Summer 8-12-2011

Communication for Teams of Networked Robots

Jonathan Fink
University of Pennsylvania, jonfink@grasp.upenn.edu

Communication for Teams of Networked Robots

Abstract

There are a large class of problems, from search and rescue to environmental monitoring, that can benefit from teams of mobile robots in environments where there is no existing infrastructure for inter-agent communication. We seek to address the problems necessary for a team of small, low-power, low-cost robots to deploy in such a way that they can dynamically provide their own multi-hop communication network. To do so, we formulate a situational awareness problem statement that specifies both the physical task and end-to-end communication rates that must be maintained. In pursuit of a solution to this problem, we address topics ranging from the modeling of point-to-point wireless communication to mobility control for connectivity maintenance. Since our focus is on developing solutions to these problems that can be experimentally verified, we also detail the design and implantation of a decentralized testbed for multi-robot research. Experiments on this testbed allow us to determine data-driven models for point-to-point wireless channel prediction, test relative signal-strength-based localization methods, and to verify that our algorithms for mobility control maintain the desired instantaneous rates when routing through the wireless network. The tools we develop are integral to the fielding of teams of robots with robust wireless network capabilities.

Degree Type

Dissertation

Degree Name

Doctor of Philosophy (PhD)

Graduate Group

Electrical & Systems Engineering

First Advisor

Vijay Kumar

Keywords

robotics, networking, coordinated control, communication modeling, multi-robot systems

Subject Categories

Robotics

COMMUNICATION FOR TEAMS OF NETWORKED ROBOTS

Jonathan Fink

A DISSERTATION

in

Electrical and Systems Engineering

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

2011

Vijay Kumar, *Professor of Mechanical Engineering and Applied Mechanics*
Supervisor of Dissertation

Roch Guerin, *Professor of Electrical and Systems Engineering*
Graduate Group Chairperson

Dissertation Committee:

Dr. Vijay Kumar, *Professor of Mechanical Engineering and Applied Mechanics*

Dr. Ali Jadbabaie, *Associate Professor of Electrical and Systems Engineering*

Dr. Alejandro Ribeiro, *Assistant Professor of Electrical and Systems Engineering*

Dr. Brian Sadler, *Army Research Laboratory*

Communication for Teams of Networked Robots

Acknowledgements

There are a number of people who I would like to thank for their contributions to the completion of this thesis. First and foremost, I would like to thank my advisor, Vijay Kumar, for the nearly limitless opportunities while always being able to connect my ideas to the "big picture" and for the countless discussions that transformed impenetrable concepts into well-ordered thoughts. I also wish to thank my committee for their time and advice which has collectively enhanced my knowledge and understanding. In particular, I would like to thank Alejandro Ribeiro for numerous discussions, insights, and advice.

Many people were instrumental to the experimental work that became a significant focus of my thesis. I would like to specifically recognize Ben Charrow, Alex Kushleyev, Christian Moore, Pedro Shiroma, Mike Shomin, and Ethan Stump. Additionally, I would like to thank collaborators Dave Cappelleri, Peng Cheng, Jason Derenick, and Ani Hsieh. Special thanks go to Nathan Michael who contributed more than his share of late nights, forward-looking vision, and ideas.

Finally, I would not have made it to this point without the unconditional love and support of my wife and parents. To them, I will be eternally grateful.

ABSTRACT

COMMUNICATION FOR TEAMS OF NETWORKED ROBOTS

Jonathan Fink

Vijay Kumar

There are a large class of problems, from search and rescue to environmental monitoring, that can benefit from teams of mobile robots in environments where there is no existing infrastructure for inter-agent communication. We seek to address the problems necessary for a team of small, low-power, low-cost robots to deploy in such a way that they can dynamically provide their own multi-hop communication network. To do so, we formulate a situational awareness problem statement that specifies both the physical task and end-to-end communication rates that must be maintained. In pursuit of a solution to this problem, we address topics ranging from the modeling of point-to-point wireless communication to mobility control for connectivity maintenance. Since our focus is on developing solutions to these problems that can be experimentally verified, we also detail the design and implantation of a decentralized testbed for multi-robot research. Experiments on this testbed allow us to determine data-driven models for point-to-point wireless channel prediction, test relative signal-strength-based localization methods, and to verify that our algorithms for mobility control maintain the desired instantaneous rates when routing through the wireless network. The tools we develop are integral to the fielding of teams of robots with robust wireless network capabilities.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this thesis, we address the problem of communication for networked teams of robots in a comprehensive way. This entails a unified examination of many problems that are typically considered separately including system architectures, wireless communication modeling, network routing, and coordinated motion planning. We focus on communication because, in the same way that joint kinematics or obstacle avoidance are fundamental constraints of single robot systems, communication is a fundamental constraint for teams of robots. It enables dynamic point-to-point communication infrastructure, an external command structure, and coordinated algorithms for control, estimation and inference. In other words, the communication capability of a team of robots is essential even for applications where it is not the primary objective.

The potential application of teams of robots to search and rescue applications is well accepted [17, 46]. Teams of low-cost, low-power, expendable robots are able to distribute themselves through disaster sites in place of human rescuers in order to locate survivors and identify dangerous locations. In the most general case, flows of sensory data from each robot must be aggregated at a fixed operation center outside the disaster area. A specialization of this application is the flow of sensory and control data necessary to teleoperate a single robot while the team act as communication relays. To accomplish these types of task, the robots must be able to estimate point-to-point communication capabilities in the environment and position themselves such that the necessary flows of data can be accommodated by their wireless network.

The primary challenge facing the design and implementation of such a system is that point-to-point radio communication is notoriously hard to predict. The variability of its performance makes it difficult to guarantee a certain rate of communication over a single link and only becomes more challenging when considering a network of links. Furthermore, gathering enough measurements to adequately characterize communication performance for any point-to-point link in a new environment is prohibitive. A secondary independent challenge of this work is that even when given an approximate prediction of point-to-point communication capability, planning coordinated motions of many robotic agents such that communication flows are maintained is a complex undertaking.

The primary challenge of uncertain point-to-point radio communication motivates the need for experimental verification. Based on our assumption that communication will be inherently unpredictable, we cannot argue that theoretical analysis or simulation results will be adequate to verify the success or performance of any solution to the communication problems faced by teams of networked robots. Consequently, in this thesis we also address the development of an experimental testbed for networked teams of robots. In doing so, we demonstrate how the properties of an experimental testbed affect the type of system architecture that can be verified. This forces us to focus on approaches to solve the communication problems for a team of robots that can be implemented and verified experimentally on our testbed. Finally, since our focus is on communication issues, we will assume the primary objective for the team, e.g. sensing or data fusion, are externally available but beyond the scope of this work.

1.1 Objective

The focus of this thesis is presented in two pieces. The first objective is to consider the design of system architectures and the implications this has on the development of an experimental testbed to investigate the wide range of research problems underly multi-robot coordination and cooperation. We detail the design criteria and evolution of a system necessary to support a large class of robots and increasing size of the operating environment. A large part of testbed design includes the evaluation and implementation of control and localization techniques for individual robots. In later experiments, this allows us to assume basic individual robot functionality as an available technology.

The second objective of this thesis is to address the communication problems faced by teams of networked robots. We begin by formulating a motivating problem statement for situational awareness and a solution that addresses the problem through the development of two core capabilities. The first capability is a system to provide statistical predictions of point-to-point radio communication. By analyzing experimental measurements, we evaluate several candidate models and develop this capability. The second capability necessary is a suite of algorithms that rely on the point-to-point communication predictions to determine how robots should move to accomplish their high-level task specification while maintaining the fundamental communication constraints. Our treatment of the communication problems faced by networked robots concludes with experimental results verifying that our approach succeeds in solving our situational awareness problem statement.

1.2 Literature Review

As is typical in robotics problems, the relevant literature extends across several fields. In the case of communication issues for robotic networks, we draw on results from radio communication, wireless networking, control theory, and robotic motion planning literature. We begin by considering the problem of modeling point-to-point radio communication. This leads us to a short tangent into the area of relative localization based on received signal strength. These ideas can be used to provide robotic networks with additional location information in unknown environments. Next, we survey recent results in the field of wireless networking, i.e. determining how to route messages from source to destination through a network with dynamically changing link topology. Finally, we examine results on coordinated control and connectivity for multi-agent systems. While much of this work relies on simplifying abstractions about the nature of the communication channels, it provides a strong theoretical basis for the development of architectures and algorithms.

1.2.1 Point-to-point Radio Communication

The basis for arbitrary communication in networked teams of robots is point-to-point radio communication. That is, the use of radio frequency transceivers to communicate control, coordination, sensing, and other data. It is well understood the capability of a wireless communication channel is dependent on the signal-to-noise ratio at the receiver [93]. Consequently, the problem of modeling and predicting the propagation of a radio signal through the environment is well studied. We are most interested in the literature that considers indoor environments and the application of models to problems with inherent node mobility.

Radio signal propagation is a complex, multi-scale process that occurs at a number of length scales. At the coarsest scale, radio signal power decays with increasing distance from the transmitter. At a medium scale, obstacles in the environment can lead to shadowing, i.e. signal power attenuation due to the obstacle. The so-called phenomena of small-scale fading occurs due to multi-path effects. That is, the radio signal reflects and refracts off of scatterers in the environment causing several “copies” of the signal to arrive at the receiver at approximately the same time with dramatically varied power characteristics. In fact, small movements in the scatterers, transmitter, or receiver can result in large changes in the received signal strength [86].

Models in the literature can be classified based on the computation of radio signal path as direct path, dominant path, or ray-tracing. The class of direct path models have the highest level of adoption due to the simplicity of their implementation. All of these models are grounded in the concept of a distance dependent path loss. However, these models are empirical and require training based on experimental measurements that may lead to varied parameterizations from environment to environment. For indoor models, the recognition that walls generate non-smooth attenuation of signal power is incorporated by

Damosso et al. in the multi-wall model that assumes a fixed attenuation, on a log scale, for each wall between the source and receiver [14]. There are several additions to this model that consider the effects of wall type [48] and thickness [58].

Based on the experimental observation that radio propagation seems to occur in two distinct regions, Cheung et al. [11] propose a direct path model that relies on two unique path-loss exponents for short-range and longer-range channels. Extending their model even further, they introduce an angle dependence to the signal attenuation attributed to obstacles in the environment and allow for one level of signal diffraction at corners in the environment.

A practical limitation of direct path methods is that they offer limited modeling of the medium-scale effects and no modeling of the small-scale fading effects. For robotic teams, accurate prediction of received signal strength and communication channel capability requires consideration of these effects since the nodes are operating in possibly cluttered environments and are mobile [74]. Fortunately, it has been observed that small-scale fading for particular environments can be approximated as a random variable with log-normal [55], Rician [62], or Rayleigh [12] distribution.

The class of ray-tracing models consider a large number of rays sent in all directions by the transmitter [67]. At each interaction with obstacles in the environment, i.e. transmission, reflection, refraction, the number of rays increases exponentially. As each rays interacts with the environment, signal attenuation is computed based on path loss and dielectric properties of the interacting obstacles in the environment. The set of rays that arrive at the receiver enumerate the multi-path effect and provide a more accurate portrayal of the small-scale fading phenomenon. Though the naive implementation of ray-tracing methods leads to exponential complexity, there are methods for efficient approximation, e.g. Gorce et al. [34] introduce a multi-resolution frequency domain ParFlow (MR-FDPF) approach for fast simulation of radio wave propagation in indoor environments.

While ray-tracing methods have the capability to accurately model radio signal propagation at several length scales, they are fragile in their dependence on precise models of the environment that include material properties such as the dielectric constant of each obstacle. Wölfle et al. [107, 109] propose the dominant path class of models that seek to efficiently predict the sequence of ray interactions that leads to the primary component of the signal power arriving at the receiver. Dominant paths are inferred by training of neural network prediction models on coarse geometric maps of the environment and path loss along the dominant path is based on known properties of the environment. They demonstrate that this paradigm outperforms both direct-path and ray-tracing models in terms of accuracy and efficiency [110].

Separate to the issue of model selection is the problem of building accurate indoor databases of received signal strength. Wahl et al. [103] address this issue by examining the inaccuracies for an indoor environment with two accurate models – one fast ray-tracing and one dominant path model. Specifically, they enumerate a number of factors that lead to inaccuracies in building databases, e.g. time varying obstacles and adjacent building structures. Neskovic et al. [76] also provide a survey of many radio propagation models and note that even temporal effects in indoor environments due to people moving through the environment can cause significant changes in received signal strength. They summarize that do date a perfect system has not been found for the prediction of radio signal strength – each has its advantages and disadvantages.

The advances in radio signal propagation modeling has had an effect on both the wireless sensor network and mobile robotics communities. Indeed, the wireless sensor network community has recognized the need to incorporate more accurate propagation models into simulation environments [36, 96]. Furthermore, both communities have turned to received signal strength as a sensory input that can be used to provide localization services. This is, in part, due to several commercial systems that offer real-time location information for warehouse and hospital environments, e.g. [22, 3, 2].

Localization techniques for wireless sensor networks can be broken into two categories – range-based and range-free methods. Awad et al. [6] examine the application of received signal strength indicators to network-centric localization of a wireless sensor network.

Specifically, they focus on methods for estimating distance from received signal strength and the challenges that arise – namely the transmit power, node orientation, and quality of reference measurements. They demonstrate that proper selection of transmission power can increase the accuracy of distance estimation. An alternate approach, presented by Yedvalli et al. [112], ignores the idea of range inference from signal strength and instead relies on the assumption that an ordering of received signal strength of point-to-point links predicts the distance-based order of those same links. By comparative analysis of their method with four state-of-the-art localization methods, they demonstrate improved performance over a range of channel conditions and node deployments.

In the robotics community, assumptions for received signal strength-based localization typically involve mobile nodes on a smaller scale than wireless sensor networks. The advantage of controlled node mobility is that a time series of measurements can help refine the location estimate. However, a theme here is that model-based approaches cannot provide adequate accuracy for mobile robot localization. Instead, most methods rely on a priori measurement campaigns at reference locations followed by probabilistic filtering to estimate mobile robot localization [53, 37]. These methods are forced to make discretizing assumptions based on the location of reference measurements while Ferris et al. [25] adopt a model-free approach by training a Gaussian process on a set of a priori measurements in the environment. In doing so, they are able to accurately model the spatial process of received signal strength in a probabilistically correct way. Recently, Zickler et al. [117] have demonstrated that a range-based approach can succeed for mobile robot localization with a small number of nodes. They pursue the idea of a data-driven probabilistic model that maps received signal strength to inter-node distance distributions. When embedded in a grid-based localization algorithm, this model effectively enables localization of a moving node.

The development of radio signal propagation models does not directly address the mobile robotics problem of mapping received signal strength in a partially unknown environment. A common robotics application requires the nodes to deploy into a new environment with limited a priori knowledge of radio signal strength propagation and learn the necessary features of that environment during operation. This presents a challenging problem. Indeed, even for environments where the received signal strength model has been learned, temporal variations may quickly invalidate that model, e.g. [113]. Furthermore, the task of capturing point-to-point training data for even a small environment can be prohibitive.

Nonetheless, there are limited examples of how signal strength maps can be generated in an online fashion by mobile robots. For example, Hsieh et al. [41] propose a method that relies on a convex cell decomposition of the environment. Within each cell, they assume that the line-of-sight channel can be predicted with reasonable accuracy based on distance-based signal transmission models. A finite set of point-to-point signal strength measurements are then made to determine signal strength between convex cells. Much of the work focuses on determining sampling plans that visit the pairs of positions in an efficient manner. Mostofi et al. [75] propose an application of compressed sensing to the mapping of signal strength with minimal sensing. Their method allows for the reconstruction of a signal strength map based only on sparse measurements.

1.2.2 Wireless Networking

The entire field of wireless networking is well studied and beyond the scope of this thesis. Here we examine a slice of that literature that provides insight to the problems faced by networked robot teams and an idea of the research trajectory that brings us to our solutions. The main issues of wireless networking arrise directly from the challenges introduced by point-to-point radio communication as described above. The inherent variability in the performance of radio communication degrades the performance of traditional networking algorithms developed for wired systems. When applied to robotic networks, where a majority of the nodes are mobile, traditional methods become entirely unsuitable as even the basic topology of the network becomes dynamic.

Early solutions to the problem of changing network topology include Ad Hoc On Demand Vector routing (AODV) [83] and Dynamic Source Routing (DSR) [45]. In both of these approaches realtime measurements are collected to determine the connectivity of the network and the algorithm determines how messages should be routed based on a minimum hop-count metric. That is, both algorithms seek to minimize the number of the links necessary to send a message from source to destination. Problems quickly arrise when these algorithms are applied to real-world dynamic networks.

Most of these problems can be reduced to the concept that the existence of a link in a wireless network is not a binary quantity. Lundgren et al. [64] first observe that the problem of detecting link existence is not straight forward. They propose the concept of a communication grey zone where link performance has degraded to a point where reception of control packets used to identify link existence is possible but reception of data packets is not. The difference in reception arises from a number of reasons, e.g. transmission rate and packet size. They propose a method whereby each possible communication link is evaluated according to a signal quality threshold before it is included in the network routing algorithm.

While eliminating communication links based on a signal quality threshold maintains performance in real world networks, it may lead to a disconnected system where nodes that could otherwise communicate at low rates are unreachable. An alternate approach, presented by De Couto et al. [18], defines the expected transmission count metric as a replacement for the minimum hop-count metric typically used when selecting routes in a dynamic wireless network. In essence, this approach avoids the false notion that hop count is a good predictor for route performance. While De Couto et al. experimentally demonstrate improved performance, Stojmenovic et al. [98] examine similar metrics based on expected hop count and the probability of delivery to develop algorithms with theoretical guarantees. The common conclusion from these studies and others is that route choice should be based on a quality metric rather than satisfaction of a threshold.

A recent thread of the literature eschews the idea of a single deterministic route for messages in an uncertain wireless network. Instead, Ribeiro et al. [89] present a framework where each link in the network is defined as a probability of successful communication. This allows them to develop stochastic routing protocols where packets are routed randomly according to probabilities that are the result of properly defined optimality criteria. Wu et al. [111] extend this work to model uncertain link reliabilities based on their mean and variance which allows for robust routing algorithms that can be formulated as convex optimization programs. By taking into account the statistical properties of the link capabilities, this approach satisfies optimality criteria in spite of link uncertainty. Furthermore, it is demonstrated that these optimization problems have a separable structure and can be computed in a distributed way through communication with one-hop neighbors. These probabilistic approaches are particularly interesting for robotics applications where node movement prevents accurate estimation of point-to-point communication capability.

1.2.3 Coordinated Control

Coordinated control refers to the ability of a team or group of robots to work together in order to accomplish an application or task. Typical coordination tasks for a team of mobile robots include formation control, cooperative manipulation, and sensor fusion. However, a large class of coordination problem can be considered as the general consensus problem. Accordingly, there has been a large amount of effort devoted to understanding the algorithms and network conditions under which consensus can be achieved. Nearly all of this work relies on a coarse abstraction for network communication – that links are a binary quantity where connections can be considered available or not. While this abstraction may seem to be an oversimplification based on the wireless communication literature reviewed above, it does allow some strong graph theoretic statements to be made.

The generic consensus problem is defined such that all agents of a team must independently determine the value of some global quantity based on interactions with other agents in the team. The precise form of interaction is generally abstracted and can be indirect, i.e. each node senses the state of its neighbors, or more explicit, i.e. exchange of data through a wireless communication channel. Much of this work can be traced back to that of Jadbabaie et al. [43] where they provide a theoretical explanation for flocking models when each agent only has knowledge of its heading and the headings of its current neighbors. They prove that the described system converges, i.e. achieves consensus, despite changes to each node's neighbor set so long as a union of the collection of interaction graphs is connected frequently enough. Similar results are extended by Ren et al. [88] for directed graphs where the condition for convergence is that a spanning tree exist frequently enough.

This work is extended even further by Olfati-Saber et al. [79] when they consider a large set of consensus problems with fixed or switching networks, time delays, and directed or undirected information flow. A primary contribution of this work is the establishment of the connection between algebraic connectivity, also know as the Fiedler eigenvalue, and the performance of linear consensus protocols. They additionally show that there is a tradeoff between robustness to time-delays and the performance of linear consensus protocols. Olfati-Saber et al. [78] go on to provide an overview on much of the theory and applications of consensus problems in networked systems. Most importantly they demonstrate that a number of seemingly different consensus algorithms are, in fact, closely related.

Martínez et al. [65, 66] diverge from analysis of consensus problems to develop a theory of time and communication complexity for motion coordination algorithms.. They do this by formalizing the notions of a robotic network with control and communication laws. They define the time complexity of a control and communication law to be the minium number of rounds of communication necessary to achieve the specified task. Using their formalization, they are able to analyze several well-known coordination algorithms and provide lower and upper bounds on their complexity.

We present this slice of the coordinated control literature to demonstrate the amount of effort that has been put into analyzing interaction systems. Indeed, this body of work has been able to make remarkable guarantees about the convergence and performance of consensus-type algorithms that act on interaction graphs as an abstraction for sensing or wireless communication. Consequently, there is an accordingly large segment of the coordinated control literature that addresses the problem of connectivity maintenance. That is, the problem of completing a primary task, be it rendezvous, sensing, or target tracking, under the constraint that the communication network must remain connected. Solutions to this problem can be classified based on the model used for point-to-point communication.

The first model, and perhaps most widely considered, is the disk model. This implies that communication capability can be directly inferred by proximity, e.g. if two nodes are less than a distance R from each other, communication is possible. Examining the models for wireless communication, it is easy to see the attractiveness of this as a conservative approximation of connectedness since communication is often quite reliable if two nodes are close enough. Most importantly, the disk-model allows for an easy geometric construction of the communication graph. Notarstefanno et al. [77] introduce a result for a system of double integrators that transforms the desired controls for each agent to find the closest input that maintains connectivity. They do this in a distributed fashion by developing the concept of a double integrator disk graph an employing distributed flow-control algorithms. More recently, Chakraborty et al. [8] formulate a convex optimization problem to maintain k-connectivity during network reconfiguration. The major contribution of this work is the development of a distributed incremental algorithm that solves their problem with local information and improved scalability over the centralized problem.

Spanos et al. [95] take an alternate approach to most of the literature reviewed here by developing methods for distributed motion planning that take connectivity constraints into account. By introducing the concept of the connectivity robustness of a network, i.e. a quantification of the freedom of individual vehicles to undergo arbitrary motions without disconnecting the network, they are able to show that distributed motion planning methods are able to efficiently solve the problem. Unlike many methods which seek to analytically construct controls, they rely on algorithmic search to produce optimal controls subject to obstacle-avoidance and communication constraints.

The second class of model-based connectivity maintenance algorithms extend the disk model by assuming some sort of smooth distance-based decay of connectivity. The primary result of this extension is that they gain the ability to analytically evaluate gradient information about connectivity constraints. Another common feature of these approaches is the use of the second smallest eigenvalue, the Fiedler value, of the graph Laplacian as a measure for connectivity of the wireless network. DeGennaro et al. [19] use a model of exponential decay for point-to-point communication and propose a decentralized algorithm for eigenvector computation that yields the desired distance for each link. A potential-based control law then drives the system toward the set-point distances that increase the connectivity. Ji et al. [44] extend this work by considering a class of problems where the team has a primary formation control style objective that must be achieved while maintaining connectivity. The problem is further refined by Zavlanos et al. [115] who formulate the connectivity requirement as a differentiable constraint on individual agent motion based on the dynamics of the Laplacian matrix. This allows for a leader-follower application where follower nodes move based on gradient information to maintain connectivity.

A limitation of the above approaches that leverage the graph Laplacian and act to increase algebraic connectivity is that they ignore the discrete nature of the network structure. Zavlanos et al. [116, 70] address this limitation by proposing a hybrid systems approach where a continuous controller maintains existing links while a discrete controller operates to delete links unnecessary for connectivity by means of a gossip algorithm. They demonstrate the effectiveness of their approach in both simulation and experiments.

Stump et al. [99] continue the work of [19] for an environment with obstacles. They do this by incorporating obstacle–based constraints and casting the resulting problem as a convex optimization that is solved at each time step. By doing so, they are able to demonstrate a reactive controller that can maintain distance-based connectivity in an environment with complex obstacles.

Schuresko et al. [92] take an alternate approach to develop algorithms that determine if a collective action will break connectivity and modifying that action to guarantee that connectivity is maintained. Like many of the methods above, they rely on the use of the second smallest eigenvalue of the graph Laplacian as a metric for connectivity but are able to deal with outdated information due to delays in communication and robot motion. They accomplish this with a game theoretic solution approach. Doing so frees them from many of the analytic restrictions on the mapping from inter-robot distance to edge weight in the connectivity graph, i.e. non-convex mappings are allowable.

A third model-based class of connectivity algorithms avoid the use of proximity as a predictor for communication capability. Instead, they make the assumption that a line-of-sight path between two nodes is a good predictor of a reliable communication channel. Indeed, some of the models from the radio communication literature do support this assumption. In particular, Stump [100] develops considers the problem of maintaining a line-of-sight chain from a fixed node to a single moving node under external control. He solves this problem by an adaptation of [31] and the application of Steiner trees to the visibility maintenance problem to compute deployments that provide a minimum length mutual visibility sequence to the moving node.

There is a growing segment of the literature that rely on empirical measurements or stochastic models of communication capability that can handle the small-scale fading phenomena that occur in most realistic environments and disrupt many implementations of the literature referenced above. Hsieh et al. [38, 40] avoid the issue of predicting the relationship between node positions and communication capability by measuring the local gradient of each channel's performance and computing controls accordingly. They are able to demonstrate that for simple communication topologies, their controller maintains connectivity to a base station in a real environment. Lindhé et al. [61] pursue a parallel problem to the connectivity maintenance problem – they seek to exploit small-scale fading to locally improve communication performance. They accomplish this by developing analysis for the optimal sampling necessary in order to determine if small local movements can reliably improve the performance of a communication link. The approach is experimentally verified but does assume that nodes are stationary and that multi-path effects are slow to change in the environment.

In the radio communication field, small-scale fading is often modeled as a random variable. Mostofi [73] explicitly considers the random fading of wireless channels and identifies the impact this fading has on decentralized mobile networks. She demonstrates a characterization of communication noise and the variance on that noise as a function of the signal-to-noise ratio and proposes a random motion planning strategy that can drive agents out of highly correlated deep fades. Ghaffarkhah et al. [32] continue this approach by addressing a specific target tracking problem where measurements of a moving target are communicated back to a base station. The positions of the robots observing the target are constrained by the quality of communication to the base station which is estimated in a realistic way that includes random fading.

Similar to Hsieh et al., Le Ny et al. [56] avoid specific models for the prediction of communication performance and instead propose an approach based on general gradient descent frameworks when the objective function is based on uncertain or dynamic values. By turning to stochastic gradient descent algorithms, they demonstrate a source seeking behavior with realistic stochastic wireless connectivity constraints that are incorporated into the random objective function.

Finally, all of the previous treatments of the wireless connectivity problem either rely on simplified communication models and use algebraic connectivity as an abstraction for the wireless network performance or rely on more accurate point-to-point communication models while avoiding the difficult issues of network topology and end-to-end connectivity. Zavlanos et al. [114] rely on deterministic functions to predict point-to-point reliability but they also make the leap that occurred in the wireless networking literature to consider the end-to-end performance of the network. In other words, their solution occurs in the joint space of network routing and node mobility. They accomplish this with a hybrid control approach where continuous motion controllers based on potential fields interact with discrete optimization of the communication variables that control network routing. Rather than consider topological metrics, communication integrity is based on the ability of the network to support desired end-to-end communication rates.

1.3 Approach

The objective of this thesis is to present and experimentally verify a system that can address communication issues for a team of networked robots. This problem is clearly divided into two pieces: first, the development of system architectures and experimental testbed design and second, the algorithms that make up our solution to a particular instance of the communication connectivity problem for a team of mobile agents.

In the first part, we begin by classifying the choice of system architecture according to the communication abstraction – nearest-neighbor, centralized, or multi-hop. The choice of system architecture presents implications for both the target deployment system as well as the experimental testbed for verification. We continue by describing the evolution of our experimental testbed from a centralized model with the ability to emulate nearest-neighbor algorithms to a fully decentralized model that can accurately implement nearest-neighbor or multi-hop algorithms. We demonstrate the effectiveness of a centralized testbed with case study on the verification of a nearest-neighbor algorithm for multi-robot manipulation.

To address the second piece, we formulate a situational awareness problem statement with a proposed solution that relies on two subsystems – one that provides point-to-point communication predictions and another that controls robot motions to maintain communication. To generate point-to-point communication predictions, we compare the performance of several direct-path models based on [14, 48, 11] and an extension of the Gaussian process approach taken by Ferris et al. [25].

In order to determine robot motions that satisfy a task specification while maintaining communication constraints, we propose a novel approach that unifies wireless networking and coordinated control. The wireless networking component is based on the premise of considering the end-to-end performance as in [18] and the stochastic routing as in [111]. By adopting a stochastic routing scheme, we are able to compute network solutions that are robust to link uncertainty. The coordinated control component of our algorithm is most closely related to [114]. By considering the problems jointly, we are able to choose solutions to the networking problem that maximize a utility function related to each node's freedom to move. Conversely, node motions are constrained such that they will not invalidate the current network solution. We present both local gradient-based methods similar to the existing literature as well as global planning-based methods that are able to operate in complex environments with obstacles.

Since the underlying point-to-point communication performance is inherently variable, we conclude with a series of experiments to verify our solution to the situational awareness problem statement. We conduct experiments in multiple environments and demonstrate that instantaneous network performance matches or exceeds the task specifications.

1.4 Thesis Contributions

This thesis makes several novel contributions to the communication problems faced by teams of networked robots.

First, we reformulate the traditional connectivity maintenance problem to incorporate noisy point-to-point channels and support quality-of-service specifications for a team of robots. This is done in a robust way by translating quality-of-service specifications into end-to-end rates that must be maintained with a certain probability in the presence of uncertain wireless communication. By defining connectivity based on on precise quality-of-service specifications rather than abstractions for connectivity based on graph theory, we are able design solutions that efficiently satisfy problem requirements when implemented on a team of mobile robots.

Second, in order to predict the achievable end-to-end rates across the network, we develop probabilistic models for the point-to-point communication rate between pairs of robots. These models can be refined as robots explore new regions of the environment. The key feature of our approach is that it models local changes in the point-to-point communication capability and is able to represent uncertainty about predictions for unexplored regions of the environment.

Third, we explore the use of robot mobility to enable and improve communication quality-of-service. Using our probabilistic model for connectivity, we develop local controllers that move agents to satisfy task specifications while optimizing connectivity and performance of the network in an open environment. Furthermore, we present a global planning algorithm that searches the joint configuration space of a team of robots to find a sequence of network deployments that satisfy task specifications for complex environments with obstacles. By employing a global planning strategy, we are able to operate in complex real-world environments in order to satisfy a large class of multi-robot tasks that are subject to communication constraints.

Fourth, through the design and implementation of a decentralized experimental testbed, we verify that a team of robots can sample point-to-point signal strength measurements in an environment, learn a probabilistic model of point-to-point communication, and make useful predictions about the performance of wireless channels in arbitrary configurations. More importantly, we demonstrate experimental verification that our algorithms are able to maintain connectivity and desired quality-of-service specifications with low-power radios in a real indoor environment. We believe our approach is the first to rely on concurrent design of routing protocols and control policies in order to address connectivity maintenance problem. Consequently, it is also the first system with experimental verification of these capabilities.

Finally, our experimental testbed supports not only the experimental verification of our approach to the connectivity maintenance problems but many other investigations in the area of multi-robot systems [30, 29, 70, 20, 9, 71, 100]. Its capabilities and flexibility demonstrate the impact our testbed has on experimental verification of multi-robot systems.

1.5 Thesis Outline

The remainder of this thesis continues as follows.

Chapter 2 begins by presenting a classification of coordination algorithms for teams of robots based on their communication requirements. These requirements influence the necessary capabilities of the experimental testbed. We finish this chapter by detailing the evolution of our experimental testbed and the types of problems it is suitable for studying.

Chapter 3 presents two example problem statements targeted for a networked team of robots. The first is a nearest-neighbor algorithm for multi-robot manipulation. The second is a generic situational awareness problem statement that serves as a motivating example for the remainder of this thesis. It formalizes our task representation and the structure of our communication requirements. This chapter finishes with a system architecture outlining our proposed solution to the situational awareness problem.

Chapter 4 details several candidate models for predicting point-to-point received signal strength by a team of robots. After experimentally analyzing the candidate models, we make some recommendations to their applicability. This chapter also includes some supplementary material towards the estimation of relative node location based on received signal strength.

Chapter 5 presents the central algorithmic pieces of this thesis as we present a unified way to consider robust network routing and constrained node mobility. We begin with a local gradient-based controller, demonstrate the local minima that lead to its shortcomings, and conclude with a global planner that allows for flexible planning of network configurations in complex environments.

Chapter 6 combines the results of Chapters 2 – 5 to present a set of experiments on our decentralized testbed that demonstrate our solution to the situational awareness problem with an experimental system. We go on to provide results that verify the performance of our solution to the communication problem based on real-world instantaneous measurements of the underlying wireless channels.

Chapter 7 finishes with some concluding remarks and thoughts for future work.

Chapter 2

System Design and Architecture

We are interested in comprehensively addressing the components necessary for a team of networked robots to accomplish their assigned task. Design goals that are central to our interests include experimental verification, the ability to compose individual subsystems, and scalability in terms of team size, capability, and operating region. In this chapter, we begin by discussing approaches and abstractions for problems of multi-agent communication and coordination. We continue by examining the design of an experimental architecture. Finally, we conclude with a description of two approaches to experimental testbed design that allow for robust verification of a wide variety of solutions to multi-agent problems.

2.1 Communication & Coordination

For robotics applications, the coordination of agents is closely tied to their communication. As a result, abstractions for inter-agent communication are a necessary first step in the design of coordinated algorithms. Furthermore, the choice of abstraction can affect the transition to deployed systems as well as scaling to complex environments or larger teams.

Architectures for communication and coordination can be separated into three groups:

- *centralized* systems where information is globally shared and it is assumed every agent can communicate with every other,

- *multi-hop* systems where direct communication is only allowed with agents belonging to a small neighborhood but multi-hop routing can pass messages across a network of agents, and

- *nearest-neighbor* systems where communication is limited to a single hop and typically confined to a small neighborhood.

One way of contrasting these paradigms is to consider the space of problems parameterized by allowable group size and workspace complexity as depicted in Fig. 2.1. It is interesting to note that both nearest-neighbor and fully centralized systems can be considered via communication abstractions which allows for ease in algorithm development and implementation but only for the class of problems amenable to these methods. On the

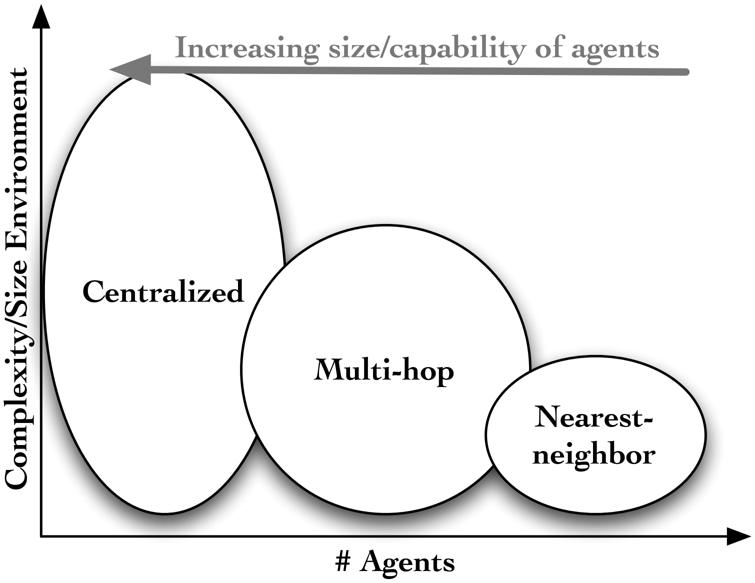

Increasing size/capability of agents
Complexity/Size Environment
Centralized
Multi-hop
Nearest-
neighbor
# Agents

Figure 2.1: Graphical depiction of the space of multi-robot systems. These systems can be classified based on allowable abstractions for their communication structure. Teams with very few agents can assume a centralized paradigm while very large teams must rely only on local information and nearest-neighbor interactions. There is a large middle category where the system explicitly computes the multi-hop network that is necessary for information sharing. Currently, centralized approaches are the most successful when dealing with large complex environments.

other hand, multi-hop methods can be applied to a larger class of problems but require more complicated implementations that explicitly address communication issues. Also, there is a general correlation that as one moves from nearest-neighbor systems to centralized systems, individual agents exhibit increasing size and capability. Currently, the increased agent capability translates to the ability to operate in large complex environments. A primary goal of multi-robot research is to increase the capabilities of multi-hop and nearest-neighbor systems.

Centralized

For small teams of highly capable agents, it is possible to consider a centralized system in which information is assumed to be globally available via high bandwidth connections between all agents. Clearly this is only tractable for small teams with large amounts of resources dedicated to inter-agent communication. However, in the limited space of problems where this is feasible, a centralized system greatly simplifies algorithm development and implementation. The requirements placed on an experimental testbed for verification of centralized algorithms are quite steep. In general, it requires a testbed that is itself centralized with excess bandwidth that can be devoted to the task of synchronizing agents.

Nearest-Neighbor

A large class of algorithms for coordinated control stem from the concept of nearest-neighbor communication with the assumption that an agent can identify and communicate only with some small set of immediate neighbors [43, 88, 78]. This communication abstraction is often typified by homogeneous teams with anonymous agents which allows for robustness to the addition or deletion of agents from the team.

A nearest-neighbor approach is warranted for applications where the system must be capable of scaling well in terms of the number of agents. However, a shortcoming of nearest-neighbor approaches is the translation of tasks from a group-level specification into rules based on local information which is often challenging. Since many nearest-neighbor algorithms rely on the existence of a connected interaction graph for convergence, these systems will work best when the density of agents is such that the frequency of connectedness is high. This limits their application in large environments.

Within the scope of nearest-neighbor systems, we advocate the adoption of an *asymmetric broadcast control* (ABC) architecture [69] where an external observer has the capability to communicate with all agents via a high-powered broadcast mechanism. In fact, we argue that as the team size grows, it becomes a mathematical necessity to define an abstraction that maps the large state space to a lower dimensional space that has a physical interpretation and is easy to control and command. This necessitates the need for a global supervisory agent that provides control inputs in the abstract state directly to all agents via a high bandwidth broadcast downlink. Individual agents then interpret the commands in the abstract space based on their local state and low-bandwidth coordination with neighbors. The asymmetric nature of the required communication when using the ABC architecture significantly reduces the complexity of network design and implementation for even very large systems.

Multi-hop

When dealing with any system not at the extremes of the space defined in Fig. 2.1, it becomes necessary to explicitly consider multi-hop communications. That is, uniquely identified agents must cooperatively route messages over a communication network. Because communications occur over shared channels, care must be given to properly allocate resources in order to satisfy problem constraints. Effectively, this means that algorithm design for coordinated control can not consider communication and coordination tasks in isolation via abstraction as is possible with both nearest-neighbor and centralized systems. Instead, the communication and coordination problems must be addressed in concert. While the requirements on a testbed to verify multi-hop systems are relatively limited, the actual implementation of these systems is generally quite complex.

2.2 Experimental Testbed

Experimental validation is particularly important in multi-robot systems research for a numer of reasons. First, the differences between models and real world conditions that may not be apparent in single robot experiments are amplified because of the large number of robots, interactions between robots, and the effects of asynchronous and distributed control, sensing, and actuation. Furthermore, accurate portrayal or modeling of some system components, such as wireless communication, is often infeasible. Finally, experimental validation forces the development of a comprehensive solution and eliminates many simplifying assumptions that cannot be engineered into a real system.

An experimental testbed must be:

- robust and reliable;
- scalable and allow the easy addition or deletion of agents;
- capable of measuring and logging state information (including ground truth) for analysis;
- extensible to a variety of applications;
- inexpensive; and
- easy to use and maintain.

Robustness and reliability is of great concern when designing an experimental testbed. Since an assumption is made on the performance of the testbed when evaluating an algorithm, uncharacterized failures prevent accurate verification. Scalability is the focus of much research and cannot be limited by the system implementation. Measurements, state information, and algorithm status provide real-time insight into the performance of the algorithms being tested and are invaluable during the debug phase of development. Furthermore, the ability to access or log such information at run-time or for post-processing is vital to the analysis of any experiment. In fact, it is this analysis of results that allows us to consider multi-robot systems in a scientific way.

Extensibility ensures that the testbed can be used to test a wide range of algorithms. We argue that over-specification of a system with respect to computation, communication, and sensing is important as it allows for the verification of a wide range of applications – the abilities of an individual robot can typically be artificially scaled to emulate less capable devices. The system must be designed to be inexpensive so that researchers can incrementally increase the size of the system. Finally, ease of use and maintenance are of great concern since the testbed consists of multiple independent units and supports collaborative research with many individuals accessing the system.

The remainder of this chapter focuses on the design decisions and specific components that make up the experimental testbed in the *Multi-robot systems laboratory* (MRSL) in the GRASP laboratory at the University of Pennsylvania. We cover the hardware, simulation tools, and two iterations of software architecture used for supporting a wide range of multi-robot experimentation. This work was completed in collaboration with Nathan Michael and with the initial pieces published in [68].

2.2.1 Hardware

The MRSL testbed consists of both ground and aerial robots. The central component of the ground fleet is the *Scarab* platform, a custom differential-drive wheeled platform. For aerial experiments, we rely on the *Hummingbird* and *Pelican* platforms designed and manufactured by Ascending Technologies GmbH [4]. A motion capture arena is available with full 6-DOF pose tracking data provided by a Vicon motion capture system [102] running at 100 Hz with millimeter accuracy.

The *Scarab* platform, in its third revision and pictured in Fig. 2.2, is driven by two Maxon A-Max brushed DC-motors which are in turn controlled with PWM by a dedicated microcontroller on the Acroname Moto 1.0 module with 3 Amp H-bridges. The Moto 1.0 module performs PID-based feedback with optical encoders to provide velocity control of each motor. The on-board computer is equipped with a Nano-ITX motherboard with a 1.5 Ghz processor, 1 GB of RAM and a 32 GB solid-state drive for storage. Wireless communications are facilitated by a high-bandwidth IEEE 802.11a device operating in the 5 GHz frequency range and a *Zigbee* device with $0.1 - 1$ mW ($-10 - 0$ dBm) power output in the 2.4 GHz frequency range [21]. Range sensing is provided by a Hokuyo URG laser range finder with a 5.5 m range and 240° field of view. Additional sensing, including cameras, can be easily added via the USB. Power is supplied via an *Inspired Energy* NL2024 smart Lithium-Ion battery. With 95Wh of capacity, it provides between 3 to 4 hours typical usage.

