A* bottleneck edge *is the longest edge in a spanning tree, i.e, that maximizes* $\text{dist}(c, c')$ *over $(c, c') \in \mathcal{T}$. A spanning tree $\mathcal{T}^*$ is a* minimum bottleneck spanning tree (or MBST) *of C, if C does not contain a spanning tree with a shorter bottleneck edge. We define $b(C)$ to be the length of the bottleneck edge of the MBST of C, i.e,*

$$b(C) := \max_{(c,c') \in \mathcal{T}^*} \text{dist}(c, c'). \tag{2.4}$$

Throughout the thesis we will refer to communication demands over the network and communication quality of a specific communication link. For reference, we present our meaning of this terminology here:

Definition 2.3.4 (Communication quality q_{ij} of a link)**.** *We refer to the communication quality of a link between communicating nodes v_i and v_j to be q_{ij}. This quantity is a dimensionless averaged signal to noise ratio related to the* Effective Signal to Noise Ratio *(ESNR) presented in the paper [59]. This quantity has a direct mapping to the instantaneous maximum rate (throughput) in Mb/s that a link can support and thus the ESNR and max rate of a link are used interchangeably in this thesis. In sections*

of the thesis where the Euclidean disk assumption is used, the communication quality is assumed to map directly to the Euclidean distance is not dependent on feedback from the wireless channel.

Definition 2.3.5 (Communication demand $\tilde{q}_j$ of client j)**.** *The communication demand $\tilde{q}_j$ of client j is the requested rate of client j. If $\tilde{q}_j \leq \max_i q_{ij}$ then we say that the client's demand is satisfied.*

Chapter 3

Proximity-based Communication for Dynamic Clients with Unknown Trajectories

3.1 Introduction

In this chapter of the thesis, we derive a problem formulation for finding router placements such that all clients receive the maximum communication quality provided by a fixed number of routers. We give explicit treatment to the case where client agents are mobile, and we provide an algorithm to solve for router placements that maintain connectivity over a computable time window. Specifically, we are interested in maintaining a network such that each client can send and receive messages to and from all other clients in the system but we do not assume client cooperation with the routers to maintain the network. This is an important assumption as it allows the mobile clients maximum freedom to change their motion plans as necessary, but it also makes communication maintenance significantly more challenging. To see why this is an important assumption, take the pedagogical example of your cellular provider. The service that you expect from the provider is the ability to communicate to any person in your contact list, without needing to know where that person is (even if they

happen to be in the same room as you, since from the point of view of the service provider this is irrelevant), and without having to alter your path (say, you wish to communicate while driving to work) in order to communicate reliably with that other person. Similarly, we wish to provide this level of freedom to our robot clients such that they may communicate to any other client in the network reliably without explicit cooperation (ie. they are viewed as *non-cooperative* agents). In this spirit, we assume the two following constraints: 1) every router can communicate reliably with any other router or client at a distance of at most R, where R is a specified communication radius and 2) a client can only communicate reliably with its nearest router. We aim to develop on-line position control for the routers such that they provide communication over the entire system, and such that the amount of time that these positions are guaranteed to maintain a connected network is maximized for the given communication radius and vehicle velocities.[1]

In order to achieve the goal of controlling robot routers to provide communication under these assumptions we must address questions such as, given n mobile clients can we allocate k mobile routers such that there exists a connected communication network? How would we compute this allocation and moreover, how can we maximize the amount of time that we are guaranteed to preserve mutual connectivity despite unpredicted client movement? What are the tradeoffs between the communication radius R, number of mobile routers k, maximum client velocities, and their moving freedom? In this work we define a formal framework, design provable algorithms, and provide empirical case studies that aim to answer these types of questions.

3.1.1 Assumptions

In this chapter of the thesis we make the following assumptions:

(i) Signal strength can be mapped directly to the Euclidean distance between two communicating nodes. We refer to this as the *Euclidean disk assumption.*

(ii) Current positions of clients and routers are known at the time that a new con-

[1]Part of this work has also been presented in [45].

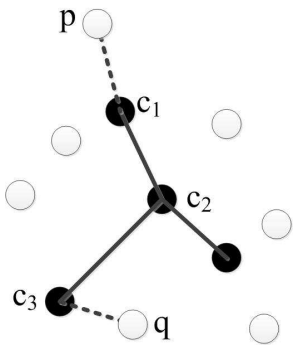


Figure 3-1: The point p can communicate with q through the path of centers p, c_1, c_2, c_3, q. The minimum required distance r for this communication is the longest edge (c_2, c_3) in this path.

figuration is computed in a centralized manner, although future client positions (more generally, their trajectories) are unknown.

(iii) Client communication demands are equal across the network.

(iv) We assume the dimension $d > 0$ to be part of the input.

(v) A communication network $G = (V, E)$ is *connected* if no edge exceeds a given radius R such that $\max_{(i,j)\in E} dist(v_i, v_j) < R$.

We later show, in Chapter 5 of the thesis, that the assumption of signal quality directly mapping to Euclidean distance can be relaxed by incorporating feedback from the wireless channels across the network. We show that the algorithms for router placement in this chapter of the thesis readily extend to the most general case, that uses real-time wireless quality feedback, via a reduction from the most general case to the specialized problem formulation in this chapter (see Section 5.5).

3.1.2 Summary of Problems Addressed

We consider that a communication network is *feasible* for a communication radius of R, and a given configuration of mobile routers C under the two constraints given in the opening paragraph if every pair of clients can mutually communicate using C. Communication occurs via message passing from a client p to its nearest router c_1, and then via multi-hop through the connected network of routers c_2, c_3 before being

delivered to its destination client q similar to the figure in 3-1. The minimum R that is needed to use this routing path is the maximum distance between any two vehicles sharing a communication edge, see Figure 3-1. We consider communication to be feasible for a configuration C if every pair of clients p, q have such a path that requires a communication radius of no more that R. This constitutes our first problem

Problem 3.1.1 (feasibility problem)**.** *Is communication* feasible *for a configuration C of mobile routers?*

Note that the solution depends on all the possible spanning trees of the set C. If the required communication radius is actually r where $r < R$, this implies that clients p and q can move while still preserving mutual connectivity which motivates the problem of finding the *minimum* such r:

Problem 3.1.2 (k-connected center problem)**.** *Viewing routers as* centers*, what is the minimum value of r such that there is a feasible set of positions C for the mobile routers, and what is that C?*

We note that even if the number of centers (routers) and mobile clients is the same ($n = k$) the solution is not trivial due to the interdependencies arising from the connectivity requirement of the centers (see Figure 3-2). Finally, we incorporate the dynamics of the vehicles, ie. control effort limitations for the routers and maximum velocities for the mobile clients, as well as the maximum communication radius R to answer the following question.

Problem 3.1.3 (reachability problem)**.** *What areas can each client and mobile router move to (reach) while keeping the network connected, and for how long can we guarantee that a connected configuration is maintainable given a set of routers C and their allowed control inputs, and maximum client velocities?*

3.1.3 Results Snapshot

We develop an algorithm that solves for optimal mobile router positions that provide a connected communication network if such a configuration exits, and that maximizes

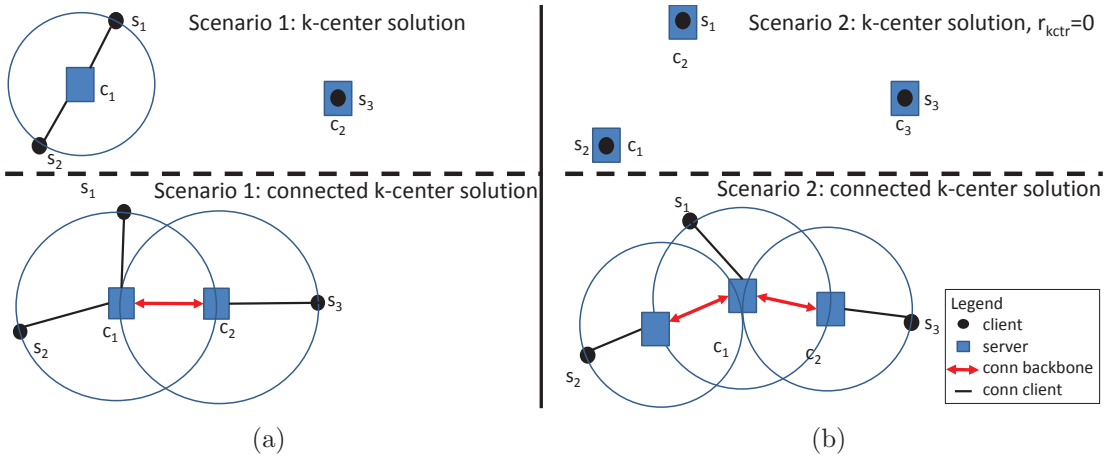


Figure 3-2: Schematic drawings showing the differences between the k-center and connected k-center solutions for the n=3, k=2 case (a) and the n=3, k=3 case (b). Note that clients have the same positions for all scenarios depicted above.

reachability for given vehicle velocities. Our algorithm also answers the feasibility problem in $O(n+(k\log k)^{4/3})$ for $\mathbb{R}^3$ for a given C by the observation that a Euclidean minimum spanning tree for C (that minimizes the sum of the length of edges) also minimizes the longest edges among all spanning trees of C [3]. For n mobile clients and k centers this algorithm has two flavors, it can be used to provide 1) an exact solution C^* with optimal cost h^* in $n^{O(k)}$ time, or 2) a faster approximate solution that takes $O(\frac{k}{\varepsilon^d})$ and returns a solution C with cost h such that $h \leq (1+\epsilon)h^*$ where $\epsilon > 0$ is an input parameter depending on the desired solution accuracy. We address the approximate solutions to this problem in Chapter 4.

We use a modified version of k clustering that we refer to as *k-connected* clustering to optimally assign k subclusters of client vehicles to each router, where the distance between any cluster of clients and its assigned router, as well as router-router distances, are minimized. The result is a position controller that drives each router to optimal positions that minimize distance to the non-cooperative group of clients, and that maintains reachability of a connected configuration if such a configuration exists for the given problem variables. For applications where the number of vehicles and/or fast vehicle dynamics makes computation of the exact solution prohibitive in real-time environments, we employ alternative approximate solutions that compute the optimal communication vehicle positions over a *subset* of the clients, and that

have asymptotic error bounds on the performance (see Chapter 4).

We provide an *expiration time*, or time window over which a configuration C of communication vehicle positions is guaranteed to maintain connectivity of the network given control input limitations and maximum client velocities. Our empirical results show that for a scenario of 5 clients moving at a constant $1m/s$ and 2 centers moving at $1.5m/s$ the expiration time is $t_e > 60$s and the exact solution can be computed in 38s using a Matlab implementation of Algorithm 1 on a Dell Latitude E4300 with an Intel core 2 processor.

3.1.4 Notation

3.2 Problem Formulation based on the k-Center and Reachability Problems

We now turn our attention to deriving a problem formulation, or cost, such that this cost can be optimized to find new router positions that satisfy a metric over the network. We base our metric on the classic k-center problem for facility location, and the well-known reachability problem for handling client dynamics.

3.2.1 Problem Statement

We wish to provide communication coverage to n *mobile* clients with positions $p_{j_t} \in \mathbb{R}^d$ at time $t \geq 0$, that are moving over unknown trajectories. We assume the following discrete time LTI model for client dynamics, for all $j \in [n]$, the input w_{j_t} is unknown but bounded to the uncertainty set W_j where Q_j is a given $d \times d$ positive definite diagonal matrix:

$$\begin{aligned} &p_{j_{t+1}} = p_{j_t} + w_{j_t},\ j \in \{1, \ldots, n\}, \\ &w_{j_t} \in W_j = \{w \in \mathbb{R}^d :\ w^T Q_j w \leq 1\}. \end{aligned} \tag{3.1}$$

We provide this communication using a set of mobile routers with positions $c_{i\tau} \in$

Table 3.1: Common Notation for Chapter 3

k centers	*is synonymous with*	k routers.
$G(V, E)$	$\triangleq$	Communication graph with nodes V and edges E.
E_C	$\triangleq$	Edges between centers in the set C.
p_j	$\triangleq$	Client j position in $\mathbb{R}^d$.
c_i	$\triangleq$	Robot router i position in $\mathbb{R}^d$.
z_{ij}	$\triangleq$	Relative state between two nodes $v_i \in V, v_j \in V$.
Ω_i	$\triangleq$	Positive definite matrix defining the control set U_i for robot router i.
Q_j	$\triangleq$	Positive definite matrix defining the allowed disturbance set W_j for client-agent j.
X	$\triangleq$	Desired connectivity set.
$r(P, C)$	$\triangleq$	k-center cost for clients P and robot routers C.
$r_b(P, C)$	$\triangleq$	k-connected center cost for clients P and robot routers C.
$b(C)$	$\triangleq$	Bottleneck edge of minimum spanning tree over $G(C, E)$.
t_e	$\triangleq$	Expiration time.
Z	$\triangleq$	The set of relative states z_{ij} for all edges in E.
$\mathcal{C}(P)$	$\triangleq$	Set of all router positions providing connected configurations for P.
τ, $\tau + 1$	$\triangleq$	Current update time (in seconds) and next update time.
$\mathcal{T}$	$\triangleq$	Minimum spanning tree over the set of centers C.
$\boldsymbol{\Omega} = \{\Omega_1, \ldots, \Omega_k\}$	$\triangleq$	Set of positive definite matrices Ω_i for $i \in [k]$.
$\boldsymbol{Q} = \{Q_1, \ldots, Q_n\}$	$\triangleq$	Set of positive definite disturbance set matrices Q_j for $j \in [n]$.
$\mathcal{Q}(P, k)$	$\triangleq$	Set of possible clusterings of points in P.

$\mathbb{R}^d,\ i \in \{1, \ldots, k\}$, whose movement we can control via the input $u_{i\tau}$. The set U_i specifies the vehicle control limitations where Ω_i is a $d \times d$ positive definite diagonal matrix:

$$\begin{aligned} c_{i\tau+1} &= c_{i\tau} + u_{i\tau},\ i \in \{1, \ldots, k\}, \\ u_{i\tau} &\in U_i = \{u \in \mathbb{R}^d :\ u^T \Omega_i u \leq 1\} \end{aligned} \tag{3.2}$$

We use different notations t and τ where t is time in seconds, and τ is the time in number of iterations for any algorithm that updates the router positions. We can think of iterations as a sequence of times $(\tau_1, \tau_2, \ldots)$ where $k = 1, 2, \ldots$, and $\tau_{k+1} = t_k + B$ and t_k is the time in seconds of the current position update for the routers and B is the elapsed time in seconds until the next update. As a shorthand, we will use the notation τ and $\tau + 1$ to denote two consecutive updates. We denote the set of router positions at iteration τ as C_τ and the set of client positions at time t as P_t, and assume the dimension d is a constant. For clarity in the remainder of the paper we use the names *robot routers* and *centers* synonymously. We discuss generalizations of this problem to accommodate general discrete time LTI dynamics as a topic of current and future work in Section 7.2.2 of this thesis chapter.

We would like to keep the heterogeneous system in a *connected configuration*. The length of the minimum bottleneck edge $b(C)$ (Definition 2.3.3), and the maximum distance between any client to its closest center determine the smallest communication radius, r^*, needed to achieve a connected system. If this value is smaller than the maximum allowed communication radius R, then the configuration is connected. Formally,

Definition 3.2.1 (Connected Configuration)**.** *Let $P \subseteq \mathbb{R}^d$ and $k \geq 1$ be an integer. Let*

$$r(P, C) := \max_{p \in P} \mathrm{Dist}(p, C) \tag{3.3}$$

where $\mathrm{Dist}(p, C) = \min_{c \in C} \mathrm{dist}(p, c)$ is the closest point to p in the set C. For a given

set C, define

$$r_b = \max\{r(P, C), b(C)\} \tag{3.4}$$

If $r_b \leq R$ then sets P and C are in a connected configuration. We denote the set of all router positions providing a connected configurations for P by $\mathcal{C}(P)$. Mathematically, our desired connectivity set X is

$$X(G) = \{z_{ij} \in \mathbb{R}^d : \ z_{ij} = c_i - p_j, \|z_{ij}\| \leq R, \ \forall \ (i,j) \in E\} \tag{3.5}$$
$$\text{for some communication graph } G(P \cup C, E)$$

The desired connectivity set is always defined with respect to a graph G but we will drop this explicit dependence for readability. Note that our desired connectivity set is compact and can be re-written as the ball $X = \{z_{ij} \in \mathbb{R}^d : \ z_{ij}^T \Psi z_{ij} \leq 1\}$ where Ψ is a positive definite diagonal $d \times d$ matrix with $1/R^2$ on the diagonal. Moreover, if a connected configuration for a set P exists for some C then a connected configuration is feasible.

We would like to address the problems of 1) evaluating whether a connected configuration is feasible for a set P and C, 2) finding the configuration C for attaining a minimum value of r_b such that the system is connected and 3) choosing a new set of positions $C_{\tau+1}$ such that the centers maintain feasibility of a connected configuration from one iteration C_τ to the next $C_{\tau+1}$ given the vehicle dynamics models from (3.3) and (3.2) and the communication radius R. Our insight for formalizing the third problem is to view this problem as that of controlling the state of a dynamical system to remain within a desired set X under bounded but unknown disturbances, commonly known as a problem of *reachability* of the set X, [8, 11, 19, 93]. In particular our formulation is most closely related to the application of reachability to differential games as in [10]. From this perspective, we view the entire heterogeneous network as a single dynamical system where the objective is to use permissible controls ($u_\tau \in U$) for placing routers such that the network remains in a connected state despite client

mobility (viewed as an unknown but bounded disturbance). Specifically, we define a relative state for each communication link at every time $t \geq 0, 1, 2, \ldots$:

$$z_{ij_t} = s_{j_t} - p_{it} \tag{3.6}$$

and for some communication graph $G(V, E)$ we wish to control the updated positions of our centers such that for every edge $(i, j) \in E$ we have that $z_{ij_{t+1}}$ satisfies $\|z_{ij_{t+1}}\| \leq R$, and is thus in a connected configuration, for any $s_{j_{t+1}}$ satisfying (3.2). If admissible controls exist for each router to accomplish this task from one iteration to the next, we say that the desired connected state is reachable for given control limitations and client dynamics. For a set X, reachability can be formally defined as:

Definition 3.2.2 (Reachability of a set X)**.** *We use the definition of reachability of a set from [8] for a one-step horizon where a sufficient condition for X to be reachable from a state x (where the state evolution is given by equations of the form* (3.2)*) is $x \in X^*$ where X^* is defined as*

$$\mathcal{E} = \{x \in \mathbb{R}^d | x + w \in X \quad \textit{for all} \quad w \in W\} \tag{3.7}$$

$$X^* = \{x \in \mathbb{R}^d | x + u \in \mathcal{E} \textit{ for some } u \in U\} \tag{3.8}$$

Specifically, the condition that $x \in X^*$ asserts that for all uncontrolled but bounded inputs w to the system where $w \in W$, there exists some permitted control $u \in U$, such that the state x can remain in the desired set X.

We wish to find the set of relative states X^* such that the condition $z_{ij} \in X^*, \ \forall (i, j) \in E$, asserts that a configuration of routers from the set $\mathcal{C}(P_{t+1})$ is reachable. Therefore we aim to choose updated center positions $C_{\tau+1}$, and edge assignments $E_{\tau+1}$ such that for the graph $G(C_{\tau+1} \cup P_t, E_{\tau+1})$ we have that $Z_{t+1} \subset X^*$ for all $t \leq \tau + 1$, where Z is the set of relative states z_{ij} for all c_i and p_j forming an edge in E. We would like to find the maximum update time $\tau + 1$ such that this condition holds. In other words, we wish to find an maximize the expiration time t_e defined

below:

Definition 3.2.3 (Expiration time t_e)**.** *For given problem parameters* $\boldsymbol{\Omega} = \{\Omega_1, \ldots, \Omega_k\}, \boldsymbol{Q} = \{Q_1, \ldots, Q_n\}, R$*, and current vehicle positions* P, C*, we define the expiration time*

$$t_e = T_E(P, C, \boldsymbol{\Omega}, \boldsymbol{Q}, R) \tag{3.9}$$

to be a lower bound on the time window over which the set of positions C *is guaranteed to maintain* $G(C \cup \tilde{P}, E)$ *in a connected configuration, for any set of positions* $\tilde{P}$ *that evolves from* P *using the update equations in* (3.2)*. Using the expiration time, we choose our update time such that* $\tau_{l+1} - \tau_l \leq t_e$*.*

3.3 A Formulation for Optimizing Router Placements

Our problem formulation for optimizing router placements is based on the classic k-center and reachability problems. We first provide the necessary concepts from the related literature upon which we build in this section of the thesis.

3.3.1 Background on the k-Center and Reachability Problems

The well known k-centers problem finds the optimal placement of centers C^*, where $|C^*| = k$, that minimizes the maximum distance from any point in the set P to a point in C^*. Formally,

Definition 3.3.1 (k-Center Problem)**.** *The* k*-center problem for a set* P *returns the set* C^* *where*

$$C^* = \arg\min_{C} r(P, C) \tag{3.10}$$

and C^* minimizes this cost over every set $C \subseteq \mathbb{R}^d$ with cardinality $|C| = k$ [53, 64]. See Figure 3-3(a).

In the next section we present a generalization of the k-center problem that takes into account not only the maximum distance between a client and its nearest center, but also the maximum distance between centers making up the backbone of the communication network.

While our generalized formulation accounts for router placements in static networks, we must account for the dynamics of our nodes using tools from reachability. In particular, given the sets from Definition 3.2.2, a *necessary and sufficient* condition for our relative states z_{ij} to be in the desired set X, despite unknown client movement, is that $z_{ij} \in X^*$ for each edge (i, j) of the network of interest, G. In order to enforce this condition we must find an analytical description of the sets $\mathcal{E}$ and X^*. Unfortunately, this is in general not tractable.

A common solution is to approximate these sets by ellipsoids. Following [8] we find an ellipsoidal approximation $\tilde{\mathcal{E}}$ such that $\tilde{\mathcal{E}} \subset \mathcal{E}$. Thus it is sufficient to find a set satisfying $\tilde{\mathcal{E}} + W \subset X$ where the disturbance set W is given in Equation (3.2), and the desired set X is given by Equation (3.6). There are several proposed methods for computing $\tilde{\mathcal{E}}$. The reference [8] uses the method of [95, 99] for finding this ellipsoidal approximation set, although the approximation error incurred in this approach cannot be bounded. Instead, we use the method from [19] for approximating this ellipsoidal set, where the resulting approximation error can be bounded using the famous Fritz John result.

Lemma 3.3.2 (John Ellipsoid)**.** *For any convex set D in $\mathbb{R}^d$ that has the property of central symmetry about the origin, there exists an (maximal volume) ellipsoid $\mathcal{E} \subset D$ such that $\mathcal{E} \subset D \subset \sqrt{d}\mathcal{E}$. If D is compact then E is unique.*

We now find a maximal volume set $\tilde{\mathcal{E}}$:

Lemma 3.3.3 (An approximation to the reachability set with bounded error [19,72])**.** *The ellipsoidal approximation $\tilde{\mathcal{E}}$ to the set $\mathcal{E}$ can be found as the maximal volume ellipsoid such that $\tilde{\mathcal{E}} + W \subset X$. For our case where X is compact, Q from Equation* (3.2)

is positive definite diagonal and Ψ *from Equation* (3.6) *is positive definite diagonal we have:*

$$\tilde{\mathcal{E}} = \{x : \ x^T F x \leq 1\} \tag{3.11}$$
$$F^{-1} = (1 - p^{-1})\Psi^{-1} + (1 - p)Q^{-1}$$

Where p *is the only solution to*

$$\frac{1}{p+\alpha} + \frac{1}{p+\beta} = \frac{2}{p(p+1)} \tag{3.12}$$

that also satisfies $p \in [m^{1/2}, M^{1/2}] \cap (1, m)$ *where* $m = \min(\alpha, \beta)$, $M = \max(\alpha, \beta)$, $\alpha = 1$ *is the ratio of the diagonal elements of* Ψ^{-1} *(which is simply 1 for a ball of radius R) and* β *is the ratio of diagonal elements of* Q^{-1}.

The case of an exact computation of the reachability set $\mathcal{E}$

The set $\tilde{\mathcal{E}}$ has zero approximation error (is exact) for the special *homothetic* case.

Lemma 3.3.4 (Exact computation of $\mathcal{E}$ for the homothetic case [19])**.** *The case where the sets* X *and* W *are homothetic ellipsoids such that* $\Psi^{-1} = \nu Q^{-1}$ *for some scalar* $\nu > 0$*, then the reachability set* $\mathcal{E}$ *itself is an ellipsoid and* $\tilde{\mathcal{E}}$ *is exact.*

We make the important note that there are many cases of interest that fall into the homothetic category. Particularly, the case where client motion is velocity-bounded meaning that $W = \{w \in \mathbb{R}^d : \ \|w\| \leq v_w\}$ where v_w is the maximum client velocity at a given iteration and $X = \{x \in \mathbb{R}^d : \ \|x\| \leq R\}$ (as in the current case of interest) where R is a minimum communication radius.

3.3.2 Controller Development

In this section we derive an algorithm for finding optimal router-client assignments and placements for our k routers that provides a connected configuration for n mobile clients assuming only client to router and router to router communication. In the rest of this section, we use the words router and center synonymously.

Our algorithm is centralized and assumes access to all current client positions at each update time in order to allow maximum flexibility in the network topology due to client movement. We develop a novel formulation for this problem, namely the k-connected centers problem. We use tools from reachability analysis to incorporate unknown client motion, control input limitations for the routers, and communication range specifications to provide a solution that is guaranteed to be both *connected* and *feasible* over a computable time window, t_e, that we maximize.

k-Connected Centers

Unfortunately, k-centers does not solve the problem of providing a fully connected network over all clients and communication vehicles. We therefore extend the k-centers problem to also minimize the maximum router-router distance. Since it is prohibitive and unnecessary to constrain the routers to a fully connected system, we instead define a spanning tree over this set, with a corresponding bottleneck edge $b(C)$ where both the minimum spanning tree and bottleneck edge can be defined given a configuration C of router placements. We define the k-connected center problem as follows:

Definition 3.3.5 (k-Connected Centers Problem)**.** *The solution to the k-connected centers problem for a set P returns the set C^* where*

$$C^* = \arg\min_{C}\{\max\{r(P,C), b(C)\}\} \tag{3.13}$$

and C^* minimizes this cost over every set $C \subseteq \mathbb{R}^d$ with cardinality $|C| = k$. We define the k-connected center cost $r_b(P,C)$:

$$r_b(P,C) := \{\max\{r(P,C), b(C)\}\} \tag{3.14}$$

See Figure 3-3(b)

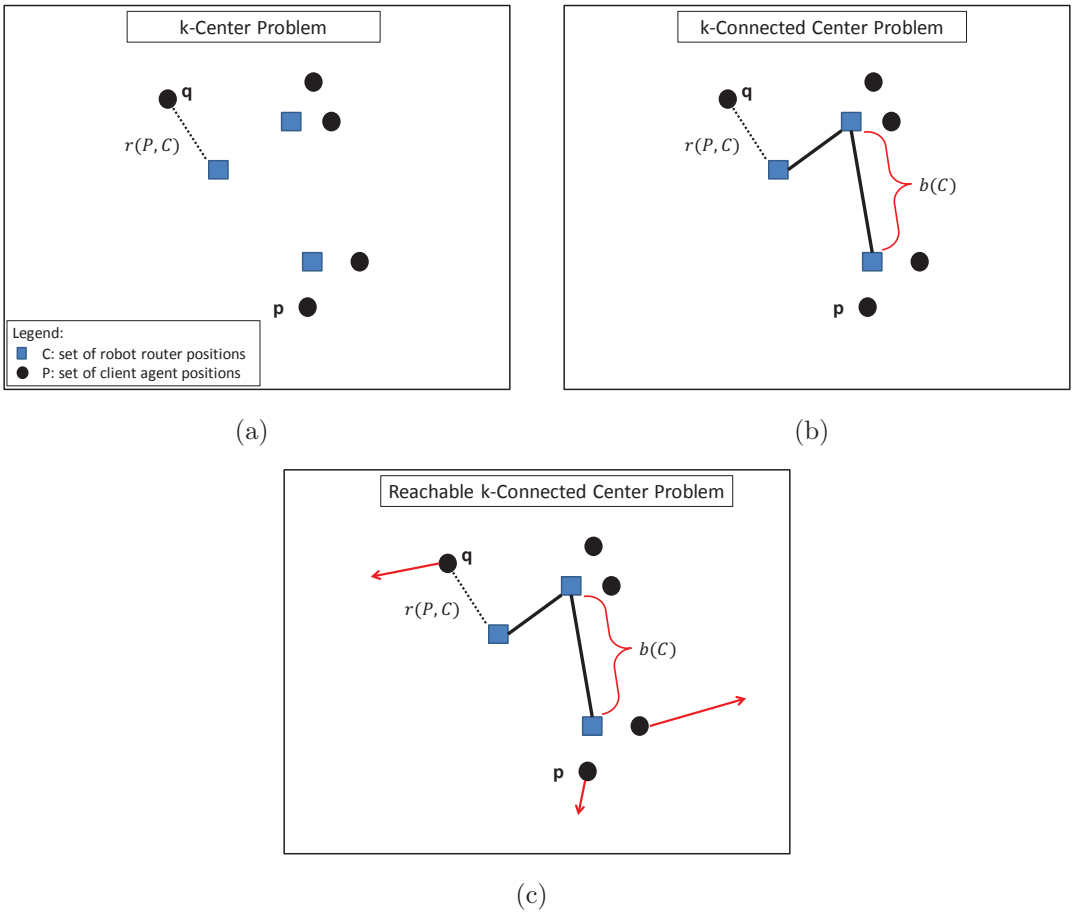


Figure 3-3: Schematic representation of the k-Center, k-Connected Center, and Reachable k-Connected Center Problem.

Exact Solution of the k-Connected Centers Problem

We can use the definition of the k-connected centers problem to devise an algorithm that returns the optimal solution for C^* via an exhaustive search over all centers and spanning tree combinations for the set of clients P. We first define the set of all possible client clusterings, $\mathcal{Q}(P,k)$, and connective edges between the k centers $E_C(k)$ and describe the mapping $\mathbb{C}$ that outputs a corresponding set of centers:

Definition 3.3.6 (The mapping $\mathbb{C}(D)$)**.** *Let $E(k)$ denote all the possible non-decreasing sequences $(\iota_1, \cdots, \iota_{k-1})$ where $\iota_1, \cdots, \iota_{k-1}$ are integers between 0 to k. Formally,*

$$E(k) := \{(\iota_1, \cdots, \iota_k) \in (0, 1, 2, \cdots, k)^k | \iota_1 \leq \iota_2 \leq \cdots \leq \iota_k\}. \tag{3.15}$$

and $\mathcal{Q}$ is the set of possible clusterings of points in P

$$\mathcal{Q}(P,k) := (P^{d+1})^k \tag{3.16}$$

In the sequel we use the shorthand $\mathcal{Q}(P,k) = \mathcal{Q}$. An element $Q \in \mathcal{Q}$ partition $(Q_1, \ldots, Q_k)$ where $Q_i \subset P$; we refer to an element Q_i as a clustering *of P. Let*

$$\mathcal{X}(P,k) := E_C(k) \times \mathcal{Q}(P,k) \tag{3.17}$$

denote all the pairs whose first item is a k-sequence from $E_C(k)$ and their second item is a sequence of $d+1$ points from P. The function $\mathbb{C} : X(P,k) \to (\mathbb{R}^d)^k$ maps a pair $D \in \mathcal{X}(P,k)$ to the sequence $\mathbb{C}(D) = (c_1, \cdots, c_k)$ of k centers as follows. Let

$$D = ((\iota_1, \cdots, \iota_k), (Q_1, \cdots, Q_k)) \in \mathcal{X}(P,k). \tag{3.18}$$

Let c_1 be the center of a ball that intersects all the $d+1$ points of Q_1. For every i, $2 \leq i \leq k$, we recursively define c_i as follows. In case $\iota_i = 0$, let c_i be the center of a ball that intersects all the $d+1$ points of Q_i. Otherwise, we define c_i to be the center of a ball that intersects the first d points of Q_i, and is also tangent to the ball that is centered at $c_{\iota_{i-1}}$.

The mapping from Definition 3.3.6 forms the basis of an algorithm to find the k-connected centers. In particular, the set $\mathcal{X}(P, k)$ can be used to define a *connectivity neighborhood* for an agent c_i

Definition 3.3.7 (Connectivity Neighborhood)**.** *For a given configuration of centers C_τ at iteration τ, a center $c_i \in C_\tau$ has an assignment Q_i of clients where $Q_i \subset P$ and possibly a connectivity edge with one other center c_j such that $c_j = T(c_i)$. The notation $T(c_i)$ is used to denote connectivity constraint between centers and is a* directed *constraint, meaning that $T(c_i) = c_j$ does not imply that $T(c_j) = c_i$. In fact we avoid this in order to prevent dependency cycles. The connectivity neighborhood $\mathcal{N}(c_i)$ of c_i is the set of vehicles that a communication vehicle is assigned to maintain connectivity to. Although connectivity constraints and neighborhoods are directed, communication is* undirected*, determined only by inter-vehicle distance.*

Reachability

Previously, we showed that the problem of finding optimal locations for placing communication vehicles such that they maintain close proximity to all clients and maintain connectivity amongst themselves can be viewed as a k-connected centers problem. The solution to the k-connected clients problem is not enough however, since uncertain client movement may render a connected configuration infeasible given the mobility of the clients and the control limit placed on the centers. We combine our k-connected centers approach (for defining connectivity neighbors) with reachability since this allows us to account for vehicle dynamics and maintain feasibility of a connected configuration over the next iteration. We also must ensure that any configuration at $C_{\tau+1}$ is physically attainable by centers located at C_τ. We would like to be able to compute an *expiration time* $T_E(C_t, P_t, \boldsymbol{Q}, \boldsymbol{\Omega}, R)$ over which we can guarantee that updated center positions at $C_{\tau+1}$ will maintain connectivity with all clients.

