Distributed Adaptive Control of Mobile Sensor Networks: Coverage and Estimation Algorithms

Submitted in partial fulfillment of the requirements

of the degree of

Doctor of Philosophy

of the
Indian Institute of Technology Bombay, India
and
Monash University, Australia

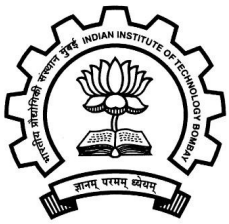
by

Rihab Abdul Razak

Supervisors:

Dr. Srikant Sukumar (IIT Bombay)

Dr. Hoam Chung (Monash University)



The course of study for this award was developed jointly by
Monash University, Australia and the Indian Institute of Technology Bombay, India
and was given academic recognition by each of them.
The programme was administrated by The IITB-Monash Research Academy

(Year 2020)

To my family and friends

Declaration

I declare that the work reported in this thesis represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I declare that I have properly and accurately acknowledged all sources used in the production of this report. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Notice 1

Under the Copyright Act 1968, this thesis must be used only under the normal conditions of scholarly fair dealing. In particular, no results or conclusions should be extracted from it, nor should it be copied or closely paraphrased in whole or in part without the written consent of the author. Proper written acknowledgement should be made for any assistance obtained from this thesis.

Notice 2

I certify that I have made all reasonable efforts to secure copyright permissions for third-party content included in this thesis and have not knowingly added copyright content to my work without the owner's permission.

Rihab Abdul Razak
IITB ID: 144234003
Monash ID: 26853922
Date : 23 October 2019

Abstract

This thesis deals with distributed adaptive control of mobile sensor networks. Mobile sensor network refers to a network of autonomous agents with sensors which can measure some phenomenon of interest. The particular problems investigated in this thesis are area coverage control problems and scalar field estimation using mobile sensor networks. The control and estimation algorithms we develop are distributed in the sense that each mobile agent or sensor requires only local information (information from the given agent and its neighbouring agents) for computing its control or update laws. The algorithms are adaptive in the sense that the density function is assumed to be unknown and the algorithms are required to adapt for the unknown density function in order to achieve its objectives. Coverage control refers to the deployment of mobile sensors in order to cover a given region in space where some event of interest described by a *scalar field* or a *density function* occurs.

This thesis can be broadly divided into three parts. In the first part of the thesis, we discuss algorithms for adaptive coverage control of differential drive robots using the locational optimization framework where the density function is unknown. We consider the full dynamic model of the differential drive robots for developing the control algorithm. We also extend the adaptive coverage control algorithm to the case where there are uncertainties present in the robot dynamics. The control algorithms are validated using simulations.

In the second part of the thesis, we propose a more general formulation of the coverage control problem, namely minimization of a suitable distance function between density functions. The core idea of this approach is to define an aggregate sensing function for the mobile agents, based on the sensing capabilities of the individual sensors. The solution to the coverage problem is then proposed as the minimization of an appropriate distance function between the aggregate sensing function of the agents and the target density

function. We show that the locational optimization problem can be viewed as a special case of this formulation under certain assumptions. We also consider the $\mathcal{L}^2$-distance as a possible alternative to achieve coverage. We discuss parameter convergence issues and compare the locational optimization framework with the $\mathcal{L}^2$ framework with experimental data using differential drive robots.

In the third and final part of the thesis, we discuss the related problem of estimating the density function or scalar field using mobile sensor networks. The density function is approximated using positive definite radial basis functions, and the corresponding parameters are estimated using methods from adaptive control and Lyapunov analysis. Compared to many of the previous works in scalar field estimation, we also provide theoretical guarantees for the convergence of the estimates assuming that the scalar field can be exactly parameterized using the radial basis functions. We develop two types of algorithms: (1) where each mobile sensor estimates the entire parameter vector, (2) where each mobile sensor estimates only a subset of the parameters. The first method is seen to be accurate but computationally expensive, while the second method is less accurate but computationally less expensive. A modified form of the second method is also seen to provide better estimates with only a slight increase in computation. Simulations are used to validate and compare the various algorithms proposed.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent years, cooperative control problems using multiple agents where multiple agents work together to achieve common tasks in an autonomous manner have become very popular. Some of the cooperative tasks which have been investigated include rendezvous where the agents try to converge to a common state, formation control where the agents try to maintain a given spatial formation and coverage control where the agents are deployed to cover a given region of interest. The applications include surveillance, patrolling, environmental monitoring, and sensing, where multiple mobile robots or UAVs are used.

The agents in the cooperative setting are typically assumed to be capable of communicating with other agents. The communication topology is described in terms of a graph with nodes and edges where the nodes correspond to the agents, and two nodes are connected by an edge if the two agents can communicate with each other. In most cases, communication graphs correspond to proximity graphs meaning that two agents communicate if they are close to each other. This also motivates the use of decentralized or distributed control strategies for an efficient solution of multi-agent problems where the control laws of individual agents are determined by the information exchange with their neighbouring agents.

The type of problems that are of interest in this thesis are sensing tasks where the mobile agent or robot is equipped with sensors to measure some quantity of interest at the agent's location. In particular, this thesis discusses decentralized or distributed control strategies using such mobile sensors for coverage control and field estimation problems. Coverage is an important cooperative control problem where the objective is deploying

the mobile agents/sensors to optimally cover a given area of interest according to some phenomenon occurring in the region characterized by a scalar field, also called a *density function*. Such problems find applications in sensing, surveillance, and rescue operations. Consider, for example, autonomous agents deployed to sense nuclear radiation in a region, or for monitoring temperature or oil spills in part of the ocean, or networks of UAVs in a reconnaissance mission for detecting enemy presence in a region. The density function may be thought of as giving the probability of the particular event to be sensed. In other applications, the density function may describe the intensity of the phenomenon to be measured. For example, in the case of mobile sensors deployed to sense nuclear radiation over a region, the density function could be the intensity of radiation over the region. In this case, we would like the mobile agents starting at initial positions to converge to an optimal configuration for sensing purpose.

In many cases, such as the nuclear radiation spill scenario, the density function may not be known, and in such cases, we may need to adapt for the unknown density function in the coverage algorithm. In this thesis, we assume that the unknown density function can be linearly parameterized in terms of known functions. We can then use adaptive algorithms to estimate the unknown density function as the mobile sensors move and cover the region. The most widely studied method for achieving coverage is the locational optimization framework, which is introduced in Chapter 2. The region to be covered is partitioned into Voronoi cells, and the optimal configuration in this framework is the centroidal Voronoi configuration, where the agents converge to the centroids of their partitions. The first part of this thesis is concerned with this problem of adaptive coverage control of differential drive robots.

In many situations where the density function is not known, it may be beneficial or even imperative to estimate the density function. Once the density function is estimated, coverage may then be achieved using the estimated density function, without requiring adaptation for the unknown density function. The second part of this thesis is concerned with the problem of estimating the density function or scalar field accurately. This is different from the adaptive control framework in the first part of the chapter in that here the objective is field estimation rather than coverage. Once the field is estimated accurately, the robots can be deployed for coverage without the need to adapt for the density function. In other words, coverage in the presence of unknown density function

may be achieved in two ways: (1) deploy robots for coverage and in the process adapt for the unknown density function, (2) first deploy robots to explore the region and estimate the density function, and then deployment of robots for coverage can be done. The first part of the thesis deals with the first case above, while the second part of the thesis deals with the second case.

More concretely, the contributions of this thesis consist of the following:

- Adaptive coverage control using nonholonomic (differential drive) mobile robots: We derive adaptive coverage control laws for differential drive robots in the locational optimization framework. The full dynamic model for differential drive robots is assumed. We then consider the case where there are unknown parameters in the robot model and derive adaptive coverage control laws for the same. This also requires the additional computation of the time derivative of the centroids. We study the algorithms extensively using simulations in which we compare the dynamics-based coverage algorithm with the algorithm based on kinematics alone.

- A new formulation of the coverage control problem in terms of distance between density functions: For each agent, we define a *sensing function*, and coverage is obtained by minimizing a distance function between an aggregated sensing function (among the agents) and the target density function. We show that the locational optimization problem can be formulated as the minimization of the K-L divergence with the agent sensing functions being Gaussian functions and the aggregate function formed by taking the max-operator among the agents. In addition, we consider the $\mathcal{L}^2$-distance as a possible means to achieve coverage. In the literature, a consensus-based adaptation law is seen to provide better convergence of parameters. We propose a modified form of consensus-based adaptation law for faster convergence of parameters. We present results from a series of experiments on differential drive robots of the coverage algorithm and compare the performance of $\mathcal{L}^2$ coverage with locational optimization-based coverage, focusing primarily on the performance of parameter estimation.

- Estimation of the density function or scalar field using a mobile sensor network: Here we consider the estimation of a scalar field or density function as the primary objective, as opposed to coverage. We approximate the unknown scalar field as a

linear combination of radial basis functions and use methods from adaptive control and Lyapunov analysis to estimate the parameters. This method of scalar field estimation has not been reported in the literature to the best of our knowledge. We consider mainly two cases: (1) where each mobile sensor estimates the entire parameter vector, which can be computationally heavy, (2) where each mobile sensor estimates only part of the parameter vector, which is computationally much lighter. We also test and verify the algorithms using simulations.

1.1 Structure of the thesis

In Chapter 2, we present a brief survey of literature related to decentralized coverage control and scalar field estimation. We will focus on the formulation of the coverage problem using the locational optimization framework, and this forms the starting point for the subsequent work in this thesis. In particular, we will focus on an adaptive algorithm which assumes that the density function is unknown. In Chapter 3, we extend the decentralized adaptive control algorithm to agents with double integrator dynamics and then to non-holonomic mobile robots. We also discuss the case where the dynamics of the robots have unknown parameters. In Chapter 4, we compare and discuss simulation results for the algorithms presented in Chapter 3. In Chapter 5, we consider a more general formulation of the coverage problem where we define coverage as achieving the minimum of a cost function defined in terms of a distance between the aggregated sensing function of the agents and the target density function. The notion of $\mathcal{L}^2$-distance based coverage is also introduced in this chapter. In Chapter 6, we discuss coverage and adaptation laws for differential drive robots in the $\mathcal{L}^2$ framework. We also discuss the use of consensus terms for improved parameter convergence in the adaptation law in Section 6.2. The experimental results for the coverage algorithms implemented on differential drive robots are presented in Sections 6.3 and 6.4. In Chapter 7, we formulate the problem of estimation of the density function, and we discuss various algorithms for estimation along with simulations for validating the algorithms. Finally, we conclude the thesis with Chapter 8, where we present possible directions for further research.

Chapter 2

Literature Survey

In this chapter, we survey the existing literature on decentralized coverage control in multi-agent systems.

2.1 Decentralized Coverage Control

In [1], the coverage problem is formulated as follows. This also forms the starting point of the work in this thesis. Consider N agents and a convex region $\mathcal{Q} \subset \mathbb{R}^n$ in which the agents are deployed to cover the region for sensing an event of interest. The position of each agent will be denoted by $p_i \in \mathbb{R}^n$, and the corresponding velocities are denoted by $\dot{p}_i$ for each $i \in \{1, 2, \ldots, N\}$. The event of interest is described by a density function $\phi : \mathcal{Q} \to \mathbb{R}_+$ over $\mathcal{Q}$ which describes the relative importance of various regions of $\mathcal{Q}$ with respect to the sensing objective i.e. the regions where ϕ has higher values are more important than the regions with lower values of ϕ and in the optimal coverage configuration, the agents should cover the region in proportion to the value of ϕ. The Voronoi partitions generated by a set of points $\{p_1, p_2, \ldots p_N\}$ is defined as

$$\mathcal{V}_i = \{q \,:\, ||q - p_i|| \leq ||q - p_j||, \quad \forall j \in \{1, 2 \ldots N\}, j \neq i\}. \tag{2.1}$$

Assume that each agent at position p_i covers a region W_i. Also, assume that the sensing reliability of a point in the domain by an agent decreases with the distance of the point from the agent's location. Then optimal coverage is formulated in terms of the following cost function ([1],[2]):

$$\mathcal{H}(p_1, \ldots, p_N, \mathcal{W}_1, \ldots, \mathcal{W}_N) = \sum_{i=1}^{N} \int_{\mathcal{W}_i} ||q - p_i||^2 \phi(q) dq. \tag{2.2}$$

The optimal coverage configuration is then the set of agent positions p_i and the corresponding coverage region of agents W_i such that cost function (2.2) is minimized. It can be shown that for a given set of agent positions, the cost function (2.2) is minimized when the coverage region of the agents $\mathcal{W}_i$ correspond to the Voronoi partitions $\mathcal{V}_i$ generated by the agent positions $\mathcal{P} = \{p_1, p_2, \ldots, p_N\}$ (see [3]). Then we have

$$\mathcal{H}(p_1, \ldots, p_N) = \sum_{i=1}^{N} \int_{\mathcal{V}_i} ||q - p_i||^2 \phi(q) dq. \tag{2.3}$$

This minimization problem is also called *locational optimization*.

The gradient of $\mathcal{H}$ with respect to the agent positions p_i (see [4],[1],[3]) is given by

$$\frac{\partial \mathcal{H}}{\partial p_i} = M_{\mathcal{V}_i}(p_i - C_{\mathcal{V}_i}), \tag{2.4}$$

where

$$L_{\mathcal{V}_i} = \int_{\mathcal{V}_i} q\phi(q) dq, \tag{2.5}$$

$$M_{\mathcal{V}_i} = \int_{\mathcal{V}_i} \phi(q) dq, \tag{2.6}$$

$$C_{\mathcal{V}_i} = \frac{L_{\mathcal{V}_i}}{M_{\mathcal{V}_i}}, \tag{2.7}$$

$M_{\mathcal{V}_i}$ is called the mass of $\phi(\cdot)$ over $\mathcal{V}_i$, $L_{\mathcal{V}_i}$ is the first moment of $\phi(\cdot)$ over $\mathcal{V}_i$ and $C_{\mathcal{V}_i}$ is the centroid of $\phi(\cdot)$ over $\mathcal{V}_i$. From equation (2.4), it can be seen that the cost function (2.3) achieves local minimum when $p_i = C_{\mathcal{V}_i}$, i.e., location of each agent corresponds to the centroid of its Voronoi partition. Such a configuration is called the *centroidal Voronoi configuration*. Thus the objective is to obtain control laws for the agents to cover the region $\mathcal{Q}$ optimally by making the agents converge to a centroidal Voronoi configuration.

Assuming that the agent dynamics are given by

$$\dot{p_i} = u_i, \tag{2.8}$$

the control law is given by

$$u_i = -k_p(p_i - C_{\mathcal{V}_i}), \tag{2.9}$$

where k_p is a positive gain. The control law is proportional to the negative gradient of the cost function and thus drives the robots to a configuration with minimum value of the cost function (see [1]).

Various extensions to this approach have been proposed in the literature. In [5], the authors extend the work of [1] to cover heterogeneous robots whose sensor footprints are not uniform. They also consider robots having a finite size for avoiding collisions, and extend the work for non-convex environments using the notion of geodesic distance. In [6], the authors propose a decentralized coverage control algorithm using the locational optimization framework for the case where the sensing performance of the individual sensors is assumed to be location-dependent. Their work assumes that the sensor performance at a location depends on both the distance of the location from the sensor as well as the location of the sensor rather than the distance of the location from the sensor alone. In [7], the authors discuss coverage for autonomous aerial robotic camera networks. They propose decentralized coverage control strategies for six degree of freedom robotic camera network such that the objective function 'information per pixel' of the camera image is maximized. The objective function assumes the form of the location optimization objective function, and the optimal configuration is the centroidal Voronoi configuration. In [8], the authors propose a modification of the Voronoi partition-based method for exploration and coverage of non-convex domains by multiple robots. They modify the distance metric (which was the euclidean distance) to a graph-based metric obtained by discretizing the domain and constructing a graph with the discrete cells as the nodes and a link between the neighbouring nodes. They also propose entropy-based formulations for exploration of a given unknown environment. In [9], the authors discuss coverage on infrastructure networks such as water pipelines where the sensors could be placed anywhere along the edges of the network or the nodes. Voronoi partition-based method is used to obtain the optimum configuration, and the sensor locations are updated from their initial conditions using a gradient law to reach the optimum configuration. They also propose methods for choosing the initial configuration so that faster convergence is achieved.

In [10], the authors propose a decentralized control algorithm for coverage of robotic sensors where a density function is defined on the domain representing the frequency of random events taking place. A gradient-based law is proposed for maximizing the joint probability of detecting a random event. In [11], the authors talk about task switching in multi-agent networks in a cooperative manner. All the agents are required to carry out multiple cooperative tasks. The consensus-like algorithm determines when to switch to the next task in a synchronous manner. In particular, the paper gives an example where the

agents do an exploration task first and then switch to a coverage task. In [12], the authors discuss coverage for a one-dimensional problem where the sensory distribution function is unknown and stochastic. The coverage cost function is optimized in terms of the *usage probabilities* of the agents, which are the probabilities of the occurrence of the random events to be sensed within each agent's Voronoi region. i.e., it tells how frequently each agent is being used. In [13] and [14], the authors discuss coverage using a method inspired from the behavior of bacteria. The method is compared with the Voronoi partition-based method of [1] and found to give similar results. In [15], the authors propose algorithms for coverage enhancement by removing redundant links in communication graph among the agents and reconfigures the agents using potential function based control laws so that the agents between whom a link was removed separates out of the communication range of each other while keeping those agents close enough between whom a communication link exists. In [16], the authors talk about coverage of a region using virtual springs. The neighbouring robots are assumed to be connected through virtual springs with three parameters - the spring constant, the damping constant, and a nominal length of the spring. The authors talk about the adaptation of the spring constant and damping constant parameters so as to achieve fast convergence to a stable configuration. In [17], the authors propose a game theory-based solution for the coverage control problem assuming the environment is unknown but modelled using a Gaussian mixture model whose parameters are updated as the algorithm runs. The utility function for the individual agents depends on the coverage worth of a given area as well as the energy required to achieve the coverage of the area. In [18], the authors use receding horizon control for the coverage of mobile robots. They develop a distributed algorithm where each robot solves a receding horizon optimal control problem at each discrete time. In [19], the authors talk about persistent awareness coverage. Persistent coverage implies that each agent visits each point in the domain periodically. Persistent coverage problems have also been investigated in [20, 21]. In [22, 23, 24], the authors talk about different generalizations of the coverage problem based on locational optimization by using different assumptions on the agent sensing capabilities. More recently, in [25], the authors develop coverage control laws for time-varying density functions, where the time variation is assumed to be known. In [26], the authors consider coverage problem where agent sensing capabilities are described by polynomial functions and uses tools from algebraic geometry to solve

the problem. In [27], the authors develop a coverage control algorithm for heterogeneous teams of robots deployed to sense multiple phenomena. In [28], the authors consider the extension of the locational optimization-based coverage problem with multiple phenomena, each modelled as a separate density function, and propose distributed algorithms for locally optimal solutions.

2.2 Adaptive Decentralized Coverage Control

In [29] and [30], the authors extend the algorithm of [1] for the case where the density function is unknown. It is assumed that the density function can be written in the form

$$\phi(q) = \mathcal{K}(q)^T a, \tag{2.10}$$

where $\mathcal{K} : \mathbb{R}^n \to \mathbb{R}^m_+$ and $a \in \mathbb{R}^m_+$ is a constant parameter vector. It is assumed that $\mathcal{K}(q)$ is known to all the agents, but a is unknown. The parameter vector a is assumed to be lower bounded and thus satisfies

$$a^i \geq a_{\min}, \qquad i = 1, 2, \ldots, m, \tag{2.11}$$

where a^i is the ith component of a, and $a_{\min}$ is the lower bound assumed to be known.

It is assumed that each agent can measure the value of density function $\phi(q)$ at its current location. Since the true parameter a is not known, each agent will use an estimate of a denoted by $\hat{a}_i$. The corresponding estimated quantities are given by $\hat{\phi}_i(q) = \mathcal{K}(q)^T \hat{a}_i$ which is the agent i's estimate of the density function, $\hat{M}_{\mathcal{V}_i} = \int_{\mathcal{V}_i} \hat{\phi}_i(q) dq$ which is the agent i's estimate of the mass of $\phi(\cdot)$ over $\mathcal{V}_i$, $\hat{L}_{\mathcal{V}_i} = \int_{\mathcal{V}_i} q\hat{\phi}_i(q) dq$ which is the agent i's estimate of $L_{\mathcal{V}_i}$ and $\hat{C}_{\mathcal{V}_i} = \frac{\hat{L}_{\mathcal{V}_i}}{\hat{M}_{\mathcal{V}_i}}$ which is agent i's estimate of the centroid $C_{\mathcal{V}_i}$. The control law for single integrator agents is given by

$$u_i = -k_1(p_i - \hat{C}_{\mathcal{V}_i}). \tag{2.12}$$

In addition the following quantities are defined, ([29]),

$$\Lambda_i(t) = \int_0^t e^{-\alpha(t-\tau)} \mathcal{K}_i(\tau) \mathcal{K}_i(\tau)^T d\tau, \tag{2.13}$$

$$\lambda_i(t) = \int_0^t e^{-\alpha(t-\tau)} \mathcal{K}_i(\tau) \phi_i(\tau) d\tau, \tag{2.14}$$

$$b_i = -\int_{\mathcal{V}_i} \mathcal{K}(q)(q - \hat{C}_{\mathcal{V}_i})^T dq\, \hat{e}_i - \gamma(\Lambda_i \hat{a}_i - \lambda_i), \tag{2.15}$$

where $\mathcal{K}_i(t) := \mathcal{K}(p_i(t))$, $\phi_i(t) = \phi(p_i(t))$ which corresponds to agent i's measurement of the density function $\phi(\cdot)$, $\hat{e}_i = \hat{C}_{\mathcal{V}_i} - p_i$ is the estimated error, α and γ are positive constants. Then the adaptation law for $\hat{a}_i$ is given by

$$\dot{\hat{a}}_i = \Gamma(b_i - I_{\beta_i} b_i), \tag{2.16}$$

where I_{β_i} is a diagonal matrix with diagonal elements given by

$$I_{\beta_i}(j) = \begin{cases} 0 & \text{for } \hat{a}_i(j) > a_{\min}, \\ 0 & \text{for } \hat{a}_i(j) = a_{\min} \,\&\, b_i(j) \geq 0, \\ 1 & \text{otherwise}, \end{cases} \tag{2.17}$$

and $\Gamma > 0$ is positive definite. The adaptation law consists of the term b_i in addition to a projection defined by (2.17) to make sure that the updated parameter value is always greater than the assumed minimum $a_{\min}$.