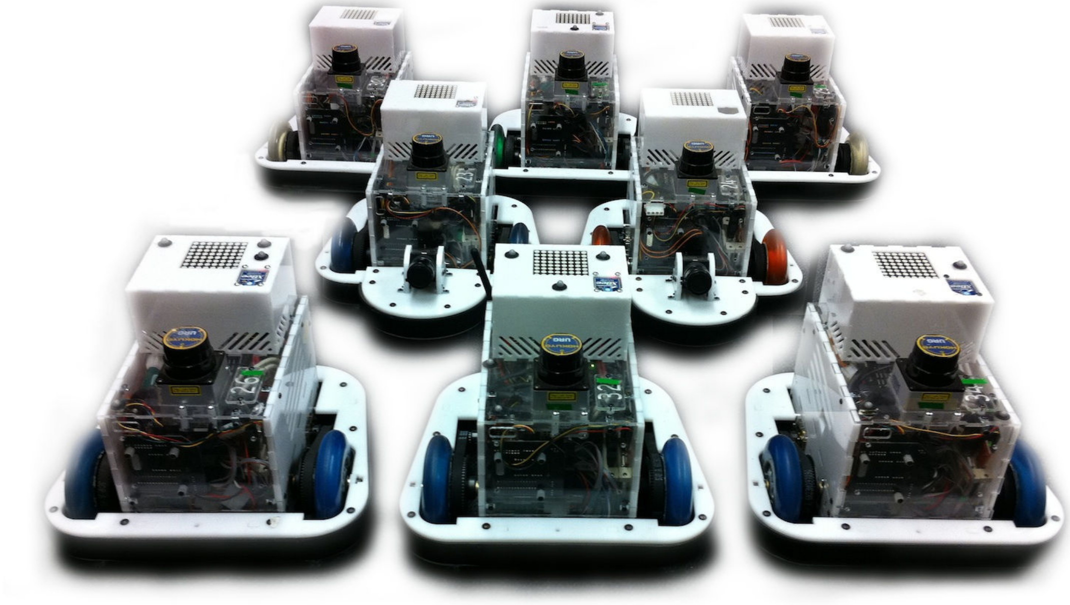


Figure 2.2: Eight *Scarab* platforms in the MRSL

2.2.2 Simulation

While it does not replace experimental validation, simulation tools are an integral part of the multi-robot testbed [26]. They allow for development and basic verification of algorithms in software before handling the additional challenges associated with deployment on a real system. Additionally, it provides opportunities to explore environments not physically available as well as the ability to gather many more trials than is feasible with real systems.

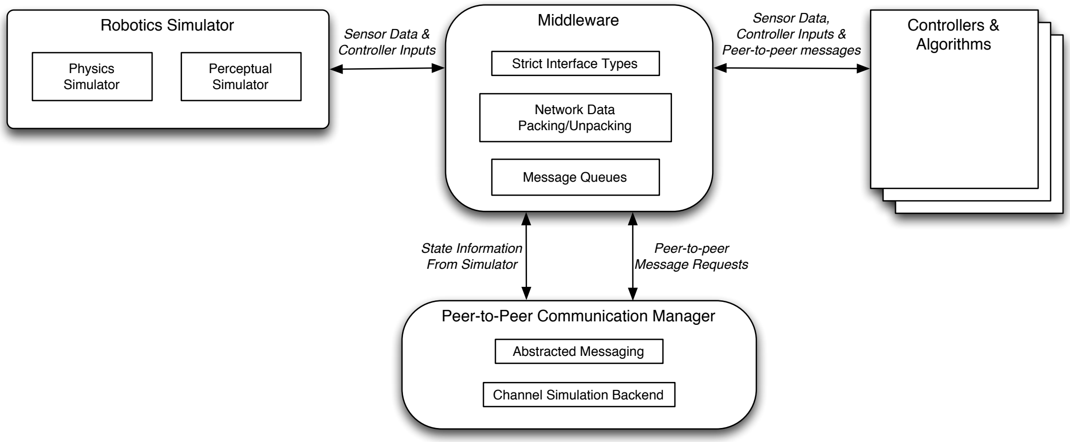

Robotics Simulator
Physics Simulator
Perceptual Simulator
Sensor Data & Controller Inputs
Middleware
Strict Interface Types
Network Data Packing/Unpacking
Message Queues
Sensor Data, Controller Inputs & Peer-to-peer messages
Controllers & Algorithms
State Information From Simulator
Peer-to-peer Message Requests
Peer-to-Peer Communication Manager
Abstracted Messaging
Channel Simulation Backend

Figure 2.3: Our basic simulation architecture is modular in design to allow for independent simulation of the physical and wireless communication systems. By relying on a middleware with proper software layering, we can switch transparently from simulation to experimental hardware.

Figure 2.3 depicts the structure of our simulation architecture. It is designed with a modular architecture with independent simulation of the physical and communication systems so that we can rely on a hierarchy of simulation fidelity. For example, research focusing on wheeled ground robots with low-power radio communication requires a simple kinematic simulation of the physical system but an accurate portrayal of the randomness inherent in wireless communication. On the other hand, a system that assumes a centralized algorithm for multi-robot manipulation is more considered with faithfully representing the physical interaction of each robot with the environment.

Central to this approach is a capable middleware, i.e., message passing software that allows for arbitrary messages and software layering. In effect, this allows us to abstract the development of algorithms from the interface with experimental hardware. The result of this abstraction is that controller and algorithm development can happen somewhat independently of the experimental hardware itself. Assuming simulation modules present the same interfaces and messaging as the actual testbed hardware, the transition from simulation to experiment is nearly transparent. Thus the challenge in designing simulation tools lies in the faithful reproduction of the interactions that take place with hardware rather than system simulation itself, i.e., physical interactions or wireless communication in our context. As more realistic simulation models become available, they can be applied to narrow the gap between simulation and experimental verification.

2.3 Experimental Architecture

In Section 2.2.2, we introduce the importance of a middleware for robotics software development. Here we build upon that to investigate the architectural issues that face experiment design for multi-robot research. It is important to note that the architecture employed for an experimental system does not directly map to the algorithms and architectures that we are researching. Indeed, we wish to design an experimental architecture that can be relied on to address a wide range of research. In a way, this makes the experimental architecture a meta-system within which we must implement the system we are interested in testing. Consequently, we must ensure that our experimental architecture sufficiently exceeds the capabilities that we assume for the system being studied. Finally, as stated in Section 2.2, ground truth information about the physical state of the system is necessary for scientific analysis. The experimental architecture must be amenable to collecting, logging, and real-time reporting of this information.

The experimental architecture for a multi-robot system is primarily a function of software engineering decisions, collection of ground truth, and communication capabilities. Each of these items is heavily influenced by the robotics middleware, self-localization capability, and radio hardware respectively.

Software engineering for a robotic system relies on the concept of a middleware that, as described in the previous section, provides arbitrary messaging and hardware abstraction layers. The framework of a robotics middleware allows us to write modular software that can be flexibly connected to build an experimental system. Within this experimental system, we implement the system and algorithms that are the focus of our research.

We rely on ROS [85], a so–called *robotics operating system*, which provides a number of development advantages. First, in terms of testing and debugging a complex system that relies on numerous drivers and algorithms, ROS adopts a flexible structure where each piece of software runs independently and connections are managed by a stable master node. This yields a number of stability benefits to the system when introducing new and untested software. ROS also maintains a low barrier for the creation of message types and communication channels. This supports modular code that can be shared across several applications. A second-order benefit of ROS is the growing community of users and availability of open-source software that address many of the necessary single-robot capabilities – namely perception, localization, and navigation.

The collection or aggregation of ground truth information presents the second driving concept behind an experimental architecture. Ground truth information includes the state of algorithms, sensor data, and the physical state of each robot in the experiment. This data must be measured and collected at a rate appropriate to the dynamics of the experiment for thorough analysis. Often, real-time access to this data is invaluable in algorithm testing and development. However, for a multi-robot system that is deployed throughout a large environment, the measurement and aggregation of this data can be challenging.

Finally, in the same sense that the experimental architecture must act as a meta-system to the multi-robot system undergoing verification, we propose that communication capabilities of the experimental architecture must be independent of those relied on by our research. On our experimental hardware, this is achieved through the use of two radios – high-powered IEEE 802.11a on the 5 GHz band for the experimental architecture and low-powered IEEE 802.15.4 (*Zigbee*) on the 2.4 GHz band for research algorithms. Relying on these two radios offers range and throughput for the experimental architecture that far exceeds the capabilities of the system being experimentally verified.

We consider two models for the experimental architecture. First, we present the centralized model, where we assume that the experimental network is always fully connected and that bandwidth is sufficient to support real-time data logging in a central location. Second, we present a decentralized model where the experimental architecture assumes each robot operates more independently with an experimental network that may have intermittent connectivity. Each of these models offers a different set of capabilities and is particularly amenable to a class of applications.

2.3.1 Centralized Model

In the centralized model, we assume that the full state information of the system is always available at a central location. This implies a high-throughput network that is fully connected and that tracking and localization of the physical state of the system is measured from a central point.

When it is possible to efficiently achieve these capabilities, the centralized model for an experimental architecture is advantageous because it offers easy methods for software development and debugging. Additionally, it allows us to study a wide class of multi-robot algorithms by assuming a subset of the experimental architecture's capabilities are available for the algorithm being investigated. However, as is expected, it becomes difficult to maintain these strict capabilities as the size of the multi-robot team and the operating environment grows.

Tracking & Localization

Initially, we developed and relied heavily on an overhead camera-based tracking system to provide a centralized tracking service for robots moving on the ground-plane. Eventually this system was replaced by the *Vicon* motion capture system [102] which provides full $6-\mathrm{DOF}$ pose measurements at 200 Hz. The overhead tracking system allows control algorithms to assume known pose in a global reference frame, thus eliminating the localization problem. Conversely, the tracking system allows the verification of localization algorithms as ground truth. It is also possible to use the tracking system in lieu of sensors that may be unavailable such as neighbor sensors or collision avoidance sensors. A centralized tracking solution such as the *Vicon* system offers the additional capability of being able to offer high fidelity pose information at high rates that can be used for studies in dynamics and advanced control.

Network

Since we need a low-latency network to communicate between agents and controllers with reasonable data rates, we rely on a dedicated 802.11a wireless access point in a frequency range not used by adjacent wireless networks to ensure that we have complete control over the bandwidth available to the robots. We have successfully experimented with tens of computers, robots, and sensors performing data-intensive experiments without a noticeable impact on the performance or latency of the network. A limitation of this centralized network infrastructure is that the robots are confined to operate within range of the access point. The IEEE 802.11abgn standard does not allow for fast enough access point switching to accomodate control algorithms.

Data Logging

A requisite component of an experimental system is logging functionality. The system design permits local or networked data logging depending on the demands of the experiment. Logging to local storage or mounted network drives on each robot is possible depending on the space and logging frequency required. Additionally, since we use PSG, a common logging interface exists that permits networked logging. As each robot communicates with other robots in the system, the same messages are sent to a computer which stores the data for post-processing. With such a design, we are able to log relevant system information without requiring significant computational overhead from the robots.

Additional Considerations

The robots and supporting computer infrastructure are networked with a dedicated local area network managed by a server with networked storage and a centralized user database.

A user remotely accesses the robots in the same way they would access a desktop computer and all working repositories and code are mounted via network drives. Since the robots and workstations all use the same x86 computer architecture, the same compiled binaries work on all platforms for easy development. Deployment is simple since the same storage is available on both robots and workstations. By viewing the team of robots as a system of networked computers and using off-the-shelf technology, we are able to effortlessly distribute changes in the code base to all of the robots. Additionally, the dedicated server hosts web server capabilities, a repository for software and documentation, and other data to facilitate research and collaboration.

2.3.2 Decentralized Model

As we transition towards algorithms and experiments that require larger and more complex environments, it is necessary to move towards a truly decentralized testbed while attempting to maintain some qualities of the centralized model. Since the centralized approach to experimental testbed design required a persistant network connection for the purposes of data logging and code deployment, its range is limited to that of a single access point. Switching between multiple IEEE802.11 access points does not occur quickly enough, e.g. on the order of 5 s, for real time control. Thus, in order to operate effectively across the environment in Fig. 2.4 we rely on several enabling technologies and components.

Localization

A second order strength of ROS is a large library of software for basic robotic tasks that has been developed and tested by Willow Garage [106]. The open-source release of the *navigation stack* with some optimizations for our platforms provides reliable laser-based localization and motion planning across very large environments as depicted in Fig. 2.4.

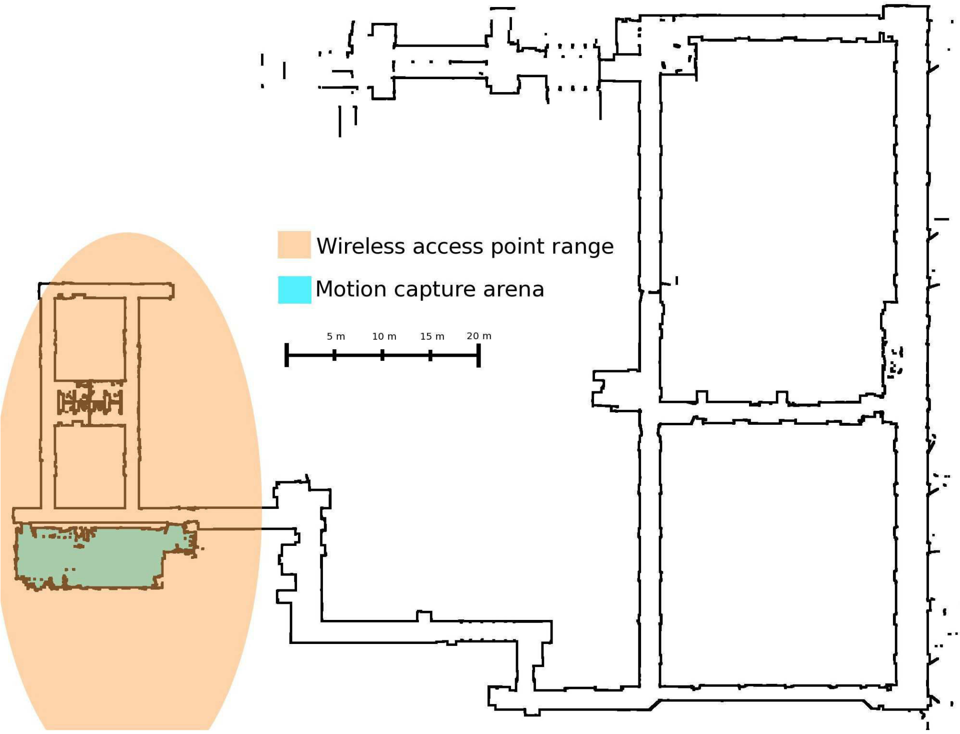


Figure 2.4: Map for laser-based localization. Note that under the centralized model, experiments were confined to the range of a single access point. The truly decentralized testbed allows for experiments that range throughout the available environment.

Mesh Network

The open80211s consortium [80] is developing an open-source implementation of the emerging IEEE 802.11s mesh networking standard and distributing it into modern linux kernels. It works effectively to allow transparent multi-hop routing of network traffic between many systems operating on the mesh where mesh routes are dynamically computed based on current conditions. By bridging the mesh network with our local wireless/wired network, we create an environment where laptops and desktops used for development are connected via the mesh network to robots across the building.

Decentralized Operation

Though the mesh network extends in the attempt to provide ubiquitous network connectivity, it is inevitable that during some experimentation, the connectivity will fail. In order to be robust to such failures, we seek to deploy ROS in a truly decentralized way. Though the focus of ROS development is on a single robotic system, the PR2, ROS message passing takes place via UDP or TCP/IP channels and can operate over a network on multiple machines. However, the presence of a central master that manages message connections makes it not well suited to deal with intermittent connectivity.

We have solved this problem by augmenting the default ROS master with a version that performs automatic service discovery of other ROS masters and allows for synchronization of communication channels that are currently being managed. In this way, an experimental system is designed such that each robot locally registers it's hardware drivers, algorithms, and communication channels with its local ROS master. When mesh connectivity is available, each robot's ROS master discovers and synchronizes its table of running software and interfaces.

Data Logging

The ROS environment provides an efficient means to log arbitrary messages to a local filesystem. This means that logging of state information, communication messages, and other sensory data on an individual robot is trivial. However, the intermittent connectivity of a truly decentralized experimental testbed complicates the matters of data logging for experimental verification. We address this problem by ensuring that each robot operates on a synchronized clock and aggregating individual logs after each experimental trial. While this does not provide quite the same ability of the centralized testbed where the full state of the system can be queried at any time during an experiment, it is a robust and reliable way to log large amounts of data without overburdening the mesh network.

Chapter 3

Coordinated Control

We present two examples of problem statements targeted for networked teams of robots. The first is a case study involving caging manipulation by a team of agents. It assumes a disc model for local communication with asynchronous broadcast control to facilitate decentralized controllers. The second example is more tightly coupled as it considers the networking and mobility problems associated with providing general situational awareness capabilities to a team of mobile robots. It serves as motivating problem statement for the remainder of this thesis.

3.1 Multi-Robot Manipulation via Caging

The concept of object closure or caging for manipulation is introduced in [82] where multiple robots are used to geometrically confine a payload to a compact set in the plane. Here we extend this method for manipulation with a decentralized algorithm for large teams of robots [27, 29] by employing sequential composition of vector fields as in [42]. Team behaviors are chosen that *approach*, *surround*, and *transport* a payload where each behavior is defined by a vector field that requires global task parameters and local state information allowing it to operate within the ABC architecture. Transition between behaviors is accomplished by local sensing and message passing as depicted in Fig. 3.1. In this way we are able to construct a system for payload transport that is robust to agent addition or deletion, the presence of obstacles, and a wide range of modeling errors with respect to the underlying manipulation task.

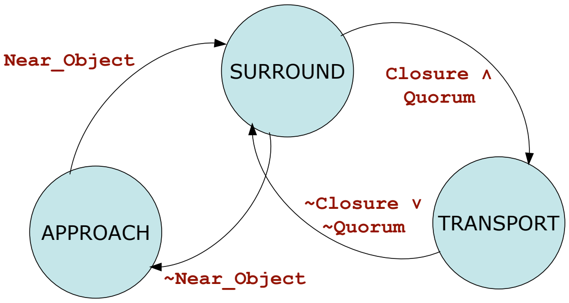


Figure 3.1: Behavior architecture. The software for each robot is identical and consists of several simple modes and the sequential composition of these modes.

3.1.1 Methodology

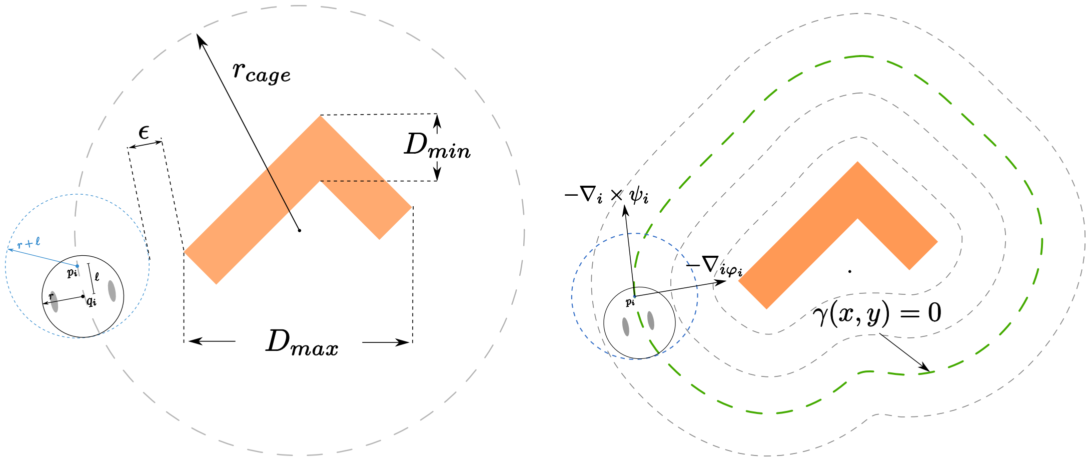
We consider a group of N planar, fully actuated robots each with kinematics given by $\dot{q}_i = u_i$ where $q_i = (x_i, y_i)^T$ and u_i denote the ith agent's position and control input. We assume the agents are disk–shaped with radius r_i and can localize themselves in a shared global coordinate frame. In addition, we assume agents are able to sense the proximity of their teammates and/or obstacles within the environment. Thus, the neighborhood of q_i is determined by the range and field of view of the sensing hardware. We denote the set of neighbors for agent i by Γ_i. For collision and obstacle avoidance purposes, we assume a circular influence range, R_i, such that collision and obstacle avoidance maneuvers are active only when agents are within this range.

Our objective is to design control inputs to enable the N–robot team to approach, surround, and transport a payload in an environment with circular obstacles from one location to another. To achieve this, we propose to construct artificial potential functions, φ, that can stabilize the agents onto a one-dimensional boundary (curve) such that they cage the desired payload and achieve closure by orbiting around the payload, all the while avoiding collisions. Once closure is obtained, the robots transport the payload from an initial position to a goal location based on a global navigation function, Φ, while maintaining closure.

Given a convex workspace $\mathcal{W}$, we assume the boundary, denoted by $\partial\mathcal{W}$, can be described by an implicit function of the form $s_{\mathcal{W}_0} - s_{\mathcal{W}}(x, y) = 0$. For the given $\mathcal{W}$, we denote the set of circular obstacles in $\mathcal{W}$ as $\mathcal{O}$ and let $q_{\mathcal{O},k}$ and ρ_k be the center and the radius of each obstacle $k \in \mathcal{O}$ respectively. Furthermore, given a payload whose centroid is denoted as q_{obj}, we assume there exists a smooth star shape, $\mathcal{S}$, whose boundary, $\partial\mathcal{S}$, is a smooth, regular, simple, closed curve of the form $s(x, y) = 0$ such that the desired payload is contained within $\partial\mathcal{S}$. In general, we can always find such a shape $\mathcal{S}$ by considering the following two parameters of the payload. First, define $D_{min}(obj)$ as the smallest gap through which the payload will fit, and second, $D_{max}(obj)$ as the maximum distance between any two points on the payload. Then, for any given payload, the circular boundary with radius given by:

$$r_{cage} = \frac{1}{2} D_{max}(obj) + \max_i r_i + \epsilon, \tag{3.1}$$

where $\epsilon > 0$ is a constant scalar, will always contain the payload as depicted in Fig. 3.2a. We refer to this circle as the *caging* circle.



(a) (b)

Figure 3.2: Visualization of the caging manipulation methodology. (a) depicts notation such as the payload properties and caging radius. (b) depicts the shape control surface along with the *descent* and *surround* components of the vector field.

Given the caging radius r_{cage} and a team of N robots each with radius $r_i > 0$ and influence range $R_i > 0$, we define $r = \max_{r_i \forall i} r_i$ and $R = \max_{R_i \forall i} R_i$. Our goal is to synthesize decentralized controllers that will allow the team to surround the payload while avoiding collisions. Therefore, the length of $\partial \mathcal{S}$, L, will naturally impose an upper bound on the number of robots, e.g. $N_{max} > 0$, that can fit on this boundary. Additionally, for a given r_{cage} and $D_{min}(obj)$, there must be at least $N_{min} > 0$ number of agents to ensure object closure. We make the following assumptions where $d_{min}(\cdot, \cdot)$ denotes the minimum Euclidean distance between a position and boundary in $\mathcal{W}$:

1. $N_{min} \leq N < N_{max}$;
2. $r_{cage} > R$;
3. $d_{min}(q_{obj}(0), \partial \mathcal{W}) > (r_{cage} + 2r)$, where $q_{obj}(0)$ denotes the inital position of the obstacle;
4. $\|q_{obj}(0) - q_k\| > (r_{cage} + 2R)$ for all $k \in \mathcal{O}$;
5. $\|q_{\mathcal{O},j} - q_{\mathcal{O},k}\| > (\rho_j + \rho_k + 2R)$ for all j,$k \in \mathcal{O}$; and
6. $\|q_{obj} - q_i\| \leq (r_{cage} + r)$ for $i = 1, ..., N$ in the Transport behavior.

Assumption 1 ensures the agents be able to surround the payload and achieve closure. Assumption 2 ensures convergence of the team to the boundary surrounding the payload. Assumptions 3 and 4 ensure the payload is initially located at a position where agents can surround and orbit it without colliding with $\partial \mathcal{W}$ or other obstacles in the environment. Assumption 5 ensures the agents will have the ability to maneuver around the obstacles in the environment. Finally, assumption 6 ensures the agents will be able to maintain closure when transporting the payload around the environment.

Our approach to caging and manipulation of payloads can be summarized by the behavior architecture in Figure 3.1. The architecture relies on three behaviors.

1. **Approach**: The robot approaches the payload while avoiding collisions with obstacles and other robots in the environment;

2. **Surround**: The robot stabilizes to a trajectory that orbits the payload while avoiding collisions with other robots; and

3. **Transport**: The team moves toward the goal configuration following a global navigation function.

As shown in Fig. 3.1, transitions between behaviors are based on conditions derived from sensor abstractions. If a robot is near the payload, a sensor sets its `Near_Object` flag causing that robot to switch to SURROUND mode. A `Quorum` flag is set based on the number of neighbors within its field of view. The `Closure` flag is set when the robot team surrounds the payload. When `Closure` and `Quorum` are both set, the robot enters the TRANSPORT mode and starts moving the payload. Reseting the flags can cause the robot to regress into a different mode.

Shape controller

We base our decentralized feedback controllers for boundary following on the ones described in [39]. This controller has been shown to be scalable to large teams and stability and convergence properties have been established with collision and obstacle avoidance guarantees for a certain class of boundaries.

Let $\partial\mathcal{S}$ be described by $s(x,y)=0$ such that $s(x,y)<0$ for all (x,y) in the interior of $\partial\mathcal{S}$ and $s(x,y)>0$ for all (x,y) in the exterior of $\partial\mathcal{S}$. Let $\gamma=s(x,y)$, and $\beta_0 = s^2_{\mathcal{W}_0}-s^2_{\mathcal{W}}(x,y)$ so that we can define the *shape navigation function*, φ, as

$$\varphi(q)=\frac{\gamma^2}{\gamma^2+\beta_0}. \tag{3.2}$$

The shape navigation function will generate an input to drive each agent towards $\partial\mathcal{S}$.

To enable the agents to surround and orbit the object in a counter-clockwise direction, let

$$\psi=\left[0,\,0,\,\frac{\gamma}{\sqrt{\gamma^2+\beta_0}}\right]^T.$$

We impose an additional input given by $-\nabla \times \psi$, where $\nabla \times \psi$ is a vector tangent to the level set curves of φ which drives each agent to travel along the boundary in a counterclockwise direction. To enable avoidance of circular obstacles in the environment, a third input is included to drive the robots around any obstacles within their influence range. Thus, the decentralized shape control law is given by:

$$\begin{aligned} u_i &= -K_N \nabla_i \varphi_i \cdot f(N_i) - \nabla_i \times \psi_i \cdot g(T_i) \\ &\quad - \sum_{j \in \Gamma_i, \|v_j\|=0} \frac{\nabla_i \times \hat{\beta}_j}{d_{ij}^{p_2}} h(q_i, q_j) \end{aligned} \tag{3.3}$$

where ∇_i denotes differentiation with respect to agent i's coordinates, $d_{ij} = \|q_i - q_j\| - (r_i + r_j)$, $\varphi_i = \varphi(q_i)$, $\psi_i = \psi(q_i)$, and $\hat{\beta}_j$ is defined as:

$$\hat{\beta}_j = \left[0,\, 0,\, \frac{d_{ij}}{\sqrt{d_{ij}^2 + \beta_0}} \right]^T .$$

The functions $f(N_i), g(T_i) \in [0, 1]$ and $h(q_i, q_j) \in [-1, 1]$ are scalar switching functions used to independently modulate each term in (3.3). As such, the components of each robot's vector field are modulated to avoid collisions depending on the positions of its neighbors and the obstacles in the environment. The first term of (3.3) drives the agents towards $\partial\mathcal{S}$, the second term drives them along the level set curves of φ in a counterclockwise direction, and the third term enables them to avoid collision with obstacles in their neighborhood as depicted in Fig. 3.2b. We note since the φ and ψ are common among all agents, robots do not have to exchange information. Instead, it suffices to know only the positions of the neighbors which be obtained via sensing alone.

Composition of behaviors

Our multi-robot manipulation system is based on a decentralized shape controller and a global navigation function with proven stability and convergence results [39, 90]. By varying certain properties of our controller, i.e. the sequential composition of (3.3) and

(3.7), each mobile agent can operate in one of several modes: Approach, Surround, or Transport. Composition of these modes result in a global vector field such that, when combined with local interactions, achieves the desired task. Transitions between these modes, as well as exceptions in the case of failure, can be defined robustly while keeping individual agent decisions a function of local sensing.

The Approach behavior is characterized by a larger gain K_N on the descent component of (3.3) to yield agent trajectories that efficiently approach the object from a distance while avoiding collisions with other agents.

In the Surround mode, agents are near the desired shape and K_N is decreased so that the agents can distribute themselves around the object. Given enough robots, this behavior will lead to object closure. For a given r_{cage} and $D_{min}(obj)$, the minimum necessary number of robots to achieve object closure is

$$N_{min} = \frac{2\pi r_{cage}}{2r + D_{min}(obj)}. \tag{3.4}$$

Additionally, to ensure convergence to the boundary of the desired shape, the size of the team must be no greater than

$$N_{max} = \frac{\pi r_{cage}}{r}. \tag{3.5}$$

In practice, we do not always require this condition as the state transition events are robust to excess robots.

The Transport mode controller relies on composition of the first two terms in (3.3) with a descent direction derived from a global navigation function [90]. Given the workspace, $\mathcal{W}$, circular obstacles $\mathcal{O}$, and a payload goal configuration q_{goal}, let

$$\beta_k(q) = \|q - q_k\|^2 - (\rho_k + 2r + r_{cage})^2,$$

where $k \in \mathcal{O}$. Then the global navigation function is given by

$$\Phi(q) = \frac{\|q - q_{goal}\|}{\left[\|q - q_{goal}\|^{2\kappa} + \Pi_{k=0}^{|\mathcal{O}|}\beta_k\right]^{1/\kappa}} \tag{3.6}$$

with the feedback controller for the payload given by

$$u_{obj} = -\nabla_{obj}\Phi(q_{obj}). \tag{3.7}$$

Here, $\Phi(q)$ is conservatively constructed by expanding the boundaries of the obstacles and by decreasing the world boundary by $(2r + r_{cage})$ so as to ensure collisions do not occur between the team and the environment.

In general, each agent's transition between modes will result from local observations of its neighbors as well as its distance to the manipulated object. An agent will initialize to the Approach mode if $D_{\text{obj}}(q_i) > D_{\texttt{near\_object}}$ (i.e. it is far from the object). As the agent approaches the desired caging shape, $D_{\text{obj}}(q_i) \leq D_{\texttt{near\_object}}$ will result in a transition to the Surround mode.

In the Surround mode, the orbiting term of the shape controller is favored so the robots are distributed around the object to be manipulated. Given at least N_{min} agents, this mode converges to an equilibrium where object closure is attained. While closure is a global property of the system, we propose an algorithm for local estimation of closure.

To locally define quorum, we introduce the concept of a *forward* neighborhood $\hat{\Gamma}^+$ and a *backward* neighborhood $\hat{\Gamma}^-$ with respect to the manipulated object and the Surround mode. For agent i, the Surround mode introduces an approach component, i.e. $\nabla_i\varphi_i$, and rotation component, i.e. $\nabla_i \times \psi_i$, so that we can define the set of robots to be in front of agent i and another set behind. If a neighborhood $\hat{\Gamma}_i$ represents the agents within a distance $D_{min}(obj)$, then

$$\hat{\Gamma}_i^{\pm} = \{j \in \Gamma_i \mid 0 < \pm(q_j - q_i)^T(\nabla_i \times \psi_i)\}. \tag{3.8}$$

Furthermore, we can define specific agents from $\hat{\Gamma}_i^+$ and $\hat{\Gamma}_i^-$,

$$i^+ = \text{argmax}_{k \in \hat{\Gamma}_i^+} \frac{(q_k - q_i)^T \nabla_i \varphi_i}{\|q_k - q_i\|}, \tag{3.9}$$

$$i^- = \text{argmax}_{k \in \hat{\Gamma}_i^-} \frac{-(q_k - q_i)^T \nabla_i \varphi_i}{\|q_k - q_i\|}, \tag{3.10}$$

to be the adjacent agents in the potential cage around the object as depicted in Fig. 3.3.

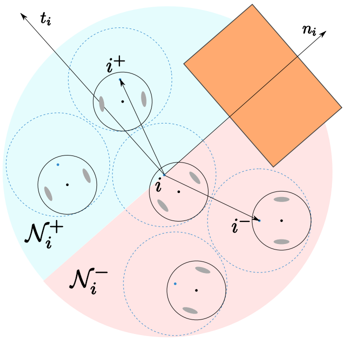


Figure 3.3: Agent i's neighborhoods $\hat{\Gamma}_i^+$ and $\hat{\Gamma}_i^-$ with i^+ and i^-

Since we are interested in strategies requiring little to no communication among the agents, consider the following update rule for $\texttt{quorum}_i$,

$$\texttt{quorum}_i = \begin{cases} 0 & \text{if } (\hat{\Gamma}_i^+ = \emptyset) \vee (\hat{\Gamma}_i^- = \emptyset), \\ N_{min} & \text{if } f(i^+, i^-) > N_{min}, \\ \min(\texttt{quorum}_{i^+}, \texttt{quorum}_{i^0}) + 1 & \text{otherwise.} \end{cases} \tag{3.11}$$

This strategy only requires the exchange of small bits of information and can be easily handled by agents with limited communication resources. We shall use $\texttt{quorum}_i$, $\texttt{quorum}_{i^+}$, and $\texttt{quorum}_{i^-}$ to determine when there is object closure.

If there is no closed loop around the object, $\texttt{quorum}_i$ will converge to the minimum of N_{min} and the shorter of the forward/backward chain of agents. On the other hand, if there is a complete loop around the object, $\texttt{quorum}_i$ will grow as large as the imposed bound N_{min}.

We define *local* closure to be

$$\begin{gathered}\texttt{closure}_i = \\ (\texttt{quorum}_i \geq N_{min}) \wedge (\texttt{quorum}_i = \texttt{quorum}_{i+}) \wedge (\texttt{quorum}_i = \texttt{quorum}_{i-}).\end{gathered} \tag{3.12}$$

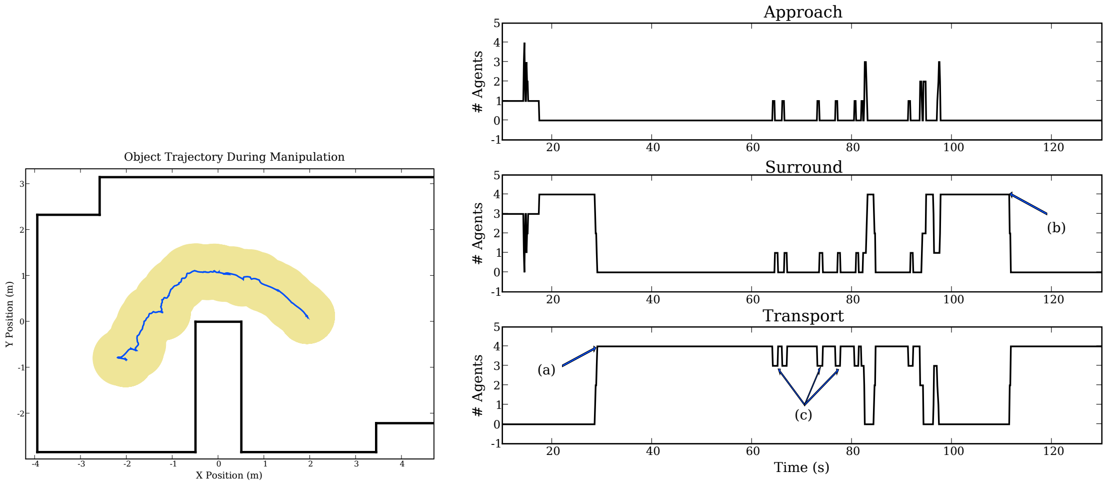
Our condition for local closure will coincide with global closure for any situation where up to $2N_{min}$ agents are used to form a cage around the object.

When an agent estimates that local closure has been attained, it will switch to the Transport mode and begin manipulation of the object. The `quorum` and `closure` events are defined such that they represent a kind of distributed consensus and as a result, the set of manipulating agents will switch into the Transport mode in a nearly simultaneous fashion. Should `closure` be lost during manipulation, each agent in the system will return to the Surround mode to re-acquire the object.

3.1.2 Results

For experimental verification, we implemented and deployed the algorithms for caging manipulation on the centralized testbed described in Section 2.3.1. Though the testbed is based on a centralized model, it is well suited for decentralized algorithms where identical software executes on each agent. Rather than address the perception problems associated with each agent performing local estimates of its neighbor's states, we simulate these measurements via global state information. Thus, we are able to test decentralized caging algorithms that rely entirely on information that is local to each agent.

Figure 3.4 depicts an experimental trial where four robots approach, surround, and transport a circular payload around an obstacle in the workspace. Figure 3.4a depicts the trajectory of the object as it is manipulated around a virtual obstacle inside the workspace. The shaded region around the trajectory denotes the conservative approximation used to represent the area occupied by the payload and robots during the execution of the task. The number of agents executing the behaviors Approach, Surround, and Transport during the experiment is shown in Fig. 3.4b to demonstrate how the agents reach consensus and transition from one behavior to the next. The trial begins with three of the



(a) (b)

Figure 3.4: (a) Object trajectory and area occupied by the object and the robots during task execution. (b) The number of agents in each mode over the duration of the experiment. Marker "a" points to the first instance when all agents have entered the Transport mode. Marker "b" denotes a time when closure was lost and the robots attempt to re–acquire the object. Marker "c" identifies instances where individual robots detect loss of closure, however, closure is regained before the object escapes.

four robots already close enough to the object to enter the surround mode. Once all four agents enter the surround mode, i.e. at 20 s, they begin exchanging messages with their neighbors to determine when closure is achieved. At approximately 30 s, all four nodes simultaneously detect closure and enter the transport state. During manipulation there are several instances where closure is lost – causing the agents to stop Transport and fall back into the Surround behavior until closure is regained. This behavior is entirely decentralized and contributes to the robustness of this method for multi-robot manipulation.

Figure 3.5 depicts the transport of an L-shaped payload through a field of obstacles. It demonstrates the composition of vector fields in a more complex environment where agents must approach the payload through a field of obstacles. Once the payload is caged, a global navigation function is followed to steer the team through the field of obstacles. As this experimental trial is generated with exactly the same decentralized controllers running

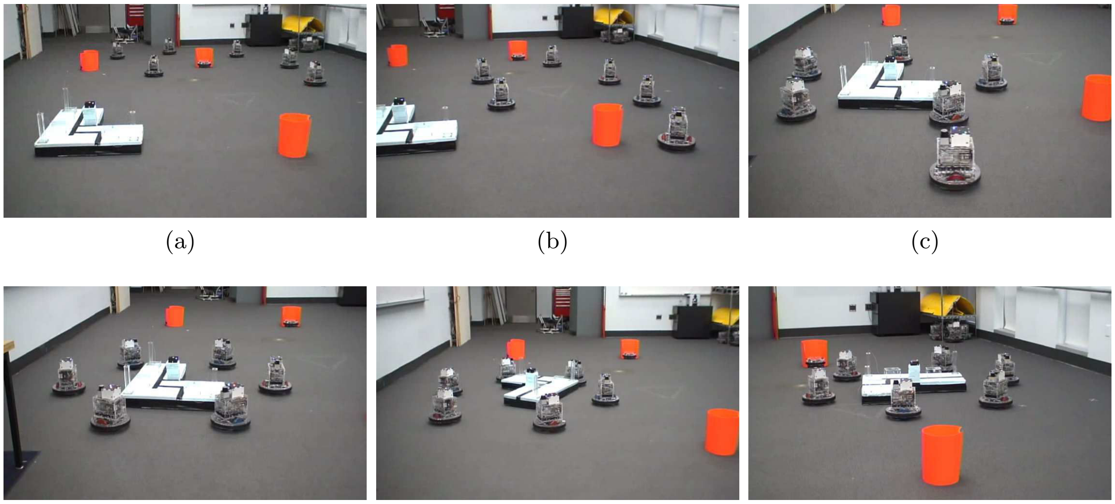

(a) (b) (c) (d) (e) (f)

Figure 3.5: Transport of an L-shaped payload through a field of obstacles with a decentralized method for multi-robot manipulation via caging. Figures. (a) and (b) depict the *approach* behavior, Figs. (c) and (d) depict the *surround* behavior and Figs. (e) and (f) the *transport* behavior.

on each robot, it serves to demonstrate the generality of our approach. No changes to the algorithm are necessary to utilize more robots or transport a payload with more complex geometry.

Through experimental verification, we have demonstrated that, with varying numbers of mobile robots and varying payload geometries, our algorithm facilitates payload transportation in complex environments. By performing manipulation based on the ideas of caging, we avoid the considerable challenges of modeling the effect of surface friction between the payload and supporting surface. Furthermore, our algorithm for decentralized caging manipulation is well suited to the asynchronous broadcast control architecture. It relies on identical anonymous agents that receive high–level task specifications, e.g. r_{cage}, $D_{min,max}$, and Φ, from a supervisory agent that requires no aggregation of information from individual agents.

3.1.3 Summary

There are currently some limitations to this work. The potential field approaches we employ will present issues when applied to more complex environments. Fortunately, the

Approach behavior we present could be substituted with a straight-forward motion planning algorithm that brings agents through an complex environment until they are within a convex region of free space that includes the payload. Also, the notion of caging manipulation implicitly assumes a conservative region of free space within which to transport the payload. More tightly coupled approaches to multi-robot manipulation are applicable if we require precise positioning in narrow environments, e.g. [10, 28]. Furthermore, while the underlying shape-control behavior will scale well to large teams of robots, the manipulation task we present here will only succeed for $N_{min} \leq N \leq N_{max}$ agents. This presents a limitation that must be resolved with resource allocation methods to ensure that an appropriate number of agents interact with the payload at one time. This kind of global allocation can be handled by the supervisory agent with the ABC architecture.

In this work, maintenance of communication constraints were simplified by several features. First, inter-agent collision avoidance is based entirely on local knowledge of neighbor states. This information is obtained either by sensing or simple one-way broadcast of state information. There is no need to reach any sort of consensus – inter-agent priority is explicitly determined based on a common control policy. Even in the case of unreliable communication of local state information, this approach would succeed so long as the dynamics of motion are slower than the wireless channel fading which is typically true.