We aim to use control of the communication vehicles to maintain a connected configuration. We note that reachability tools are typically applied over a horizon of more than 1, however our choice to compute reachability over only a one step horizon is for two main reasons. Firstly, infinite-time reachability is infeasible for a fixed k

since clients are not constrained to bounds within the environment, and secondly, we prefer to allow for changes in the connectivity neighborhoods $\mathcal{N}(c_i)$ at each iteration τ in order to maximize flexibility of the solution and the motion freedom of the clients. Note that one iteration $(\tau_{l+1} - \tau_l)$ is equivalent to a several second window.

For a given communication graph $G(C \cup P, E)$, our objective is to maintain the resulting relative states $z_{ij} \in \tilde{\mathcal{E}}_{ij}$ for every edge in G. Because $\tilde{\mathcal{E}}_{ij}$ from Equation (3.11) is an approximation to $\mathcal{E}_{ij}$, where $\tilde{\mathcal{E}}_{ij} \subset \mathcal{E}_{ij}$ as derived in Lemma 3.3.3 is the maximum volume ellipsoid contained in $\mathcal{E}_{ij}$, the condition $z_{ij} \in \tilde{\mathcal{E}}_{ij}$ is a weaker sufficient condition for reachability of a connected graph (for the general case). In contrast, for the special homothetic case we get an exact solution so that $z_{ij} \in \tilde{\mathcal{E}}$ is a stronger condition for reachability of a connected graph. We may now formally define our reachable k-connected center problem:

Definition 3.3.8 (Reachable k-Connected Centers)**.** *For a given graph at time t, $G(C_\tau \cup P_t, E_t)$, where E_t are the set of edges between each center $c_i \in C_\tau$ and all nodes in its communication neighborhood $\mathcal{N}(c_i)$ from Definition 3.3.7, the solution to the reachable k-connected centers problem for a set P_t returns the set C^* where*

$$C^* = \arg\min_C r_b(P_t, C) + \gamma \tag{3.19}$$

$$s.t.\ c_i - c_{i\tau} \in U_\tau, \quad \forall c_i \in C, \forall c_{i\tau} \in C_\tau \tag{3.20}$$

$$(c_i - p_j)^T F_{ij} (c_i - p_j) \leq \gamma \quad \forall p_j \in \mathcal{N}(c_i),\ \forall c_i \in C \tag{3.21}$$

Where F_{ij} is the reachability matrix from (3.11) *computed for each link in the communication graph, and C^* minimizes this cost over every set constrained set C with cardinality $|C| = k$. See Figure 3-3(c).*

By Lemma 3.3.3 we have that the updated configuration of routers $C_{\tau+1} = C^*$ (where C^* is the solution to the reachable k-connected center problem for the set of clients P_t) is guaranteed to maintain connectivity of the graph $G(C_{\tau+1} \cup P_{t+1}, E_{\tau+1})$ for any P_{t+1} that evolves from P_t according to the equations in (3.2), and for all $t \leq \tau + 1$ where $\tau + 1 = t + t_e$. The expiration time t_e is the amount of time for

which connectivity is guaranteed and can be computed using the ellipsoidal equations for the sets $\tilde{\mathcal{E}}_{ij}$. In particular, for any communication graph $G(V, E)$ and for router dynamics given by (3.3) and client dynamics given by (3.2) we have that

$$t_e = \sqrt{r_\chi^*}/v \tag{3.22}$$

where r_χ^* is the solution to

$$r_\chi^* = \min_{(i,j)\in E} r_{\chi_{ij}} \tag{3.23}$$

and

$$\begin{aligned} r_{\chi_{ij}} = \min_x (x - z_{ij})^T Q_j (x - z_{ij}) \\ s.t.\ x^T F_{ij} x = 1 \end{aligned}$$

where $v = 1/\sqrt{\lambda_m}$ and λ_m is the smallest eigenvalue of Q_{j^*} where (i^*, j^*) is an edge that achieves the minimum in (3.23), and F_{ij} and the matrices describing the sets $\tilde{\mathcal{E}}_{ij}$ from (3.11). For special cases Equation (3.22) can be solved in closed form for t_e. For example, the case where the Q_j's and F_{ij}'s are diagonal, which in particular is true if all sets X,W, and U are balls (i.e. the velocity bounded cases). Using the expiration time t_e we can now make guarantees on the connectivity of our graph:

Lemma 3.3.9 (Guaranteed connectivity over a window of t_e seconds)**.** *For given communication neighborhoods $\mathcal{N}(c_i), \forall i \in \{1, \ldots, k\}$ where E is the set of edges between c_i and all $p_j \in \mathcal{N}(c_i)$ for every $c_i \in C_\tau$, a connected configuration is reachable if the solution from Equation* (3.19) *returns $\gamma^* \leq 1$. Let t^* denote the number of seconds for which the communication graph $G(C_{\tau+1} \cup P_{t+1}, E_{\tau+1})$ is guaranteed to remain in a connected configuration (i.e. such that $z_{ij} \in X$ for every edge $(i, j) \in E$) for all P_{t+1} evolving from P_t according to Equation* (3.2) *and for all $t + 1 \leq t^*$. By Lemma 3.3.2 and the construction of $\tilde{\mathcal{E}}_{ij}$, we have that $t_e \leq t^* \leq \sqrt{2} t_e$ for t_e computed*

via Equation (3.22).

For the special homothetic case described in 3.3.4 we have that $\tilde{\mathcal{E}}$ is an exact reachability set and the bound on t_e is exact:

Lemma 3.3.10 (The special case of an exact time bound)**.** *By Lemma 3.3.4 we have that $z_{ij} \in \tilde{\mathcal{E}}_{ij}$ maintains connectivity of the communication graph $G(C_{\tau+1} \cup P_{t+1}, E_{\tau+1})$ for any $t + \delta$ where $\delta \leq t_e$ where P_{t+1} evolves from P_t via the Equations* (3.2)*. Here $t_e = t^*$ where t^* is an exact lower bound for the connectivity time window.*

Using the reachable k-connected center formulation we can now derive a convex program that can be optimized over all connectivity graphs $G(C_\tau \cup P_t, E_\tau)$ at each iteration $\tau = t_1 + t_{e1}, t_2 + t_{e2}, t_3 + t_{e3}, \ldots$ to find a configuration of routers $C_{\tau+1}$ that are guaranteed to maintain connectivity of the graph for the next t_e seconds, where t_e is maximized. The following convex program can be optimized to find C^* and t_e for given communication neighbors provided by $\mathcal{N}(c_i)$ for every center.

$$\min_{C,r} \; r + \gamma \tag{3.24}$$

$$s.t. \|c_i - p_j\| \leq r, \; \forall p_j \in \mathcal{N}(c_i), \; \forall c_i \in C \tag{3.25}$$

$$(c_i - c_{it}) \in U_i, \; i \in \{1, \ldots, k\}, c_{it} \in C_\tau$$

$$(c_i - p_j) \in \tilde{\mathcal{E}}_{ij}, \; \forall c_i \in C, \forall p_j \in \mathcal{N}(c_i), \tilde{\mathcal{E}}_{ij} \text{ from Eq (3.11)} \tag{3.26}$$

3.4 Algorithm for Finding k-Connected Centers with Reachability Constraints

We combine the results from the previous two subsections on k-connected centers and reachability analysis to provide an algorithm for returning communication vehicle placements that minimize the reachable k-connected centers cost from (3.19) over all communication graphs.

Algorithm 1 Exact Algorithm for k-connected centers with Reachability

input : Set of clients P, number of centers k, current center locations C_t

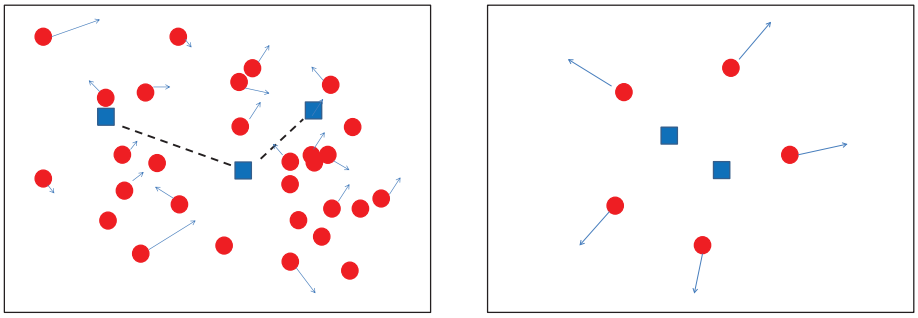
output: Optimal centers C_{t+1}, cost r^* where optimality is wrt Definition (3.3.8), expiration time t_e

```
for Every partition 𝒩 ∈ 𝒬(P, k) from (3.16) do
    for Every spanning tree 𝒯 over k nodes do
        (C, r) ← Equation (3.24)    // Evaluate convex program for reachable
        k-connected centers
        if r < r* then
            r* = r; C* = C; γ* = γ; 𝒩* = 𝒩
C_{t+1} = C*
t_e ← Equation (3.22)  // Evaluate expiration time using the neighborhoods
𝒩* and router positions C*
return : r*, C_{t+1}, t_e
```

Theorem 3.4.1. *Let P be a set of points in $\mathbb{R}^d$ and $k \geq 1$ an integer. Algorithm 1 provides an exact solution to the reachable k-connected centers problem from Definition 3.3.8, where the resulting set of centers C^* is the set of centers of cardinality $|C^*| = k$ that minimizes the cost $r_b(P, C)$ over every set C of centers that satisfy the constraints . Moreover, the configuration C^* guarantees connectivity over the entire heterogeneous network for a minimum of t_e timesteps where t_e is the expiration time for C^* and is defined in* (3.22)*. The algorithm runs in $n^{O(k)}$ time.*

Proof. The proof for the optimality of the solution C^* under the constraints follows from the exhaustive search enumeration for the exact algorithm we presented for k-connected center. It follows from Lemma 3.3.9 that the configuration C^* guarantees connectivity over t_e timesteps. The Q loop in Algorithm 1 runs computations over subsets of P of size $\binom{n}{d+1}$ for each of the k centers, and the inner loop performs computations over all spanning trees for a graph with k nodes. Since convex quadratically constrained quadratic programs can be solved in polynomial time and we assume $n \geq k$ we have that the dominating complexity is $n^{O(k)}$. □

Like its cousin the classic k-center problem, our derived formulation, the connected k-center problem is exponential in its complexity. Therefore, for large problem sizes we cannot expect an exact solution that is computable in real time. For this reason,



(a) (b)

Figure 3-4: Schematic of both scenarios of our case study where sensors (red circles) either move randomly through the environment or move radially outward and test the communication limits provided by the centers (blue squares).

in the next chapter of the thesis we focus on developing approximate solutions that can be computed efficiently and whose approximation error can be bounded.

3.5 Numerical Studies

In this section we provide empirical results that study i) the effect of sensor velocity and ratio of centers k to sensors n on the expiration time t_e (see Figure 3-5) and ii) computation vs. expiration time and how the expiration time can be used as a guide for switching between exact and approximate methods (see Figure 3-6). The empirical results of this section use computation of the set $\tilde{\mathcal{E}}$ from [8].

We examine two sensor behaviors, where 1) sensors are distributed uniformly over a fixed area but vary their maximum velocities (Figure 3-4(a)) and 2) sensors begin at the center of the environment and move outwards radially at a speed of $1m/s$ permitting worst-case analysis (Figure 3-4(b)). For the first case we investigate the effect of increasing sensor velocities on the *expiration time*, or minimum bound on the time that the centers generated by Algorithm 1 are guaranteed to maintain communication with the sensors. It must hold that the maximum allowable velocity of the centers is at least that of the sensors in order to maintain connectivity. We also vary the ratio of sensors to communication vehicles in order to demonstrate how a growing density of communication vehicles increases the attainable expiration times.

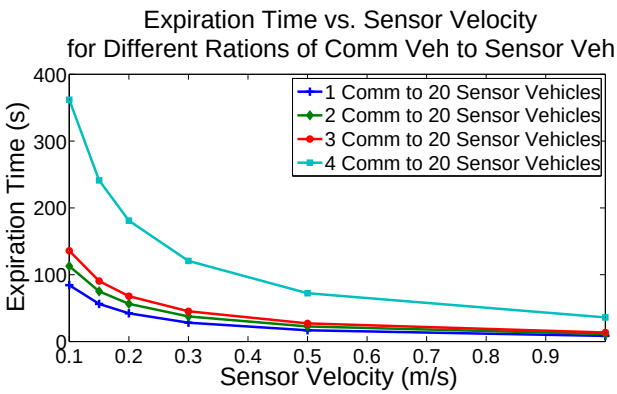


Figure 3-5: Effect of sensor velocity on expiration time studied for a fixed $R = 60$m and communication (router) vehicle speed $v_C = 1.5m/s$. Different curves correspond to different ratios of comm vehicles to sensors k/n and the slope of the curves become shallower as the number of router vehicles k increases per unit area.

The second case assumes a fixed number of 2 centers that must provide connectivity for 5 sensors that are moving radially outward from the environment center. Figure 3-6 demonstrates the motivation for switching from an exact optimization method, to an approximate method that uses a coreset. We perform this switch when the computation time of the exact algorithm from Algorithm 1 reaches the expiration time and thus the center locations must be updated more quickly in order to maintain connectivity. As the sensor vehicles move farther outwards towards the communication limits of the centers we can switch to a $(3 + 2\alpha)$-approximation algorithm. The derivation of algorithms for solving the approximate version of this problem can be found in the next chapter of the thesis.

3.6 Conclusion

In this chapter of the thesis we presented a novel optimization formulation for computing router-client assignments and router placements to maintain connectivity of an ad-hoc network over a computable time window such that the connectivity is robust to unknown but bounded client motion. For this purpose, we formulate a *reachable k-connected center* problem that is related to the classic k-center and reachability for dynamic system problems. We present an algorithm to solve for router placements such that for a given number of k routers, the resulting configuration is optimal in

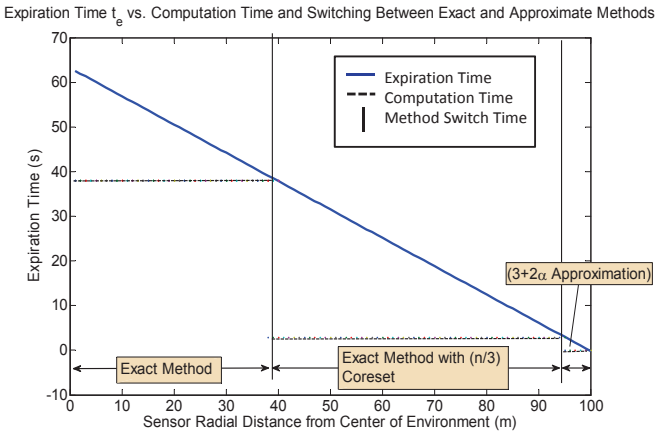


Figure 3-6: Plot of t_e vs. radial sensor distance from center of environment. Computation time (horizontal dashed lines) is compared against expiration time (solid blue line) and switch times (solid vertical lines) between exact and approximate methods is demonstrated.

terms of the best assignment of routers to clients. While this algorithm can be executed over the entire input set for a small number of clients, its exponential running time makes an exact solution intractable for large input sets. Therefore we dedicate the next chapter of this thesis to solving the router placement problem efficiently even for large sets of clients.

Chapter 4

Efficient Computation of Router Placements

4.1 Introduction

In the previous chapter of the thesis we presented a problem formulation, the k-connected center with reachability problem, for assigning a cost to a particular configuration of routers (router placement and client assignments) and an algorithm for finding an optimal router configuration that minimizes this cost and provides a connected network over a computable time window. However, like the classic k-center problem from which our formulation is derived, computing an optimal solution for router placements is computationally expensive. In particular, it is exponential in the number of clients. For large numbers of sensors n, computing the exact solution for the k-connected center problem can consume a prohibitive amount of time (i.e., the case $\tau_{l+1} - \tau_l > t_e$). It would be desirable to instead compute the exact solution over a carefully chosen subset of sensors such that a solution can be computed more efficiently and in a way that the induced approximation error can be bounded. The expiration time computed in the previous chapter, compared against required computation time, provides a lower bound estimate of whether there is enough time to compute the exact solution or whether the approximate solution should be computed instead. Intuitively, for large systems or for cases where the positions of the client

routers must be updated quickly due to mobile clients with high velocities operating near the communication limits of the routers, the faster approximate version of the algorithm is used.[1]

In this chapter we focus on finding approximate solutions to the router placement problem that can be computed efficiently and where the resulting approximation error can be bounded. We may then use the expiration time from the previous chapter, compared against the required computation time for solving for router placements, in order to provide a lower bound estimate of whether there is enough time to compute the exact solution or whether the approximate version of the algorithm should be used. We provide large-scale numerical simulations (thousands of clients), as well as small-scale hardware experiments (ten clients), to show that our approximation algorithms allow for updating router positions in real time as clients move through the environment, and for both large (thousands) and small (order of ten) sets of clients.

4.1.1 Assumptions

In this chapter of the thesis we make the following assumptions as in the previous chapter:

(i) Signal strength can be mapped directly to the Euclidean distance between two communicating nodes. We refer to this as the *Euclidean disk assumption*.

(ii) Current positions of clients and routers are known at the time that a new configuration is computed in a centralized manner, although future client positions (more generally, their trajectories) are unknown.

(iii) Client communication demands are equal across the network.

(iv) A communication network $G = (V, E)$ is *connected* if no edge exceeds a given radius R such that $\max_{(i,j)\in E} dist(v_i, v_j) < R$.

[1] This work has also been presented in [31, 45]

Chapter 5 of the thesis relaxes these assumptions by replacing the Euclidean disk assumption with real-time channel feedback for each link in the network.

4.1.2 Results Snapshot

Inspired by recent results in computation geometry, information theory and classic results from reachability theory, our main technique for approximation is to maintain a representative set S of $O(k)$ points and compute a solution of router placements over this representative set. We prove that running any approximation algorithm for the k-connected centers or the reachable k-connected centers on the small set S, would yield a $(1+\varepsilon)$-approximation to the exact result of running the same algorithm on the original set of n points. In recent years, such small sets S (known as coresets) were constructed for other optimization problems such as the k-center problem [7]. We now give an overview of the results presented in this chapter of the thesis.

(i) *Algorithm for computing a static coreset*: We provide an algorithm for computing a representative set $S \subseteq \mathbb{R}^d$ for which running Algorithm 1 returns a $(1+\varepsilon)$-approximation to the k-connected centers problem with reachability

(ii) *A constant factor algorithm for router placement*: We provide a generalization of the Gonzalez 2-approximation algorithm for k-center that can be applied to our k-connected center problem and returns a solution of router placements that is a $(3+2\alpha)$-approximation and takes linear time $O(nk)$ to compute. More generally, we prove that any γ-approximation for the classic k-center problem is also an $(3+2\gamma)$-approximation for the k-connected center problem.

The results stated above find a static representative set of client agents at each time for which a solution to the k-connected centers with reachability problem is computed. While the development of sparse representative sets with the $(1+\varepsilon)$ approximation bound property did not previously exist for this class of problems, and thus constitute a contribution on its own, this is still not an ideal solution for our dynamic problem. This is because there may be many such representative sets

that satisfy the same error bound but will correspond to different router placements. Thus we develop a new data structure suited for dynamic applications, a *kinematic coreset*, and algorithms for computing and quickly updating such a data structure.

(iii) *Algorithm for computing a kinematic coreset*: an algorithm to compute a kinematic coreset over the clients, that can be used as input for computing server positions to form a connected communication network for the set of moving client vehicles. This coreset differs from the *static* coreset in that the kinematic coreset can be updated quickly, is reactive to client movement, and also provides consistency across iterations such that for marginal client movement the same coreset can be maintained.

(iv) *Experimental results*: We test the computational efficiency and approximation error properties of our kinematic coresets on large problem sizes of up to $n = 2000$ in simulation as well as on a small scale hardware implementation using two Kuka Youbot robots (servers) that must react online to five clients moving over *a priori* unknown trajectories.

4.1.3 Notation

Table 4.1: Common Notation for Chapter 4

k centers	*is synonymous with*	k routers.
$G(V, E)$	$\triangleq$	Communication graph with nodes V and edges E.
ε	$\triangleq$	A given error tolerance where $\varepsilon > 0$.
p_j	$\triangleq$	Client j position in $\mathbb{R}^d$.
c_i	$\triangleq$	Robot router i position in $\mathbb{R}^d$.
S	$\triangleq$	A coreset for a set of clients P.
$r(P, C)$	$\triangleq$	k-center cost for clients P and robot routers C.
$r_b(P, C)$	$\triangleq$	k-connected center cost for clients P and robot routers C.
$b(C)$	$\triangleq$	Bottleneck edge of minimum spanning tree over $G(C, E)$.
t_e	$\triangleq$	Expiration time.

4.2 Problem Statement

In this section of the thesis we are interested in constructing a *coreset*, or small representative set for the clients such that a solution to the reachable k-connected centers problem can be computed over this set and the resulting approximation error can be bounded. We would like to find a set S with the property that a solution to the k-connected center with reachability problem (Definition 3.3.5) computed over S will provide an $O(1+\varepsilon)$ approximate solution compared to the optimal solution computed over the entire input set P.

Definition 4.2.1 (An α-approximation)**.** *For a positive scalar $\alpha \geq 1$, we say that a cost $f(P,C)$ is an α-approximation to the original cost $h(P,C)$ if the optimal solution $f^*(P)$ has the property that $f^*(P) \leq \alpha h^*(P)$ where $h^*(P)$ is the optimal solution of $h(P,C)$.*

The mathematical definition of a coreset can be given as follows:

Definition 4.2.2 (A (k,ε)-coreset)**.** *Let C^* denote the set that minimizes $r(P,C)$ over every set C of k centers. For a given $\varepsilon > 0$, a* (k,ε)-coreset *for P is a subset S such that for every point $p \in P$ we have*

$$\text{Dist}(p,S) \leq \varepsilon r(P,C^*). \tag{4.1}$$

In particular, for every set C of k centers we have

$$r(S,C) \leq r(P,C) \leq (1+\varepsilon) r(S,C). \tag{4.2}$$

Such a coreset approximates the cost of P for every given query set of k centers, up to an additive error of εr^* where $r^* = r(P,C^*)$ is the optimal k-center of P. Since $\varepsilon r^* < \varepsilon r(P,C)$ for every set C of k-centers, such a coreset also yields a multiplicative factor of $(1 \pm \varepsilon)$ for the cost $r(P,C)$. We wish to find a coreset with the same property for our k-connected centers cost $r_b(P,C)$ with reachability from Definition 3.3.8.

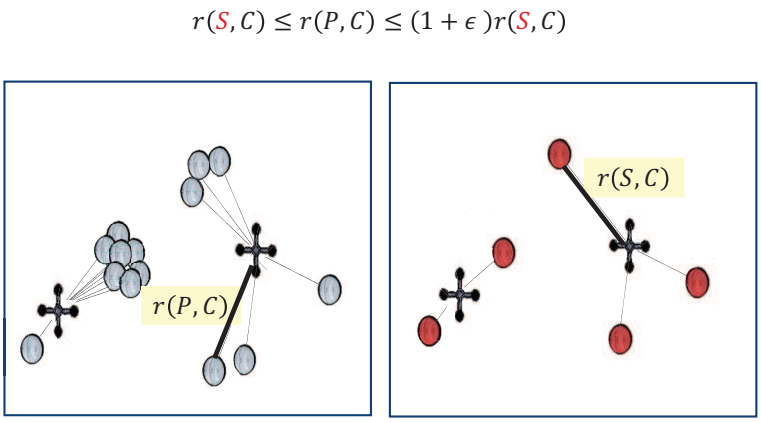


Figure 4-1: Schematic interpretation of the property of a (k, ε)-coreset.

Secondly, we would like to maintain the error bounding properties of this representative set as clients move. We refer to a data structure having these properties as a *kinematic coreset*. A *kinematic* (k, ε)*-coreset* is a data structure that dynamically updates the coreset whenever a client updates its new position. More precisely, the data structure consists of a (k, ε)-coreset S for the set of clients P, and an update method MOVE(p, p') that gets a client's position $p \in P$ and replaces it by $p' \in \mathbb{R}^d$. That is, both P and S are updated in each call to the MOVE method. We wish to maintain such a set S of size as small as possible, and also to minimize the execution time of a call to MOVE. Our objective is to determine the kinematic coreset, or sparse representative set, of clients and control the servers to track this coreset.

Finally, we wish to find a constant factor solution that can compute a configuration of routers, $\tilde{C}$, in linear time $O(nk)$ such that $r_b(P, \tilde{C}) \le (3 + 2\alpha) r_b^*$ where $\alpha \ge 1$ is a given approximation error for the original k-center problem and $r_b^* = r(P, C^*)$.

4.3 Tracking a Representative Set of Clients

To produce gains in efficiency we use a data structure called a coreset for the k-center problem. Our first result is to prove that the same coreset has the desired

properties for the k-connected center problem. Hence, running our exact algorithm (Algorithm 1) on the coreset S instead of P, would yield a $(1+\varepsilon)$-approximation to the k-centers of P. The corresponding running time would be reduced then from $n^{O(k)}$ to $|S|^{O(k)}$. Clearly, $S = P$ is a coreset for P. However, if we can compute a coreset of size $|S| \ll n$ the running time on the coreset would be significantly smaller.

In this section we prove that such a coreset $\hat{S}$ of size $O(\frac{k \log n}{\varepsilon^d})$ exists for every given set P of robots. Moreover, we provide an algorithm that computes $\hat{S}$ in $O(nk)$ time; see Algorithm 2.

Theorem 4.3.1. *Let $S \subseteq P$ be a set of points that is returned by a call to Algorithm 2 with $S \subseteq \mathbb{R}^d$, $k \geq 1$ and $\varepsilon > 0$. Then, with arbitrarily high probability of at least $1 - 1/n$, S is a (k, ε)-coreset for P. That is, for every set C of k centers in $\mathbb{R}^d$ we have*

$$r_b(S, C) \leq r_b(P, C) \leq (1+\varepsilon) r_b(S, C). \tag{4.3}$$

The running time of the algorithm is $O(nk)$.

Proof. We show that a (k, ε)-coreset S for the k-center problem is also a (k, ε)-coreset for the reachable k-connected center problem. Indeed, if Algorithm 2 returns a (k, ε)-coreset S for the k-center problem, then we have that the same set S is a (k, ε)-coreset for the k-connected center with reachability problem. This follows from (4.3) since the bottleneck edge is only a function of C and not the input set P and $r_b(P, C) = \max\{r(P, C), b(C)\}$. Since configurations $\tilde{C}$ that satisfy the reachability condition from (3.21) are a subset of C, and Equation (4.3) holds for all C then this property also holds for any constrained sets $\tilde{C}$. From the proof in [33] we have that Algorithm 2 returns a coreset S for the k-center problem with a known probability of $1/2$ in time $O(nk)$. Thus by running Algorithm 2 γ times the probability can be boosted arbitrarily high to $1 - (1/2)^\gamma$ where letting $\gamma = log(n)$ returns a (k, ε)-coreset with probability of at least $1 - 1/n$ and the total running time for finding this coreset is still $O(nk)$. □

A (k, ε)-coreset S for the k-center problem is also a (k, ε)-coreset for the reachable k-connected center problem. This is formalized in the following corollary:

Corollary 4.3.2. *Let P be a set of n points and let $\varepsilon > 0$ be a constant. Then a $(1+\varepsilon)$-approximation for the reachable k-connected center of P can be computed in $O(nk) + (\log n)^{O(k)}$ time. In particular, let P be a set of n robot locations in $\mathbb{R}^d$, $k \geq 1$ be an integer, and $\varepsilon > 0$ be a constant. Let $\hat{S}$ be the output of Algorithm 2, and let $\tilde{C}$ be the reachable k-connected center of $\hat{S}$. Then $\tilde{C}$ is a $(1+\varepsilon)$-approximation to the reachable k-connected center of P.*

Overview of Algorithm 2. We pick a small random sample T_1 from P. Such a random sample has the property that it "hits" large clusters Q_1 of robots, but probably misses outliers. Hence, in Line 4 we remove only the half closest robots to T_1, which are approximated well, and keep the remaining robots. We continue recursively until no robots are left. This yields $O(\log n)$ iterations that corresponds to $O(\log n)$ sample sets. Since $T \subseteq P$, where T is the union of sample sets, we have that $r(\hat{S}, C^*) \leq r(P, C^*)$ for the k-center of P. In Lines 10–12 we turn this $O(1)$-factor approximation into $O(\varepsilon)$-approximation by constructing the grid G_p around every robot p in $\hat{S}$. The distance between two points that are in the same cell of the grid is at most $\varepsilon r_b(P, C^*)$

4.3.1 Empirical Results for Static Approximations

Figure 4-2 shows the improvement in the induced error, ε, of using a coreset as a representative set for P vs. the size of the representative set. We compute a 2-approximation to the k-center cost on different sized input sets,$|P| = 30$ and $|P| = 300$. This plot shows the ratio of this k-center cost computed over a representative set to the k-center cost computed over the entire input set P. We contrast the performance of using a uniform random sample (dashed line) to that of using a coreset (solid line) and show that the coreset provides better performance with approximation error $\varepsilon \leq 0.34$ for only $n/2$ sample points (when $n = 30$) and $\varepsilon \leq 0.05$ for less than $n/3$ sample points (when $n = 300$). Since the computation time for computing the

exact k-center cost is exponential, even for small input sets the computational savings is significant using coresets.

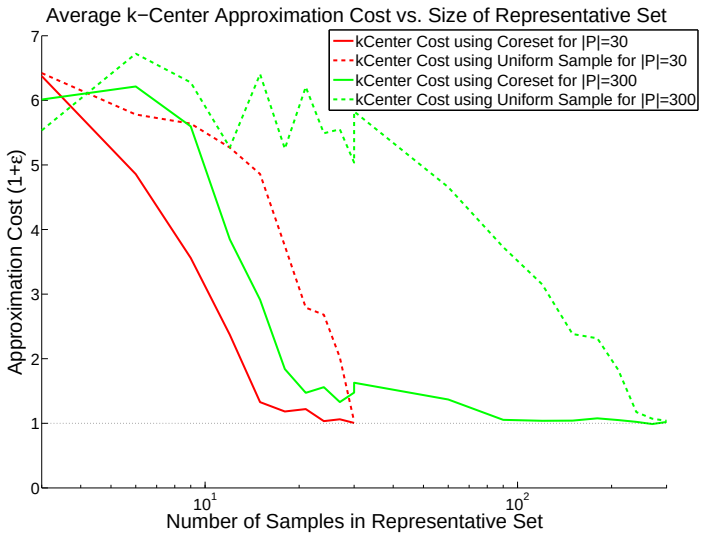


Figure 4-2: This plot shows aggregate results over 1000 runs for the error induced by using a representative set of size $|S|$ for the input set P vs. increasing representative set size. The plot shows that a *coreset* (solid line) provides better performance with approximation error $\varepsilon \leq 0.34$ (vs $\varepsilon = 3.8$ for a uniform random sample (dashed line)) for only $|P|/2$ sample points when $|P| = 30$, and $\varepsilon \leq 0.05$ (vs. $\varepsilon = 2.7$ for a uniform random sample) with less than $|P|/3$ sample points when $|P| = 300$.

4.3.2 An Improved Dynamically Updated Approximate Solution

The main tool we propose to handle problems where we wish to maintain a representative set, or coreset, of dynamic client vehicles is a *kinematic coreset*. This tool yields a sparse representative set of the n clients that provably approximates their maximum distance to any possible positioning of the k servers at any given time. Since our coresets are small and can be updated quickly, we are able to apply exact (optimal) solutions that would otherwise be intractable. This yields dynamic positioning of the servers that provably approximates the optimal solutions on the full set of clients. Since the running times are exponential in k, our coresets improve the performance even for a small number $n < 10$ of clients.