We now state the main theorem in [29],[30].

Theorem 2.1. *Consider n agents with dynamics given by*

$$\dot{p}_i = u_i.$$

With the control law given by (2.12) and the update law for $\hat{a}_i$ given by (2.16), the following holds:

(a) $\lim_{t\to\infty}(p_i - \hat{C}_{\mathcal{V}_i}) = 0, \quad \forall i \in \{1, 2, \ldots, n\}$,

(b) $\lim_{t\to\infty} \mathcal{K}_i(\tau)\tilde{a}_i(t) = 0, \quad \forall \tau \ s.t\ 0 \leq \tau \leq t \ and\ \forall i \in \{1, 2, \ldots, n\}$,

where $\tilde{a}_i = \hat{a}_i - a$ is the parameter estimation error of the i-th agent.

In [31], the authors extend the work to nonholonomic sensors with the agent models given by the kinematic equations. The authors use the kinematic equations for unicycle robots in polar coordinates. We will consider differential drive kinematics together with the complete dynamics of the robots to derive the coverage and adaptation laws in chapter 3. In [32], a continuum model is considered where a large number of agents forming a swarm is considered as defining a flow. Ideas from optimal transport are used for generating a gradient flow to move the initial configuration to a target configuration. In [33], coverage control is performed by minimizing the statistical distance called the f-divergence. This

work relates to aspects of work in the current thesis in chapter 5, where we generalize the coverage problem in terms of distance functions, though our formulation is more general and we do not consider f-divergence in this thesis.

2.3 Field Estimation using Mobile Sensor Networks

Several works have studied field estimation using wireless sensor networks. See, for example [34, 35]. In [36] the scalar field is assumed to be modelled using a partial differential equation, and finite element methods are used for estimating the field. In [37, 38, 39, 40, 41, 42], the field is modelled as a spatial random process and estimated using samples from the sensor nodes. In [43], field reconstruction is posed as an optimization problem constrained by linear dynamics, and a gradient-based method is used to solve the problem. In [44], the scalar field is assumed to be linearly parameterized in terms of Gaussian basis functions and the measurements from the sensors are fused together to form an estimate for the scalar field. In most of these cases, the sensors are assumed to be fixed and distributed over the region of interest. Usually, a large number of sensors are required to be installed for achieving enough spatial resolution. Using mobile sensor networks can be highly advantageous since they can move around the region of interest and collect measurements adaptively; the number of sensors required is greatly reduced. In [45, 46], scalar field estimation is done with mobile sensor network by fusing sensor measurements using consensus filters. The region of interest is divided into cells, and the scalar field value at each cell is estimated by fusing sensor measurements of the various mobile sensors at the cell using a consensus algorithm whose weights are updated to ensure improvement in the confidence level of the estimate. The mobile sensors move in a formation with a leader which moves so that the sensors are able to cover all the cells. In [47], information about a scalar field is obtained by exploring the level surfaces of the field using a mobile sensor network. In [48], a static sensor network is used along with a mobile robot to estimate a scalar field by combining the robot measurements with the sensor network measurements and planning the robot trajectory to minimize some reconstruction error.

Chapter 3

Adaptive Coverage Control of Mobile Robots

In this chapter, we will extend the adaptive coverage control algorithm discussed in section 2.2 to mobile robots. As in section 2.1, we consider N mobile robots or agents deployed over a bounded and convex region $\mathcal{Q} \subset \mathbb{R}^n$. The density function $\phi : \mathcal{Q} \to \mathbb{R}^n_+$ describes a phenomenon of interest with respect to which the robots are deployed to cover the region. The individual robot positions are denoted as p_i, $i = 1, 2, \ldots, N$. See figure 3.1.

It is assumed that the density function $\phi(\cdot)$ is unknown to the agents. We make the same assumptions on the density function $\phi(\cdot)$ as described in section 2.2, which are re-stated below. It is assumed that the density function can be written in the form

$$\phi(q) = \mathcal{K}(q)^\top a, \tag{3.1}$$

where $\mathcal{K} : \mathbb{R}^n \to \mathbb{R}^m_+$ and $a \in \mathbb{R}^m_+$ is a constant parameter vector. It is assumed that $\mathcal{K}(\cdot)$ is known to all the agents whereas a is unknown. $\mathcal{K}(q)^\top = [\mathcal{K}_1(q)\ \mathcal{K}_2(q)\ \ldots\ \mathcal{K}_m(q)]$ can be interpreted as a set of basis functions whose weighted combination gives the density function $\phi(q)$. The parameter vector a is assumed to be lower bounded and thus satisfies

$$a^i \geq a_{\min}, \qquad i = 1, 2, \ldots, m, \tag{3.2}$$

where a^i is the ith component of a. This assumption is required to make sure that the estimate of $\phi(\cdot)$ never becomes zero, and thus the control inputs are always well defined.

It is assumed that each agent can measure the value of density function $\phi(\cdot)$ at its current location. Since the true parameter a is not known, each agent will use an estimate

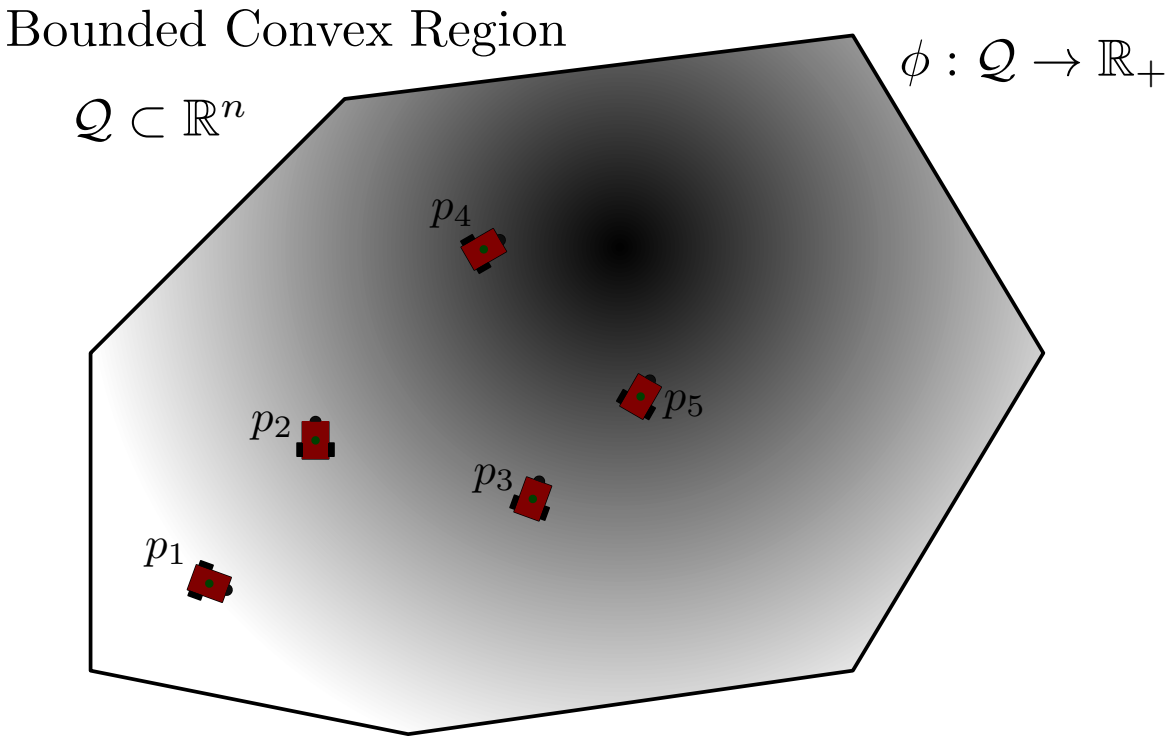


Figure 3.1: The coverage problem

of a denoted by $\hat{a}_i$. Adaptation law will be derived for $\hat{a}_i$ so that the desired objectives are achieved. We also define the corresponding estimated quantities $\hat{\phi}_i(q) = \mathcal{K}(q)^T \hat{a}_i$ which is the agent i's estimate of the density function, $\hat{M}_{\mathcal{V}_i} = \int_{\mathcal{V}_i} \hat{\phi}_i(q)dq$ which is the agent i's estimate of the mass of $\mathcal{V}_i$, $\hat{L}_{\mathcal{V}_i} = \int_{\mathcal{V}_i} q\hat{\phi}_i(q)dq$ which is the agent i's estimate of $L_{\mathcal{V}_i}$ and $\hat{C}_{\mathcal{V}_i} = \frac{\hat{L}_{\mathcal{V}_i}}{\hat{M}_{\mathcal{V}_i}}$ which is agent i's estimate of the centroid of $\mathcal{V}_i$.

First, we derive the control and adaptation law for agents with double integrator dynamics. Then we extend the algorithm to nonholonomic mobile dynamics. Next, we discuss the extension of the control and adaptation laws to the case where the robot dynamics are unknown.

3.1 Double Integrator Dynamics

We consider the following dynamics for the individual agents:

$$\ddot{p}_i = u_i, \qquad i = 1, 2, \ldots, N. \tag{3.3}$$

This can be rewritten as

$$\begin{aligned} \dot{p}_{i_1} &= p_{i_2}, \\ \dot{p}_{i_2} &= u_i, \end{aligned} \tag{3.4}$$

where $p_{i_1} := p_i$ is the position of agent i and $p_{i_2} := \dot{p}_i$ is the velocity of agent i.

Theorem 3.1 below gives an adaptive decentralized control law for agents with dynamics (3.4) to converge to centroidal Voronoi configuration under the above stated assumptions. The control law is given by

$$u_i = -k_1 \hat{M}_{\mathcal{V}_i}(p_{i_1} - \hat{C}_{\mathcal{V}_i}) - k_2 p_{i_2} \tag{3.5}$$

$$= -k_1 \hat{M}_{\mathcal{V}_i}(p_i - \hat{C}_{\mathcal{V}_i}) - k_2 \dot{p}_i. \tag{3.6}$$

The control is of P-D type with a position error term and a velocity feedback term. In addition the following quantities are defined [29],

$$\Lambda_i(t) = \int_0^t e^{-\alpha(t-\tau)} \mathcal{K}_i(\tau) \mathcal{K}_i(\tau)^T d\tau, \tag{3.7}$$

$$\lambda_i(t) = \int_0^t e^{-\alpha(t-\tau)} \mathcal{K}_i(\tau) \phi_i(\tau) d\tau, \tag{3.8}$$

$$b_i = -k_1 \int_{\mathcal{V}_i} \mathcal{K}(q)(q - p_i)^T dq \, p_{i_2} - \gamma(\Lambda_i \hat{a}_i - \lambda_i), \tag{3.9}$$

where $\mathcal{K}_i(t) := \mathcal{K}(p_{i_1}(t))$, $\phi_i(t) = \phi(p_{i_1}(t))$ which corresponds to agent i's measurement of the density function $\phi(q)$, α, and γ are positive constants. $\Lambda_i(t)$ and $\lambda_i(t)$ can be obtained using the following filter equations with zero initial conditions:

$$\dot{\Lambda}_i = -\alpha \Lambda_i(t) + \mathcal{K}_i(t) \mathcal{K}_i(t)^T, \tag{3.10}$$

$$\dot{\lambda}_i = -\alpha \lambda_i(t) + \mathcal{K}_i(t) \phi_i(t). \tag{3.11}$$

Then the adaptive law for $\hat{a}_i$ is given by

$$\dot{\hat{a}}_i = \Gamma(b_i - I_{\beta_i} b_i), \tag{3.12}$$

where I_{β_i} is a diagonal matrix with diagonal elements given by

$$I_{\beta_i}(j) = \begin{cases} 0 & \text{for } \hat{a}_i(j) > a_{\min}, \\ 0 & \text{for } \hat{a}_i(j) = a_{\min} \,\&\, b_i(j) \geq 0, \\ 1 & \text{otherwise}, \end{cases} \tag{3.13}$$

and $\Gamma > 0$ is positive definite. The adaptation law consists of the term b_i in addition to a projection defined by (3.13) to make sure that the updated parameter value is always greater than the assumed minimum $a_{\min}$.

We now state the theorem.

Theorem 3.1. *Consider N agents with dynamics given by (3.4). With the control law given by (3.6) and the update law for $\hat{a}_i$ given by (3.12), the following holds:*
(a) $\lim_{t\to\infty}(p_{i_1} - \hat{C}_{\mathcal{V}_i}) = 0$,
(b) $\lim_{t\to\infty} p_{i_2} = 0$,
(c) $\lim_{t\to\infty} \mathcal{K}_i(\tau)\tilde{a}_i(t) = 0, \quad \forall \tau \ s.t \ t - T \leq \tau \leq t$ *for any finite positive* T,
for all $i \in \{1, 2, \ldots, N\}$.

Proof. The proof essentially follows the proof of the main theorem in [29]. Define the function

$$V(t) = k_1\mathcal{H} + \frac{1}{2}\sum_{i=1}^{N} \tilde{a}_i^T\Gamma^{-1}\tilde{a}_i + \frac{1}{2}\sum_{i=1}^{N} p_{i_2}^T p_{i_2}, \tag{3.14}$$

where $\tilde{a}_i = \hat{a}_i - a$. This function is non-negative and hence, lower bounded. We next prove that its time derivative is non-positive. i.e., V is non-increasing.

$$\begin{aligned}\dot{V} &= \sum_{i=1}^{N}\left\{k_1\frac{\partial\mathcal{H}}{\partial p_{i_1}}\dot{p}_{i_1} + \tilde{a}_i^T\Gamma^{-1}\dot{\hat{a}}_i + p_{i_2}^T\dot{p}_{i_2}\right\} \\ &= \sum_{i=1}^{N}\left\{-k_1\int_{\mathcal{V}_i}(q - p_{i_1})^T\phi(q)dq\, p_{i_2} + \tilde{a}_i^T\Gamma^{-1}\dot{\hat{a}}_i + p_{i_2}^T u_i\right\}.\end{aligned}$$

Putting $\phi(q) = \mathcal{K}(q)^T a = \mathcal{K}(q)^T(\hat{a}_i - \tilde{a}_i) = \hat{\phi}(q) - \mathcal{K}(q)^T\tilde{a}_i$ and also substituting for u_i from (3.6), we get

$$\begin{aligned}\dot{V} &= \sum_{i=1}^{N}\left\{-k_1\int_{\mathcal{V}_i}(q - p_{i_1})^T\hat{\phi}(q)dq\, p_{i_2} + k_1\int_{\mathcal{V}_i}\tilde{a}_i^T\mathcal{K}(q)(q - p_{i_1})^T dq\, p_{i_2} + \tilde{a}_i^T\Gamma^{-1}\dot{\hat{a}}_i + p_{i_2}^T u_i\right\} \\ &= \sum_{i=1}^{N}\Big\{-k_1\hat{M}_{\mathcal{V}_i}(\hat{C}_{\mathcal{V}_i} - p_{i_1})^T p_{i_2} + \tilde{a}_i^T k_1\int_{\mathcal{V}_i}\mathcal{K}(q)(q - p_{i_1})^T dq p_{i_2} + \tilde{a}_i^T\Gamma^{-1}\dot{\hat{a}}_i \\ &\qquad + p_{i_2}^T\Big(-k_1\hat{M}_{\mathcal{V}_i}(p_{i_1} - \hat{C}_{\mathcal{V}_i}) - k_2 p_{i_2}\Big)\Big\}.\end{aligned}$$

The first term in the above expression cancels out leading to

$$\dot{V} = \sum_{i=1}^{N}\left\{\tilde{a}_i^T k_1\int_{\mathcal{V}_i}\mathcal{K}(q)(q - p_{i_1})^T dq p_{i_2} + \tilde{a}_i^T\Gamma^{-1}\dot{\hat{a}}_i - k_2 p_{i_2}^T p_{i_2}\right\}.$$

Substituting for $\dot{\hat{a}}_i$ with the adaptation law given by equations (3.12) and (3.13), we get

$$\dot{V} = \sum_{i=1}^{N}\left\{\tilde{a}_i^T\left[-\gamma(\Lambda_i\hat{a}_i - \lambda_i) - I_{\beta_i}b_i\right]\right\} - \sum_{i=1}^{N} k_2 p_{i_2}^T p_{i_2}$$

$$
\begin{aligned}
&= \sum_{i=1}^{N} \left\{ -\gamma \tilde{a}_i^T (\Lambda_i \hat{a}_i - \lambda_i) - \tilde{a}_i^T I_{\beta_i} b_i \right\} - \sum_{i=1}^{N} k_2 p_{i_2}^T p_{i_2} \\
&= \sum_{i=1}^{N} \left\{ -\gamma \tilde{a}_i^T \left[\int_0^t e^{-\alpha(t-\tau)} \mathcal{K}_i(\tau) \mathcal{K}_i^T(\tau) \hat{a}_i d\tau \; - \; \int_0^t e^{-\alpha(t-\tau)} \mathcal{K}_i(\tau) \phi_i(\tau) d\tau \right] - \tilde{a}_i^T I_{\beta_i} b_i \right\} \\
&\quad - \sum_{i=1}^{N} k_2 p_{i_2}^T p_{i_2} \\
&= \sum_{i=1}^{N} \left\{ -\gamma \int_0^t e^{-\alpha(t-\tau)} (\mathcal{K}_i^T(\tau) \tilde{a}_i(t))^2 d\tau - \tilde{a}_i^T I_{\beta_i} b_i \right\} - \sum_{i=1}^{N} k_2 p_{i_2}^T p_{i_2}.
\end{aligned}
$$

It is clear that the first and the last terms in the above expression are non-positive. We can also show that the second term is non-positive as follows: From equation 3.13, it is clear that when $\hat{a}_i(j) > a_{\min}$ or when $\hat{a}_i(j) = a_{\min}$ and $b_i(j) \geq 0$, the term $\tilde{a}_i^T I_{\beta_i} b_i$ vanishes. When $\hat{a}_i(j) = a_{\min}$ and $b_i(j) < 0$, we have that $\tilde{a}_i(j) \leq 0$ by assumption (2.11). This means that the term $\tilde{a}_i^T I_{\beta_i} b_i \geq 0$, making the second term in the above $\dot{V}$ expression non-positive. Thus we have shown that $\dot{V}$ is non-positive or non-increasing.

Now, since V is positive (and thus is bounded below by zero) and its time derivative $\dot{V}$ is non-positive, it follows that $\lim_{t\to\infty} V(t) < \infty$ is finite. This implies that $\dot{V}$ is integrable and $\lim_{t\to\infty} \int_0^t \dot{V} dt < \infty$ is finite. This together with the fact that p_{i_2}, $\dot{p}_{i_2}$ are bounded allows us to conclude that

$$\lim_{t\to\infty} p_{i_2} = 0, \tag{3.15}$$

using Barbalat's lemma. In a similar manner, we can show that

$$\lim_{t\to\infty} \int_0^t e^{-\alpha(t-\tau)} (\mathcal{K}_i^T(\tau) \tilde{a}_i(t))^2 d\tau = 0. \tag{3.16}$$

Equation (3.15) along with the fact that $\dot{p}_{i_2}$ is uniformly continuous (this is because each term of $\ddot{p}_{i_2} = \dot{u}_i$ is bounded, see Lemma 1 in the appendix of [30]) implies that $\lim_{t\to\infty} \dot{p}_{i_2} = 0$ using Barbalat's lemma. This along with the fact that $\dot{p}_{i_2} = u_i = -k_1 \hat{M}_{\mathcal{V}_i} (p_{i_1} - \hat{C_{\mathcal{V}_i}}) - k_2 p_{i_2}$ allows us to conclude that

$$\lim_{t\to\infty} (p_{i_1} - \hat{C}_{\mathcal{V}_i}) = 0. \tag{3.17}$$

Now consider equation (3.16). The integrand is always non-negative which implies that the integral can be zero only if the integrand converges to zero. This, in turn, implies that

$$\lim_{t\to\infty} e^{-\alpha(t-\tau)} (\mathcal{K}_i(\tau) \tilde{a}_i(t))^2 = 0, \quad \forall \tau \text{ s.t } 0 \leq \tau \leq t, \tag{3.18}$$

for all $i \in \{1, 2, \ldots, N\}$. Now, given any finite $T > 0$, $e^{-\alpha(t-\tau)} > 0$ for $t - T < \tau < t$. Then for equation (3.18) to hold, we require

$$\lim_{t\to\infty} (\mathcal{K}_i(\tau)\tilde{a}_i(t)) = 0, \quad \forall \tau \text{ s.t } t - T \leq \tau \leq t. \tag{3.19}$$

Thus equations (3.15), (3.17), (3.19) hold, and the theorem is proved. □

Remark 3.2. Equation (3.19) implies

$$\lim_{t\to\infty} \mathcal{K}_i(\tau)^T \hat{a}_i(t) = \mathcal{K}_i(\tau)^T a,$$

$\forall \tau$ s.t $t - T \leq \tau \leq t$ and $\forall i \in \{1, 2, \ldots, n\}$ where a is the true parameter, i.e., the estimated density function $\hat{\phi}$ converges to the true value for points on the robot trajectory and not necessarily for all points in $\mathcal{Q}$. Since the agents converge only towards the estimated centroids, we will call such a configuration *near-optimal configuration*. See also [29, 30]. In [30], it is shown that if $\lim_{t\to\infty} \Lambda_i(t)$ is positive definite, then $\lim_{t\to\infty} \tilde{a}_i(t) = 0$ and thus, the estimated centroid $\hat{C}_{\mathcal{V}_i}$ converges to the true centroid $C_{\mathcal{V}_i}$ which corresponds to the *optimal coverage configuration*. This condition is similar to (but weaker than) the *persistency of excitation* condition. We will discuss parameter convergence in more detail in chapter 6.