Second, the one part of this work that requires communication for coordination is the determination of object closure. For this, we assume reliable communication between any agents with range R. Since message passing only happens between neighbors that are closer than $D_{min}(\text{obj})$, i.e., close enough that the object cannot escape, we must only ensure that $D_{min}(\text{obj}) < R$. In fact, we can assert this inequality for any object and maintain a conservative notion of closure. Of course, a system where it is impossible to guarantee reliable communication for any range R will require further analysis to demonstrate that consensus is reached with regards to payload closure.

3.2 Situational Awareness

The general problem of maintaining "situational awareness" for a team of networked mobile robots depends on the ability of the team to deploy into an environment and accomplish a sensing task while relaying the result of sensing to an operating center that coordinates high–level tasks. A typical scenario is search and rescue missions in hazardous situations where a team is deployed to scout points of interest. While a designated lead member of the team moves to a specified location, the remaining robots provide mission support. Critical for task accomplishment is the availability of wireless communications. Communication is required to exchange information between robots as well as to relay information to and from the human operators.

Availability of wireless communication infrastructure is unlikely in the harsh environments in which autonomous robot teams are to be deployed. Rather, we want the robots to self organize into a wireless network capable of supporting the necessary information exchanges.

3.2.1 Problem Statement

Consider a team of N robots and denote their positions as x_i, for $i = 1, \ldots, N$. We assume the robots are kinematic and fully controllable which allows us to consider simple models of the form

$$\dot{x}_i(t) = u_i(t), \tag{3.13}$$

where $u_i(t)$ is the input. A fixed operation center that we index as $i = 0$ is located at position x_0. Further define vectors $\mathbf{x} := (x_0, \ldots, x_N) \in \mathbb{R}^{2(N+1)}$ and $\dot{\mathbf{x}} = (\dot{x}_0, \ldots, \dot{x}_N) \in \mathbb{R}^{2(N+1)}$. The task assigned to the team is specified through a generic scalar convex task potential function $\Psi : \mathbb{R}^{2(N+1)} \to \mathbb{R}$. If the potential minimum $\Psi_{\min}$ is attained at $\mathbf{x}^*$, i.e., if $\Psi(\mathbf{x}^*) = \Psi_{\min}$, the configuration $\mathbf{x}^*$ satisfies task completion. E.g., if a designated leader agent ℓ must visit a target location $x_{\ell,goal} \in \mathbb{R}^2$, we can define

$$\Psi(\mathbf{x}) = \|x_\ell - x_{\ell,goal}\|^2. \tag{3.14}$$

The minimum $\Psi_{\min} = 0$ is attained by any configuration $\mathbf{x}^* = (x_0, \dots, x_\ell, \dots, x_N)$ for which $x_\ell = x_{\ell,goal}$, or equivalently by any member of the set

$$\mathbf{x}^* \in \{\mathbf{x} = (x_0, \dots, x_\ell, \dots, x_N) : x_\ell = x_{\ell,goal}\}. \tag{3.15}$$

Irrespective of the particular form of $\Psi(\mathbf{x})$, the control problem is to find inputs $\dot{\mathbf{x}}(t)$ such that at some time t_f the team configuration $\mathbf{x}(t_f) = \mathbf{x}(0) + \int_0^{t_f} \dot{\mathbf{x}}(t)dt$ satisfies task completion in that we have $\Psi(\mathbf{x}(t_f)) = \Psi_{\min}$. Mathematically, we can write this mobility control formulation as

$$\begin{aligned} \min_{\dot{\mathbf{x}}(t), t\in[0,t_f]} \quad & \Psi(\mathbf{x}(t_f)) \\ \text{subject to} \quad & \mathbf{x}(t) = \mathbf{x}(0) + \int_0^t \dot{\mathbf{x}}(u)\ du\ . \end{aligned} \tag{3.16}$$

As robots move to accomplish their task, they maintain *end-to-end* data communication flows between members of the team and/or members of the team and the operation center. Information flows are indexed as $k = 1, \dots, K$. Flows may have multiple sources and multiple destinations. The set of destinations of the k-th information flow is denoted as dest(k). For agent i and flow k, the variable $a_{i,\min}^k$ represents the required communication rate between agent i and any of the agents in the set of destinations dest(k). E.g., if the only communications of interest are from the lead robot ℓ to the operation center, there are only $K = 1$ flows. Since the flow $k = 1$ is intended to the operating center, dest$(1) = 0$ and $a_{\ell\min}^1$ denotes the minimum level of service for the communication from the leader to the operating center. All other variables $a_{i\min}^k = 0$ are null.

We model *point-to-point* connectivity through a rate function $R_{ij}(\mathbf{x}) = R_{ij}(x_i, x_j)$ that determines the amount of information that agent i at position x_i can send to agent j at

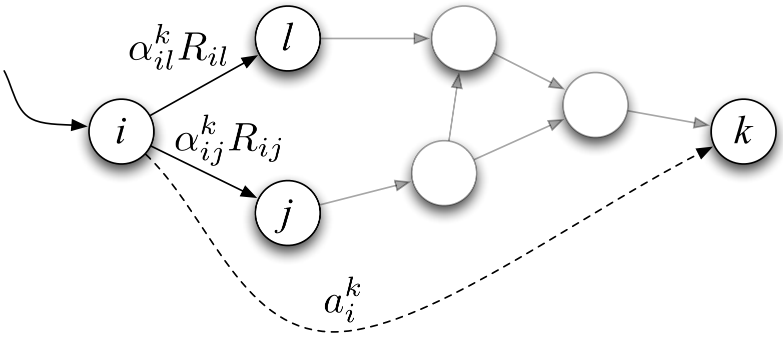


Figure 3.6: A network of nodes where we are interested in end-to-end rates from node i to destination (or flow) k. Routing variables α_{ij}^k determine the fraction of time node i sends packets to node j for flow k. R_{ij} is defined as the supported rate of the wireless channel from node i to node j.

position x_j. Since direct communication between the source and the destination of an information flow is not always possible, terminals self-organize into a multihop network to relay packets for each other. Packet relaying is determined by routing variables α_{ij}^k which describe the fraction of time node i spends transmitting data for flow k to node j; see Fig. 3.6. Thus, the product $\alpha_{ij}^k R_{ij}(\mathbf{x})$ determines the rate of point-to-point information transmission from i to j. If we consider the transmission to all neighboring terminals for which $R_{ij}(\mathbf{x}) > 0$, the total rate at which packets leave agent i is $\sum_{j=0}^{N} \alpha_{ij}^k R_{ij}(\mathbf{x})$. Likewise, the total rate at which i receives packets from other terminals is $\sum_{j=0, j\notin \mathrm{dest}(k)}^{N} \alpha_{ji}^k R_{ji}(\mathbf{x})$. The information rate $a_i^k(\boldsymbol{\alpha}, \mathbf{x})$ available for flow k at source i is the difference between outgoing and incoming rates

$$a_i^k(\boldsymbol{\alpha}, \mathbf{x}) = \underbrace{\sum_{j=0}^{N} \alpha_{ij}^k R_{ij}(\mathbf{x})}_{\text{Outgoing packets}} - \underbrace{\sum_{j=0, j\notin \mathrm{dest}(k)}^{N} \alpha_{ji}^k R_{ji}(\mathbf{x})}_{\text{Incoming packets}}, \tag{3.17}$$

where we define the vector $\boldsymbol{\alpha}$ grouping all routing variables α_{ij}^k. Notice that the variables α_{ij}^k represent time slot shares and must therefore satisfy $0 \leq \alpha_{ij}^k \leq 1$ for all i, j, and k. It must also be that $\sum_{j,k} \alpha_{ij}^k \leq 1$ for all i to ensure that the sum of all time shares at terminal i does note exceed 1. It is possible to, alternatively, require $\sum_{i,j,k} \alpha_{ij}^k \leq 1$ if we wish for only one link to be active at any time across the entire network.

Routing variables $\boldsymbol{\alpha}$ and configuration-dependent rates $R_{ij}(\mathbf{x})$ determine the set $a_i^k(\boldsymbol{\alpha}, \mathbf{x})$ of end-to-end communication rates from each node i and flow k as per (3.17). The task specification requires that end-to-end rates exceed the minimum threshold $a_{i,\min}^k$. Therefore, integrity of the communication network necessitates that for all i and k

$$a_i^k(\boldsymbol{\alpha}, \mathbf{x}) \geq a_{i,\min}^k \quad \text{for all } i, k. \tag{3.18}$$

Notice that $a_i^k(\boldsymbol{\alpha}, \mathbf{x})$ is a function of positions $\mathbf{x}$ and routing variables $\boldsymbol{\alpha}$. To control end-to-end connectivity, i.e., to satisfy (3.18), we can resort to control of positions $\mathbf{x}$, to control of routes $\boldsymbol{\alpha}$, or both.

Since communication is necessary for task completion, the mobility control problem as summarized in (3.16) is redefined. The new goal is to find algorithms and control policies that govern robot motions in order to satisfy the task specifications in (3.16) and (3.18). Reducing $\Psi(\mathbf{x})$ as per (3.16) and ensuring network integrity as per (3.18) may be conflicting requirements. We therefore replace (3.16) by a concurrent search of trajectories $\mathbf{x}(t)$ and routes $\boldsymbol{\alpha}(t)$ so that the task potential is minimized without ever breaking communication connectivity. Mathematically, we write this objective as the optimization problem

$$\begin{aligned} \min_{\boldsymbol{\alpha}(t), \dot{\mathbf{x}}(t), t \in [0, t_f]} \quad & \Psi(\mathbf{x}(t_f)) \\ \text{subject to} \quad & a_i^k(\boldsymbol{\alpha}(t), \mathbf{x}(t)) \geq a_{i,\min}^k \\ & \mathbf{x}(t) = \mathbf{x}(0) + \int_0^t \dot{\mathbf{x}}(u)\ du \end{aligned} \tag{3.19}$$

where rates $a_i^k(\boldsymbol{\alpha}(t), \mathbf{x}(t))$ are given by the expression in (3.17) with $\boldsymbol{\alpha} = \boldsymbol{\alpha}(t)$ and $\mathbf{x} = \mathbf{x}(t)$.

Figure 3.7 depicts an architectural diagram of our proposed solution to the concurrent routing and mobility problem (3.19). There is a module performing task specification, a second module computing the control law and networking solutions, and a third module conducting actuation and state estimation. A parallel module estimates and predicts point-to-point communication capabilities. The task specification module interfaces with

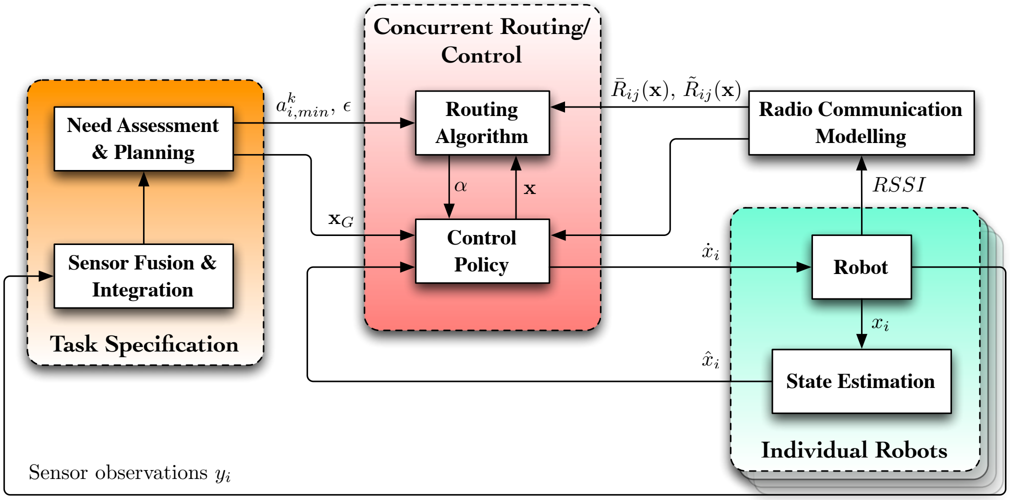

Concurrent Routing/ Control
Task Specification
Need Assessment & Planning
Sensor Fusion & Integration
$a^k_{i,min}$, ϵ
$\mathbf{x}_G$
Routing Algorithm
Control Policy
α
$\mathbf{x}$
$\bar{R}_{ij}(\mathbf{x})$, $\tilde{R}_{ij}(\mathbf{x})$
Radio Communication Modelling
$RSSI$
$\dot{x}_i$
Robot
x_i
$\hat{x}_i$
State Estimation
Individual Robots
Sensor observations y_i

Figure 3.7: Full system architecture. *Task Specification* here represents a generic spatial application defined by a convex task potential function $\Psi(\mathbf{x})$ while providing a stream of data to the operator. *Individual robot* components consist of the low-level robot control, estimation, and communication. A subsystem is available to build an online model of radio communication in the environment. Concurrent methods are developed for solving the routing and mobility control problem (3.19).

the operator and integrates robot observations and requirements to determine specifications that it passes on to the control module. These specifications come in the form of a potential function $\Psi(\mathbf{x})$ that must be minimized and communication rates $a^k_{i,\text{min}}$ that must be maintained at all times. The control module then determines control inputs $\dot{\mathbf{x}}(t)$ and network variables $\boldsymbol{\alpha}(t)$ that are conducive to task completion. It relies on estimates of achievable point-to-point rates $R_{ij}(\mathbf{x})$ which are provided by the communication modeling module.. Individual robots implement the control law $\dot{\mathbf{x}}(t)$ and route packets according to variables $\boldsymbol{\alpha}(t)$. Robots also take observations $y_i(t)$, e.g., a video feed, that they relay to task planning and perform position estimation $\hat{x}_i(t)$ that they feedback to the control block. Using available technologies for mapping, control, and state estimation, each robot estimates its position $\hat{x}_i(t)$ and controls its velocity $\dot{x}_i(t)$ with respect to a common known map of the environment; see e.g., [101].

This thesis continues by developing the techniques and algorithms necessary to implement the architecture in Fig 3.7 on an experimental system. We first address the issues of communication modeling in Chapter 4. We go on to propose control and planning algorithms that provide solutions to the concurrent routing and mobility problem in Chapter 5. For the scope of this work, we focus on straight-forward task specifications, e.g. a single leader node that must visit a series of waypoints while maintaining a minium rate of communication back to a fixed operating center.

Chapter 4

Communication Modeling & Mapping

The primary goal of communication modeling is to predict signal strength in order to infer communication capability between two nodes. The secondary objective is to provide a measurement model that can be used for sub-tasks such as relative localization and mapping. We begin by introducing radio signal propagation and its connection with wireless channel capacity. We then proceed by describing a series of candidate models for received signal strength with increasing complexity focusing on regression techniques for learning each model from measurements in an environment. Next, we compare the performance of these models with experimental data from two indoor environments with differing architecture. We conclude with a treatment of the secondary objective which is to understand if we can leverage communication models in order to provide relative localization.

4.1 Received Signal Strength and Channel Capacity

Received radio signal power is a complex multi–scale process that is a function of the distance from the source, shadowing due to obstacles, electromagnetic interference, and multi–path phenomena that arise as a result of reflections and refractions of the signal with the environment. While spatially and temporally averaged behavior can be fit to deterministic path loss models, small-scale fading due to multi–path can cause variations to received signal strength on the order of ±10 dB over small length scales of even a few centimeters. Though small–scale fading can be modeled by complex ray–tracing methods [34], we adopt a stochastic approach that assumes small–scale fading is drawn from a probability distribution [33].

Given our fundamental assumption that radio signal propagation is a stochastic process, the most basic models we can consider are stationary models which are purely a function of the distance between source and receiver – all environmental features are incorporated into the stochasticity of the model. However, this type of model presents difficulties as the size of the environment grows. As more environmental features are captured by the stochastic component, the model becomes more and more coarse. That is, it becomes more general and an underapproximation of the actual behavior for a particular point–to–point channel. Thus, we consider not only distance–dependent models, but also methods for *mapping* the radio signal propagation in large complex environments. This includes the considerations necessary to handle measurements are a function of position, i.e. $\mathbb{R}^2$, position and relative orientation, $SE(2)$, and point-to-point channels, $\mathbb{R}^2 \times \mathbb{R}^2$. In general, we describe the received signal strength at x_j when transmitted from x_i as a random variable with known mean and variance, $\bar{P}_R(x_i, x_j)$ and $\tilde{P}_R(x_i, x_j)$ respectively.

Ultimately, we seek to describe a model that relates node positions x_i, x_j with the supported rate of communication R_{ij} between them. Since it is well known that communication rate is a function of the bit-error-rate of a channel and that bit-error-rate is a function of the signal-to-noise ratio [93], we employ a cascaded model that first determines the received signal strength due to radio signal propagation, i.e. $P_R(x_i, x_j)$, and then the bit-error-rate and uses this as a measure of the supported communication rate over a channel.

From [93], bit error rate (p_b) and received power level $P_R(x_i, x_j)$ are a function of the modulation scheme but can be generically related by

$$p_b(P_R(x_i, x_j)) = \operatorname{erfc}\left(\sqrt{\frac{P_R(x_i, x_j)}{P_{N_0}}}\right) \tag{4.1}$$

where P_{N_0} is the noise power (we use N_0 to refer to the noise power in dBm on a logarithmic scale with respect to 1 mW) and $\operatorname{erfc}(x)$ is the complementary error function. Zigbee (802.15.4) employs a modulation scheme: O-QPSK with half-sine pulse shaping (MSK) with 16 channels spaced at 5 Mhz and has transmission rates of 250 kb/s [54].

The rate function for a channel from a node at x_i to another node at x_j is then

$$R_{ij}(x_i, x_j) \propto 1 - p_b(P_R(x_i, x_j)) \tag{4.2}$$

Since we know that received signal strength suffers from small–scale fading and must be considered as a random variable, it is necessary to consider $R_{ij}(x_i, x_j)$ as a random variable. Specifically, we will consider the statistics on R_{ij} – that is, the mean, $\bar{R}_{ij}$, and variance, $\tilde{R}_{ij}$. We can apply the delta method to compute the variance of the bit-error-rate function due to the variance on received power via its Taylor series expansion [49]

$$\tilde{R}_{ij}(P_R(x_i, x_j)) = \operatorname{var}\left[R_{ij}(P_R(x_i, x_j))\right] \approx \left(R'_{ij}(\bar{P}_R(x_i, x_j))\right)^2 \tilde{P}(x_i, x_j). \tag{4.3}$$

Figure 4.1 depicts the resulting distribution on channel reliability as a function of distance when using a standard log-normal model

$$\begin{aligned} \bar{P}_R(x_i, x_j) &= L_0 - 10n\log_{10}(\|x_1 - x_2\|) && (4.4)\\ \tilde{P}_R(x_i, x_j) &\sim \mathcal{N}(0, \sigma_{\mathcal{F}}^2). && (4.5) \end{aligned}$$

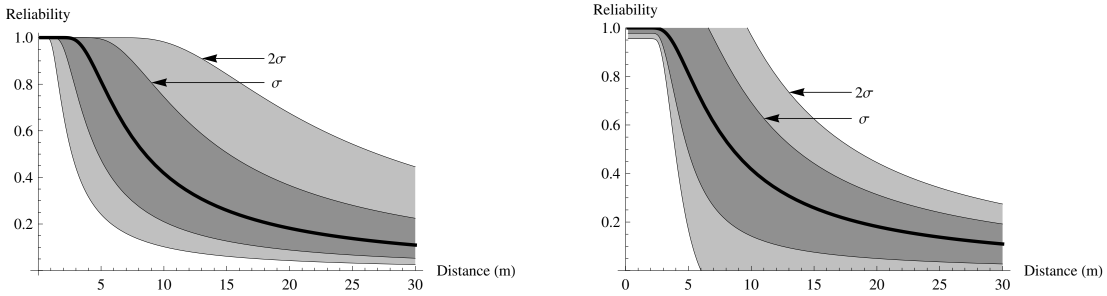
Note that (4.4) and (4.5) represent received power relative to 1 mW in the units of dBm and will be described in more detail in Section 4.3. We visualize the distribution by plotting the mean behavior $\bar{R}_{ij}(x_i, x_j)$ and several confidence intervals. The confidence levels in Fig. 4.1a are computed numerically. For each distance $d = \|x_i - x_j\|$, we plot the surfaces

$$R_{ij}\left(\bar{P}_R(x_i, x_j) - k\sqrt{\tilde{P}_R(x_i, x_j)}\right), \tag{4.6}$$

for $k = -2, -1, 0, 1, 2$. Figure 4.1b is based on the approximation from (4.3) and depicts

$$\bar{R}_{ij}(x_i, x_j) - k\sqrt{\tilde{R}_{ij}(x_i, x_j)}, \tag{4.7}$$

for $k = -2, -1, 0, 1, 2$. While modeling point–to–point channel capabilities as a normal random variable is an approximation, it is instrumental in later applications for robust communication. A topic of future work is the identification of more accurate probability distributions for point–to–point reliability.



(a) True Confidence Intervals

(b) Delta-method approximation

Figure 4.1: Model of the mapping from distance to link reliability shown with several confidence intervals due to the log-normal fading on received-signal power. While the delta-method provides a variance on reliability, there are some artifacts of approximation as can be seen when comparing confidence intervals in (a) and (b).

4.2 Gaussian Processes

As we consider environments of increasing complexity, it becomes difficult to predict received signal strength with distance-based models and we turn to spatial mapping techniques. In particular, we investigate the use of a Gaussian process (GP) which is a generalization of a typical Gaussian distribution and describes a distribution over functions. In fact, using a Gaussian process to model radio signal strength is specifically considered in [25], where Ferris et al. demonstrate the utility of Gaussian processes for robotic localization tasks and continue in [24] by addressing the simultaneous localization and mapping problem when receiving transmissions from multiple base stations. Our work differs from these approaches in that we are more interested in the predictive capabilities of the GP – specifically the ability to represent increasing uncertainty about unexplored regions of the environment [30].

Adopting the function–space view defined in [87], a GP describes a process $f(\mathbf{x})$ as a distribution over functions that is parameterized by a mean function $\mu(\mathbf{x})$ and covariance function $k(\mathbf{x}, \mathbf{x}')$ which are defined as

$$\mu(\mathbf{x}) = \mathbb{E}\left[f(\mathbf{x})\right], \tag{4.8}$$

$$k(\mathbf{x}, \mathbf{x}') = \mathbb{E}\left[(f(\mathbf{x}) - \mu(\mathbf{x}))(f(\mathbf{x}') - \mu(\mathbf{x}'))\right]. \tag{4.9}$$

The GP is then written as

$$f(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')). \tag{4.10}$$

For data–driven regression on radio signal strength, we consider (4.10) to be a prior distribution and incorporate training points $(\mathbf{X}_T, \mathbf{y}_T)$ with the interpretation that $\mathbf{y}_i = f(\mathbf{x}_i) + \mathcal{F}$, i.e. a noisy measurement model where random variations of the received signal strength on a log-scale are considered to be normally distributed with variance $\sigma^2_{\mathcal{F}}$. The goal of regression is then to determine a posterior GP that can be used to make predictions about sample points $\mathbf{X}_* = [\mathbf{x}_{i*}]$.

As a convention, define $K(\mathbf{X}, \mathbf{X}')$ to be the matrix of $k(\cdot, \cdot)$ evaluated pairwise across the elements of $\mathbf{X}$ and $\mathbf{X}'$. Furthermore, to make things more compact, define $\mathbf{K}_A = K(\mathbf{X}_A, \mathbf{X}_A)$ and $\mathbf{K}_{A,B} = K(\mathbf{X}_A, \mathbf{X}_B)$. For a single test point $\mathbf{x}_*$, we write $\mathbf{k}_*$ to represent the vector obtained by computing $k(\mathbf{x}_*, \cdot)$ for all points in $\mathbf{X}_T$. The joint distribution of measurements $\mathbf{y}$ and sample outputs $\mathbf{f}_* = f(\mathbf{X}_*)$ is then given by

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu(\mathbf{X}) \\ \mu(\mathbf{X}_*) \end{bmatrix}, \begin{bmatrix} \mathbf{K}_T + \sigma_{\mathcal{F}}^2 \mathbf{I} & \mathbf{K}_{T,*} \\ \mathbf{K}_{*,T} & \mathbf{K}_* \end{bmatrix}\right). \tag{4.11}$$

By conditioning on the observations, we can write the posterior predictive distribution as

$$\begin{aligned} \mathbf{f}_*|\mathbf{X}_*, \mathbf{X}_T, \mathbf{y}_T \sim \mathcal{N}\big(&\mu(\mathbf{X}*) + \mathbf{K}_{*,T}\left[\mathbf{K}_T + \sigma_{\mathcal{F}}^2\mathbf{I}\right]^{-1}(\mathbf{y}_T - \mu(\mathbf{X}_T)), \\ &\mathbf{K}_* - \mathbf{K}_{*,T}\left[\mathbf{K}_T + \sigma_{\mathcal{F}}^2\mathbf{I}\right]^{-1}\mathbf{K}_{T,*}\big) \end{aligned} \tag{4.12}$$

or, for a single test point $\mathbf{x}_*$, the *measurement* prediction as

$$\bar{\mathbf{y}}_* = \mu(\mathbf{x}*) + \mathbf{k}_*^{\mathrm{T}}\left[\mathbf{K}_T + \sigma_{\mathcal{F}}^2\mathbf{I}\right]^{-1}(\mathbf{y}_T - \mu(\mathbf{X}_T)), \tag{4.13}$$

$$\operatorname{var}[\mathbf{y}_*] = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^{\mathrm{T}}\left[\mathbf{K}_T + \sigma_{\mathcal{F}}^2\mathbf{I}\right]^{-1}\mathbf{k}_* + \sigma_f^2 \tag{4.14}$$

where $\bar{\mathbf{y}}_*$ and $\operatorname{var}[\mathbf{y}_*]$ give the mean and variance of the predicted measurement respectively. This is the distribution we are interested in as it predicts the observed signal strength of the channel.

So far, we have not explicitly defined $\mu(\mathbf{x})$ or $k(\mathbf{x}, \mathbf{x}')$. The prior for the mean function can be any parametric function of the input state $\mathbf{x}$ and it is somewhat standard in the Gaussian process literature to begin by considering a squared-exponential covariance function, i.e.,

$$k_{se}(x_1, x_2) = \sigma_k \exp\frac{-\|x_1 - x_2\|^2}{2\ell^2}, \tag{4.15}$$

to model a wide range of spatial processes. By adopting a stationary kernel, i.e. one that varies as a function of the difference between points, several of the calculations to determine the posterior are simplified. The key idea of the squared-exponential covariance function is that measurements are highly correlated within a characteristic length scale ℓ.

Examining the structure of $\operatorname{var}[\mathbf{y}_*]$, it is evident that prediction variance is controlled by σ_f^2 and the *position* of training measurements. Consequently, as we make predictions farther from the region of the environment where training is performed, the variance of the prediction will increase. This serves to represent our uncertainty about the behavior of the modeled process in regions where we have not applied training.

4.3 Candidate Models

Most work with signal strength mapping is conducted with the assumption of a static source so that the input space of the model is $\mathbb{R}^2$ and the source location is a fixed parameter in $\mathbb{R}^2$. However, in a general application, both nodes of a point–to–point channel will be mobile so that the true input space for mapping is $\mathbb{R}^2 \times \mathbb{R}^2$. Consequently, we will consider regression techniques where a single measurement consists of an received signal strength indicator (RSSI) y_i and tuple of positions in $\mathbb{R}^2$, $\mathbf{x}_i = (x_s, x_r) \in \mathbb{R}^4$. A set of N measurements is then given by $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, $\mathbf{y} = \{y_1, \ldots, y_N\}$. To match our experimental measurements and for simplicity of presentation, we denote received signal strength power in units of dBm, i.e., relative to 1 mW. We consider a series of models presented in order of increasing complexity to explore the tradeoffs that are implicit to both predictive performance and parameter estimation. We denote the unknown parameters of each model as the vector θ and the estimated parameters based on measurements as $\hat{\theta}$.

All of the models we examine consider fading in a stochastic manner. The idea of modeling fading power as a random variable is well accepted [74] and it is generally considered to be drawn from a log–normal, Rayleigh, or Rician distribution. Any of these distributions is parameterized based on an expected value and a single parameter representing the magnitude of the random fading. We assume a log–normal distribution so that fading power can be considered as an additive Gaussian on the log-scale that we will typically use for convenience.

M1: *Distance–dependent path loss with log–normal fading.* The most basic model we consider is a function solely of the distance between source x_s and receiver x_r. It is parameterized by L_0 which denotes the received power at 1 m, decay term n, and fading variance

$\sigma_{\mathcal{F}}^2$,

$$P_{R,1}(x_s, x_r) = \underbrace{L_0 - 10n\log_{10}(\|x_s - x_r\|)}_{\text{Path loss}} + \underbrace{\mathcal{F}}_{\text{Fading}}. \tag{4.16}$$

$\mathcal{F}$ models the small–scale fading as a normal random variable where $\mathcal{F} \sim \mathcal{N}(0, \sigma_{\mathcal{F}}^2)$. Additionally, note that L_0 aggregates the transmitted power, which we assumed to be fixed, and attenuation due to the amplifier–antenna system on both transmitter and receiver. Parameters of this model are $\theta_1 = (L_0, n, \sigma_{\mathcal{F}})$.

M2: *Distance–dependent path loss with log–normal fading and fixed attenuation for non line of sight (NLOS) channels.* If some knowledge of the environment's geometry is available, it is possible to explicitly model the radio signal attenuation for non-line-of-sight channels by introducing a term $W(x_s, x_r)$ that represents a fixed signal–strength attenuation. We introduce a function $\mathcal{V}(x_s, x_r)$ as an indicator for the existence of line of sight or *visibility* between two nodes at x_s and x_r. $\mathcal{V}(x_s, x_r) = 1$ when there is visibility and 0 otherwise so that

$$W(x_s, x_r) = \begin{cases} w & \text{if } \mathcal{V}(x_s, x_r) = 0, \\ 0 & \text{otherwise} \end{cases} \tag{4.17}$$

where w represents the fixed attenuation due to obstacles in the environment. Fixed attenuation due to the shadowing effects of obstacles is then incorporated into (4.16) so that

$$P_{R,2}(x_s, x_r) = L_0 - 10n\log_{10}(\|x_s - x_r\|) + W(x_s, x_r) + \mathcal{F}. \tag{4.18}$$

with $\theta_2 = (L_0, n, w, \sigma_{\mathcal{F}})$.

M3: *Distance–dependent path loss with models for log–normal fading that are dependent on line-of-sight and non-line-of-sight channels.* Extending the idea of (M2), we define a distance–dependent path loss model with log–normal fading where the parameters of the model are explicitly fit for line–of–sight and non–line–of–sight regimes so that

$$P_{R,3}(x_s, x_r) = L_0(x_s, x_r) - 10n(x_s, x_r)\log_{10}(\|x_s - x_r\|) + \mathcal{F}(x_s, x_r) \tag{4.19}$$

where

$$
\begin{aligned}
L_0(x_s, x_r) = & \begin{cases} L_0 & \text{if } \mathcal{V}(x_s, x_r) = 1, \\ L_{0,\mathrm{NLOS}} & \text{otherwise} \end{cases} \\
n(x_s, x_r) = & \begin{cases} n & \text{if } \mathcal{V}(x_s, x_r) = 1, \\ n_{\mathrm{NLOS}} & \text{otherwise.} \end{cases}
\end{aligned}
\tag{4.20}
$$

The random fading term in this model is a function of the channel's line-of-sight status. For example, if we assume a log-normal distribution then,

$$
\begin{aligned}
\mathcal{F}(x_s, x_r) \sim & \quad \mathcal{N}(0, \sigma_{\mathcal{F}}(x_s, x_r)), \\
\sigma_{\mathcal{F}}(x_s, x_r) = & \begin{cases} \sigma_{\mathcal{F}} & \text{if } \mathcal{V}(x_s, x_r) = 1, \\ \sigma_{\mathcal{F},\mathrm{NLOS}} & \text{otherwise.} \end{cases}
\end{aligned}
\tag{4.21}
$$

The parameters of this model are then $\theta_3 = (L_0, n, \sigma_{\mathcal{F}}, L_{0,\mathrm{NLOS}}, n_{\mathrm{NLOS}}, \sigma_{\mathcal{F},\mathrm{NLOS}})$. At the cost of complexity, this model can be extended to consider an arbitrary number of channel classes, e.g. to handle variations in obstacle material that affect the extent of shadowing or multi–path fading.

M4: *Gaussian process with naive uniform prior.* In the spirit of (M3), consider the scenario where we continually introduce new channel classes and accompanying parameters as the environment is explored. In the limit, this can be modeled by a Gaussian process. Models of radio signal propagation using a GP are able to capture a wide range of shadowing and environmental effects without prior knowledge of the environment. This contrasts with (M3) where we assume knowledge of the environment's geometric structure and that the geometry provides a good predictor for radio signal propagation.

Remembering that $\mathbf{x} = (x_s, x_r)$ we can restate (4.12), (4.13), and (4.14) to write the received power for a particular point–to–point link as the posterior GP

$$\begin{aligned} P_{R,4}(x_s, x_r) \sim \mathcal{GP}\big(&\mu(\mathbf{x}) + \mathbf{K}_{\mathbf{x},T}\left[\mathbf{K}_T + \sigma_{\mathcal{F}}^2\mathbf{I}\right]^{-1}(\mathbf{y}_T - \mu(\mathbf{X}_T)), \\ &\mathbf{K}_{\mathbf{x}} - \mathbf{K}_{\mathbf{x},T}\left[\mathbf{K}_T + \sigma_{\mathcal{F}}^2\mathbf{I}\right]^{-1}\mathbf{K}_{T,\mathbf{x}} + \sigma_{\mathcal{F}}^2\mathbf{I}\big). \end{aligned} \tag{4.22}$$

For (M4), we will temporarily suspend any prior knowledge we have of radio signal propagation and set $\mu(\mathbf{x}) = -90$ dBm to coincide with the minimum threshold of the radios in our experimental testbed – making the prior assumption that communication is *not* possible between two arbitrary points. For processes where the behavior is correlated at multiple length scales, one can build more complicated covariance functions based on the sum of several squared exponential functions with different length parameters.

We choose a covariance function that is the sum of two squared exponentials

$$k(\mathbf{x}, \mathbf{x}') = \sigma_{k,1}^2 \exp\frac{-d(\mathbf{x}, \mathbf{x}')^2}{2\ell_1^2} + \sigma_{k,2}^2 \exp\frac{-d(\mathbf{x}, \mathbf{x}')^2}{2\ell_2^2}. \tag{4.23}$$

The intent is that one length scale ℓ_1 is longer and represents path loss components of the process while the other length scale ℓ_2 relates to shadowing due to obstacles in the environment. We rely on a non-euclidian distance function $d(\cdot, \cdot)$ to model our assumption that channels are symmetric. While it may turn out that this assumption is incorrect, it serves to significantly reduce the sampling burden in the training phase. We define the distance function between points $\mathbf{x} = (x^s, x^r)$ and $\mathbf{x}' = (x^{s'}, x^{r'})$ as

$$d(\mathbf{x}, \mathbf{x}') = \min\left\{ \left\| \begin{pmatrix} x^s \\ x^r \end{pmatrix} - \begin{pmatrix} x^{s'} \\ x^{r'} \end{pmatrix} \right\|, \left\| \begin{pmatrix} x^s \\ x^r \end{pmatrix} - \begin{pmatrix} x^{r'} \\ x^{s'} \end{pmatrix} \right\| \right\}. \tag{4.24}$$

Given our choice of $\mu(\mathbf{x})$ and $k(\mathbf{x}, \mathbf{x}')$, the so–called hyperparameters of the GP used to represent (M4) are $\theta_4 = (\sigma_{\mathcal{F}}^2, \sigma_{k,1}^2, \ell_1, \sigma_{k,2}^2, \ell_2)$.

M5: *Gaussian process with model–based prior.* While it is feasible to consider a dense but limited sampling of $\mathbb{R}^2$ as a single robot explores the environment, it is impractical to make this assumption for the $\mathbb{R}^2 \times \mathbb{R}^2$ space of all point–to–point links that we are interested in. As a result, large quantities of training data are necessary before (M4) can make useful predictions. We address this difficulty by incorporating the basic path–loss model used in (M1) into the mean function prior. In fact, any parametric model, e.g., (M1) – (M3), can be used as the prior. Then,

$$\mu(\mathbf{x}) = L_0 - 10n\log_{10}(\|x_s - x_r\|). \tag{4.25}$$

Consequently, the covariance function can be simplified to be

$$k(\mathbf{x}, \mathbf{x}') = \sigma_{k,1}^2 \exp\frac{-d(\mathbf{x}, \mathbf{x}')^2}{2\ell_1^2} \tag{4.26}$$

since it only needs to take into account shadowing due to obstacles in the environment. With the introduction of a parameterized prior, the hyperparameters of the GP in (M5) are $\theta_5 = (\sigma_{\mathcal{F}}^2, \sigma_{k,1}^2, \ell_1, L_0, n)$. The posterior for (M5) has the same form as (4.22), i.e.

$$\begin{aligned} P_{R,5}(x_s, x_r) \sim \mathcal{GP}\big(&\mu(\mathbf{x}) + \mathbf{K}_{\mathbf{x},T}\left[\mathbf{K}_T + \sigma_{\mathcal{F}}^2\mathbf{I}\right]^{-1}(\mathbf{y}_T - \mu(\mathbf{X}_T))\,, \\ &\mathbf{K}_{\mathbf{x}} - \mathbf{K}_{\mathbf{x},T}\left[\mathbf{K}_T + \sigma_{\mathcal{F}}^2\mathbf{I}\right]^{-1}\mathbf{K}_{T,\mathbf{x}}\big) \end{aligned} \tag{4.27}$$

where $\mu(\mathbf{x})$ and $k(\mathbf{x}, \mathbf{x}')$ are redefined in (4.25), (4.26). The idea of (M5) is that once a coarse model for path loss and small–scale fading has been determined, reasonable predictions can be made anywhere in the space and local deviations are incorporated into the posterior GP by the covariance function.

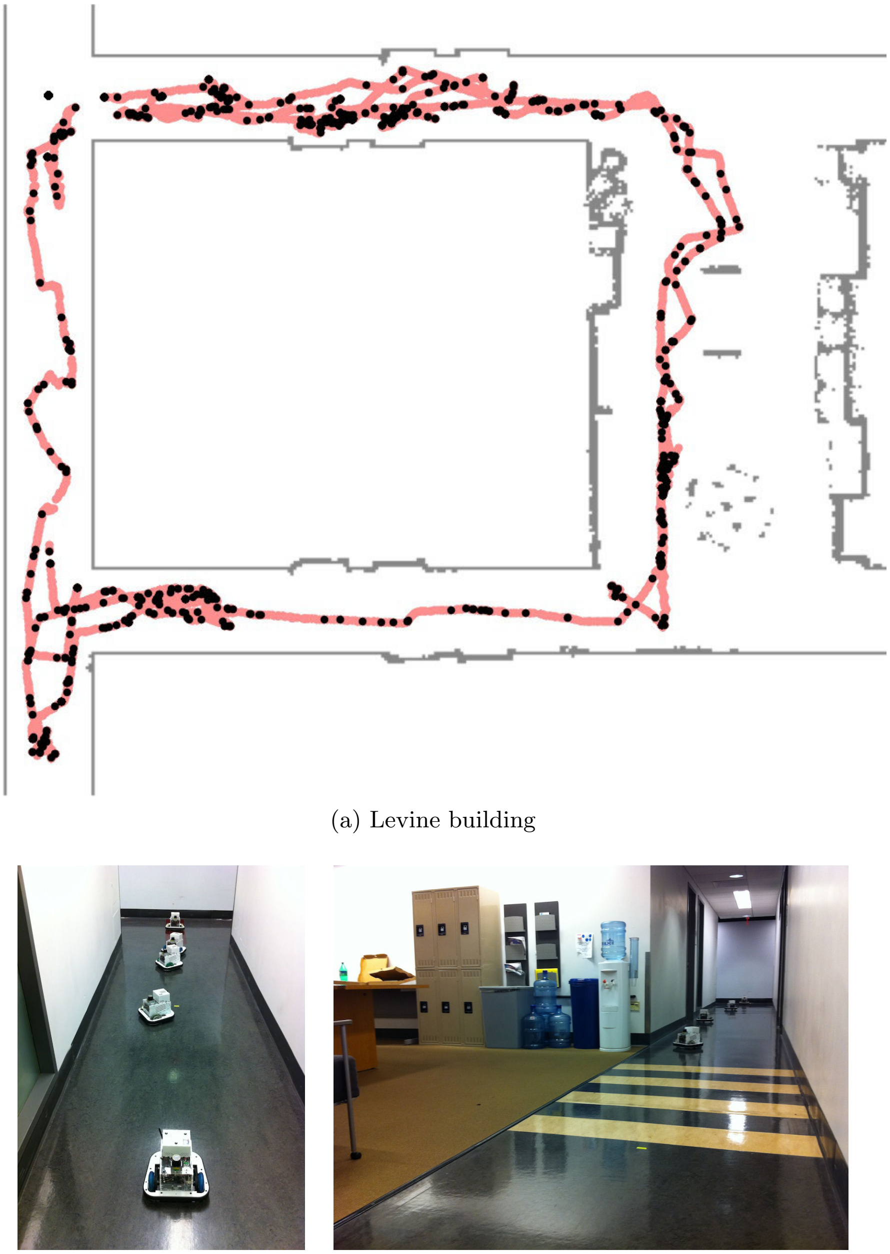
4.4 Experimental Analysis and Evaluation

Ultimately we are interested in online estimation and adaptation to obtain accurate models of received signal strength in an unknown environment. However, in order to compare the basic representations or models we have proposed above, we analyze each model's ability to incorporate measurements and then make predictions in a real environment. To demonstrate the generalization of each model, we perform this comparison in two indoor environments of drastically different design and materials on the University of Pennsylvania campus. With construction occurring in 1996 and 2003, the Levine building, depicted in Fig. 4.2, offers modern construction – interior walls are primarily made up of wood or metal framing with drywall. The Towne building, built in 1903–1906 and depicted in Fig. 4.3, offers drastically different construction materials including brick and concrete walls.

We collect data about the point–to–point links within a team of n mobile robots as they execute a pre-planned deployment within the desired workspace. Per the testbed description in Section 2.2, each robot is capable of self–localization in the environment and communication with its neighbors via a 2.4 GHz *Zigbee* radio. This means that at each measurement round, the system can collect on the order of n^2 measurements in the desired $\mathbb{R}^2 \times \mathbb{R}^2$ input space. Each robot embeds its current pose x^s in the world–frame into a message packet and broadcasts to its neighbors at a rate of 5 Hz. On arrival, the receiving robot includes its current pose x^r and the received signal strength y and logs the measurement. After an experimental trial, signal strength measurements are aggregated into a single dataset of all measurements $\{\mathbf{X}, \mathbf{y}\}$. We use a subset of the data $\{\mathbf{X}_T, \mathbf{y}_T\}$ to train each model. Training entails least–squares parameter estimation for (M1)–(M3) and normal Gaussian process regression with *ad hoc* selection of maximum likelihood hyperparameters for (M4), (M5).