We derive a kinematic coreset with the following properties: it 1) can be updated

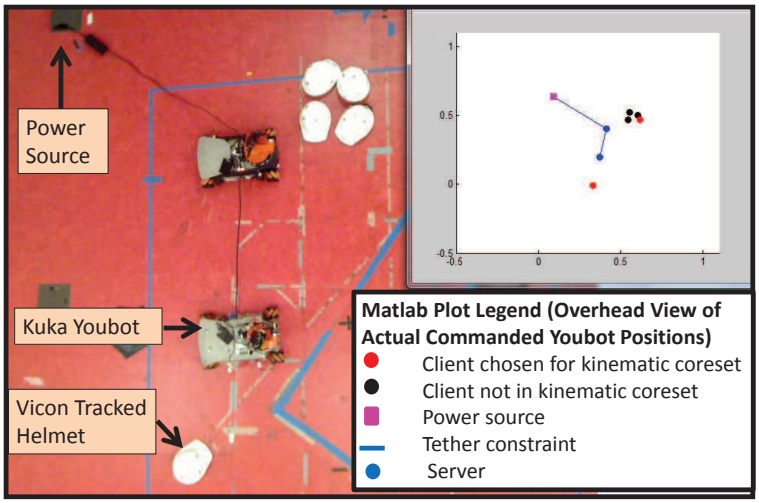


Figure 4-3: Overhead view of our hardware setup of two Kuka Youbot robots and white helmets with vicon tracking markers that were worn by five adults, or "clients", moving around the motion capture room at a walking pace as shown in figure 4-5. This figure shows the constraint that the first Youbot was tethered to a power source at the top left corner of the environment and the second Youbot was tethered to the first Youbot for power. The Matlab plot in the top right corner demonstrates the two kinematic coreset points (red) and the optimal configuration of the Youbots (blue) computed for the client positions under the given tethering constraints.

quickly and adapt to client motion, 2) provides consistency such that the same coreset can be maintained for marginal client motion and 3) can provide approximation error bounds for the k-center and k-connected center problems as well as constrained versions of these problems. Our system contains the first implementation of kinematic coresets, and includes several improvements to the state of the art, both in terms of theoretical guarantees and practical usage.

4.3.3 Algorithm for Maintaining a Kinematic Coreset

In Algorithms 2–5 we define the main procedure MOVE for updating the coreset, together with its sub-routines. We provide a sample optimization problem that we run on the coreset in our experiments, for computing a set of servers that is close to the clients, with additional restriction on maximum distance between servers. The procedure INIT(P) (Algorithm 1) is called once with the initial position of clients

set P. It runs the static version of our coreset construction from [33]. The data structure maintains the coreset in each call to $\textsc{Move}(p, p_a, i)$, and correctness follows from Theorem 4.3.3.

To minimize the changes to the coreset, our data structure maintains a partition of the clients in $O(\log n)$ resolution levels. The first level is large and represents the "main stream" or large dense clusters of the clients that are not sensitive to a small fraction of clients that may change their position. The last level is very small and consists of few "outliers" or isolated clients that change their location frequently. Each of the $O(\log n)$ levels has its own coreset of $m = O(k \log(n)/\varepsilon^d)$ clients. This yields a coreset of size $O(k \log^2(n)/\varepsilon^d)$. When a client updates its location the method $\textsc{Move}(p, p_a, i)$ is called. The parameter p denotes the last recorded position, and p_a denotes the new position. The last recorded position can be saved on the client or server side. The data structure then computes which of the following actions should be taken:

No update. Our data structure maintains the distances of each client to its closest coreset point at its level. These distances are stored in a binary heap for fast updates. A binary heap has the property that a value in a heap's node is always larger than its childs node. Therefore, the root of the heap contains the largest distance from a client to the coreset of the heap in that level. When a client updates its new location, it also sends a pointer to its node in the heap. If the heap is still valid after the change (the new distance is still larger then the node's child and smaller than its parent) no other action is taken. This is the fastest update type and takes constant $O(1)$ time.

Heap update. When the updated distance of a client to the closest coreset point in its level does not preserve the structure of the heap (the new distance is smaller than the node's childs or larger than its parent), we need to "heapify" the node down or up the heap to preserve its structure. The update time depends on the number ℓ of such switches with other nodes. Since the height of the heap is $O(\log n)$ these changes take time $\ell \leq O(\log n)$.

Level update. After a series of heap updates, a client may be the farthest from

its level coreset, and reaches the root of its level's heap. In this case, if a new client is added to the level with a smaller distance to the coreset, we remove the client in the root to a different resolution level, or even ℓ levels. The update time for such a change is $O(\ell)$ where $\ell \leq \log_2 n$ is the difference between the current level and the new level of the client.

Coreset update. Every level maintains its coreset, which is a uniform random sampling of size k from its clients positions during different times ("snapshots"). That is, when a client position is chosen for the coreset, the client itself may continue to move, but its "recorded" coreset point is static until it is removed from the coreset. We thus call the coreset points "virtual clients".

When too many clients (constant factor) have entered or left the level's heap using heap updates, the coreset should also be updated. Updating a point in the coreset may affect all the points that it serves in the level, and also next levels. However, since the coreset is a random sample, update should occur very rarely in the higher levels (which contain large clusters) and may occur frequently in the lower levels (the small sets of outliers). Based on this observation, we prove that the overall expected running time of such an update is at most $O(\log n)$.

Algorithm 4 handles the case where p is inserted or deleted from the ith level. That is, p was one of the $|Q_i|$ closest points to S_i but not after the call to MOVE, or vice versa. Intuitively, p has left its cluster and has moved from the "main stream" toward a different level of resolution. In this case, we insert (respectively, remove) p to its new (respectively, old) heap and continue recursively to update p in the next level.

In case the size of the heap of Q_i is not cP_i for some $c \in (1/4, 3/4)$, then we also need to balance the heaps by moving the root r of one of the heaps to the other one and recursively update this change in the next pair (Q_{i+1}, S_{i+1}).

Finally, we handle the case where there is no $c \in (1/4, 3/4)$ such that $|S_i| \in c(k + \log n)$. That is, p_i was removed or inserted to S_i. In this case, we recompute all the data structures that correspond to the pairs $(Q_i, S_i), \cdots, (Q_{|D|}, S_{|D|})$. As we prove in our main theorem, this event is rare (happens with probability at most $1/n$).

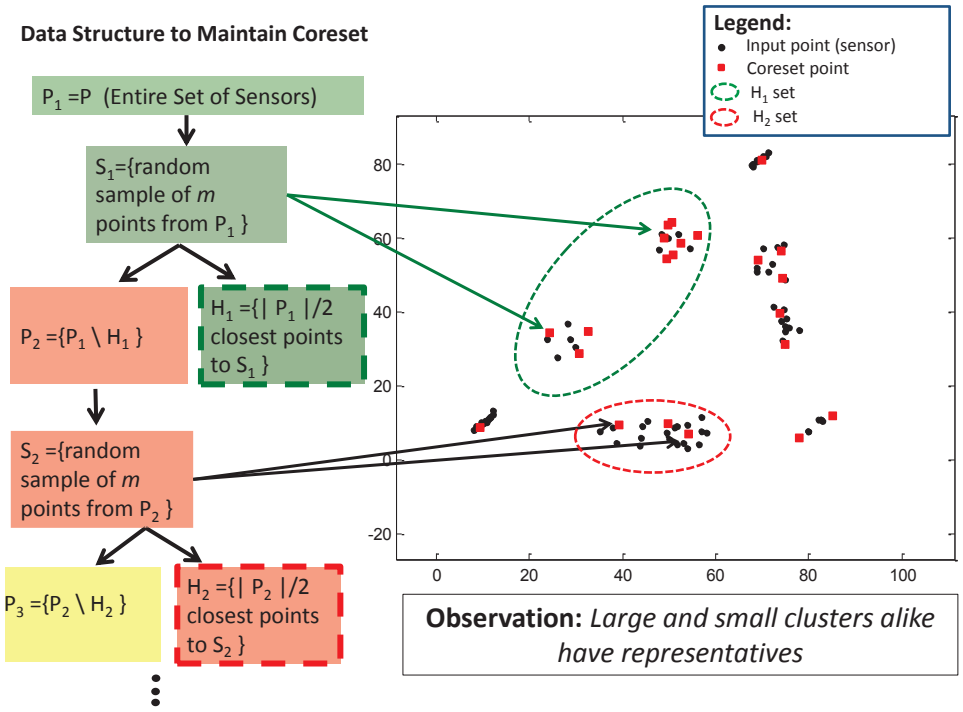


Figure 4-4: Typical test scenario where clients move randomly between clusters forming large and small clusters. The red squares are sampled input points from a (k, ε)-coreset demonstrating that most clusters (both large and small) are fairly sampled in contrast to uniform random sampling where several small clusters are often missed thus adversely affecting approximation quality. On the left, key properties of the construction of a (k, ε)-coreset are explained.

4.3.4 Analysis

In this section we prove that the algorithms presented in the previous section provide a kinematic coreset S that 1) is updated in time polynomial in $\frac{k \log(n)}{\varepsilon}$ for an input set of n sensors, and 2) captures information about clients most critically influencing the cost by reactively updating itself to maintain an upper bound on the approximation error as compared to the optimal cost by a factor of at most $(1 + \varepsilon)$.

First we show that a coreset computed by the static approach in Algorithm INIT provides the desired $(1 + \varepsilon)$, $\varepsilon \in (0, 1/2)$, error bound on the optimal cost for the k-centers. A corollary to this provides that any algorithm obeying three key properties of the INIT algorithm also produces a (k, ε)-coreset for our communication costs.

Proposition 4.3.8 further generalizes this claim so that any (k, ε)-coreset for P is also a coreset for any perturbed set P_A as long as the magnitude of the perturbation is bounded below some constant factor. Finally we prove our main result: that our Algorithm MOVE for updating a *kinematic* coreset in time polynomial in $\frac{k \log(n)}{\varepsilon}$ indeed satisfies Corollary 4.3.4 and thus provides the error guarantees for an arbitrarily moving set of sensors.

Theorem 4.3.3. *Let P be a set of n points, $k \geq 1$ be an integer and $\varepsilon > 0$ be a constant. Let S denote the output of the algorithm* INIT(P, m) *for an appropriate $m = O(\frac{k \log(n)}{\varepsilon^d})$. Then, with arbitrarily high probability of at least $1 - 1/n$, S is a (k, ε)-coreset for P of size $|S|$ that is polynomial in $\frac{k \log(n)}{\varepsilon}$.*

Proof. The algorithm INIT is a small modification to the Static BiCriteria algorithm from [33] and thus the proof from that paper holds with minor modification to account for a different constant factor $c \in (\frac{1}{4}, \frac{3}{4})$ of points taken from P_i at every iteration. □

A result of this theorem is that any algorithm that maintains the properties of the coresets S_i from Algorithm INIT as an invariant also produces a (k, ε)-coreset for P. In particular we state the following Corollary:

Corollary 4.3.4. *Let P be a set of n clients, $j \geq 1$ be an integer, and $(H_1, \cdots, H_j)$ be a partition of P. Let $S \subseteq P$ and $(S_1, \cdots, S_j)$ be a partition of S. Let $m = O(\frac{k \log(n)}{\varepsilon^d})$ be defined as in the previous theorem. Suppose that the following properties 4.3.5-4.3.7 hold for every $i = 1, \cdots, j-1$:*

Property 4.3.5. *S_i is a random sample of size $|S_i| \geq m$ from P_i*

Property 4.3.6. *H_i is the set of $c|P_i|$ points $p \in P_i$ with the smallest* $\text{Dist}(p, S_i)$*, for some $c \in (1/4, 3/4)$.*

Property 4.3.7. *$P_{i+1} = P_i \setminus H_i$*

Then S is a (k, ε)-coreset for P

Since it is inefficient and costly for the coreset to change as client vehicles move over small distances that do not have a significant effect on cost, we must be able to

show that a coreset S for an input set P is also a coreset for a perturbed set P_A if the perturbation is small in magnitude. The following proposition defines tolerable perturbations such that this property holds and is a key component of our kinematic update algorithm MOVE.

Proposition 4.3.8 (Coresets for Perturbed Sets)**.** *A (k, ε)-coreset, S, for an input set P, is also a (k, ε)-coreset for any other set P_A if for every $q \in P_A$ there exists a* unique *point $p \in P$ such that $dist(p, q) \leq \frac{1}{2} Dist(p, S_i)$, where i is arbitrary and corresponds to the coreset level H_i (from Algorithm* INIT*) to which p belongs. Then the coreset assumption $Dist(q, S) \leq O(1)\varepsilon r_b^* \ \forall q \in P_A$ holds for all points $q \in P_A$ up to a constant factor $O(1)$ where $S = \cup_{i=1}^{\log(n)} S_i$.*

Proof. Let p be the unique virtual position of a point with actual position $q \in P_A$. The virtual position p of a point is the last recorded position of p and the set P contains all the virtual points' positions. From the Proposition assumption we have that every $q \in P_A$ has a virtual point $p \in P$ such that $dist(p, q) \leq \frac{1}{2}\text{Dist}(p, S_i)$. Indeed for some $p \in H_i$ for any level i, if the above claims are satisfied we have

$$\begin{aligned} \text{Dist}(q, S_i) &\leq \text{dist}(q, p) + \text{Dist}(p, S_i) \\ &\leq \frac{1}{2}\text{Dist}(p, S_i) + \text{Dist}(p, S_i) \leq \frac{3}{2}\varepsilon r_b^* \end{aligned}$$

where the first line follows from the triangle inequality, the second line follows from our assumption, and the last inequality proves the proposition. □

Lastly we prove our results on the time complexity and accuracy of our kinematic coreset resulting from a call to the MOVE algorithm. The proof for the following theorem is an extension of similar proofs from [32, 33] and is previously presented in our paper [31].

Theorem 4.3.9. *Let P' be a set of n clients in $\mathbb{R}^2$, $k \geq 1$, and $\varepsilon \in (0, 1/2)$. Let P denote the set of clients after a call to* INIT$(P', k/\varepsilon^2)$ *followed by a finite sequence of calls to the* MOVE *algorithm. Then, the following holds (i) S is a (k, ε)-coreset of P,*

(ii) $|S|$ is of size polynomial in $\frac{k\log(n)}{\varepsilon}$, (iii) The expected execution time of each such call to MOVE *is polynomial in $\frac{k\log(n)}{\varepsilon}$, using an appropriate implementation.*

Proof. (Sketch) First we prove that the procedure UPDATESAMPLE maintains a random sample S_i from P_i. Indeed, by the two last lines of the procedure, we have that S_i contains all the points in P_i that were assigned random values $r(p) \in (0,1)$ that are less than $k\log_2 n/|P_i|$. Hence, S_i is a uniform random sample of expected size $k\log_2 n$ from P_i. By Hoeffding-Chernoff inequality, the size of S_i is $\Theta(k\log n)$.

The correctness of the INIT algorithm was proved in [33]. We claim that after moving a client using the MOVE procedure, then the coreset has the same properties as the output of the INIT algorithm. Indeed, it is easy to check that UPDATESAMPLE maintains a uniform random sample of the set of points in each level. Since the update takes expected $O(\log n)$ time and there are $O(\log n)$ levels, the overall update time is polynomial in $\log n$ if none of the sample sets is updated.

By Lines 5 and 6 of Algorithm 6, the keys $r(q)$ of the samples $q \in S$ have values at most $k\log n/|P_i|$. When we insert or delete a point from $|P_i|$ and update the sample, this threshold changes very little, from $k\log n/|P_i|$ to $k\log n/(|P_i| \pm 1)$. The probability that the random number $r(q) \in (0,1)$ is in this gap for one of the points $q \in P_i$ is $1/|P_i|$, so, in expectation, we will update the sample S once in every $O(n)$ insertions/deletions to a set P_i of size n, and therefore the amortized update time when the sample set is updated is only $O(1)$.

Note that for the last levels the sample (consisting of a very small set of points) will be changing very frequently (since $|P_i|$ is small), however, since these sets are small, reconstructing the corresponding heaps will also be fast. In general, if the reconstruction time and size of a set of points in a specific level is m, then its sample will be updated only once every $O(m)$ times, which yields an overall of $O(1)$ update time per level.

Assuming that there is no change at the sample, we update in Algorithm 3, our heaps of closest and farthest points from the current coreset. When a point moves, each heap changes by at most one inserted/deleted point, and thus can be easily updated in $O(1)$ time. Since we have only $O(\log n)$ levels, the total update is again

polynomial in $O(\log n)$. □

4.4 A Constant Factor Approximation for the k-Connected Center Problem

We make a brief observation that constant factor (α-approximate) solutions, $\tilde{C}$, to the k-center problem are also constant factor (β-approximate) solutions to the k-connected center problem where $\alpha > 1, \beta > 1$. In particular the following result was proved in the paper [45]

Theorem 4.4.1. *For every $\alpha \geq 1$, an α-approximation to the k-center of a set $P \subseteq \mathbb{R}^d$ is an $(3 + 2\alpha)$-approximation to the k-connected center of P.*

This implies that well-known algorithms, such as the 2-approximation algorithm by [53], can be applied to our k-connected center problem where the resulting approximation error ($\beta > \alpha$) can be computed. We note however, that this may not be true for the k-connected center problem with reachability. The latter question is a topic of future research.

4.5 Empirical Results

We extensively test the computational time efficiency of computing a kinematic coreset, and of computing a k-center or connected k-center cost over this coreset, as well as the resulting approximation costs as a function of desired coreset size. Simulation results over large, up to $n = 2000$ points, data sets show the asymptotic properties of our coresets. We also implement our algorithm on a small $n = 5$ example problem in a hardware implementation to provide intuition behind why coresets work and demonstrate online adaptation of robots to *a priori* unknown client movement. We will describe both our hardware implementation and numerical studies in detail in the following sections.

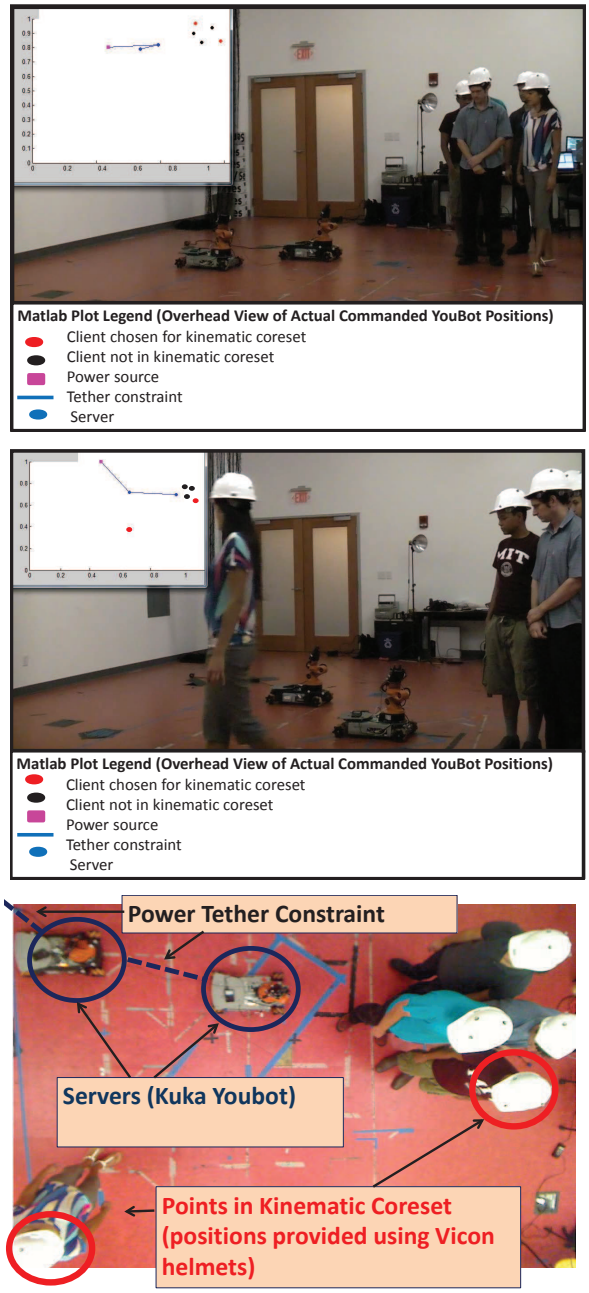


Figure 4-5: Side and overhead views of hardware experiments for heterogeneous Kuka Youbot (server) and human (client) systems. Arbitrary intial positions with coreset (left), servers tracking moving client (middle), overhead view of clients divided into two clusters and resulting coreset (right). Matlab plots show computed kinematic coreset points (red), commanded Youbot positions (blue), and power tether constraints (blue line).

4.5.1 Hardware Experiment

We implemented our kinematic coreset algorithm for a heterogeneous system consisting of $n = 5$ human clients and $k = 2$ robot servers. The five clients were instructed to walk for 10 minutes within the sensing envelope of a Vicon motion capture system, where their positions were sent in realtime to a single Intel Core 2 Duo 2.4 GHz computer running our algorithm. No *a priori* knowledge of the clients' movements was provided to the two servers, which were Kuka Youbot omnidirectional ground robots running the Robot Operating System. Figure 4-3 shows an overhead view of our hardware setup.

Using Matlab R2012a and the CVX convex optimization software [56], kinematic coresets of both two and three points were maintained and used to calculate the connected k-center costs from Equation (3.3.1). Figure 4-5 shows the movement of the clients around the room and the resulting choice of the two point coreset S maintained by repeated calls to MOVE. In addition, the optimal cost over the five clients was calculated and used for positioning the servers as described in [46]. These optimal cost calculations of $n^{O(k)}$ computation time were made possible due to the small number of agents in the system, and thus were used to evaluate our algorithm's performance. Table 4.2 shows the computation time and solution accuracy of our algorithm compared to the optimal connected k-center solution and a naïve sampling strategy.

With our experiment, we were able to demonstrate the ability of our algorithm to detect newly formed clusters. The plots in Figure 6-5 show that although the coreset is only of two points, a representative point (red) is found in every cluster of clients, which is a driving factor for the low resulting error of $\varepsilon = 0.14$ with respect to the optimal solution. In addition, the server position computation takes 2.2 s, which is a factor of 97$\times$ faster. In contrast, a sample set of two points chosen randomly often misses one of the client clusters, thus resulting in a higher $\varepsilon = 0.5$ approximation cost. We expect that for the case where all clients are equally distanced, the solution computed over a kinematic coreset would produce similar approximation costs to

that of a uniform random sample. However, the clustering of clients often arises in practice, especially for large data sets. This small scale implementation demonstrates that the properties we prove for large systems similarly holds for small systems where the $O(\cdot)$ notation is irrelevant.

Metric	KC 2 Pts	U 2 Pts	KC 3 Pts	U 3 Pts
AvgCost/OPT	1.14	1.50	1.02	1.30
VarCost	0.10	0.16	0.08	0.11
OPTtime/Time	97	102	19	19
VarTime (sec)	0.29	0.34	0.41	1.19

Table 4.2: This tables summarizes the result of our hardware experiment employing two servers and five clients. Computing new server positions over a kinematic coreset (KC) of two points is 97× faster with approximation cost of $\varepsilon = 0.14$ compared to performing computation over entire input set of five points. In contrast, naïvely sampling two input points at random (U) produces an approximation cost of $\varepsilon = 0.5$ at comparable computational speed. Calculations of server positions using three points shows similar trends.

4.5.2 Numerical Simulation

We present empirical results for update time, and quality of the coreset S against different input set sizes n. Our test scenario is of an input set of points, P, moving randomly between depots located at three corners of the environment. In particular, at the beginning of a run we randomly select a subset of 10 points from all depots that choose with equal probability one other depot to move to. This is representative of a situation where clients are scouting three areas of major interest where some vehicles may be recalled to other areas of higher interest. We compare the performance of *uniform random sampling, static bicriteria* and *kinematic coresets* for maintaining a representative set of the input P. We specificy our three compared methods below where poly(x) means "polynomial in x":

Uniform Random Sampling: the sample set S is a uniform sample of $m = \text{poly}(\frac{k\log(n)}{\varepsilon})$ points from the input set P.

Static Coreset: the sample set S in this case is of cardinality $m = \text{poly}(\frac{k\log(n)}{\varepsilon})$ and is a (k, ε)-coreset returned from Algorithm INIT computed from scratch every time the positions of input set P are updated.

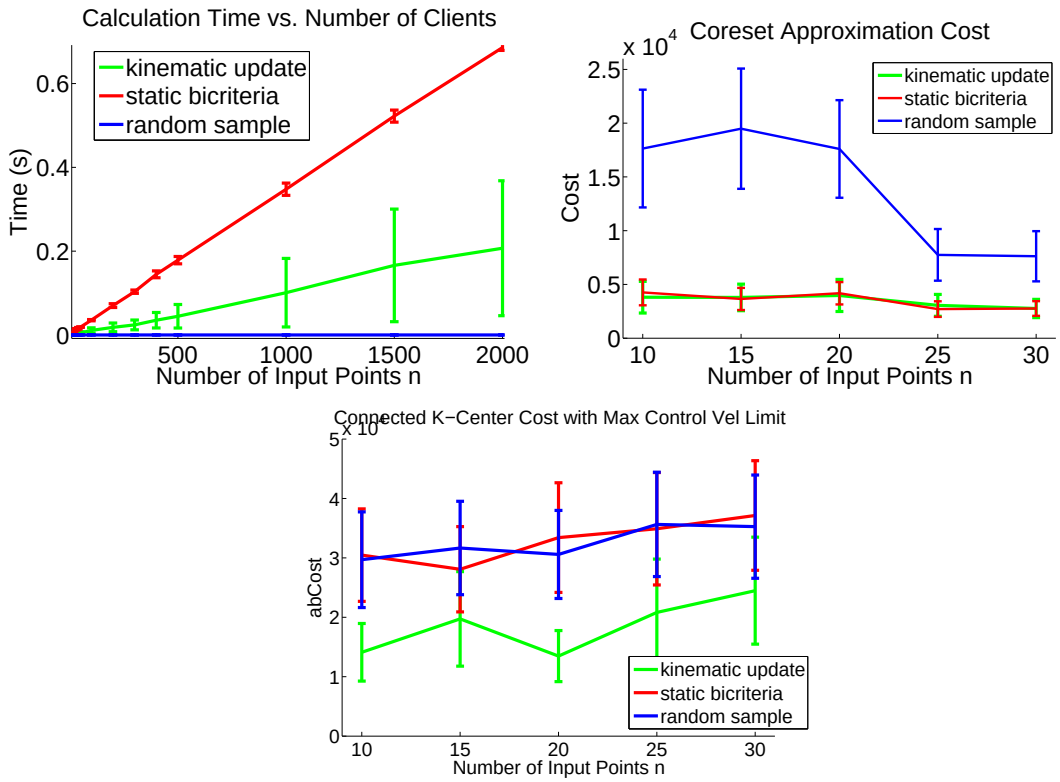


Figure 4-6: These plots show calculation time to compute updated representative set after each position update for points in the input set P for up to $n = 2000$ input points averaged over 500 runs, and the cost over each representative set after each position update for points in the input set P for up to $n = 20$ input points averaged over 500 runs. Results show that kinematic updates perform comparably to using a static coreset in terms of accuracy and random sampling performs up to $5\times$ worse than both of these methods. The last plot shows the case of server velocity constraints where a kinematically updated coreset outperforms a statically updated coreset since in the former case there is consistency between iterations.

Kinematic Coreset: the sample set S in this case is of cardinality $m = \text{poly}(\frac{k\log(n)}{\varepsilon})$ and is a (k, ε)-coreset that is updated using Algorithm MOVE each time the positions of the input points in P are updated.

We measure performance between all three methods in three different ways. First we compare the time needed to update the representative set S. Secondly, we compute the coreset cost $\text{dist}_{p\in\text{P}}(p, S)$ which is how well the entire input set P is approximated by S. Lastly, we analyze the connected k-center cost from Equation (3.13) that takes into account a communication constraint between centers, with an added dynamic constraint on the vehicles that limits how far the centers can move between consecutive iterations, true for any physical system. Calculation of the connected k-center cost demonstrates that kinematically updated coresets are the most cost effective for physical systems that cannot tolerate arbitrarily different solutions (since centers cannot move infinitely fast).

Figure 4-6 bolsters our main time complexity result from Theorem 4.3.9 and indeed demonstrates that the updates for the kinematic coreset are updated much faster, providing a larger computational complexity advantage over the updates for the Static Bicriteria coreset as n increases. Additionally, Figure 4-6 demonstrates that the coreset which is updated kinematically provides similar k-center cost as compared to the Static Bicriteria coreset in stark contrast to a purely random sample of the input client set which has minimal computational complexity but performs up to $5\times$ worse than both the kinematic and static bicriteria algorithms.

For our simulation we do not compute the exact connected k-center cost which takes exponential time in k to compute as discussed in [46], rather we compute a relaxation where we pair every coreset point in S to a unique center as described in Algorithm 7. Figure 4-6 shows that the kinematic coreset performs *better* than the coreset computed using Static BiCriteria for a cost that takes into account displacement constraints on the centers between iterations. This is because the MOVE algorithm updates the coreset intelligently as points move whereas the static bicriteria calculates a new coreset from scratch each iteration and thus has no consistency between iterations.

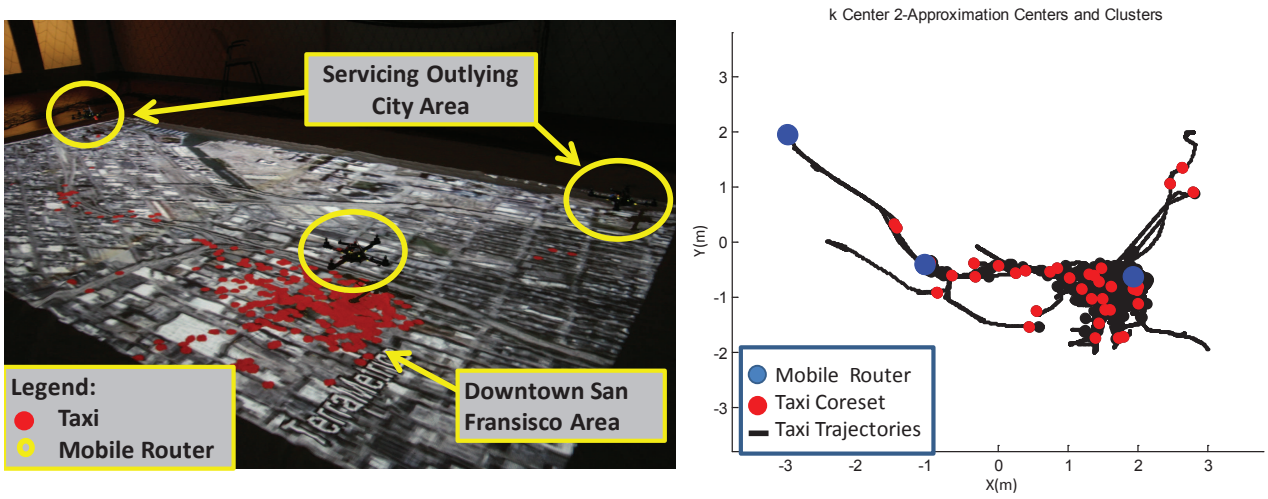


Figure 4-7: A hardware implementation of a kinematic coreset over real San Fransisco taxi data for 500 taxis. Taxi trajectories were reflected in real time over a window of 10 minutes in San Fransisco. These data points were taken from an online source [69] and our router placement algorithms were not given information regarding these trajectories *a priori*.

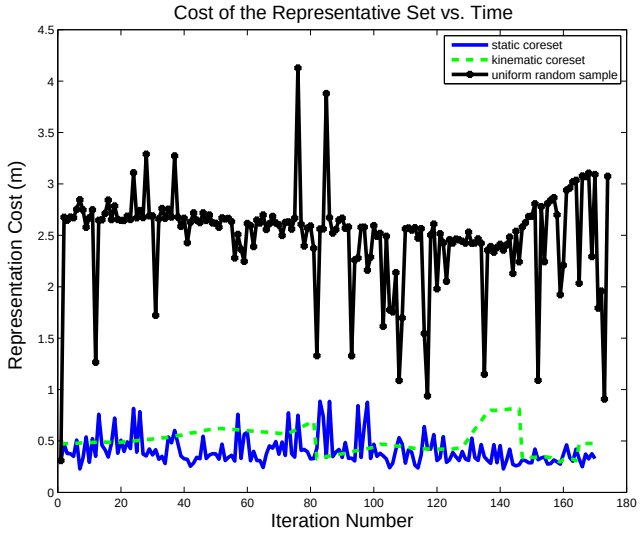


Figure 4-8: A plot of the cost $\max_{p \in P} \min_{s \in S} dist(p, s)$, ie. how well the input set of clients P is represented by the sparse set S. The sparse set S is computed as a uniform random sample of points from P (black), a static coreset (blue), or a kinematic coreset (green).

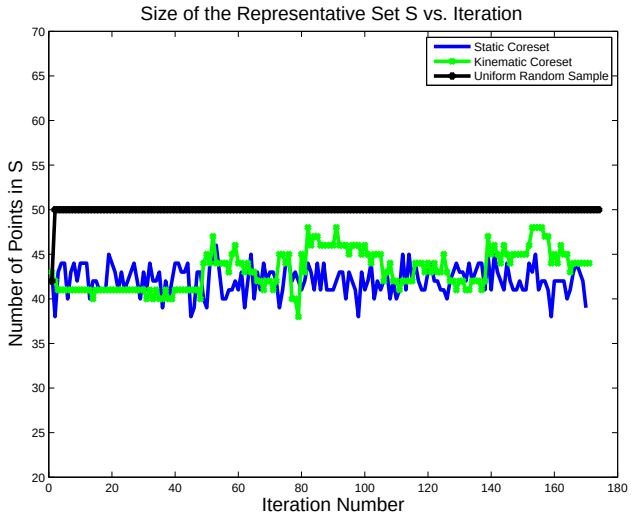


Figure 4-9: This plot compares the size of the representative set S for every iteration where a new router placement is computed. The data shows that a constant sized representative set of 50 taxi points are used for the uniform random sample whereas both the static and kinematic coresets < 50 points but represent the entire client set with a much smaller cost.