3.2 Extension to dynamics of nonholonomic robots

In the literature, control of mobile robots is often accomplished using the kinematic model (unicycle model, for example). In this section, we will extend the result for coverage control to nonholonomic agents using dynamic models. In particular we will consider mobile robots of differential drive type. The general model for mobile robots obtained using the Euler-Lagrange equation is (see [49])

$$M_i(q_i)\ddot{q}_i + V_{mi}(q, \dot{q}_i)\dot{q}_i + F_i(\dot{q}_i) + G(q_i) = B(q_i)\tau_i - A_i^T(q_i)\eta_i, \tag{3.20}$$

where the subscript i denotes the ith robot, $q_i \in \mathbb{R}^q$ gives the generalized coordinates, $M_i(q_i) \in \mathbb{R}^{q\times q}$ is inertia matrix (symmetric, positive definite), $V_m(q_i, \dot{q}_i) \in \mathbb{R}^{q\times q}$ is the centripetal and Coriolis matrix, $F_i(\dot{q}_i) \in \mathbb{R}^q$ is the vector representing the surface friction, $G_i(q_i) \in \mathbb{R}^q$ is the vector representing the gravitational force, $B_i(q_i) \in \mathbb{R}^{q\times l}$ is the input transformation matrix, $\tau_i \in \mathbb{R}^l$ is the input vector, $A_i(q_i) \in \mathbb{R}^{c\times q}$ is the constraint matrix

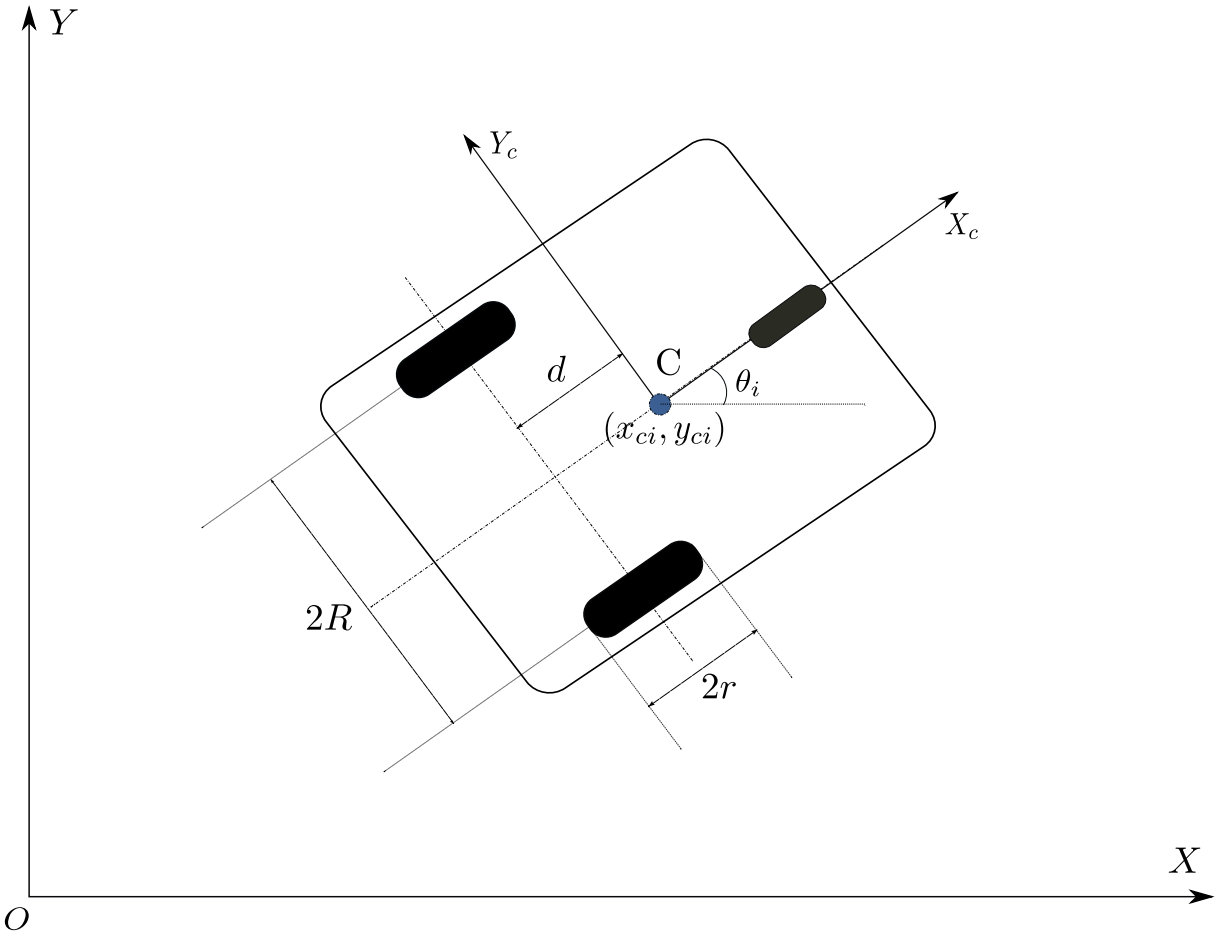


Figure 3.2: Mobile robot

and $\eta_i \in \mathbb{R}^c$ is the vector of constraint forces. The non-holonomic constraints are given by

$$A_i(q_i)\dot{q}_i = 0. \tag{3.21}$$

We also define a matrix $S_i(q_i) \in \mathbb{R}^{q\times(q-c)}$ whose columns span the null space of $A_i(q_i)$,

$$S_i^T(q_i)A_i^T(q_i) = 0. \tag{3.22}$$

Now we consider the dynamic model for a nonholonomic planar mobile robot with two actuated wheels, which also integrates the kinematic model as presented in [49]. We will assume that the robots are identical, with mass m, radius of the wheels equal to r, length of the axle between the two wheels equal to $2R$ and the distance between the center of mass and the axle equal to d. See figure 3.2. In this case $q_i = [x_{ci}\ y_{ci}\ \theta_i]^T$ where (x_{ci}, y_{ci}) gives the coordinates of the center of mass of the ith robot and θ_i gives the orientation of the ith robot. The nonholonomic constraints are given by (assuming no-slip condition)

$$\dot{y}_{ci}\cos(\theta_i) - \dot{x}_{ci}\sin(\theta_i) - d\dot{\theta}_i = 0, \tag{3.23}$$

which gives

$$A_i(q_i) = [\,-\sin(\theta_i)\ \cos(\theta_i)\ \ -d\,],$$

and $S_i(q_i)$ is given by

$$S_i(q_i) = \begin{pmatrix} \cos(\theta_i) & -d\sin(\theta_i) \\ \sin(\theta_i) & d\cos(\theta_i) \\ 0 & 1 \end{pmatrix}. \tag{3.24}$$

We get the kinematic equations as

$$q_i = S_i(q_i)v_i, \tag{3.25}$$

where $v_i := [u_i\ \omega_i]^T$. Here u_i is the linear velocity of the center of mass and ω_i is the angular velocity. The other dynamical quantities are given by

$$M(q) = \begin{pmatrix} m & 0 & md\sin(\theta_i) \\ 0 & m & -md\cos(\theta_i) \\ md\sin(\theta_i) & -md\cos(\theta_i) & I \end{pmatrix},$$

$$V_i(q_i, \dot{q}_i) = \begin{pmatrix} md\dot{\theta}_i^2\cos(\theta_i) \\ md\dot{\theta}_i^2\sin(\theta_i) \\ 0 \end{pmatrix}, \quad G_i(q_i) = 0,$$

$$B_i(q_i) = \frac{1}{r}\begin{pmatrix} \cos(\theta_i) & \cos(\theta_i) \\ \sin(\theta_i) & \sin(\theta_i) \\ R & -R \end{pmatrix}, \quad \tau_i = \begin{pmatrix} \tau_{ir} \\ \tau_{il} \end{pmatrix},$$

where τ_{ir} and τ_{il} are the torque inputs to the right and left wheels of the ith robot, respectively. We can convert the above dynamical model given by (3.20) into the following form, as presented in [49].

$$\dot{q}_i = S_i(q_i)v_i, \tag{3.26}$$

$$\bar{M}_i(q_i)\dot{v}_i = \bar{B}_i\tau_i - \bar{V}_{mi}(q_i, \dot{q}_i)v_i - \bar{F}(v_i), \tag{3.27}$$

where $\bar{M}_i = S_i^T M_i S_i$ is a symmetric and positive definite inertia matrix, $\bar{V}_{mi} = S_i^T(M_i\dot{S}_i + V_{mi}S_i)$ is the centripetal and Coriolis matrix, $\bar{F}_i(v_i)$ is the surface friction term, $\bar{B}_i = S_i^T B_i$ is a constant non-singular matrix. We will use the model given by equations (3.26) and (3.27) in what follows. Also, we will assume that all the parameters and other quantities in the model equations are fully known for deriving the coverage control.

It can be shown that the matrix $\dot{\bar{M}}_i - 2\bar{V}_{mi}$ is skew symmetric (see [49]), i.e.,

$$\frac{1}{2}\zeta^T(\dot{\bar{M}}_i - 2\bar{V}_{mi})\zeta = 0, \quad \forall\, \zeta \in \mathbb{R}^2, \text{ and all } i \in \{1, 2, \ldots, N\}. \tag{3.28}$$

We also note that $q_i = [x_{ci}\ y_{ci}\ \theta_i]^T = [p_i^T\ \theta_i]^T$ where $p_i = [x_{ci}\ y_{ci}]^T$ is the location of the ith robot. Correspondingly, we divide the matrix $S_i(q_i)$ as

$$\begin{bmatrix} \dot{p}_i \\ \dot{\theta}_i \end{bmatrix} = \begin{bmatrix} S_i^1(q_i) \\ S_i^2(q_i) \end{bmatrix} v_i. \tag{3.29}$$

We now have the following theorem.

Theorem 3.3. *Consider N agents with dynamics given by (3.26) and (3.27). With the control law for the ith given by*

$$\tau_i = \bar{B}_i^{-1}\left\{-k_1\hat{M}_{\mathcal{V}_i}(S_i^1)^T(p_i - \hat{C}_{\mathcal{V}_i}) - k_2 v_i + \bar{F}_i\right\}, \tag{3.30}$$

where k_1, k_2 are positive scalar gains, and the adaptation law for $\hat{a}_i$ given by

$$\dot{\hat{a}}_i = \Gamma(b_i - I_{\beta_i} b_i), \tag{3.31}$$

with

$$b_i = -k_1 \int_{\mathcal{V}_i} \mathcal{K}(q)(q - p_i)^T dq\, S_i^1 v_i - \gamma(\Lambda_i \hat{a}_i - \lambda_i), \tag{3.32}$$

$\Gamma > 0$ a positive definite gain matrix, Λ_i given by equation (3.7), λ_i given by equation (3.8) and I_{β_i} given by equation (3.13), the following holds:
(a) $\lim_{t\to\infty}(p_i - \hat{C}_{\mathcal{V}_i}) = 0$,
(b) $\lim_{t\to\infty} v_i = 0$,
(c) $\lim_{t\to\infty} \mathcal{K}_i(\tau)\tilde{a}_i(t) = 0, \quad \forall \tau\ s.t\ t - T \le \tau \le t$ *for any finite positive T,*
for all $i \in \{1, 2, \ldots, N\}$.

Proof. The proof proceeds in the same manner as in the proof of Theorem 3.1 by considering

$$V(t) = k_1\mathcal{H} + \frac{1}{2}\sum_{i=1}^{N} \tilde{a}_i^T \Gamma^{-1}\tilde{a}_i + \frac{1}{2}\sum_{i=1}^{N} v_i^T \bar{M}_i v_i. \tag{3.33}$$

Taking the derivative,

$$\dot{V} = \sum_{i=1}^{N}\left\{k_1\frac{\partial\mathcal{H}}{\partial p_i}\dot{p}_i + \tilde{a}_i^T\Gamma^{-1}\dot{\hat{a}}_i + v_i^T\bar{M}_i\dot{v}_i + \frac{1}{2}v_i^T\dot{\bar{M}}_i v_i\right\}$$

$$= \sum_{i=1}^{N} \left\{ -k_1 \int_{\mathcal{V}_i} (q - p_i)^T \phi(q) dq \, S_i^1 v_i + \tilde{a}_i^T \Gamma^{-1} \dot{\hat{a}}_i + v_i^T (\bar{B}_i \tau_i - \bar{V}_{mi} v_i - \bar{F}_i) + \frac{1}{2} v_i^T \dot{\bar{M}}_i v_i \right\}$$

$$= \sum_{i=1}^{N} \left\{ -k_1 \int_{\mathcal{V}_i} (q - p_i)^T \phi(q) dq \, S_i^1 v_i + \tilde{a}_i^T \Gamma^{-1} \dot{\hat{a}}_i + p_{i_2}^T (\bar{B}_i \tau_i - \bar{F}_i) + \frac{1}{2} v_i^T (\dot{\bar{M}}_i - 2\bar{V}_{mi}) v_i \right\}.$$

Using the skew-symmetric property (3.28) and the equation for control τ_i given by (3.30), we get

$$\dot{V} = \sum_{i=1}^{N} \left\{ -k_1 \int_{\mathcal{V}_i} (q - p_i)^T \phi(q) dq \, S_i^1 v_i + \tilde{a}_i^T \Gamma^{-1} \dot{\hat{a}}_i + v_i^T (-k_1 \hat{M}_{\mathcal{V}_i} (S_i^1)^T (p_i - \hat{C}_{\mathcal{V}_i})) - k_2 v_i^T v_i \right\}$$

$$= \sum_{i=1}^{N} \left\{ -k_1 \hat{M}_{\mathcal{V}_i} (\hat{C}_{\mathcal{V}_i} - p_i)^T S_i^1 v_i + \tilde{a}_i^T k_1 \int_{\mathcal{V}_i} \mathcal{K}(q) (q - p_i)^T dq S_i^1 v_i + \tilde{a}_i^T \Gamma^{-1} \dot{\hat{a}}_i \right\}$$
$$+ \sum_{i=1}^{N} \left\{ v_i^T \left(-k_1 \hat{M}_{\mathcal{V}_i} (S_i^1)^T (p_i - \hat{C}_{\mathcal{V}_i}) \right) - k_2 v_i^T v_i \right\}.$$

Substituting the adaptation law (3.31) and simplifying, we get

$$\dot{V} = \sum_{i=1}^{n} \left\{ -\gamma \int_0^t e^{-\alpha(t-\tau)} (\mathcal{K}_i^T(\tau) \tilde{a}_i(t))^2 d\tau - \tilde{a}_i^T I_{\beta_i} b_i \right\} - \sum_{i=1}^{n} k_2 v_i^T v_i.$$

As in the proof of Theorem 3.1, it can be shown that all the three terms in the above expression are positive, which means that $\dot{V}$ is non-increasing. Since V is non-negative (bounded below by zero) and its time derivative $\dot{V}$ is non-positive, it follows that $\lim_{t\to\infty} V(t) < \infty$ (is finite). This implies that $\dot{V}$ is integrable and $\lim_{t\to\infty} \int_0^t \dot{V} dt < \infty$ (is finite). This together with the fact that v_i, $\dot{v}_i$ are bounded allows us to conclude that

$$\lim_{t\to\infty} v_i = 0, \tag{3.34}$$

using Barbalat's lemma. In a similar manner, we can show that

$$\lim_{t\to\infty} \int_0^t e^{-\alpha(t-\tau)} (\mathcal{K}_i^\top(\tau) \tilde{a}_i(t))^2 d\tau = 0. \tag{3.35}$$

Equation (3.34) along with the fact that $\dot{v}_i$ is uniformly continuous (this is because each term of $\ddot{v}_i = \frac{d}{dt} \left[\bar{M}_i^{-1} \left(-k_1 \hat{M}_{\mathcal{V}_i} (S_i^1)^\top (p_i - \hat{C}_{\mathcal{V}_i}) - k_2 v_i - \bar{V}_{mi}(q_i, \dot{q}_i) v_i \right) \right]$ is bounded; also see Lemma 1 and 2 in the appendix of [30]) implies that $\lim_{t\to\infty} \dot{v}_i = 0$ using Barbalat's lemma. This along with the fact that the closed loop dynamics are given by $\bar{M}_i \dot{v}_i = -k_1 \hat{M}_{\mathcal{V}_i} (S_i^1)^\top (p_i - \hat{C_{\mathcal{V}_i}}) - k_2 v_i - \bar{V}_{mi}(q_i, \dot{q}_i) v_i$ allows us to conclude that

$$\lim_{t\to\infty} (p_{i_1} - \hat{C}_{\mathcal{V}_i}) = 0 \tag{3.36}$$

since S_i^1 is non-singular. Now consider equation (3.35). Similar to the proof of Theorem 3.1, we can conclude from equation (3.35) that

$$\lim_{t\to\infty}(\mathcal{K}_i(\tau)\tilde{a}_i(t)) = 0, \quad \forall\tau \text{ s.t } t - T \leq \tau \leq t, \tag{3.37}$$

given any finite $T > 0$. Thus equations (3.34), (3.36), (3.37) hold and the statements (a), (b) and (c) of the theorem are proved. □

Remark 3.4 (Regarding orientations of the robots). Theorem (3.3) states that utilizing the control law and adaptation law given by (3.30) and (3.31), the robot positions converge to the estimated centroids. Regarding the orientation of the robots, we can only conclude from the above theorem that $\dot{\theta}_i$ is bounded, and $\lim_{t\to\infty}\dot{\theta}_i = 0$. This is the case since $\lim_{t\to\infty} v_i = 0$ implies $\lim_{t\to\infty}\omega_i = 0$ which in turn implies that $\lim_{t\to\infty}\dot{\theta}_i = 0$.

3.3 Adaptive Coverage Control with Model Uncertainties

In practical scenarios, the parameters in the robot dynamics such as friction coefficients or inertia parameters may not be well known. In this section, we extend the adaptive coverage control algorithm for the case where the robot dynamics contain unknown parameters. We use an additional adaptation law to account for the unknown parameters in the design of the control law.

We assume that the kinematics of the robots are fully known i.e., the entries of the matrix $S(q)$ in equation (3.26) is known. The dynamics equation given by (3.27) contains unknown/uncertain parameters. We also make the following assumption concerning the unknown parameters:

Assumption 3.5.

$$\bar{M}_i(q_i)\dot{z} + \bar{V}_{mi}(q_i,\dot{q}_i)z + \bar{F}(v) := W_i(q,\dot{q},v,z,\dot{z})^T\eta, \qquad \forall z \in \mathbb{R}^2, \tag{3.38}$$

where $\eta \in \mathbb{R}^d$ is a constant vector.

We cannot use the Lyapunov function from Theorem 3.3 in section 3.2 for deriving the control since it will lead to a detectability obstacle and the desired stability cannot be established. The problem is that v_i tending to zero (see equation (3.34)), does not imply

from the closed loop dynamics that the position error $p_i - \hat{C}_{\mathcal{V}_i}$ goes to zero, since the parameters are not exactly known and thus the control input cannot exactly cancel the terms containing the parameters. The control law, in this case, can be designed using the backstepping approach. Considering the kinematic equation (3.26) with v_i as the control and the Lyapunov function

$$V_1 = k_1 \mathcal{H} + \sum_{i=1}^{N} \frac{1}{2} \tilde{a}_i^T \Gamma_a^{-1} \tilde{a}_i. \tag{3.39}$$

Its derivative is given by

$$\dot{V}_1 = k_1 \sum_{i=1}^{N} \left\{ \hat{M}_{\mathcal{V}_i} (p_i - \hat{C}_{\mathcal{V}_i})^T \dot{p}_i + \tilde{a}_i^T \int_{\mathcal{V}_i} \mathcal{K}(q)(q - p_i)^T dq \, \dot{p}_i \right\} + \sum_{i=1}^{N} \tilde{a}_i^T \Gamma_a^{-1} \dot{\hat{a}}_i. \tag{3.40}$$

Using the adaptation law for $\hat{a}_i$ given by equation (3.31) and the control law for v_i as

$$v_i = v_{id} := -(S^1(q))^T (p_i - \hat{C}_{\mathcal{V}_i}), \tag{3.41}$$

we get

$$\dot{V}_1 = -k_1 \hat{M}_{\mathcal{V}_i} \sum_{i=1}^{N} \| (S^1(q))^T (p_i - \hat{C}_{\mathcal{V}_i}) \|^2 - \sum_{i=1}^{N} \left\{ \gamma \int_0^t e^{-\alpha(t-\tau)} (\mathcal{K}_i^T(\tau) \tilde{a}_i(t))^2 d\tau + \tilde{a}_i^T I_{\beta_i} b_i \right\}. \tag{3.42}$$

The fact that V_1 is lower bounded and $\dot{V}_1$ is non-positive implies that $\lim_{t\to\infty} V_1(t)$ exists. This, in turn, implies that $\dot{V}_1$ is integrable. This along with the fact that the first term of $\dot{V}_1$ is uniformly continuous and $S^1(q)$ is non-singular allows us to conclude that $\lim_{t\to\infty}(p_i - \hat{C}_{\mathcal{V}_i}) = 0$. Also, the statement (c) of Theorem 3.3 holds in this case.

Now we consider the full dynamical model given by equations (3.26) and (3.27). Using assumption 3.5, we define

$$\bar{M}_i(q_i)\dot{v}_{id} + \bar{V}_{mi}(q_i, \dot{q}_i) v_{id} + \bar{F}(v_i) = W_i(q, \dot{q}, v_i, v_{id}, \dot{v}_{id})^T \eta := W_i^T \eta_i, \tag{3.43}$$

where $\eta_i \in \mathbb{R}^d$ is an unknown constant parameter. We will use $\hat{\eta}_i$ to denote the estimate of η_i computed by agent i. We now have the following theorem:

Theorem 3.6. *Consider N agents with dynamics given by* (3.26) *and (3.27). With the control law for the ith agent given by*

$$\tau_i = \bar{B}_i^{-1} \left\{ W_i^T \hat{\eta}_i - k_1 \hat{M}_{\mathcal{V}_i} (S^1(q))^T (p_i - \hat{C}_{\mathcal{V}_i}) - k_2 (v_i - v_{id}) \right\}, \tag{3.44}$$

with v_{id} given by (3.41), the adaptation law for $\hat{a}_i$ given by

$$\dot{\hat{a}}_i = \Gamma_a(b_i - I_{\beta_i} b_i), \tag{3.45}$$

the adaptation law for $\hat{\eta}_i$ given by

$$\dot{\hat{\eta}}_i = \Gamma_\eta W_i(v_i - v_{id}), \tag{3.46}$$

b_i and I_{β_i} given by equations (3.9) and (3.13), Γ_a and Γ_η are positive definite gain matrices, Λ_i and λ_i given by equations (3.7) and (3.8), the following holds:
(a) $\lim_{t\to\infty}(p_i - \hat{C}_{\mathcal{V}_i}) = 0$,
(b) $\lim_{t\to\infty} v_i = 0$,
(c) $\lim_{t\to\infty} \mathcal{K}_i(\tau)\tilde{a}_i(t) = 0, \quad \forall \tau \ s.t \ t - T \leq \tau \leq t$ *and any finite positive* T,
for all $i \in \{1, 2, \ldots, N\}$.