Least–squares parameter estimation for models (M1)–(M3) will implicitly maximize their log-likelihood. For the Gaussian process models, hyperparameters are chosen by maximizing the log–marginal likelihood of the training data

$$\begin{aligned}\log p(\mathbf{y}_T|\mathbf{X}_T, \theta) = &-\frac{M}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{K}_T + \sigma_{\mathcal{F}}^2| \\ &-\frac{1}{2}(\mathbf{y}_T - \mu(\mathbf{x}_T))^{\mathrm{T}}\left(\mathbf{K}_T + \sigma_{\mathcal{F}}^2\mathbf{I}\right)^{-1}(\mathbf{y}_T - \mu(\mathbf{x}_T))\end{aligned} \tag{4.28}$$

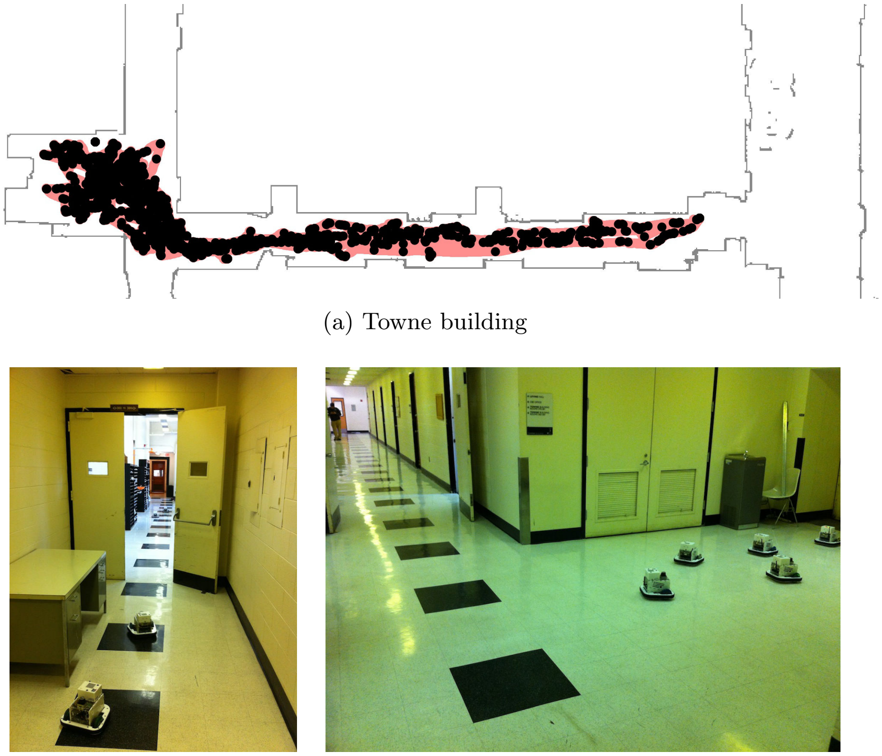


(a) Levine building

(b)

(c)

Figure 4.2: (a) Projection of full measurement dataset $\mathbf{X}$ onto $\mathbb{R}^2$ for the Levine building environment. Though measurements are made between pairs of robots in a 6 robot team, i.e., $\mathbb{R}^2 \times \mathbb{R}^2$ we display the measurement locations by projecting onto $\mathbb{R}^2$ and only displaying the location of the receiving node for each measurement. The uniform training set $\mathbf{X}_T$ is determined by choosing $1,000$ of the measurements – displayed here as black points. (b) and (c) depict snapshots from the environment



(a) Towne building

(b)

(c)

Figure 4.3: (a) Projection of full measurement dataset $\mathbf{X}$ onto $\mathbb{R}^2$ for the Towne building environment. Though measurements are made between pairs of robots in a 6 robot team, i.e., $\mathbb{R}^2 \times \mathbb{R}^2$ we display the measurement locations by projecting onto $\mathbb{R}^2$ and only displaying the location of the receiving node for each measurement. The uniform training set $\mathbf{X}_T$ is determined by choosing $1,000$ of the measurements – displayed here as black points. (b) and (c) depict snapshots from the environment

where M is the number of training points in $\mathbf{X}_T$. It is possible to compute the gradient of this function and employ standard gradient-ascent algorithms to find hyperparameters that achieve local maxima of the log–marginal likelihood. A complexity difficulty arises when considering GP–based regression on the large datasets we encounter (e.g. a system of 6 robots broadcasting messages at 5 Hz can capture up to $10,000$ measurements in less than a minute). In fact, standard GP regression with M training points requires the inversion of an $M \times M$ matrix which happens in $\mathrm{O}\big(M^3\big)$ time. There are a number of modern approaches to sparse approximations to GP regression [13, 94, 104, 84]. We employ the sparse pseudo-input Gaussian process (SPGP) method proposed in [94] to make Gaussian process regression and prediction tractable on the extremely large datasets we capture.

The log-likelihood, or evidence, of each model with parameters based on training measurements $\{\mathbf{X}_T, \mathbf{y}_T\}$ is then computed for the full dataset $\{\mathbf{X}, \mathbf{y}\}$. To make this a fair comparison between the Gaussian process and parametric models, we make the following standardizations. First, we assume that each measurement is independently distributed. For all models (M1)–(M5), we note that the received power at a single point can be represented as a normally distributed random variable

$$P_{R,i}(x_s, x_r) \sim \mathcal{N}(\bar{P}_{R,i}(x_s, x_r), \tilde{P}_{R,i}(x_s, x_r)) \tag{4.29}$$

with mean $\bar{P}_{R,i}(x_s, x_r)$ and variance $\tilde{P}_{R,i}(x_s, x_r)$. Then, we define the likelihood for each model i based on N measurements $\{\mathbf{X}, \mathbf{y}\}$, training $\{\mathbf{X}_T, \mathbf{y}_T\}$, and estimated parameters $\hat{\theta}_i$ to be

$$\begin{aligned} \mathcal{L}(\mathrm{M}i) &= \log \mathrm{P}\left[\mathbf{y} \,|\, \mathbf{X}, \mathbf{X}_T, \mathbf{y}_T, \mathrm{M}i, \hat{\theta}_i\right] \\ &= -\frac{N}{2}\log 2\pi - \sum_{j=1}^{N} \frac{\log \tilde{P}_{R,i}(x_{s,j}, x_{r,j})}{2} - \sum_{j=1}^{N} \frac{\left(y_j - \bar{P}_{R,i}(x_{s,j}, x_{r,j})\right)^2}{\tilde{P}_{R,i}(x_{s,j}, x_{r,j})}. \end{aligned} \tag{4.30}$$

To normalize, we also consider the average likelihood $\bar{\mathcal{L}}(\mathrm{M}i) = \frac{1}{N}\mathcal{L}(\mathrm{M}i)$. Models with larger likelihood do a better job of explaining the captured measurements. By training on a subset of the data and then computing the likelihood over the entire set of measurements in a given environment, we test for each model's ability to provide predictions throughout the environment.

We consider two testing paradigms that affect the choice of training subset $\{\mathbf{X}_T, \mathbf{y}_T\}$. In the first, we seek to identify a baseline comparison between models by uniformly choosing training measurements across the entire domain. In the second paradigm, we explicitly test the extrapolation capabilities of each model by training on a local subset of $\mathbf{X}$ chosen by uniformly sampling measurements taken in the initial phases of a deployment.

Uniform Training Set

There are $N = 98,454$ measurements in the Levine dataset. We choose $N_T = 1,000$ training points such that $\{\mathbf{X}_T, \mathbf{y}_T\}$ is uniformly distributed across the measurement domain using hierarchical clustering methods [16]. The estimated model parameters $\hat{\theta}$ found by least–squares regression on the training measurements $\{\mathbf{X}_T, \mathbf{y}_T\}$ for models (M1)–(M3) are given in Table 4.1. Likewise, the hyperparameters that lead to maximum likelihood for the GP–based models are given in Table 4.2. Finally, the likelihoods $\mathcal{L}(\text{M}i)$ for each model are given in Table 4.3.

	L_0	n	w	$L_{0,\text{NLOS}}$	n_{NLOS}	$\sigma^2_{\mathcal{F}}$	$\sigma^2_{f,\text{NLOS}}$
Results for Levine building							
(M1) *Path loss*	-50.6	2.75	-	-	-	40.5	-
(M2) *Fixed attenuation for walls*	-51.3	2.07	-7.58	-	-	31.6	-
(M3) *Piecewise parameters*	-52.7	1.78	-	-50.1	3.03	24.2	29.6
Results for Towne building							
(M1) *Path loss*	-52.6	2.51	-	-	-	37.9	-
(M2) *Fixed attenuation for walls*	-53.3	2.20	-4.7	-	-	34.8	-
(M3) *Piecewise parameters*	-53	2.25	-	-62.2	1.81	40.13	22.3

Table 4.1: Least–squares parameter estimation for models (M1)–(M3) based on a uniform training set with $N_T = 1,000$ points as depicted in Figs. 4.2 and 4.3.

	$\sigma^2_{\mathcal{F}}$	$\sigma^2_{k,1}$	ℓ_1	$\sigma^2_{k,2}$	ℓ_2	L_0	n
Results for Levine building							
(M4) *GP with constant prior*	8	93	5.75	14	0.25	-	-
(M5) *GP with (M1) prior*	15	44	2.25	-	-	-50.6	2.75
Results for Towne building							
(M4) *GP with constant prior*	13	93	5.25	13.5	0.25	-	-
(M5) *GP with (M1) prior*	14	41.5	1.75	-	-	-52.6	2.51

Table 4.2: Hyperparameters for Gaussian process models (M4)–(M5)

We can draw several conclusions from this data. First, note that as the model complexity is increased, the $\sigma^2_{\mathcal{F}}$ parameter representing fading variance decreases. This implies that as the model becomes more expressive, deviations in the underlying process can be explicitly modeled and are not subsumed into the randomness used to model small–scale

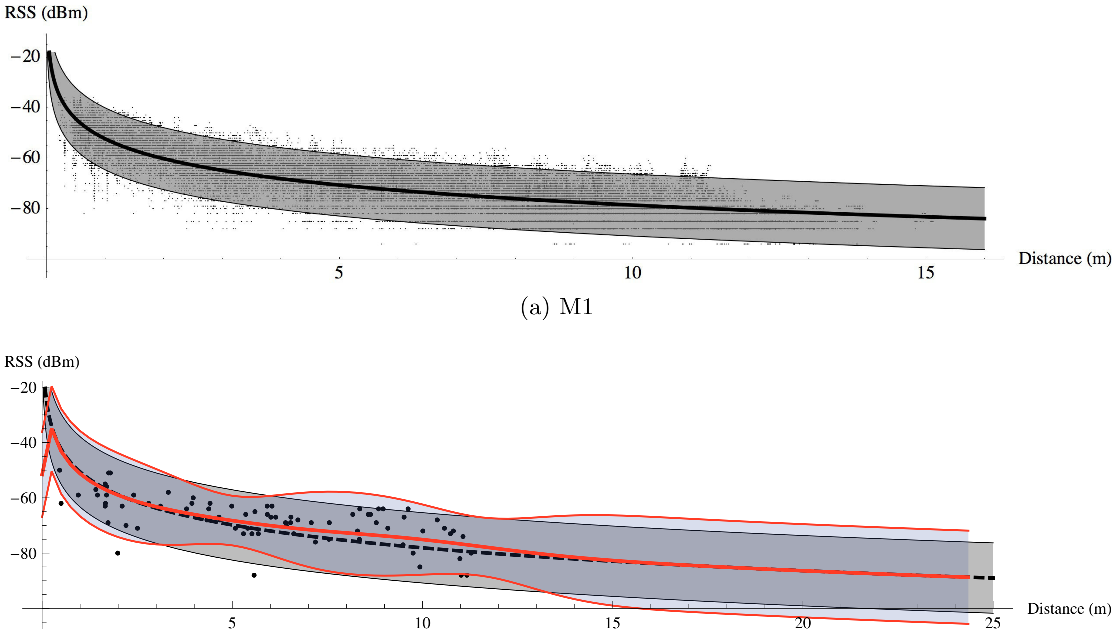
	M1	M2	M3	M4	M5
Levine uniform $\mathbf{X}_T$	-3.25	-3.13	-3.11	-3.14	-3.08
Towne uniform $\mathbf{X}_T$	-3.27	-3.24	-3.24	-3.21	-3.21

Table 4.3: Comparison of each model's average log–likelihood $\bar{\mathcal{L}}(\mathrm{M})$.

fading. The alternate explanation, that increasing model complexity leads to overfitting, is countered by the fact that model likelihood $\bar{\mathcal{L}}$ is computed using points outside the training set and also shows an inverse relationship with the sequence of increasing complexity models (M1) – (M5). However, while there is a trend of increasing model likelihood, the actual improvement in performance as measured by the likelihood is minimal. This is due to the fact that the local behavior of the underlying process is dominated by small–scale fading, see Fig. 4.4. For example, consider (M2) fit to the Levine building environment. Based on samples across the entire environment, the model extracts that an additional 5.8 dBm attenuation is present for non-line-of-sight channels while the variance associated with small–scale fading is $\sigma^2_{\mathcal{F}} = 34.4$. This amounts to a standard deviation of 5.87 dBm meaning that 32% of the time, a channel will instantaneously exhibit changes to received signal strength as dramatic as losing line of sight.

The increase in performance from (M1) to (M2) serves to demonstrate the benefit that can be derived from geometric information about the environment, e.g., to determine if a wireless channel is line-of-sight. However, if we examine the performance of (M2) with (M4) in Table 4.3, it is clear that the two models perform similarly. This occurs despite the fact that the Gaussian process-based (M4) incorporates no knowledge about the environment. Furthermore, this serves to highlight a feature of spatial mapping approaches based on the Gaussian process – they are able to represent un-modeled received signal strength phenomena.

Finally, a strength of the parametric models (M1) – (M3) is that a physics–based explanation can be assigned to the estimated parameters. However, this clearly breaks down when we consider the multi–modal fit of (M3) for the Levine building. In this case, the best fit for non–line–of–sight channels inflates the received signal strength at 1 m,



(a) M1

(b) M5

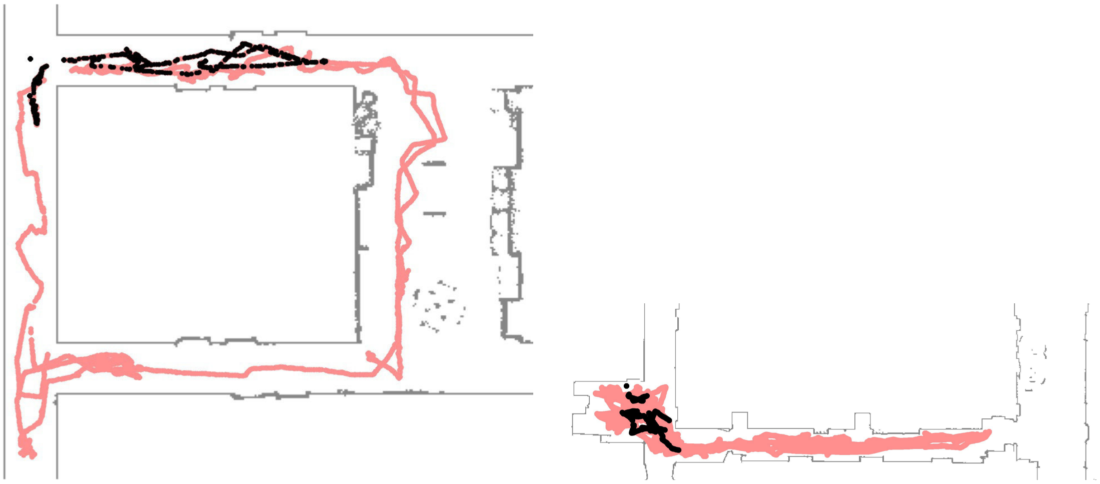
Figure 4.4: Predictive output of candidate models (M1) and (M5) for received signal strength in the Levine building. The models' output is visualized as a function of distance with an envelope depicting $\pm 2\sigma$ and is overlaid with actual measurement values. Since (M5) is spatially correlated, its prediction is made along the upper hallway in Fig. 4.2 and compared to nearby measurements. Note that predictions farther from measurements incur larger uncertainty.

i.e. $L_{0,\text{NLOS}} > L_0$, and compensates with increased fading exponent $n_{\text{NLOS}} = 3.49$. This behavior occurs because no short range non–line–of–sight measurements are captured in this dataset. The models (M1) – (M3) rely on deterministic parameters and have no representation for uncertainty due to lack of adequate measurements. This contrasts with the GP–based models (M4) and (M5) which explicitly model predictive uncertainty as a function of training measurements as depicted in Fig. 4.4b.

Similar to the results in the Levine building, the Towne building dataset includes $N = 181,376$ measurements with $N_T = 1,000$ training measurements across the domain of the environment as depicted in Fig. 4.3. Parameter estimates and model likelihoods are reported for this dataset in Tables 4.1, 4.2, and 4.3. We observe the same trend of increasing model likelihood with complexity. Additionally, examining the coarse model (M1) for the Levine and Towne building, it is evident that a single model with a large random component can be used as a conservative model across both environments. Furthermore, the nature of the GP–based models, where correlations are determined by local kernel functions, admits the best of both scenarios where a coarse deterministic prior is used in conjunction with local measurements to refine the model in different regions of the environment.

Local Training

The results presented above that rely on uniform training measurements across the entire measurement domain are useful to provide a baseline comparison of candidate models but do not adequately represent many applications. Here we alter the paradigm used to choose a subset of measurements for training to be more realistic. In both the Levine and Towne datasets, the team of six robots begin located in close proximity and are deployed into the environment. To produce a "local" training set, we perform uniform sampling of measurements captured in the first 100 s of each deployment. This results in a training set $\mathbf{X}_T$ for each environment as depicted in Fig. 4.5.



(a)

(b)

Figure 4.5: Projection of full measurement dataset $\mathbf{X}$ onto $\mathbb{R}^2$ for the (a) Levine and (b) Towne buildings. In order to display the data in $\mathbb{R}^2$, only the source location is depicted for each measurement. The local training set $\mathbf{X}_T$ is determined by choosing 2% of the measurements in a subset of the environment – displayed here as black points.

Following the same procedures as above, we find the best parameterizations of the candidate models based on these drastically reduced training sets as reported in Tables 4.4 and 4.5.

	L_0	n	w	$L_{0,\text{NLOS}}$	n_{NLOS}	$\sigma^2_{\mathcal{F}}$	$\sigma^2_{f,\text{NLOS}}$
Results for Levine – local							
(M1) *Path loss*	-52.4	2.07	-	-	-	43	-
(M2) *Fixed attenuation for walls*	-52.2	1.95	-3.9	-	-	41.3	-
(M3) *Piecewise parameters*	-52.4	1.9	-	-49.5	3.11	39.7	42.4
Results for Towne – local							
(M1) *Path loss*	-53.6	2.18	-	-	-	41.4	-
(M2) *Fixed attenuation for walls*	-54	1.95	-6.3	-	-	39	-
(M3) *Piecewise parameters*	-54	1.83	-	-54.2	2.69	35.7	57.1

Table 4.4: Least–squares parameter estimation for models (M1)–(M3) based on a local training set as depicted in Figs. 4.5a and 4.5b.

	$\sigma^2_{\mathcal{F}}$	$\sigma^2_{k,1}$	ℓ_1	$\sigma^2_{k,2}$	ℓ_2	L_0	n
Results for Levine – local							
(M4) *GP with constant prior*	8	93	5.75	14	0.25	-	-
(M5) *GP with (M1) prior*	15	44	2.25	-	-	-50.6	2.75
Results for Towne – local							
(M4) *GP with constant prior*	13	93	5.25	13.5	0.25	-	-
(M5) *GP with (M1) prior*	14	41.5	1.75	-	-	-52.6	2.51

Table 4.5: Hyperparameters for Gaussian process models (M4)–(M5) based on a local training set as depicted in Figs. 4.5a and 4.5b

Comparison of these parameters with those found based on the more "ideal" uniform training set reveals a systematic underapproximation of the received signal strength. Indeed, the model likelihoods reported in Table 4.6 clearly demonstrate reduced performance over their counterparts above.

A key feature of the GP–based model (M5) is its representation of increased predictive uncertainty away from measurements used for training. The local training set illustrates this as depicted in Fig. 4.6 where we train the model–based prior on short range line–of–sight measurements and then make a series of predictions. Since each prediction is given

	M1	M2	M3	M4	M5
Levine local $\mathbf{X}_T$	-3.46	-3.27	-3.13	-4.03	-3.45
Towne local $\mathbf{X}_T$	-3.31	-3.29	-3.29	-3.50	-3.44

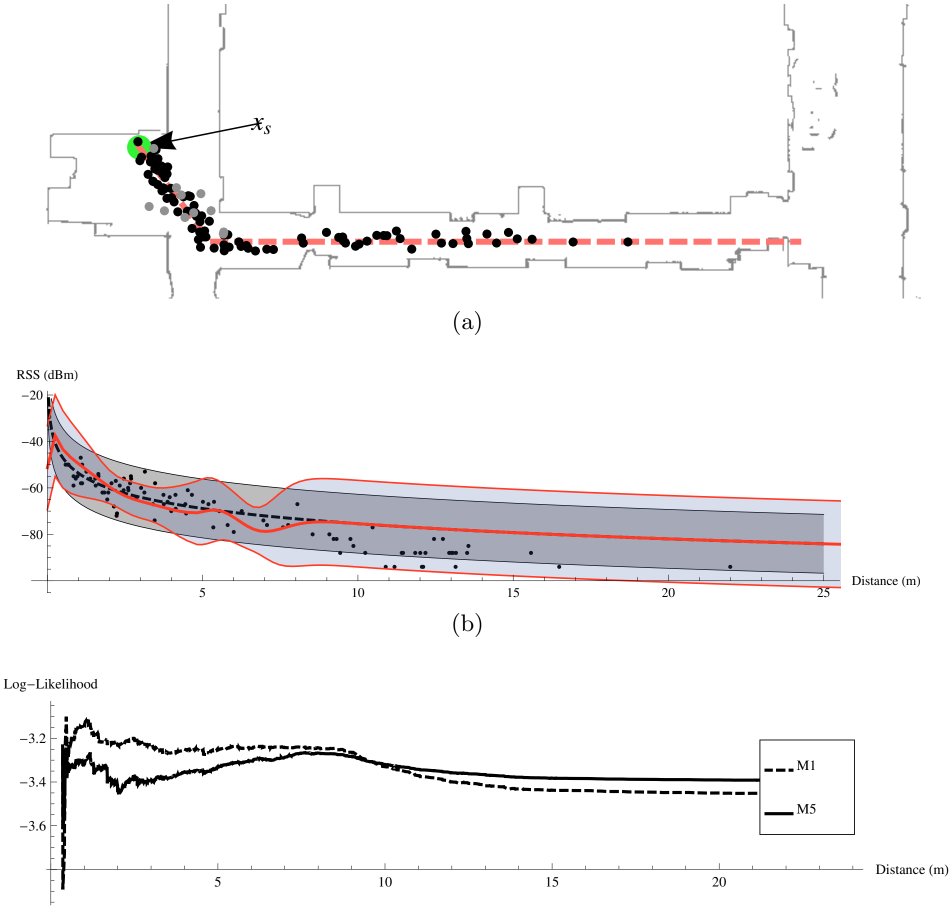
Table 4.6: Comparison of each model's average log–likelihood $\bar{\mathcal{L}}(\mathrm{M})$. Note that some models, e.g. (M3), cannot not be fit for training sets with no non–line–of–sight measurements such as *Towne local.*

by a normal distribution, the mean of that distribution matches the performance of (M1) but increased variance indicates decreased certainty about the received signal strength.

To specifically analyze the performance of the GP–based model with a path loss model as we make predictions farther from the set of training measurements, consider Fig. 4.6c which depicts the evolution of each model's likelihood based on validation measurements. By the time measurements are more than 10 m from the source, the performance of the GP–based model exceeds that of its path loss-based prior. Since the expected value of predictions from the GP model rely entirely on the prior here, the increased likelihood is due to the uncertainty representation that grows in regions where we have not captured measurements for training.

Summary

Our results, as expected, indicate that fitting performance increases with model complexity in cases where the training data is uniformly distributed across the domain of the final test data. This indicates that, in the absence of computational complexity concerns, the Gaussian process-based model (M5) is the best choice of model. However, the issue becomes more complicated when we consider the local training examples where models (M2) and (M3) perform better than GP–based models. In these particular examples, knowledge of the environment geometry allows for accurate extrapolation of performance in untrained regions of the environment. On the other hand, the GP–based models (M4) and (M5) capture model uncertainty in regions where training samples have not been collected.



(a)

(b)

(c)

Figure 4.6: Illustration of increased model uncertainty in unexplored regions for GP–based model (M5) (a) depicts the sample source location x_s, measurements used to train the model (gray points), measurements used to validate the model (black points), and the dashed line along which predictions are made. (b) is the predictive output of the GP with the darker dashed line and envelope depicting the prediction from (M1) while the solid line with lighter envelope depicts the prediction from the GP in (M5). (c) depicts the evolution of the likelihood of both models (M1) and (M5) as we consider points increasingly far from the source and training data.

In conclusion, model choice is not as straightforward as the complexity to performance relationship. However, it is true that the inclusion of accurate environment geometry does increase the extrapolation performance. The real benefit of the computational complexity incurred by GP-based models is twofold: first, the ability to represent signal strength features that are not predicted by geometry and second, the capability to report increased uncertainty away from training samples.

4.5 Localization

Our work towards communication modeling is primarily driven by the situational awareness problem statement from Section 3.2. However, a common alternate application of received signal strength modeling is relative localization of mobile agents [2, 3, 6, 7, 15, 22, 24, 25, 35, 37, 81, 91, 97, 112]. As demonstrated in Sections 4.3 and 4.4, small–scale fading in indoor environments presents itself as a significant source of noise that poses difficulties to any localization method that relies on received signal strength.

4.5.1 Radio Signal Source Localization

Here we consider the problem of a single static node broadcasting messages from an unknown position x_s. An individual or team of mobile agents with self–localization capabilities move through the environment receiving messages from x_s and measuring received signal strength. The goal is to determine an estimate of the source location $\hat{x}_s$ based on an inferred model of radio signal propagation. In practice, the estimation of a static node at an unknown position can be applied to a wide range of applications including search and rescue based on cellular phone signals or connectivity maintenance to a static access point.

In the context of using a Gaussian process to model received signal strength, the problem of localizing an unknown source location can be formulated as a maximum likelihood estimation on the parameters of the Gaussian process prior function [30]. Consider the candidate model (M5) which relies on a parametric function of path loss as a prior

$$P_R(x_s, x_r) = \underbrace{L_0 - 10n \log_{10}(\|x_s - x_r\|)}_{\text{Path loss}} + \underbrace{\mathcal{F}}_{\text{Fading}} . \tag{4.31}$$

In the communication mapping work developed in Section 4.3, we assume the parameters of this equation are $(L_0, n, \sigma_{\mathcal{F}})$ and that the source location for each received signal strength measurement is known. To perform source localization, we extend the parameters of (4.31) to be $(L_0, n, \sigma_{\mathcal{F}}, x_s)$ where $x_s \in \mathbb{R}^2$ represents the location of an unknown node that is broadcasting messages. Given a set of measurements $\{\mathbf{X}, \mathbf{y}\}$ where each measurement $\mathbf{x} \in \mathbf{X}$ is now in $\mathbb{R}^2$, we utilize gradient–based maximum likelihood estimation to find the source location x_s along with the other hyperparameters θ of the Gaussian process. Given this method for maximum likelihood estimation of the source location, we propose an active control strategy that drives a mobile agent to collect the samples necessary for accurate source localization.

Maximum likelihood source estimation

The Gaussian process model described in (M4) and (M5) denotes θ as the hyperparameters of the process, i.e. the parameters of the priors on mean and covariance functions. Here, we further decompose hyperparameters θ into those affecting the mean prior as θ_m and those affecting the covariance or kernel function as θ_k. For the source localization work, we assume accurate prior knowledge of the covariance hyperparameters θ_k. Given a mean function prior similar to (4.31), we denote it as an explicit function of parameters θ_m so that $\mu(x, \theta_m)$ incorporates the signal source location and parameters describing path loss in the environment. Then, the marginal log-likelihood function of the associated Gaussian process with N measurements $\{\mathbf{X}, \mathbf{y}\}$ is

$$\begin{aligned}\mathcal{L}(\theta_m) =& \log \mathrm{P}\left[\mathbf{y}|\mathbf{X}, \theta_m\right] \\ =& -\frac{1}{2}(\mathbf{y} - \mu(\mathbf{X}, \theta_m))^{\mathrm{T}}(\mathbf{K} + \sigma_{\mathcal{F}}^2 I)^{-1}(\mathbf{y} - \mu(\mathbf{X}, \theta_m)) \\ & -\frac{1}{2}\log\left|\mathbf{K} + \sigma_{\mathcal{F}}^2 I\right| - \frac{N}{2}\log 2\pi.\end{aligned} \tag{4.32}$$

To find the maximum likelihood estimate $\hat{\theta}_m$ of the mean function parameters θ_m, we consider the gradient of the log-likelihood $\partial\mathcal{L}(\theta_m)/\partial\theta_m$

$$\begin{aligned}\frac{\partial\mathcal{L}(\theta_m)}{\partial\theta_m} &= \frac{\partial \log \mathrm{P}\left[\mathbf{y}|\mathbf{X},\theta_m\right]}{\partial\theta_m} \\ &= (\mathbf{y}-\mu(\mathbf{X},\theta_m))^{\mathrm{T}}(\mathbf{K}+\sigma_{\mathcal{F}}^2 I)^{-1}\frac{\partial\mu(\mathbf{X},\theta_m)}{\partial\theta_m} \\ &= \left(\frac{\partial\mu(\mathbf{X},\theta_m)}{\partial\theta_m}\right)^{\mathrm{T}}(\mathbf{K}+\sigma_{\mathcal{F}}^2 I)^{-1}\left(\mathbf{y}-\mu(\mathbf{X},\theta_m)\right)\end{aligned} \tag{4.33}$$

and perform gradient ascent to find $\hat{\theta}_m$. Note that if we assume to know the kernel parameters, this optimization can be evaluated with a single computation of the inverse $(\mathbf{K}+\sigma_{\mathcal{F}}^2 I)^{-1}$ which avoids the typically prohibitive $O(N^3)$ cost when computing Gaussian process models for large datasets.

Furthermore, as shown in [1], we can use the second derivative of the log-likelihood

$$\begin{aligned}\frac{\partial^2\mathcal{L}(\theta_m)}{\partial\theta_{m,i,j}} = &\left(\frac{\partial^2\mu(\mathbf{X},\theta_m)}{\partial\theta_{m,i,j}}\right)^{\mathrm{T}}(\mathbf{K}+\sigma_{\mathcal{F}}^2 I)^{-1}\left(\mathbf{y}-\mu(\mathbf{X},\theta_m)\right) - \\ &\left(\frac{\partial\mu(\mathbf{X},\theta_m)}{\partial\theta_{m,i}}\right)^{\mathrm{T}}(\mathbf{K}+\sigma_{\mathcal{F}}^2 I)^{-1}\frac{\partial\mu(\mathbf{X},\theta_m)}{\partial\theta_{m,j}}\end{aligned} \tag{4.34}$$

to compute the Fisher information matrix

$$\mathcal{I}_{i,j} = -\frac{\partial^2\mathcal{L}(\theta_m)}{\partial\theta_{m,i,j}} \tag{4.35}$$

which, via the Cramer-Rao bound, provides a lower bound on the covariance of the maximum likelihood estimate $\hat{\theta}_m$. Thus, we describe the uncertainty of a maximum likelihood source estimate to be normally distributed, e.g.

$$\theta_m \sim \mathcal{N}(\hat{\theta}_m, \Sigma_m) \quad \text{where} \quad \Sigma_m = \mathcal{I}^{-1}. \tag{4.36}$$

Control law

Our focus is on using a mobile robot to continuously drive through an environment and sample the signal strength. We assume that signal strength mapping is the only task assigned to the robot so that it has full freedom to choose control directions that are most informative with respect to signal strength mapping. Here we take an exploration-exploitation approach similar in spirit to [51]. When the robots first enter an environment and have very few training samples, the estimate of $\hat{x}_s$ is poor or sometimes impossible to determine – random exploration is the best or only strategy. However, when an estimate is available for $\hat{x}_s$, the controller can choose directions that are more informative.

To formalize, we define two control inputs $\mathbf{u}_{explr}$ and $\mathbf{u}_{explt}$. The exploration input $\mathbf{u}_{explr}$ is chosen to locally reduce the entropy of the Gaussian process by following the gradient of the predictive variance (4.14) at the current position x_*, i.e.

$$\begin{aligned} \mathbf{u}_{explr}(x_*) &= \alpha_{explr} \nabla \mathrm{var}\left[y_*\right] \\ &= \alpha_{explr} \left(\frac{\partial k(x_*, x_*)}{\partial x_*} - 2 \frac{\partial \mathbf{k}_*}{\partial x_*}^{\mathrm{T}} \mathbf{Q} \mathbf{k}_* \right) \end{aligned} \tag{4.37}$$

where $\mathbf{Q} = (\mathbf{K} + \sigma_{\mathcal{F}}^2 \mathbf{I})^{-1}$. The exploitation input is chosen based on the current maximum likelihood estimate of the source location $\hat{x}_s$

$$\mathbf{u}_{explt}(x_*) = \alpha_{explt} \left(\hat{x}_s - x_*\right). \tag{4.38}$$

The intuition behind the exploit control input is that for estimating the parameters of a log-based function, samples must be collected near the source and away from the flat tail of the function where the local variation will be within that explained by small–scale fading.

Control inputs are chosen at discrete intervals indexed by k and applied for a fixed distance based on the characteristic length of the Gaussian process as determined by the length hyperparameter ℓ in the covariance function $k(\cdot, \cdot)$. This ensures that statistically independent samples of the received signal strength are collected. Each control input $\mathbf{u}_k$ is chosen with equal probability to be either $\mathbf{u}_{explr}(x(k))$ or $\mathbf{u}_{explt}(x(k))$. The first control input $\mathbf{u}_0$ is chosen entirely at random since there is no prior knowledge of $\nabla \text{var}[y_*]$ or $\hat{x}_s$. Control gains α_{explr} and α_{explt} are chosen based on velocity constraints of the mobile robot.

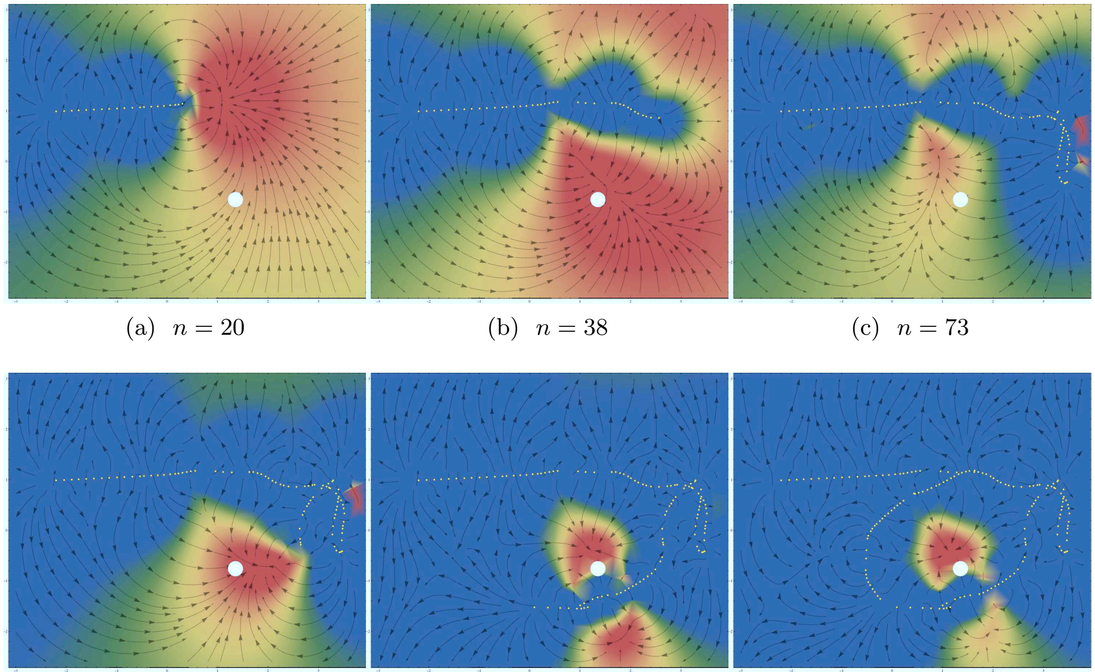
Experimental Results

We place a stationary *Scarab* robot in an open environment and have it broadcast packets via its *Zigbee* radio at 2 Hz. A mobile robot is started in the same obstacle-free region and follows the controller defined above as depicted in Fig. 4.7. After a random initial control direction in Fig. 4.7a, there is an estimated source location in the $+x$-direction and the $\mathbf{u}_{explt}$ action is selected in Fig. 4.7b. Figure 4.7c and 4.7d depict the $\mathbf{u}_{explr}$ control action while Figs. 4.7e and 4.7f depict execution of $\mathbf{u}_{explt}$. Note that as samples are collected, multiple local maxima of the likelihood function $\mathcal{L}(\theta_m)$, e.g. Fig. 4.7e, are resolved to the true maximum shown in Fig. 4.7f.

In a complex hallway environment, allowable control directions are limited, precluding the use of the control law defined in (4.37) and (4.38). Instead, we perform a constrained exploration by driving the length of the hallway as depicted in Fig. 4.8a. After collecting non–collinear data, there is a single maximum likelihood estimate of the source location $\hat{x}_s$ that closely approximates the true source location.

4.5.2 Bearing–based Localization

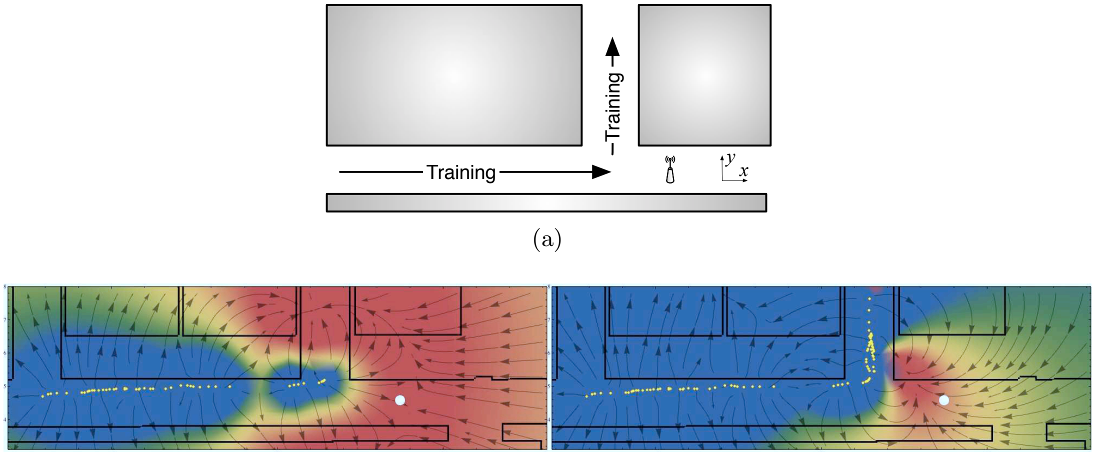
All of our signal strength modeling up until now has focused on the location and distance between the transmitting and receiving nodes while making the assumption that the antennas are isotropic. While antenna design typically strives to be isotropic in its sensitivity, real antennas will inevitably have some orientation dependence. In fact, any real antenna implementation will have a “lobe” structure where there exist bearings along which the



(a) $n = 20$ (b) $n = 38$ (c) $n = 73$

(d) $n = 85$ (e) $n = 112$ (f) $n = 155$

Figure 4.7: The evolution of the likelihood function $\mathcal{L}(\theta_m)$ with respect to the signal source location x_s after n samples. As the experiment progresses from (a)–(f), the measurements are incorporated into the Gaussian process, affecting the likelihood of the source location. The trial concludes when there is a clear global maximum of the likelihood function. Samples are represented by yellow points, the vector field depicts the gradient of $\mathcal{L}(\theta_m)$. The white circle in each figure represents the actual signal source location.

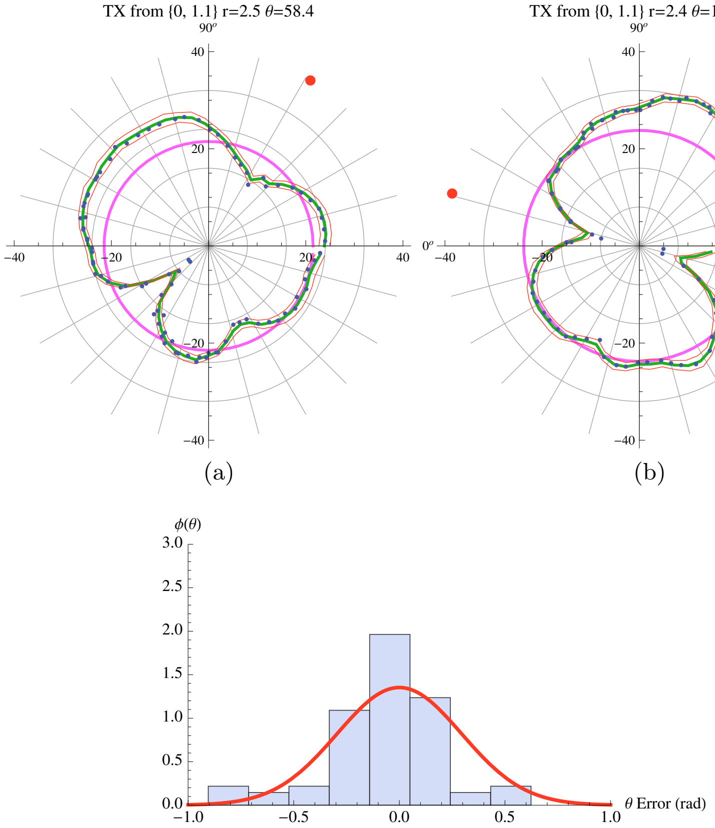


(a)

(b) $n = 48$ (c) $n = 130$

Figure 4.8: In complex environments such as (a), constraints on sampling make it difficult to accurately estimate the source location. In this example, a clear maximum in the likelihood function $\mathcal{L}(\theta_m)$ is not found until the robot turns the corner in (c). In (b) and (c), the actual source location is marked by a white disk.

antenna is much less sensitive as demonstrated in the experimental results depicted in Fig. 4.9.