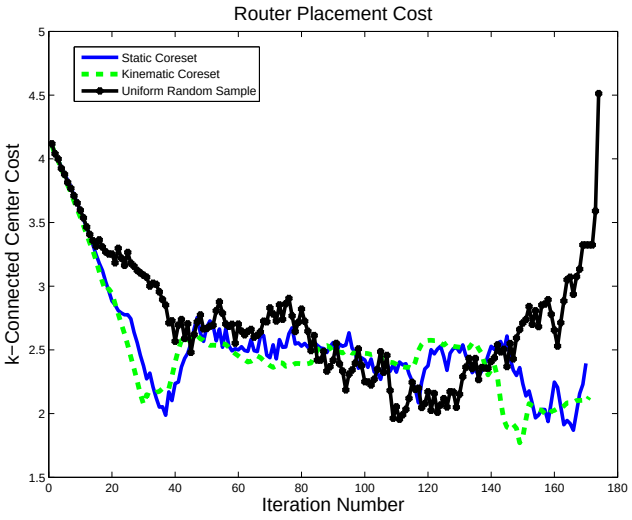


Figure 4-10: This plot shows the resulting k-connected center cost for each router placement over 180 iterations. For the case where taxis roam to the outskirts of the city (iterations 140-180) the uniform random sample set cannot maintain a bounded error on the solution and thus grows arbitrarily large due to misrepresentation of these outlier points in the client set.

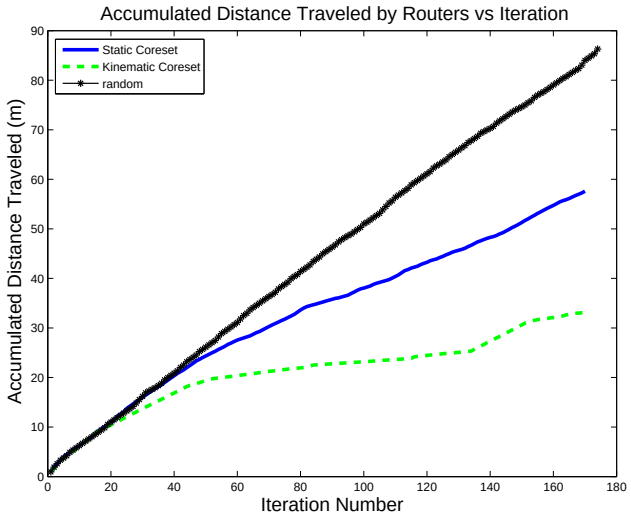


Figure 4-11: This plot shows the accumulated distance traveled by the routers and compares the stability of the resulting router trajectories for each choice of representative set. The advantage of using a kinematic coreset over a static coreset for dynamic points is clear from this plot which shows that the consistency provided by a kinematic coreset across iterations allows for the same error bound in cost as a static coreset but with much less travel required of the routers.

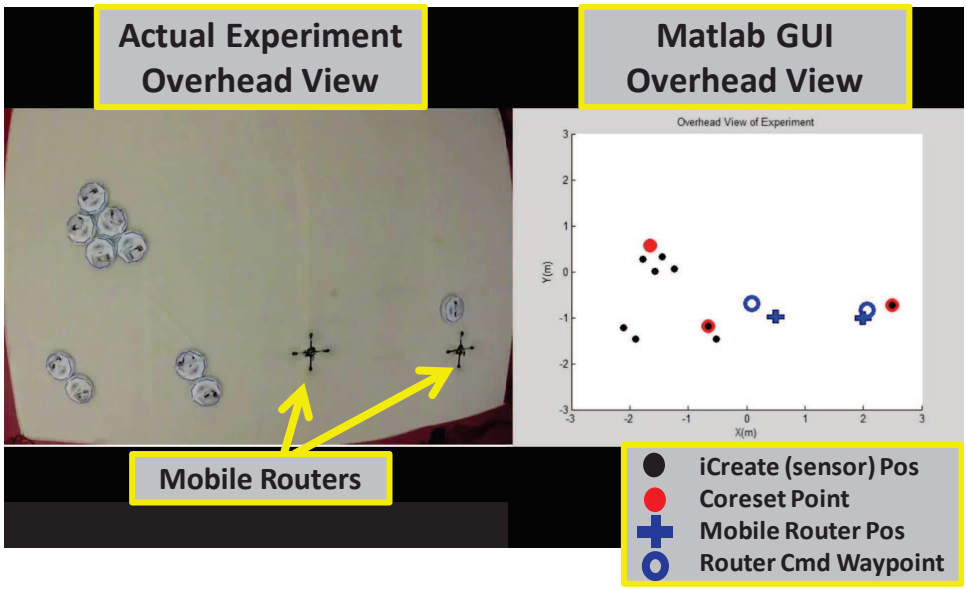


Figure 4-12: Snapshot of hardware experiment involving a heterogeneous robot platform with 10 mobile sensor clients (iRobot Create vehicles) and two mobile routers (AscTec Hummingbird quadrotors).

4.6 Conclusion

In this chapter we have provided an algorithm for maintaining a sparse set of representative clients that is updated as the client vehicle team moves arbitrarily through the environment. Additionally we present theory that guarantees that our representative set can be updated in time polynomial in $(k \log(n)/\varepsilon)$ and provide the same error bounded approximate k-center cost as the case of computing the entire representative set from scratch using static coresets. Our empirical results additionally show that for systems of practical interest that have physical limitations on how fast center vehicles can move, updating the existing coreset kinematically is favorable over computing a static coreset since consistency is maintained over consecutive iterations.

Algorithm 2 A (k, ε)-coreset $\hat{S}$ for P

input : A set $P \subseteq \mathbb{R}^d$ of n robots, $k \geq 1$ centers, and a constant $\varepsilon > 0$
output: An (ε, k)-coreset $\hat{S}$ of size $O(k \log n/\varepsilon^d)$ for P

$i \leftarrow 0$, $S_0 \leftarrow P$
while $|P_i| > k$ **do**
    Pick a random set $T_i \subseteq P_i$ of k robots from S_i
        Remove half of the closest robots $Q_i \subseteq P_i$ to T_i `/* Continue recursively with the remaining robots. */`
    $P_{i+1} \leftarrow P_i \setminus Q_i$
    $i \leftarrow i + 1$
$T \leftarrow T_1 \cup \cdots \cup T_{i-1} \cup P_i$
for *each* $p \in T$ **do**
    Construct a d-dimensional grid G_p of side length $(\varepsilon/\sqrt{d}) \cdot \text{Dist}(P, T)$ that is centered at p Pick an arbitrary representative robot $q \in P$ from every non-empty cell of G_p
$\hat{S} \leftarrow$ the union of representatives that were selected at Line 11 Return $\hat{S}$

Algorithm 3 INIT(P, m)

Input: A set P of n clients, and an integer $m \geq 1$
Output: A set S that satisfies Theorem 4.3.3.

$i \leftarrow 1$; $P_1 \leftarrow P$
while $|P_i| > m$ **do**

    $S_i \leftarrow$A uniform random sample of m points from P_i, with replacement.

    $H_i \leftarrow$ A set of $c_i|P_i|$ points $p \in P_i$ with the smallest distance $\text{Dist}(p, S_i)$ for some $c_i \in (1/4, 3/4)$.

    $P_{i+1} \leftarrow P_i \setminus H_i$ $i \leftarrow i + 1$
$S_i \leftarrow P_i$; $H_i \leftarrow P_i$ $S \leftarrow S_1 \cup \cdots \cup S_i$
return S

Algorithm 4 UPDATE$(p, i, action)$ Insert/Delete p from P_i

Input: A client $p \in P_i$, and $action \in \{\text{insert}, \text{delete}\}$

UPDATESAMPLE$(p, i, action)$
if *S_i was changed during the execution of previous line* **then**
    Reconstruct levels $i, i + 1, i + 2, \ldots$ **return**
Insert/Delete p to/from its heap $H \in \{H_i, \overline{H}_i\}$. In case of ties, choose smallest heap.
if $H = \overline{H}_i$ **then**
    UPDATE$(p, i + 1, action)$
BALANCE(i) **return**

Algorithm 5 BALANCE(i) Balance the pair of heaps at level i

Input: A coreset level $i \geq 1$

if $|\overline{H}_i| \notin [1/4, 3/4]$ **then**
  $p \leftarrow$ root of the larger heap in $\{H_i, \overline{H}_i\}$
    UPDATE($p, i,$ delete)
    UPDATE($p, i,$ insert) `/* p is inserted to the smaller heap */`

Algorithm 6 UPDATESAMPLE($p, i, action$): Update the sample S_i with the deletion/insertion of p

Input: A client p that should be inserted/deleted from P_i according to $action \in \{\text{insert}, \text{delete}\}$.

if $action =$ insert **then**
  $P_i \leftarrow P_i \cup p$
  $r(p) \leftarrow$ A random number, sampled uniformly over the interval $[0, 1]$
else
  $P_i \leftarrow P_i \setminus \{p\}$
Remove from S_i every $q \in S_i$ such that $r(q) > k \log_2 n/|P_i|$ Insert to S_i every $q \in P_i$ such that $r(q) \leq k \log_2 n/|P_i|$

Algorithm 7 RELAXATION(P, C, γ) Compute connected centers that are attainable from C_{t-1}

Input: A set P of k clients, the current set $c'_1, \cdots, c'_k$ of centers, and max velocity bound $\gamma > 0$
Output: A set C of k centers and their cost r

for $i \leftarrow 1$ *to* k **do**
  `/* Uniquely assign each center to a close client */`
14 $p_i \leftarrow \arg\min_{p \in P} \|c'_i - p\|$
  $P \leftarrow P \setminus \{p_i\}$

$$(C, r) \leftarrow \arg \min_{C=\{c_1,\cdots,c_k\} \subseteq \mathbb{R}^2, r \geq 0} r$$
$$\text{s.t. } \forall i = 1, \cdots, k$$
$$\|c_i - p_i\| \leq r,$$
$$\|c_i - {c'}_i\| \leq \gamma.$$

return (C, r)

Algorithm 8 MOVE(p, p_a, i): Move $p \in P_i$ to its actual position p_a.

Input: A virtual client $p \in P_i$, its actual position p_a, and an integer $i \geq 1$.

$h_i \leftarrow \max_{q \in H_i} \text{Dist}(q, S_i)$ /* S_i is the coreset of level i. */
if $p_a, p > h_i$ **then**
  MOVE$(p_a, p, i+1)$
    /* Check next levels recursively */
else if $p \in H_i$ *and* $\text{dist}(p_a, p) \leq \text{Dist}(p, S_i)/2$ **then**
  return /* No update */
else if $\text{dist}(p, S_i), \text{dist}(p_a, S_i) \leq h_i$ **then**
  Replace p with p_a in H_i /* Heap update */
  if $p \in S_i$ **then**
    Reconstruct levels $i, i+1, i+2, \ldots$ /* Coreset update */
  return
/* At this line $p \in H_i$ and $p_a \notin H$, */
/* or vice versa */
if $\text{dist}(p, S_i) \leq h_i$ **then**
  /* $p \in H_i$ but $p_a \notin H$ */
  UPDATE(p, i, delete) UPDATE(p_a, i, insert)
else
  /* $p_a \in H_i$ but $p \notin H_i$ */
  UPDATE(p_a, i, delete) UPDATE(p, i, insert)
return

Chapter 5

Real-world Communication with Unknown Client Locations

5.1 Introduction

In this chapter of the thesis our focus is to develop a new problem formulation for the router placement problem that gives a realistic treatment to the wireless communication aspect. In other words we replace the assumption that signal quality is idealistically mapped to Euclidean distance. Along these lines we provide two main contributions: 1) measure a new mapping for signal quality along each link using a novel technique involving signal processing of the full channel for each wireless link and 2) we use the resulting realtime channel feedback to formalize a new optimization problem that can be solved for placing routers such that the required communication demands of the clients are satisfied over the network. We connect this chapter of the thesis to earlier chapters by showing that our new optimization problem is surprisingly simple (comprised of quadratic link costs) and can be reduced to our previous formulation; this allows us to use Algorithm 1 from Chapter 3 for solving for router placements as before. This results in router placements that are simple to compute, yet capable of supporting variable communication demands of the clients in general environments.[1]

[1]This work has appeared in [47].

5.1.1 Assumptions

In this chapter of the thesis we relax many of the assumptions of the previous chapters of the thesis. In particular, we assume that

(i) Signal strength is based on a directional signal strength map that we derive (*not* the Euclidean disk model).

(ii) Current positions of client vehicles are *not* known, router positions are known.

(iii) Client communication demands are allowed to be heterogeneous with respect to other client demands, and also variable over time.

(iv) We mainly consider dimension $d = 2$ although concepts are extensible to $d = 3$. Though $d = 3$ is a topic of future work.

(v) We consider a single interference region so that all nodes communication over the same frequency.

(vi) Routers travel along piecewise linear paths where the length of the linear portion of the path must be $\geq \lambda/2$ and the router velocity over this displacement is assumed constant (or alternatively, router positions must be known to mm accuracy) in order to perform SAR.

(vii) We assume omni-directional antennas.

(viii) A signal strength profile in constructed for every straight-line path of length $\geq \lambda/2$. It is assumed that the wireless channel is constant (ie. client and environment are static) during the time window needed for a router to travel a distance of $\lambda/2$ which for our implementation is on the order of $6cm$.

5.1.2 Results Snapshot

First, we introduce an innovative approach for mapping communication quality to robot placement. Our approach calculates a mapping between a robot's current position and the signal strength that it receives along *each spatial direction*, for every

wireless link with other robots. This is in contrast to existing methods [34, 116], which compute an aggregate signal power at each position but cannot distinguish the amount of signal power received from each spatial direction. In this spirit, the paper [29] uses received signal strength over a rotating antenna (by rotating the antenna's robot platform) in an attempt to infer the direction of highest RSSI and learn bearing information relative to another robot. However, using the full channel information (phase in addition to received signal strength) allows us to gain much richer information, i.e. a full directional profile of signal strengths capturing complex phenomena such as multipath, by moving along an arbitrary straight-line path. Our approach combines the best attributes of both the disk model and stochastic methods: Like the disk model, we can compute our mapping without knowledge of the environment and its obstacles, or a model of channel's distribution. Like the stochastic methods, our approach accurately captures the mapping of the signal strength directionality and hence can help multi-robot systems satisfy their desired communication demands in real-world environments.

Second, we construct an optimization for positioning a team of robot routers for providing communication coverage to an independent set of client vehicles using the directional information provided by our mapping. We aim for a solution that is adaptive to variable communication quality demands by the clients, as well as changes in the wireless channels due to natural fluctuations or a dynamic environment. Being able to measure the profile of signal strength across spatial directions in real-time yields a much more capable controller. For example, the controller uses the profile to find directions of movement that yields better communication quality. The profile also helps estimate the confidence with which the controller can improve signal power by navigating the robot along any of these directions. The confidence can then be used to control the speed of the robot, thereby improving stability and convergence time. Furthermore, the controller can leverage the entire profile of signal strength across directions, to optimize communication with multiple robots by choosing a direction of movement corresponding to a strong signal that strikes trade-offs between competing demands. Interestingly, we show that such optimizations can be formulated in terms

of simple quadratic costs, similar in spirit to the Euclidean disk model. Further, they can be made independent of environment-dependent parameters, or even client positions.

A key question remains: How do we calculate the signal strength along each spatial direction? The naive approach would use directional antennas, a type of antenna that receives signals only from a cone in space. Unfortunately, directional antennas are bulky and unwieldy [87], making them ill-suited for small agile robots. To address this problem, we employ Synthetic Aperture Radar (SAR), a technique that leverages movement to emulate a high-resolution directional antenna [37]. In order to achieve this, we must derive a method for implementing SAR using off-the-shelf wireless cards, a challenging task since these devices are not intended for this purpose.

We implement our method in a multi-robot testbed that has two robotic routers serving three robotic clients. We conduct our experiments in different indoor environments without providing the robotic controller the environment map or the clients' positions. We observe the following: 1) Our system consistently positions the robotic routers to satisfy the robotic client demands, while adapting to changes in the environment and fluctuations in the wireless channels; 2) Compared to the disk model [23,65] and the stochastic approach [74, 100] under identical settings, our system converges to accurately satisfy the communication demands, unlike the disk model, while significantly out-performing the stochastic method in terms of empirical convergence rate (see Fig. 5-13 in Sec. 5.6.4).

5.1.3 Contributions Summary

The contributions of this chapter are three-fold: 1) We present a method to enable a robotic receiver to find the profile of signal strength across spatial directions for each sender of interest. To this end, we perform synthetic aperture radar (SAR) techniques using standard Wi-Fi packets exchanged between two independent nodes; 2) We develop an optimization that leverages this directional signal profile to position robotic routers to satisfy heterogeneous communication demands of a network of robotic clients, while adapting to real-time environmental changes; 3) We implement

our design and demonstrate its empirical gains in comparison to both the disk model and the stochastic method.

5.1.4 Notation

Table 5.1: Common Notation for Chapter 5

$G(V, E)$	$\triangleq$	Communication graph with nodes V and edges E.
ε	$\triangleq$	A given error tolerance where $\varepsilon > 0$.
p_j	$\triangleq$	Client j position in $\mathbb{R}^d$.
c_i	$\triangleq$	Robot router i position in $\mathbb{R}^d$.
$\text{dist}_M(x, y)$	$\triangleq$	Mahalanobis distance between $x \in \mathbb{R}^d$ and $y \in \mathbb{R}^d$.
$f_{ij}(\theta)$	$\triangleq$	Mapping from angles (relative to router heading) to signal strength for link (i, j).
$\vec{v}_{\theta_{\max}}$	$\triangleq$	Relative direction of highest signal strength.
σ_{ij}	$\triangleq$	Confidence in the max signal strength direction for link (i, j).
θ_{max}	$\triangleq$	Direction of maximum signal strength for a particular link.
α_j	$\triangleq$	Importance of client j, $\alpha_j > 0$.
w_{ij}	$\triangleq$	Service discrepancy for link (i, j), $w_{ij} \geq 0$.
$\tilde{q}_{ij}$	$\triangleq$	Actual link quality (ESNR) for link (i, j) (one-to-one mapping with throughput of link in Mb/s).
q_j	$\triangleq$	Demanded communication quality (ESNR) for client j.

5.2 Problem Formulation

We now formally describe our problem formulation. A schematic interpretation can be seen in Figure 5-1.

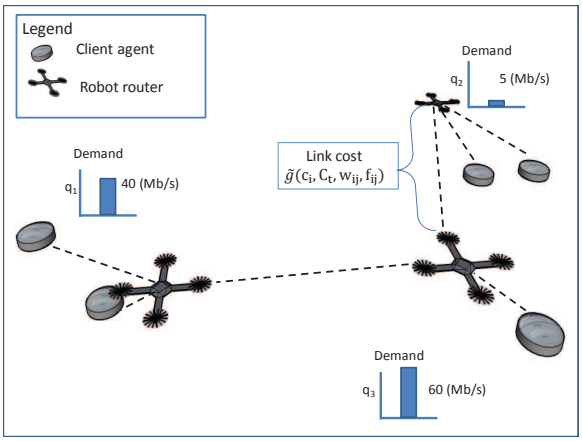


Figure 5-1: A problem sketch demonstrating our goal of positioning routers to satisfy heterogeneous communication demands of different clients over the network.

5.2.1 Problem Statement

Our goal is to position the robotic routers to provide adaptive wireless communication coverage to the clients, while allowing variable communication quality demands for all clients, and where exact client positions are unknown. Note that all quantities in this section are time-dependent; we omit this explicit dependency in our notation for simplicity. For each client $j \in [n] = \{1, \ldots, n\}$, we define demanded communication quality $q_j > 0$ and achieved communication quality $\tilde{q}_{ij}$ to each router i (where $i \in [k]$), both expressed in terms of *Effective Signal to Noise Ratio* (ESNR) that has a direct mapping to rate in Mb/s [59]. We choose to work with ESNR values rather than rates since the rates supported on a link are discretized (non-continuous). Additionally, let every client j be given an importance $\alpha_j > 0$. We define the notion of *service discrepancy* for each pair of robots (i, j) to be the difference between the demanded and achieved communication quality scaled by the importance of the client.

$$w_{ij} = \max(\alpha_j \frac{(q_j - \tilde{q}_{ij})}{q_j}, 0) \tag{5.1}$$

Physically, this is the fraction of the client's communication demand that remains to be satisfied, scaled by α_j. Denote by $c_i \in \mathbb{R}^d$ the position of the ith robot router

and by $p_j \in \mathbb{R}^d$ the position of the jth client and $C_t = \{c_{1_t}, \ldots, c_{k_t}\}$ is the set of all router positions at time t. Given a cost g as a function of signal quality, communication demands, and agent positions, we wish to position each robotic router to minimize the largest discrepancy of service between routers and clients. However, the true form of this function g has an intricate dependence on the environment and the positions of the client and router. Thus an inherent challenge to solving this problem is approximating the influence of spatial positioning on communication quality in a way general to different environments. We have a joint goal to 1) find $f_{ij} : [-\pi, \pi] \to \mathbb{R}$ (a relation capturing directional information about the signal quality between i and j), and an approximation $\tilde{g}$ of g that is a cost characterizing the anticipated communication quality for the router-client pair (i, j) at a proposed router position c_i, and 2) use this cost to optimize router positions to minimize the service discrepancy to each client. Formally,

Problem 5.2.1. *Find a mapping*

$$f_{ij} : [-\pi, \pi] \to \mathbb{R} \tag{5.2}$$

that maps spatial direction to wireless signal strength directly from channel measurements, and a cost for every link (i, j)

$$\tilde{g}(c_i, C_t, w_{ij}, f_{ij}) > 0 \tag{5.3}$$

that is independent of the environment and client positions, has a simple quadratic form, and whose minimization over c_i *directly relates to increasing signal quality. We aim to find robot router positions,* C_{t+1}*, that minimize the maximum service discrepancy over all clients* j *by solving the following min-max optimization problem:*

$$C_{t+1} = \arg\min_{C}\{\max_{j}\min_{c_i \in C}\tilde{g}(c_i, C_t, w_{ij}, f_{ij})\} \tag{5.4}$$

Intuitively, the solution to this optimization problem favors "fair" solutions where the maximum service discrepancy is minimized over all clients. We note that it is important that the resulting cost $\tilde{g}$ is i) reflective of real time wireless signal information, unlike the Euclidean disk model where $\tilde{g}(p, c) = dist(p, c)$ and the dependence on signal quality f_{ij} is ignored, and ii) endowed with a simple structure that is independent of the environment, unlike stochastic sampling approaches where $\tilde{g}$ and may have a complex form and f_{ij} may capture properties of a signal quality distribution that is estimated using received signal strength measurements and is dependent on environmental parameters.

We dedicate the next sections to i) developing a method that computes f_{ij} as the profile of signal qualities along each direction θ for each link (i, j) found directly from channel measurements; and ii) developing an optimization framework that utilizes this directional information to handle trade-offs between competing client demands and position all routers to jointly minimize the maximum service discrepancy over the links in the communication network.

In this section of the paper we show that the problem of positioning a team of k mobile routers to service the heterogeneous communication demands of n clients can be framed as a connected k-center problem. We first derive the framework assuming that we have channel feedback over each link in the network in the form of the directional f_{ij} map introduced in the previous section, as well as a measure of the instantaneous link quality $\tilde{q}_{ij}$ for each link. We then detail our method for attaining this map in Subsection 5.3. Finally we tie together our control framework with our channel feedback technique to obtain an algorithm for positioning mobile routers to achieve a wireless network that is adaptive to sensor mobility and variable communication demands.

5.3 Derivation of Directional Signal Quality Maps

In this section, we develop the first component of the solution of Problem 1; namely, we derive a method to calculate $f(\theta)$, the mapping that captures the strength of the signal from a robotic client to its router along each direction θ, where this mapping can be updated often, roughly once every 6cm of motion.[2]

Before we explain how we compute $f(\theta)$, we describe this function to help understand the information it captures. Assume we have a robotic client and router, where the router moves along some trajectory. We will define the direction θ relative to the tangent to the router's trajectory at each point. Consider the scenario in Fig.5-3(a), where the robotic client is in line-of-sight at $-50°$ relative to the robotic router, which is moving along the horizontal axis. In this case, one would expect $f(\theta)$ to have a single dominant peak at $-50°$, as shown in Fig.5-3(b). Now consider the more complex scenario in Fig.5-3(c), where the environment has some obstacles and one of these obstacles obstructs the line-of-sight path between the router and its client. In this case, $f(\theta)$ would show two dominant peaks at $20°$ and $-30°$ that correspond to the two reflected paths from surrounding obstacles, as shown in Fig.5-3(d).

Advantage over Sampling Methods: One may estimate $f(\theta)$ by sampling the signal power similar to stochastic techniques [74, 100, 116]. In this case, one has to move the router along each direction, compute the power in all these new positions relative to the first, and draw the profile $f(\theta)$. Unfortunately, this approach leads to much wasted exploration. This is because the signal power does not change reliably when the robot moves. For example, if the robot moves for 5 or 10 centimeters, it is very likely that the resulting change in the signal power is below the variability in noise. Hence, measurements of power over short distances are likely to be marred by noise. To obtain reliable measurements of changes in the signal power, the robot has to move significantly along potentially counter-productive paths.

To address this limitation, our approach relies on the channel phase as opposed to the power. Specifically, at any position the wireless channel can be expressed

[2]For simplicity, we denote $f_{ij}(\theta)$ as $f(\theta)$ as we consider only the single link between robotic router i and client j for the rest of this section.

as a complex number $h(t)$ [92]. The magnitude of this complex channel captures the signal power (more accurately, its square-root). The phase of the channel has traditionally been ignored by robotic systems. However, the phase changes rapidly with motion. For Wi-Fi signals at a frequency of 5 GHz, the phase of the channel rotates by π every 3 cm. This far exceeds any rotation due to noise variability. Thus, by measuring channels as complex numbers and tracking changes in its phase as the robot moves, we reliably estimate signal variation without much exploration.

In the next sections, we describe some necessary background in order to understand the signal processing techniques for computing our directional signal strength profiles, or the mapping $f(\theta)$. First we describe the technique of beamforming where the signals received by an array of multiple antennas are post-processed in order to estimate the signal arriving from a desired direction (relative angle) in the presence of noise and interference. Then we describe Synthetic Aperture Radar (SAR) techniques that are more amenable to mobile platforms and in particular we describe how we are able to adapt SAR to be applied on our single omni-directional antenna platforms in a non-radar setting. We explain how to apply SAR for extracting the received signal strength along each direction from changes in channel phase. Note that SAR does not need exploring all directions; the robot can move along its path without extra exploration or sampling. SAR uses the resulting variations in channel phase over distances of a few centimeters to find $f(\theta)$.

5.3.1 Beamforming

In this section we provide some background on beamforming techniques as adapted from the paper [110]. Beamforming is a technique that takes signal inputs from a set of J sensors aligned as shown in Figure 5-2, and linearly combines received signals in order to amplify those arriving from a specific direction. The sensor array is a static, linear array of multiple equally spaced sensors.

The signals $h(k)$ are sampled at each sensor at a given time k and are processed

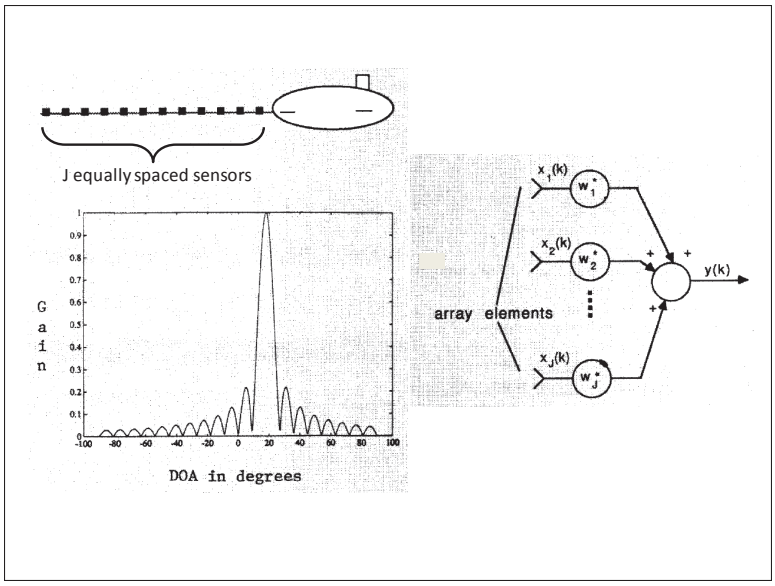


Figure 5-2: A schematic showing a sensor antenna array and a corresponding direction of arrival (DOA) plot for a beamformer steered to 18 degrees. On the right a schematic shows how the received signals are weighted and processed to produce the output from J sensors at time k [110].

to produce the output $y(k)$ according to:

$$y(k) = \sum_{i=1}^{J} w_i(k) * h_i(k) \tag{5.5}$$

where $*$ denotes the complex conjugate and $w_i(k)$ are weights [110] (where the weights are often selected as part of the beamformer design). Because each of the sensors is placed at a different spatial location, a signal propagating from the same source will traverse a different distance when arriving at each of the sensors and this results in a relative phase shift. If we represent the signal as a complex number with a certain phase and amplitude then the narrow beam beamformer output becomes:

$$y(k) = e^{j\omega k} \sum_{i=1}^{J} w_i^* e^{-j\omega\Delta_i(\theta)} \tag{5.6}$$

where the phase elements $d(\theta, \omega) = [1, e^{j\omega\tau_2(\theta)}, \ldots, e^{j\omega\tau_J(\theta)}]^H$ represent the "steering vector" and the $\tau_i(\theta)$, for $2 \leq i \leq J$ are the time delays due to differences in the signal propagation when received at each sensor. This sets the stage for the signal processing

tools that can be used to combine received signals across a multi-sensor array for a particular DOA of interest. Now we shift our attention to SAR techniques where we wish to use a mobile antenna array (or an emulated static array) to produce a full directional profile (along every direction, not just a single direction) in a non-radar setting.

5.3.2 Synthetic Aperture Radar (SAR)

Synthetic Aperture Radar (SAR) enables a single antenna mounted on a mobile device to estimate the strength of the signal received along every spatial direction. We leverage the natural motion of a robotic router to implement SAR and measure $f(\theta)$ for each of its robotic clients. To do so, the robotic router measures the channel $h(t)$ from its client as it moves along any straight line. Note that in this context we wish to perform SAR in a non-radar setting meaning that we wish to use the forward and backward channels (transmitted and received) where the received signal is transmitted from a different node in the network altogether. This introduces a host of new challenges that we discuss in the next section.

The straight line motion that is used to collect channel measurements for computing $f(\theta)$ can be as short as a few centimeters (on the order of $\lambda/2$ which for our implementation was $6cm$). This means that the router can have an updated measurement of $f(\theta)$, for all values of θ, after every straight-line traversal of only a few centimeters. During this short period we assume the channel is constant (for example, that the source is static). Specifically, Let $h(t)$ for $t \in \{t_0, \ldots, t_m\}$ be the $m+1$ most recent channel measurements, corresponding to the robot moving a distance $d(t_0) \ldots d(t_m)$. SAR computes the received signal strength across spatial directions $f(\theta)$ as:

$$f(\theta) = \left| \sum_t h(t) e^{-j\frac{2\pi}{\lambda} d(t) cos\theta} \right|^2, \tag{5.7}$$

where λ is the wavelength of the Wi-Fi signal. We refer the reader to [102] for

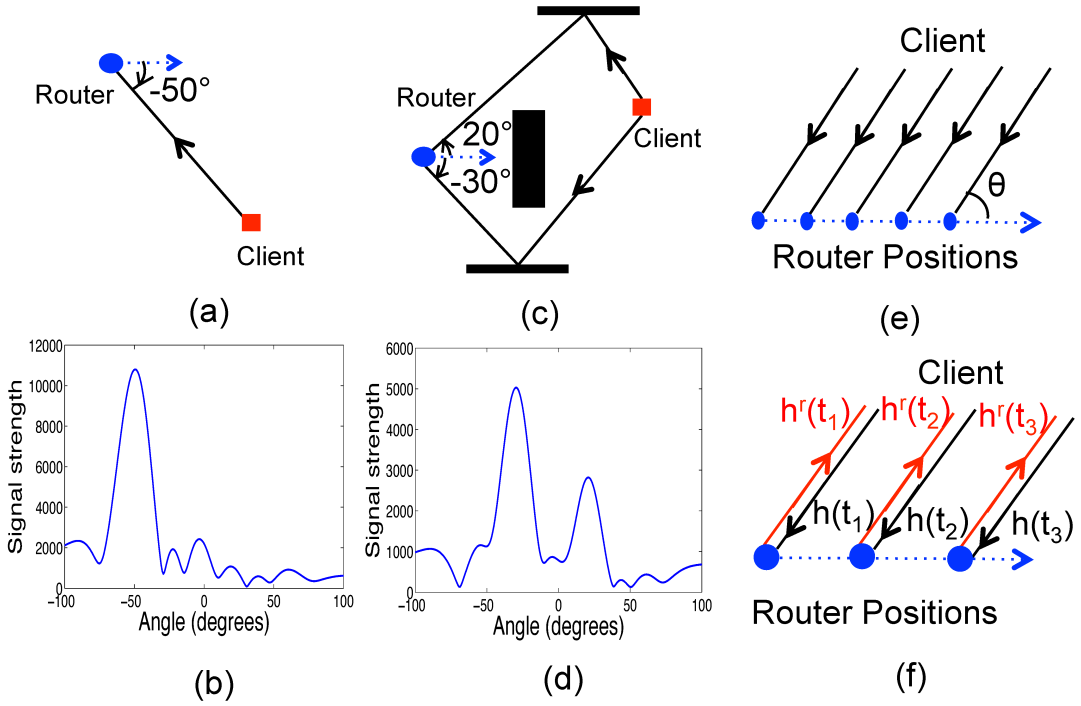


Figure 5-3: (a)/(c) LOS and NLOS topologies annotated with signal paths. (b)/(d) $f(\theta)$ of the signal in LOS and NLOS. (e) Shows how θ is defined in SAR. (f) Shows $h(t_i)$, the forward channel from transmitter to receiver and $h^r(t_i)$, the reverse channel from receiver to transmitter at time t_i.

the analysis of this standard SAR equation. At a high level, the multiplying terms $e^{-j\frac{2\pi}{\lambda}d(t)cos\theta}$ in Eqn. 5.7 project the channels $h(t)$ along the direction of interest θ by compensating for incremental phase rotations introduced by the robot's movement to all paths of the signal arriving along θ.