Proof. Consider the Lyapunov function

$$V_2 = V_1 + \frac{1}{2}\sum_{i=1}^{N} \tilde{v}_i^\top \bar{M}\tilde{v}_i + \frac{1}{2}\sum_{i=1}^{N} \tilde{\eta}_i^\top \Gamma_\eta^{-1}\tilde{\eta}_i, \tag{3.47}$$

where $\tilde{\eta}_i = \hat{\eta}_i - \eta_i$, $\tilde{v}_i = v_i - v_{id}$.
Computing the derivative of V_2, we have

$$\begin{aligned}
\dot{V}_2 &= \dot{V}_1 + \sum_{i=1}^{N}\left[\tilde{v}_i^\top \bar{M}\dot{\tilde{v}}_i + \frac{1}{2}\tilde{v}_i^\top \dot{\bar{M}}\tilde{v}_i\right] + \sum_{i=1}^{N} \tilde{\eta}_i^\top \Gamma_\eta^{-1}\dot{\hat{\eta}}_i \\
&= \sum_{i=1}^{N}\Big\{k_1 \hat{M}_{\mathcal{V}_i}(p_i - \hat{C}_{\mathcal{V}_i})^\top S^1(q)v_i + \tilde{v}_i^\top\left[\bar{B}_i\tau_i - \bar{V}_{mi}v_i - F - \bar{M}\dot{v}_{id}\right] \\
&\qquad + \frac{1}{2}\tilde{v}_i^\top \dot{\bar{M}}\tilde{v}_i + \tilde{\eta}_i^\top \Gamma_\eta^{-1}\dot{\hat{\eta}}_i - \gamma\int_0^t e^{-\alpha(t-\tau)}(\mathcal{K}_i^\top(\tau)\tilde{a}_i(t))^2 d\tau - \tilde{a}_i^\top I_{\beta_i} b_i\Big\}.
\end{aligned}$$

Rearranging terms and using equation (3.43),

$$\begin{aligned}
\dot{V}_2 = k_1\sum_{i=1}^{N} \hat{M}_{\mathcal{V}_i}(p_i - \hat{C}_{\mathcal{V}_i})^\top S^1(q)v_i + \sum_{i=1}^{N}\left\{\tilde{v}_i^\top\left[\bar{B}_i\tau_i - W_i^T\eta_i\right] + \frac{1}{2}\tilde{v}_i^\top(\frac{1}{2}\dot{\bar{M}} - \bar{V}_{mi})\tilde{v}_i\right\} \\
+ \sum_{i=1}^{N} \tilde{\eta}_i^\top \Gamma_\eta^{-1}\dot{\hat{\eta}}_i - \sum_{i=1}^{N}\left\{\gamma\int_0^t e^{-\alpha(t-\tau)}(\mathcal{K}_i^\top(\tau)\tilde{a}_i(t))^2 d\tau + \tilde{a}_i^\top I_{\beta_i} b_i\right\}.
\end{aligned}$$

Using the skew-symmetry property (3.28), control law (3.44) and the adaptation law (3.46) for $\hat{\eta}_i$, we get

$$\dot{V}_2 = k_1\sum_{i=1}^{N} \hat{M}_{\mathcal{V}_i} v_{id}^\top [S^1(q)]^\top (p_i - \hat{C}_{\mathcal{V}_i}) - k_2\sum_{i=1}^{N} \tilde{v}_i^\top \tilde{v}_i$$

$$
\begin{aligned}
&\qquad\qquad - \sum_{i=1}^{N} \left\{ \gamma \int_0^t e^{-\alpha(t-\tau)} (\mathcal{K}_i^\top(\tau)\tilde{a}_i(t))^2 d\tau + \tilde{a}_i^\top I_{\beta_i} b_i \right\} \\
&= -k_1 \sum_{i=1}^{N} \hat{M}_{\mathcal{V}_i} \|(S^1(q))^\top (p_i - \hat{C}_{\mathcal{V}_i})\|^2 - k_2 \sum_{i=1}^{N} \|\tilde{v}_i\|^2 \\
&\qquad\qquad - \sum_{i=1}^{N} \left\{ \gamma \int_0^t e^{-\alpha(t-\tau)} (\mathcal{K}_i^\top(\tau)\tilde{a}_i(t))^2 d\tau + \tilde{a}_i^\top I_{\beta_i} b_i \right\}.
\end{aligned}
$$

V_2 is non-negative and $\dot{V}_2$ is non-increasing. Using a similar analysis to that in the proof of Theorem 3.3, it can be shown that for each $i \in \{1, 2, \ldots, N\}$,

$$\lim_{t\to\infty} \|(S^1(q))^\top (p_i - \hat{C}_{\mathcal{V}_i})\| = 0, \tag{3.48}$$

$$\lim_{t\to\infty} \|v_i - v_{id}\| = 0, \tag{3.49}$$

$$\lim_{t\to\infty} \mathcal{K}_i(\tau)\tilde{a}_i(t) = 0, \quad \forall \tau \text{ s.t } t - T \le \tau \le t, \quad \text{where } T > 0. \tag{3.50}$$

Statement (a) of the theorem follows from equation (3.48) and the fact that $S^1(q)$ is non-singular. Equations (3.48) and (3.49) imply statement (b) of the theorem. Statement (c) of the theorem follows from equation (3.50). □

Remark 3.7. Regarding the orientation of the agents, as in the case of Theorem 3.3, we have that $\dot{\theta}_i$ is bounded and $\lim_{t\to\infty} \dot{\theta}_i = 0$ for all $i \in \{1, 2, \ldots, N\}$.

Remark 3.8. Computing the adaptive control law (3.44) requires computing $\dot{v}_{id}$ which in turn requires the knowledge of $\dot{\hat{C}}_{\mathcal{V}_i}$. Computation of $\dot{\hat{C}}_{\mathcal{V}_i}$ requires computing the time derivative of spatial integrals whose area of integration depends on time and is thus a non-trivial task. We look at the computation of $\dot{\hat{C}}_{\mathcal{V}_i}$ for planar robots below.

Computing $\dot{\hat{C}}_{\mathcal{V}_i}$

We have

$$\hat{C}_{\mathcal{V}_i} = \frac{\hat{L}_{\mathcal{V}_i}}{\hat{M}_{\mathcal{V}_i}}, \text{ where } \hat{L}_{\mathcal{V}_i} = \int_{\mathcal{V}_i} q\hat{\phi}(q)dq, \; \hat{M}_{\mathcal{V}_i} = \int_{\mathcal{V}_i} \hat{\phi}(q)dq. \tag{3.51}$$

Then,

$$\dot{\hat{C}}_{\mathcal{V}_i} = \frac{\hat{M}_{\mathcal{V}_i}\dot{\hat{L}}_{\mathcal{V}_i} - \hat{L}_{\mathcal{V}_i}\dot{\hat{M}}_{\mathcal{V}_i}}{\hat{M}_{\mathcal{V}_i}^2}. \tag{3.52}$$

To compute $\dot{\hat{C}}_{\mathcal{V}_i}$, we thus need to compute the time derivatives $\dot{\hat{L}}_{\mathcal{V}_i}$ and $\dot{\hat{M}}_{\mathcal{V}_i}$. Both $\hat{L}_{\mathcal{V}_i}$ and $\hat{M}_{\mathcal{V}_i}$ are of the form

$$I_i := \int_{\mathcal{V}_i} f(q,t)dq, \quad \text{with } f(q,t) = \begin{cases} \phi(q) & \text{for } \hat{M}_{\mathcal{V}_i}, \\ q\,\phi(q) & \text{for } \hat{L}_{\mathcal{V}_i}, \end{cases}$$

where the region of integration $\mathcal{V}_i$ is a function of agent positions p_i's. The derivatives can be computed as

$$\dot{\hat{L}}_{\mathcal{V}_i} = \frac{\partial \hat{L}_{\mathcal{V}_i}}{\partial t} + \sum_{j=1}^{n} \frac{\partial \hat{L}_{\mathcal{V}_i}}{\partial p_j}\dot{p}_j, \qquad \dot{\hat{M}}_{\mathcal{V}_i} = \frac{\partial \hat{M}_{\mathcal{V}_i}}{\partial t} + \sum_{j=1}^{n} \frac{\partial \hat{M}_{\mathcal{V}_i}}{\partial p_j}\dot{p}_j, \tag{3.53}$$

where

$$\frac{\partial \hat{L}_{\mathcal{V}_i}}{\partial t} = \int_{\mathcal{V}_i} \mathcal{K}(q)^T \dot{\hat{a}}_i dq, \qquad \frac{\partial \hat{M}_{\mathcal{V}_i}}{\partial t} = \int_{\mathcal{V}_i} q\mathcal{K}(q)^T \dot{\hat{a}}_i dq. \tag{3.54}$$

It remains to compute $\frac{\partial \hat{L}_{\mathcal{V}_i}}{\partial p_j}$ and $\frac{\partial \hat{L}_{\mathcal{V}_i}}{\partial p_j}$. Let $\mathcal{N}_i$ denote the set of Voronoi neighbours of agent i. For $j \notin \mathcal{N}_i \cup \{i\}$,

$$\frac{\partial \hat{L}_{\mathcal{V}_i}}{\partial p_j} = \frac{\partial \hat{M}_{\mathcal{V}_i}}{\partial p_j} = 0 \tag{3.55}$$

since the Voronoi region $\mathcal{V}_i$ does not depend on p_j. For $j \in \mathcal{N}_i \cup \{i\}$,

$$\frac{\partial I_i}{\partial p_j} = \begin{bmatrix} \frac{\partial I_i}{\partial p_{j_x}} & \frac{\partial I_i}{\partial p_{j_y}} \end{bmatrix},$$

where (see [50])

$$\frac{\partial I_i}{\partial p_{j_x}} = \int_{\partial \mathcal{V}_i} f(q,t)\, n_q^T \frac{\partial q}{\partial p_{j_x}} dq,$$
$$\frac{\partial I_i}{\partial p_{j_y}} = \int_{\partial \mathcal{V}_i} f(q,t)\, n_q^T \frac{\partial q}{\partial p_{j_y}} dq,$$

where $\partial\mathcal{V}_i$ is the boundary of $\mathcal{V}_i$ and n_q is the normal to $\partial\mathcal{V}_i$ at the point q. This can be further simplified as (see figure 3.3)

$$\frac{\partial I_i}{\partial p_{j_x}} = \int_{c_{ij}} f(q,t)\, n_q^T \frac{\partial q}{\partial p_{j_x}} dq \quad (j \neq i), \qquad \frac{\partial I_i}{\partial p_{i_x}} = \sum_{j \in \mathcal{N}_i} \int_{c_{ij}} f(q,t)\, n_q^T \frac{\partial q}{\partial p_{i_x}} dq, \tag{3.56}$$

$$\frac{\partial I_i}{\partial p_{j_y}} = \int_{c_{ij}} f(q,t)\, n_q^T \frac{\partial q}{\partial p_{j_y}} dq \quad (j \neq i), \qquad \frac{\partial I_i}{\partial p_{i_y}} = \sum_{j \in \mathcal{N}_i} \int_{c_{ij}} f(q,t)\, n_q^T \frac{\partial q}{\partial p_{i_y}} dq, \tag{3.57}$$

where c_{ij} is the segment of $\partial\mathcal{V}_i$ that is shared between agents i and j. The terms $n_q^T \frac{\partial q}{\partial p_{j_x}}$ and $n_q^T \frac{\partial q}{\partial p_{j_y}}$ can be computed to be (see Lemma 2.2 in [51])

$$n_q^T \frac{\partial q}{\partial p_{j_x}} = \frac{\cos(\theta_q)}{2} + \sin(\theta_q)\frac{d_{qc}}{d_{ij}}, \qquad n_q^T \frac{\partial q}{\partial p_{i_x}} = \frac{\cos(\theta_q)}{2} - \sin(\theta_q)\frac{d_{qc}}{d_{ij}}, \tag{3.58}$$

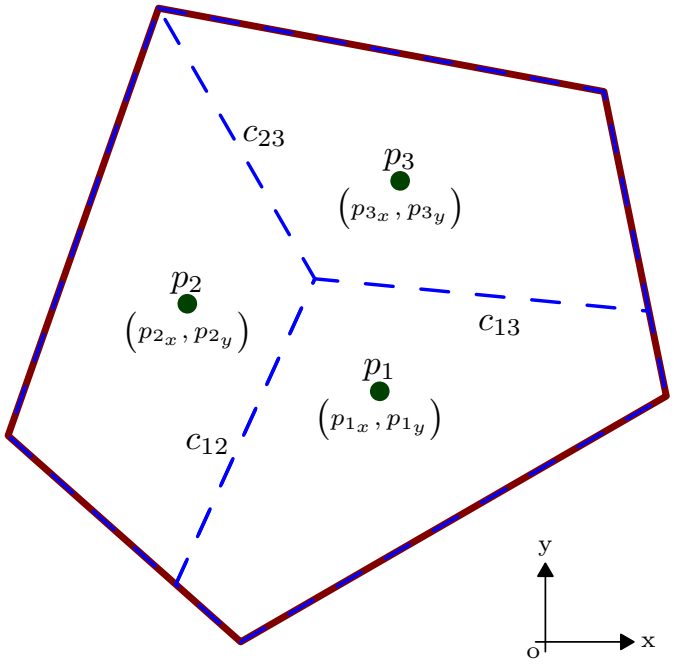


Figure 3.3: A Voronoi diagram with 3 agents

$$n_q^T \frac{\partial q}{\partial p_{j_y}} = \frac{\sin(\theta_q)}{2} + \cos(\theta_q)\frac{d_{qc}}{d_{ij}}, \qquad n_q^T \frac{\partial q}{\partial p_{i_y}} = \frac{\sin(\theta_q)}{2} - \cos(\theta_q)\frac{d_{qc}}{d_{ij}}, \tag{3.59}$$

where θ_q is the angle (in radians) between the x-axis and n_q, $d_{ij} = \|p_i - p_j\|$ is the distance between agents i and j, d_{qc} is the distance between point q of $\partial\mathcal{V}_i$ and the point $c = \frac{(p_i+p_j)}{2}$.

3.4 Note on failure of agents

In this section, we briefly look at the effects of agents failing or becoming faulty. Two modes of failure which may happen frequently are:

1. Communication breakdown: The communication system of individual agents may fail, and the agents will not be able to communicate with their Voronoi neighbours: in this case, the faulty agent will not be able to compute its Voronoi region since it requires communication between Voronoi neighbours. If q agents fail in this way, the remaining $n - q$ agents may perform the coverage task.

2. Mobility breakdown: The agents may break down so that they may not be able to move; the faulty agents are still assumed to be able to communicate with other agents. Suppose q agents fail. Then we have (assuming single integrator dynamics

for agents)

$$\dot{x}_i = 0, \qquad i \in I_q.$$

Then the time-derivative of the cost function $\mathcal{H}$ (equation (2.3)) becomes

$$\dot{H} = \sum_{i=1}^{N} \frac{\partial \mathcal{H}}{\partial x_i} \dot{x}_i \tag{3.60}$$

$$= \sum_{i=1, i \neq I_q}^{N} \frac{\partial \mathcal{H}}{\partial x_i} \dot{x}_i, \tag{3.61}$$

where I_q is the set of indices of the faulty agents. In this case, using the gradient control law (for the non-faulty agents) will lead to the non-faulty agent converging to the centroids of their respective Voronoi partitions whereas the faulty agents remain at their initial positions.

Chapter 4

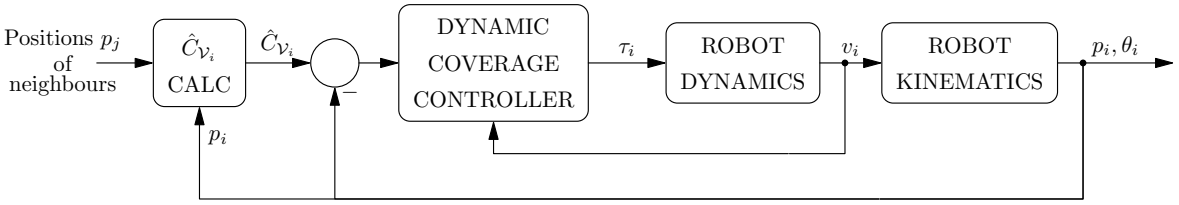
Simulation based study of Mobile Robots for Coverage

In this chapter, we provide a simulation-based study of the algorithms presented in chapter 3 for mobile robots. We compare the performance of the coverage algorithm developed based on the dynamic model with that based on the kinematic model alone. We also compare the performance of the adaptive algorithm (in case of model uncertainties) with that of the non-adaptive algorithm with wrong values of the model parameters. Block diagram representations of the coverage controllers for mobile robots discussed in the previous chapter are shown in Figure 4.1.

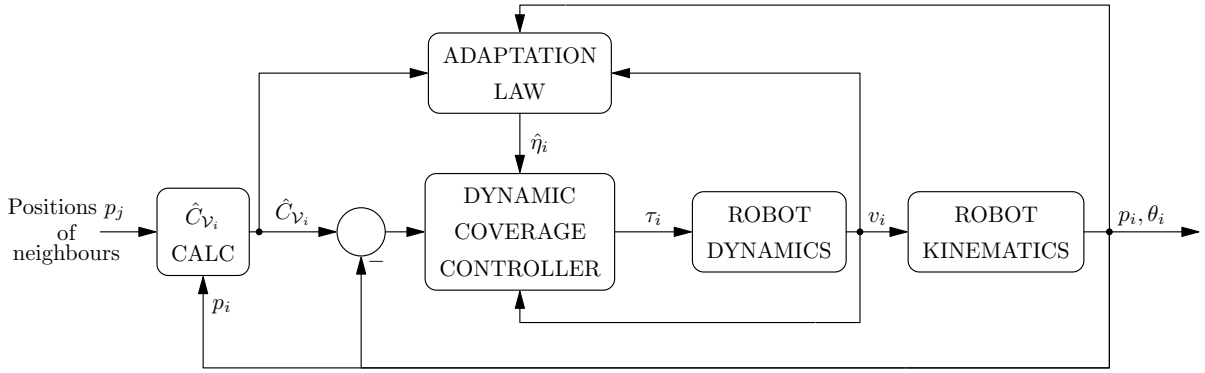
For simulations, we consider the unit square region $\mathcal{Q}$ with $N = 10$ agents. The density function $\phi(.)$ is a combination of two Gaussians and a constant term: $\phi(q) = \mathcal{K}(q)^\top a$ where $\mathcal{K}(q) = [\,\mathcal{K}_1(q),\, \mathcal{K}_2(q),\, \mathcal{K}_3(q)\,]^\top$ and $\mathcal{K}_i(q) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{(q-\mu_i)^\top (q-\mu_i)}{2\sigma^2}\right)$ for $i = 1, 2$, $\mathcal{K}_3(q) = 1$. The values of various constants used are given in Table 4.1. One of the Gaussians is weighted more than the other and forms the major component of the density function $\phi(\cdot)$.

4.1 Coverage with known dynamics

The simulation results are shown in Figure 4.3– 4.4. In the plots, we also give a comparison of the proposed controller (with known dynamics) with that of the coverage controller derived based on kinematics alone. The motivation is to study to what extent the controllers derived above based on the full dynamics give advantage over controllers



(a) Controller for agent i: Known dynamics



(b) Controller for agent i: Unknown dynamics

Figure 4.1: Block diagram of the proposed decentralized optimal coverage controllers

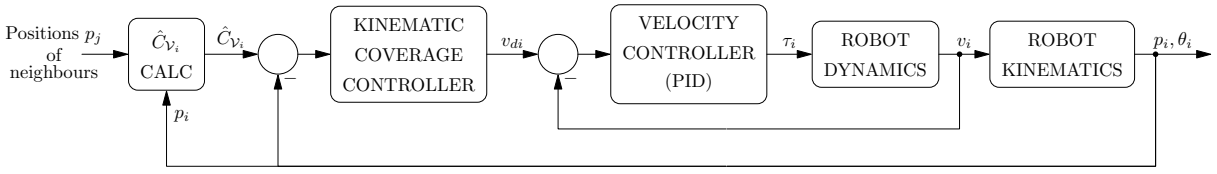


Figure 4.2: Block diagram of the kinematic optimal coverage controller used in comparison

Table 4.1: Simulation parameters: $\mathbf{I}$ represents the identity matrix

Parameter	Value	Parameter	Value
n	10	k_1	5
m	3	k_2	10
μ_1	[0.5, 0.3]	Γ_η	$10\mathbf{I}$
μ_2	[0.7, 0.5]	m	10 kg
σ^2	0.2	d	0.25 m
a	$[100, 0.3, 0.3]^\top$	r	0.05 m
Γ_a	$\mathbf{I}$	R	0.5 m
γ	300	I	5 kg.m^2
α	1	k_f	0.3 N.s.m^{-1}

derived based on kinematics alone. The kinematic model is given by (3.26) and the coverage control law for the kinematic model is given by $v_i = kv_{id}$ where v_{id} is given by (3.41). A block diagram representation of a robot with kinematic coverage controller is shown in Figure 4.2. A separate PID-based velocity controller is used for tracking the velocity command for the mobile robots. The parameters of the velocity controller was tuned to obtain similar average position error estimates as that with the dynamic model based coverage control. The values for the PID gains were $k_p = 100, k_d = 5, k_i = 5$.