(a)

(b)

(c)

Figure 4.9: Bearing estimator results. Figures 4.9a and 4.9b both depict raw data that enters the bearing estimation process. The magenta circle in each example illustrates $\bar{\psi}_{i,j}$ and the red dot indicates the actual bearing. Figure 4.9c depicts the estimator error with a histogram over 100 trials to justify our Gaussian approximation.

A generic two-lobe structure can be approximated on a dB scale by

$$\Psi_2(\theta) = 20\log_{10}\left(\left|\frac{-\cos\left(\frac{1}{2}\pi\cos\theta\right)}{\pi\cos\theta}\right|\right) + 20\log_{10}\pi + \eta \tag{4.39}$$

where η is a normalization term to account for the magnitude of the received signal at a particular location. More general lobe structures can be generated by composition of several parameterized versions of (4.39) where we indicate the number of lobes by the subscript of $\Psi(\theta)$.

Given these antenna profiles, our approach is to gather received signal strength data while rotating the antenna and identify the most likely bearing to the signal source. Let an individual RSSI measurement be $\psi_{i,j}(\theta_i)$ and assume that a relative bearing estimate is computed after sampling $\theta_i \in [0, 2\pi)$. Then, a function

$$\mathcal{E}_\ell(\phi_{i,j}) = \int_0^{2\pi} \left(\psi_{i,j}(\theta_i) - \Psi_\ell(\theta_i - \phi_{i,j})\right)^2 d\theta_i \tag{4.40}$$

describes the error of the collected measurements given a relative bearing $\phi_{i,j}$. The normalization parameter η is set $\eta = \max_{\theta_i \in [0,2\pi)} \psi_{i,j}(\theta_i)$. Based on empirical evidence, we attempt to fit both one and two-lobe models to the measured data. The best estimate for $\phi_{i,j}$ is then given by the minimum over $\mathcal{E}_1(\phi_{i,j})$ and $\mathcal{E}_2(\phi_{i,j})$,

$$\hat{\phi}_{i,j} = \underset{\phi_{i,j} \in [0,2\pi), \ell \in \{1,2\}}{\operatorname{argmin}} \mathcal{E}_\ell(\phi_{i,j}). \tag{4.41}$$

By relying solely on the relative deviations in signal strength during a rotation, we divorce ourselves from the necessity to learn environmental characteristics of radio signal propagation. However, the symmetric nature of antenna profiles dictate that this method of bearing measurement will suffer from directional ambiguity, motivating the development of tools for solving bearing–only localization problems in the presence of a π-ambiguity.

It is clear that for a cycle of robots making π-ambiguous relative bearing measurements as depicted in Fig. 4.10, the number of possible solutions is combinatoric. However, it is possible to discover the true configuration via two observations. First, we prove that for position-only localization, the combinatoric set of feasible solutions reduces to 2. Second, we show that it is possible to disambiguate feasible solutions through node motion, see e.g., [20].

Experimental Results

In order to properly characterize the noise model of our RSSI–based bearing estimator, we conduct over 100 trials where one robot transmits packets and another rotates in place so as

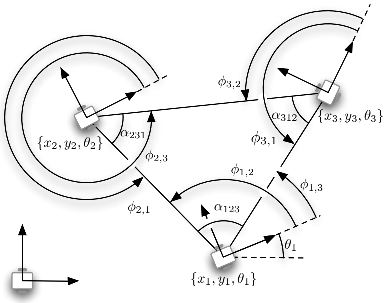
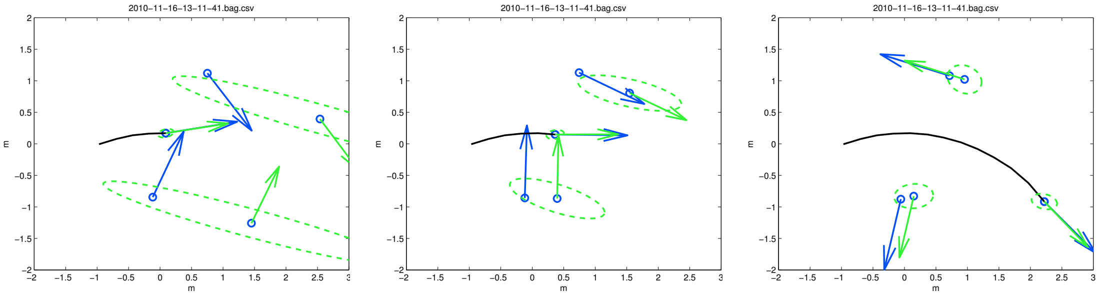


Figure 4.10: Notation for agent state x_i, y_i, θ_i, relative bearing measurements $\phi_{i,j}$, and the interior angles of a cycle α_i for three agents.

to compute the bearing measurement (4.41). For the purposes of this experiment, we rely on the ground truth data to properly resolve the direction ambiguity in the measurement. Figures. 4.9a and 4.9b show sample datasets. Figure 4.9c depicts a histogram of the error, demonstrating that it can be approximated by a Gaussian distribution with $\sigma_{z_{\hat{w}_i}} = 17°$.



(a) 5 Rounds

(b) 6 Rounds

(c) 12 Rounds

Figure 4.11: Snapshots of the three robot experiment demonstrating the convergence of the proposed filter with relative bearing estimates based on RSSI measurements from off-the-shelf *Zigbee* radios. The state estimate is provided relative to the root node which starts at $(-1, 0)$ m. Ground truth is depicted by blue arrows while the EKF estimate is indicated with green arrows and confidence ellipsoids.

A three robot experiment is conducted as depicted in Fig. 4.11 where two agents are stationary while a third agent, moves through an elliptical trajectory to ultimately disambiguate the state hypotheses and perform localization. In this trial, all robots belong to the same neighborhood so that relative bearing measurements are made between all pairs of robots resulting in fast convergence of team-state depicted in Fig. 4.12. Observe that after only 7 measurement rounds, the team obtains a formation that is nearly consistent with scale of 0.87 and has a mean–squared-error (MSE) over the interior angles and relative position of $8 \times 10^{-3} rads^2$ and 0.027 m^2. After completing 12 measurement rounds, the scale is 0.92 with MSE values being 1×10^{-3} rads^2 and 0.033 m^2. The interior angle error is used as a metric to capture the scale–free convergence of our method.

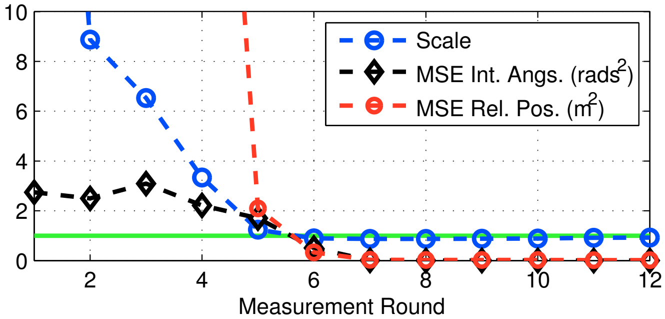


Figure 4.12: The scale, MSE of formation interior-angles, and relative position with respect to the root node in the experimental trial depicted in Fig. 4.11.

4.6 Summary

In this chapter, we have thoroughly explored several methods for communication modeling and its applications for teams of mobile robots. Our primary focus was the presentation of several candidate models for received signal strength and extensive experimental results to compare the performance of these models. We have also explored the application of models for received signal strength to the problem of relative localization for a team of mobile robots.

The results in this chapter can be distilled into several specific points. First, geometric knowledge of the environment, e.g., a map built from laser range scans, provides information that effectively improves the quality of received signal strength predictions. In cases where this geometric map is unavailable or fails to adequately capture the full nature of radio signal propagation, Gaussian proceses can serve as a framework for spatial mapping of point-to-point wireless channels. GP-based methods can model a wide range of signal strength phenomena and have the ability to represent increased uncertainty in untrained or unexplored regions of the environment. Further, we demonstrate that the

Gaussian process model for received signal strength can be readily applied to the localization problem by examining the marginal likelihood of the model with respect to an unknown source location. Finally, though we spend considerable effort developing and experimentally verifying models that ignore the relative orientation of radios, it is the case that real antennas are rarely isotropic and, in fact, demonstrate significant variation in their performance based on the orientation to the signal source. We demonstrate that this variation is highly correlated such that we can build a relative localization method for teams of robots based on purely relative information. That is, we avoid the complexities of learning a global model for radio signal propagation.

In aggregate, we have defined a framework in which we can capture point-to-point measurements of received signal strength with a team of mobile robots in a complex indoor environment, extract a model or map of received signal strength, and use this to predict the supported rate between robots at arbitrary points in the environment. The prediction of point-to-point communication rate consists of both a mean and variance and thus provides an essential capability to the development of connectivity control algorithms that are focused on maintaining specific rates of end-to-end communication.

Chapter 5

Concurrent Mobility Control & Robust Routing

A drawback of our formulation to the situational awareness task in (3.19) is the difficulty of ensuring that the end-to-end rate constraints in (3.18) are satisfied. As seen in (3.17), rates $a_i^k(\boldsymbol{\alpha}, \mathbf{x})$ depend on the link reliabilities $R_{ij}(\mathbf{x})$, which are difficult to estimate as demonstrated in Chapter 4. The challenge here is that $\bar{R}_{ij}(\mathbf{x})$ estimates are needed not only for configuration $\mathbf{x}(t)$, but for nearby configurations where the robots will move. The high variability of wireless channels makes $\bar{R}_{ij}(\mathbf{x}(t))$ a poor predictor of $R_{ij}(\mathbf{x})$ even if $\mathbf{x}$ is close to $\mathbf{x}(t)$. In formal terms, uncertainty of channel estimates means that the variances $\tilde{R}_{ij}(\mathbf{x})$ of channel estimates $\bar{R}_{ij}(\mathbf{x})$ are large for potential future positions $\mathbf{x}$. Based on the models developed in Chapter 4, we assume that the communication modeling algorithm provides mean, $\bar{R}_{ij}(\mathbf{x})$, and variance, $\tilde{R}_{ij}(\mathbf{x})$, estimates for arbitrary network configurations $\mathbf{x}$.

Thus, we seek to redefine (3.19) in a manner that takes into account the probabilistic formulation of channel rates. The important observation here is that if point-to-point link rates become random, so do the rates $a_i^k(\boldsymbol{\alpha}, \mathbf{x})$ of end-to-end communication flows. Consequently, it is not possible to guarantee satisfaction of the constraints in (3.18).

Rather, we introduce a reliability tolerance ϵ and require that for all i and k,

$$\mathrm{P}\left[a_i^k(\boldsymbol{\alpha}, \mathbf{x}) \geq a_{i,\min}^k\right] \geq \epsilon. \tag{5.1}$$

That is, we require that the end-to-end link between all sources i and the destinations of all corresponding flows k exceed their minimum required level of service with probability larger than ϵ. As in the case of the rate requirement in (3.18), we can satisfy (5.1) by controlling $\boldsymbol{\alpha}$ and $\mathbf{x}$.

In order to robustly satisfy the networking constraints in (5.1), the concurrent routing and mobility problem (3.19) is replaced by

$$\begin{aligned} \min_{\boldsymbol{\alpha}(t), \dot{\mathbf{x}}(t), t \in [0, t_f]} \quad & \Psi(\mathbf{x}(t_f)) \\ \text{subject to} \quad & \mathrm{P}\left[a_i^k(\boldsymbol{\alpha}, \mathbf{x}) \geq a_{i,\min}^k\right] \geq \epsilon \\ & \mathbf{x}(t) = \mathbf{x}(0) + \int_0^t \dot{\mathbf{x}}(u)\ du \end{aligned} \tag{5.2}$$

The problem formulation in (5.2) inherits some standard complications from the control formulation in (3.16). The problem is infinite dimensional and due to, e.g., the presence of obstacles, not convex. The concurrent optimization in (5.2) is further complicated by the entanglement of the routing and mobility problems. We deal with this entanglement by fixing $\mathbf{x}$ and selecting $\boldsymbol{\alpha}$ in a manner that optimizes the reliability $\mathrm{P}\left[a_i^k(\boldsymbol{\alpha}, \mathbf{x}) \geq a_{i,\min}^k\right]$. We follow up with local and global searches for positions $\mathbf{x}$ that minimize $\Psi(\mathbf{x})$ while maintaining communication reliabilities above the ϵ threshold.

5.1 Robust Routing

The major difficulty in solving (3.19) is the uncertainty in achievable transmission rates between nearby agents. Assuming that actual channels $R_{ij}(\mathbf{x})$ coincide with their estimates $\bar{R}_{ij}(\mathbf{x})$ may result in a drastic difference between predicted and actual end-to-end rates. A simple way to account for the uncertainty in $R_{ij}(\mathbf{x})$ is to artificially decrease

$\bar{R}_{ij}(\mathbf{x})$ to reduce the likelihood of having actual rates less than the predicted value. A better way to account for this uncertainty is to recall that end-to-end failures rather than point-to-point failures are relevant. In fact, it is possible to exploit spatial redundancy to devise robust routes that guarantee small changes in end-to-end rates despite large variability in $R_{ij}(\mathbf{x})$ [111].

To develop robust routing algorithms, we start by noticing that computing the probability in (5.1), which is part of the problem formulation in (5.2), necessitates modeling the probability distribution of $a_i^k(\boldsymbol{\alpha}, \mathbf{x})$. This is difficult in general. However, if we explicitly consider the stochastic model of point-to-point links via their means and variances, we can compute the mean and variance of end-to-end rates $a_i^k(\boldsymbol{\alpha}, \mathbf{x})$ as

$$\bar{a}_i^k(\boldsymbol{\alpha}, \mathbf{x}) := \mathbb{E}\left[a_i^k(\boldsymbol{\alpha}, \mathbf{x})\right] = \sum_j \alpha_{ij}^k \bar{R}_{ij}(\mathbf{x}) - \sum_{j \notin \text{dest}(k)} \alpha_{ji}^k \bar{R}_{ij}(\mathbf{x}), \tag{5.3}$$

$$\tilde{a}_i^k(\boldsymbol{\alpha}, \mathbf{x}) := \text{var}\left[a_i^k(\boldsymbol{\alpha}, \mathbf{x})\right] = \sum_j (\alpha_{ij}^k)^2 \tilde{R}_{ij}(\mathbf{x}) + \sum_{j \notin \text{dest}(k)} (\alpha_{ji}^k)^2 \tilde{R}_{ij}(\mathbf{x}). \tag{5.4}$$

A substitution for the probability in (5.1) is the difference between $a_i^k(\boldsymbol{\alpha}, \mathbf{x})$ and its mean $\bar{a}_i^k(\boldsymbol{\alpha}, \mathbf{x})$ normalized by its standard deviation $\sqrt{\tilde{a}_i^k(\boldsymbol{\alpha}, \mathbf{x})}$. Indeed, for any probability distribution, we can invoke Chebyshev's inequality to claim that

$$\frac{\bar{a}_i^k(\boldsymbol{\alpha}, \mathbf{x}) - a_{i,\min}^k}{\sqrt{\tilde{a}_i^k(\boldsymbol{\alpha}, \mathbf{x})}} \geq \sqrt{\frac{1}{\epsilon}}. \tag{5.5}$$

is a sufficient condition for satisfying (5.1). Using specific assumptions on the distribution of $R_{ij}(\mathbf{x})$ tighter bounds can be obtained. If, e.g., we assume that $R_{ij}(\mathbf{x})$ has a Gaussian distribution, then (5.1) is equivalent to

$$\frac{\bar{a}_i^k(\boldsymbol{\alpha}, \mathbf{x}) - a_{i,\min}^k}{\sqrt{\tilde{a}_i^k(\boldsymbol{\alpha}, \mathbf{x})}} \geq \Phi^{-1}(\epsilon) \tag{5.6}$$

where $\Phi^{-1}(\epsilon)$ is the inverse of the normal distribution's cumulative distribution function. The idea in robust routing algorithms is to reduce end-to-end uncertainty by taking advantage of spatial redundancy. To exploit spatial redundancy it is necessary to split traffic among various different routes. Indeed, (5.6) is satisfied by either increasing the mean $\bar{a}_i^k(\boldsymbol{\alpha}, \mathbf{x})$ or decreasing the variance $\tilde{a}_i^k(\boldsymbol{\alpha}, \mathbf{x})$. Since the mean is a linear function of $\boldsymbol{\alpha}$, traffic splitting has a minor effect on $\bar{a}_i^k(\boldsymbol{\alpha}, \mathbf{x})$. The variance is a quadratic function of $\boldsymbol{\alpha}$ so that traffic splitting reduces $\tilde{a}_i^k(\boldsymbol{\alpha}, \mathbf{x})$ by a factor proportional to the splitting – recall that $\alpha_{ij} \leq 1$. Thus, traffic splitting tends to increase the slack with which (5.6) is satisfied because it keeps $\bar{a}_i^k(\boldsymbol{\alpha}, \mathbf{x})$ more or less constant but reduces $\tilde{a}_i^k(\boldsymbol{\alpha}, \mathbf{x})$ significantly.

For given positions $\mathbf{x}$, we want to find routing variables $\boldsymbol{\alpha}$ that satisfy (5.6) or (5.5). In either case there is some indeterminacy because there is a non-unique set of variables $\boldsymbol{\alpha}$ that satisfy the corresponding inequality. This indeterminacy provides a degree of freedom that can be used to increase the reliability beyond the required level. For doing so we introduce a slack variable a_Δ and write the following optimization problem which seeks to maximize the minimum $a_{i,\min}^k$ threshold that can be maintained with probability ϵ.

$$\begin{aligned} \boldsymbol{\alpha}(\mathbf{x}) = \underset{\boldsymbol{\alpha},\, a_\Delta}{\operatorname{argmax}} \quad & a_\Delta \\ \text{subject to} \quad & \frac{\bar{a}_i^k(\boldsymbol{\alpha}, \mathbf{x}) - (a_{i,\min}^k + a_\Delta)}{\sqrt{\tilde{a}_i^k(\boldsymbol{\alpha}, \mathbf{x})}} \geq \Phi^{-1}(\epsilon), \end{aligned} \tag{5.7}$$

where $\bar{a}_i^k(\boldsymbol{\alpha}, \mathbf{x})$ and $\tilde{a}_i^k(\boldsymbol{\alpha}, \mathbf{x})$ are given as in (5.3) and (5.4), the routing variables $\boldsymbol{\alpha}$ are further constrained to $0 \leq \alpha_{ij}^k \leq 1$ and $\sum_{j,k} \alpha_{ij}^k \leq 1$, and the probability constraints are required for all i and k.

The original goal as stated in (5.6) is to find routing variables that make $a_i^k(\boldsymbol{\alpha}, \mathbf{x}) \geq a_{i,\min}^k$ with probability ϵ. The formulation in (5.7) seeks the best margin a_Δ for which we can have $a_i^k(\boldsymbol{\alpha}, \mathbf{x}) \geq a_{i,\min}^k + a_\Delta$ with probability ϵ. A large value of a_Δ implies that the constraints in (5.6) are satisfied with significant slack and that there is significant liberty to change the physical configuration without violating communication constraints. The goal of maximizing a_Δ is to find routing variables that maximize the flexibility to change the configuration for which $\boldsymbol{\alpha}$ was chosen. This flexibility is instrumental for the mobility control algorithms.

The optimization problem in (5.7) can be formulated as a second order cone program (SOCP) [63] as long as $\epsilon > 0.5$ so that $\Phi^{-1}(\epsilon) > 0$ which maintains (5.8) as a convex second–order cone constraint. Start by noticing that the route allocation constraints $0 \leq \alpha_{ij}^k \leq 1$ and $\sum_{j,k} \alpha_{ij}^k \leq 1$ are linear in $\boldsymbol{\alpha}$. The probability constraints can be rewritten as

$$\sqrt{\tilde{a}_i^k(\boldsymbol{\alpha}, \mathbf{x})} \leq \frac{\bar{a}_i^k(\boldsymbol{\alpha}, \mathbf{x})}{\Phi^{-1}(\epsilon)} - \frac{a_\Delta}{\Phi^{-1}(\epsilon)} - \frac{a_{i,\min}^k}{\Phi^{-1}(\epsilon)}. \tag{5.8}$$

We define the $N \times N$ matrix $A_k = (\alpha_{ij}^k)$ and $\boldsymbol{\alpha} = (\mathrm{vec}(A_1), \ldots, \mathrm{vec}(A_K))$. Likewise, define the rate matrix $R = (R_{ij})$ which yields vectors $\bar{\mathbf{r}} = \mathrm{vec}(\bar{R})$ and $\tilde{\mathbf{r}} = \mathrm{vec}(\tilde{R})$ that represent the aggregate rate means and variances respectively. Then, define the matrix $B = \mathrm{diag}\Big((\sqrt{\tilde{\mathbf{r}}}, \ldots, \sqrt{\tilde{\mathbf{r}}}, 0)\Big)$ with K instances of $\sqrt{\tilde{\mathbf{r}}}$ and the variable $\mathbf{y} = (\boldsymbol{\alpha}, a_\Delta)$. For each node, define the vector $c_i = (\mathrm{vec}(\bar{R} \cdot S), \ldots, \mathrm{vec}(\bar{R} \cdot S), -1)/\Phi^{-1}(\epsilon)$ with K instances of $\mathrm{vec}(\bar{R} \cdot S)$ where $\bar{R} \cdot S$ is the component wise multiplication of $\bar{R}$ with a sign matrix such that $\bar{a}_i^k(\boldsymbol{\alpha}, \mathbf{x}) = \Phi^{-1}(\epsilon) c_i^{\mathrm{T}} \mathbf{y}$. Finally, define the constant $d_i = -a_{i,min}/\Phi^{-1}(\epsilon)$ and rewrite (5.8) as

$$\|A\mathbf{y}\| \leq c_i^T \mathbf{y} + d_i \tag{5.9}$$

The constraint in (5.9) defines a second order cone because it constrains the norm of a vector with a linear function. A problem with conic and linear constraints and a linear objective is, by definition, an SOCP. SOCPs are a particular case of convex optimization problem that can be efficiently solved with primal-dual potential reduction or interior point methods [63].

The computational complexity of solving (5.7) is represented as a polynomial function of the dimension of the SOCP problem M, $\mathrm{O}\big(M^3\big)$ [63]. The dimension of the SOCP in (5.7) is based on the number of agents, N, and communication flows, K, where $M = K \cdot N^2$. However, in practical implementations, the N^2 term can be reduced by eliminating links where $\bar{R}_{ij}$ is below a certain threshold. The actual computation time for an SOCP, using a primal-dual potential reduction method on a 2.5 GHz processor, is depicted in Fig. 5.1 for

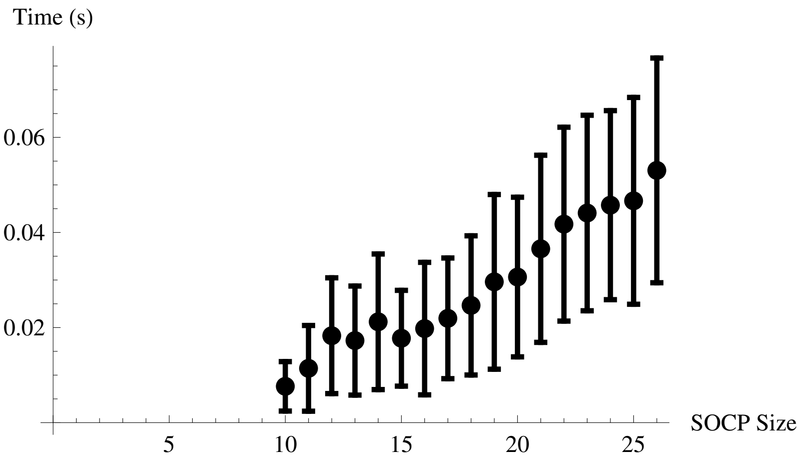


Figure 5.1: Running time statistics for the solution to the SOCP (5.7) on a 2.5 Ghz processor. The SOCP size refers to the dimension of $\mathbf{y}$ in (5.9). Error bars indicate one standard deviation.

a number of different problem sizes where the problem size is the dimension of $\mathbf{y}$ in (5.9).

5.2 Coordinated Control of Robot Mobility

As per the robust concurrent routing and mobility control problem (5.2), the objective of mobility control is to find robot trajectories that decrease $\Psi(\mathbf{x})$ while satisfying the probability constraints in (5.6) which are equivalent to $\mathrm{P}\left[a_i^k(\boldsymbol{\alpha},\mathbf{x}) \geq a_{i,\min}^k\right] \geq \epsilon$. To check for the feasibility of a joint mobility and network configuration $(\boldsymbol{\alpha},\mathbf{x})$, we define the probability margin as the minimum slack in probability constraints across all flows and sources,

$$\nu(\boldsymbol{\alpha},\mathbf{x}) := \min_{i,k}\left[\frac{\bar{a}_i^k(\boldsymbol{\alpha},\mathbf{x}) - a_{i,\min}^k}{\sqrt{\tilde{a}_i^k(\boldsymbol{\alpha},\mathbf{x}')}} - \Phi^{-1}(\epsilon)\right] \tag{5.10}$$

Notice that a necessary and sufficient condition for feasibility of the *physical* component of the configuration $\mathbf{x}$ is to have $\nu(\boldsymbol{\alpha}(\mathbf{x}),\mathbf{x}) \geq 0$, with routing variables $\boldsymbol{\alpha}(\mathbf{x})$ given by the unique solution of the convex optimization problem in (5.7). A sufficient condition for the feasibility of physical configuration $\mathbf{x}'$ is the existence of a network configuration $\boldsymbol{\alpha}$

for which $\nu(\boldsymbol{\alpha}, \mathbf{x}') \geq 0$. In particular, for $\mathbf{x}'$ close to $\mathbf{x}$ we expect to have $\nu(\boldsymbol{\alpha}(\mathbf{x}), \mathbf{x}') \geq 0$ since the channel statistics at $\mathbf{x}$ and $\mathbf{x}'$ are close.

5.2.1 Gradient Control Law

Since joint optimization over $\boldsymbol{\alpha}$ and $\mathbf{x}$ is problematic, e.g. there is a non-convex relationship between positions $\mathbf{x}$ and end-to-end rates $a_i^k(\boldsymbol{\alpha}, \mathbf{x})$ via the rate function $R(\mathbf{x})$, we pursue a controller that uses gradient descent to find locally optimal solutions to (5.2). That is, we choose robot velocities $\dot{\mathbf{x}}(t)$ proportional to the negative gradient of $-\nabla\Psi(\mathbf{x}(t))$. However, since the problem in (5.2) is subject to communication constraints, a local controller should be based on gradients of the potential function $\Psi(\mathbf{x}(t)$ projected onto the feasible set $\nu(\boldsymbol{\alpha}, \mathbf{x}) \geq 0$. However, the complex description of the feasible set precludes computation of projected gradients. Instead, we consider the probability margin $\nu(\boldsymbol{\alpha}(\mathbf{x}), \mathbf{x})$ and modify the potential function $\Psi(\mathbf{x})$ by incorporating the probability margin constraint into the objective through a barrier function with scaling parameter μ,

$$\Omega(\mathbf{x}) := \Psi(\mathbf{x}) - \mu \log\big(\nu(\boldsymbol{\alpha}(\mathbf{x}), \mathbf{x})\big). \tag{5.11}$$

Since non-negativity is necessary and sufficient for feasibility of the physical configuration $\mathbf{x}$, the potential function $\Omega(\mathbf{x})$ in (5.11) is defined if and only if physical configuration $\mathbf{x}$ is feasible.

The control law for the team of robots is defined to implement gradient descent on the modified potential $\Omega(\mathbf{x})$ introduced in (5.11), i.e.

$$\mathbf{u}(t) = -\nabla\Psi(\mathbf{x}(t)) + \mu \frac{\nabla_{\mathbf{x}}\nu\big[\boldsymbol{\alpha}(\mathbf{x}(t)), \mathbf{x}(t)\big]}{\nu\big[\boldsymbol{\alpha}(\mathbf{x}(t)), \mathbf{x}(t)\big]}. \tag{5.12}$$

The term $\nabla\Psi(\mathbf{x}(t))$ in (5.12) drives the system to satisfy the mobility task. The term $\mu\nabla_{\mathbf{x}}\nu\big[\boldsymbol{\alpha}(\mathbf{x}(t)), \mathbf{x}(t)\big]/\nu\big[\boldsymbol{\alpha}(\mathbf{x}(t)), \mathbf{x}(t)\big]$ serves as a barrier that drives robots away from configurations for which there is a low probability of exceeding the desired reliability in end-to-end rates. The scaling parameter μ is initialized to $\mu_0 = 1$ and robots are driven so that $\dot{\mathbf{x}}(t) = \mathbf{u}(t)$ from (5.12) until we reach a stationary point for which $\mathbf{u}(x(t)) = \mathbf{0}$. We define the stationary point for a given μ to be $\mathbf{x}_\infty(\mu) = \lim_{t\to\infty} x(t)$. For a task that cannot be satisfied without violating the communication constraints, consecutive reduction of μ leads to configurations that further minimize $\Psi(\mathbf{x}_\infty(\mu))$.

Proposition 5.2.1. *Solving* (5.2) *by application of* (5.12) *and* (5.7) *represents a family of algorithms that are parameterized by* μ. *Each instance will converge to a stationary point* $\mathbf{x}_\infty(\mu) = \lim_{t\to\infty} \mathbf{x}(t)$ *such that one of the following is true*

1. $\Psi(\mathbf{x}_\infty(\mu)) = \Psi_{min}$ *(i.e. the task is accomplished),*
2. $\Psi(\mathbf{x}_\infty(\mu)) \leq \Psi(\mathbf{x})\ \forall\ \left\{\mathbf{x} \in \mathcal{B}_{\mathbf{x}_\infty(\mu)} : \nu(\boldsymbol{\alpha}, \mathbf{x}) \geq \mu \frac{\nabla_{\mathbf{x}}\nu\left(\boldsymbol{\alpha}(\mathbf{x}),\mathbf{x}\right)}{\nabla_x \Psi(\mathbf{x})}\right\}$.

Proof. Both of these cases represent a stationary point of the controller where $\mathbf{u}(\mathbf{x}_\infty(\mu)) = 0$,

$$\nabla_x \Psi(\mathbf{x}_\infty(\mu)) = \mu \frac{\nabla_{\mathbf{x}}\nu\big(\boldsymbol{\alpha}(\mathbf{x}_\infty(\mu)), \mathbf{x}_\infty(\mu)\big)}{\nu\big(\boldsymbol{\alpha}(\mathbf{x}_\infty(\mu)), \mathbf{x}_\infty(\mu)\big)} \tag{5.13}$$

for all components of $\mathbf{x}$. Case (1) occurs when the system globally minimizes the task potential function so that $\nabla_x \Psi(\mathbf{x}) = 0$, i.e., $\Psi(\mathbf{x}_\infty(\mu)) = \Psi_{min}$, and $\nabla_x \nu(\boldsymbol{\alpha}(\mathbf{x}), \mathbf{x}) = 0$ which corresponds to a local maxima of the team's probability margin. Since $\nabla_x \Psi(\mathbf{x}) = 0$ only at the global minima of the task potential function, the condition of local optimality in case (2) occurs when (5.13) is true and $\nabla_x \Psi(\mathbf{x}_\infty(\mu)) > 0$. We can rearrange (5.13) to be

$$\nu\big(\boldsymbol{\alpha}(\mathbf{x}_\infty(\mu)), \mathbf{x}_\infty(\mu)\big) = \mu \frac{\nabla_{\mathbf{x}}\nu\big(\boldsymbol{\alpha}(\mathbf{x}_\infty(\mu)), \mathbf{x}_\infty(\mu)\big)}{\nabla_x \Psi(\mathbf{x}_\infty(\mu))}$$

for all components of $\mathbf{x}$. Given the definition of $\boldsymbol{\alpha}(\mathbf{x})$ as the unique argmax of an SOCP (5.7), it is true that

$$\nu\big(\boldsymbol{\alpha}(\mathbf{x}_\infty(\mu)), \mathbf{x}_\infty(\mu)\big) = \mu \frac{\nabla_{\mathbf{x}}\nu\big(\boldsymbol{\alpha}(\mathbf{x}_\infty(\mu)), \mathbf{x}_\infty(\mu)\big)}{\nabla_x \Psi(\mathbf{x}_\infty(\mu))} \geq \nu\big(\boldsymbol{\alpha}, \mathbf{x}_\infty(\mu)\big)$$

for all $\boldsymbol{\alpha}$ and all components of $\mathbf{x}$. This is a necessary condition for $\mathbf{x}_\infty(\mu)$ to be a stationary point. Therefore a sufficient condition for $\mathbf{x}$ to be a non-stationary point is that

$$\nu\big(\boldsymbol{\alpha},\mathbf{x}\big) \geq \mu\frac{\nabla_{\mathbf{x}}\nu\big(\boldsymbol{\alpha}(\mathbf{x}),\mathbf{x}\big)}{\nabla_x\Psi(\mathbf{x})}.$$

Thus, gradient-based optimization with (5.12) will converge to a point where

$$\Psi(\mathbf{x}_\infty(\mu)) \leq \Psi(\mathbf{x}) \;\forall\; \left\{\mathbf{x} \in \mathcal{B}_{\mathbf{x}_\infty(\mu)} : \nu(\boldsymbol{\alpha},\mathbf{x}) \geq \mu\frac{\nabla_{\mathbf{x}}\nu\big(\boldsymbol{\alpha}(\mathbf{x}),\mathbf{x}\big)}{\nabla_x\Psi(\mathbf{x})}\right\} \tag{5.14}$$

□

Note that case (2) in Proposition 5.2.1 refers to a condition of local optimality for (5.11). As $\mu \to 0$, the condition (5.14) represents a network configuration that cannot move closer to the goal configuration without violating the networking constraint $\nu(\boldsymbol{\alpha},\mathbf{x}) \geq 0$.

5.2.2 Algorithm for Coordinated Control

By assumption, computing $\nabla\Psi(\mathbf{x}(t))$ is straight forward. However, computing $\nabla_{\mathbf{x}}\nu(\boldsymbol{\alpha}(\mathbf{x}),\mathbf{x})$ is difficult because: (i) Values of $\nu(\boldsymbol{\alpha}(\mathbf{x}),\mathbf{x})$ depend on reliabilities $R_{ij}(\mathbf{x})$ that are not known in analytic form but queried from a channel estimation module; see Fig. 3.7 and Chapter 4. (ii) The margin $\nu(\boldsymbol{\alpha}(\mathbf{x}),\mathbf{x})$ depends on $\boldsymbol{\alpha}(\mathbf{x})$ that is computed as the solution of the SOCP in (5.7). To approximate $\nabla_{\mathbf{x}}\nu(\boldsymbol{\alpha}(\mathbf{x}),\mathbf{x})$, we consider a finite set of perturbations $\mathbf{X}'$ of the position $\mathbf{x}$ and define $\widehat{\nabla}_{\mathbf{x}}\nu(\boldsymbol{\alpha}(\mathbf{x}),\mathbf{x})$ as the average of the finite difference ratios,

$$\widehat{\nabla}_{\mathbf{x}}\nu(\boldsymbol{\alpha}(\mathbf{x}),\mathbf{x}) = \sum_{\mathbf{x}'\in\mathbf{X}'}\frac{\nu(\boldsymbol{\alpha}(\mathbf{x}),\mathbf{x}) - \nu(\boldsymbol{\alpha}(\mathbf{x}'),\mathbf{x}')}{\|\mathbf{x}-\mathbf{x}'\|}(\mathbf{x}-\mathbf{x}'). \tag{5.15}$$

To compute $\widehat{\nabla}_{\mathbf{x}}\nu(\boldsymbol{\alpha}(\mathbf{x}),\mathbf{x})$ in (5.15) it is necessary to solve the SOCP in (5.7) for all positions $\mathbf{x}' \in \mathbf{X}'$. To reduce computational cost, we define a simplified gradient estimate $\widehat{\nabla}^{(s)}_{\mathbf{x}}\nu(\boldsymbol{\alpha}(\mathbf{x}),\mathbf{x})$ as

$$\widehat{\nabla}^{(s)}_{\mathbf{x}}\nu(\boldsymbol{\alpha}(\mathbf{x}),\mathbf{x}) = \sum_{\mathbf{x}'\in\mathbf{X}'}\frac{\nu(\boldsymbol{\alpha}(\mathbf{x}),\mathbf{x}) - \nu(\boldsymbol{\alpha}(\mathbf{x}),\mathbf{x}')}{\|\mathbf{x}-\mathbf{x}'\|}(\mathbf{x}-\mathbf{x}'), \tag{5.16}$$

Algorithm 1 Search for feasible node velocities

```
Require: Routing solution at $\mathbf{x}(t)$, $\boldsymbol{\alpha}(\mathbf{x})$. Prediction function for link rates, $R(x_i, x_j)$. Desired velocity for each node $\dot{x}_i^{des}$. Initial position of nodes at time $t$, $\mathbf{x}(t)$. Increment to use when scaling velocities, $\Delta v$
$v_j = 0$ for all $j \in J$ {Initialize all node velocities to zero}
for $k = 0, \dots, K$ do {each flow $k$}
    $Q$ = Empty queue
    $M$ = Empty set {To store scaled nodes}
    Enqueue($Q, k$)
    while $Q$ not empty do
        $j$ = Dequeue($Q$)
        $u_j = \dot{x}_j^{des} / \|\dot{x}_j^{des}\|$
        $v_j = \|\dot{x}_j^{des}\|$
        while $v_j \geq 0$ and $\nu(\boldsymbol{\alpha}(\mathbf{x}(t)), \mathbf{x}(t) + \mathbf{v}(t)T) < 0$ do
            $v_j = v_j - \Delta v$ {Scale velocity of node $j$}
            $Add(M, j)$ {Mark node $j$ as done}
        end while
        for $i \in \{J : \alpha_{ij}^k > 0\}$ do {each child of $j$}
            if $i \notin M$ then
                Enqueue($Q, i$)
            end if
        end for
    end while
end for
```

where the routing solution $\boldsymbol{\alpha}(\mathbf{x})$ is used as an approximation to $\boldsymbol{\alpha}(\mathbf{x}')$. Though the lower compuatational cost of $\widehat{\nabla}_{\mathbf{x}}^{(s)} \nu(\boldsymbol{\alpha}(\mathbf{x}), \mathbf{x})$ facilitates real-time implementation, use of the gradient approximation $\widehat{\nabla}_{\mathbf{x}} \nu(\boldsymbol{\alpha}(\mathbf{x}), \mathbf{x})$ will yield final configurations with lower task potential.

On a real system, the proposed local controller operates in discrete time intervals of duration T. At times $t = kT$ for $k = 1, 2, \dots$ optimal routes $\boldsymbol{\alpha}(\mathbf{x}(kT))$ are computed as per (5.7) and control inputs $\mathbf{u}(kT)$ are computed as per (5.12). Network variables $\boldsymbol{\alpha}(\mathbf{x}(kT))$ are used to route packets through the network of robots and the physical positions of the robots are updated as

$$x_i((k+1)T) = x_i(kT) + \beta_i T u_i(kT) \tag{5.17}$$

where β_i scale control inputs $u_i(kT)$ to ensure $\nu(\boldsymbol{\alpha}(\mathbf{x}(kT)), \mathbf{x}((k+1)T)) \geq 0$. That is, constants β_i guarantee that routing variables $\boldsymbol{\alpha}(\mathbf{x}(kT))$, optimized for configuration $\mathbf{x}(kT)$, are a feasible operating point for configuration $\mathbf{x}((k+1)T)$.

Scale factors β_i in (5.17) are chosen so that the predicted probability margin $\nu(\boldsymbol{\alpha}(\mathbf{x}(kT)), \mathbf{x}((k+1)T))$ after time horizon T remains positive. A simple solution is to uniformly chose all $\beta_i = \beta$ such that

$$\nu(\boldsymbol{\alpha}(\mathbf{x}(kT)), \mathbf{x}(kT) + T(\beta\dot{\mathbf{u}}(t))) \geq 0. \tag{5.18}$$

A suitable value for β can then be found through line search algorithms. However, values of β tend to be prohibitively conservative. To see this, consider the case where one link in the communication network is utilized such that any decrease in its reliability will invalidate $\nu(\boldsymbol{\alpha}(\mathbf{x}(kT)), \mathbf{x}(kT) + \delta) \geq 0$. Uniform scaling would force $\beta = 0$ and stop the motion of all nodes.

A method to select β_i separately is shown in Algorithm 1. We consider the nodes participating in each flow k according to the tree induced by the routing solution $\boldsymbol{\alpha}(\mathbf{x}(t))$ in the following breadth-first fashion. Starting at dest(k) for the first flow, we set $\beta_{\text{dest}(k)} = 1$ and then scale β_i for all nodes $i \in \{j : \alpha^k_{j,\text{dest}(k)} > 0\}$, i.e those sending data directly to dest(k), so that

$$\nu(\boldsymbol{\alpha}(\mathbf{x}(kT)), x_i(kT) + \beta_i u_i(kT)) \geq 0 \tag{5.19}$$

As above, the search for each β_i is a line search. However, after scaling β_i for immediate neighbors of dest(k), the algorithm procedes with the 2-hop neighbors and so forth in a breadth-first search facilitated with a first in, first out queue.