Note that SAR finds the signal power from every angle θ simply by measuring the channels[3], without needing prior tuning to any given direction. In fact, moving by around a wavelength (about 6 cm) is sufficient to measure the full profile of $f(\theta)$.

Therefore, SAR is a natural choice for autonomous robotic networks since it exploits the mobility of the robots to compute $f(\theta)$. Further, it only requires the robot to move along a small straight line along any arbitrary direction, and does not require it to explore directions counter-productive to the overall coordination goal. Note that SAR requires only the relative position of the robotic router $d(t)$ and both the

[3]Of course, the resolution at which θ is available depends on the number of channel measurements.

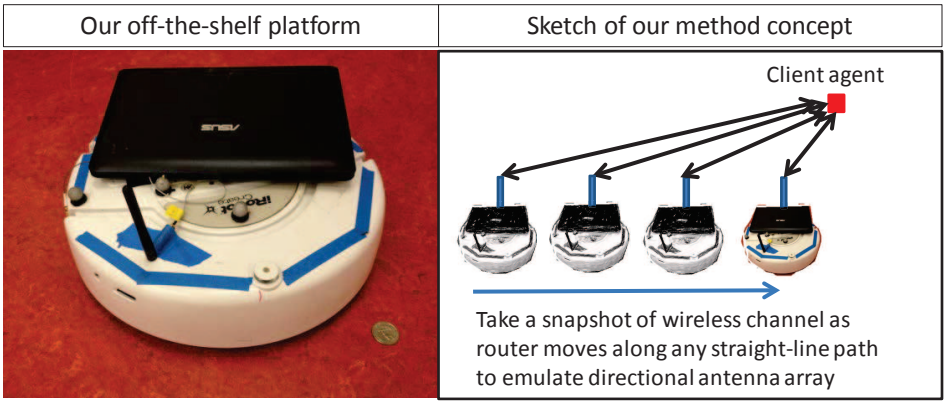


Figure 5-4: This figure shows an image of our off-the-shelf platform, as well as a schematic of our technique for gathering wireless channel information for constructing our directional signal strength maps (this supplements the sketch in Figure 5-3).

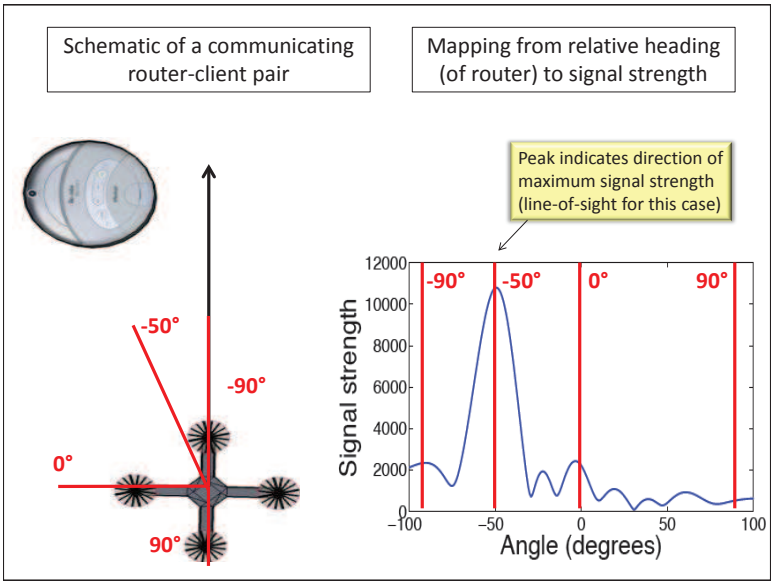


Figure 5-5: A directional signal strength map computed during our experiments. The accompanying schematic shows an interpretation of the information contained in this signal strength map.

magnitude and phase of the channel $h(t)$. It does not require the topology of the environment nor the exact location of the transmitter.

5.3.3 Challenges in Implementing SAR on Independent Wireless Devices

A key challenge in adapting SAR to multi-robot systems is that all past SAR-based solutions [2, 37, 114] are for radar-like applications, where a single device transmits a radar signal and receives its reflections off an imaged object, e.g., an airplane. However, in our scenario the transmitter and receiver are completely independent wireless devices (i.e., the robotic client and router, respectively). This means that the transmitter robot and the receiver robot have different frequency oscillators. In practice, there is always a small difference between the frequency of two independent oscillators. Unfortunately, even a small offset Δ_f in the frequency of the oscillators introduces a time varying phase to the wireless channel.

For instance, let $h(t_0)$, $h(t_1)$, ..., $h(t_m)$ be the actual wireless channel from the robotic client to the robotic router at times $t_0, t_1, \ldots, t_m$. The channel observed by the router from its client $\hat{h}(t_0)$, $\hat{h}(t_1)$, ..., $\hat{h}(t_m)$ are given by:

$$\hat{h}(t_0) = h(t_0), \quad \hat{h}(t_1) = h(t_1)e^{-2\pi\Delta_f(t_1-t_0)}, \ldots, \quad \hat{h}(t_m) = h(t_m)e^{-2\pi\Delta_f(t_m-t_0)}. \quad (5.8)$$

Hence, the phase of the channels are corrupted by time-varying values due to the frequency offset between the transmitter and the receiver. Fortunately, we can correct for this offset using the well-known concept of channel reciprocity [92]. Specifically, let $h^r(t)$ denote the reverse channel from the robotic router to its client, as shown in Fig. 5-3(f). Reciprocity states that the ratio of the forward and reverse channels stays constant over time, subject to frequency offset, i.e., $h^r(t) = \gamma h(t)$, where γ is constant. Further, the frequency offset in the reverse direction Δ_f^r is negative of the offset in the forward direction, i.e., $\Delta_f^r = -\Delta_f$. Thus, the observed reverse channels

$\hat{h}^r(t_0)$, $\hat{h}^r(t_1)$, ..., $\hat{h}^r(t_m)$ are given by:

$$\hat{h}^r(t_0) = h^r(t_0), \quad \hat{h}^r(t_1) = h^r(t_1)e^{2\pi\Delta_f(t_1-t_0)}, \quad \ldots, \quad \hat{h}^r(t_m) = h^r(t_m)e^{2\pi\Delta_f(t_m-t_0)}. \tag{5.9}$$

Multiplying Eqn. 5.8 and 5.9 above and using $h^r(t) = \gamma h(t)$, we have $\hat{h}(t)\hat{h}^r(t) = h(t)h^r(t) = \gamma h(t)^2 \Rightarrow h(t) = \sqrt{\hat{h}(t)\hat{h}^r(t)/\gamma}$. Hence we re-write Eqn. 5.7 as:

$$f(\theta) = \left| \sum_t \sqrt{\hat{h}(t)\hat{h}^r(t)} e^{-j\frac{2\pi}{\lambda}d(t)cos\theta} \right|^2, \tag{5.10}$$

where the constant scaling γ is dropped for simplicity. Hence, to measure $f(\theta)$ the router and client simply need to measure their channels at both ends.[4] This signal processing is done on each node (decentralized). In the next section, we explain how we leverage $f(\theta)$ on each link to control the position of multiple robotic routers to meet the clients' communication demands.

5.4 Capturing Real-World Communication in a Simple Quadratic Controller

In this section, we target the problem of placing a team of mobile router vehicles at locations such that they provide wireless coverage to client vehicles, each with different communication demands. Specifically, using as input the channel feedback $f_{ij}(\theta)$ derived in the previous section, we aim to find a function $\tilde{g}$ that can be optimized over router positions such that:

$$C_{t+1} = \arg\min_C \{\max_j \min_{c_i \in C} \tilde{g}(c_i, C_t, w_{ij}, f_{ij})\} \tag{5.11}$$

[4] In practice, the router and client transmit packets back-to-back with a small gap $\delta \approx 200\mu s$ to obtain $\hat{h}^r(t+\delta)$ and $\hat{h}(t)$, respectively. The router collects these values and approximates $\hat{h}(t)\hat{h}^r(t)$ as $\hat{h}(t)\hat{h}^r(t+\delta)e^{-j2\Delta_f\delta}$. The router computes this 10 times per second (an overhead of just 0.1%).

Our focus in this section is to find a function $\tilde{g}$ that has three desirable properties: 1) It is quadratic; 2) It allows for trade-offs between clients with competing demands as captured by the service discrepancies w_{ij}; and 3) It is independent of client positions p_j. In the rest of this section, we show how to capitalize the rich spatial information provided by $f_{ij}(\theta)$, to derive a cost $\tilde{g}$ possessing the three desired qualities. We can then optimize this cost to complete our objective of robot router placement that best satisfies the communication demands of the clients.

5.4.1 A Generalized Distance Metric for Incorporation of Channel Feedback

Our first goal is to translate signal quality over all directions, $f_{ij}(\theta)$, to a cost $\tilde{g}$ that can be optimized over router positions. We begin with the case where all positions are known and extend to the position independent case in Section 5.4.3. Consider a single router-client pair (i, j) located at positions (c_i, p_j). A disk model approach to service this client does not use $f_{ij}(\theta)$ at all. Instead, it relates improving communication quality between the router and client to reducing the Euclidean distance between them, i.e. the cost $\tilde{g} := dist(p_j, c_i)$. The appeal of such a cost is in its simple quadratic form that can be easily optimized. Unfortunately, the cost is oblivious to the actual wireless channel at the client and fails to capture the current service discrepancy which can be large even at small distances (say, due to obstacles).

Our system avoids this pitfall, while retaining simplicity, by incorporating real-time channel feedback into a generalized distance metric. In particular, we do not assume that the shortest distance for enabling better communication between two robots is the straight line path between them, but rather the path along the $\theta_{\max}$, the direction of maximum signal strength from the mapping $f_{ij}(\theta)$. Thus, the client is recommended to move towards $\vec{v}_{\theta_{\max}}$, the unit vector along $\theta_{\max}$.

Importantly, the recommended heading direction $\vec{v}_{\theta_{\max}}$ may exhibit variation due to noise or multipath affecting the wireless link. To account for these effects, while not over-fitting to noise, we leverage the entire f_{ij} signal profile to design a *confidence*

metric σ_{ij} in the recommended heading direction. The purpose of this confidence metric is to incorporate second-order information from f_{ij} that captures the presence of noise, or multipath, and can be used to alter the behavior of the controller accordingly. The derivation of this confidence metric is the subject of the following section.

A Confidence Measure from Channel Feedback

We design a parameter σ_{ij} that is derived from the mapping f_{ij} and that we refer to as a "confidence" in the recommended heading direction $\vec{v}_{\theta_{\max}}$. Intuitively, σ_{ij} captures the"variance" of f_{ij} around $\theta_{\max}$. We define σ_{ij} mathematically as the ratio of two quantities, σ_f and σ_N. All quantities in this subsection are with respect to a particular wireless link (i, j) but we subsequently drop i, j subscripts for readability. We define

$$F = \sum_{\theta} f(\theta) \tag{5.12}$$

$$\sigma_f = \sum_{\theta} (\theta - \theta_{\max})^2 \frac{f(\theta)}{F} \tag{5.13}$$

$$\sigma_N = \sum_{\theta} (\theta - \theta_{\max})^2 \frac{F}{L} \tag{5.14}$$

$$\sigma = \frac{\sigma_f}{\sigma_N} \tag{5.15}$$

where L is the total number of θ values that make up the plot $f_{ij}(\theta)$. The term σ_f is the variance of the plot f_{ij} around its maximum $\theta = \theta_{\max}$ and σ_N is a normalization factor (it is the variance around $\theta_{\max}$ in the case that the mass under the $f_{ij}(\theta)$ curve was distributed evenly over the θ values). The ratio of these two quantities, σ_f/σ_N, characterizes the amount signal strength (mass under the $f_{ij}(\theta)$ curve) that is concentrated under the peak direction $\theta_{\max}$ versus the remaining parts of the curve. A ratio of $\sigma_f/\sigma_N = 1$ would mean that the $f_{ij}(\theta)$ plot does not provide evidence that the max direction $\theta_{\max}$ is of much significance and that indeed the plot is entirely noise. On the other hand a ratio $\sigma_f/\sigma_N < 1$ indicates that a significant portion of the

signal strength curve in $f_{ij}(\theta)$ is concentrated around the max $\theta_{\max}$ and thus this peak is considered to have "high confidence." Lastly, the case where $\sigma_f/\sigma_N > 1$ indicates the presence of high signal strength in other parts of the $f_{ij}(\theta)$ curve other than the $\theta_{\max}$ direction which suggests the presence of multipath. Indeed we can characterize the range of σ_f and σ_N as

$$0 \leq \sigma_f = \sum_{\theta} (\theta - \theta_{\max})^2 f(\theta)/F \qquad \leq L^2 F \tag{5.16}$$

$$\frac{L(L+2)}{12} \leq \sigma_N = \sum_{\theta} (\theta - \theta_{\max})^2 F/L \qquad \leq \frac{(L+1)(2L+1)}{6} \tag{5.17}$$

By comparing the maximum variance of σ_f and σ_N corresponding to the case where there exists another peaked signal strength at a θ value far from $\theta_{\max}$, we see that $L^2 F > \frac{(L+1)(2L+1)}{6}$ for this case and thus $\sigma_f/\sigma_N > 1$. This case is a strong indication of the presence of multipath, where there is another high signal-strength path along which transmission on the link (i, j) may be received. We formally define these three cases below:

Definition 5.4.1 (Confidence in the direction of highest signal strength $\theta_{\max}$)**.** *We define three cases captured by our confidence metric* $\sigma = \frac{\sigma_f}{\sigma_N}$*:*

(i) ***High confidence peak:*** $\sigma < 1$

(ii) ***Noise:*** $\sigma = 1$

(iii) ***Multipath:*** $\sigma > 1$

See Figure 5-6 for examples of these regions identified automatically from actual experimental data.

Experimental results in the basement of the Stata Center building at MIT show that the regions of high confidence, noise, and multipath defined above can be identified automatically from data using the confidence metric from Eq. (5.15) (see Figure 5-7).

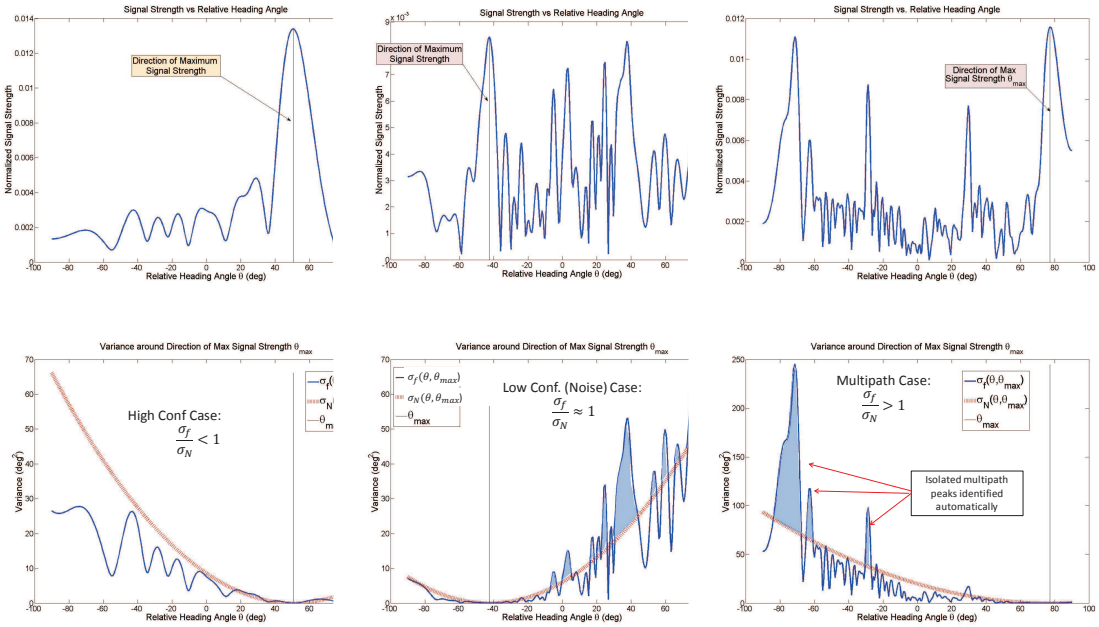


Figure 5-6: These plots show directional signal strength profiles from actual experiments. They demonstrate how the confidence metric identifies cases of high confidence, low confidence, and multipath automatically from the f_{ij} signal strength profile. The dotted red line is the variance,σ_N, of a uniform signal strength profile $f_{N_{ij}}(\theta) = 1/L$ centered around $\theta_{\max}$. Comparing the variance (bottom row) σ_f to σ_N indicates which of the three cases are occuring in the f_{ij} plot (top row): high confidence $\theta_{\max}$ ($\sigma_f < \sigma_N$), low confidence (noise) $\theta_{\max}$ ($\sigma_f \approx \sigma_N$), or multipath around $\theta_{\max}$ ($\sigma_f > \sigma_N$).

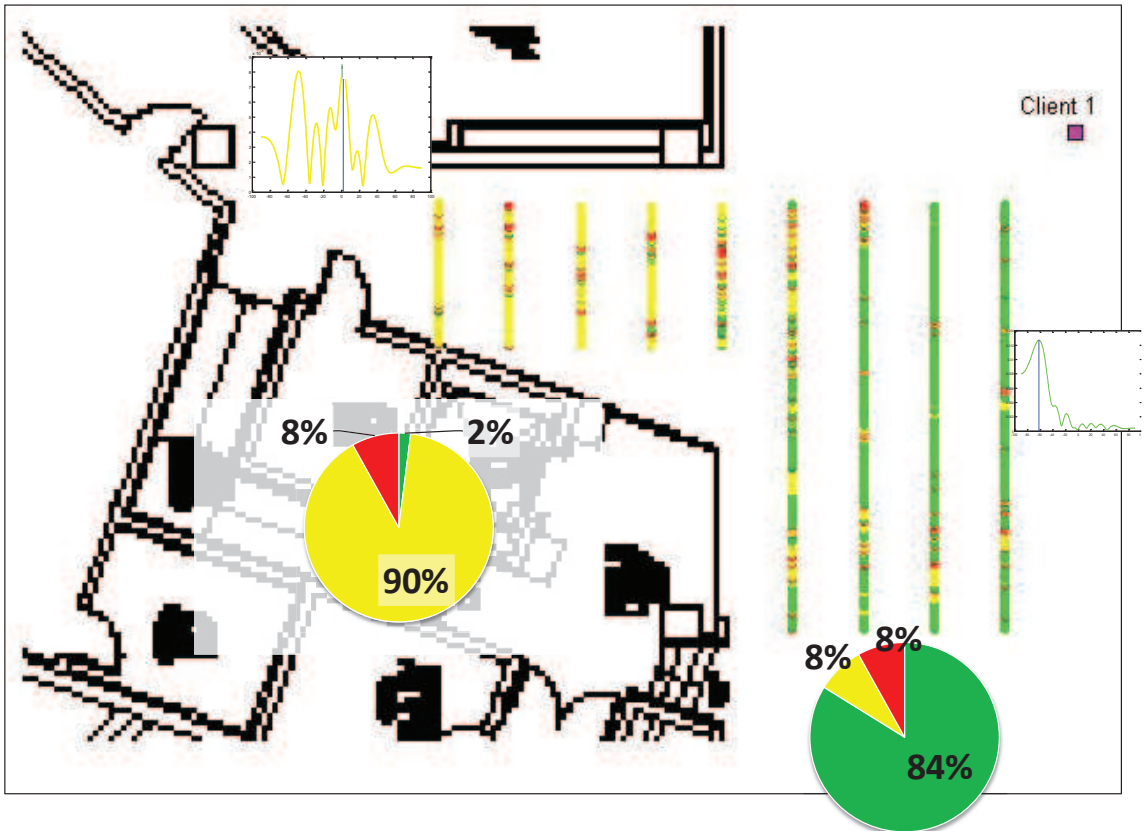


Figure 5-7: Data collected for a one-link system of one router and one client where the client is stationary at the top right corner of a basement environment and a mobile router is driven in a lawn mower pattern throughout the environment through line-of-sight and non-line-of-sight regions. Each colored data point represents an acquired directional signal profile (two example profiles are shown) and the color of the data point is the result of automatic mode detection from the data using the confidence metric from Eq. (5.15) where red=noise, yellow=multipath, and green=high confidence peak.

An important observation from the collected data in Figure 5-7 is that even in line-of-sight regions of the environment (relative to the position of the client) there may be significant multipath present due to reflections off of nearby concrete walls and this may cause the direction profile to demonstrate peaks in heading directions that are non-intuitive. Therefore this data suggests that simple geometric metrics, including visibility graphs, do not adequately capture the complexities of wireless signal quality in general environments.

Using $\vec{v}_{\theta_{\max}}$ and σ to Define a Generalized Distance Metric

We would like to encode the quantity σ_{ij} into our controller such that $\vec{v}_{\theta_{\max}}$ directions of high confidence are followed more aggressively (larger displacements along these directions), and the opposite is true of $\vec{v}_{\theta_{\max}}$ directions with low confidence.

Specifically, σ_{ij} falls under the following categories: 1) $\sigma_{ij} < 1$: Indicates a high confidence in $\vec{v}_{\theta_{\max}}$ due to a sharp peak in f_{ij}. The robot is moved at higher speeds; 2) $\sigma_{ij} \approx 1$: Indicates that f_{ij} is noisy, so the robot must move slowly; 3) $\sigma_{ij} > 1$: Indicates that f_{ij} has multiple significant peaks owing to multi-path. We study this case, and particularly the opportunity it presents for making trade-offs between clients, more elaborately in Sec. 5.4.2.

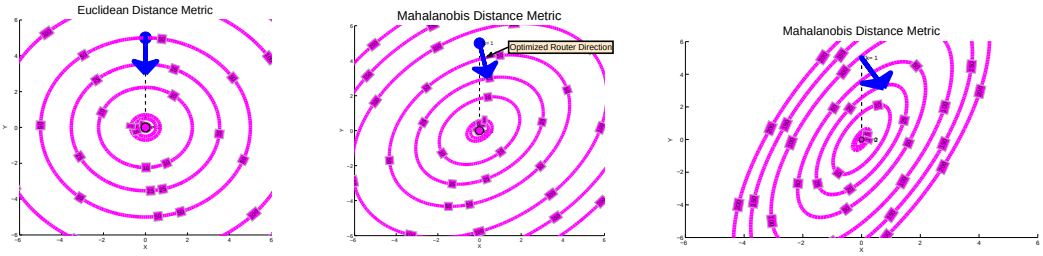
We can use the heading direction and confidence to design a cost function $\tilde{g}$ that accurately captures the cost of communication in the spatial domain. Interestingly, we can express this cost as a generalized distance metric called the *Mahalanobis distance*. The square of the Mahalanobis distance is a cost function (paraboloid) with ellipsoidal level sets (Fig. 5-8). We design our cost by orienting these level sets so that the direction of steepest descent is along $\vec{v}_{\theta_{\max}}$. We then skew the ellipsoidal level sets using the confidence σ_{ij}, so that a higher confidence translates to a steeper descent. Mathematically, the Mahalanobis distance is given by:

Definition 5.4.2 (Mahalanobis Distance)**.** *Given a positive definite matrix $M \in \mathbb{R}^{d\times d}$, a vector $x \in \mathbb{R}^d$, and a vector $y \in \mathbb{R}^d$, the Mahalanobis Distance between x and y is:*

$$\text{dist}_M(x, y) = \sqrt{(x-y)^T M (x-y)} \tag{5.18}$$

Euclidean distance is a special case of the Mahalanobis distance (see Fig. 5-8(a)) with $M = I$ where I is the identity matrix of appropriate dimension.

Here, $M = Q\Lambda Q^T$ is a positive-definite matrix, where Q consists of orthogonal eigen-vectors and Λ contains the corresponding eigen-values. We simply set one of the eigen-vectors of Q to the heading direction $\vec{v}_{\theta_{\max}}$. To skew the ellipsoid, we set



(a) Euclidean Distance (b) Mahalanobis (Low Conf) (c) Mahalanobis (High Conf)

Figure 5-8: These plots show the level sets of a Euclidean distance function and a Mahalanobis distance function.

the ratio of the eigen values $\{\lambda_1, \lambda_2\}$ in Λ to the confidence σ^2, i.e. $\lambda_2/\lambda_1 = \sigma^2$, where λ_1 is the eigen-value corresponding to $\vec{v}_{\theta_{\max}}$. For example, in Fig. 5-8(b), where $\sigma \approx 1$ (i.e. poor confidence), the level sets are nearly circular, leading to a shallow descent in cost; while Fig. 5-8(c), where $\sigma < 1$ (i.e high confidence), the level sets are skewed, leading to a steep descent in cost along $\vec{v}_{\theta_{\max}}$. In other words, the cost function has an elegant geometric interpretation, akin to Euclidean distance, but is derived directly from channel measurements. Further, the cost function $\tilde{g} := \text{dist}_{M_{ij}}(p_i, c_j)$ from Eqn. 5.18 is quadratic, a desirable property for optimizations.

5.4.2 Network Trade-offs

In this section, we show how our optimization framework readily extends to a multi-agent scenario and study the different trade-offs. We show that via the setting of two parameters, both set automatically from wireless channel data, the resulting positional controller can be made to greedily optimize one client's needs or alternatively, strike trade-offs between multiple clients. First, we focus on managing service discrepancies specified by w_{ij}. The quantity w_{ij} aims to bias the controller by assigning higher weight to users with larger service discrepancies. To do this, we scale the cost function $\tilde{g} = \text{dist}_{M_{ij}}(p_i, c_j)$ by the square of the discrepancy w_{ij}^2 to optimize:

$$r_M(P, C) = \max_{p_j \in P} \min_{c_i \in C} \{w_{ij}^2 \text{dist}_{M_{ij}}(p_i, c_j)\} \tag{5.19}$$

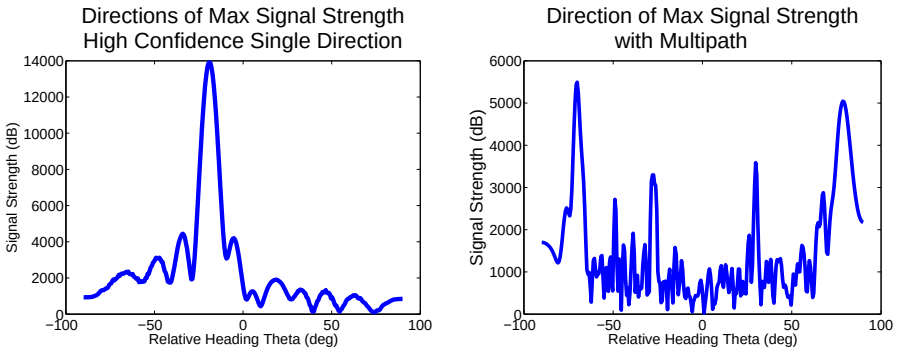
Second, we highlight the subtle role played by the confidence σ_{ij} in managing network trade-offs. For instance, consider a scenario with two clients: 1 and 2, where client-1 demands greater communication quality (as specified by w_{ij}'s). Suppose client-1 has a highly confident $\vec{v}_{\theta_{\max}}$ as shown in Fig. 5-9(a) (i.e $\sigma_{ij} < 1$). As expected, the robotic router is directed towards client-1 as shown in Fig. 5-9(c). In the more interesting scenario in Fig. 5-9(b), client-1's confidence is poor due to multiple peaks in the signal profile f_{ij} (i.e $\sigma_{ij} > 1$). Here, the router strikes a trade-off and services client-2 instead, as this may potentially benefit client-1 as well due to the multipath recognized in client-1's $f_{ij}(\theta)$ map. The intuition behind this is simple. Eqn. 5.21 above, scales the ellipsoidal cost function based on the discrepancies w_{ij}'s. However, recall that the ellipsoidal cost function is steep (or shallow) depending on whether the confidence is high (or low) and this is attained by setting the ratio of eigenvalues λ_2/λ_1 of M_{ij}. In extremely low confidence scenarios such as Fig. 5-9(b), the higher value of discrepancy of client-1 is masked by its low value of confidence. Hence, this balances the trade-off in favor of client-2, despite having a lower discrepancy.

5.4.3 A Position-Independent Solution

A simple relaxation to the cost from the previous section frees the optimization of using client positions, while maintaining its simple structure and desirable properties developed above. Consider a given stepsize $\gamma > 0$. We replace client positions p_j in Eqn. (5.21) with "virtual" positions p'_{ij}:

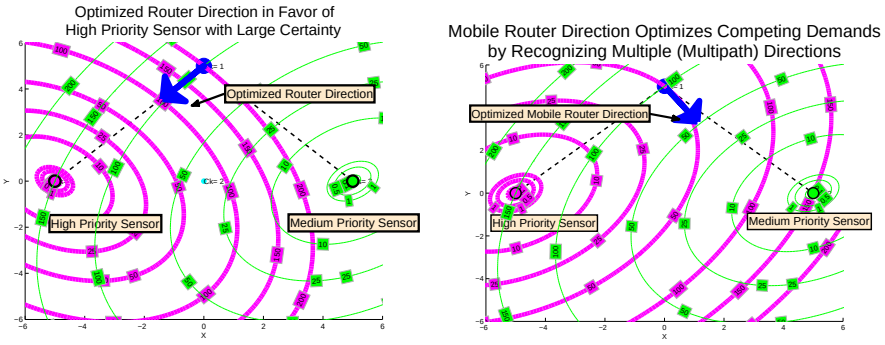
$$p'_{ij} = c_{i,t} + \gamma w_{ij} \vec{v}_{\theta_{\max}} \tag{5.20}$$

Loosely, a client is no longer directly observed but rather estimated to be along the relative direction $\vec{v}_{\theta_{\max}}$ and at a distance of γw_{ij} with respect to the ith router. As before, $\vec{v}_{\theta_{\max}}$ is the heading direction associated with the maximum stength signal direction $\theta_{\max}$. As a client's demand is better satisfied by router i, the service discrepancy w_{ij} tends to 0 and the client is perceived as being closer to router i. The intuition here is that routers better equipped to service a particular client as reflected by the w_{ij} term, will view the client as "closer" and those routers with a weaker sig-



(a) High Certainty Direction

(b) Multipath Directions



(c) Client Favored

(d) Client Tradeoff Multipath

Figure 5-9: Trade-offs between Clients: (a)-(b) show the $f_{ij}(\theta)$ map for the high demand client; (c)-(d) show the optimized router direction;

nal to the same client will view this client as farther away. This results in a natural method of assigning client nodes to routers by effectively sensing over the wireless channels. Our final cost takes the form:

$$r_M(C) = \max_{j \in \{1,\ldots,n\}} \min_{c'_i \in C'} \{\text{dist}_{M_{ij}}(c_{i,t} + \gamma w_{ij} \vec{v}_{\theta_{\max}}, c'_i)\} \tag{5.21}$$

By expanding the squared per-link cost $\text{dist}^2_{M_{ij}}(c_i + \gamma w_{ij} \vec{v}_{\theta_{\max}}, c'_i)$ from Eqn. 5.21:

$$(c'_i - c_{i,t})^T M_{ij} (c'_i - c_{i,t}) - 2\gamma w_{ij} \lambda_{\theta_{ij}} \vec{v}^T_{\theta_{\max}} (c'_i - c_{i,t}) + \gamma^2 w^2_{ij} \lambda_{\theta ij} \tag{5.22}$$

we note that as $w_{ij} \to 0$ the first term in Eqn. (5.22) favors stable solutions where $c'_i = c_{i,t}$, ie. the router reaches a static solution when all of its assigned clients have zero service discrepancy.

Finally for a set of routers with positions C, $r_M(C)$ reflects the cost of the client with the largest service discrepancy. As defined in our problem statement, Problem 5.2.1, we have found a set of quadratic costs $g(p_j, c_i, C_t, w_{ij}, f_{ij}) = (p'_{ij}(c_{i,t}, w_{ij}, \vec{v}_{\theta_{\max}}) - c_i)^T M_{ij} (p'_{ij}(c_{i,t}, w_{ij}, \vec{v}_{\theta_{\max}}) - c_i)$ that can be optimized in the desired min-max formulation from (5.4) in order to find an optimized robotic router placement for our wireless network.

5.5 A Reduction to the Router Placement Problem

We show that the most general problem formulation from Equation (5.22) can be reduced to the k-connected center problem formulation from Definition 3.3.5. This observation allows us to use the algorithm for router placement, Algorithm 1, for finding a configuration of routers C^* that optimizes (5.21).

Because the k-center problem is defined over any distance metric, showing that the cost from this section (Equation (5.21)) reduces to the k-connected center problem is equivalent to showing that the Mahalanobis distance $\text{dist}_M(x, y)$ satisfies the

properties of a metric.