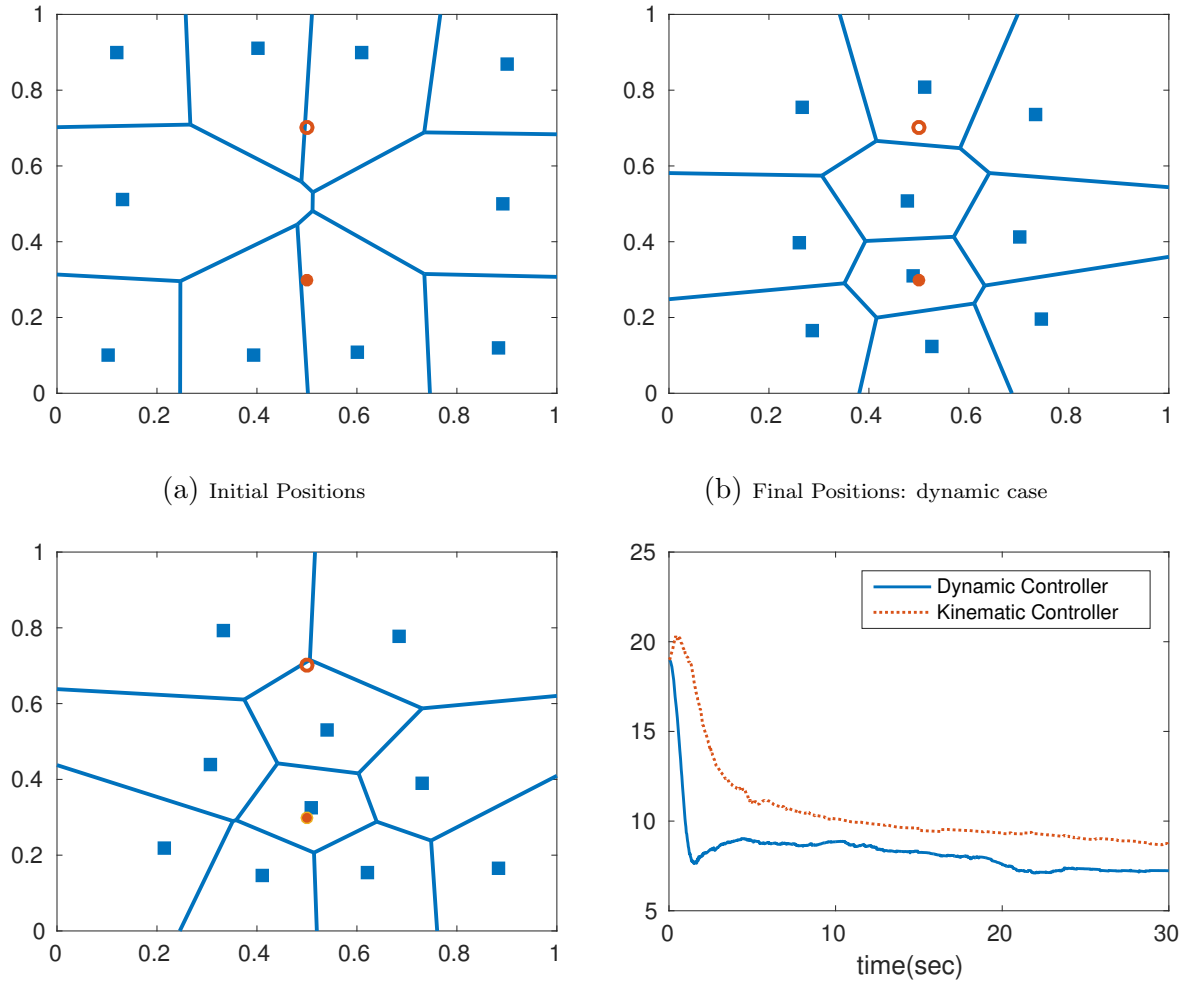
The initial and final positions of the robots and the corresponding Voronoi regions are shown in Figures 4.3a, 4.3b and 4.3c, respectively. The averaged position error $(p_i - \hat{C}_{\mathcal{V}_i})$ and velocity v_i are plotted in Figures 4.4a and 4.4b, respectively. The averaged integrated parameter error which is given by $\int_0^t \mathrm{e}^{-\alpha(t-\tau)} \left(\mathcal{K}_i^\top \tilde{a}_i(t)\right)^2 d\tau$ is plotted in figure 4.4c. This quantity is a weighted integral of the error in the density function estimate at time t along the path the agent has traversed till time t, and corresponds to the assertion (c) in theorems 3.3 and 3.6. It can be seen that the quantities approach zero with time. The average control torques are plotted in Figure 4.4d. It can be seen that the kinematic controller requires much higher torque initially as compared to the dynamic controller. The peak torque value for the kinematic case is 0.94 N.m and for the dynamic case is 0.24 N.m. In practical scenarios, the higher torque required may cause actuator saturation and wheel slips. The coverage cost function given by (2.3) is plotted in Figure 4.3d.

It can be seen that the cost function settles at a lower value for the dynamic case as compared to the kinematic case. The lower value of the true cost indicates better true coverage using the dynamic model based decentralized coverage law. The orientations of the agents and the average parameter error (across the agents) with the proposed dynamic coverage controller are plotted in figures 4.4e and 4.4f respectively, which indicates that the orientations converge to some constant value. We have not shown the plot of orientations for the kinematic controller case, as it is similar. We also see that the parameter errors converges to some constant value (not necessarily zero), but the integrated parameter error (shown in figure 4.4c) is guaranteed to converge to zero, as per Theorem 3.3. We find from the simulations that the dynamics based optimal coverage controller performs much better in terms of the position error convergence, coverage as well as the peak input torque required for the agents. It may also be noted that the initial conditions have an impact on the robot's final configuration since the final configuration the robots try to settle into is the minimizer of the coverage cost function (2.3). Convergence is guaranteed only to a local minimum of the cost function, and thus, different initial positions can lead to different final positions, although the behavior of the system (the nature of various plots) does not change with different initial conditions. The adaptation of the parameter estimates also changes with different initial conditions, which may also lead to changes in the robot trajectories, although convergence of the robots is guaranteed as per the theorems 3.3 and 3.6.

In figure 4.5, we also show how the agents reorganize themselves in case of failures discussed in section 3.4. The agents were simulated for 30 seconds, with the fault occuring at time $t = 10$ seconds. The initial positions and the final positions without faults is given in figure 4.3. In the case of communication breakdown, it is as if the faulty agent does not exist, since the other agents are not able to detect the faulty agent. In this case the whole configuration of agents seems to be skewed to the right where the faulty agent is. We do not further discuss the issue of faults in agents, and leave this for future work.

4.2 Coverage with unknown dynamics

We assume that the friction term $\bar{F}(v)$ is given by $\bar{F}(v) = k_f \text{sgn}(v)$ where k_f is unknown. For simplicity, the other parameters in the agent dynamics are assumed to be



(a) Initial Positions

(b) Final Positions: dynamic case

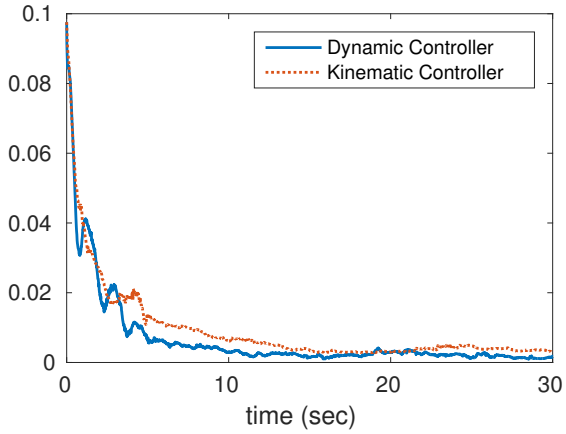
(c) Final Positions: kinematic case

(d) Cost Function

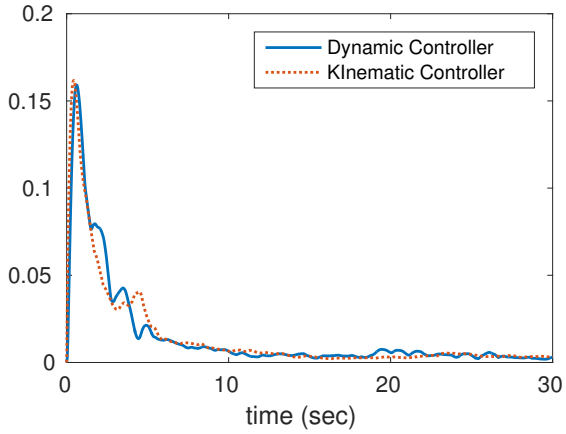
Figure 4.3: Simulations of optimal coverage control for the known dynamics case: The blue squares indicate the robot positions. The filled red circle indicates the mean of the Gaussian with the higher weight and the unfilled red circle indicates the mean of the other Gaussian component.

known. The true value of k_f is given in Table 4.1. The initial value for the adaptation law of k_f is assumed to be 0.1. In the plots, we also compare the performance of the adaptive algorithm (given in Theorem 3.6) with that of the non-adaptive algorithm (given in Theorem 3.3) with the unknown parameter assuming the value $k_f = 0.1$ (same as the initial value for the adaptive case). The initial and final positions of the robots with the corresponding Voronoi regions are shown in figures 4.6a, 4.6b, and 4.6c, respectively. The plots for average position error and average velocity is given in figures 4.7a and 4.7b, respectively. The average integrated parameter error is shown in figure 4.7c. The average control torque magnitude is plotted in figure 4.7d, and the cost function is plotted in figure 4.6d.

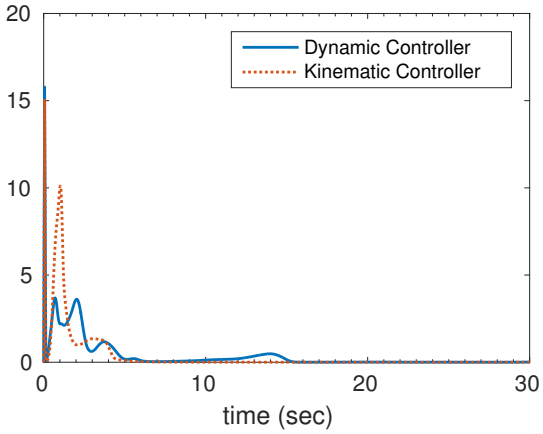
Comparing the adaptive and the non-adaptive controller case with an offset initial parameter value, we find that the average estimated position error is significantly larger for the latter due to the error in the parameter value whereas the adaptive algorithm is able to successfully converge towards the estimated centroid although the convergence rate is slower compared to the case with known dynamics. The average control torque required is much higher than the adaptive case (Peak torque magnitude values: adaptive case - 0.17 N.m, non-adaptive case - 1.03 N.m). Further, the true coverage cost is also much lower for the adaptive coverage controller. The orientations of the agents for the adaptive algorithm are shown in figure 4.7e. We have not shown the plot of orientations for the non-adaptive case as it is similar. We find from the above simulations that the adaptive controller performs much better compared to the non-adaptive controller with inexact values of the parameter.



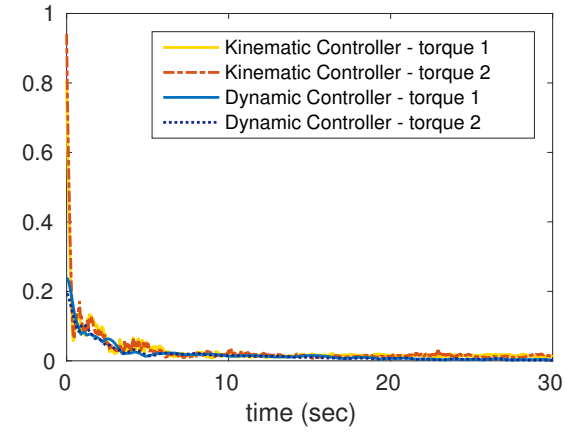
(a) Average Position Error



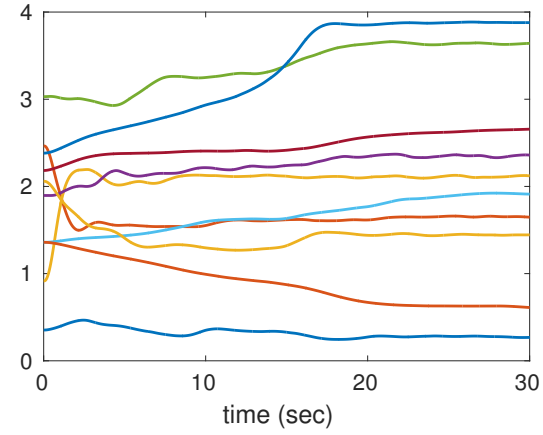
(b) Average Velocity



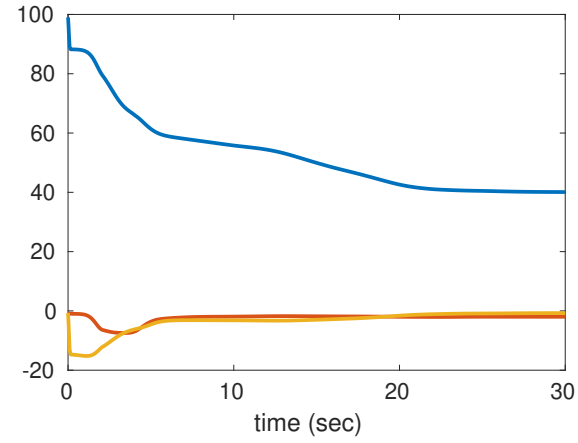
(c) Avg Integrated Parameter Error: Final value $(t = 30s) \sim 10^{-4}$



(d) Avg Control Torque

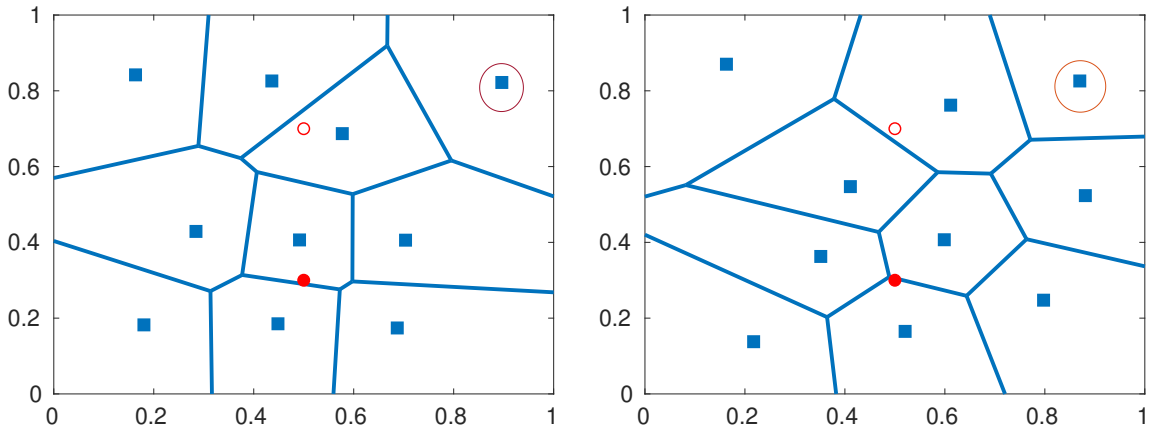


(e) Orientations: with dynamic controller



(f) Avg parameter errors: with dynamic controller

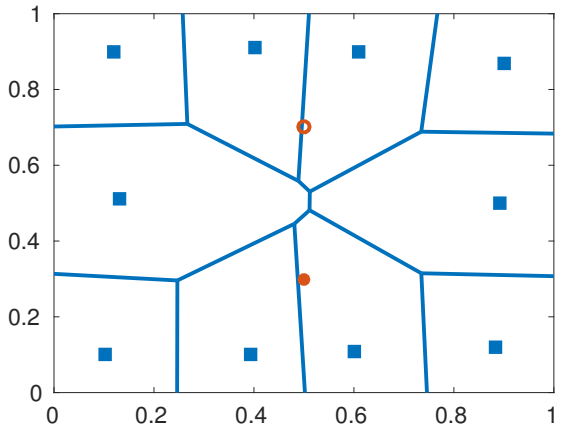
Figure 4.4: Simulation of decentralized optimal coverage controller for the known dynamics case: Comparison of dynamic and kinematic coverage controller



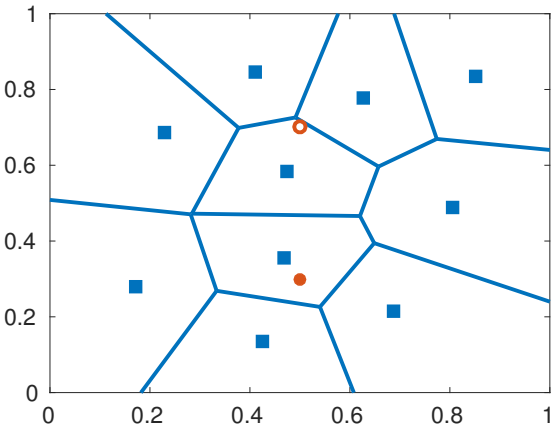
(a) Mobility breakdown

(b) Communication breakdown

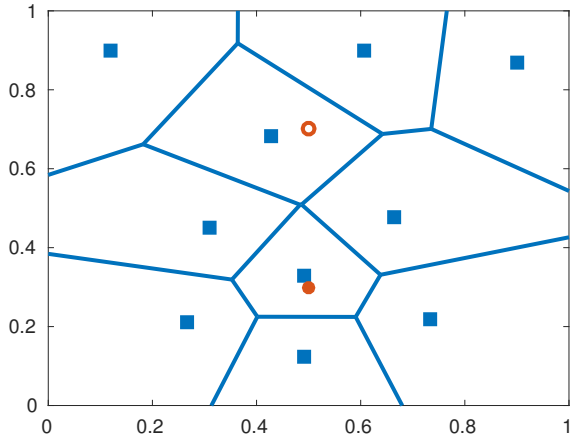
Figure 4.5: Final positions of agents after faults: the faulty agent is marked with a brown circle.



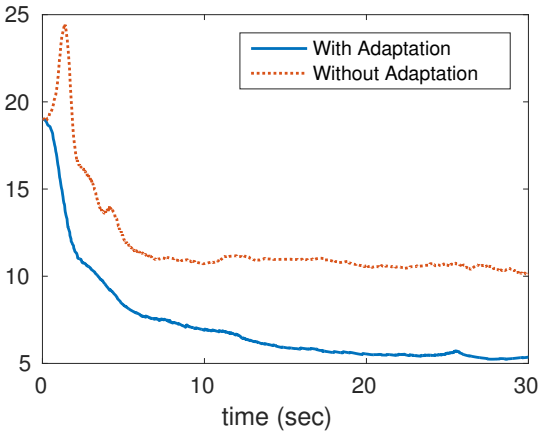
(a) Initial Positions



(b) Final Positions: adaptive controller

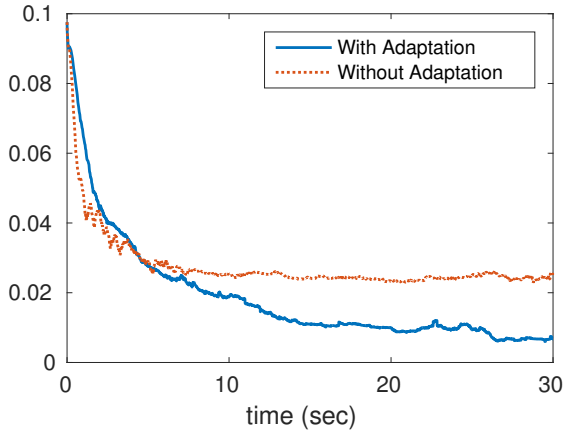


(c) Final Positions: non-adaptive controller

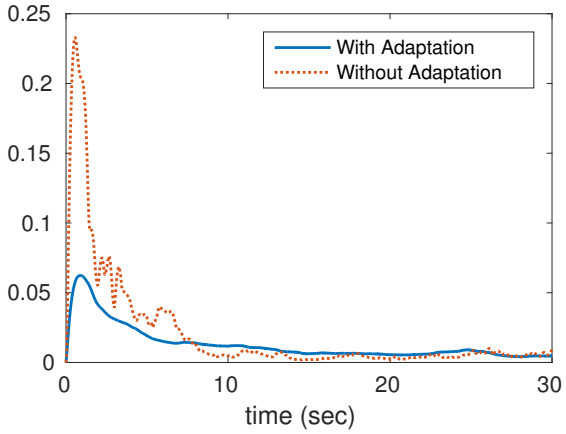


(d) Cost Function

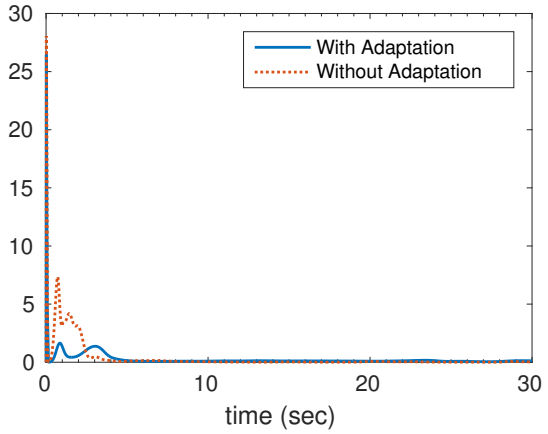
Figure 4.6: Simulation of optimal coverage control for the unknown dynamics case: The blue squares indicate the robot positions. The filled red circle indicates the mean of the Gaussian with the higher weight and the unfilled red circle indicates the mean of the other Gaussian component.