5.2.3 Simulation Results

We implement in simulation the local controller with position updates as in (5.17), mobility control inputs given by (5.12), communication variables obtained from (5.7), and using the velocity search in Algorithm 1. Allowable communication rates are chosen according to (M1) from Chapter 4. Computing controls based on local optimization of the network-level end-to-end rates allows for a method of realizing team deployment while maintaining the necessary level of network connectivity. Figure 5.2 depicts an example deployment trajectory with three robots for a time-varying task potential,

$$\Psi(\mathbf{x}(t)) = \begin{cases} x_{2,goal} = (4,0) & t < 40 \\ x_{2,goal} = (8,0) & t < 60 \\ x_{2,goal} = (6,5) & t \geq 60. \end{cases} \tag{5.20}$$

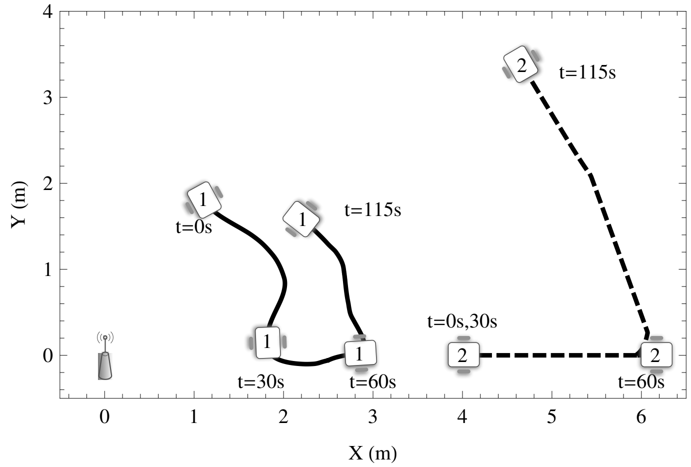
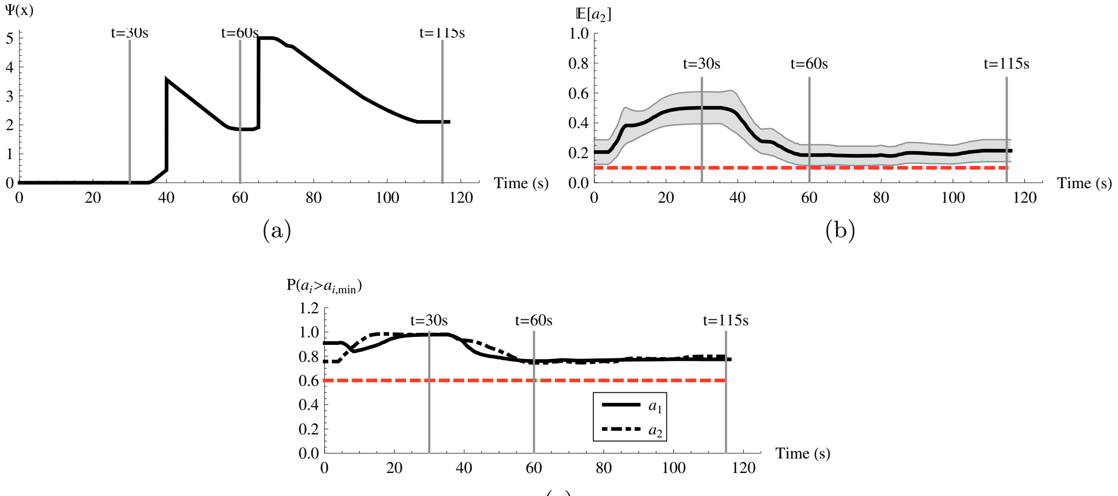


Figure 5.2: Deployment via local control law (5.12) for a system with a fixed access point, relay node x_1 and lead node x_2 which is controlled by a time-varying task potential $\Psi(\mathbf{x}(t))$.

Most importantly, this example demonstrates convergence of the task potential $\Psi(\mathbf{x})$ while maintaining $\mathrm{P}\left[a_i \geq a_{i,min}\right] > \epsilon$ as depicted in Figs. 5.3a and 5.3c. However, it is also interesting to observe that when task potential is minimized, e.g. $t \leq 30$ s, the local control law (5.12) maximizes the probability of each end-to-end rate exceeding its minium threshold. When the task potential switches so that $\Psi(\mathbf{x})$ is no longer minimized, the probability margin is reduced so that the primary objective, minimization of $\Psi(\mathbf{x})$, is prioritized. Finally, Fig. 5.3b depicts the end-to-end rate of the node x_2 that must remain above $a_{2,min} = 0.1$. Recall that $R_{ij}(\mathbf{x})$ is a random variable that affects the end-to-end rate. The envelope around $\bar{a}_2$ in Fig. 5.3b depict the effect that different realizations of communication channels $R_{ij}(\mathbf{x})$ will have on the end-to-end rate. Since the pursuit of minimization on $\Psi(\mathbf{x})$ is constrained to have a probability margin $\nu(\boldsymbol{\alpha}(\mathbf{x}), \mathbf{x}) \geq 0$, the end-to-end rate exceeds its threshold in the presence of deviations to $R_{ij}(\mathbf{x})$.



(a)

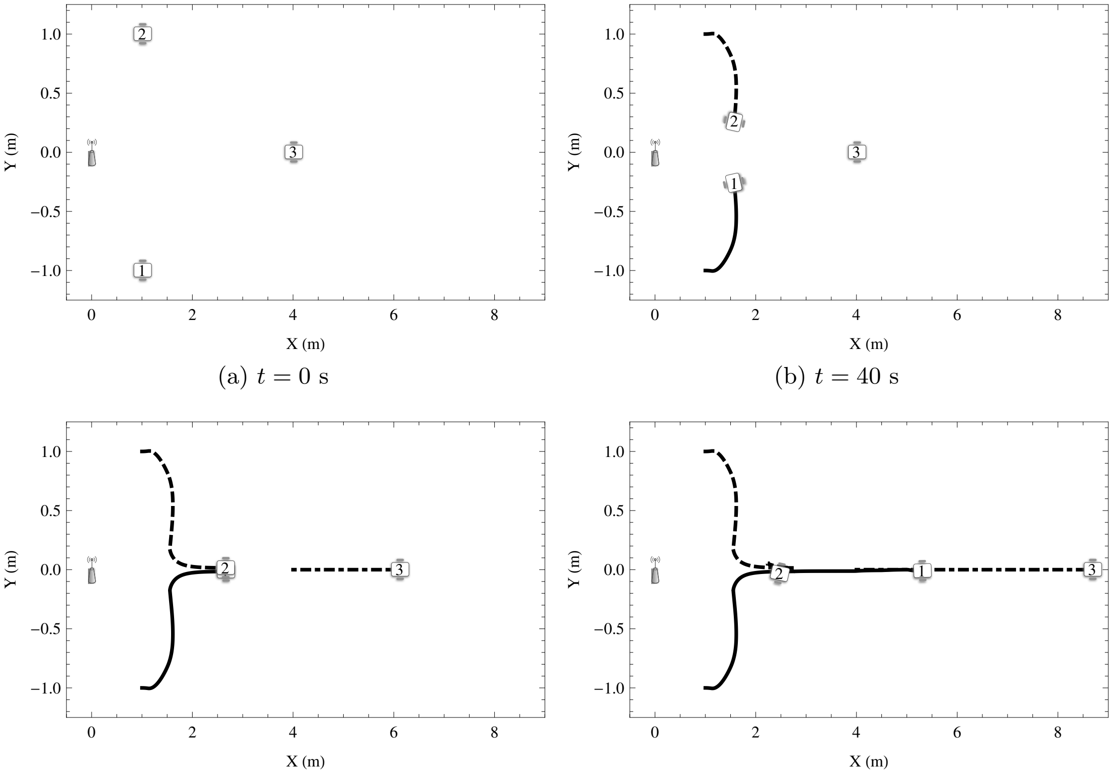
(b)

(c)

Figure 5.3: Performance of the local control law (5.12) demonstrating convergence of $\Psi(\mathbf{x}(t))$ in (a), the maintenance of expected end-to-end rate greater than the threshold of $a_{2,min} = 0.1$ in (b), and the $\mathrm{P}\left[a_i \geq a_{i,min}\right] > 0.6$. The envelope surrounding $\mathbb{E}\left[a_2\right]$ in (b) depicts the 60% confidence interval for realizations of the end-to-end rate with stochastic $R_{ij}(\mathbf{x})$.

We perform a four robot simulation, depicted in Fig. 5.4, to demonstrate how the complexity of the objective function (5.11) increases with more nodes. Similiar to the three robot simulation above, the task potential function is time-varying so that $\Psi(\mathbf{x}) = 0$ for $t < 50$ s and $\Psi(\mathbf{x}) = \|x_3 - (9,0)\|$ for $t \geq 50$ s. When $t < 40$ and the task potential is already minimized, control of x_1 and x_2 is based soley on the maximization of $\log \nu(\boldsymbol{\alpha}(\mathbf{x}), \mathbf{x})$. From the symmetric initial configuration, the maximization drives x_1 and x_2 towards a local maxima where they would be positioned at the same point.

To further illustrate the "terrain" of the objective function (5.11) that drives local control we observe the convergence of $\Psi(\mathbf{x}(t))$ as depited in Fig. 5.5a. For times 70 s $<$ $t <$ 90 s, convergence of $\Psi(\mathbf{x}(t))$ slows, $\nu(\boldsymbol{\alpha}(\mathbf{x}), \mathbf{x})$ as depicted in Fig. 5.5c shows little change, and x_1 and x_2 cease progress. Fortunately, the gradient of $\Psi(\mathbf{x})$, though small, is enough to pull the system out of the local minima so that it can achieve a global minimum at $t = 140$ s. As the number of agents increases, the frequency of local minima in $-\nu(\boldsymbol{\alpha}(\mathbf{x}), \mathbf{x})$ becomes more and more of an issue for local control. The addition of obstacles into the environment adds further difficulties as it not only affects feasible $\mathbf{x}$



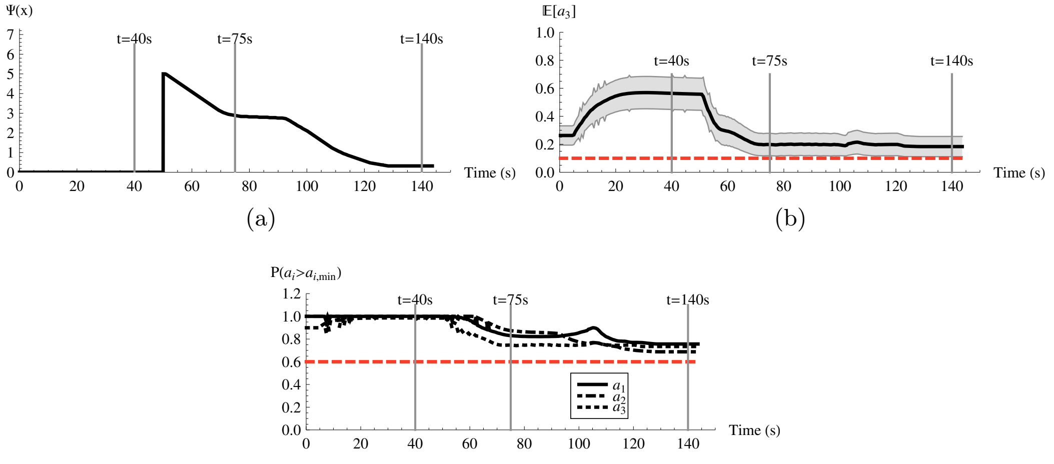
(a) $t = 0$ s

(b) $t = 40$ s

(c) $t = 75$ s

(d) $t = 140$ s

Figure 5.4: Snapshots from a four robot trial. The end-to-end rates and probability of meeting the problem specifications are depicted in Fig. 5.5. $\Psi(\mathbf{x}(t)) = \|x_3 - (9,0)\|$ for $t > 50$ s and $\Psi(\mathbf{x}(t)) = 0$ for $t \leq 50$ s.



(a)

(b)

(c)

Figure 5.5: The (a) convergence of $\Psi(\mathbf{x})$, (b) end-to-end rates, and (c) probability of success for the four robot trial depicted in Fig. 5.4.

due to collision constraints, but also introduces non-smooth components in the underlying point-to-point communication links $R_{ij}(\mathbf{x})$.

5.3 Optimal Configurations

As demonstrated in Section 5.2, solutions to (5.2) that rely on local gradient–based control will converge to stationary points of the modified objective function $\Omega(\mathbf{x})$ given in (5.11). Even when the task potential is globally minimized, i.e. $\Psi(\mathbf{x}) = \Psi_{min}$, there are often multiple configurations $\mathbf{x}$ that achieve local maxima of the "barrier" term in the modified potential function $\Omega(\mathbf{x})$. We seek to characterize these so–called optimal configurations in order to understand the local minima that will disrupt gradient–based controllers and to inform our configuration search as we move towards global planning methods.

Rather than consider all possible task potential functions $\Psi(\mathbf{x})$, we will restrict our attention to the joint optimization of $\nu(\boldsymbol{\alpha}, \mathbf{x})$ over all routes $\boldsymbol{\alpha}$ and robot configurations $\mathbf{x}$ for the simplest task potential function. We will begin by examining the problem set of three nodes restricted to operate along a straight line. There is one fixed operating center at $x_0 = 0$, one mobile node x_1 acting as a relay, and one fixed node x_2, e.g. x_2 is placed such that $\Psi(\mathbf{x}) = \Psi_{min}$. Thus, the joint space of physical and network configurations are described by $(x_1, \alpha_{1,0}, \alpha_{2,0}, \alpha_{2,1})$ and the problem is parameterized by $x_2, a_{1,min}, a_{2,min}$ as depicted in Fig. 5.6. We will continue to assume the point-to-point communication capability is given by (M1) where, for notational convenience, we will use $\bar{R}(d), \tilde{R}(d)$ to denote the mean and variance respectively of the supported rate over a

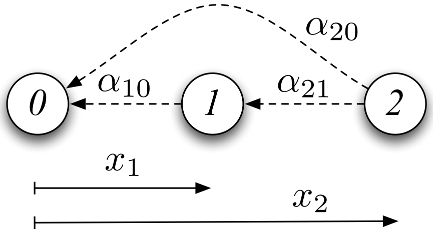


Figure 5.6: Configuration of three nodes restricted to operate along a straight line. x_0 and x_2 are fixed, leaving x_1 to move in order to optimize the probability of achieving the desired rates.

distance d. Remembering the SOCP formulation in (5.7), we seek to maximize $\nu(\boldsymbol{\alpha}, \mathbf{x}) = \min \mathrm{P}\left[a_i(\boldsymbol{\alpha}, \mathbf{x}) \geq a_{i,min}\right]$ by maximizing a_Δ, the largest end-to-end rate margin that can be maintained with probability ϵ. We can write this same optimization, now over both routing variables $\mathbf{x}$ as well as node positions $\mathbf{x}$, as

$$\begin{array}{ll} \underset{x_1,\alpha_{1,0},\alpha_{2,0},\alpha_{2,1},a_\Delta}{\text{maximize}} & a_\Delta \\ \text{subject to} & \dfrac{\alpha_{2,0}\bar{R}(x_2) + \alpha_{2,1}\bar{R}(x_2 - x_1) - a_{2,min} - a_\Delta}{\sqrt{\alpha_{2,0}^2\tilde{R}(x_2) + \alpha_{2,1}^2\tilde{R}(x_2 - x_1)}} \geq \Phi^{-1}(\epsilon) \\ & \dfrac{\alpha_{1,0}\bar{R}(x_1) - \alpha_{2,1}\bar{R}(x_2 - x_1) - a_{1,min} - a_\Delta}{\sqrt{\alpha_{1,0}^2\tilde{R}(x_1) + \alpha_{2,1}^2\tilde{R}(x_2 - x_1)}} \geq \Phi^{-1}(\epsilon). \end{array} \tag{5.21}$$

We make the assumption that the fixed position of x_2 is such that it cannot maintain the desired rate of communications with the operating center with a direct link, i.e.

$$\frac{\bar{R}(x_2) - a_{2,min}}{\sqrt{\tilde{R}(x_2)}} < \Phi^{-1}(\epsilon) \tag{5.22}$$

Furthermore, based on (M1), we can assume that $\bar{R}(d)$ is monotonically decreasing and that $\tilde{R}(d)$ is monotonically increasing with distance d. This introduces a series of additional inequalities

$$\bar{R}(x_2) \leq \bar{R}(x_2 - x_1) \tag{5.23}$$

$$\bar{R}(x_2) \leq \bar{R}(x_1) \tag{5.24}$$

$$\tilde{R}(x_2) \geq \tilde{R}(x_2 - x_1) \tag{5.25}$$

$$\tilde{R}(x_2) \geq \tilde{R}(x_1) \tag{5.26}$$

The problem in (5.21) seeks to maximize the minimum end-to-end rate margin that can be maintained with probability ϵ. This means that its solution is a Pareto optimal point with the necessary condition that the two constraint slacks be equivalent, i.e.

$$\begin{aligned} &\alpha_{2,0}\bar{R}(x_2) + \alpha_{2,1}\bar{R}(x_2 - x_1) - a_{2,min} - \Phi^{-1}(\epsilon)\sqrt{\alpha_{2,0}^2\tilde{R}(x_2) + \alpha_{2,1}^2\tilde{R}(x_2 - x_1)} \\ &= \alpha_{1,0}\bar{R}(x_1) - \alpha_{2,1}\bar{R}(x_2 - x_1) - a_{1,min} - \Phi^{-1}(\epsilon)\sqrt{\alpha_{1,0}^2\tilde{R}(x_1) + \alpha_{2,1}^2\tilde{R}(x_2 - x_1)}. \end{aligned} \tag{5.27}$$

We can make some well-founded assumptions about the network routing solution at an optimal configuration. First, $\alpha_{1,0} = 1$ as this will only serve to maximize a_Δ with no ill effects. Second, $\alpha_{2,0} = 1 - \alpha_{2,1}$ since in an optimal configuration, node x_2 will maximize its outgoing channel. Then, by manipulating (5.27), we arrive at a fourth-order polynomial in $\alpha_{2,1}$. The roots of this polynomial yield four solutions for $\alpha_{2,1}$ as a function of $\bar{R}(x_2), \bar{R}(x_2 - x1), \tilde{R}(x_2), \tilde{R}(x_2 - x_1)$. We can numerically verify that one of these solutions matches the solution to the SOCP. While this approach yields an analytic solution to the optimal routing variables for a small problem, optimization of a_Δ is still a function of the mean and variance of the rate function, $\bar{R}_{ij}(\mathbf{x}), \tilde{R}_{ij}(\mathbf{x})$, which are non-convex.

However, for small systems like this, it is possible to exhaustively sample the space of node configurations $\mathbf{x}$ and use the SOCP (5.7) that results when positions $\mathbf{x}$ are fixed in (5.21) to compute $\boldsymbol{\alpha}(\mathbf{x})$. Choosing a parameterization of the three robot system described above, i.e. $L_0 = -53$ dBm, $n = 2.52$, $N_0 = -70$ dBm, $\sigma_{\mathcal{F}}^2 = 40$, $a_{1,min} = 0$, $a_{2,min} = 0.1$, $\epsilon = 0.8$, and $x_2 = 13$ m, we can sample $0 < x_1 < x_2 = 13$ m to find the point where a_Δ is maximized as depicted in Fig. 5.7a. it is tractable to do this over two dimensions as in Fig. 5.7b where we sample over all $0 \leq x_1 \leq x_2 \leq 17$ m. Figure 5.7 demonstrates that a_Δ for a one–relay system is well behaved as we vary $\mathbf{x}$ and has a unique maximum placing the relay near the midpoint of x_2 and the operating center at x_0. Thus, we can expect gradient–based optimization to converge to the optimal configuration.

We continue by examining the four robot case, i.e. one fixed operating center x_0, one fixed node x_3 with $a_{3,min} > 0$, and two mobile relay nodes x_1, x_2 as depicted in Fig.5.8. The increased complexity of this problem is immediately evident as the system of variables describing solutions is now $(x_1, x_2, \alpha_{10}, \alpha_{20}, \alpha_{21}, \alpha_{30}, \alpha_{31}, \alpha_{32})$. Furthermore, the additional relay node precipitates an additional constraint when writing the optimization

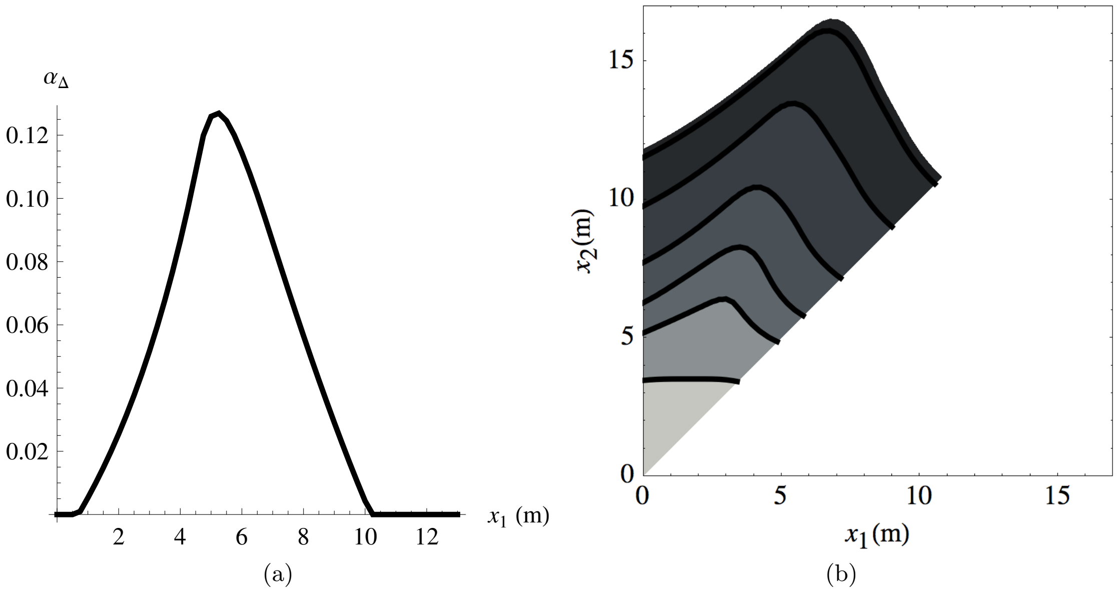


Figure 5.7: Exhaustive sampling of $\mathbf{x}$ to determine maximum of a_Δ for the one–relay case depicted in Fig. 5.6 where $x_2 = 13$ m. In (a), $x_2 = 13$ m while in (b), we sample across $x_1 \leq x_2 \leq 17$ m.

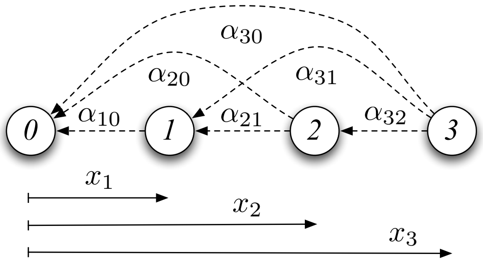


Figure 5.8: System of four robots

(5.21). Rather than attempt an analytic solution to this formulation, we procede with the exhaustive sampling approach whereby we fix values for x_1, x_2 and solve the resulting SOCP to determine $\boldsymbol{\alpha}(\mathbf{x})$ and the optimal a_Δ. Figure 5.9 depicts the optimal a_Δ for combinations of x_1, x_2 when the fourth node is fixed at $x_3 = 15$ m. Unlike the one relay case where there is a single $\mathbf{x}$ that locally maximizes a_Δ, the introduction of a second relay leads to multiple configurations that achieve local maxima. It is interesting to note that the two local maxima appearing in Fig. 5.9 have intuitive explanation. One maxima, at $x_1 = x_2 \approx 6.2$ m, corresponds to a "diversity" type of solution where traffic is split between the two relay nodes. The other, at $x_1 = 3.9$ m, $x_2 = 8$ m, corresponds to a "multihop"

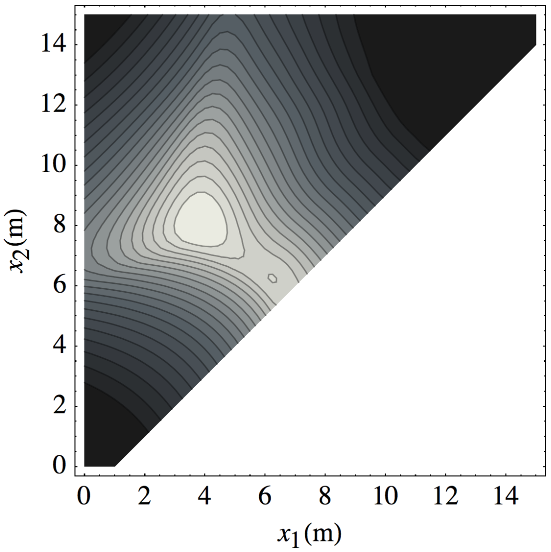
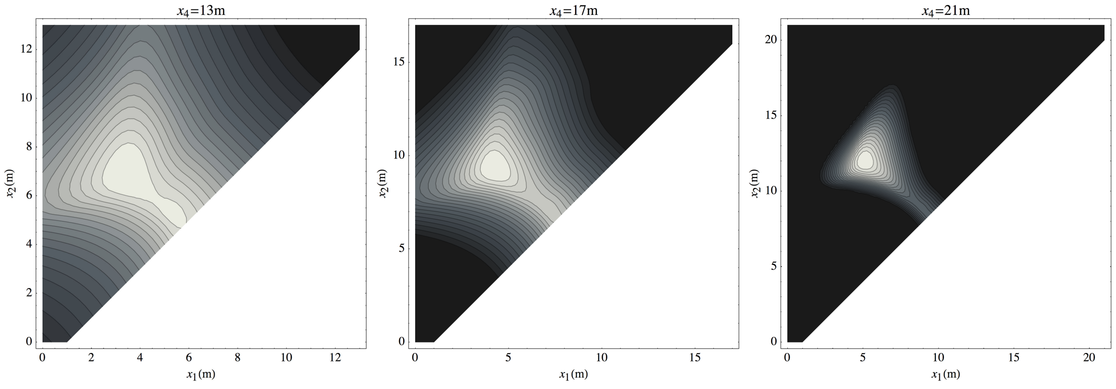


Figure 5.9: Exhaustive sampling of $\mathbf{x}$ to determine maximum of $\nu(\boldsymbol{\alpha}, \mathbf{x})$ for the two–relay case depicted in Fig. 5.8 where $x_3 = 15$ m.

type of solution where traffic is sent serially from x_3 to x_2 to x_1 to x_0. A gradient–based controller will converge to one of these two local maxima based on its initial condition.

As x_3 is increased, the "diversity" solution becomes infeasible and the system returns to a regime with a single optimal configuration as depicted in Fig.5.10. Thus, the complexity



(a) (b) (c)

Figure 5.10: Exhaustive sampling of $\mathbf{x}$ to determine maximum of a_Δ for the two–relay case while varying x_3 – (a) $x_3 = 13$ m, (b) $x_3 = 17$ m, and (c) $x_3 = 21$ m.

associated with local maxima of a_Δ is not only a function of the number of relays but also the parameterization of the problem, i.e. the position of the fixed node that is sending data back to the operating center. As we consider larger teams with even more complicated network topologies and complex environments with obstacles, local maxima of a_Δ become more of an issue.

5.4 Global Planning

Local control will drive the system towards local minima of (5.11). We propose that a global search of (5.2) is necessary in order to accomplish the high-level situational–awareness tasks we are interested in.

To consider global search of (5.2), we redefine the problem to be more amenable to motion planning approaches from the robotics literature. Let X be a bounded, connected open subset of $\mathbb{R}^{2N}$ that represents the full joint state space for the team of robots where $\mathbf{x}_{init}$ is the initial configuration of the team. In general, the goal region will be defined as $X_g = \{\mathbf{x} : \Psi(\mathbf{x}) < \Psi_{min} + \delta\}$. In an application where $\Psi(\mathbf{x}) = \|x_\ell - x_{\ell,g}\|^2$ where a lead agent must visit $x_{\ell,g}$, $X_g = \{\mathbf{x} : \|x_\ell - x_{\ell,g}\| < \delta\}$. The obstacle region X_{obs} contains any configuration that places an individual robot on a physical obstacle and the infeasible region represents configurations where it is infeasible to satisfy the network constraint (5.6),

$$\begin{aligned} X_{inf} = \Bigg\{ \mathbf{x} : & \frac{\bar{a}_i^k(\boldsymbol{\alpha}, \mathbf{x}) - a_{i,min}^k}{\sqrt{\tilde{a}_i^k(\boldsymbol{\alpha}, \mathbf{x})}} \geq \Phi^{-1}(\epsilon) \\ & \forall\ \boldsymbol{\alpha} \in \Big\{ \alpha_{ij}^k : 0 \leq \alpha_{ij}^k \leq 1, \sum_{j,k} \alpha_{ij}^k \leq 1 \Big\} \Bigg\}. \end{aligned} \tag{5.28}$$

The free space X_{free} is then $X \setminus (X_{obs} \cup X_{infeasible})$. Finally, a path in X is parameterized by a scalar $s \geq 0$ and given by $\boldsymbol{\sigma} : [0, s] \to X$. Concatenation of paths is defined by $\boldsymbol{\sigma} = \boldsymbol{\sigma}_1 | \boldsymbol{\sigma}_2$. A feasible path, and solution to our global-planning problem, is then $\boldsymbol{\sigma} : [0, s] \to X_{free}$ such that $\boldsymbol{\sigma}(0) = \mathbf{x}_{init}$ and $\boldsymbol{\sigma}(s) \in X_g$.

The dimensionality of our problem and the high computational cost of verifying a state is in X_{free} makes deterministic search algorithms impractical. Instead we turn to proba-

Algorithm 2 Structure of the rapidly exploring random tree algorithm

```
Require: Initial state $\mathbf{x}_0$, goal region $X_g$, representation of the bounded configuration space $X$.
$\mathcal{T}.init(\mathbf{x}_0)$
while $i < N$ do
    $\hat{\mathbf{x}} \leftarrow$ RandomState$(X, \mathcal{T})$
    $\mathbf{x}_{\min} \leftarrow$ Nearest$(\mathcal{T}, \hat{\mathbf{x}})$
    if $\mathbf{x}_{new} \leftarrow$ Extend$(\mathbf{x}_{\min}, \hat{\mathbf{x}})$ then
        $\mathcal{T}$.add_vertex$(\mathbf{x}_{new})$
        $\mathcal{T}$.add_edge$(\mathbf{x}_{\min}, \mathbf{x}_{new})$
        if $\mathbf{x}_{new} \in X_g$ then
            return $\mathcal{T}$
        end if
    end if
end while
return $\mathcal{T}$
```

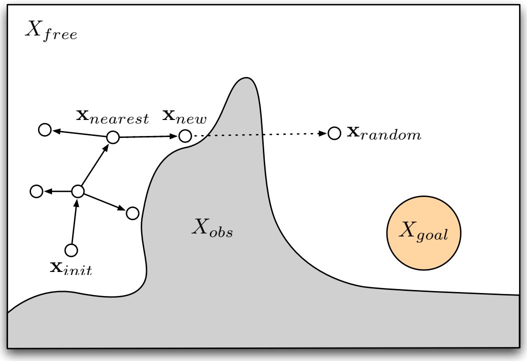


Figure 5.11: Graphical depiction of the RRT search process visualized in $\mathbb{R}^2$.

bilistic search methods than offer good space-filling properties and efficient exploration of an unknown space *e.g.* rapidly–exploring random tree (RRT) algorithms [52]. The basic structure of an RRT as detailed in Algorithm 2 and depicted in Fig. 5.11 is to start with an initial point $\mathbf{x}_0$ and expand to fully explore the workspace, adding states in a tree-like structure until a feasible point is added such that $\mathbf{x} \in X_g$. The tree is expanded by picking a random state $\hat{\mathbf{x}} = \textsc{RandomState}(X, \mathcal{T})$, finding the closest point on the existing tree $\mathbf{x}_{\min} = \textsc{Nearest}(\mathcal{T}, \hat{\mathbf{x}})$, and attempting to add a new point by extending from $\mathbf{x}_{\min}$, $\mathbf{x} = \textsc{Extend}(\mathbf{x}_{\min}, \hat{\mathbf{x}})$.

A common problem encountered when applying RRT algorithms to high-dimensional state spaces is that computation of Nearest is inefficient for increasing tree sizes. We adopt the strategy of storing the tree $\mathcal{T}$ in a KD-tree data structure which stores states in $\mathbb{R}^d$ by recursively subdividing based on alternating axis-aligned hyperplanes [5]. This enables approximate nearest neighbor calculations that maintain performance even as the dimension increases. However, there are two additional difficulties that arise when applying standard RRT algorithms to solve the specific high-dimensional network connectivity problem in (5.2): the verification of feasible states as Extend is used to expand the tree towards $\hat{\mathbf{x}}$ and the prohibitive cost of uniformly exploring X_{free} for our high-dimensional problem with slow-to-compute constraints.

5.4.1 Efficient verification of feasible states

The Extend$(\mathbf{x}_{from}, \mathbf{x}_{to})$ algorithm attempts to virtually drive the system from $\mathbf{x}_{from}$ towards $\mathbf{x}_{to}$ by successively verifying that points along the line connecting $\mathbf{x}_{from}$ and $\mathbf{x}_{to}$ are in X_{free}. It returns the state $\mathbf{x}_{new}$ as the closest state to $\mathbf{x}_{to}$ such that all states sampled with precision $\Delta\mathbf{x}$ between $\mathbf{x}_{from}$ and $\mathbf{x}_{new}$ are in X_{free}. In traditional motion planning applications, verification that $\mathbf{x} \in X_{free}$ is based on an algebraic constraint or collision query with a multitude of efficient methods for doing so [60, 72, 59]. While the necessary computation to determine $\mathbf{x} \notin X_{obs}$ is typically small, computation of $\mathbf{x} \notin X_{inf}$ requires a solution of the SOCP (5.7) and can be costly (see Fig. 5.1).

Consequently, we store $\boldsymbol{\alpha}(\mathbf{x})$ for every node in $\mathcal{T}$ and reuse methods from Section 5.2 to extend new states. By relying on the fact that an optimal robust routing solution $\boldsymbol{\alpha}(\mathbf{x})$ will be feasible for neighboring states, it is often possible to extend $\mathbf{x}$ towards $\hat{\mathbf{x}}$ without the costly overhead of numerical optimization as detailed in Algorithm 3.

5.4.2 Biased space sampling

Random states $\hat{\mathbf{x}}$ are chosen to sample the space $X \subset \mathbb{R}^{2N}$ according to a probability distribution $p_{\mathbf{x}}(\mathbf{x})$ representing the belief about configuration $\mathbf{x}$ being part of a feasible

Algorithm 3 EXTEND($\mathbf{x}_{from}, \mathbf{x}_{to}$)

```
Require: Initial state $\mathbf{x}_{from}$, desired final state $\mathbf{x}_{to}$, verify segment over $K$ steps
$\mathbf{x}_{new} \leftarrow \mathbf{x} \leftarrow \mathbf{x}_{from}$
$\boldsymbol{\alpha} \leftarrow$ argmax (5.7) for rates $R_{ij}$ in configuration $\mathbf{x}_{new}$.
$\Delta\mathbf{x} \leftarrow (x_{to} - x_{from})/K$
while $\mathbf{x}_{new} \neq \mathbf{x}_{to}$ and $\boldsymbol{\alpha} \neq \emptyset$ do
  $\mathbf{x}_{new} \leftarrow \mathbf{x}$
  $\mathbf{x} \leftarrow \mathbf{x}_{new} + \Delta\mathbf{x}$
  if $\nu(\boldsymbol{\alpha}, \mathbf{x}_{new}) \leq 0$ then
    {Recompute $\boldsymbol{\alpha}$ if the probability margin at $\mathbf{x}_{new}$ is negative}
    $\boldsymbol{\alpha} \leftarrow$ argmax (5.7) for rates $R_{ij}$ in configuration $\mathbf{x}_{new}$.
  end if
end while
if $\mathbf{x}_{new} = \mathbf{x}_{from}$ then
  return $\emptyset$
else
  return $\mathbf{x}_{new}$
end if
```

path $\boldsymbol{\sigma}(s)$. If nothing is known about $\boldsymbol{\sigma}(s)$, we choose $p_{\mathbf{x}}(\mathbf{x})$ uniform in the space X; i.e., we make $p_{\mathbf{x}}(\mathbf{x}) = 1/v(X)\mathbb{I}\{\mathbf{x} \in X\}$ where $\mathbb{I}\{\mathbf{x} \in X\}$ denotes the indicator function of the set X and $v(X)$ the volume of set X. In general, the final configuration is known in that $\boldsymbol{\sigma}(s) \in X_g$. We can then bias the distribution towards X_g by making

$$p_{\mathbf{x}}(\mathbf{x}) = \frac{p_g}{v(X_g)}\mathbb{I}\{\mathbf{x} \in X_g\} + \frac{1-p_g}{v(X \setminus X_g)}\mathbb{I}\{\mathbf{x} \notin X_g\}. \tag{5.29}$$

When $p_g = v(X_g)/v(X)$ the distribution in (5.29) corresponds to uniform sampling. Larger values of p_g make $\hat{\mathbf{x}}$ more likely to hit X_g than what corresponds to its volume $v(X_g)$. Goal biasing as in (5.29) improves efficiency of RRT algorithms by reducing the number of samples necessary to find a feasible path $\boldsymbol{\sigma}(s)$ in the high dimensional space $X \subset \mathbb{R}^{2N}$.

In some cases of interest, the volume of X_g is comparable to the volume of X. In these cases goal biasing offers little improvement over uniform sampling. In, e.g., the application where $\Psi(\mathbf{x}) = \|x_\ell - x_{\ell,g}\|^2$ the goal position of the leader $x_{\ell,g}$ is known, but the positions of the remaining robots are free. Thus, goal biasing would reduce the exploration cost along the components associated with x_ℓ but keep the cost of exploring the remaining

$2(N-1)$ dimensions fixed. To further reduce exploration cost in this case we construct a prediction $\tilde{X}_g \subset X_g$ of the final configuration and bias sampling towards this configuration prediction by making the sampling distribution

$$p_{\mathbf{x}}(\mathbf{x}) = \frac{p_g}{v(\tilde{X}_g)} \mathbb{I}\left\{\mathbf{x} \in \tilde{X}_g\right\} + \frac{1-p_g}{v(X \setminus \tilde{X}_g)} \mathbb{I}\left\{\mathbf{x} \notin \tilde{X}_g\right\}. \tag{5.30}$$

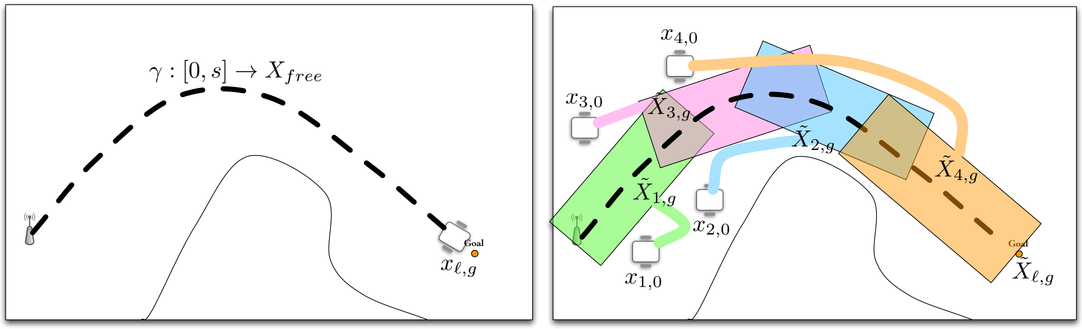
Constructing a final configuration prediction $\tilde{X}_g$ is task-specific. We describe here a method applicable to the telepresence–type application where the final position of a lead node is specified. To determine the configuration prediction $\tilde{X}_g$ we determine configuration predictions $\tilde{X}_{i,g}$ for each robot and compute $\tilde{X}_g$ as the Cartesian product of these individual sets, i.e., $\tilde{X}_g = \prod_{i=1}^N \tilde{X}_{i,g}$. Notice that for the lead robot we can make $\tilde{X}_{\ell,g} = X_{\ell,g} = \{x_\ell \in \mathbb{R}^2 : \|x_\ell - x_{\ell,g}\| < \delta\}$.

Observe now that $X \subset \mathbb{R}^{2N}$ is the Cartesian product $X = \prod_{i=1}^N X_i$ of the N decoupled spaces $X_i \in \mathbb{R}^2$ corresponding to each individual robot. If we further assume a homogeneous team of robots then all robots operate in the same space $X_i = Y$, with a common set of physical obstacles Y_{obs}, and consequently a common free space $Y_{free} = Y \setminus Y_{obs}$. It follows that the joint free space X_{free} is also a Cartesian product of N identical sets Y_{free} minus those configurations for which a network cannot be established with sufficient reliability,

$$X_{free} = (Y_{free})^N \setminus X_{inf}. \tag{5.31}$$

While infeasible network configurations are captured by X_{inf} as given in (5.28), X_{free} can otherwise be described by the free space of individual robots.

To exploit this observation, we first determine an obstacle free path $\gamma : [0, s] \to \mathbb{R}^2$ such that $\gamma(0) = x_0$ is the position of the operating center and $\gamma(s) \in X_{\ell,g}$; see Fig. 5.12a. This path can be determined by a RRT algorithm [47] or other discrete planning algorithms [57]. Since the dimensionality of the space and the goal set $X_{\ell,g}$ are small, it is possible to find this path with small computational cost. The obstacle-free path $\gamma : [0, s] \to Y_{free}$ is split into $N-1$ equal length segments γ_k such that $\gamma_k : [0, s_k] \to \gamma : [\frac{ks}{N-1}, \frac{(k+1)s}{N-1}]$. The



(a)

(b)

Figure 5.12: Illustration of the biased space sampling. Since we only know one component of the goal state $x_{g,\ell}$ and it is expensive to expand our search space in the high-dimensional state of the entire system, it is beneficial to bias our search towards configurations that are deemed likely to succeed. In (a), an obstacle-free path $\gamma : [0, s] \to \mathbb{R}^2$ is found between the access point and goal location. In (b), the path is divided into $N - 1$ segments and enlarged to represent a class of possible goal configurations $\tilde{X}_{i,g}$.

ith robot is then assigned to a segment by the function $k(i)$ based on euclidian distance to its midpoint such that $\sum_{i \neq 0,\ell} \|\gamma_{k(i)}(s/2) - x_{i,0}\|$ is minimized; see Fig. 5.12b. Segments are then enlarged to define the region $\tilde{X}_{i,g}$ for $i \neq 0, \ell$. Since this is a heuristic for the goal configuration, the only requirement on $\tilde{X}_{i,g}$ is that $\gamma_{k(i)} : [0, s] \to \tilde{X}_{i,g}$. A typical choice is

$$\tilde{X}_{i,g} = \{x_i : \min_s \|x_i - \gamma_{k(i)}(s)\| < \tilde{d}_g\}$$

where $\tilde{d}_g$ is a parameter controlling the enlarged size of $\tilde{X}_{i,g}$. The predicted final configuration is then computed as the Cartesian product $\tilde{X}_g = \prod_{i=1}^{N} \tilde{X}_{i,g}$.

This procedure is summarized in Algorithm 4. In lines 1–3, the predicted goal configuration $\tilde{X}_g$ is constructed. A random sample $\hat{\mathbf{x}}$ is then drawn uniformly from $\tilde{X}_g$ with probability p_g or from $X \setminus \tilde{X}_g$ otherwise. It should be noted that the construction of $\tilde{X}_g$ described above is based on the heuristic that a feasible goal configuration in an environment with obstacles will *resemble* a line-of-sight communication chain. Increasing the size of $\tilde{X}_g$ with large values of $\tilde{d}_g$ limits the implication of this assumption.