Lemma 5.5.1 (The Mahalanobis Distance is a Metric [78])**.** *The Mahalanobis distance* $\text{dist}_M(x, y)$ *is a metric satisfying the properties of 1)non-negativity 2)symmetry and 3) the triangle inequality.*

Proof. Because the Mahalanobis distance is defined as $\text{dist}_M(x, y) = (x-y)^T M(x-y)$ where for our case M is always a strictly positive-definite symmetric matrix, we have that the properties of non-negativity $\text{dist}_M(x, y) \geq 0$ and $\text{dist}_M(x, y) = 0 \rightarrow x = y$ and symmetry hold. For the triangle inequality it must hold that $\text{dist}_M(x, y) + \text{dist}_M(y, z) \geq \text{dist}_M(x, z)$. By writing $\text{dist}_M(u, w) = \|M(u - w)\|$ and defining $\tilde{u} = Mu, \tilde{w} = Mw$ we immediately see that the triangle inequality holds for the Mahalanobis distance by an application of the Euclidean space triangle inequality to $\tilde{x}, \tilde{y}$ and $\tilde{z}$. □

Therefore we have shown that a configuration of routers C^* that minimizes the cost in Equation (5.21) can be found by replacing the cost (3.24) in Algorithm 1 with the cost (5.21) from this section which handles realistic wireless communication phenomena and heterogeneous communication quality demands.

5.6 Empirical Results

We evaluated our system on a five-node testbed with two routers and three clients. Each node was an ASUS 1015PX netbook equipped with an Intel 5300 Wi-Fi card mounted on an iRobot Create robot. We implemented SAR by modifying the iwlwifi driver on Ubuntu 10.04. We used the 802.11 CSI tool [60] to obtain channel information ($\hat{h}(t)$ in Eqn. 5.10). The routers communicated with a central laptop emulating the base for control information and human input. We performed our experiments in a room with a Vicon motion capture system to aid robot navigation. Our testbed contains obstacles to simulate both line-of-sight and non-line-of-sight scenarios.

In this section, all of the plots showing the effective signal to noise ratio (ESNR) have been averaged over small time windows in order to smooth out the small-scale

variation in signal strength that is characteristic of wireless signals.

5.6.1 Computing Direction of Maximum Signal Strength

We first observe how effectively our system computes the direction of maximum signal strength θ_{max}, on a wireless link. We consider a single client, serviced by a robot router, that is: 1) In direct line-of-sight (LOS) as shown in Fig. 5-10(a). 2) In possible non-line-of sight (NLOS) scenarios due to obstacles as shown in Fig. 5-10(b). We drive the robot router in a lawn-mover pattern and get θ_{max} at regular intervals.

Results: Fig. 5-10(a) and 5-10(b) depict the gradient field with the arrows indicating θ_{max} in LOS and NLOS, respectively. The gradient field in LOS accurately directs the robot router towards the client regardless of its initial position. In NLOS, the robot is directed away from obstacles so that controller can route around obstacles to improve signal strength. We stress that θ_{max} is found locally at the router purely via wireless channels and its own position, *without* prior knowledge of the environment. Further, the plots are not static and naturally *change over time*, especially in dynamic settings. Thus our system obtains instantaneous θ_{max} values locally in real-time.

Fig. 5-10(c) and 5-10(d) plot $f_{ij}(\theta)$, the power profile of the signal along different directions, for a candidate location in line-of-sight and non-line-of-sight scenarios, respectively. Clearly, the power profile in line-of-sight is dominated by a single peak at θ_{max}, directed along the line-of-sight path to the client. In contrast, the power profile in non-line-of-sight close to an obstacle has two significant peaks, each corresponding to reflected paths along walls or other objects in the environment.

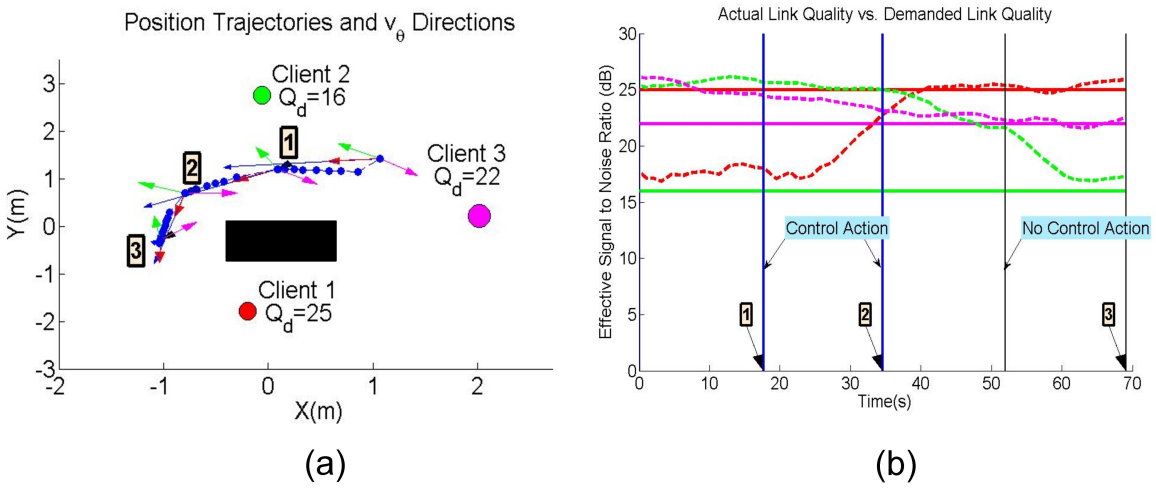


Figure 5-11: (a) Depicts testbed with robot router servicing three clients in a candidate non-line-of-sight setting. The blue line depicts the trajectory, and colored arrows indicate instantaneous θ_{max} for the corresponding clients. (b) Plots the ESNR across time (as dotted lines) for each client through the experiment. Solid lines denote client demands.

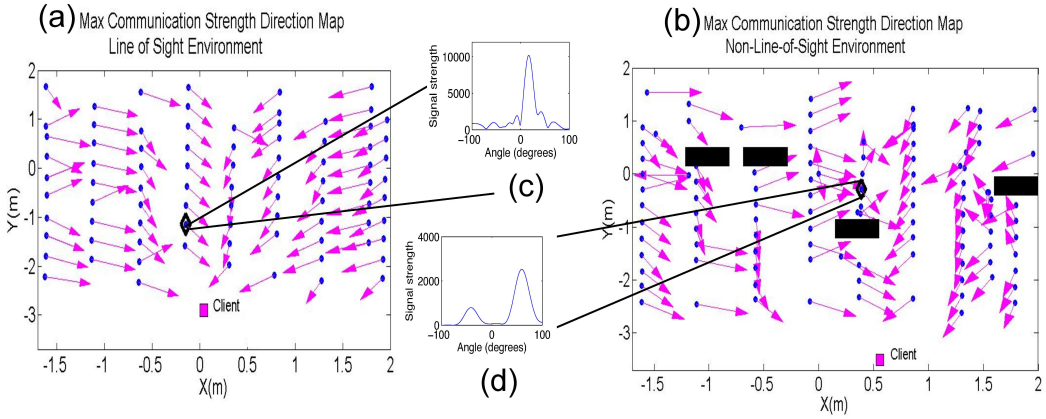


Figure 5-10: Gradient field of θ_{max} and power profile for (a) Line-of-sight and (b) Non-Line-of-Sight.

5.6.2 Controlling Router Trajectory to satisfy Client Demands

We evaluate how a single robotic router finds a trajectory to satisfy the demands of three clients (specified in terms of effective signal-to-noise ratio or ESNR) using θ_{max} on each link. We consider the candidate non-line-of-sight setting in Fig. 5-11(a). The router is unaware of exact client positions or the layout of the environment.

Results: Fig. 5-11(a) depicts the trajectory of the robotic router in blue. The colored arrows denote the recommended $\vec{v}_{\theta_{\max}}$ directions for each client at every control point. The figure shows how the robot performs non-zero control actions until it eventually satisfies network demands. Fig. 5-11(b) tracks the ESNR of the clients across time (dotted lines). The plot shows that the ESNR demands of each client (solid lines) are satisfied upon convergence. Note that the whenever the robot decides to follow the $\vec{v}_{\theta_{\max}}$ of a client at a control point (vertical line), the client's ESNR increases. This validates our claim that following a heading direction based on $\vec{v}_{\theta_{\max}}$ indeed improves the ESNR of the corresponding client.

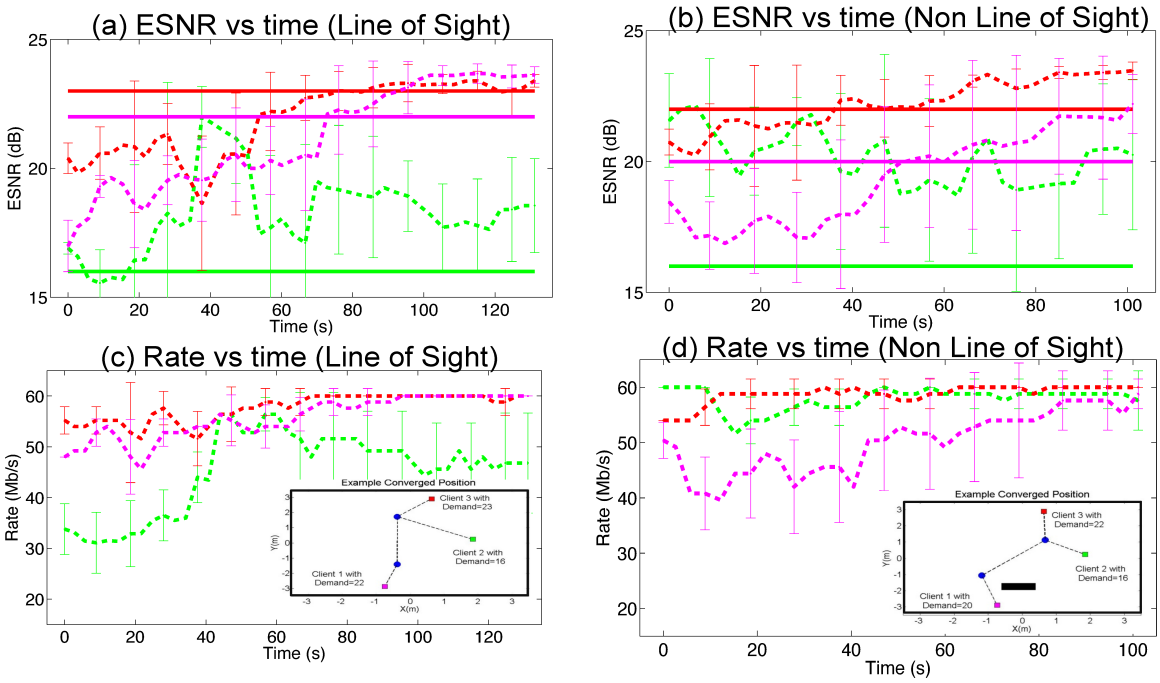


Figure 5-12: Statistical results obtained over 5 runs. Our plots show that demands are consistently met even in the presence of obstacles as demonstrated by the candidate converged solutions.

5.6.3 Aggregate System Results

We evaluate our full system with two robot routers serving three clients with different ESNR demands. We perform the experiment in line-of-sight (LOS) and non-line-of-sight (NLOS) settings as shown in the inset maps of Fig. 5-12(b) and 5-12(d)

respectively. We repeat the experiment five times in each setting and plot the results.

Results: Fig. 5-12(a) and 5-12(b) plot the mean and variance of ESNR over time across experiments for each client (dotted colored lines) in LOS and NLOS. Clearly, each client's ESNR demand (solid lines) is satisfied at the converged position across experiments. Fig. 5-12(c) and 5-12(d) plot the corresponding aggregate link rate across time, which follows the same trend as the ESNR [59].[5] The inset plots in Fig. 5-12(c) and 5-12(d) depict the final converged position of the routers (blue dots) in LOS and NLOS. The results show that our system consistently satisfies client demands while adapting to real-time changes in wireless channels, even in the presence of obstacles.

5.6.4 Comparison with Existing Schemes

We test our method against two other popular approaches to the communication problem in robotics: 1) Euclidean Disk Model as used in [23] and [65], where communication constraints are in terms of Euclidean distance; 2) Stochastic Gradient Approach, where we implement the Simultaneous Perturbation method (SPSA) ([100], [74]) for estimating the gradient of signal power by sampling the ESNR (which provides greater granularity than RSSI), along randomized directions, similar to the approach utilized by [74]. We note that although there are many sampling approaches in the literature, they share many commonalities (for example, the need for exploratory sampling of the RSSI) and many of the same drawbacks as a result. Thus we chose a sampling method that we believe is representative of many sampling techniques even though there are many to choose from.

For the generation of each direction in the SPSA method we use a Bernoulli random variable (as in [100]) and diminishing step sizes satisfying the conditions stated in [100] for convergence. Our largest step size was allowed to be the same maximum vehicle velocity of v_c for all experiments. We consider a robotic router and three clients, each with an ESNR demand of 20 dB. We repeat the experiment

[5]Note that the data-rate is capped by 60 Mb/s causing the plot to appear flat at times unlike ESNR.

five times in the non-line-of-sight environment in Fig. 5-13(b)-(d). In each instance, we measure r_{max}, the maximum ratio of ESNR demand versus the ESNR achieved among all three clients. In particular, r_{max} is below one at the converged position (i.e. all client demands are satisfied), and above one otherwise.

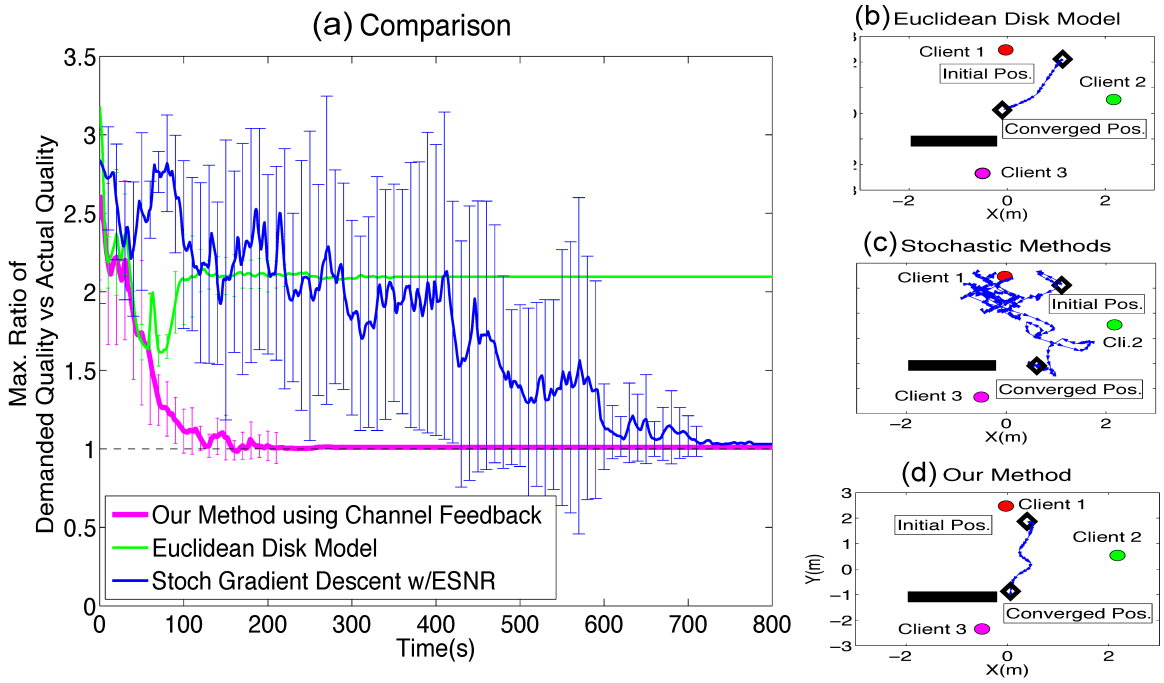


Figure 5-13: Plots comparing our method against the Euclidean disk model and a stochastic gradient descent method based on ESNR. Our method both converges to a position that meets communication demands, and converges quickly along an efficient path.

Results: Fig. 5-13(a) plots the aggregate mean and variance of r_{max} across time, for all the three approaches. Fig. 5-13(b)-(d) show a candidate trajectory adopted by the robotic router for the three schemes. The plots demonstrate while the disk model converges quickly to a solution, ignorance of the wireless channels leads to solutions not meeting client demands; especially in non-line-of-sight settings. In contrast, the stochastic gradient approach (in blue), which sample the instantaneous ESNR, eventually satisfies network demands. However, the convergence is often laborious as the router often traverses counter-productive directions (see Fig. 5-13 (c)). Indeed such techniques are noisy at low signal power, as even a large change in distance translates to a small change in signal power (a well-studied problem in communications litera-

ture [18, 67, 115]). Fig. 5-13(c) shows that this leads to areas at non-line-of-sight or far distances from the client, where the robot easily gets lost.

Our method leverages full information of the channel, including signal power and phase, to find the direction of signal power as opposed to its magnitude. The result is an algorithm that converges to positions that satisfy network demands while not necessitating counter-productive exploration steps of a pure sampling approach.

5.6.5 Robustness to Dynamic Obstacle Positions

We evaluate how our system adapts to changes in the environment without an *a priori* known map. Consider two robotic routers and three clients in an environment with an obstacle located initially as shown in Fig. 5-14(a). We allow the robot routers to navigate to their converged positions. At $t = 120$ sec, we move the obstacle to a different location as in Fig. 5-14(c), and let the routers re-converge.

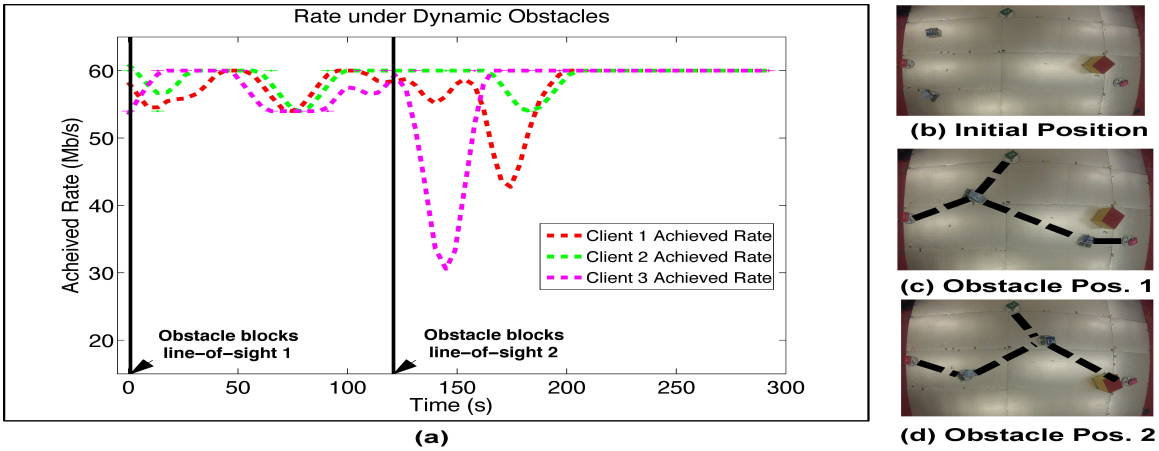


Figure 5-14: These plots show the result of disturbing the wireless channels via movement of a line-of-sight obstructing obstacle. Actual testbed snapshots are shown on the right.

Results: Fig. 5-14(b) and Fig. 5-14(c) depict the converged position of the routers before and after the obstacle was moved. Fig. 5-14(d) plots the data-rate across time for each client. The plot shows that our system satisfies client demands at the initial position. Further it recovers from the sharp fall in data-rate to one of the clients to successfully re-converge after the obstacle is moved.

5.6.6 Complex Indoor Environments

We evaluate our system in a large complex indoor environment with concrete walls and columns. We place a robotic router and client and line-of-sight (LOS) and non-line-of-sight (NLOS) as in Fig. 5-15. In this setup, the client is stationary at the upper right-hand corner of the environment and a mobile router is driven in a lawn-mower pattern to attain directional profile data and acquire a gradient field similar to those in Figures5-10(a) and 5-10(b). Although our signal profiles provide information on signal quality over all heading directions, we take the highest power direction to be the gradient and thus the gradient field in our plots shows the direction of the greatest amplitude peak (compare with Figure 5-10).

We note that mapping the entire environment is not necessary for our controller since we make local decisions using the most recent directional signal profile and purposefully avoid using stale information (acquired at earlier times). This is because dynamic environments can obviate older measurements. However, it is possible to map the wireless signals in a static environment using our techniques and this is what is demonstrated by Figure 5-15. We trace the router's gradient field towards the client starting from multiple initial positions.

In this experiment no vicon (motion capture) positioning system was used and no knowledge of the client position, nor the router position were needed to acquire the data. Instead, the assumption that the router was moving at a constant velocity is sufficient for producing the directional signal maps. This suggests that exact knowledge of router positions may not be necessary if the velocity of the platform can be assumed constant during the data acquisition phase.

Similar works such as [108] use samples of the received signal strength (RSS) to estimate the direction of the gradient of signal quality in the environment and for identifying connectivity boundaries and source localization. Application of our current technique that has access to the full directional profile of signal strengths (not just the direction of greatest signal improvement, or gradient) to this class of

problems could be an interesting topic of future work.

Results: Fig. 5-15 (a) and (b) plot of candidate trajectories (from gradient field) in LOS and NLOS across initial locations. The plots show that our system successfully navigates towards the client to satisfy its demands, without knowledge of the environment or client location.

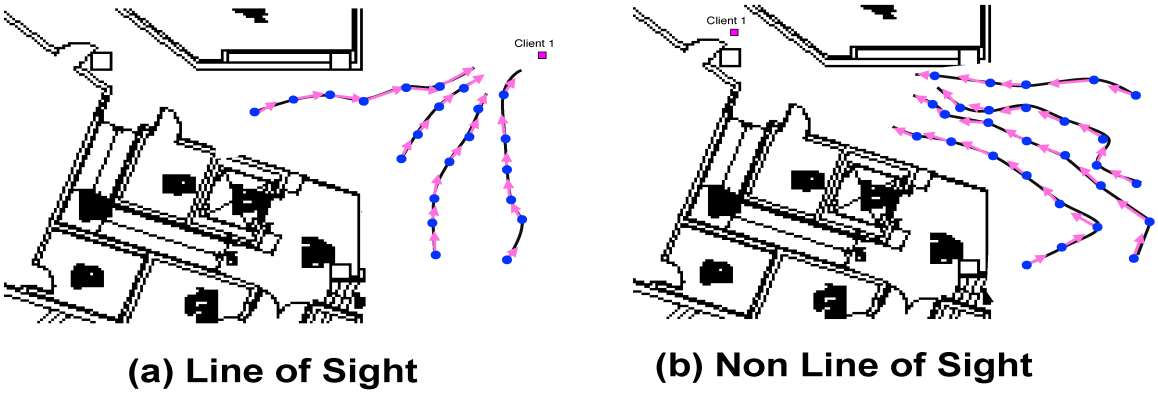


Figure 5-15: Trajectories using measured $\vec{v}_{\theta_{\max}}$ directions satisfy a client's demand in line-of-sight and non-line-of-sight settings in complex indoor environments.

5.6.7 Discussion on SAR in Higher Dimensions

We briefly note that while the discussion of the theory and implementation in this chapter focuses on two-dimensional ($\mathbb{R}^2$) applications, the concepts discussed should be extensible to higher dimensions though this is a topic of future work. It should be mentioned that SAR in its traditional radar form has enjoyed extensive application in terrain mapping as in [54] where signals from a transmitter mounted on an airplane are bounced off of the earth's surface as the plane traverses over straight-line paths. These reflected signals can be post-processed to recover a map of the terrain. In order to resolve the additional uncertainty and ambiguity of the depression angle necessary in this higher dimensional application of SAR, one technique that is described in [54], is to an additional aperture separated by a vertical distance. This may provide a fruitful avenue for extending the concepts of the current chapter to higher dimensions as well.

5.6.8 Conclusion

In this chapter, we present a framework to satisfy real-time variable communication demands in a constantly changing network. We develop a solution enabling a robotic receiver to find the profile of signal strength across spatial directions for each sender of interest. While our design focuses on optimizing communication quality, it can be readily integrated as a set of simple quadratic constraints to build upon prior work on multi-agent coordination such as coverage [23], consensus [89], formation control [65], etc. We believe our system provides the necessary robustness to bring the benefits of these seminal contributions to practical robotic systems.

Chapter 6

A Distributed Router Placement Formulation for Static Clients

6.1 Introduction

In this chapter we propose a nonsmooth, gradient-based approach to positioning routers to optimize a communication network. We achieve this objective via careful design of an appropriate cost function that is then minimized by the placement of the routers. As in the rest of the thesis, we refer to routers as the subset of the network graph that we control, and "nodes" is a general term that refers to both routers and clients[1].

A common approach to distributed minimization of a cost function is to design a gradient-based controller where agents follow a distributed gradient descent on that cost function. We design a cost function that incorporates the Signal-To-Interference Ratio (SIR) from the communication literature, which is a physically-based, continuous measure of link quality between any two communicating agents [58]. Local minima of this cost function achieve a tradeoff between maximizing the SIR for any single link, and equalizing the communication capability, also SIR, over all links in the graph. We model signal strength between two nodes that degrades with distance and drops non-smoothly to zero outside of the communication radius R. The ad-

[1]This work has been presented in [49]

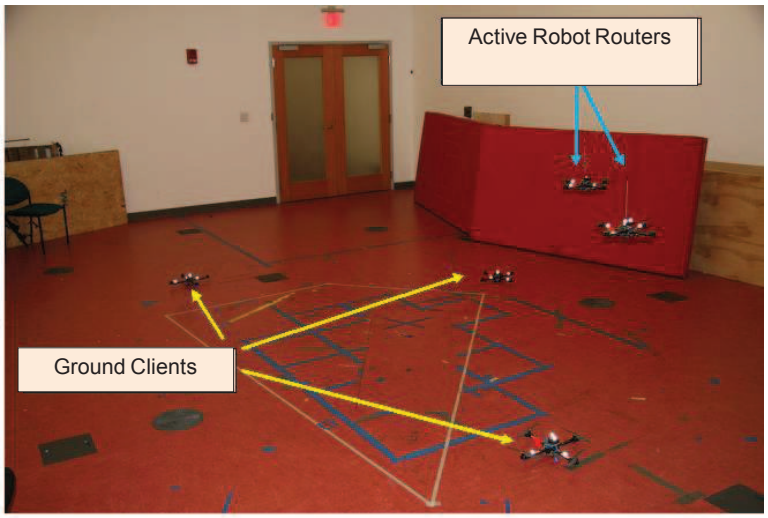
vantage of using this approach over existing approaches is two-fold: 1) the handling of a non-smooth cost function makes this method robust to failures and/or agents moving in and out of communication range where in general such changes in topology are difficult to handle in distributed systems and 2) the cost explicitly accounts for interference from neighboring transmitters which is necessary for a real ad-hoc network.

The non-differentiability due to nodes entering or leaving the communication radius of one another necessitates the use of results from the nonsmooth stability analysis literature [24] to prove convergence to local minima of the cost function. Furthermore, for certain initial conditions and controller parameter values, we prove that routers will never move in such a way so as to disconnect the communication graph.

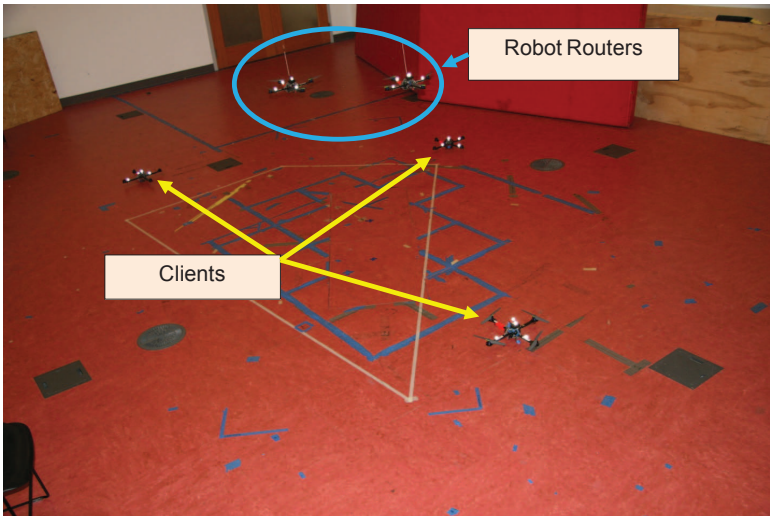
6.1.1 Notation

Table 6.1: Common Notation for Chapter 6

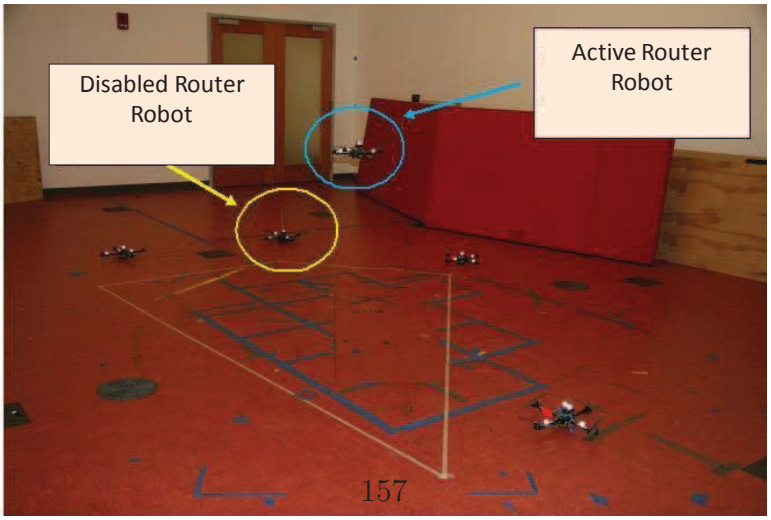
$G(V, E)$	$\triangleq$	Communication graph with nodes V and edges E.
ε	$\triangleq$	A given error tolerance where $\varepsilon > 0$.
p_j	$\triangleq$	Client j position in $\mathbb{R}^d$.
c_i	$\triangleq$	Robot router i position in $\mathbb{R}^d$.
SIR_{ij}	$\triangleq$	Signal-to-Interference Ratio for link (i, j).
f_{ij}	$\triangleq$	Signal quality of link (i, j).
H	$\triangleq$	Cost based on idealized physical model and SIR.
∂	$\triangleq$	Generalized gradient operator.
$Ln(\partial H)(x)$	$\triangleq$	Generalized gradient vector field.


Active Robot Routers
Ground Clients

(a)


Robot Routers
Clients

(b)


Disabled Router Robot
Active Router Robot

(c)

6.2 Decentralized (local) Cost Development based on Physical Model

6.2.1 Problem

The communication coverage problem for static clients considers n clients with positions $p_j \in \mathbb{R}^d, j \in [m] = \{1, \ldots, m\}$, that are performing a collaborative task (such as coverage, search, or exploration of an environment). These vehicles are required to communicate over distances greater than their communication radius R in order to achieve their assigned task. We propose the use of a group of k routers to provide a communication network for the ground vehicles, where the aerial robots follow a distributed control law and are placed at locations that optimize communication link quality amongst all vehicles according to a specific cost H that we derive. The cost function associated with this approach is described in Equation (6.4). This cost function formally defines our cost $H : \mathbb{R}^{(p \times N)} \to \mathbb{R}$ over all vehicle positions where $N = k+n$ is the total number of vehicles. We denote communication vehicle positions as $c_i \in \mathbb{R}^d$ where $i \in [k] = \{1, \ldots, k\}$.

Assumptions

We assume that 1) k is large enough to provide a connected network amongst ground vehicles, 2) that communication only exists amongst neighbors within a distance radius R where signal strength is modelled by f_{ij} described later in this section, and that 3) the ground vehicle dynamics are zero as necessary for the mathematical proof, although in the practical setting we may allow ground vehicles to move given that their velocities are much smaller than those of the routers. We note that assumption 3 is common for problems using Lyapunov-type proofs of stability. Due to the distributed nature of our problem, all agents have access only to local information and thus will be unaware of disconnected subclusters. Therefore we must also assume that the communication network composed of both air and ground vehicles is initially in a connected state, although our controller is robust to changes in the network

including agents arriving or exiting. Our hardware results demonstrated in Figure 6-4 include such a scenario, where an router is disabled and the remaining router positions themselves to compensate for the loss of the router.

Objective

We aim to ensure connectivity of the graph in a continuous fashion by either placing a requirement that the initial conditions of the system are below some critical cost, or by adjusting a design parameter ρ in our cost function to ensure that routers will never break existing connections.

Routers are controlled via a gradient descent method, where we allow for a non-smooth cost function that is non-differentiable at the points where nodes come in and out of communication radius of each other. Due to the local non-differentiability of the cost function, we must instead use the generalized gradient of the cost function which we denote $\frac{\partial H}{\partial c_i}$ throughout. We find the direction of descent for the resulting nonsmooth gradient vector field such that the controller takes the form

$$\dot{c_i} = -Ln(\partial H)(c_i). \tag{6.1}$$

Where $Ln(\partial H)(x) : \mathbb{R}^d \to \mathbb{R}^d$ is the generalized gradient vector field, and $-Ln(\partial H)(c_i)$ is a direction of descent of H at $c_i \in \mathbb{R}^d$ [24]. In Section 4.3.4 we find the generalized gradient vector field of the cost function and show that the resulting positions of the routers converge to critical points of this cost function.