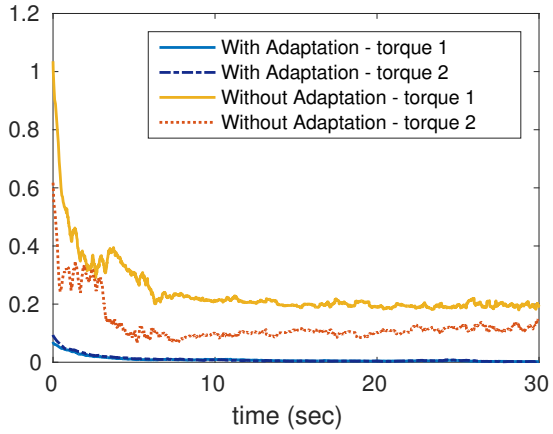
(a) Average Position Error



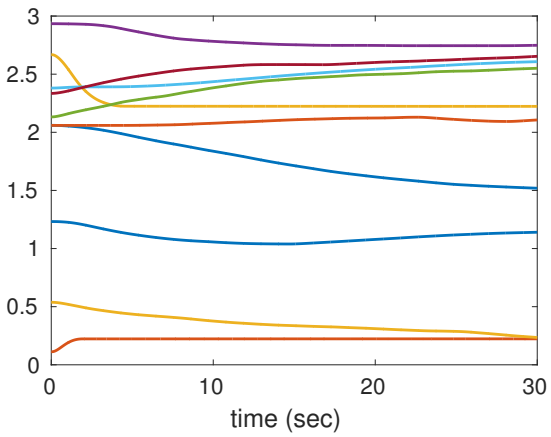
(b) Average Velocity



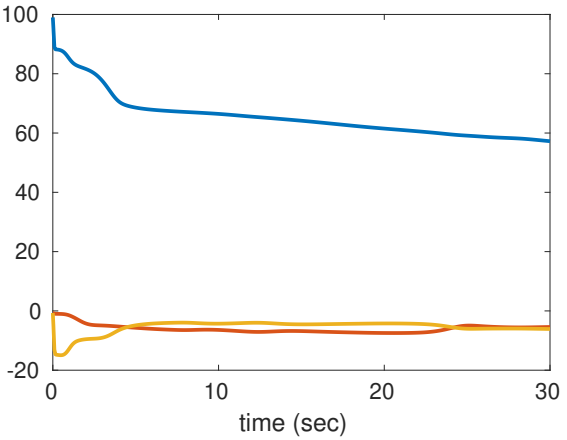
(c) Avg Integrated Parameter Error: Final value ($t = 30s$) $\sim 10^{-2}$



(d) Avg Control Torque



(e) Orientations: with adaptive controller



(f) Avg parameter errors: with adaptive controller

Figure 4.7: Simulation of decentralized optimal coverage control for the unknown dynamics case

Chapter 5

Towards a more general Coverage Problem Formulation

In this chapter, we try to formulate the coverage problem in a more general context, as an optimization problem in the space of density functions. In much of the works cited in the literature (see chapter 2), coverage problems are studied in the locational optimization framework, and generalized by using different weight functions describing the sensing capabilities of agents. In this chapter, we pose the coverage problem in a more general framework. We have the target density function which represents the phenomenon with respect to which coverage is to be achieved. We can also define an *agent sensing function* corresponding to each agent depending on the position as well as the sensing capabilities of the agent. Then the multiple agents located at different positions in the domain can be thought of as defining an *aggregate density function* of the agents. The coverage problem can then be posed as an optimization problem that seeks to minimize the *distance* between the target density function and the aggregate density function of the agents for an appropriately defined distance. This way of looking at coverage can lead to different notions of coverage by using different *distance* and *agent density* functions, depending on the application. We focus on the use of $\mathcal{L}^2$-distance as a means of achieving coverage.

5.1 Coverage as Distance between Densities

We consider the coverage problem over a region $\mathcal{Q} \subset \mathbb{R}^n$. The density function describing the event of interest with respect to which coverage is to be obtained (we will call this the *target density function*) is given by $\phi : \mathcal{Q} \to \mathbb{R}_+$.

Consider N agents to be deployed in region $\mathcal{Q}$ in order to cover the region with respect to the target density function $\phi(.)$. The location of the agents in Q will be denoted by p_i, $i = 1, 2, \ldots, N$. We will use P to denote the set of all agent positions. i.e. $P = \{p_1, p_2, \ldots, p_N\}$ Each agent is assumed to have a sensing capability which decreases with the distance from the agent location. We quantify this sensing capability of each agent as the *sensing function* denoted by $f_i(p_i, q)$. The sensing function describes the capability of the agent at position p_i to sense event at point $q \in \mathcal{Q}$. In this work, we assume isotropic sensors whose sensing is independent of direction. We can thus represent the sensing function as $f_i(\|p_i - q\|)$. We require $f_i : \mathbb{R}_+ \to \mathbb{R}_+$ to be an appropriate decreasing function of its argument. An illustration of the one-dimensional case is shown in figure 5.1. Given N agents each with its sensing functions, we define an *aggregate agent density*

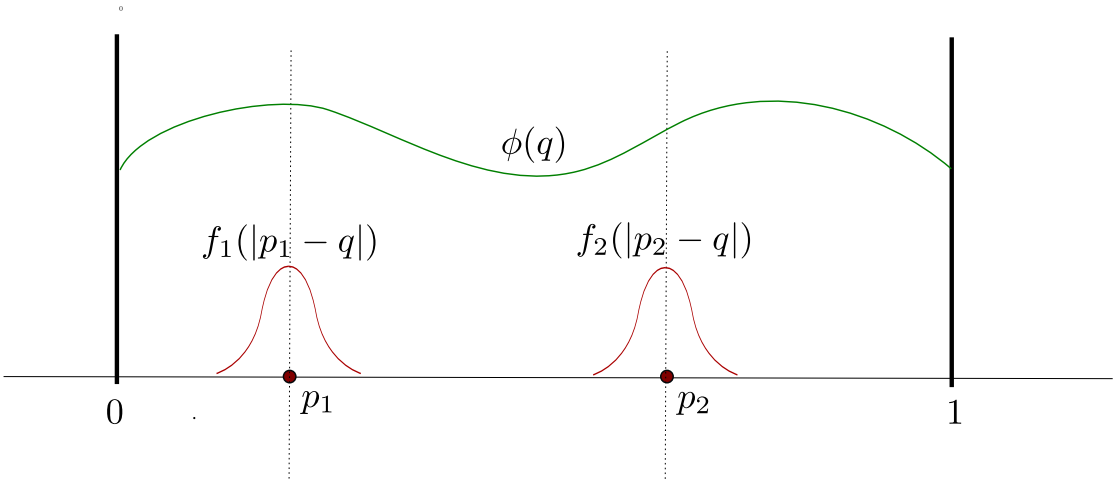


Figure 5.1: Two agents covering the interval $\mathcal{Q} = [0, 1]$: p_1 and p_2 are positions of two agents, with q representing a generic point of $\mathcal{Q}$.

function defined by

$$f_P(q) := \alpha(f_1(\|p_1 - q\|), f_2(\|p_2 - q\|), \ldots, f_N(\|p_N - q\|)),$$

where $\alpha : \mathbb{R}_+^N \to \mathbb{R}_+$ is an aggregation function that is to be chosen appropriately. The aggregate agent density gives a measure of the quality of sensing of the region $\mathcal{Q}$ using

all the N agents. The general coverage problem can then be posed as the following optimization problem:

$$\min_{P=\{p_1,p_2,\ldots,p_N\}} d(\phi, f_P), \tag{5.1}$$

where $d(\cdot,\cdot)$ is an appropriate measure of divergence/distance between ϕ and f_P. The agents should move to a configuration such that the *distance* between the target density ϕ and the aggregate agent density f_P is minimized. The optimal locations of the agents is then given by

$$P^* = \{p_1^*, p_2^*, \ldots, p_N^*\} = \arg\min_P d(\phi, f_P). \tag{5.2}$$

The choice of $d(\cdot,\cdot)$ will generally be determined by the class of functions we work with which depends on the nature of the coverage density function $\phi(\cdot)$ as well as the nature of sensors in the mobile agents employed. Similarly for the aggregate density function, there are no general properties required to be satisfied. But depending on the choice of the distance function, some requirements may be enforced on $\alpha(\cdot)$. For example, in the sequel we consider the squared integral cost, which requires both $\alpha(\cdot)$ and $\phi(\cdot)$ to be square integrable.

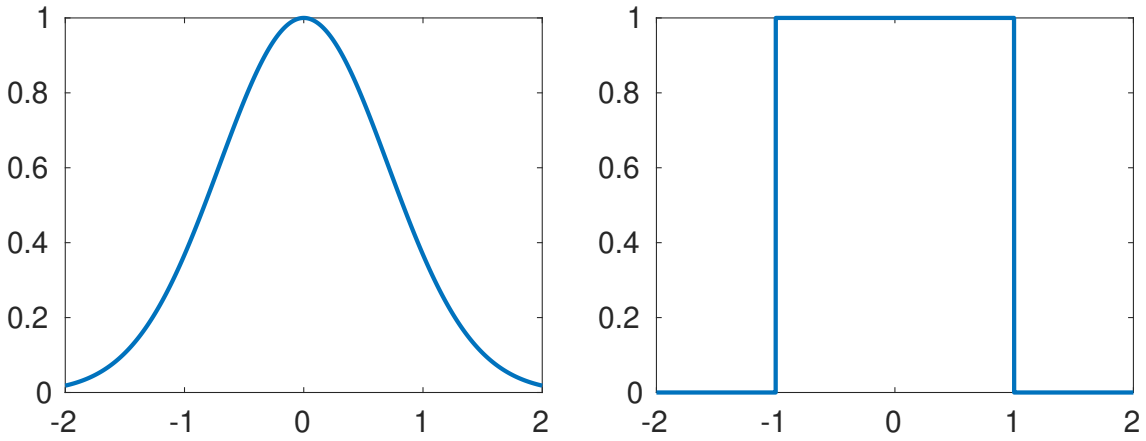
5.1.1 Choice of the agent sensing functions and aggregate density

The agent sensing functions $f_i(\|p_i - q\|)$ as mentioned above is required to be a non-increasing function of its argument, and depends on the nature of sensors. Some examples which we can be used are given below:

1. Gaussian function
$$f_i(\|p_i - q\|) = A_i \exp\left(-\frac{\|p_i - q\|^2}{\sigma_i^2}\right). \tag{5.3}$$

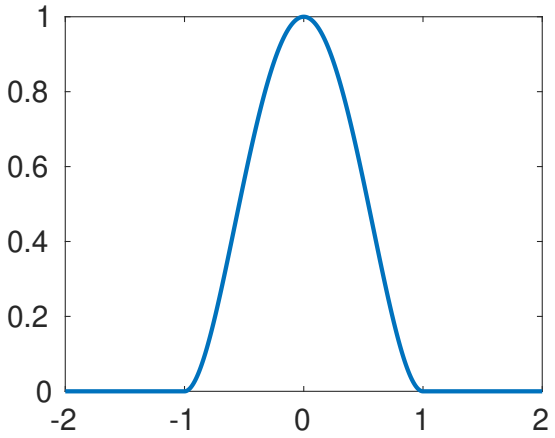
2. Constant sensing function
$$f_i(\|p_i - q\|) = \chi_{\{\|p_i - q\| \le r_i\}}, \tag{5.4}$$
where χ_S is the characteristic function of the set S, and $r_i > 0$ is some constant.

3. Quartic function
$$f_i(\|p_i - q\|) = \begin{cases} \frac{M}{r_i^4}\left(\|p_i - q\|^2 - r_i^2\right)^2 & \text{if } \|p_i - q\| < r_i, \\ 0 & \text{if } \|p_i - q\| \ge r_i, \end{cases} \tag{5.5}$$

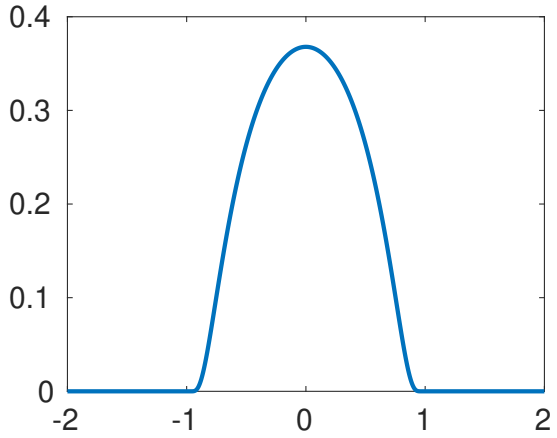


(a) Gaussian

(b) Constant



(c) Quartic



(d) Bump

Figure 5.2: Examples of sensing functions

for some constants $M > 0, r_i > 0$.

4. Bump function

$$f_i(\|p_i - q\|) = \begin{cases} \exp\left\{-\frac{1}{1-\|p_i-q\|^2}\right\} & \text{if } \|p_i - q\| < 1, \\ 0 & \text{if } \|p_i - q\| \geq 1. \end{cases} \tag{5.6}$$

The constant sensing function, the bump function and the quartic function (unlike the Gaussian function) allow us to model sensors with a finite sensing range since they take zero value outside a finite region of radius r_i around the agent position. Accordingly, we consider these functions as examples of *limited range* sensing functions, and the Gaussian function as an example of a *full range* sensing function. An illustration of the various one-dimensional functions are given in figure 5.2. The choice of aggregate agent density

$f_P(\cdot)$ could be made in different ways. We give here two natural choices:

1. Average/sum function
$$f_P(q) = \gamma \sum_{i=1}^{N} f_i(\|p_i - q\|), \tag{5.7}$$

2. Max function
$$f_P(q) = \gamma \max_i f_i(\|p_i - q\|), \tag{5.8}$$

where γ is a positive constant. The max function defines the aggregate density at a point as the value of the agent sensing function with the maximum value at the point. If the agents are identical, then the aggregate density at a point, in this case, is the agent density corresponding to the agent that is closest to the point. We will see that this naturally leads to a partition of the region $\mathcal{Q}$ and allows using distributed control schemes.

5.2 Locational Optimization as a special case

Now, we consider the locational optimization problem defined by the cost function (2.3) (see also [1])

$$\mathcal{H}(p_1, \ldots, p_N) = \int_Q \min_i \|p_i - q\|^2 \phi(q) dq \tag{5.9}$$
$$= \sum_{i=1}^{N} \int_{\mathcal{V}_i} \|p_i - q\|^2 \phi(q) dq, \tag{5.10}$$

where the domain $\mathcal{Q}$ is assumed to be compact and convex, p_i are the locations of the agents and $\{\mathcal{V}_i\}$ is the Voronoi partition of $\mathcal{Q}$ corresponding to $\{p_i\}$. The minimizer of the function $\mathcal{H}$ corresponds to the centroidal Voronoi configuration. These are the points p_i^* such that $p_i^* = C_{\mathcal{V}_i}$ where $C_{\mathcal{V}_i} = (\int_{\mathcal{V}_i} q\phi(q)dq)/(\int_{\mathcal{V}_i} \phi(q)dq)$. See chapters 2 and 3 for more on locational optimization.

We wish to pose the locational optimization problem in the general framework discussed above (equation (5.1)). In order to do that we consider the relative entropy or Kullback-Leibler divergence of two integrable functions $f : \mathcal{Q} \to \mathbb{R}_+$ and $g : \mathcal{Q} \to \mathbb{R}_+$ where $g(q) > 0$, defined as

$$d_{KL}(f, g) = \int_Q f(q) \log(\frac{f(q)}{g(q)}) dq. \tag{5.11}$$

d_{KL} is not a metric (since it is not symmetric), but it is a measure of divergence (distance) of g with respect to f and $d_{KL}(f, f) = 0$ (see [52, 53]). If we interpret the density function $\phi(\cdot)$ as defining the probability density of events occurring in the domain $\mathcal{Q}$, and agent sensing functions as defining the probability of detecting an event, then minimizing $d_{KL}(\phi, f_P)$ would mean that we are choosing those agent positions which results in maximum likelihood of detection of these events.

Proposition 5.1. *Assume that*

- *The target density $\phi(\cdot)$ is integrable.*
- *The sensing functions of the agents are Gaussian (equation (5.3)) with $A_i = 1$ and $\sigma_i = 1$.*
- *The aggregate density function of the agents is given by the max function (5.8) with $\gamma = 1$.*

Under the above assumptions,

$$\arg\min_P \; d_{KL}(\phi, f_P) = \arg\min_P \; \mathcal{H}(P),$$

i.e., minimizing $d_{KL}(\phi, f_P)$ is equivalent to minimizing $\mathcal{H}(P)$.

Proof. For each agent i

$$f_i(\|p_i - q\|) = \exp(-\|p_i - q\|^2),$$

and the aggregate density is given by

$$f_P(q) = \max_i \exp(-\|p_i - q\|^2).$$

Then,

$$\begin{aligned} d_{KL}(\phi, f_P) &= \int_Q \phi(q) \log \frac{\phi(q)}{f_P(q)} dq \\ &= \int_Q \phi(q) \log \phi(q) dq - \int_Q \phi(q) \log f_P(q) dq. \end{aligned}$$

Since the first term is independent of P, we have

$$\begin{aligned} \min_P d_{KL}(\phi, f_P) &= \min_P - \int_Q \phi(q) \log f_P(q) dq \\ &= \min_P - \int_Q \phi(q) \log \max_i f_i(\|p_i - q\|) dq \end{aligned}$$

$$
\begin{aligned}
&= \min_P - \int_Q \phi(q) \max_i \log f_i(\|p_i - q\|) dq \\
&= \min_P \int_Q \phi(q) \min_i \|p_i - q\|^2 dq \\
&= \min_P \mathcal{H}(P).
\end{aligned}
$$

□

As discussed in chapter 2, it can be shown (see [1]) that for agents with single integrator dynamics $\dot{p}_i = u_i$, the gradient control law given by

$$u_i = -\frac{\partial \mathcal{H}}{\partial p_i} = -kM_{\mathcal{V}_i}(p_i - C_{\mathcal{V}_i})$$

allows the agents to converge to the centroidal Voronoi configuration.

5.3 $\mathcal{L}^2$-norm-based coverage

We assume that the target density $\phi(\cdot)$ belong to $\mathcal{L}^2(\mathcal{Q})$[1]. We further assume that $\phi(\cdot)$ is lower bounded, i.e., $\phi(q) \geq \beta$ for some constant $\beta > 0$. Assuming that the aggregate density of the agents also belong to $\mathcal{L}^2(\mathcal{Q})$, we can define the following cost function

$$d_2(\phi, f_P) = \int_Q |\phi(q) - f_P(q)|^2 dq. \tag{5.12}$$

We now investigate the multi-agent coverage problem using the above cost function. Taking the derivative of $d_2(\phi, f_P)$ with respect to the p_i's

$$
\begin{aligned}
\frac{\partial d_2}{\partial p_i}(\phi, f_P) &= \int_{\mathcal{Q}} \frac{\partial}{\partial p_i} \left(|\phi(q) - f_P(q)|^2 \right) dq \\
&= -\int_{\mathcal{Q}} 2 \left(\phi(q) - f_P(q) \right) \frac{\partial}{\partial p_i} f_P(q)\, dq.
\end{aligned} \tag{5.13}
$$

Any minima of the cost function $d_2(.,.)$ with respect to P satisfies the condition $\frac{\partial d_2}{\partial p_i} = 0$ for every i.

As a simple example, consider $\mathcal{Q} = [0, 1]$ with one agent with $f(|p-q|) = \exp\{-|p-q|^2\} =: f_P(q)$. Assume the density function is given by $\phi(q) = \exp\{-|c - q|^2\}$ where $c \in (0, 1)$. Minimizing the $\mathcal{L}^2$ norm implies $\frac{\partial d_2}{\partial p} = 0$ which implies from (5.13) that $p = c$.

In the following, we consider the $\mathcal{L}^2$-distance with the max aggregate density function for two cases: (1) $f_i(\|p_i - q\|) > 0 \ \forall q \in \mathcal{Q}$ i.e. f_i is a full range sensing function, and (2) $f_i(\|p_i - q\|) = 0 \ \forall q \in \mathcal{Q} \setminus C_i$ for a given set $C_i \subset \mathcal{Q}$ i.e. f_i is a limited range sensing function.

[1] i.e., $\int_{\mathcal{Q}} |\phi(q)|^2 dq < \infty$.