Algorithm 4 RandomState(X)

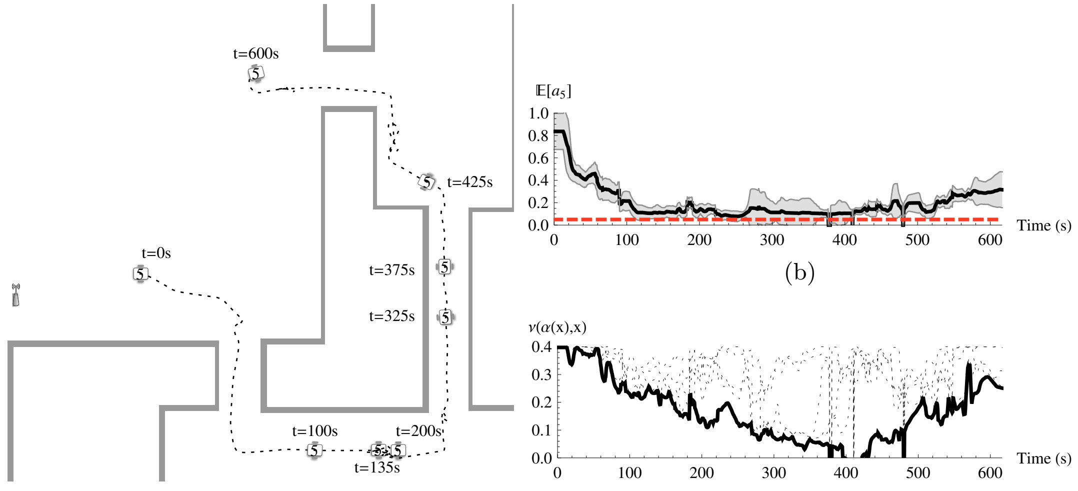
```
Require: Configuration space description $X$, obstacle-free path $\gamma(s) \to \mathbb{R}^2$ such that $\gamma(0) = x_0$ and $\gamma(s) = x_{\ell,g}$, probability $p_g$.
$\tilde{X}_{\ell,g} = \{x_\ell \in \mathbb{R}^2 : \|x_\ell - x_{\ell,g}\| < \delta\}$
$\tilde{X}_{i,g} \leftarrow \text{Enlarge}\left(\gamma_{k(i)}\right)$
$\tilde{X}_g \leftarrow \prod_{i=1}^{N} \tilde{X}_{i,g}$
$p \leftarrow \text{Uniform}[0, 1]$
if $p > p_g$ then
    $\hat{\mathbf{x}} \leftarrow \text{Uniform}(X \setminus \tilde{X}_g)$
else
    $\hat{\mathbf{x}} \leftarrow \text{Uniform}(\tilde{X}_g)$
end if
return $\hat{\mathbf{x}}$
```

5.4.3 Simulation Results

The randomized motion planner is able to find feasible configurations that allow target servicing at positions not attainable with the local control approach from Section 5.2. It can additionally provide a feasible sequence of configurations to get to the target configuration. A centralized controller is employed to drive individual robots through the sequence of feasible network configurations in a coordinated fashion.

To test the global planner on a system with one fixed access point and 5 robots, we introduce a sequence of task potential functions $\Psi_1(\mathbf{x}), \Psi_2(\mathbf{x}), \ldots, \Psi_M(\mathbf{x})$ that require the *lead* node x_5 to visit a sequence of positions while the remaining 4 robots act as relays to support end-to-end communication with the access point of $a_{5,min} = 0.05$. As with the local control examples in Section 5.2.3, we require that this end-to-end rate be satisfied with probability $\epsilon = 0.6$. The problem is made more complicated by the introduction of obstacles that not only block robot motions but also degrade received signal strength by 9.5 dBm when line-of-sight is lost as in (M2).

Algorithm 2 is queried to find a feasible path $\sigma_i : [0, s] \to \mathbb{R}^{10}$ for each task $\Psi_i(\mathbf{x})$ in order such that $\sigma_i(0) = \sigma_{i-1}(s)$ and $\sigma_1(0) = \mathbf{x}_{init}$. The trajectory of the lead node x_5 for the concatenation of paths $\sigma = \sigma_1 | \ldots | \sigma_M$ is depicted in Fig. 5.13a. After solving for a feasible path, $\dot{\mathbf{x}}^{des}(t)$ is computed so that $\mathbf{x}(t)$ follows σ. Before considering the state of



(a)

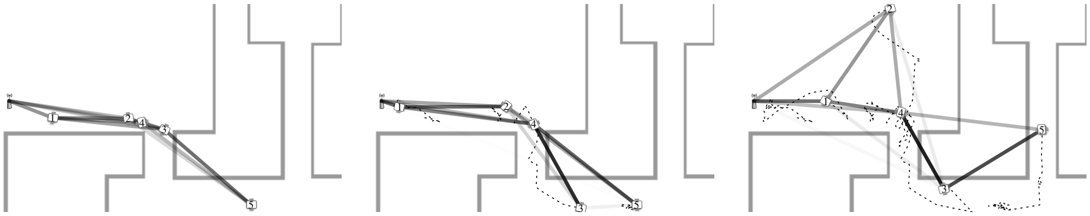
(b)

(c)

Figure 5.13: Global planning with 5 robots in a complex environment. (a) depicts the solution trajectory for the lead node x_5. Positions of other nodes in $\mathbf{x}$ are omitted for clarity. (b) depicts the expected end-to-end rate $a_5(\boldsymbol{\alpha}(\mathbf{x}(t)), \mathbf{x}(t))$ for x_5 with an envelope representing $\pm 0.5\sigma$ (to approximately correspond with $\epsilon = 0.6$). (c) depicts the probability margin $\nu(\boldsymbol{\alpha}(\mathbf{x}(t)), \mathbf{x}(t))$ as a solid line with individual probabilities $P(a_i > a_{i,min}) - \Phi(\epsilon)$ shown dashed.

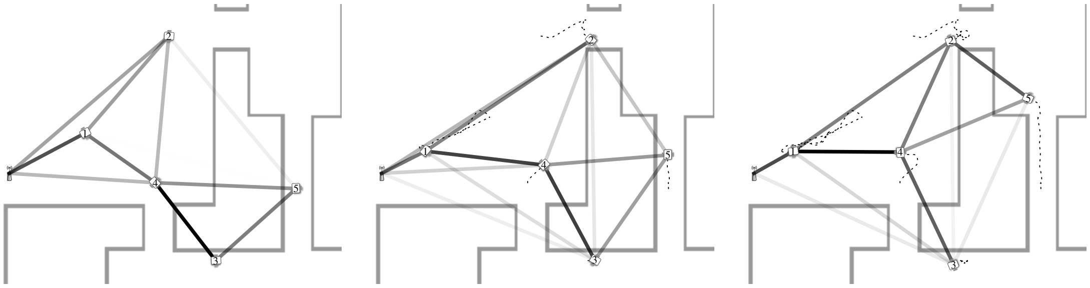
the full system, note that $\nu(\boldsymbol{\alpha}(\mathbf{x}(t)), \mathbf{x}(t)) > 0$ for nearly the entire trajectory as depicted in Fig. 5.13c. The three distinct points where the probability margin drops below zero are situations where the system is operating at the edge of its feasible workspace and individual robot control fails to keep the node precisely on the planned path. While optimal solutions to $\boldsymbol{\alpha}(\mathbf{x}(t))$ effectively maximize the probability margin for a particular $\mathbf{x}(t)$, the global planner only satisfies these constraints. Large probability margin corresponds to robustness to controller errors as well as stochastic channels.

The interplay between expected end-to-end rate $\bar{a}_i$ and variance $\tilde{a}_i$ and their relationship with the satisfaction of probability-based constraints can be observed in Fig. 5.13b. Early in the trajectory, $t < 100$ s, the expected end-to-end rate can be maximized such that even with a large variance, $\nu(\boldsymbol{\alpha}(\mathbf{x}), \mathbf{x}) > 0$. However, as the system extends farther from the access point, expected end-to-end rate decreases and the variance of the solution must be decreased.



(a) $t = 135$ s (b) $t = 200$ s (c) $t = 300$ s

Figure 5.14: Snapshots of the team controlling to follow the feasible path σ for time $t = 135$ s – 300 s. For this interval, X_g requires the lead robot x_5 to reach the designated point in the right hallway. The line weight between nodes indicates the magnitude of each allocation for the optimal robust routing solution.



(a) $t = 325$ s (b) $t = 375$ s (c) $t = 425$ s

Figure 5.15: Snapshots of the team controlling to follow the feasible path σ for time $t = 325$ s – 425 s. For this interval, X_g requires the lead robot x_5 to reach the designated point in upper-right corner. The line weight between nodes indicates the magnitude of each allocation for the optimal robust routing solution. This segment of the deployment specifically represents reconfiguration behavior not possible with local control – the movement of node x_2 to provide connectivity to node x_5 as it moves to minimize $\Psi(\mathbf{x})$.

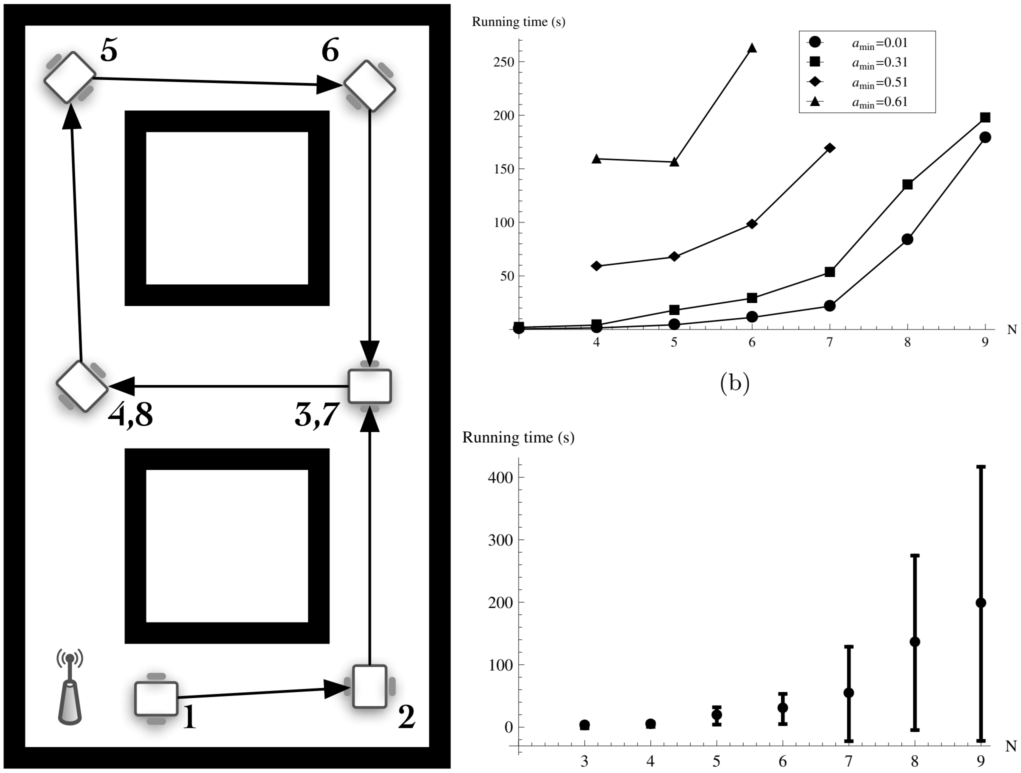
Figure 5.14 depicts three snapshots from the trajectory control $\mathbf{x}(t)$ to follow the feasible path σ during the times 135 s $\leq t \leq$ 300 s. In this example, the progress of x_5 is stopped until x_3 moves to support better end-to-end communication at time $t = 200$ s.

Figure 5.15 depicts three snapshots from the trajectory control $\mathbf{x}(t)$ to follow the feasible path σ during the times 324 s $\leq t \leq$ 425 s. This segment of the deployment is included to demonstrate the large-scale shifts in topology of the communication network that are found with global planning methods. Specifically, in the configuration at $t = 135$ s depicted in Fig. 5.15a, the heuristically determined goal region from (5.31) is achieved by nodes x_1, x_4, x_3, *i.e.* the *lower* line of sight path. However, in order to support the network constraints for x_5 at time $t = 425$ s as depicted in Fig. 5.15c, the entire network must reconfigure to support the *upper* line of sight path. In a sense, it is these dramatic changes to the topology of the underlying communication network that are only found by global planning.

The computational cost of exploring a large or complex environment can be high. The primary expense for our motion planning problem is the verification of a feasible state which can require the solution of several SOCP optimization problems per iteration as described in Section 5.4.1. With regards to the running time of Alg. 2, we note that it is difficult to characterize the performance of randomized search algorithms. One factor is the complexity of X_{free} which is determined both by physical obstacles in the environment as well as the constraints placed on feasible network configurations, *e.g.* solution to $\boldsymbol{\alpha}(\mathbf{x})$. Another component in determining running time is the "planning-horizon" or, roughly, the number of states that must be expanded in order to find a solution. For example, the global planning problem illustrated in Fig. 5.13 is the result of 5 sequential global plans with a cumulative running time of 1200 s.

In order to better characterize the performance of our global planning algorithm, we construct a benchmark task that can be solved many times with different problem parameterizations. The task, depicted in Fig. 5.16a, requires the lead robot to visit a series of positions in the environment (labeled 1–8) while communicating data at a specified rate, a_{min}, to the operating center located near waypoint 1. We parameterize the task by the number of robots N and the end-to-end rate of communication that must be maintained, a_{min}. Point-to-point communication is simulated according to model (M2) and end-to-end rates are maintained with probability $\epsilon = 0.8$.

The performance is measured by the running time to compute the series of network configurations necessary for the lead robot to visit its sequence of waypoints. The average performance is depicted in Fig. 5.16b based on 10 trials per task parameterization. As expected, increasing the number of robots adds to the complexity of both the individual SOCP solutions as well as the randomized search algorithm. Increasing the minimum end-to-end rate, a_{min}, has a similar effect on the complexity. Intuitively, increased a_{min}



(b)

(a)

(c)

Figure 5.16: Running time for benchmark environment with global planner solving the task depicted in (a). The average running time for tasks with different a_{min} and number of robots N are depicted in (b). The variance of the running time for a particular task is depicted in (c).

corresponds to increasing the complexity of the space of feasible configurations. This increases the planning effort necessary to explore the workspace and find a feasible path. Additionally, as we increase the end-to-end rate requirement, more agents are necessary for task completion.

Randomized planning algorithms can only offer the guarantee of probabilistic completeness. That is, the probability of finding a solution, *if one exists*, approaches 1 as time spent planning increases. Since there is no precise way to determine when a task cannot be solved with the current configuration, we test for task infeasibility by stopping the planning process after a specified timeout period. For the purposes of this benchmarking, that timeout is 300 s for each subtask. An artifact of this timeout is that tasks in extremely complex spaces (e.g. $a_{min} = 0.51, N = 8$) are not solved though we know a solution exists (e.g. the solution for $a_{min} = 0.51, N = 7$ is a subset of the possible solutions with $N = 8$). As the complexity of the task increases, so does the variance of the running time as depicted in Fig. 5.16c for a particular task, $a_{min} = 0.61$. Since the planning algorithm is randomized, the running time is also random.

5.5 Summary

In Chapter 3, we describe a situational awareness problem statement (3.19) that encodes a mobility task for a team of robots with a potential function $\Psi(\mathbf{x})$ while maintaining a set of communication requirements. Instead of considering communication requirements with an abstraction such as graph connectivity, we require the maintenance of specific end-to-end rates across the network. This problem is difficult for two reasons. First, the problem statement is cast as a joint-optimization problem over all robot positions and routing variables. For N robots and even a single flow of data, this means we are searching a $(2N + N^2)$-dimensional space. Second, the end-to-end-rate capabilities rely explicitly on the performance of point-to-point channels which, as demonstrated in Chapter 4, are difficult to accurately predict.

In this chapter, we address the difficulties in solving (3.19), develop algorithms that find solutions to (3.19), and present simulation results that demonstrate these solutions. We manage the uncertain nature of point-to-point wireless channels by modeling their capability as a random variable and transforming the end-to-end communication constraints of (3.19) into constraints on the probability of exceeding the desired threshold. This leads to a robust reformulation of the problem statement in (5.2). The joint optimization of positions and routing variables in this robust formulation is a non-convex problem due to the models for point-to-point wireless channel capability. However, if we fix the positions of the robots, the optimization of the remaining N^2 routing variables is a convex problem and can be cast as an SOCP and efficiently solved for a unique solution.

Given an efficient way to solve for optimal network routes, we propose two methods for searching the remaining $2N$ position variables that describe the physical configuration of the robots. In the first method, we rely on a gradient controller to minimize an objective function that incorporates the task potential function and a barrier to maintain communication constraints that are computed with the optimal network routing solution. However, as demonstrated through simulation results, gradient-based controllers may drive the system into local minima of the objective function without satisfying the task specification. Dealing with environments that include obstacles complicates matters even further. Consequently, we develop a randomized motion planning algorithm that globally searches for physical configurations that achieve the task specification. By customizing traditional randomized motion planning algorithms to the particularities of end-to-end communication constraints and relying on an optimal solution for network routing variables, we are able to efficiently search the $2N$-dimensional space of physical configurations in order to find deployments that satisfy the task specification in real indoor environments.

The work presented in this chapter makes several key contributions to the existing connectivity control literature for teams of mobile agents that require communication. First, we move entirely away from the concept of a disc-based model of reliable communication and adopt a probabilistic model. In fact, we make no assumption about the specific model for point-to-point communication except that, for a given source and receiver position, a prediction of the supported communication rate is made with mean and variance. Second, rather than rely on graph-based abstractions for connectivity, we consider quality-of-service metrics that can be cast as probabilistic constraints on the end-to-end rates of the network. In this chapter, we present simulation results that are based on communication models obtained as a result of extensive experiments in Chapter 4. However, these simulations do not address the instantaneous realization of the end-to-end rate in a real network which leads us to pursue experimental verification of the entire system next in Chapter 6.

Chapter 6

Robust Routing & Mobility Control Experiments

In this chapter we describe a set of experiments used to validate our approach to the situational awareness problem described in Section 3.2. The communication modeling developed in Chapter 4 and algorithms developed in Chapter 5 enable implementation of the system architecture depicted in Fig. 3.7. Experiments allow us to verify that the instantaneous realization of point-to-point communication links yield end-to-end rates that satisfy our design requirements.

6.1 Methodology

The full implementation of the system architecture in Fig. 3.7 relies on a closed-loop structure where the model of point-to-point communication capability is continually updated as measurements are collected. Here, we realize an instantiation of this architecture where point-to-point communication measurements are collected a priori and used to fit a probabilistically correct model of communication for the environment. For now, we adopt model (M2) as described in Chapter 4 to make predictions of received signal strength and supported communication rate that are a function of the distance and line-of-sight nature of the link connecting transmitter with receiver.

We rely on centralized implementations of the algorithms for concurrent solutions to the robust routing and mobility control problem that are presented in Chapter 5. Since the algorithms we present in Chapter 5 implicitly maintain a connected network of agents, coordinated control commands can be robustly routed through the wireless network. Furthermore, a centralized implementation is not a shortcoming for the problem sizes we consider, i.e. 5 to 20 agents, since typical scalability problems such as state aggregation are not yet an issue.

Each experiment consists of a situational-awareness type task, e.g. (3.19), that requires a single lead robot, indexed by ℓ, to visit one or more locations in the environment while maintaining a desired end-to-end communication rate, $a_{\ell,min}$, with a fixed operating center. This type of task is amenable to a straightforward quadratic task potential function, $\Psi(\mathbf{x}) = \|x_\ell - x_{\ell,g}\|^2$, where $x_{\ell,g}$ represents the desired location for the lead robot. The algorithms introduced in Chapter 5 yield feasible configurations for the team – $\boldsymbol{\alpha}(t)$ and $\mathbf{x}(t)$ which represent the network and physical configurations respectively. During an experiment, each robot probes the communication channels with its neighbors to determine actual instantaneous measurements of the point-to-point received signal strength at a rate of 5 Hz. This data is logged locally and aggregated after each experiment to compute the supported communication rate $\hat{R}_{ij}(t)$ between node i and j at time t. Using these measurements in conjunction with the network routing solution $\boldsymbol{\alpha}(t)$, we can predict the actual supported end-to-end rate at time t for each node i, $\hat{a}_i(\boldsymbol{\alpha}(t), \mathbf{x}(t))$.

On a real system, the implementation of stochastic routing is such that node i sends packets that are part of flow k to node j with probability α_{ij}^k. The rate interpretation of this routing policy is that traffic for flow k from node i to node j assumes the rate $\alpha_{ij}^k R_{ij}$. This rate of traffic affects the end-to-end rate of both node i and node j and our robust routing solution is such that $\mathrm{P}\left[a_i^k(\boldsymbol{\alpha}, \mathbf{x}) > a_{i,min}\right] \geq \epsilon$ is satisfied optimally. While we have not focused on the particulars of such an implementation, for the purposes of these experiments we will assume that each node can meter its outgoing traffic so that the actual rate $\alpha_{ij}^k \hat{R}_{ij}$ does not exceed the expected rate $\alpha_{ij}^k \bar{R}_{ij}$. This assumption is reasonable since we are typically concerned with actual point-to-point rates $\hat{R}_{ij}$ that are worse than the expected value.

Recall that the problem statement in (5.2) requires that

$$\mathrm{P}\left[a_i^k(\boldsymbol{\alpha}, \mathbf{x}) \geq a_{i,\min}^k\right] \geq \epsilon$$

for all nodes i. Indeed, the structure of end-to-end rates with stochastic routing dictates that in our experimental verification each node must be able to maintain

$$\hat{a}_i(\boldsymbol{\alpha}(t), \mathbf{x}(t)) > a_{i,min} \tag{6.1}$$

with probability ϵ. To achieve the desired end-to-end rates, *all* nodes in the team must satisfy this constraint. Thus, in experimental analysis we will evaluate (6.1) across the duration of the experiment to determine the percent of time $\hat{a}_i(\boldsymbol{\alpha}(t), \mathbf{x}(t)) > a_{i,min}$ and use this as a metric for the success of that trial.

6.2 Local Control

The algorithm for local control described in Section 5.2 does not take obstacles or collisions with other robots into account. Accordingly, we modify the velocity search algorithm, i.e. Algorithm 1, to limit individual robot velocities so that they remain out of collision with each other over the next T seconds.

We verify the local control algorithms on a four robot team where one robot acts as the fixed operating center, x_0, two act as relays, x_1, x_2, and one is the lead robot, x_3. The lead robot must maintain a rate of $a_{3,min} = 0.4$ with probability $\epsilon = 0.9$ to the fixed operating center. The relay nodes must satisfy rates $a_{1,min} = a_{2,min} = 0$ to maintain queue stability as they support the flow of data from the lead node to the fixed operating center. The task potential function for this experiment is $\Psi(\mathbf{x}) = \|x_3 - (4, 0)\|$. In this experiment, the time horizon is set to be $T = 2$ s, i.e. all robot velocities are constrained such that network constraints are probabilistically guaranteed for the next 2 s. Another way of interpreting the time horizon T is that the current robust routing solution is feasible for T seconds before a new solution must be computed and, more importantly in the case of a centralized implementation, deployed to the individual agents.

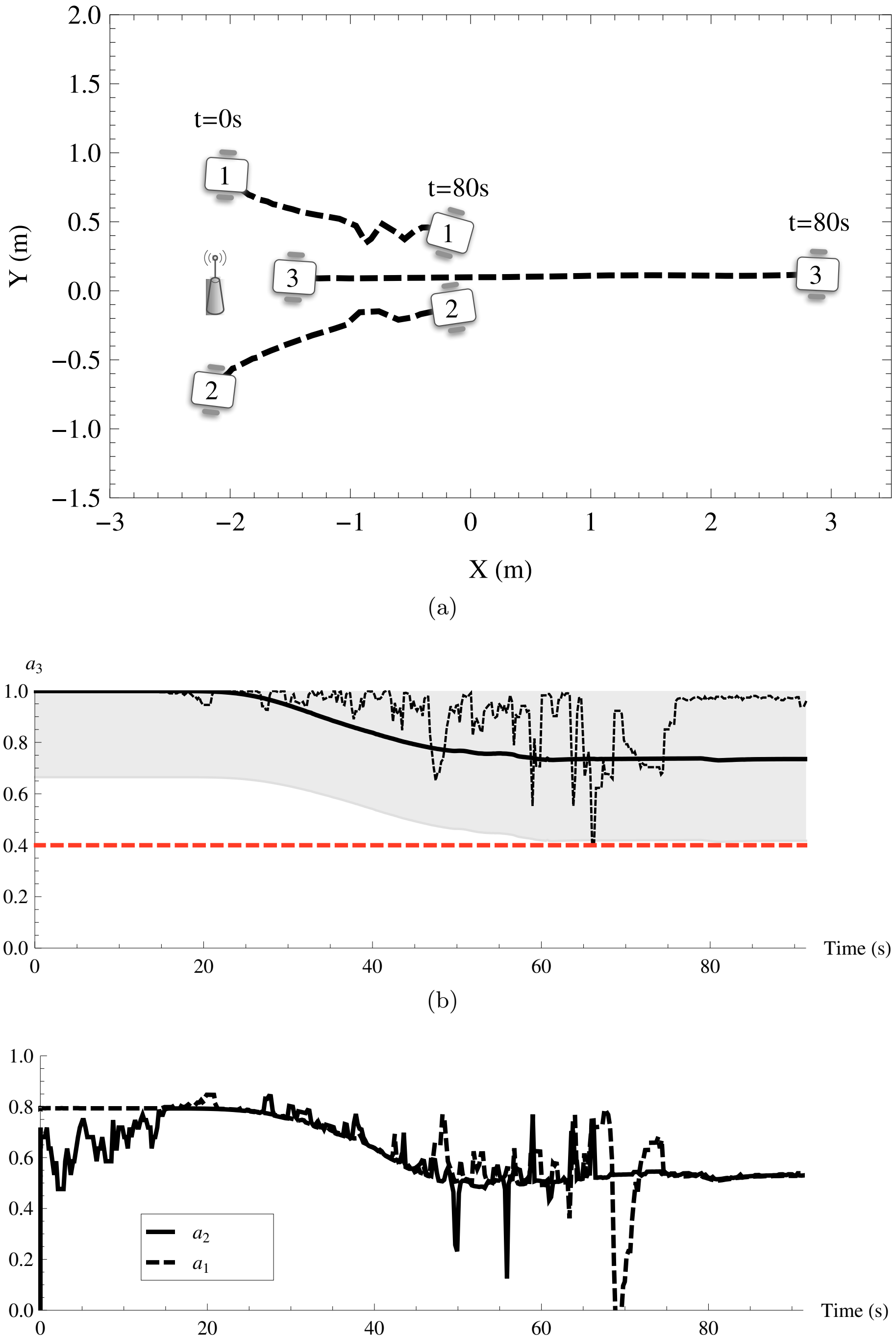
The robot trajectories for an experimental trial in the Levine building are depicted in Fig. 6.1a. Actual end-to-end rates $\hat{a}_i(\boldsymbol{\alpha}(t), \mathbf{x}(t))$ are depicted in Figs. 6.1b and 6.1c. By $t = 80$ s, the system reaches a stationary point and x_3 cannot move to further minimize $\Psi(\mathbf{x})$ without violating the constraint $\nu(\boldsymbol{\alpha}(\mathbf{x}), \mathbf{x}) \geq 0$.

6.3 Global Planning

For global planning experiments, we return to the two environments considered in Chapter 4 – the Levine and Towne buildings on the University of Pennsylvania campus. We perform tests in both environments to demonstrate the versatility of our methods to indoor environments of drastically different construction. In each experiment we demonstrate the ability of the global motion planning algorithm to find a sequence of network configurations that enable the lead robot to visit a series of pre-specified waypoints. We then verify that the instantaneous end-to-end rate $\hat{a}_i(\boldsymbol{\alpha}(\mathbf{x}(t)), \mathbf{x}(t)) \geq a_{i,min}$ with probability ϵ over the duration of the experiment.

Levine Environment

Figure 6.2 depicts the series of waypoints that the lead node, x_5 must visit in the Levine building experiment. Four additional mobile nodes, x_1, x_2, x_3, x_4 are available to relay data back to the fixed access point indicated in the lower left of Fig. 6.5. In this experiment, the lead node must maintain and end-to-end rate of $a_{5,min} = 0.25$ with probability $\epsilon = 0.75$ while each relay node must maintain end-to-end rates greater than zero.



(a)

(b)

(c)

Figure 6.1: Local control experiment with two relay nodes. (a) depicts the trajectory of each node. (b) and (c) depict the rates $\hat{a}_i(\boldsymbol{\alpha}(t), \mathbf{x}(t))$ for the lead and relay nodes respectively. In (b), the solid line with shaded envelope depicts $\bar{a}_3$ and variations that occur with probability $\epsilon = 0.9$ based on $\tilde{a}_3$. The dashed line represents the instantaneous end-to-end rate $\hat{a}_3$

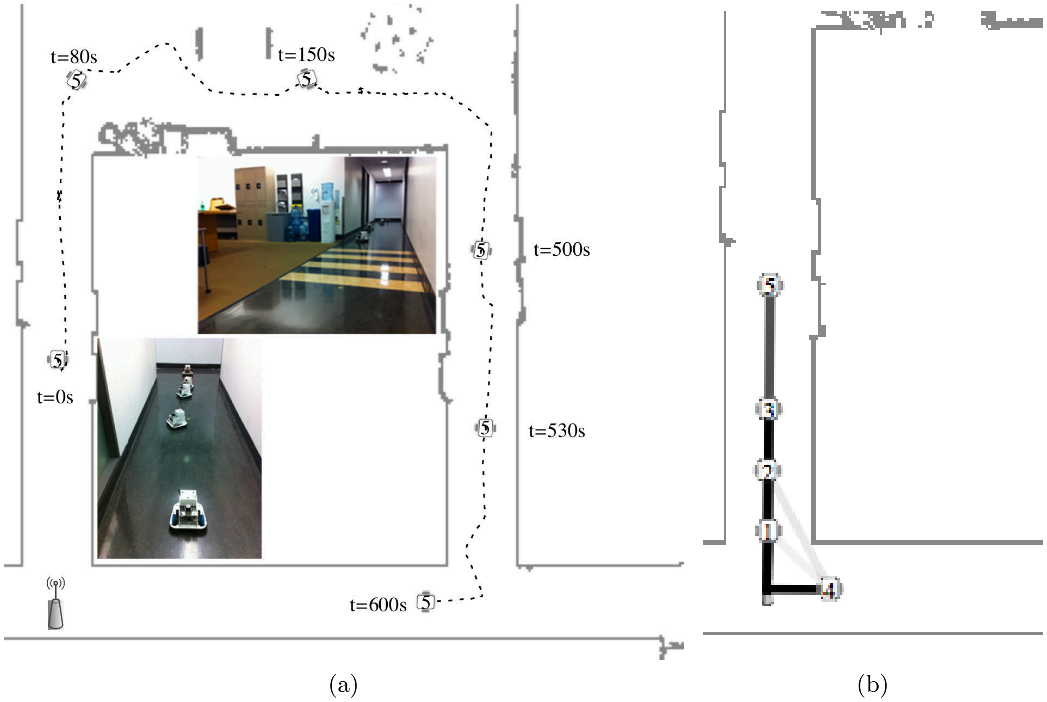
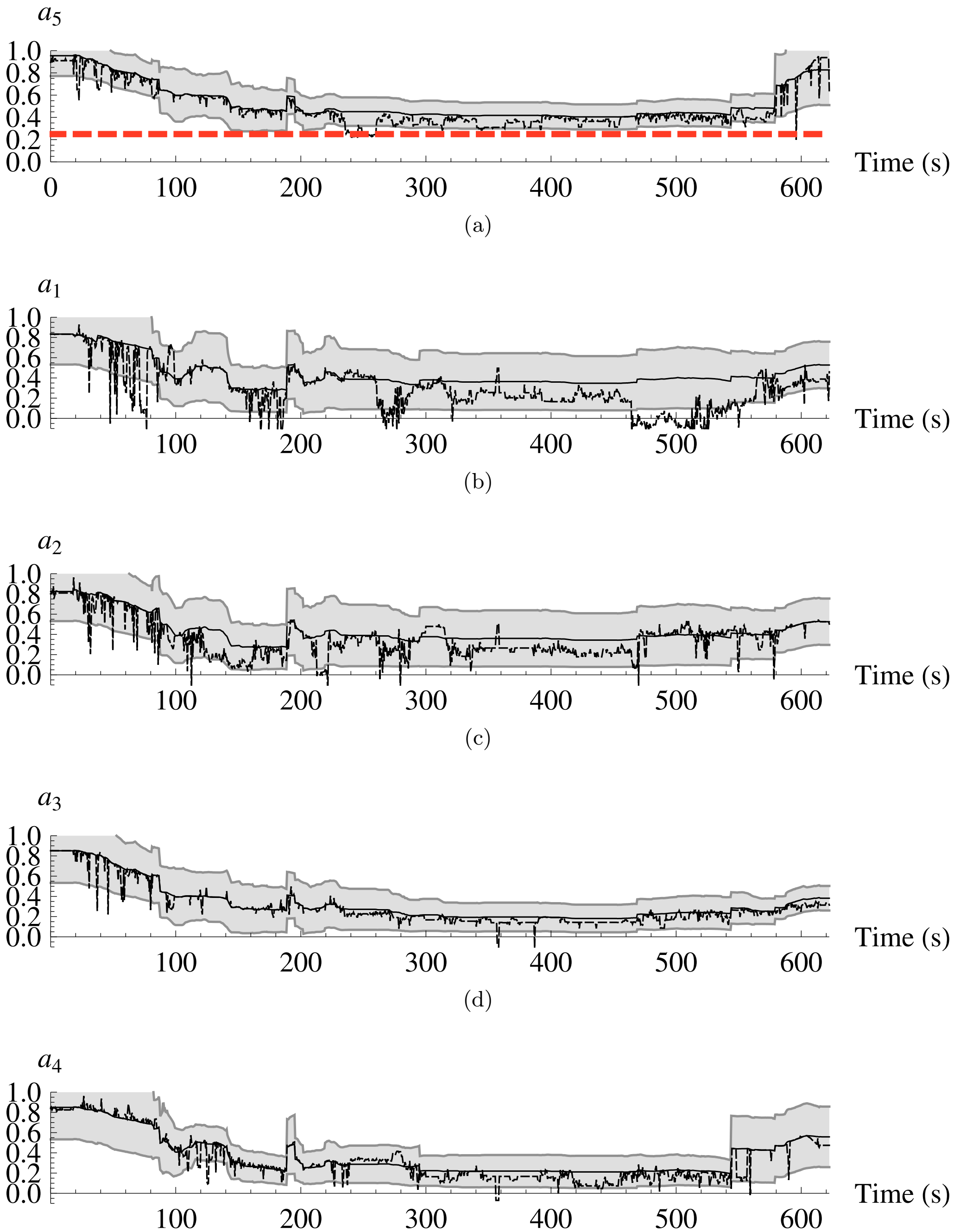


Figure 6.2: The task specification for the Levine building experiment. It requires the lead node, x_5 to follow a sequence of waypoints that take it in a loop through the environment as depicted in (a). The initial configuration of the team is depicted in (b).

The predicted and measured end-to-end rates of each node are depicted in Fig. 6.3. First, notice that the instantaneous rate $\hat{a}_5(\boldsymbol{\alpha}(t), \mathbf{x}(t))$ is almost always above its minimum threshold of $a_{5,min} = 0.25$. In fact, it drops below the minimum threshold only 2.9% of the time, well within the allowable 25% for this problem specification. However, for that rate to be maintained in an end-to-end sense across the network, each node must be able to support the necessary rate margin $a_{i,min}$. The corresponding fraction of time spent below the minimum threshold for each of the instantaneous node rates $\hat{a}_1, \hat{a}_2, \hat{a}_3, \hat{a}_4$ is 9.2%, 0.8%, 0.3%, and 0.6%.

Representative network configurations are depicted in Fig. 6.4. In Fig. 6.4a, at $t =$ 100 s, the predicted goal state $\tilde{X}_g$ assumes the shortest line of sight path which is the left hallway, i.e. a similar result to the reactive methods in our local control algorithm. As the system transitions to Fig. 6.4b, where the lead node x_5 has been tasked to a waypoint in the right hallway, the prediction for $\tilde{X}_g$ shifts to a chain of relays going through the



(a)

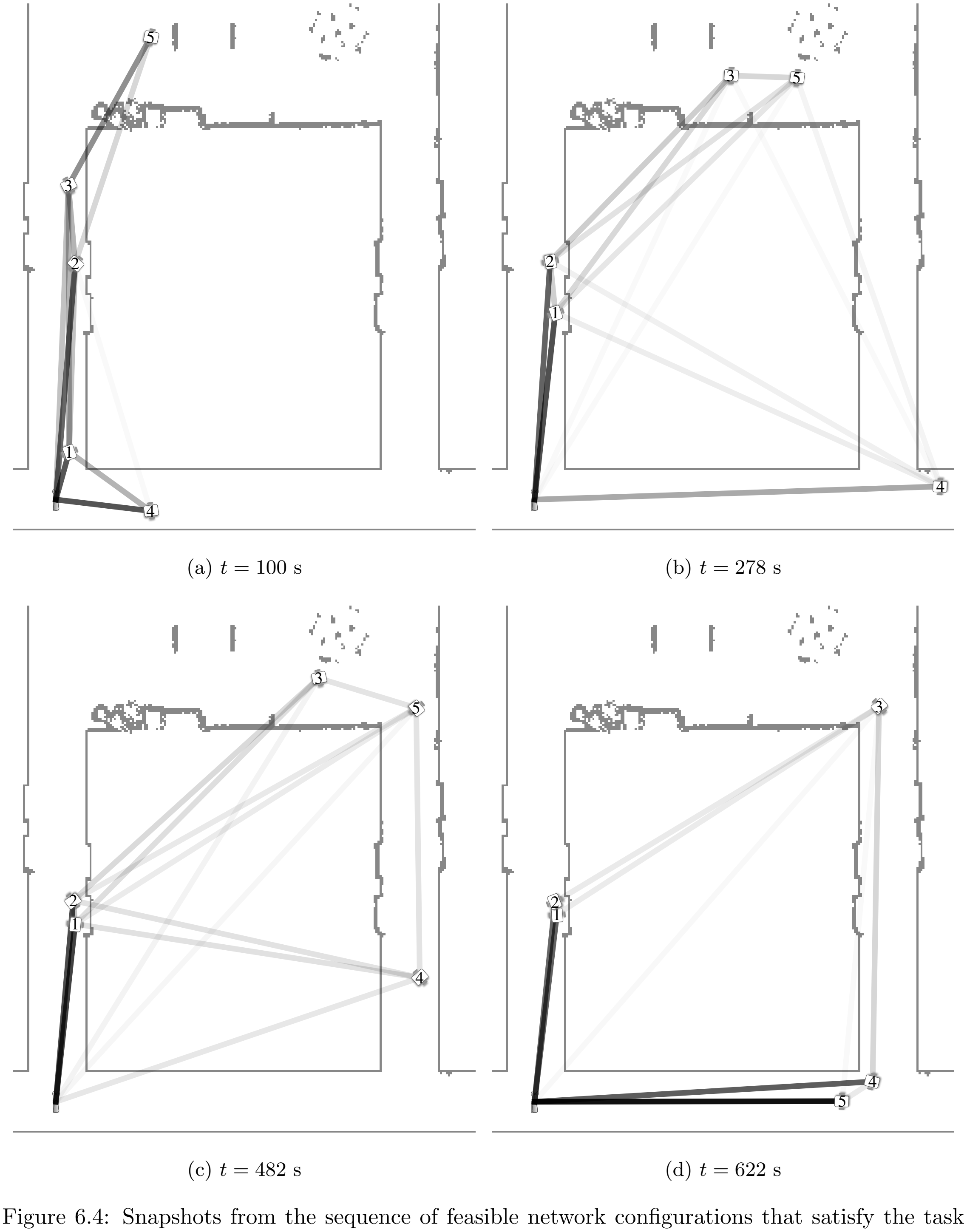
(b)

(c)

(d)

(e)

Figure 6.3: The end-to-end rates of the nodes during the Levine building experiment depicted in Fig. 6.2. (a) depicts the prediction, $\bar{a}_5, \tilde{a}_5$, and instantaneous, $\hat{a}_5$, end-to-end rate for the leader and (b) – (e) depict the instantaneous rates of the relay nodes. In each plot, the solid line with shaded envelope depicts $\bar{a}_i$ and variations that occur with probability $\epsilon = 0.75$ based on $\tilde{a}_i$. The dashed black line represents the instantaneous end-to-end rate $\hat{a}_i$. The dashed red link in (a) depicts the threshold $a_{5,min} = 0.25$.



(a) $t = 100$ s

(b) $t = 278$ s

(c) $t = 482$ s

(d) $t = 622$ s

Figure 6.4: Snapshots from the sequence of feasible network configurations that satisfy the task depicted in Fig. 6.2.

right hallway. This shift in the basic topology of $\tilde{X}_g$ focuses exploration of the joint state space so that x_4 moves towards a configuration that will *lower* the performance of the network over the short term. As node x_5 completes the desired loop, it utilizes x_4 as a relay channel and is able to maintain the desired end-to-end rate. It is this dramatic shift in network topology that highlights the advantage of our global planning approach as we are able to accomplish continuous end-to-end rate maintenance that would not be possible with a purely reactive method.

Towne Environment

In the second experiment, depicted in Fig. 6.5, the structure of the environment does not require any dramatic shifts in the topology of the network routing as in the prior experiment. However, in this experiment we train the communication model on a subset of the environment, i.e. the labeled "training region" in Fig. 6.5. After deploying to the leader state indicated at time $t = 900$ s in Fig. 6.5, the leader is tasked with returning to its initial condition.

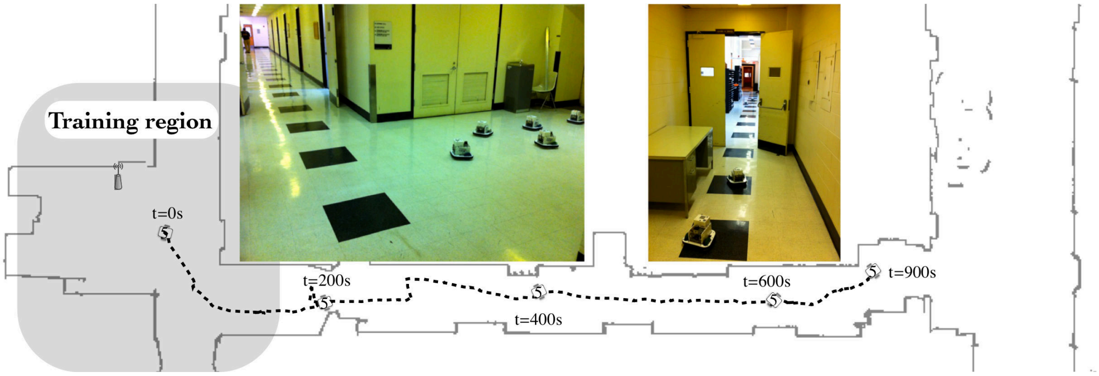


Figure 6.5: The task specification for the Towne building experiment. It requires the lead node, x_5 to follow a sequence of waypoints that take it as far as possible from the fixed operating center.