We design our cost function to incorporate a physically-based, continuous, measure of signal quality called the Signal-to-Intereference Ratio (SIR) [58]. The SIR value of the link i-j improves with increasing communication strength between agents i and j and decreases with increasing environmental noise N_i and interfering communication amongst i's other neighbors as seen from the definition of SIR:

$$SIR_{ij}(c_i, p_j) = \frac{f_{ij}(c_i, p_j)}{N_i + \sum_{l \in \mathcal{N}_{i \setminus j}} f_{il}(c_i, p_l)} \tag{6.2}$$

Where $\mathcal{N}_{i \setminus j}$ is the set of neighbors of i not including j. The communication strength over link i-j is denoted f_{ij}. We choose an example model for the signal strength that drops off proportional to ${d_{ij}}^{-\beta}$, but we emphasize that other, more problem specific models for signal strength can be used with our controller so long as this function is locally Lipschitz and regular and models no communication outside of the radius R. These properties are important for the analysis of our controller but we defer this discussion to section 4.3.4. We define f_{ij} as

$$f_{ij}(c_i, p_j) = \begin{cases} \frac{P_0}{d_{ij}^{\beta}+1} - C & , \quad d_{ij} \leq R \\ 0 & , \quad d_{ij} \geq R \end{cases} \tag{6.3}$$

where $C = \frac{P_0}{R^{\beta}}$ is a constant to ensure continuity at $d_{ij} = R$, and we define $d_{ij} = \|c_i - p_j\|$. Thus the communication strength model reaches a maximum value of $\frac{P_0}{d_{ij}^{\beta}+1} - C$ at $d_{ij} = 0$ and drops off by β as $d_{ij} > 0$ with a non-smooth transition to zero at $d_{ij} = R$ as seen in Figure 6-2. This non-smooth transition is necessary to model loss of communication between two agents at a distance larger than R from each other. Finally, we present our cost function $H : \mathbb{R}^{(d \times N)} \to \mathbb{R}$ over all vehicle positions ${c^t}_i \in \mathbb{R}^d, {p^t}_i \in \mathbb{R}^d$ for $N = n + k$ vehicles at iteration t as:

$$H({c^t}_1, \ldots, {c^t}_k, {p^t}_1, \ldots, {p^t}_n) = \sum_{i=1}^{k} \sum_{j=1}^{n} -SIR_{ij} + \frac{\rho}{SIR_{ij} + \delta}, \tag{6.4}$$

where the term $\delta \in (0, 1]$ is included to ensure that the cost function H is continuous at the point where agents become disconnected and the value of $SIR_{ij} = 0$. For clarity we will subsequently drop the explicit dependence of H on agent positions as this dependence is captured through SIR_{ij}, and we will not carry around the

iteration number t as this is implicit throughout. A smaller δ value has the effect of putting more weight on the second term of the cost function. It is evident that the cost function is global and thus uses position information for all agents. However, as shown in equation (6.8) the control for each agent is local, as all non-neighbor information drops out in the derivative. Figure 6-6 shows optimization of a non-smooth H as agents enter the communication neighborhoods of others.

Minimization of this cost function corresponds to a compromise of two competing goals. The first term in the cost function favors increased SIR over all communication links in the graph while the second term favors equal SIR over each individual link, which can be thought of as equal resource allocation where SIR measures communication ability of each link. The design parameter $\rho \geq 0$ is used to adjust the weighting of the first term versus the second term in the cost function. A higher weighting on the second term corresponds to agents seeking to equalize their SIR values amongst all of their neighbors whereas a higher weighting on the first term will result in agents greedily improving individual SIR links. In Section 4.3.4 we prove that there exists a critical value of ρ, ρ_{cr}, that prevents agents from disconnecting from existing neighbors and demonstrate this range of behaviors for the controller in Figure 6-3.

Because the cost function H is non-smooth due to the non-differentiability of f_{ij} at $d_{ij} = R$, we next characterize its a stability even at non-smooth points of the cost function.

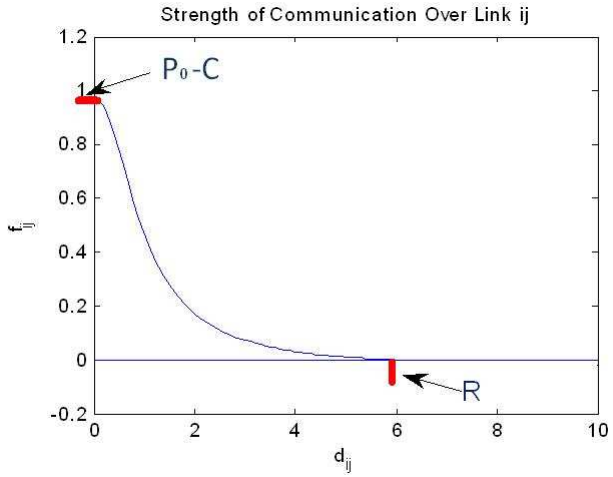


Figure 6-2: Plot of f_{ij}.

6.3 Non-Smooth Analysis of Controller

The cost function H presented above is non-smooth at the point where agents move in and out of the communication radius R of each other. This is reflected as a non-smooth transition to zero in the function f_{ij} at the point $d_{ij} = R$. As a result, the derivative does not exist at this point and we must instead use the generalized gradient and generalized gradient vector field of our cost function in order to build the appropriate controller.

6.3.1 Generalized Gradient and the Generalized Gradient Vector Field

The generalized gradient $\frac{\partial H}{\partial c_i}$ of a function f at a point of non-differentiability, x, is presented in [24], as the convex hull of the all the possible limits of the gradient at neighboring points where the gradient is defined. More precisely:

$$\frac{\partial H}{\partial x} = co\left\{lim_{z_i \to x} \nabla H(z_i) \ \forall z_i : z_i \to x, z_i \notin \Omega_H\right\}, \tag{6.5}$$

where *co* denotes convex hull, $H : \mathbb{R}^d \to \mathbb{R}$, is a locally Lipschitz function, and $\Omega_H \subset \mathbb{R}^d$ denotes the set of points where H fails to be differentiable. Moreover, the *generalized gradient vector field*, $Ln(\frac{\partial H}{\partial x}) : \mathbb{R}^d \to \mathbb{R}^d$, is defined in [24] where $Ln : \mathcal{B}(\mathbb{R}^d) \to \mathcal{B}(\mathbb{R}^d)$ *is a set-valued map that associates to each subset* $\mathcal{S}$ *of* $\mathbb{R}^d$ *the set of least-norm elements of its closure* $\bar{\mathcal{S}}$. Most importantly, $-Ln(\frac{\partial H}{\partial x})$ is a direction of descent of H at $x \in \mathbb{R}^d$ [24]. Finding the generalized gradient for an arbitrary nonsmooth function can be a daunting task, however for our case, because the function f_{ij} is smooth everywhere except at R, the generalized gradient is equivalent to the normal gradient at all points outside of R, where at R it takes the value zero. The generalized gradient vector field of f_{ij} for our problem is:

$$Ln[\frac{\partial f_{ij}}{\partial c_i}] = \begin{cases} \{-\frac{\beta P(c_i - p_j)\|c_i - p_j\|^{\beta-2}}{(\|c_i - p_j\|^{\beta}+1)^2}\} & , \quad d_{ij} < R \\ \{0\} & , \quad d_{ij} \geq R \end{cases} \tag{6.6}$$

Knowing the generalized vector field for f_{ij} is sufficient for finding the generalized vector field of the cost function H. This relies on the fact that f_{ij} is Lipschitz and regular. A function is said to be locally Lipschitz at $x \in \mathbb{R}^d$ if there exist a L_x and $\epsilon \in (0, \infty)$ such that $\|f(y) - f(y')\| \leq L_x \|y - y'\|$ for all $y, y' \in B(x, \epsilon)$ where $B(x, \epsilon)$ is a ball centered at x of radius ϵ. A function is said to be regular when its right directional derivative $f'(x; v)$ is equal to its generalized directional derivative $f^0(x; v)$, [24], where:

$$f^0(x; v) = \lim_{h \to 0^+} \sup_{y \to x} \frac{f(y + hv) - f(y)}{h} \tag{6.7}$$

The proof of f_{ij} Lipschitz and regular, as well as the final form of the controller using the generalized vector field of H is presented in the next subsection.

6.3.2 Stability of Controller

Our next objective is to prove the stability of the controller. This requires several supporting results from the nonsmooth analysis literature that we present in the following background section.

Background

We summarize some supporting results from [24]. The first results are the *Sum Rule* and *Quotient Rule* for algebraic operations on nonsmooth functions summarized in [24]. These results are important for conserving Lipschitz and regular properties of nonsmooth functions and for finding the generalized gradient of a function that is an algebraic composition of such functions.

Sum Rule: If f_1,f_2:$\mathbb{R}^d \to \mathbb{R}$ are locally Lipschitz and regular at $x \in \mathbb{R}^d$ and $p_1, p_2 \in \mathbb{R}$, then the function $p_1 f_1 + p_2 f_2$ is locally Lipschitz and regular at x and the generalized gradient $\partial(p_1 f_1 + p_2 f_2)(x) = p_1 \partial f_1 + p_2 \partial f_2$.

Quotient Rule: If f_1,f_2: $\mathbb{R}^d \to \mathbb{R}$ are locally Lipschitz and regular at $x \in \mathbb{R}^d$ and $p_1, p_2 \in \mathbb{R}$, then the function f_1/f_2 is locally Lipschitz and regular at x and the generalized gradient $\partial(f_1/f_2)(x) = (1/f_2^2(x))(f_2 \partial f_1 - f_1 \partial f_2)$.

We combine the results *Theorem 1* and *Theorem 2* of Jorge Cortes' *Discontinuous Dynamical Systems* to produce a result similar to Proposition 11 of the same work. We state this result here as Lemma 1.

Lemma 1. *Let $H : \mathbb{R}^d \to \mathbb{R}$ be locally Lipschitz and regular. Then, the strict minimizers of H are strongly stable equilibria of the nonsmooth gradient flow of H. Furthermore, if there exists a compact and strongly invariant set for the nonsmooth dynamics in (6.1), then the solutions of the nonsmooth gradient flow asymptotically converge to the set of critical points of H [24].*

Stability and Convergence

We are now ready to state and prove our theorem for stability and convergence properties of our controller in (6.1).

Theorem 1. *Routers following the direction of descent of the generalized gradient of H such that $\dot{c}_i(t) = -Ln(\frac{\partial H}{\partial c_i})$ will asymptotically converge to the critical points of H where the strongly stable critical points are local minima of H.*

Proof. The proof of this theorem follows readily from Lemma 1, using the fact that H is locally Lipschitz and regular, and that there exists a compact and strongly invariant

set for (6.1). The maximum of a finite set of continuously differentiable functions is a locally Lipschitz and regular function [24]. Thus the function f_{ij} is regular because it can be written as $f_{ij} = \max\{\frac{P_0}{d_{ij}^{\beta}+1} - C, 0\}$ where both $f(d_{ij}) = \frac{P_0}{d_{ij}^{\beta}+1} - C$ and $f(d_{ij}) = 0$ are continuously differentiable functions and thus f_{ij} is a locally Lipschitz and regular function. Combining equations (6.3) and (6.2), it is clear that H,from (6.4), is an algebraic composition of signal strength functions. Since the signal-strength function f_{ij} is Lipschitz and regular, by applying the *Sum Rule* and *Quotient Rule* it follows that H is both Lipschitz and Regular. Lastly, we show that there exists a compact and strongly invariant set for the dynamical system in (6.1). The generalized gradient $\frac{\partial H}{\partial c_i}$ for agent i goes to zero when agent i is outside of the communication radius R for all other $N-1$ agents and thus we define the set, $\mathcal{M}$, to be the set of points for which the generalized gradient is non-zero. Let $\mathcal{M} \subseteq \mathbb{R}^d$ be the set of all points inside the radius $2R(N-1)$ from the origin where, for the case of one ground robot g, we place g at the origin. By definition this set is both closed and bounded in a ball $B(0, 2R(N-1))$ and is thus compact. This generalizes readily to the case of more than one ground robot if we find the union of all such sets. Furthermore, a solution to (6.1) with any initial condition $x_0 \in \mathcal{M}$ remains in $\mathcal{M}$ because $\frac{\partial H}{\partial c_i}(p) = 0 \; \forall p \notin \mathcal{M}$ and so $\mathcal{M}$ is a strongly invariant set.

□

Using the Product Rule and the Sum Rule, and the fact that f_{ij} is Lipschitz and regular, we now present the final form of our controller from (6.1).

$$\begin{aligned} \dot{c_i} &= -Ln[\frac{\partial H}{\partial c_i}] \\ &= \sum_{i=1}^{N}\sum_{j=1}^{N} -\frac{\partial SIR_{ij}}{\partial c_i}(1+\rho(SIR_{ij}+\delta)^{-2}). \end{aligned} \tag{6.8}$$

Where $\frac{\partial f_{ij}}{\partial c_i}$ was defined above in (6.6) and $\frac{\partial SIR_{ij}}{\partial c_i}$ is

$$\frac{\partial SIR_{ij}}{\partial c_i} = \frac{\frac{\partial f_{ij}}{\partial c_i}}{N_i + \sum_{l \in \mathcal{N}_i \setminus j} f_{il}} - f_{ij} \frac{\frac{\partial N_i}{\partial c_i} + \sum_{l \in \mathcal{N}_i \setminus j} \frac{\partial f_{il}}{\partial c_i}}{(N_i + \sum_{l \in \mathcal{N}_i \setminus j} f_{il})^{-2}} \tag{6.9}$$

6.3.3 Maintaining Network Connectivity

We use the fact that the routers are following a gradient descent on the cost function H to identify initial conditions that prevent agents from moving to disconnect the communication graph. Because of the distributed nature of our controller, we do not employ any global checks on graph connectivity and thus require that the communication graph is initially connected. We present two approaches to maintaining graph connectivity. The first approach identifies the minimum cost of a disconnected network and requires that the initial conditions of any network are below this value. The second approach is to find a critical value of ρ in (6.4) such that routers will never move outside of a radius R from their neighbors and thus will remain connected. The main difference between these two approaches is that the first approach is a check on initial conditions to ensure that connectivity is maintained, while the second approach is a design perspective where a value of the parameter ρ is chosen as a function of other parameters in (6.4) to prevent disconnection.

Theorem 2. *Given that the network begins in a connected state, the routers will not move in such a way to disconnect the graph under either of the two following conditions:*

(i) The initial cost of the system H begins below the minimum cost of a disconnected graph $H_{d_{min}}$.

(ii) The design parameter, ρ, in (6.4) takes a value $\rho \geq \rho_{crit}$ where ρ_{crit} is the value at which the dot product $\frac{\partial H}{\partial c_i}^T (c_i - p_j) = 0$ for the pair i-j where $d_{ij}^* = \max \|c_i - p_j\|$ s.t. $d_{ij}^* < R$.

Proof. We identify the minimum cost of a disconnected graph that we call $H_{d_{\min}}$. Because our controller requires that agents will move to decrease the cost, H, if the initial cost of the system $H_0 < H_{d_{\min}}$ then the network will remain connected. For the second part of the theorem we identify a value of the parameter ρ such that an agent will never disconnect from its neighbors in the worst-case scenario. Namely, we ensure that the dot product $\frac{\partial H}{\partial c_i}^T(c_i - p_j) = 0$ in the limit as $d_{ij} \to R$ so that agent i's velocity component in the direction away from j is zero and thus will never disconnect an existing connection. This is depicted graphically in Figure 6-3.

Minimum cost of a Disconnected Network

The cost of disconnecting an edge in the communication graph, or equally, the cost of a missing connection in the communication graph is given by:

$$H_{ij}|_{d_{ij}=R} = \frac{\rho}{\delta} \tag{6.10}$$

To find the minimum cost of a disconnected graph, we find the minimum number of missing connections for a disconnected graph. If we look at the case of two disconnected subgraphs, the number of elements in each subgraph is p and $N - p$ respectively, where N is the total number of elements. The function $c(p) = p(N - p)$ denotes the number of missing connections between the two subgraphs (we assume subgraphs are fully connected). Therefore the minimum number of edge disconnections occurs when a single node is separated from the larger subgraph. Therefore we find that the minimum number of directed edge disconnections for a disconnected graph is $2(N - 1)$ and the cost for this graph is:

$$H_d = 2(N-1)\frac{\rho}{\delta} + \sum_{u \neq p}\sum_{w \neq p} -SIR_{uw} + \rho(SIR_{uw} + \delta)^{-1} \tag{6.11}$$

Furthermore, we are interested in the *minimum* cost of such a graph. The theoretical minimum of Equation (6.4) would be acheived when the SIR value for all the agents

in the second subgraph is maximal. The maximum theoretical value of the SIR_{ij} from Equation (6.2) is acheived when the distance of the two agents i and j goes to zero and when interfering communication from i's neighbors, or environmental noise N_i is not accounted for. This maximum is the same maximum as that of f_{ij} and is $\max\{SIR_{ij}\} = P_0 - C$. Plugging this into the cost function we find the minimum possible H for a disconnected graph:

$$H_{d_{\min}} = 2(N-1)\frac{\rho}{\delta} - (N-1)(N-2)\left((P_0 - C) - \rho((P_0 - C) + \delta)^{-1}\right) \tag{6.12}$$

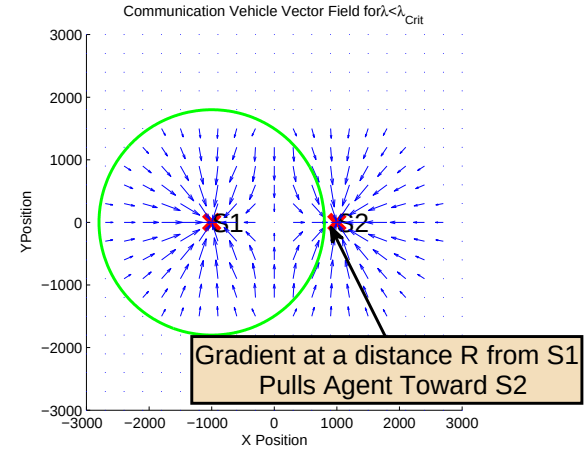
Therefore we conclude that if the initial configuration has a cost $H_{\text{initial}} < H_{d_{\min}}$ then the routers will remain connected for all time.

Finding Critical Value of ρ to Ensure Connectivity

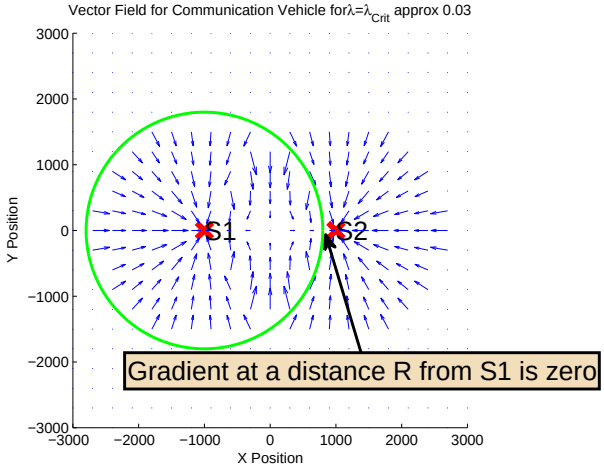
We find the ρ for which two nodes that are currently neighbors, will not move a distance larger than R from each other. The intuition behind this critical ρ value is the observation that as the distance between two nodes d_{ij} approaches the communication radius R, ρ can be chosen such that the generalized gradient $\frac{\partial H}{\partial c_i}$ will have a zero component in the direction pointing away from node j, and thus the node i will never move further than the distance R away from node j, $\forall j \in \mathcal{N}_i$. This corresponds to the ρ that forces

$$-\frac{\partial H}{\partial c_i}^T (c_i - p_j) = 0 \tag{6.13}$$

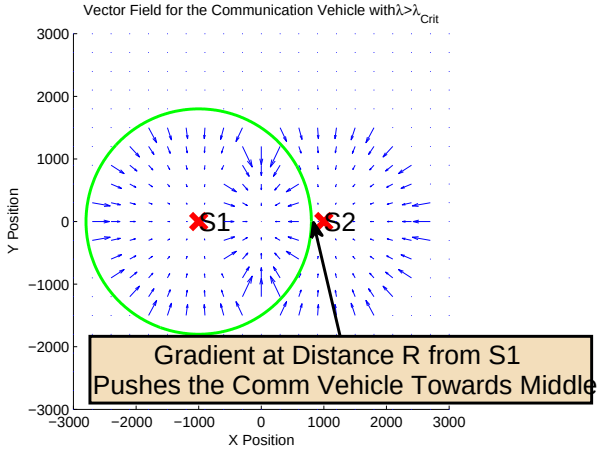
Where the vector $(c_i - p_j)$ points from node j to node i and node j is a neighbor of i at a distance approaching R. We expand Equation (6.13):



(a)



(b)



(c)

Figure 6-3: This plot shows the force felt by a communication vehicle in the presence of two clients, S1 and S2. It demonstrates the effect of the design parameter ρ on the communication vehicle gradient field where connectivity is maintained for $\rho \geq \rho_{\text{Crit}}$.

$$-\Big(\frac{\partial H_{ij}}{\partial c_i}+\frac{\partial H_{ji}}{\partial c_i}+\sum_{\{u,w\}\neq\{i,j\},\{j,i\}}\frac{\partial H_{uw}}{\partial c_i}\Big)^T(c_i-p_j)=0 \tag{6.14}$$

Where

$$\frac{\partial H_{uw}}{\partial c_i}=-\frac{\partial SIR_{uw}}{\partial c_i}(1+\rho(SIR_{uw}+\delta)^{-2}) \tag{6.15}$$

As seen in Equation (6.14) and (6.15), the gradient-based controller for node i is a combination of the gradients of the SIR values between node i and node l, $\forall l \in \mathcal{N}_i$, weighted by the inverse of the value of the SIR for that pair c_i-p_l. This weighting is directly influenced by ρ, but goes to zero when $\rho = 0$. Therefore, it is intuitive that a larger ρ value will amplify the effect of the value $SIR_{uw} \to 0$ in Eq (6.15), and thus the contribution of the gradient on i from the node whose distance is approaching R will dominate for larger values of ρ. Solving for ρ from Equation (6.14), we find:

$$\rho \;=\frac{-\sum_u^N\sum_w^N\frac{\partial SIR_{uw}}{\partial c_i}^T(c_i-p_j)}{\sum_u^N\sum_w^N(SIR_{uw}+\delta)^{-2}\frac{\partial SIR_{uw}}{\partial c_i}^T(c_i-p_j)} \tag{6.16}$$

As the distance $d_{ij} \to R$, we note that:

$$\frac{\partial SIR_{ij}}{\partial c_i}^T(c_i-p_j)\to\beta P_0(R^\beta+1)^{-2}R^{\beta-2}(Ni+\sum_{l\in\mathcal{N}_i}f_{il})^{-1}R^2. \tag{6.17}$$

and

$$SIR_{ij}=SIR_{ji}\to\frac{1}{\delta}. \tag{6.18}$$

To find ρ_{crit} we must analyze the upper bound to the equation (6.16). This corresponds to finding the case where the link i-j is most easily disconnected. From the Equation (6.14) we see that the upper bound is when the gradient dot product

$\frac{\partial H_{uw}}{\partial c_i}^T(c_i - p_j)$ is maximized, or equivalently, when all agents $p_l \neq p_j$ have a maximum value of the gradient $\frac{\partial H_{uw}}{\partial c_i}$ in the direction exactly opposite to the vector $(c_i - p_j)$. If we ignore node interference in the Signal-to-Interference Ratio to get a upper bound on H_{uw}, this is the case where all nodes not including node j are co-located at a point that is opposite of the direction $(c_i - p_j)$ with respect to node i so that the vector exactly opposite to $(c_i - p_j)$ is $(p_w - c_i)$. We place all $N-2$ nodes at a distance $(R - \xi)$ from node i, where

$$\xi = \arg\max_{\xi} \frac{\partial H_{wi}}{\partial c_i}^T (p_w - c_i). \tag{6.19}$$

We place all $N-2$ agents at a distance $(R-\xi)$ from node i and later solve to find the ξ for which $\frac{\partial SIR_{uw}}{\partial c_i}^T(u-w)$ is maximized. This case also corresponds to the minimum cost disconnected graph configuration. We find ρ_{crit} for this case:

Thus the smallest value of lambda for which we are guaranteed to preserve connectivity is:

$$\begin{aligned}
\rho_{\text{crit}} = -\Big(& - \frac{\beta P R^\beta}{(R^\beta + 1)^2 (N_i + (N-2)P)} - \frac{\beta P R^\beta}{(R^\beta + 1)^{-2} N_j} \\
& + \sum_w^N \frac{\partial SIR_{iw}}{\partial c_i}^T (c_i - p_j) + \frac{\partial SIR_{wi}}{\partial c_i}^T (c_i - p_j) \Big) \\
& * \Big(2(\frac{1}{\delta})^2 + \sum_w^N (SIR_{iw} + \delta)^{-2} \frac{\partial SIR_{iw}}{\partial c_i}^T (c_i - p_j) \\
& + (SIR_{wi} + \delta)^{-2} \frac{\partial SIR_{wi}}{\partial c_i}^T (c_i - p_j) \Big)^{-1}
\end{aligned} \tag{6.20}$$

$$\frac{\partial SIR_{iw}}{\partial c_i}^T (c_i - p_j) = -\frac{a_{iw}}{N_i}(R - \xi)R - \frac{\frac{P_0}{(R-\xi)^\beta} - C}{N_i^2}(\frac{\partial N_i}{\partial c_i}^T (c_i - p_j) + a_{ij}R^2) \tag{6.21}$$

and

$$\frac{\partial SIR_{wi}}{\partial c_i}^T (c_i - p_j) = -\frac{a_{wi}}{N_w}(R-\xi)R \tag{6.22}$$

$$SIR_{iw} = (\frac{P_0}{(R-\xi)^\beta + 1} - C)(N_i)^{-1} \tag{6.23}$$

$$SIR_{wi} = (\frac{P_0}{(R-\xi)^\beta + 1} - C)(N_w)^{-1} \tag{6.24}$$

$$a_{iw} = a_{wi} = \beta P_0((R-\xi)^\beta + 1)^{-2}(R-\xi)^{\beta-2} \tag{6.25}$$

Because we have found the minimum value of ρ for which $-\frac{\partial H}{\partial c_i}^T(c_i - p_j) = 0, \forall j$, we have shown that if we choose $\rho \geq \rho_{\text{crit}}$, node c_i will never move out of the ball of radius R centered at p_j.

□

6.4 Emperical Results

In this section we present the results of implementing our decentralized gradient descent controller on a quadrotor hardware testbed, hardware-in-the-loop simulations, and MATLAB simulations.

We implement the controller on a group of AscTec Hummingbird flying quadrotor robots providing network coverage for ground vehicles, using xBee-PRO modules for wireless communication. We present aggregate results of ten hardware experiment trials, demonstrating positioning of a team of three quadrotor routers to provide optimized communication for a group of three ground clients. We also present the results of hardware-in-the-loop simulations for up to three routers and four ground vehicles, and MATLAB simulation results for up to eight routers and eight ground vehicles. Our MATLAB simulations also show that we can adjust the behavior of the routers to optimize SIR values over individual links, or an equalization of SIR values over all links in the communication graph, by adjusting a design parameter ρ in the cost function H.

Hardware Implementation

We tested our controller on a group of three routers which are AscTec Hummingbird flying quad-rotor robots each with an ARM micro-processor and 2.4 GHz xBee modules for wireless communication, and three ground vehicles. We conducted the experiments in a room equipped with a Vicon motion capture system where position information was broadcasted wirelessly to each robot and all computation was performed onboard each of the robots in real time.

For our hardware experiments we set the controller parameters $\rho = 1 > \rho_{\text{crit}}$ and $\delta = 0.001$, and the communication parameter $\beta = 2$. We demonstrate that the routers acheive a configuration that locally minimizes the cost H. Figure 6-4 shows minimization of the cost function H averaged over ten trials with errorbars indicating the one standard deviation around the mean. Each experiment lasted on the order of one minute.

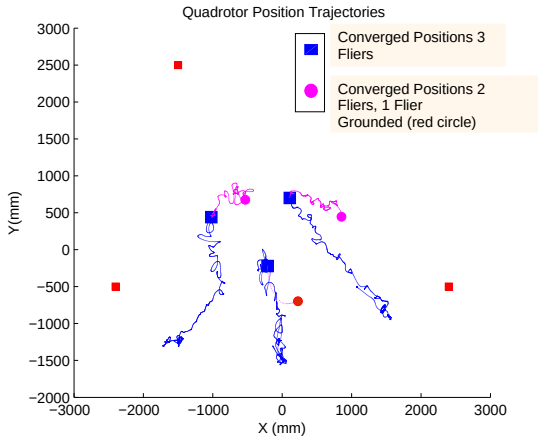
We demonstrated the adaptive capabilities of the controller by disabling one of the routers and relocating this router to a fixed position on the ground. As shown in Figure 6-4, the remaining routers re-adjust their equilibrium position to compensate for this change in the system. (Figure 6-4).

Hardware-in-the-Loop Simulation

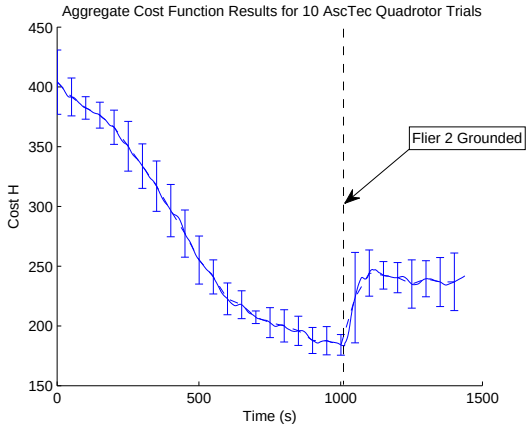
We tested the controller on a total of 7 ARM micro-controllers communicating wirelessly via xBee-XSC wireless modules. The tests were conducted on four ground vehicles, and three aerial communication vehicles with control parameters $\rho = 1 > \rho_{\text{crit}}$ and $\delta = 0.001$. Figure 6-5 shows the minimization of the cost and Figure the trajectories of the aerial vehiles with final equilibrium positions marked as blue circles.

MATLAB Simulation

We tested a configuration with 16 total vehicles, where 8 are ground clients and the remaining 8 are aerial communication vehicles. We set the control parameters $\delta = 0.001$ and the ρ parameter to $\rho = 10 > \rho_{crit}$ to target equalized SIR values

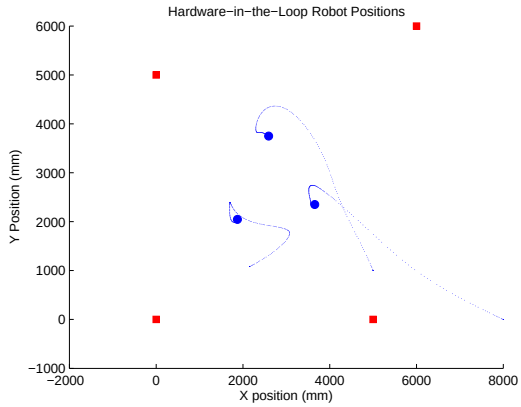


(a) Initial and Converged Positions for Hardware Trial.

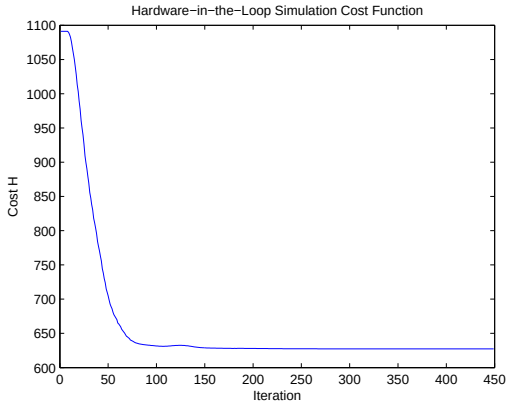


(b) Aggregate Cost Function over Ten Hardware Trials.

Figure 6-4: Position trajectories and aggregate cost function for three routers (shown as blue solid line in Fig 6-6(a)) with router equilibrium positions marked as blue squares and ground client positions marked as red squares. After reaching equilibrium one of the routers is deactivated and moved to the side while the remaining routers find a new equilibrium position (post-deactivation trajectories shown in dotted magenta line).



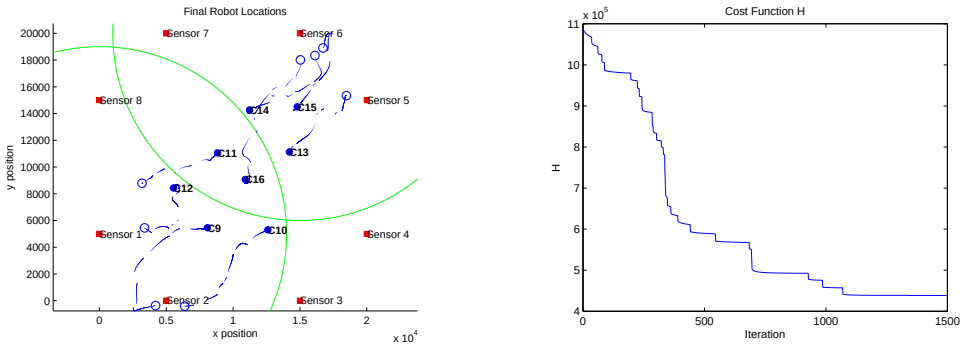
(a) Hardware-in-the-Loop Initial and Converged Positions.