5.3.1 Full range sensing function f_i

Consider a convex and compact region $\mathcal{Q} \subset \mathbb{R}^q$ with N agents and let

$$f_P(q) = \gamma \max_i f(\|p_i - q\|),$$

where the agent sensing function $f(.)$ is assumed to be the same for all the agents. We also assume that the support of the agent density $f(.)$ is the region $\mathcal{Q}$ itself (i.e., $f(\|p_i - q\|) > 0 \ \forall q \in \mathcal{Q}$). Then

$$\begin{aligned} d_2(\phi, f_P) &= \int_{\mathcal{Q}} |\phi(q) - f_P(q)|^2 \, dq \\ &= \int_{\mathcal{Q}} \left| \phi(q) - \gamma \left[\max_i f(\|p_i - q\|) \right] \right|^2 dq. \end{aligned} \tag{5.14}$$

Since $f(.)$ is a decreasing function of its argument, we can write the above as

$$d_2(\phi, f_P) = \sum_{i=1}^{N} \int_{\mathcal{V}_i} |\phi(q) - \gamma.f(\|p_i - q\|)|^2 \, dq, \tag{5.15}$$

where $\mathcal{V}_i$ are the Voronoi partitions defined by (see chapters 2 and 3)

$$\mathcal{V}_i = \{q \ : \ \|p_i - q\| \leq \|p_j - q\|, \quad \forall j \neq i\}.$$

Lemma 5.1. *The gradient of the $\mathcal{L}^2$ cost function (5.15) with respect to p_i is given by*

$$\frac{\partial d_2}{\partial p_i} = \int_{\mathcal{V}_i} \frac{\partial}{\partial p_i} |\phi(q) - \gamma.f(\|p_i - q\|)|^2 \, dq.$$

Proof. Computing the gradient with respect to p_i,

$$\begin{aligned} \frac{\partial d_2}{\partial p_i} = \frac{\partial}{\partial p_i} &\int_{\mathcal{V}_i} |\phi(q) - \gamma.f(\|p_i - q\|)|^2 \, dq \\ &+ \sum_{j \in \mathcal{N}_i} \frac{\partial}{\partial p_i} \int_{\mathcal{V}_j} |\phi(q) - \gamma.f(\|p_j - q\|)|^2 \, dq, \end{aligned} \tag{5.16}$$

where $\mathcal{N}_i$ denotes the set of neighbours of agent i. It should be noted that in the expression above the regions of integration $\mathcal{V}_i$ and $\mathcal{V}_j$ are themselves functions of p_i. Thus we have (see [4, 50])

$$\begin{aligned} \frac{\partial d_2}{\partial p_i} = &\int_{\mathcal{V}_i} \frac{\partial}{\partial p_i} |\phi(q) - \gamma.f(\|p_i - q\|)|^2 \, dq \\ &+ \int_{\partial \mathcal{V}_i} |\phi(q) - \gamma.f(\|p_i - q\|)|^2 \, n_{iq}^{\top} \frac{\partial q}{\partial p_i} dq \\ &+ \sum_{j \in \mathcal{N}_i} \int_{c_{ij}} |\phi(q) - \gamma.f(\|p_j - q\|)|^2 \, n_{jq}^{\top} \frac{\partial q}{\partial p_i} dq, \end{aligned} \tag{5.17}$$

where $\partial\mathcal{V}_i$ is the boundary of the Voronoi region $\mathcal{V}_i$, $\mathcal{N}_i$ is the set of Voronoi neighbours of agent i, n_{iq} is the unit normal at point q pointing outward from the region $\mathcal{V}_i$, and c_{ij} is the boundary segment of Voronoi region $\mathcal{V}_i$ that is shared with the Voronoi region of agent j ($\mathcal{V}_j$) (see also [4]). The boundary $\partial\mathcal{V}_i$ consists of segments c_{ij} and possibly parts of $\partial\mathcal{Q}$ (the boundary of $\mathcal{Q}$). The integrand of the second term is zero over $\partial\mathcal{Q}$. Thus the second and third terms in the above expression cancel each other (since the outward normals n_{iq} and n_{jq} in the two terms point opposite to each other), and the proof is complete. □

Now consider the Gaussian agent density function, $f(\|p_i - q\|) = \exp\{-\frac{\|p_i-q\|^2}{\sigma^2}\}$, $i = 1, 2, \ldots, N$. In this case, the gradient becomes

$$\begin{aligned}
\frac{\partial d_2}{\partial p_i} &= \int_{\mathcal{V}_i} \frac{\partial}{\partial p_i} \left|\phi(q) - \gamma.f(\|p_i - q\|)\right|^2 dq \\
&= \frac{4\gamma}{\sigma^2} \int_{\mathcal{V}_i} \left(\phi(q) - \gamma.f(\|p_i - q\|)\right)(p_i - q) \exp\left\{-\frac{\|p_i - q\|^2}{\sigma^2}\right\} dq \\
&= \frac{4\gamma}{\sigma^2} \left\{ p_i \int_{\mathcal{V}_i} \left[\phi(q) - \gamma. \exp\left\{-\frac{\|p_i - q\|^2}{\sigma^2}\right\}\right] \exp\left\{-\frac{\|p_i - q\|^2}{\sigma^2}\right\} dq \right. \\
&\qquad \left. - \int_{\mathcal{V}_i} q \left[\phi(q) - \gamma. \exp\left\{-\frac{\|p_i - q\|^2}{\sigma^2}\right\}\right] \exp\left\{-\frac{\|p_i - q\|^2}{\sigma^2}\right\} dq \right\} \\
&= \frac{4\gamma}{\sigma^2} \left\{ p_i \int_{\mathcal{V}_i} \lambda_i(q) dq - \int_{\mathcal{V}_i} q\lambda_i(q) dq \right\},
\end{aligned} \tag{5.18}$$

where

$$\lambda_i(q) := \left[\phi(q) - \gamma \cdot \exp\left\{-\frac{\|p_i - q\|^2}{\sigma^2}\right\}\right] \exp\left\{-\frac{\|p_i - q\|^2}{\sigma^2}\right\}. \tag{5.19}$$

Setting the gradient $\frac{\partial d_2}{\partial p_i} = 0$ gives

$$p_i = \frac{\int_{\mathcal{V}_i} q\lambda_i(q) dq}{\int_{\mathcal{V}_i} \lambda_i(q) dq} = \frac{L^{\lambda}_{\mathcal{V}_i}}{M^{\lambda}_{\mathcal{V}_i}}, \tag{5.20}$$

with $L^{\lambda}_{\mathcal{V}_i} = \int_{\mathcal{V}_i} q\lambda_i(q) dq$ and $M^{\lambda}_{\mathcal{V}_i} = \int_{\mathcal{V}_i} \lambda_i(q) dq$.

Lemma 5.2. *For $0 < \gamma \leq \beta$ (recall that β is the lower bound on ϕ), $\lambda_i(q) \geq 0$ for $q \in \mathcal{V}_i$ and $i = 1, 2, \ldots, N$.*

Proof. The second term in equation (5.19) an exponential and is always positive. The first term (in the square brackets) is non-negative if $0 < \gamma \leq \beta$. Thus proved. □

Remark 5.1. Choosing γ as per the above lemma guarantees that the aggregate density function of the agents $f_P(.)$ is scaled below the lower bound of $\phi(.)$ function. This might be detrimental to matching f_P and ϕ if the lower bound β is very small and there is a

large variation in $\phi(.)$. An alternative could be to add a constant bias to the density function. We will not, however, consider this case here and leave it for future work.

Remark 5.2. Although we have derived the expression for λ_i's assuming that the agent density functions f_i are Gaussians, the above computations carry over to any definition of f_i's which are decreasing as a function of $\|p_i - q\|^2$.

We define

$$\mathcal{C}^{\lambda}_{\mathcal{V}_i} := \frac{L^{\lambda}_{\mathcal{V}_i}}{M^{\lambda}_{\mathcal{V}_i}} = \frac{\int_{\mathcal{V}_i} q\lambda_i(q)dq}{\int_{\mathcal{V}_i} \lambda_i(q)dq}. \tag{5.21}$$

Thus the critical points of the $\mathcal{L}^2$ optimization problem is described by $p_i = \mathcal{C}^{\lambda}_{\mathcal{V}_i}$. λ_i defines a density on $\mathcal{V}_i$ and $C_{\mathcal{V}_i}$ is the generalized centroid of $\mathcal{V}_i$ corresponding to the density function λ_i. We call the critical point defined by $p_i = \mathcal{C}^{\lambda}_{\mathcal{V}_i}$; $i = 1, 2, \ldots, N$ as a generalized centroidal Voronoi configuration corresponding to the λ_i's. From the definition of $\mathcal{C}^{\lambda}_{\mathcal{V}_i}$ and Lemma 5.2, we can conclude that $\mathcal{C}^{\lambda}_{\mathcal{V}_i} \in \mathcal{V}_i$ for $0 < \gamma \leq \beta$, i.e., $\mathcal{C}^{\lambda}_{\mathcal{V}_i}$ is contained in the convex set $\mathcal{V}_i$.

5.3.1.1 Gradient control laws for single integrator agents

Consider N agents deployed in region $\mathcal{Q} \subset \mathbb{R}^n$. The agent dynamics are given by

$$\dot{p}_i = u_i, \qquad i = 1, 2, \ldots, N, \tag{5.22}$$

where $p_i \in \mathcal{Q}$ is the position of agent i and $u_i \in \mathbb{R}^n$ is the control input. Then we have the following result.

Theorem 5.3. *For agents with dynamics given by 5.22, the control law given by*

$$u_i = -K_p(p_i - \mathcal{C}^{\lambda}_{\mathcal{V}_i}), \tag{5.23}$$

with $0 < \gamma \leq \beta$ and $K_p > 0$, drives the agents to a minimum of the cost function (5.12) which is the generalized centroidal Voronoi configuration with respect to the λ_i's.

Proof. Consider the function

$$V(t) = d_2(\phi, f_P).$$

Taking the derivative,

$$\begin{aligned}\dot{V} &= \sum_{i=1}^{N} \frac{\partial d_2}{\partial p_i}^{\top} \dot{p}_i = \sum_{i=1}^{N} \frac{\partial d_2}{\partial p_i}^{\top} u_i \\ &= \sum_{i=1}^{N} 4\gamma M^{\lambda}_{\mathcal{V}_i}(p_i - \mathcal{C}^{\lambda}_{\mathcal{V}_i})^{\top} u_i.\end{aligned}$$

Substituting the control law, we get

$$\dot{V} = -\sum_{i=1}^{N} 4\gamma K_p M_{\mathcal{V}_i}^{\lambda} \|p_i - \mathcal{C}_{\mathcal{V}_i}^{\lambda}\|^2.$$

V is continuously differentiable on the compact set $\mathcal{Q}$, and $\mathcal{Q}$ is positively invariant with respect to the closed-loop dynamics. Since $\dot{V} \leq 0$, from LaSalle invariance principle ([54]), we conclude that the trajectories converge to the largest invariant set in $\{p_i : \|p_i - \mathcal{C}_{\mathcal{V}_i}^{\lambda}\| = 0\}$, which is the set itself. Since the cost function V is decreasing with time, we see that it converges to a minimum. This concludes the proof. □

Remark 5.4. The above control law is similar to the control law derived for the locational optimization case discussed in section 2 ([1]) except that in defining the generalized centroid, $\phi(q)$ is replaced by the modified function $\lambda_i(q)$ for $q \in \mathcal{V}_i$.

5.3.2 Limited range sensing function f_i

In this section, we discuss the $\mathcal{L}^2$ cost function case with the max aggregate density where the agent sensors have a finite range or the support of the agent sensing function is finite. Let C_i represent the support of function $f(\|p_i - q\|)$, i.e. C_i, $f(\|p_i - q\|)$ is zero outside of C_i. Let us define $e(q) := |\phi(q) - f_P(q)|$ and $e_i(q) := |\phi(q) - \gamma f(\|p_i - q\|)|$. Then the $\mathcal{L}^2$ cost functional can be written as

$$\begin{aligned} d_2(\phi, f_P) &= \int_{\mathcal{Q}} (e(q))^2 \, dq \\ &= \int_{\bigcup_i C_i} (e(q))^2 \, dq + \int_{\mathcal{Q}\setminus\bigcup_i C_i} (e(q))^2 \, dq \\ &= \sum_{i=1}^{N} \int_{D_i} (e_i(q))^2 \, dq + \int_{\mathcal{Q}\setminus\bigcup_k D_k} |\phi(q)|^2 dq, \end{aligned} \tag{5.24}$$

where the domains D_i are defined by:

$$D_i := \{q \in C_i \ : \ \|p_i - q\| \leq \|p_j - q\|, \quad \forall j \neq i\}. \tag{5.25}$$

Thus the set D_i represent a Voronoi partition of the set $\bigcup_{i=1}^{N} C_i$. In other words, $D_i = \mathcal{V}_i \cap C_i$.

Lemma 5.3. *The gradient of the $\mathcal{L}^2$ cost function (5.24) is given by*

$$\frac{\partial d_2}{\partial p_i} = \int_{D_i} \frac{\partial}{\partial p_i} |\phi(q) - \gamma f(\|p_i - q\|)|^2 dq.$$

Proof. Computing the gradient of d_2,

$$\begin{aligned}\frac{\partial d_2}{\partial p_i} = &\int\limits_{D_i} \frac{\partial}{\partial p_i} e_i^2(q)dq + \int\limits_{\partial D_i} e_i^2(q) n_q^\top \frac{\partial q}{\partial p_i} dq \\ &+ \sum_{j\in\mathcal{N}_i} \int\limits_{\partial D_j} e_j^2(q) n_q^\top \frac{\partial q}{\partial p_i} dq + \int\limits_{\partial\{\mathcal{Q}\setminus\bigcup_k D_k\}} |\phi(q)|^2 n_q^\top \frac{\partial q}{\partial p_i} dq.\end{aligned} \tag{5.26}$$

The boundaries ∂D_i of the partitions D_i in consists of a union of spherical sectors and disks in 3D space ($n = 3$). In 2D ($n = 2$) space, they consist of union of circular arcs and line segments. We will use the 2D case to develop the expressions further. Similar extensions apply to the 3D case, although the computations are more involved. Let the line segments in ∂D_i be denoted by $l_{i,j}$ where $j \in \mathcal{N}_i$ and the circular arcs be denoted by $c_{i,j}$ where $j = 1, 2, \ldots, n_{c_i}$. Then we have

$$\begin{aligned}\frac{\partial d_2}{\partial p_i} = &\int\limits_{D_i} \frac{\partial}{\partial p_i} e_i^2(q)dq + \sum_{j=1}^{n_{c_i}} \int\limits_{c_{i,j}} e_i^2(q) n_q^\top \frac{\partial q}{\partial p_i} dq + \sum_{j\in\mathcal{N}_i} \int\limits_{l_{i,j}} e_i^2(q) n_q^\top \frac{\partial q}{\partial p_i} dq \\ &+ \sum_{j\in\mathcal{N}_i} \int\limits_{l_{i,j}} e_j^2(q) n_q^\top \frac{\partial q}{\partial p_i} dq + \int\limits_{\partial\{\mathcal{Q}\setminus\bigcup_k D_k\}} |\phi(q)|^2 n_q^\top \frac{\partial q}{\partial p_i} dq,\end{aligned}$$

where n_{c_i} is the number of circular arc components in ∂D_i. The two integrals over $l_{i,j}$ cancel each other since on $l_{i,j}$, $f_i(q) = f_j(q)$ and the normal vector n_q point opposite each other resulting in

$$\begin{aligned}\frac{\partial d_2}{\partial p_i} = &\int_{D_i} \frac{\partial}{\partial p_i} |\phi(q) - \gamma f_i(q)|^2 dq + \sum_{j=1}^{n_{c_i}} \int_{c_{i,j}} |\phi(q) - \gamma f_i(q)|^2 n_q^\top \frac{\partial q}{\partial p_i} dq \\ &+ \int_{\partial\{\mathcal{Q}\setminus\bigcap D_i\}} |\phi(q)|^2 n_q^\top \frac{\partial q}{\partial p_i} dq.\end{aligned} \tag{5.27}$$

Noting that $\partial\{\mathcal{Q} \setminus \bigcup_k D_k\} = \bigcup_j c_{i,j}$, we get

$$\begin{aligned}\frac{\partial d_2}{\partial p_i} = &\int_{D_i} \frac{\partial}{\partial p_i} |\phi(q) - \gamma f_i(q)|^2 dq + \sum_{j=1}^{n_{c_i}} \int_{c_{i,j}} |\phi(q) - \gamma f_i(q)|^2 n_q^\top \frac{\partial q}{\partial p_i} dq \\ &+ \sum_{j=1}^{n_{c_i}} \int_{c_{i,j}} |\phi(q)|^2 n_q^\top \frac{\partial q}{\partial p_i} dq.\end{aligned} \tag{5.28}$$

Since the agent sensing function $f(\|p_i - q\|)$ reduces to zero on $c_{i,j}$ (since the $c_{i,j}$'s are part of the boundary of the support of the sensing function), we see that the second and third terms in the above expression cancel each other (the unit normals n_q in the two integrals are opposite each other) resulting in,

$$\frac{\partial d_2}{\partial p_i} = \int\limits_{D_i} \frac{\partial}{\partial p_i} \left|\phi(q) - \gamma f_i(\|p_i - q\|)\right|^2 dq, \tag{5.29}$$

and the proof is complete. □

Now consider the quartic sensing function given by equation (5.5).

$$f_i(\|p_i - q\|) = \begin{cases} \frac{1}{r_s^4}\left(\|p_i - q\|^2 - r_s^2\right)^2 & \text{if } \|p_i - q\| < r_s, \\ 0 & \text{if } \|p_i - q\| \geq r_s, \end{cases}$$

for some constant $r_s > 0$. Computing the gradient using Lemma 5.3 for this function, we have

$$\frac{\partial d_2}{\partial p_i} = \frac{8\gamma}{r_s^4} M_{D_i}^{\mu}\left(p_i - \mathcal{C}_{D_i}^{\mu}\right), \tag{5.30}$$

where $M_{D_i}^{\mu} = \int\limits_{D_i} \mu_i(q)dq$, $L_{D_i}^{\mu} = \int\limits_{D_i} q\mu_i(q)dq$, $\mathcal{C}_{D_i}^{\mu} = \frac{L_{D_i}^{\mu}}{M_{D_i}^{\mu}}$, and $\mu_i(q) = \left[\phi(q) - \frac{\gamma}{r_s^4}\left(\|p_i - q\|^2 - r_s^2\right)^2\right]\left(r_s^2 - \|p_i - q\|^2\right)$. Similar to Lemma 5.2, we have the following result.

Lemma 5.4. *For $0 < \gamma < \beta \cdot r_s^4$, $\mu_i(q) \geq 0$ for all $q \in D_i$ and $i = 1, 2, \ldots, N$.*

Theorem 5.5 (Gradient control law - limited range case)**.** *Consider N agents with single integrator dynamics $\dot{p}_i = u_i$. Assume that the agent sensing function is given by the quartic function, and the aggregate density is given by the* max *function with $0 < \gamma < \beta \cdot r_s^4$. Then the control law given by*

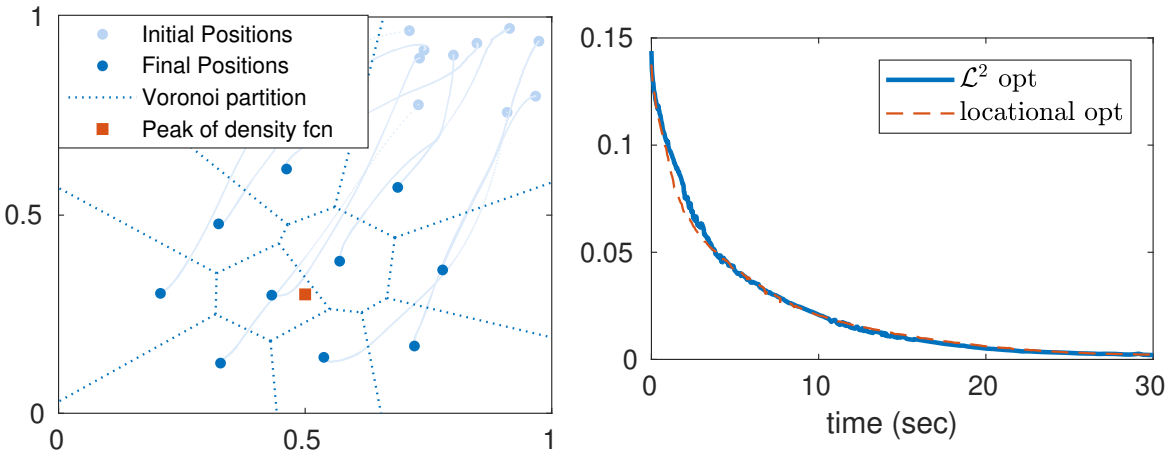
$$u_i = -K_p\left(p_i - \mathcal{C}_{D_i}^{\mu}\right)$$

drives the agents to a minimum of the cost function (5.15)*.*

Proof. The proof is omitted here as it is identical to the proof of Theorem 5.3. □

5.4 Simulations

In this section, we present a few simulation results for the $\mathcal{L}^2$-distance based coverage algorithm. We consider the unit square planar region $\mathcal{Q}$ with $N = 10$ agents. The density function $\phi(.)$ is assumed to be the sum of three components: $\phi(q) = \mathcal{K}(q)^\top a$ where $\mathcal{K}(q) = [\mathcal{K}_1(q)\ \mathcal{K}_2(q)\ \mathcal{K}_3(q)]^\top$ and $\mathcal{K}_i(q) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(\frac{(q-\mu_i)^\top(q-\mu_i)}{2\sigma^2}\right)$ for $i = 1, 2$, $\mathcal{K}_3(q) = 1$. The a vector represents the strength of the individual components of the density function. The values of various parameters are given in table 5.1. The density function is dominated by the Gaussian function $\mathcal{K}_1(q)$. The simulations are run for 30 seconds.



(a) Initial and Final Positions

(b) Average position error

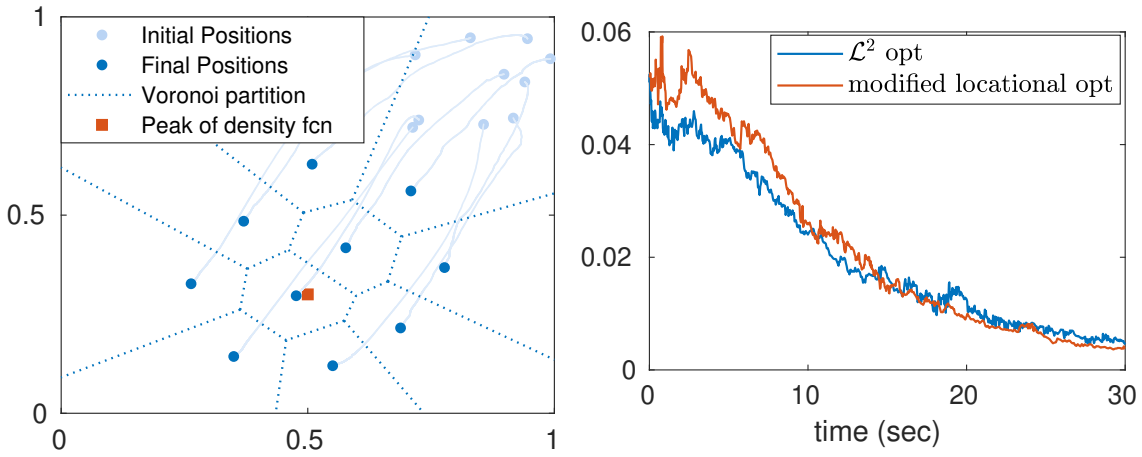
Figure 5.3: Simulation results: Gaussian Sensing function

5.4.1 Gaussian sensing function

The simulation result for the $\mathcal{L}^2$-coverage algorithm described in section 5.3.1 is shown in figure 5.3a. In the plot, the dark circular spots refer to the final positions of the agents, and the light circular spots refer to the initial positions of the agents. The red square represents the center of the dominant component of the density function. The agent sensing functions $f_i(.)$ are assumed to be Gaussian (see section 5.1.1) and the value of γ is chosen to be 1. The average position error defined by $e = \frac{1}{N}\sum_{i=1}^{N} \|p_i - \mathcal{C}^{\lambda}_{\mathcal{V}_i}\|$ is plotted in figure 5.3b. In the plot, we also show the average position error for coverage control using the locational optimization cost function (2.3) under the same conditions. We see that there is not much difference in the rate of convergence in the two cases, although it must be noted that the cost functions are different, and the two cases lead to different final configurations. Thus it is very hard to compare the quality of coverage in the two cases. Developing methods to compare different notions of coverage is part of future work.