The predicted and measured end-to-end rates of each node are depicted in Fig. 6.6. Despite the limited training data, the performance in this environment is comparable to the Levine experiment. The empirical failure rate for each node is 8.2%, 3.3%, 17%, 0.5%, and 16.7% for a_1 through a_5 respectively. The actual end-to-end rate, $\hat{a}_5$, for $300 \leq t \leq$

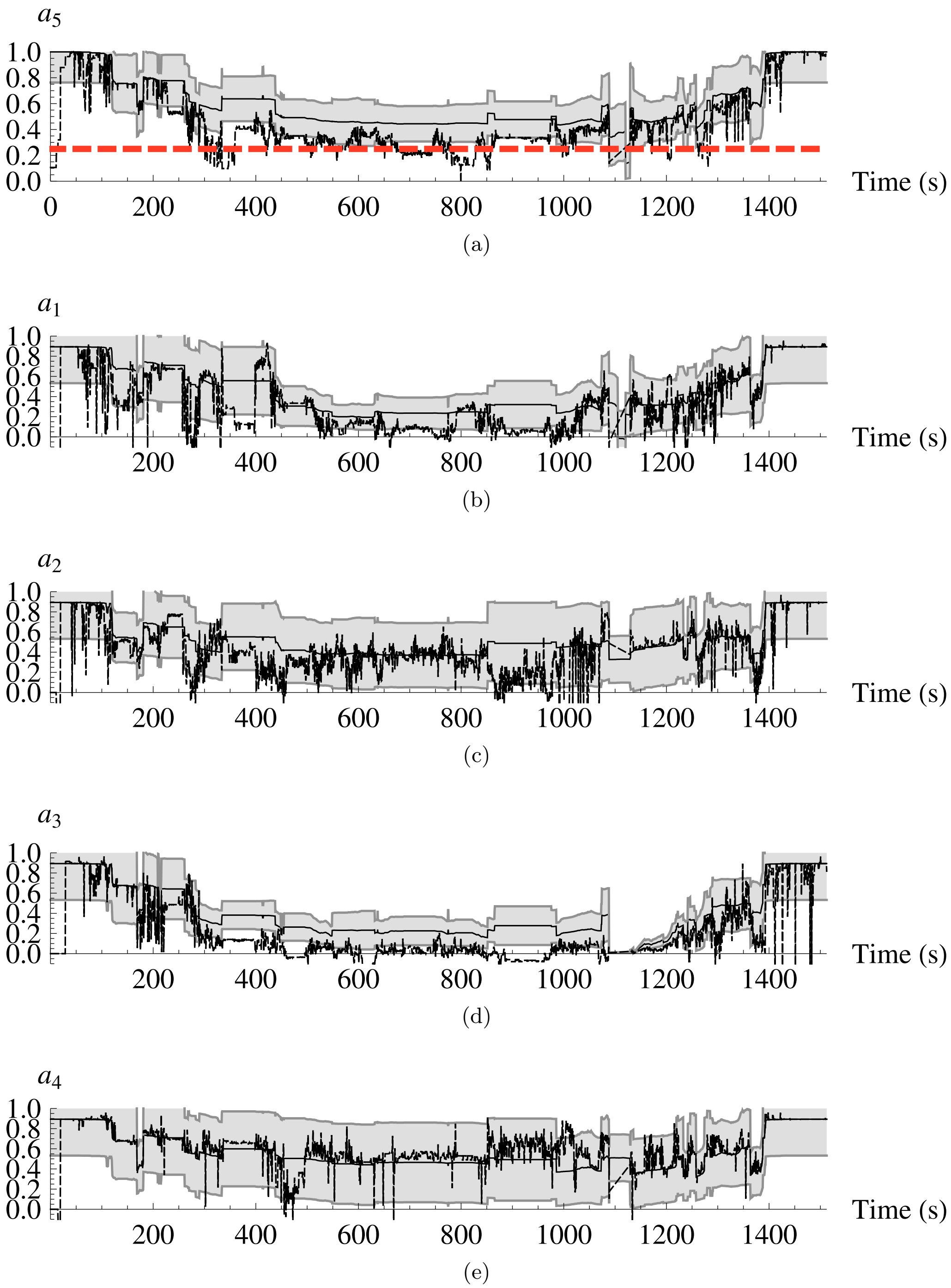


Figure 6.6: The end-to-end rates of the nodes during the Towne building experiment depicted in Fig. 6.5. (a) depicts the prediction, $\bar{a}_5, \tilde{a}_5$, and instantaneous, $\hat{a}_5$, end-to-end rate for the leader and (b) – (e) depict the instantaneous rates of the relay nodes. In each plot, the solid line with shaded envelope depicts $\bar{a}_i$ and variations that occur with probability $\epsilon = 0.75$ based on $\tilde{a}_i$. The dashed line represents the instantaneous end-to-end rate $\hat{a}_i$. The dashed red link in (a) depicts the threshold $a_{5,min} = 0.25$.

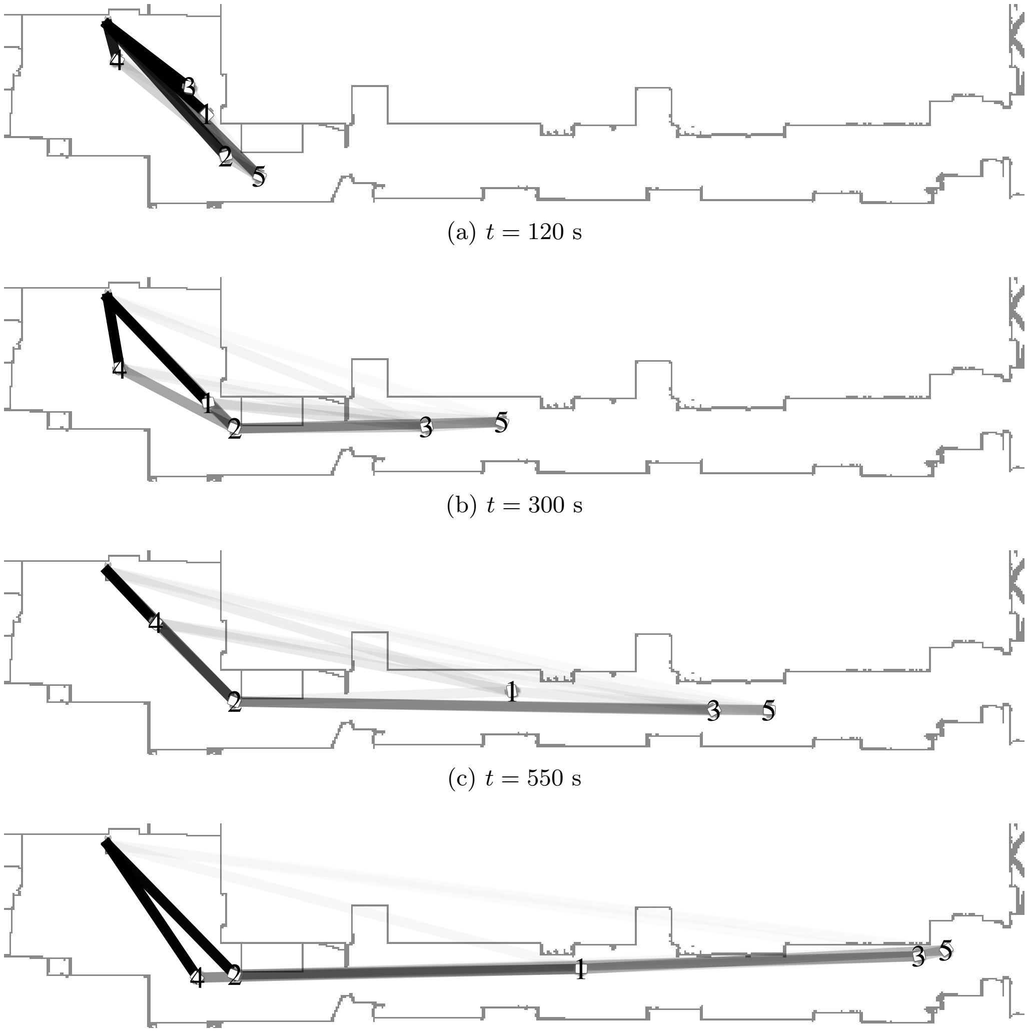
900 s is consistently offset from the prediction since this robot is operating far from the training region. This result is expected based on the communication modeling experiments conducted in Chapter 4. However, it also demonstrates an important capability of the robust methods we employ since we are still able to maintain communication within the desired parameters when the model systematically overestimates the performance of point-to-point links. The use of a more complex radio communication model, such as the Gaussian process method (M5), would incorporate increased uncertainty in this region and require more conservative configurations from the global planner – leading to improved performance.

Representative network configurations for this experiment are depicted in Fig. 6.7. Due to the communication model and geometric nature of this environment, line-of-sight links are favored as the lead robot extends farther from the fixed operating center. Though the physical state of the robots resembles a pure multi-hop solution due to the global planning heuristic for $\tilde{X}_g$, the robust network solution still splits traffic and routes data over links with lower expected rates in order to decrease the variance of the end-to-end rate.

6.4 Summary

In this chapter, we address the experimental verification of algorithms developed in Chapter 5 for concurrent control of mobility and network routing. These algorithms are focused on providing solutions to a situational awareness task that require the completion of a physical objective while maintaining specified end-to-end communication rates across the network. In this chapter, we focus on verifying that instantaneous realizations of the random end-to-end rates across the network are maintained. By conducting a series of experiments in complex, indoor environments, we show that the end-to-end rates based on real wireless channels do exceed the threshold with the desired probability.

In these experiments, we chose to employ the distance-based model with fixed attenuation for non-line-of-sight channels, i.e. (M2), presented in Chapter 4. While GP-based models may provide tighter predictions, this model is provides fast predictions and re-



(a) $t = 120$ s

(b) $t = 300$ s

(c) $t = 550$ s

(d) $t = 900$ s

Figure 6.7: Snapshots from the sequence of feasible network configurations that satisfy the task depicted in Fig. 6.5.

quires minimal training. Furthermore, the success of these results serves to demonstrate the robustness of our network routing and mobility control methodology in that it succeeds despite a coarse model of point-to-point wireless communication. This raises an important question for future work which will seek to determine how model complexity in the prediction of point-to-point wireless channels affects the solution to situational awareness problems within our framework.

Finally, these experimental results are heavily dependent on the capabilities of the decentralized testbed presented in Chapter 2. However, the success and flexibility of these experiments is also the product of a large base of software that implements the system architecture pictured in Fig 3.7. For example, this software base consists of a network layer for the radios, efficient motion planning algorithms, a simulation architecture for wireless communication, and many miscellaneous tools for the definition and setup of each experimental trial. The result of this, not explicitly referenced in this chapter, is that it requires minimal effort to take an arbitrary set of robots to a new environment, build a coarse model of communication, and perform network deployments that maintain the specified quality-of-service metrics. This capability opens the door to a wide range of multi-robot experiments in nearly arbitrary environments.

Chapter 7

Conclusions

In this thesis, we addressed the problems inherent to wireless networking for a team of mobile robots. We do this in a comprehensive way that focuses on real implementation and places significant emphasis on experimental validation. Consequently, an intermediate accomplishment of this work was the development of a flexible experimental testbed for distributed multi-robot systems. In fact, this testbed impacts multi-robot experiments beyond the scope of this thesis and has also been applied to research on multi-robot manipulation and perception tasks. The main accomplishment of the work in this thesis is a system-based solution to the situational awareness problem which relies on a probabilistic model of point-to-point communication capabilities and a set of algorithms that find concurrent solutions the network routing and mobility control for a team of robots. The methods we have described in this thesis provide the basic communication capabilities necessary for the robust operation of teams of robots in complex environments.

7.1 Summary of the Thesis

In Chapter 2 we explore a classification of system architectures based on tools for the abstraction of communication capabilities. Specifically, we look at how this translates to the design criteria and evolution of our experimental testbed. We describe the details and core capabilities of our decentralized multi-robot testbed that allows for a wide range of experimental validation across a large indoor environment.

Building on the design of our experimental testbed, in Chapter 3 we examine two case studies in coordinated control that have been evaluated on our testbed. The first example relies on the nearest-neighbor abstraction for communication and addresses the problem of multi-robot manipulation via caging. We are able to demonstrate how a decentralized team of anonymous agents are able to approach and transport a payload through a field of obstacles in a robust manner. This chapter concludes with the definition of a situational awareness problem statement that serves to explicitly motivate the remainder of this thesis. Our solution to this problem relies on the development of two core capabilities: probabilistic communication modeling and concurrent solutions to the network routing and mobility control problems.

In Chapter 4 we develop a framework to provide probabilistic communication predictions for arbitrary point-to-point channels in an indoor environment. By providing an experimental comparison of several candidate models, we draw conclusions about their effectiveness. Two key results area product of this evaluation. First, it is possible to rely on geometric maps of the indoor environment to increase the predictive performance of a communication model. Second, though the use of a Gaussian process-based model increases the complexity of training and predictions, it offers the powerful capability to represent model uncertainty in regions where training data has not been collected.

This chapter also demonstrates two methods for relative localization. In the first method, we rely directly on a Gaussian process-based model and show the ability to infer the location of a static source. In the second method, we ignore typical signal-strength based localization schemes that require extensive training and instead rely on the observation that physical antennas are not isotropic and have a predictable bearing-based attenuation pattern. By implementing a multi-hypothesis filter, we are able to demonstrate the relative localization of a team of mobile agents. Furthermore, this approach serves to demonstrate that modeling point-to-point channels in $\mathbb{R}^4$ discards a significant amount of useful information that should be considered for future predictive models.

Chapter 5 addresses the second core capability necessary for our solution to the situational awareness problem presented in Chapter 3, i.e. a concurrent solution to the network routing and mobility control problem. Our goal is to develop a unified understanding of the interplay between network routing solutions and control of node mobility. We do this by first relying on the statistical point-to-point predictions of wireless channel performance to design an optimal robust network routing solution given a fixed physical configuration for the team of robots. Then, we employ two techniques that seek to jointly solve the mobility control problem with the network routing problem. Both techniques utilize the fact that our solution to the network routing problem is robust to node motions that can lead to increased channel uncertainty.

The first technique relies on an optimization problem with a representation of network constraints in optimization objective function through a barrier function. Gradient descent is then used to move towards minima of the combined objective function in a component-wise fashion, i.e. iterative optimization of the physical configuration with optimization of the network solution. While we are able to demonstrate that this method is successful in maintaining the desired network connectivity, it is not able to deal with complex environments and suffers from issues of local minima in the combined objective function that arise as a result of complicated network topologies.

The second technique we propose for concurrent solutions to the network routing and mobility control problem takes cues from random motion planning algorithms that have found great success in the robotics literature. We have developed a random motion planning algorithm that explores the joint configuration space of a team of robots that are subject to the solution of our robust routing algorithm formulated as a convex optimization program. Using this algorithm, we are able to demonstrate the ability to efficiently solve robust deployment plans for teams with up to ten mobile robots, i.e. operating in the physical space of $\mathbf{x} \in \mathbb{R}^{20}$. These solutions demonstrate the ability of randomized planning to discover non-trivial shifts in network topology that allow us to address situational awareness problems that are infeasible with reactive methods.

Finally, in Chapter 6, we present a series of experiments that validate our solution to the situational awareness problem described in Chapter 3. We deploy a team of robots utilizing our decentralized testbed based on the result of our global planning algorithm. During deployment, each robot continuously measures the instantaneous point-to-point communication capability with its neighbors. Using this data, we are able to provide the first experimental validation of communication connectivity maintenance with real wireless channels. Furthermore, our definition of connectivity is such that it will support a task-specified end-to-end rate for each node.

7.2 Main Contributions

In this thesis, we address communication problems faced by teams of networked robots and make several key contributions. We reformulate traditional connectivity maintenance problems to present a generic situational awareness problem that combines a physical task specification with quality-of-service constraints for networked communication. By incorporating quality-of-service constraints into the problem formulation, we gain the ability to satisfy the underlying physical task while ensuring the necessary rates of communication across the network. This provides a stronger statement about communication capabilities than traditional methods that seek only to maintain graph-theoretic notions of connectivity. However, our approach does increase the complexity of the problem in two ways.

Frist, any solution to our situational awareness problem requires accurate prediction of point-to-point communication capability between arbitrary points. This leads us to develop Gaussian process-based methods for building spatial maps of point-to-point wireless channel performance. These GP-based models offer two key advantages over existing models. Primarily, they can represent variations in signal strength that cannot be predicted without extremely accurate geometric maps and information about the dielectric properties of the environment. Secondarily, GP-based models provide the capability to represent increased uncertainty about communication capability in unexplored regions of the environment.

Second, the introduction of quality-of-service constraints dictates that network routing solutions must be considered in parallel with robot motion. This increases the complexity of the non-convex situational awareness problem. However, we are able to demonstrate that for a fixed configuration of robot positions, the networking problem can be formulated and efficiently solved as a convex second order cone program. Using this fact, we propose two methods for searching the space of physical configurations to find solutions to the situational awareness problem. The second of these methods, a randomized global search technique, incorporates both communication and physical constraints to find solutions to the network deployment problem. Furthermore, we provide experimental verification that the actual end-to-end rates during deployment satisfy the specified constraints. This represents a major step forward in the capabilities of networked teams of robots.

A key component of our experimental verification is the demonstration that end-to-end rates are maintained even when planning relies on a coarse model for point-to-point capabilities. This feature can be leveraged to support a generalization of the specific problem we consider in this thesis. If we consider a scenario where prior knowledge of the communication model is low and there is no time to train a specific model, planning for deployment can rely on a very coarse model that under approximates point-to-point performance. In this way, our framework will find and utilize conservative team configurations that are robust to large uncertainties in point-to-point channels.

Furthermore, while this thesis focuses on a particular instance of the situational awareness problem statement, i.e. a single lead node sending a flow of data back to a fixed operating center, the methods we describe are immediately flexible to address several scenarios. For instance, the introduction of multiple communication flows or destinations is facilitated by additional routing variables and end-to-end rate constraints in the robust routing formulation. Likewise, a large class of physical tasks can be represented by task potential functions. The nature of our randomized motion planning algorithm makes it amenable to any constraint whose satisfaction can be queried. On the other hand, some applications require more complicated physical task specifications. In these cases, the gradient-based control approach is more natural as it can incorporate desired robot velocities from an arbitrary controller.

A parallel contribution of this thesis is the development of an experimental testbed that supports a wide range of multi-robot research. Experimental work with the multi-hop communication paradigm for a team of robots requires a complex system of algorithms that interface with hardware, re-create traditional network layers, and provide individual robot capabilities such as localization and navigation. This is in addition to the algorithms that are the focus of research. Consequently, the key component of a testbed for multi-robot research is a modularized software architecture. This allows for the development of stable algorithms that provide low-level capabilities such as self-localization and navigation. An easy to use software architecture additionally allows for efficient hardware abstractions and for algorithm development in simulation with few simplifying assumptions. We have demonstrated this capability through the experiments described in this thesis as well as a large body of previous work [30, 29, 70, 20, 9, 71, 100].

7.3 Future Work

While we have experimentally demonstrated a nearly complete solution to the situational awareness problem as we posed it, there are several challenges facing deployment of this type of system in a real-world environment. Here we identify key directions for future work that address these challenges.

In Chapter 4, we examine several candidate models for radio signal propagation and point-to-point communication capability with a relatively large stochastic component to account for uncertainty and small-scale fading. We explicitly modeled large stochastic components so that our robust network routing solutions in Chapter 5 would yield conservative solutions. One clear direction for future work is to find models that allow for tighter bounds on the uncertainty. We see this being achieved in two ways. First, radio communication literature seems to indicate that dominant path models have greater predictive ability than the direct path models we tested [108, 107, 109]. Second, none of our candidate models explicitly allow for the ability to predict non-stationary fading. For a Gaussian process model, this amounts to adopting a non-stationary covariance function which increases complexity but may offer significant improvements in the tightness of our predictions.

The other major challenge that we avoid in our treatment of communication modeling is online or iterative updates to the model parameters during operation. We took a batch approach in Chapter 4 to provide a clear basis for comparison amongst several candidate models based on a static dataset. When we turned to experimental verification of our solution to the situational awareness problem, we assumed an a priori dataset to generate communication predictions. Clearly this assumption will not be valid in many applications. Online solutions to the parameters of our parametric models via extended Kalman filtering [105] or iterative Gaussian-process methods [13] are both likely avenues for success.

As we transition to online models for point-to-point channel prediction, global planning methods will have to be updated accordingly. Since initial deployment plans will be based on a prior models, they may be conservative with respect to the actual point-to-point communication. In fact, it is easy to imagine situations where the necessary deployment is infeasible given the a priori communication model. As online updates to the point-to-point model make predictions more accurate, it is necessary to update or re-plan team deployments. The problem of re-planning within randomized[23] and deterministic frameworks [50] is well-studied in the robotics literature.

In Chapter 5 we begin by introducing a local gradient-based method that seeks to drive the system towards optimal configurations with respect to both the task specification and network utility. We quickly abandon this approach for complex environments in favor of randomized global planning methods that explore the space of feasible configurations without accounting for the optimality of a configuration with respect to the networking properties. A direction for future research is the unification of these approaches where local control techniques can be used in a virtual sense to extend the state expansion capabilities during global planning.

Throughout this work we have focused on a multi-hop communication model but made the implicit assumption that modeling, control, and planning algorithms are implemented in a centralized manner. We argue that this assumption is acceptable given the team sizes we consider and the fact that we explicitly maintain connectivity of the network. However, an important extension of the work in this thesis is the adaptation of our methods and algorithms to decentralized implementations with increased robustness and flexibility.

Bibliography

[1] M. Abt and W.J. Welch. Fisher information and maximum-likelihood estimation of covariance parameters in Gaussian stochastic processes. *Canadian Journal of Statistics*, 26(1):127–137, 1998.

[2] Healthcare Solutions: Real–time Location, Status, and Condition of Assets, Patients and Staff Throughout Your Hospital. WWW page, 2010. http://www.aeroscout.com/content/healthcare.

[3] Applanix Mobile Mapping and Positioning Solutions – Accurately and Reliably Capture and Measure the World Around Us. WWW page, 2010. http://www.applanix.com/.

[4] Ascending Technologies, GmbH, 2010. `http://www.asctec.de`.

[5] A. Atramentov and S. M. LaValle. Efficient nearest neighbor searching for motion planning. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 632–637, 2002.

[6] Abdalkarim Awad, Thorsten Frunzke, and Falko Dressler. Adaptive distance estimation and localization in wsn using rssi measures. In *In 10th EUROMICRO Conference on Digital System Design - Architectures, Methods and Tools (DSD 2007)*, pages 471–478. IEEE, 2007.

[7] J. Biswas and M. Veloso. WiFi Localization and Navigation for Autonomous Indoor Mobile Robots. In *IEEE International Conference on Robotics and Automation, 2010. ICRA '10*, May 2010.

[8] N. Chakraborty and K. Sycara. Reconfiguration algorithms for mobile robotic networks. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5484–5489, 2010.

[9] P. Cheng, J. Fink, S. Kim, and V. Kumar. Cooperative towing with multiple robots. In *Workshop on Algorithmic Foundations of Robotics*, Guanajuato, Mexico, December 2008.

[10] Peng Cheng, Jonathan Fink, and Vijay Kumar. Abstractions and algorithms for cooperative multiple robot planar manipulation. In *Robotics: Science and Systems (RSS)*, Zurich, Switzerland, June 2008.

[11] K.W. Cheung, J.H.M. Sau, and R.D. Murch. A new empirical model for indoor propagation prediction. *IEEE Transactions on Vehicular Technology*, 47(3):996–1001, 1998.

[12] RH Clarke. A statistical theory of mobile-radio reception. *Bell Systems Technical Journal*, 47(6):957–1000, 1968.

[13] L. Csató. *Gaussian Process: iterative sparse approximations.* PhD thesis, University of Aston in Birmingham, 2002.

[14] E. Damosso and L. Correira. *Digital Mobile Radio: COST 231 View on the Evolution towards 3rd Generation Systems.* The European Comission, 1998.

[15] K. Dantu, P. Goyal, and G.S. Sukhatme. Relative bearing estimation from commodity radios. In *2009 IEEE international conference on Robotics and Automation*, pages 3215–3221, 2009.

[16] S. Dasgupta and P.M. Long. Performance guarantees for hierarchical clustering. *Journal of Computer and System Sciences*, 70(4):555–569, 2005.

[17] A. Davids. Urban search and rescue robots: From tragedy to technology. *Intelligent Systems, IEEE*, 17(2):81–83, 2002.

[18] D. DeCouto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multihop wireless routing. In *International ACM Conference on Mobile Computing and Networking*, pages 134–146, San Diego, CA, September 2006.

[19] M. C. DeGennaro and A. Jadbabaie. Decentralized control of connectivity for multiagent systems. In *Proc. 45th IEEE Conference on Decision and Control*, pages 3628–3633, San Diego, CA, December 2006.

[20] Jason Derenick, Jonathan Fink, and Vijay Kumar. Localization using ambiguous bearings from radio signal strength. Submitted to 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2011.

[21] Inc. Digi International. XBee®& XBee-Pro® 802.15.4 OEM RF Modules, 2010. `http://www.digi.com/pdf/ds_xbeezbmodules.pdf`.

[22] Ekahau Real-Time Location System (RTLS) Overview. WWW page, 2010. http://www.ekahau.com/products/real-time-location-system/overview.html.

[23] D. Ferguson, N. Kalra, and A. Stentz. Replanning with rrts. In *IEEE International Conference on Robotics and Automation*, pages 1243–1248. IEEE, 2006.

[24] B. Ferris, D. Fox, and N. Lawrence. WiFi-SLAM using Gaussian process latent variable models. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 2480–2485, 2007.

[25] B. Ferris, D. Hahnel, and D. Fox. Gaussian processes for signal strength-based location estimation. In *Robotics: Science and Systems*, Philadelphia, PA, August 2006.

[26] J. Fink, T. Collins, V. Kumar, Y. Mostofi, J. Baras, and B. Sadler. A simulation environment for modeling and development of algorithms for ensembles of mobile microsystems. In *Proc. of the SPIE Conf. on Micro- and Nanotechnology Sensors, Systems, and Applications*, Orlando, Fl., April 2009.

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

[28] Jonathan Fink, Peng Cheng, and Vijay Kumar. Cooperative towing with multiple robots. In *ASME 2008 International Design Engineering Technical Conference (IDETC)*, New York, August 2008.

[29] Jonathan Fink, M. Ani Hsieh, and Vijay Kumar. Multi-robot manipulation via caging in environments with obstacles. In *2008 IEEE International Conference on Robotics and Automation*, Pasedena, CA, May 2008.

[30] Jonathan Fink and Vijay Kumar. Online methods for radio signal mapping with mobile robots. In *2010 IEEE International Conference on Robotics and Automation (IRCA)*, Anchorage, AK, May 2010.

[31] A. Ganguli, J. Cortes, and F. Bullo. Distributed deployment of asynchronous guards in art galleries. In *American Control Conference*, Minneapolis, MN, 2006.

[32] A. Ghaffarkhah and Y. Mostofi. Communication-aware navigation functions for cooperative target tracking. In *American Control Conference, 2009. ACC'09.*, pages 1316–1322. IEEE, 2009.

[33] A. Goldsmith. *Wireless communications.* Cambridge University Press, 2005.

[34] J. M. Gorce, K. Jaffres-Runser, and G. de la Roche. Deterministic approach for fast simulations of indoor radio wave propagation. *IEEE Transactions on Antennas and Propagation*, 55(3):938–948, March 2007.

[35] J. Graefenstein, A. Albert, P. Biber, and A. Schilling. Wireless node localization based on rssi using a rotating antenna on a mobile robot. In *6th Workshop on Positioning, Navigation and Communication (WPNC 2009)*, pages 253–259, 2009.

[36] M. Güneş, M. Wenig, and A. Zimmermann. Improving manet simulation results - deploying realistic mobility and radio wave propagation models. *12th IEEE Symposium on Computers and Communications*, pages 39–44, July 2007.

[37] A. Howard, S. Siddiqi, and G. S. Sukhatme. An experimental study of localization using wireless ethernet. In *Field and Service Robotics*, volume 24 of *Springer Tracts in Advanced Robotics*, pages 145–153. Springer Berlin, July 2006.

[38] A. Hsieh, A. Cowley, V. Kumar, and C. J. Taylor. Towards the deployment of a mobile robot network with end-to-end performance guarantees. In *IEEE Int'l Conf. on Robotics and Automation*, Orlando, Florida, May 16-18 2006.

[39] M. A. Hsieh. *Communication maintenance and motion control strategies for robot teams in environments with obstacles.* PhD thesis, Univ. of Pennsylvania, Philadelphia, PA, August 2007.

[40] M. A. Hsieh, A. Cowley, V. Kumar, and C. J. Taylor. Maintaining network connectivity and performance in robot teams. *Journal of Field Robotics*, 25(1-2):111–131, January 2008.

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

[42] M.A. Hsieh, S.G. Loizou, and V. Kumar. Stabilization of multiple robots on stable orbits via local sensing. In IEEE, editor, *IEEE International Conference on Robotics and Automation*, April 2007.

[43] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6), June 2003.

[44] M. Ji and M. Egerstedt. Distributed coordination control of multiagent systems while preserving connectedness. *IEEE Transactions on Robotics*, 23(4):693–703, August 2007.

[45] D.B. Johnson and D.A. Maltz. Dynamic source routing in ad hoc wireless networks. *Mobile computing*, pages 153–181, 1996.

[46] G. Kantor, S. Singh, R. Peterson, D. Rus, A. Das, V. Kumar, G. Pereira, and J. Spletzer. Distributed search and rescue with robot and sensor teams. In *Field and Service Robotics*, pages 529–538. Springer, 2006.

[47] S. Karaman, E. Frazzoli, and R. Altendorfer. Incremental Sampling-based Algorithms for Optimal Motion Planning. *Arxiv preprint arXiv:1005.0416*, 2010.

[48] JM Keenan and AJ Motley. Radio coverage in buildings. *British Telecom Technology Journal*, 8(1):19–24, 1990.

[49] L. R. Klein. *A Textbook of Ecometrics.* Row, Peterson, 1953.

[50] S. Koenig and M. Likhachev. D^* lite. In *Proceedings AAAI National Conference on Artificial Intelligence*, pages 476–483, 2002.

[51] A. Krause and C. Guestrin. Nonmyopic active learning of Gaussian processes: an exploration-exploitation approach. In *Proceedings of the 24th international conference on Machine learning*, pages 449–456. ACM New York, NY, USA, 2007.

[52] J.J. Kuffner Jr and S.M. LaValle. RRT-connect: An efficient approach to single-query path planning. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2000.

[53] A. M. Ladd, K. E. Bekris, A. P. Rudys, D. S. Wallach, and L. E. Kavraki. On the feasibility of using wireless ethernet for indoor localization. *IEEE Transactions on Robotics and Automation*, 20(3):555–559, June 2004.

[54] Khanh Tuan Le. Ieee 802.15.4 and zigbee compliant radio transceiver design. http://www.hometoys.com/htinews/feb05/articles/chipcon/zigbee.htm, 2005.

[55] W. Lee and Y. Yeh. On the estimation of the second-order statistics of log normal fading in mobile radio environment. *IEEE Transactions on Communications*, 22(6):869–873, 1974.

[56] J. LeNy and G.J. Pappas. Sensor-based robot deployment algorithms. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 5486–5492. IEEE, 2010.

[57] M. Likhachev, G. Gordon, and S. Thrun. Ara*: Anytime a* with provable bounds on sub-optimality. *Advances in Neural Information Processing Systems*, 16, 2004.

[58] A.G.M. Lima and L.F. Menezes. Motley-keenan model adjusted to the thickness of the wall. *Microwave and Optoelectronics, 2005 SBMO/IEEE MTT-S International Conference on*, pages 180–182, July 2005.

[59] M. C. Lin and J. F. Canny. Efficient algorithms for incremental distance computation. In *Proceedings IEEE International Conference on Robotics & Automation*, 1991.

[60] M. C. Lin, D. Manocha, J. Cohen, and S. Gottschalk. Collision detection: Algorithms and applications. In J.-P. Laumond and M. H. Overmars, editors, *Algorithms for Robotic Motion and Manipulation*, pages 129–142. A.K. Peters, Wellesley, MA, 1997.

[61] M. Lindhé, H. Johansson, and A. Bicchi. An experimental study of exploiting multipath fading for robot communications. In *Robotics: Science and Systems*, Atlanta, GA, June 2007.

[62] W. Lindsey. Error probabilities for rician fading multichannel reception of binary and n-ary signals. *IEEE Transactions on Information Theory*, 10(4):339–350, 1964.

[63] M.S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra and its Applications*, 284(1-3):193–228, 1998.

[64] H. Lundgren, E. Nordström, and C. Tschudin. Coping with communication gray zones in ieee 802.11 b based ad hoc networks. In *Proceedings of the 5th ACM international workshop on Wireless mobile multimedia*, pages 49–55. ACM, 2002.

[65] S. Martinez, F. Bullo, J. Cortes, and E. Frazzoli. On synchronous robotic networks, part i: Models, tasks and complexity notions. In *IEEE Conference on Decision and Control*, pages 2847–2852, Seville, Spain, December 2005.

[66] S. Martínez, F. Bullo, J. Cortés, and E. Frazzoli. On synchronous robotic networks, part ii: Time complexity of renezvous and deployment algorithms. *IEEE Transactions on Automatic Control*, 52(12):2214–2226, 2007.

[67] J.W. McKown and R.L. Hamilton Jr. Ray tracing as a design tool for radio networks. *IEEE Network*, 5(6):27–30, 1991.

[68] N. Michael, J. Fink, and V. Kumar. Experimental testbed for large multi-robot teams: Verification and validation. *IEEE Robotics and Automation Magazine*, 15(1):53–61, March 2008.

[69] N. Michael, J. Fink, S. G. Loizou, and V. Kumar. Architecture, abstractions, and algorithms for controlling large teams of robots: Experimental testbed and results. In *International Symposium of Robotics Research*, Hiroshima, Japan, November 2007.

[70] N. Michael, M. M. Zavlanos, V. Kumar, and G. J. Pappas. Maintaining connectivity in mobile robot networks. In *International Symposium on Experimental Robotics*, Athens, Greece, July 2008.

[71] Nathan Michael, Jonathan Fink, and Vijay Kumar. Cooperative manipulation and transportation with aerial robots. In *Robotics: Science and Systems (RSS)*, Seattle, WA, June 2009. Receieved Best Paper.

[72] B. Mirtich. V-Clip: Fast and robust polyhedral collision detection. Technical Report TR97-05, Mitsubishi Electronics Research Laboratory, 1997.

[73] Y. Mostofi. Communication-aware motion planning in fading environments. In *Proc. Intnl. Conf. on Robotics and Automation*, pages 3169–3174, 2008.

[74] Y. Mostofi, A. Gonzalez-Ruiz, A. Gaffarkhah, and L. Ding. Characterization and modeling of wireless channels for networked robotic and control systems - a comprehensive overview. In *Proc. IEEE/RSJ Intnl. Conf. on Intelligent Robots and Systems*, pages 4849–4854, 2009.

[75] Y. Mostofi and P. Sen. Compressed mapping of communication signal strength. In *IEEE Military Communications Conference (MILCOM 2008)*. IEEE, 2009.

[76] A. Neskovic, N. Neskovic, and G. Paunovic. Modern approaches in modeling of mobile radio systems propagation environment. *IEEE Communications Surveys*, 3(3):1–12, 2000.

[77] G. Notarstefano, K. Savla, F. Bullo, and A. Jadbabaie. Maintaining limited-range connectivity among second-order agents. In *Proc. 2006 American Control Conference*, pages 2124–2129, Minneapolis, MN, June 2006.

[78] R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proc. of the IEEE*, 95(1):215–233, January 2007.

[79] R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9), Sptember 2004.

[80] Open 802.11s Consortium, 2010. `http://open80211s.com/`.

[81] N. Patwari, A. O. Hero, M. Perkins, N. S. Correal, and R. J. O'Deaa. Relative location estimation in wireless sensor networks. *IEEE Transactions on Signal Processing*, 51(8):2137–2148, August 2003.

[82] G. A. S. Pereira, V. Kumar, and M. F. Campos. Decentralized algorithms for multi-robot manipulation via caging. *International Journal of Robotics Research*, 23(7/8):783–795, 2004.

[83] C. Perkins, E. Belding-Royer, and S. Das. Rfc3561: ad hoc on-demand distance vector (aodv) routing. *Internet RFCs*, 2003.

[84] Y.A. Qi, A.H. Abdel-Gawad, and T.P. Minka. Sparse-posterior Gaussian Processes for general likelihoods. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*, pages 450–457. Citeseer, 2010.

[85] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully B. Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. Ros: an open-source robot operating system. In *International Conference on Robotics and Automation*, Open-Source Software workshop, 2009.

[86] T.S. Rappaport et al. *Wireless communications: principles and practice*, volume 207. Prentice Hall PTR New Jersey, 1996.

[87] C.E. Rasmussen and C.K.I. Williams. *Gaussian processes for machine learning.* Springer, 2006.

[88] W. Ren and R. W. Beard. Consensus seeking in multi-agent systems under dynamically changing interaction topologies. *IEEE Transactions on Automatic Control*, 50(5), May 2005.

[89] A. Ribeiro, Z.-Q. Luo, N. D. Sidiropoulos, and G. B. Giannakis. Modelling and optimization of stochastic routing for wireless multihop networks. In *Proc. 26th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pages 1748–1756, Anchorage, Alaska, May 2007.

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

[91] P. Rong and M. L. Sichitiu. Angle of arrival localization for wireless sensor networks. In *SECON'06: Proceedings of the 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Networks*, pages 374–382, Piscataway, NJ, USA, 2006. IEEE Press.

[92] M. Schuresko and J. Cortes. Distributed motion constraints for algebraic connectivity of robotic networks. *Journal of Intelligent and Robotic Systems*, 56(1-2):99–126, September 2009.

[93] Rajeev Shorey, A. Ananda, Mun Choon Chan, and Wei Tsang Ooi. *Mobile, Wireless and Sensor Networks: Technology, Applications and Future Directions.* Wiley, John & Sons, Inc., 2006.

[94] E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. *Advances in Neural Information Processing Systems*, 18:1257, 2006.

[95] D. P. Spanos and R. M. Murray. Motion planning with wireless network constraints. In *Proc. 2005 American Control Conference*, Portland, OR, June 2005.

[96] K.N. Sridhar, S. Hao, M.C. Chan, and A.L. Ananda. Egress: Environment for generating realistic scenarios for simulations. *Tenth IEEE International Symposium on Distributed Simulation and Real-Time Applications*, pages 15–24, Oct. 2006.

[97] S. Srirangarajan, SFA Shah, and A.H. Tewfik. DIRECTIONAL BEACON BASED POSITIONING SYSTEM USING RF SIGNALS. In *16th European Signal Processing Conference (EUSIPCO 2008)*, volume 1, page L2, Lausanne, Switzerland, August 2008.

[98] I. Stojmenovic, A. Nayak, and J. Kuruvila. Design guidelines for routing protocols in ad hoc and sensor networks with a realistic physical layer. *IEEE Communications Magazine*, 43:101–106, March 2005.

[99] E. Stump, A. Jadbabaie, and V. Kumar. Connectivity management in mobile robot teams. In *Proc. IEEE International Conference on Robotics and Automation*, pages 1525–1530, Pasadena, CA, May 2008.

[100] Ethan Stump. *Control for Localization and Visibility Maintenance of an Independent Agent Using Robotic Teams.* PhD thesis, University of Pennsylvania, 2009.

[101] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics.* The MIT Press, Cambridge, MA, 2005.

[102] Vicon Motion Systems, Inc., 2010. `http://www.vicon.com`.

[103] R. Wahl, O. Staebler, and M.J. Gallardo. Requirements for indoor building databases to increase the accuracy of the propagation results. *Mobile and Wireless Communications Summit, 2007. 16th IST*, pages 1–4, July 2007.

[104] C. Walder, K.I. Kim, and B. Schölkopf. Sparse multiscale Gaussian process regression. In *Proceedings of the 25th international conference on Machine learning*, pages 1112–1119. ACM, 2008.

[105] G. Welch and G. Bishop. An introduction to the kalman filter. *University of North Carolina at Chapel Hill, Chapel Hill, NC*, 7(1), 1995.

[106] Willow Garage, 2010. `http://www.willowgarage.com/`.

[107] G. Wölfle and FM Landstorfer. Dominant paths for the field strength prediction. In *48th IEEE Vehicular Technology Conference*, volume 1, pages 552–556. IEEE, 1998.

[108] G. Wölfle, FM Landstorfer, R. Gahleitner, and E. Bonek. Extensions to the field strength prediction technique based on dominant paths between transmitter and receiver in indoor wireless communications. *ITG FACHBERICHT*, pages 29–36, 1997.

[109] G. Wölfle, R. Wahl, P. Wertz, P. Wildbolz, and F. Landstorfer. Dominant path prediction model for indoor scenarios. In *German Microwave Conference (GeMIC)*, 2005.

[110] G. Wölfle, P. Wertz, and F. M. Landstorfer. Performance, accuracy and generalization capability of indoor propagation models in different types of buildings. In *IEEE Int. Symposium on Personal, Indoor, and Mobile Radio Communications*, Osaka, Japan, September 1999.

[111] Y. Wu, A. Ribeiro, and G. B. Giannakis. Robust routing in wireless multi-hop networks. In *Conference on Information Sciences and Systems*, pages 637–642. Johns Hopkins Univ., March 14-16, 2007.

[112] Kiran Yedavalli, Bhaskar Krishnamachari, Sharmila Ravula, and Bhaskar Srinivasan. Ecolocation: A sequence based technique for rf localization in wireless sensor networks. In *In Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks (IPSN*, pages 285–292. Inc, 2005.

[113] Jie Yin, Qiang Yang, and L.M. Ni. Learning adaptive temporal radio maps for signal-strength-based location estimation. *IEEE Transactions on Mobile Computing*, 7(7):869 –883, July 2008.

[114] Michael Zavlanos, Alejandro Ribeiro, and George Pappas. Mobility and routing control in networks of robots. In *Conference on Decision Control*, Atlanta, GA, December 15–17 2010.

[115] M.M. Zavlanos and G.J. Pappas. Potential fields for maintaining connectivity of mobile networks. *IEEE Transactions on Robotics*, 23(4):812–816, 2007.

[116] M.M. Zavlanos and G.J. Pappas. Distributed connectivity control of mobile networks. *IEEE Transactions on Robotics*, 24(6):1416–1428, 2008.

[117] S. Zickler and M. Veloso. RSS-Based Relative Localization and Tethering for Moving Robots in Unknown Environments. In *IEEE International Conference on Robotics and Automation*, May 2010.