(b) Hardware-in-the-Loop Cost Function.

Figure 6-5: Position data and cost function for hardware-in-the-loop simulation where router trajectories are shown as blue lines and converged positions as blue dots. The ground clients are plotted as red squares in this figure.

amongst routers. The routers shown in blue have initial positions at a depot in the top right and bottom left corners. Green circles denote the communication radius of the farthest clients, clients 1 and 6, to demonstrate that routers are initialized out of communication range with other clients and routers in the team. The resulting node trajectories and cost function demonstrates non-smooth transitions for the points where agents enter each others communication radius as shown in Figure 6-6.



(a) Initial and Converged Positions.

(b) Non-smooth Cost Function.

Figure 6-6: Matlab simulation results of converged positions and position trajectories for 8 routers and 8 ground vehicles with non-smooth cost function H. Initial router positions are shown as blue circles, converged positions are shown as filled blue circles, and trajectories are shown as a blue line in Figure 6-6(a). Communication radius of clients 1 and 6 shown in green demonstrate that not all nodes are in communication initially. Trajectories as well as cost function show non-smooth transitions at the points where nodes enter each others communication neighborhood.

Chapter 7

Conclusions, Lessons Learned, and Future Work

In this thesis we have presented algorithms for computing router placements for servicing a non-cooperative set of client vehicles. We have examined centralized formulations for finding router placements that are optimal with respect to router-client assignments, and decentralized formulations for the case where client vehicles are assumed to be static. This thesis has maintained a pronounced emphasis on developing a theoretical framework, as well as novel techniques in hardware, whose combination leads to a treatment of the communication coverage problem that can be deployed in real-world settings. In order to do this, we developed the necessary machinery to optimize over large-scale systems. In particular, we show that as the number of client agents in the system increase, we can continue to solve our router placement problem efficiently through the use of coresets. This allows us to update router configurations in real-time, while providing an exact bound on the approximation error. We also develop novel techniques and a generalized optimization problem for router placements that take into account realistic wireless communication phenomena; resulting in systems that can optimize over competing client demands while retaining a simple quadratic structure.

7.1 Lessons Learned

One valuable lesson learned throughout the course of this thesis is that working on the intersection of theory and practice is a very difficult endeavour, with the reward that a success in this domain achieves generality due to a strong theoretical backing, and applicability to real-world implementation. In particular, the frontier between wireless communication and robotics has been especially challenging due to the inherent incongruence between the two fields. In robotics, performance metrics are intimately tied to positions and orientations in Euclidean spaces and thus a model that ties spatial references and performance is necessary. In wireless communications, models tying signal quality to relative positioning are often rejected or given second-tier status to experimental approaches due to the inherent complexity of signal propagation. However, as with most things, simplicity is highly valued. So merging these two seemingly disconnected perspectives into a simple controller that leverages spatial positioning of routers that achieves the desired communication quality objectives in real world environments seemed like an impossible task at times. However, our ability to ultimately produce a contribution in this domain was possible through the melding and collaboration of many distinct fields: namely computational geometry, robotics, and communications. Therefore the lesson here is that cross-disciplinary collaboration allows for new perspectives on problems that otherwise, from the vantage point of a single discipline, may seem impossible.

There were many challenges in moving from theory to practice. In particular, for optimizations over wireless signal quality (that is notoriously difficult to predict), aiming for solutions that are globally optimal with respect to the entire environment still remains an open problem in our opinion. One may be tempted to map out the signal strength field in the entire environment, similar to an artificial potential field indicating the directions of increased signal strength, and then running a globally optimal path planner on this artificial potential field. While this is a plausible approach, it necessitates static environments which is a very strong assumption particularly for robotic applications. From our experimentations we found that even a person walking

through the room is enough disturbance to change this artificial potential field map. In fact, this disturbance to the signal strength on the wireless links is so substantial that it has been recently used to track people moving behind walls in office environments [2]. Also, this map is expensive to generate since it requires a mobile router to physically move through the environment in order to construct it. Therefore any changes to this map are expensive to re-compute.

Finally, we learned that even naive approaches have a lot of inherent value for their simplicity and should not be disregarded. In the case of this thesis, a simplified first approach to handling the communication coverage problem was to treat the communication quality as an open-loop deterministic quantity by using the Euclidean disk assumption. While this does not always reflect reality in practice (particularly for indoor environments), it provides a lot of power in its simplicity and allowed for a theoretical formalization of the problem. By understanding the key elements of 1) the optimization formulation, whose backbone is the k-center and reachability problems (based on a Euclidean distance metric) and 2) the feedback necessitated by a real world communication model, we were able to find a generalization of the naive formulation to capture real communication phenomena without sacrificing the simplicity of the naive approach.

7.2 Future Work

We believe that the recent works in the area of communication and robotics, including the current thesis, are just scraping the surface of the realm of possibilities. In terms of future work for the topics investigated in this thesis, we are interested in 1) problems of investigating time constants and their effect on keeping up with mobile clients, 2) autonomous rate adaptation to determine the ideal rates to provide client agents (not just based on their demands) in order to maximize global throughput of the network, and 3) various extensions of the scalability of our systems for increased generality.

7.2.1 The Effects of Time Delays and Positional Uncertainties on the Network

While in this thesis we strived to mitigate many challenges that we found non-amenable to robotics applications such as allowing for arbitrary client motion, unknown environments, and large-scale networks, there remain many challenges that may necessitate solutions for certain implementations. We have begun research directions along the lines of mathematical characterization of the effect of time delays and position uncertainties on the system. For example, in a centralized computation scheme where a leader node or a central home station computes router positions at each iteration, the time delay associated with disseminating waypoint commands may make the computed solution stale if these delays are significant. In the work of the thesis we consider that these time delays are insignificant, but depending on the hardware platform this may not be the case. For the case of positional uncertainties, in the thesis we provide results for the case where client positions may be unknown. This relaxation to the problem is of great use particularly because we view client-agents as non-cooperative members of the network and thus predicting their positions is both difficult and irrelevant for the problem statement of providing requested levels of communication service. However, we do assume that router positions are known throughout the thesis. This is especially important for the sections of the thesis that measure the signal strength to relative heading angle mappings. This is because signal processing techniques are used to derive these mappings using a single antenna mounted to a iRobot platform that is moving at continuous velocity in a straight-line path. Also for the case where we wish to use reachability for tracking mobile clients, the reachability formulation would have to be altered to account for uncertainties in the router positions. This topic is discussed in the work [8] and we have begun research along these directions.

7.2.2 Generalized Router and Client Dynamics

Future work includes generalizations to the problems presented in this chapter of the thesis. Particularly for the cases of 1) general linear time-invariant (discrete-time) dynamics and 2) where the cost of interest is based on the k-means problem as opposed to the k-center problem. We discuss these two approaches briefly in this section of the thesis.

General Discrete-time Linear Dynamics

We consider general linear dynamics for both the router and client vehicles. In this case we have:

$$p_{j_{t+1}} = Ap_{j_t} + Gw_{j_t},\ j \in \{1, \ldots, n\}, \tag{7.1}$$

$$w_{j_t} \in W_j = \{w \in \mathbb{R}^d :\ w^T Q_j w \leq 1\}, \tag{7.2}$$

$$c_{i\tau+1} = Ac_{i\tau} + Bu_{i\tau},\ i \in \{1, \ldots, k\}, \tag{7.3}$$

$$u_{i\tau} \in U_i = \{u \in \mathbb{R}^d :\ u^T \Omega_i u \leq 1\} \tag{7.4}$$

Where A, G, B are given matrices of appropriate dimension, and Q_j, Ω_i are given positive definite matrices for all $i \in \{1, \ldots, k\}$ and all $j \in \{1, \ldots, n\}$. Given a time window of interest, T, the question of whether k routers can be steered to a set of waypoints $C_{\tau+1}$ generated by a solution to the k-connected centers problem can be determined by solving for the T-step transition matrix $\Gamma_T = [A^{T-1}B|A^{T-2}B|\ldots|B]$ and applying the *Cayley-Hamilton* Theorem:

Theorem 7.2.1 ([27] Thm 22.2)**.** *For* $0 \leq k \leq d \leq l$

$$Ra(\Gamma_k) \subseteq Ra(\Gamma_d) = Ra(\Gamma_l) \tag{7.5}$$

where $Ra(M)$ *is the range of the matrix* M*. Following from the above equation, the set of states reachable in some finite number of steps by appropriate choice of control is the subspace of states reachable in* d *steps.*

Therefore, given that $Ra(\Gamma_T) = \mathbb{R}^d$ (i.e. Γ_T is full rank), and the reachability set $\tilde{\mathcal{E}}$ (updated from (3.11) to include generalized dynamics) is not empty for the given client dynamics in Equation (7.1), there will exist a sequence of control actions $u_1, \ldots, u_l$, $l \leq T$, such that each router can reach the desired configuration C^*.

For the case where the C^* is a configuration of router placements that is reachable and results from a k-means optimization, an LQR formulation can be solved to find linear controls to move the centers into the desired configuration. We discuss this next.

7.2.3 A k-Means Formulation

We consider the case where the cost of interest is a k-means cost as defined below:

Definition 7.2.2 (Connected k-Means Problem)**.** *The solution to the connected k-means problem for a set P, a minimum spanning tree $\mathcal{T}^*$ over the centers, and a set of connectivity neighborhoods $\mathcal{N} \in \mathcal{Q}(P,k)$ returns the set C^* where*

$$r_k(P,C) := \min_{\mathcal{N} \in \mathcal{Q}(P,k)} \sum_{i=1}^{k} \sum_{p_j \in \mathcal{N}(c_i)} \|p_j - c_i\| \tag{7.6}$$

$$b_k(C) := \sum_{(c,c') \in \mathcal{T}^*} \|c - c'\| \tag{7.7}$$

$$C^* = \arg\min_{C} \{r_k(P,C) + b_k(C)\} \tag{7.8}$$

and C^* minimizes this cost over every set $C \subseteq \mathbb{R}^d$ with cardinality $|C| = k$.

For all edges $(i,j) \in E$ corresponding to $c_i \in C_\tau$ and $p_j \in \mathcal{N}(ci)$ where $\mathcal{N} \in \mathcal{Q}(P,k)$, we define a relative state vector

$$z_i = [(c_i - q_1), (c_i - q_2), \ldots, (c_i - q_L)]^T \tag{7.9}$$

$$\text{where } L = |\mathcal{N}(c_i)| \tag{7.10}$$

and a corresponding diagonal block matrix $\tilde{E}$ with the matrices $\tilde{\mathcal{E}}_{ij}$ (from Equa-

tion (3.11)) on its diagonal for every edge $(i, j) \in E$. Consider the following LQR problem from [8]

$$J^* = \min u^T \mathcal{R} u + z_{t+1}^T \tilde{E} z_{t+1} \tag{7.11}$$

$$u = [(c_{1t+1} - c_{1t})^T, \ldots, (c_{kt+1} - c_{kt})^T] \tag{7.12}$$

$$z_{ij_{t+1}} = z_{ij_t} + u_{it} \tag{7.13}$$

where $\mathcal{R}$ is a diagonal block matrix with the control limit matrices Ω on its diagonal and $J^* \leq 1$ implies that there is a control sequence attaining connectivity. We can see that, for a given assignment of clients to centers $\mathcal{N} \in \mathcal{Q}(P, k)$, there is a strong relationship between the minimizer of the above LQR problem and that of the connected k-means problem. It is well known from optimal control theory that there exists a closed-form solution to Equation (7.11) and linear controllers for moving a set of points C_t to the solution C^* of the LQR problem in (7.11). Thus this also suggests a linear control solution for the connected k-means problem from Definition 7.2.2, for a given router-client assignment $\mathcal{N}$.

The extension of our scalability results to incorporate more general communication quality models (beyond the Euclidean models), and for different communication costs such as k-means is an interesting topic of future work. Router placements based on a k-means formulation would minimize the sum of squared distances to all the clients, as opposed to our current approach which minimizes the maximum distance between any client and its nearest router. This would be useful for cases where outlier clients are not given much importance with respect to the majority of the clients (this approach would not be suitable for life-critical implementations such as on a battlefield where you would not want to ignore any soldier even if they are in a remote part of the environment).

Automatic Rate Selection and Multiple Interference Regions

The work of this thesis considers a single interference region where one frequency is used for communication and thus all nodes have the potential to collide transmissions. In future work we consider the case where some of these transmissions fall into different interference regions and where the division of these regions would become part of the optimization. For example, the equation for service discrepancy (Equation (7.14)) would include an interference region normalization parameter N_{ij}

$$w_{ij} = \max(\alpha_j \frac{(q_j/N_{ij} - \tilde{q}_{ij})}{q_j/N_{ij}}, 0), \tag{7.14}$$

where N_{ij} denotes the number of interfering nodes in the same interference region as the link (i, j). We also consider the problem of automatic rate selection. Currently we position routers to provide a rate greater than or equal to the maximum requested rate for each client. By adding a rate selection optimization we can increase the amount of throughput on the network by adapting to the lowest necessary transmission rate to achieve the desired throughput. We emphasize however that this objective is separate from the positioning of the routers.

Bibliography

[1] N. Roy A. Bachrach, R. He. Autonomous flight in unknown indoor environments. *International Journal of Micro Air Vehicles*, 1(4):217–228, December 2009.

[2] Fadel Adib and Dina Katabi. See through walls with wi-fi. In *SIGCOMM*, 2013.

[3] P. Agarwal, H. Edelsbrunner, Otfried Schwarzkopf, and Emo Welsl. Euclidean minimum spanning trees and bichromatic closest pairs. *Discrete and Computational Geometry*, 6(1), 1991.

[4] P. Agarwal, S. Har-Peled, and K. Varadarajan. Geometric approximation via coresets. *Discrete and Computational Geometry*, 52, 2005.

[5] Nora Ayanian, Vijay Kumar, and Daniel Koditschek. Synthesis of controllers to create, maintain, and reconfigure robot formations with communication constraints. In *Intelligent Robotic Systems, IEEE International Conference on*, 2009.

[6] A. Bachrach, A. de Winter, Ruijie He, G. Hemann, S. Prentice, and N. Roy. Range - robust autonomous navigation in gps-denied environments. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 1096–1097, 2010.

[7] Mihai Bdoiu, Sariel Har-peled, and Piotr Indyk. Approximate clustering via core-sets. In *In Proc. 34th Annu. ACM Sympos. Theory Comput*, pages 250–257, 2002.

[8] Dimitri Bertsekas. *Control of Uncertain Systems with A Set-Membership Description of the Uncertainty*. PhD thesis, Massachusetts Institute of Technology, Cambridge, 1971.

[9] Dimitri Bertsekas. Infinte-time reachability of state-space regions by using feedback control. *Automatic Control, IRRR Transactions on*, 17:604–613, 1972.

[10] Dimitri Bertsekas and I. Rhodes. On the minimax reachbility of target sets and target tubes. *Automatica*, 7:233–247, 1971.

[11] Franco Blanchini and Stefano Miani. *Set-Theoretic Methods in Control*. Birkhauser, Boston, 2008.

[12] Adam Bry, Abraham Bachrach, and Nicholas Roy. State estimation for aggressive flight in gps-denied environments using onboard sensing. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2012)*, St Paul, MN, 2012.

[13] O Burdakov, P Doherty, K Holmberg, J Kvarnstrom, and P R Olsson. Positioning unmanned aerial vehicles as communication relays for surveillance tasks. In *Robotics Science and Systems, Conference on*, 2009.

[14] Mark Campbell1, Magnus Egerstedt, Jonathan P. How, and Richard M. Murray. Autonomous driving in urban environments: approaches, lessons and challenges. *Philosophical transactions of the royal society*, 368(1928):4649–4672, October 2010.

[15] T.M. Chan. Dynamic coresets. *Discrete & Computational Geometry*, 42(3):469–488, 2009.

[16] M. Charikar, S. Khuller, D. M. Mount, and G. Narasimhan. Algorithms for facility location problems with outliers. In *Proc. 12th Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 642–651, 2001.

[17] G.L. Charvat, Leo C. Kempel, E.J. Rothwell, C.M. Coleman, and E. Mokole. An ultrawideband (uwb) switched-antenna-array radar imaging system. In *Phased Array Systems and Technology (ARRAY), 2010 IEEE International Symposium on*, pages 543–550, 2010.

[18] Hsieh-Chung Chen, Tsung-Han Lin, H.T. Kung, Chit-Kwan Lin, and Youngjune Gwon. Determining rf angle of arrival using cots antenna arrays: A field evaluation. In *MILCOM*, 2012.

[19] Felix Chernousko. *State Estimation for Dynamic Systems*. CRC Press, 1994.

[20] K. Chetty, G.E. Smith, and K. Woodbridge. Through-the-wall sensing of personnel using passive bistatic wifi radar at standoff distances. *Geoscience and Remote Sensing, IEEE Transactions on*, 50(4):1218–1226, 2012.

[21] Space Data Co. Space data skysat: balloon powered communication for military and rural areas.

[22] Alejandro Cornejo, Ruy Ley-Wild, Fabian Kuhn, and Nancy Lynch. Keeping mobile robot swarms connected. Technical report, MIT-CSAIL, June 2009.

[23] J Cortes, S Martinez, T Karatas, and F Bullo. Coverage control for mobile sensing networks. In *IEEE Transactions of Robotics and Automation*, 2004.

[24] Jorge Cortes. Discontinuous dynamical systems. *Control Systems Magazine,IEEE*, 28:36–73, 2008.

[25] Jorge Cortes, Sonia Martinez, and Francesco Bullo. Spatially-distributed coverage optimization and control with limited-range interactions. In *ESAIM. Control, Optimisation and Calculus of Variations*, pages 691–719, 2004.

[26] E.M. Craparo, J.P. How, and E. Modiano. Throughput optimization in mobile backbone networks. *Mobile Computing, IEEE Transactions on*, 10(4):560–572, 2011.

[27] Mohammed Dahleh, Munther A. Dahleh, and George Verghese. Lectures on dynamic systems and control.

[28] M.C. De Gennaro and A. Jadbabaie. Decentralized control of connectivity for multi-agent systems. In *Decision and Control, 2006 45th IEEE Conference on*, 2006.

[29] Jason Derenick, J. Fink, and V. Kumar. Localization using ambiguous bearings from radio signal strength. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 3248–3253, 2011.

[30] Jerry Ding, Eugene Li, Haomiao Huang, and Claire J. Tomlin. Reachability-based synthesis of feedback policies for motion planning under bounded disturbances. In *ICRA*, pages 2160–2165, 2011.

[31] D. Feldman, S. Gil, R. Knepper, B. Julian, and D. Rus. K-robots clustering of moving sensors using coresets. In *International Conference on Robotics and Automation (ICRA), 2013*, 2013.

[32] D. Feldman and Daniel Golovin. Dynamic bicriteria approximations. *Manuscript*, 2011.

[33] Dan Feldman, Amos Fiat, Micha Sharir, and Danny Segev. Bi-criteria linear-time approximations for generalized k-mean/median/center. In *Symposium on Computational Geometry'07*, pages 19–26, 2007.

[34] J. Fink, A. Ribeiro, and V. Kumar. Robust control for mobility and wireless communication in cyber-physical systems with application to robot teams. *Proceedings of the IEEE*, 2012.

[35] J. Fink, A. Ribeiro, and V. Kumar. Robust control of mobility and communications in autonomous robot teams. *Access, IEEE*, 1, 2013.

[36] J. Fink, A. Ribeiro, V. Kumar, and B.M. Sadler. Optimal robust multihop routing for wireless networks of mobile micro autonomous systems. In *MILITARY COMMUNICATIONS CONFERENCE, 2010 - MILCOM 2010*, pages 1268–1273, 2010.

[37] Patrick J. Fitch. *Synthetic Aperture Radar*. Springer-Verlag, 1988.

[38] Sue Marek for Wireless Week. Skysites move closer up to reality firm's weather balloons rise to fill gaps in rural areas.

[39] E Frazzoli and F Bullo. Decentralized algorithms for vehicle routing in a stochastic time-varying environment. In *Decision and Control, IEEE International Conference on*, 2004.

[40] Eric W. Frew. Information-theoretic integration of sensing and communication for active robot networks. *Mobile Networks and Applications*, 14:267–280, 2009.

[41] Stefan Geirhofer, Lang Tong, and Brian M Sadler. Dynamic spectrum access in wlan channels: empirical model and its stochastic analysis. In *Proceedings of the first international workshop on Technology and policy for accessing spectrum*, page 14. ACM, 2006.

[42] Stefan Geirhofer, Lang Tong, and Brian M Sadler. A measurement-based model for dynamic spectrum access in wlan channels. In *Military Communications Conference, 2006. MILCOM 2006. IEEE*, pages 1–7. IEEE, 2006.

[43] Stefan Geirhofer, Lang Tong, and Brian M Sadler. Cognitive radios for dynamic spectrum access-dynamic spectrum access in the time domain: Modeling and exploiting white space. *Communications Magazine, IEEE*, 45(5):66–72, 2007.

[44] Stefan Geirhofer, Lang Tong, and Brian M Sadler. Cognitive medium access: constraining interference based on experimental models. *Selected Areas in Communications, IEEE Journal on*, 26(1):95–105, 2008.

[45] S. Gil, D. Feldman, and D. Rus. Communication coverage for independently moving robots. In *IROS*, 2012.

[46] S. Gil, D. Feldman, and D. Rus. Communication coverage for independently moving robots. *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems*, 2012.

[47] Stephanie Gil, Swarun Kumar, Dina Katabi, and Daniela Rus. Adaptive communication in multi-robot systems using directionality of signal strength. In *Proceedings of the 16th International Symposium on Robotics Research*, 2013.

[48] Stephanie Gil, Samuel Prentice, Nicholas Roy, and Daniela Rus. Decentralized control for optimizing communication with infeasible regions. In *Proceedings of the 15th International Symposium on Robotics Research*, 2011.

[49] Stephanie Gil, Mac Schwager, Brian Julian, and Daniela Rus. Optimizing communication in air-ground robot networks using decentralized control. In *Proc. ICRA*, 2010.

[50] Jeremy H. Gillula, Gabriel Hoffmann, Haomiao Huang, Michael P. Vitus, and Claire J. Tomlin. Applications of hybrid reachability analysis to robotic aerial vehicles. *I. J. Robotic Res.*, 30(3):335–354, 2011.

[51] Juan J. Glvez, Pedro M. Ruiz, and Antonio F.G. Skarmeta. Multipath routing with spatial separation in wireless multi-hop networks without location information. *Computer Networks*, 55(3):583 – 599, 2011.

[52] Andrea Goldsmith. *Wireless Communications*. Cambridge University Press, 2005.

[53] T. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38, 1985.

[54] L.C. Graham. Synthetic interferometer radar for topographic mapping. *Proceedings of the IEEE*, 62(6):763–768, 1974.

[55] Gereon Grahling, piotr inyk, and christian sohler. Sampling in dynamic data streams and applications. *International Journal of Computational Geometry and Applications*, 18(01n02):3–28, 2008.

[56] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 1.21. `../../cvx`, April 2011.

[57] Piyush Gupta and P. R. Kumar. The capacity of wireless networks, 1999.

[58] Piyush Gupta and P. R. Kumar. The capacity of wireless networks, 1999.

[59] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. Predictable 802.11 packet delivery from wireless channel measurements. In *CCR*, 2010.

[60] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. Tool release: Gathering 802.11n traces with channel state information. *ACM SIGCOMM CCR*, 2011.

[61] S. Har-Peled and S. Mazumdar. On coresets for k-means and k-median clustering. In *Proc. 36th Ann. ACM Symp. on Theory of Computing (STOC)*, pages 291–300, 2004.

[62] Sariel Har-peled and Soham Mazumdar. Coresets for k-means and k-median clustering and their applications. In *In Proc. 36th Annu. ACM Sympos. Theory Comput*, pages 291–300, 2003.

[63] D. Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Inf. Comput.*, 100(1):78–150, 1992.

[64] Dorit S. Hochbaum. Easy solutions for the k-center problem or the dominating set problem on random graphs. In *Analysis and Design of Algorithms for Combinatorial Problems*, volume 109, pages 189 – 209. North-Holland, 1985.

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

[66] Ali Jadbabaie, Jie Lin, and A.S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *Automatic Control, IEEE Transactions on*, 48:988–1001, 2003.

[67] Kiran Joshi, Steven Hong, and Sachin Katti. Pinpoint: localizing interfering radios. In *NSDI*, 2013.

[68] S. Karaman and E. Frazzoli. Sampling-based optimal motion planning for non-holonomic dynamical systems. In *IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany, 2013.

[69] David Kotz and Tristan Henderson. A community resource for archiving wireless data.

[70] Richard J Kozick and Brian M Sadler. Source localization with distributed sensor arrays and partial spatial coherence. *Signal Processing, IEEE Transactions on*, 52(3):601–616, 2004.

[71] Swarun Kumar, Lixin Shi, Nabeel Ahmed, Stephanie Gil, Dina Katabi, and Daniela Rus. Carspeak: a content-centric network for autonomous driving. *SIGCOMM*, 2012.

[72] Alexander Kurzhanski and Istvan Valyi. *Ellipsoidal Calculus for estimation and control*. Birkhauser, 1997.

[73] Google X Research Labs. Loon for all: balloon powered internet for everyone.

[74] Jerome Le Ny, A. Ribeiro, and G.J. Pappas. Adaptive communication-constrained deployment of mobile robotic networks. In *American Control Conference (ACC), 2012*, pages 3742–3747, 2012.

[75] M. Lindh, K. Johansson, and A. Bicchi. An experimental study of exploiting multipath fading for robot communications. In *Proceedings of Robotics: Science and Systems*, Atlanta, GA, USA, June 2007.

[76] Magnus Lindhé and Karl Henrik Johansson. Adaptive exploitation of multipath fading for mobile sensors. In *ICRA*, 2010.

[77] Yonghe Liu and E. Knightly. Opportunistic fair scheduling over multiple wireless channels. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 2, pages 1106–1115 vol.2, 2003.

[78] P. C. MAHALANOBIS. On the generalized distance in statistics. *Proceedings of the National Institute of Sciences (Calcutta)*, 2:49–55, 1936.

[79] M. MalmirChegini and Y. Mostofi. On the spatial predictability of communication channels. *Wireless Communications, IEEE Transactions on*, 11(3):964–978, 2012.

[80] Sonia Martinez, Jorge Cortes, and F. Bullo. Motion coordination with distributed information. *Control Systems Magazine, IEEE*, 2007.

[81] Nimrod Megiddo and Arie Tamir. On the complexity of locating linear facilities in the plane. In *Operations research letters*, volume 1, 1982.

[82] Mehran Mesbahi and Magnus Egerstedt. *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010.

[83] Nathan Michael, Michael Zavlanos, Vijay Kumar, and George Pappas. Maintaining connectivity in mobile robot networks. In *Experimental Robotics*, volume 54 of *Springer Tracts in Advanced Robotics*, pages 117–126. Springer Berlin/Heidelberg, 2009.

[84] Gerald M.Knoblach and Eric A.Frische. Airborne constellation of communications platforms and method. *U.S. Patent 6,628,941*, 1999.

[85] L Moreau. Stability of continuous-time distributed consensus algorithms. In *Decision and Control,Proc of International Conference on*, 2004.

[86] E. Mueller, S.Z. Yong, M. Zhu, and E. Frazzoli. Anytime computation algorithms for stochastically parametric approach-evasion differential games. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2013.

[87] Aruba Networks. Outdoor antennas and rf coverage strategies . http://www.arubanetworks.com/vrd/outdoormimovrd/wwhelp/wwhimpl/common/html/wwhelp.htm#context=OutdoorMIMOVRD&file=chap4.html.

[88] A. Nosratinia, T.E. Hunter, and A. Hedayat. Cooperative communication in wireless networks. *Communications Magazine, IEEE*, 42(10):74–80, 2004.

[89] R. Olfati-Saber, J.A. Fax, and R.M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 2007.

[90] M Pavone, E Frazzoli, and F Bullo. Distributed policies for equitable partioning: Theory and applications. In *Decision and Control, IEEE International Conference on*, 2008.

[91] Tien Pham and Brian M Sadler. Wideband array processing algorithms for acoustic tracking of ground vehicles. *US Army Research Laboratory, report. Available at: http://www. arl. army. mil/sedd/acoustics/reports. htm*, 1997.

[92] Hariharan Rahul, Swarun Kumar, and Dina Katabi. MegaMIMO: Scaling Wireless Capacity with User Demands. In *ACM SIGCOMM 2012*, Helsinki, Finland, August 2012.

[93] S.V. Rakovic, E.C. Kerrigan, D.Q. Mayne, and J. Lygeros. Reachability analysis of discrete-time systems with disturbances. *Automatic Control, IEEE Transactions on*, 51(4):546–561, 2006.

[94] Ketan Savla, Giuseppe Notarstefano, and Francesco Bullo. Maintaining limited-range connectivity among second-order agents. *SIAM Journal on Control and Optimization*, 48(1), 2009.

[95] F. Schlaepfer and F. Schweppe. Continuous-time state estimation under disturbances bounded by convex sets. *Automatic Control, IEEE Transactions on*, 17(2):197–205, 1972.

[96] R.O. Schmidt. Multiple emitter location and signal parameter estimation. *Antennas and Propagation, IEEE Transactions on*, 34(3):276–280, 1986.

[97] M. Schuresko and J. Cortes. Distributed tree rearrangements for reachability and robust connectivity. *Intell. Robot. Syst.*, 56(1-2):99–126, sept. 2009.

[98] M Schwager, B Julian, and D Rus. Optimal coverage for multiple hovering robots with downward-facing cameras. In *Robotics and Automation,Proc of International Conference on*, 2009.

[99] F.C. Schweppe. Recursive state estimation: Unknown but bounded errors and system inputs. *Automatic Control, IEEE Transactions on*, 13(1):22–28, 1968.

[100] J.C. Spall. Adaptive stochastic approximation by the simultaneous perturbation method. *Automatic Control, IEEE Transactions on*, 2000.

[101] D. Spanos and R. Murray. Motion planning with wireless network constraints. In *Proceedings of the ACC*, 2005.

[102] Petre Stoica and Randolph L. Moses. *Spectral Analysis of Signals*. Prentice Hall, 2005.

[103] Petre Stoica and randolph moses. *Spectral analysis of signals*. Prentice Hall, 2005.

[104] William C. Stone. *Electromagnetic signal attenuation in construction materials*. U.S. Dept of Commerce, Technology Administration, National Institute of Standards and Technology, 1997.

[105] E Stump, A Jadbabaie, and V Kumar. Connectivity management in mobile robot teams. In *Robotics and Automation, IEEE International Conference on*, pages 1525–1530, 2008.

[106] A. Tahbaz-Salehi and A. Jadbabaie. On recurrence of graph connectivity in vicsek's model of motion coordination for mobile autonomous agents. In *American Control Conference, 2007. ACC '07*, pages 699 –704, july 2007.

[107] O. Tekdas, W. Yang, and V. Isler. Robotic routers: Algorithms and implementation. *Int. Journal of Robotics Research*, 29(1), 2010.

[108] Jeffrey N. Twigg, Jonathan Fink, Paul L. Yu, and Brian M. Sadler. Efficient base station connectivity area discovery. 2013.

[109] Jeffrey N Twigg, Jonathan R Fink, PL Yu, and Brian M Sadler. Rss gradient-assisted frontier exploration and radio source localization. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 889–895. IEEE, 2012.

[110] B.D. Van Veen and K.M. Buckley. Beamforming: a versatile approach to spatial filtering. *ASSP Magazine, IEEE*, 5(2):4–24, 1988.

[111] Vijay V. Vazirani. *Approximation Algorithms*. Springer, 2002.

[112] Marcos A. M. Vieira, Matthew E. Taylor, Prateek Tandon, Manish Jain, Ramesh Govindan, Gaurav S. Sukhatme, and Milind Tambe. Mitigating multi-path fading in a mobile mesh network. *Ad Hoc Netw.*, 11(4), June 2013.

[113] Jue Wang, Fadel Adib, Ross Knepper, Dina Katabi, and Daniela Rus. RF-Compass: Robot object manipulation using rfids. In *MobiCom*, 2013.

[114] Jue Wang and Dina Katabi. Dude, where's my card? RFID positioning that works with multipath and non-line of sight. In *SIGCOMM*, 2013.

[115] Jie Xiong and Kyle Jamieson. Arraytrack: a fine-grained indoor location system. In *NSDI*, 2013.

[116] Yuan Yan and Y. Mostofi. Co-optimization of communication and motion planning of a robotic operation under resource constraints and in fading environments. *Wireless Communications, IEEE Transactions on*, 12(4):1562–1572, 2013.

[117] Yuan Yan and Yasamin Mostofi. Co-optimization of communication and motion planning of a robotic operation under resource constraints and in fading environments. *IEEE Transactions on Wireless Communications*, 12(4), 2013.

[118] Seung-Kook Yun and Daniela Rus. Optimal distributed planning of multi-robot placement on a 3d truss. In *Intelligent Robots and Systems,Proc of IEEE International Conference on*, 2007.

[119] Howard A. Zebker and Richard M. Goldstein. Topographic mapping from interferometric synthetic aperture radar observations. *Journal of Geophysical Research: Solid Earth*, 91(B5):4993–4999, 1986.

[120] Qing Zhao and B.M. Sadler. A survey of dynamic spectrum access. *Signal Processing Magazine, IEEE*, 24(3):79–89, 2007.