Table 5.1: Simulation parameters

Par.	Value	Par.	Value	Par.	Value
n	10	K_p	1	a	$[100, 0.3, 1]^\top$
μ_1	$[0.5, 0.3]^\top$	μ_2	$[0.5, 0.7]^\top$	σ^2	0.2



(a) Initial and Final Positions

(b) Average position error

Figure 5.4: Simulation results: Limited range (quartic) sensing function

5.4.2 Limited range sensing function

This case corresponds to that discussed in section 5.3.2. The agent sensing functions are assumed to be the quartic function (see section 5.1.1). The value of the radius r_s was chosen to be 0.25, and γ was chosen to be 0.0035. The trajectories are shown in figure 5.4a. The average position error defined by $e = \frac{1}{N}\sum_{i=1}^{N} \|p_i - \mathcal{C}^{\mu}_{D_i}\|$ is shown in figure 5.4b. The convergence seems to be slower in this case compared to the case with the gaussian sensing function. We also plot the result for a modified version of locational optimization with limited range sensing. The convergence is seen to be similar in both cases.

Chapter 6

Hardware Implementation and More on Parameter Convergence

In this chapter, we discuss the hardware implementation of the coverage algorithm on differential drive robots. ROS (Robot Operating System) is used for implementing the algorithms. Control and adaptation laws for differential drive robots are derived in the $\mathcal{L}^2$ framework. Experimental results are used to compare the locational optimization method with the $\mathcal{L}^2$-distance method, focusing on the issue of parameter convergence. Using a consensus term in the adaptation law was shown to improve parameter convergence [30]. We discuss a method to improve upon the existing consensus algorithm to provide faster convergence of parameters.

6.1 Control and Adaptation laws

In this section, we describe the coverage control and adaptation laws for the mobile robots in the $\mathcal{L}^2$-distance framework. The robot model we consider is the differential drive kinematic model described in chapter 3:

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \end{bmatrix} = \begin{bmatrix} \cos(\theta_i) & -d\sin(\theta_i) \\ \sin(\theta_i) & d\cos(\theta_i) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_i \\ \omega_i \end{bmatrix}. \tag{6.1}$$

Here, (x_i, y_i) is the position of the center point of robot i, which corresponds to the point which is tracked by the localization system). θ_i is the orientation of robot i. u_i and ω_i

are the linear and angular velocity commands. d is the distance of the center point from the wheel axis.

We now derive the $\mathcal{L}^2$ coverage control and adaptation laws for the differential drive robots. The derivation is similar to that in Section 3.3. We denote the estimate of parameter a of agent i at time t by $\hat{a}_i(t)$ and the measurement of $\phi(\cdot)$ by the agent i by $\phi_i(t) = \phi(p_i(t))$, where $p_i(t)$ is the position of agent i at time t. Now we define the following filters:

$$\begin{aligned} \dot{\Lambda}_i(t) &= -\alpha\Lambda_i(t) + \mathcal{K}_i(t)\mathcal{K}_i(t)^\top, \\ \dot{\lambda}_i(t) &= -\alpha\lambda_i(t) + \mathcal{K}_i(t)\phi_i(t), \end{aligned} \tag{6.2}$$

where $\mathcal{K}_i(t) = \mathcal{K}(p_i(t))$, $\Lambda_i(0) = 0$, $\lambda_i(0) = 0$.

The control and adaptation laws are then given by

$$v_i = -k_1 \left[S_i^1(q)\right]^{-1} (p_i - \hat{C}_{\mathcal{V}_i}^{\lambda}), \tag{6.3}$$

$$\dot{\hat{a}}_i = \Gamma(b_i - I_{\beta_i} b_i), \tag{6.4}$$

$$b_i = -k_2 \int_{\mathcal{V}_i} e^{-\frac{\|p_i - q\|^2}{\sigma^2}} \mathcal{K}(q)(q - p_i)^T dq \; S_i^1(q) v_i - \gamma(\Lambda_i \hat{a}_i - \lambda_i), \tag{6.5}$$

where $k_1, k_2 > 0$ are positive gains and $S_i^1(q)$ is the matrix defined as

$$S_i^1(q) = \begin{bmatrix} \cos(\theta_i) & -d\sin(\theta_i) \\ \sin(\theta_i) & d\cos(\theta_i) \end{bmatrix}. \tag{6.6}$$

We can now state the following theorem.

Theorem 6.1. *Consider N differential drive robots deployed in the region $\mathcal{Q}$ for covering the region. Assume that the robots implement the control law* (6.3) *and the adaptation law* (6.4) *and* (6.5)*. Then the following statements hold:*

1. $\lim_{t\to\infty} \|p_i - \hat{C}_{\mathcal{V}_i}^{\lambda}\| = 0$,
2. $\lim_{t\to\infty} \|v_i\| = 0$,
3. $\lim_{t\to\infty} \mathcal{K}_i(\tau)\tilde{a}_i(t) = 0$, *for* $\forall\tau \;\; s.t \;\; t - T < \tau < t$, *where* $T > 0$,

for all $i = 1, 2, \ldots, N$.

Proof. Consider the Lyapunov function

$$V = d_2(\phi, f_P) + \frac{1}{2}\sum_{i=1}^{N} \tilde{a}_i^\top \Gamma^{-1} \tilde{a}_i.$$

Taking the derivative,

$$\dot{V} = \sum_{i=1}^{N} \frac{\partial d_2}{\partial p_i}^{\top} \dot{p}_i + \sum_{i=1}^{N} \tilde{a}_i^{\top} \Gamma^{-1} \dot{\hat{a}}_i.$$

Using (5.18), we get

$$\dot{V} = \sum_{i=1}^{N} \frac{4\gamma}{\sigma^2} M_{\mathcal{V}_i}^{\lambda} (p_i - C_{\mathcal{V}_i}^{\lambda})^{\top} \dot{p}_i + \sum_{i=1}^{N} \tilde{a}_i^{\top} \Gamma^{-1} \dot{\hat{a}}_i.$$

Using the definitions of $M_{\mathcal{V}_i}^{\lambda}$ and $C_{\mathcal{V}_i}^{\lambda}$ given in section 5.3.1,

$$\dot{V} = \frac{4\gamma}{\sigma^2} \sum_{i=1}^{N} \int_{\mathcal{V}_i} (p_i - q)^{\top} \lambda_i(q) dq\, \dot{p}_i + \sum_{i=1}^{N} \tilde{a}_i^{\top} \Gamma^{-1} \dot{\hat{a}}_i. \tag{6.7}$$

Now $\lambda_i(q)$ can be written as

$$\begin{aligned}
\lambda_i(q) &= e^{-\frac{\|p_i - q\|^2}{\sigma^2}} \left[\phi(q) - \gamma e^{-\frac{\|p_i - q\|^2}{\sigma^2}} \right] \\
&= e^{-\frac{\|p_i - q\|^2}{\sigma^2}} \left[\mathcal{K}(q)^{\top} (\hat{a}_i - \tilde{a}_i) - \gamma e^{-\frac{\|p_i - q\|^2}{\sigma^2}} \right] \\
&= e^{-\frac{\|p_i - q\|^2}{\sigma^2}} \left[\hat{\phi}(q) - \gamma e^{-\frac{\|p_i - q\|^2}{\sigma^2}} \right] - \tilde{a}_i^{\top} \mathcal{K}(q) e^{-\frac{\|p_i - q\|^2}{\sigma^2}} \\
&= \hat{\lambda}_i(q) - \tilde{a}_i^{\top} \mathcal{K}(q) e^{-\frac{\|p_i - q\|^2}{\sigma^2}},
\end{aligned}$$

where $\hat{\lambda}_i(q) := e^{-\frac{\|p_i - q\|^2}{\sigma^2}} \left[\hat{\phi}(q) - \gamma e^{-\frac{\|p_i - q\|^2}{\sigma^2}} \right]$.
Using this in equation (6.7), we get

$$\dot{V} = \frac{4\gamma}{\sigma^2} \sum_{i=1}^{N} \left\{ \int_{\mathcal{V}_i} (p_i - q)^{\top} \hat{\lambda}_i(q) dq\, \dot{p}_i + \tilde{a}_i^{\top} \int_{\mathcal{V}_i} \mathcal{K}(q) e^{-\frac{\|p_i - q\|^2}{\sigma^2}} (p_i - q)^{\top} dq\, \dot{p}_i \right\} + \sum_{i=1}^{N} \tilde{a}_i^{\top} \Gamma^{-1} \dot{\hat{a}}_i.$$

Using (6.1), the control law (6.3), adaptation law (6.4) and simplifying, we get

$$\dot{V} = -\frac{4\gamma}{\sigma^2} k_1 \sum_{i=1}^{N} \|p_i - \hat{C}_{\mathcal{V}_i}^{\lambda}\|^2 - \sum_{i=1}^{N} \tilde{a}_i^{\top} I_{\beta_i} b_i - \tilde{a}_i^{\top} \int_0^t e^{-\alpha(t-\tau)} \mathcal{K}_i(\tau) \mathcal{K}_i(\tau)^{\top} d\tau\, \tilde{a}_i.$$

It can be shown that all three terms of $\dot{V}$ above are non-positive (see the proof of Theorem 3.3). Since V is bounded below by zero and its time derivative is non-positive, it follows that $\lim_{t\to\infty} V(t)$ is finite. This implies that $\dot{V}$ is integrable. Using Barbalat's lemma, we can conclude that $\lim_{t\to\infty} \dot{V} = 0$. Statements 1 and 3 of the theorem follow immediately. Statement 2 follows from statement 1 and equation (6.3). The proof is thus complete. □

Remark 6.2. From the statements of Theorem 6.1, we also observe that $\lim_{t\to\infty} \dot{\theta}_i = 0$ for all $i = 1, 2, \ldots, N$.

Remark 6.3. The control and adaptation laws in the component form is given by

$$
\begin{aligned}
u_i &= -k_1 \left(\cos(\theta_i)(x_i - C^{\lambda}_{\mathcal{V}_i,x}) + \sin(\theta_i)(y_i - C^{\lambda}_{\mathcal{V}_i,y})\right), \\
\omega_i &= -k_1 \left(-\sin(\theta_i)(x_i - C^{\lambda}_{\mathcal{V}_i,x}) + \cos(\theta_i)(y_i - C^{\lambda}_{\mathcal{V}_i,y})\right), \\
\dot{\hat{a}}_i &= \Gamma \left(b_i - I_\beta b_i\right), \\
b_i &= -k_1 \int_{\mathcal{V}_i} \mathcal{K}(q) \left[(q_x - x_i)(x_i - C^{\lambda}_{\mathcal{V}_i,x}) + (q_y - y_i)(y_i - C^{\lambda}_{\mathcal{V}_i,y})\right] \exp\{-\|q - p_i\|^2\} dq \\
&\quad - \gamma(\Lambda_i \hat{a}_i - \lambda_i),
\end{aligned}
$$

for the $\mathcal{L}^2$ coverage case. A similar derivation for locational optimization yields

$$
\begin{aligned}
u_i &= -k_1 \left(\cos(\theta_i)(x_i - C_{\mathcal{V}_i,x}) + \sin(\theta_i)(y_i - C_{\mathcal{V}_i,y})\right), \\
\omega_i &= -k_1 \left(-\sin(\theta_i)(x_i - C_{\mathcal{V}_i,x}) + \cos(\theta_i)(y_i - C_{\mathcal{V}_i,y})\right), \\
\dot{\hat{a}}_i &= \Gamma \left(b_i - I_\beta b_i\right), \\
b_i &= -k_1 \int_{\mathcal{V}_i} \mathcal{K}(q) \left[(q_x - x_i)(x_i - C_{\mathcal{V}_i,x}) + (q_y - y_i)(y_i - C_{\mathcal{V}_i,y})\right] dq - \gamma(\Lambda_i \hat{a}_i - \lambda_i).
\end{aligned}
$$

6.2 Consensus for better parameter convergence

From the proof of Theorem 6.1 as well as Theorems 3.3 and 3.6, it can be observed that the parameter estimate $\hat{a}_i$ converges to the true value a if the matrix

$$
\lim_{t \to \infty} \int_0^t e^{-\alpha(t-\tau)} \mathcal{K}_i(\tau) \mathcal{K}_i(\tau)^\top d\tau \tag{6.8}
$$

is positive definite. In [30], a consensus term is proposed to be included in the adaptation law to improve parameter convergence. It is shown that using a consensus term in the adaptation law makes the parameter estimates of the agents converge to a common value and thus also weakens the sufficient richness condition required for parameter convergence. The modified adaptation law is given by

$$
\dot{\hat{a}}_i = \Gamma \left(b_i - I_\beta b_i\right), \tag{6.9}
$$

$$
b_i = -F_i(p_i) - \gamma(\Lambda_i \hat{a}_i - \lambda_i) - \zeta \sum_{j=1}^{N} l_{ij}(\hat{a}_i - \hat{a}_j), \tag{6.10}
$$

where $F_i(p_i)$ is the integral term in the adaptation law. The underlying graph used for consensus is the Delaunay graph where the agents sharing an edge of Voronoi partition

have the corresponding coefficients l_{ij} non-zero. In [30], the authors propose that l_{ij} be equal to the length of the shared Voronoi edge ($|\mathcal{V}_i \cap \mathcal{V}_j|$) between agent i and j. An important consequence of using consensus-based adaptation law is the following corollary [30].

Corollary 6.4 (Corollary 2, [30])**.** *Using the consensus adaptation law, in addition to the convergence of position and velocity, if the agent paths are such that*

$$\sum_{i=1}^{N} \int_{0}^{t} e^{-\alpha(t-\tau)} \mathcal{K}_i(\tau) \mathcal{K}_i(\tau)^\top d\tau$$

is positive definite, each agent's parameter estimate converges to the true value of the parameter.

The above condition is weaker since, with consensus, the positive definiteness condition is over the sum of trajectories of all agents as opposed to the individual trajectories for each of the agents.

6.2.1 Directed Consensus

It can be seen that the agents whose trajectories follow a certain path estimate certain parameters to large accuracy, whereas for other parameters, they have poor estimates. This can be observed from the adaptation term $-(\Lambda_i \hat{a}_i - \lambda_i)$ which means that the error between the measured and the estimated value of $\phi(.)$ is weighted by the corresponding regressor element $\mathcal{K}_i^{(j)}$ for updating the corresponding parameter estimate $\hat{a}_i^{(j)}$. Thus if the agent trajectory is such that the regressor element always takes a low value, then the corresponding parameter estimate is also very poor. This means that using a consensus term can sometimes reduce the accuracy and/or convergence speed of parameter estimates of those agents which are otherwise able to accurately estimate the parameter.

Based on the above observation, we propose a modified consensus law. Corresponding to each parameter $a^{(j)}$, we construct a directed sub-graph $\mathcal{G}^{(j)}(t)$ of the Delaunay graph $\mathcal{G}(t)$ as follows: a directed edge between Voronoi neighbours i and l exists if

$$\mathcal{K}_i^{(j)}(t) \geq \mathcal{K}_l^{(j)}(t). \tag{6.11}$$

The weights for the directed edges are taken as constant. This protocol creates a separate directed sub-graph of the undirected Delaunay graph corresponding to each parameter at each time t. An illustration is shown in figure 6.1.

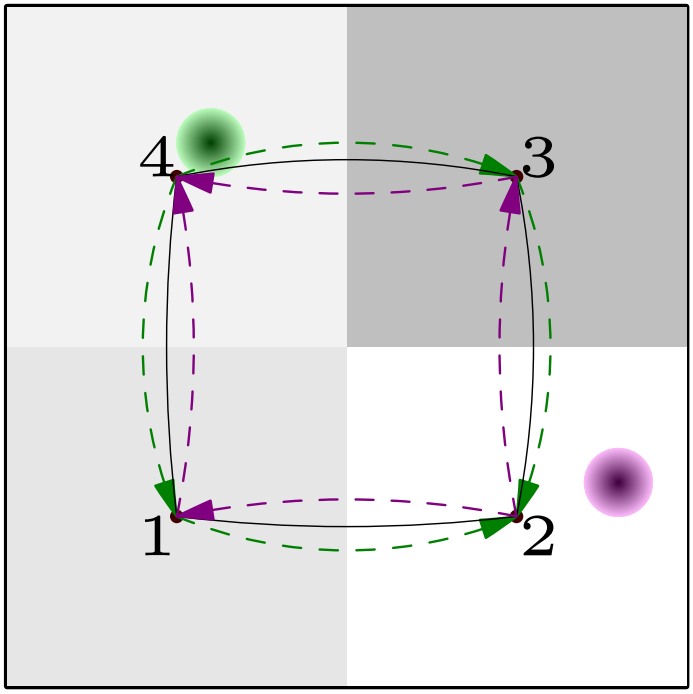
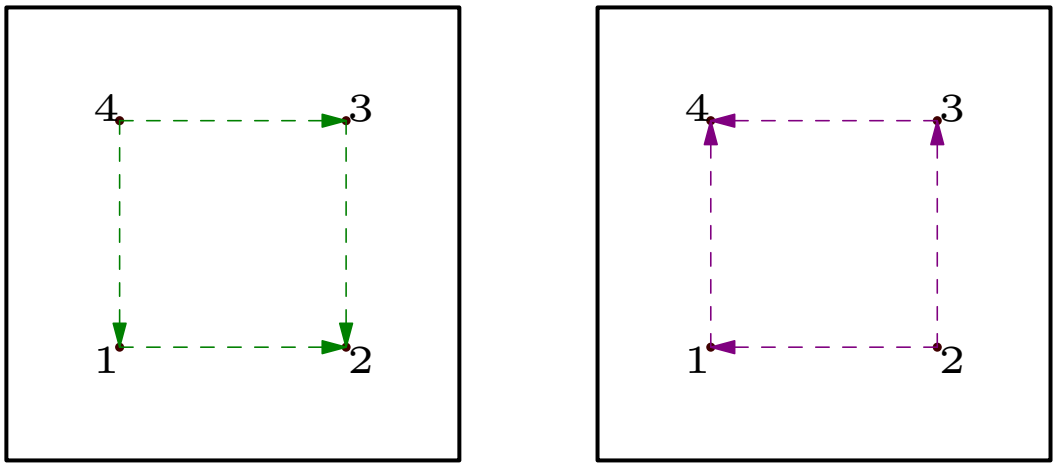


Figure 6.1: Illustration of directed graphs for consensus: 4 agents with 2 parameters - the peak of the basis functions are shown in color with the directed graph for the corresponding parameter shown in the same color; the black color shows the original Delaunay graph.

Lemma 6.1. *If the Delaunay graph $\mathcal{G}(t)$ is connected and the basis functions in $\mathcal{K}(\cdot)$ are radial functions (i.e., the functions have their peak value at some point and the value reduces with distance from that point), then the directed graphs $\mathcal{G}^{(j)}(t)$ for each j has a rooted tree. The root of the tree is the agent having the maximum value of $\mathcal{K}_i^{(j)}(t)$ among all agents $i = 1, 2, \ldots, N$.*

Proof. For any j and each pair of agents (i, l) which are Voronoi neighbours, we see from condition (6.11) that there is always a directed edge either from i to l or from l to i. Then the agent with the largest value of $\mathcal{K}_i^{(j)}(t)$, say n_j, will only have outgoing edges, and the agent with the smallest value of $\mathcal{K}_i^{(j)}(t)$ will only have incoming edges. For the illustration in Figure 6.1, the directed edges for each of the two parameters are shown seperately in Figure 6.2. Given any node (agent) l, there always exists at least one path in the Delaunay graph $\mathcal{G}(t)$ (since $\mathcal{G}(t)$ is connected) from agent n_j to l. In particular, there exists a path such that the sequence of nodes starting from n_j in the path is in decreasing order with respect to the value of $\mathcal{K}^j(\cdot)$, i.e., $\mathcal{K}^j(p_{n_j}) \geq \cdots \geq \mathcal{K}^j(p_l)$. This assertion can be proved as follows: Given node n_j, either n_j is connected to l, or it is not. If it is connected, we are done. If it is not connected, there exists another node which is at



(a) Parameter 1: 4 is the root agent.

(b) Parameter 2: 2 is the root agent.

Figure 6.2: The directed graph and edges corresponding to the parameters in Figure 6.1.

a lower distance from node n_j as compared to node l. This new node is either connected to l, or there has to exist another node which is at a smaller distance from this new node as compare to node l. This can be continued until all such nodes are exhausted. If all such nodes are exhausted, and the node l is still not connected to one of these nodes, it means that the graph is disconnected which is not possible. Thus there exists a sequence of nodes in increasing order of distance from the node n_j. This, along with the fact that the functions $\mathcal{K}^j(\cdot)$ are radial, proves the assertion. Using (6.11), it easily follows that in the directed graph $\mathcal{G}^{(j)}(t)$, there always exists a directed path from the root node n_j to any other node. This completes the proof. □

Now we modify the adaptation law with each parameter j having separate consensus law according to the directed sub-graph $\mathcal{G}^{(j)}(t)$. Thus we have the following adaptation law for agent i's parameter estimate:

$$b_i = -F_i(p_i) - \gamma(\Lambda_i \hat{a}_i - \lambda_i) - \zeta \sum_{j=1}^{N} L_{ij}(\hat{a}_i - \hat{a}_j) \tag{6.12}$$

$$= -F_i(p_i) - \gamma(\Lambda_i \hat{a}_i - \lambda_i) - \sum_{\alpha=1}^{p} \zeta \sum_{j=1}^{N} l_{ij}^{\alpha} \left(\hat{a}_i^{(\alpha)} - \hat{a}_j^{(\alpha)} \right), \tag{6.13}$$

where $L_{ij} = \text{diag}(\{l_{ij}^{\alpha}\}_{\alpha=1}^{p})$ and $L^{\alpha} = \left[l_{ij}^{\alpha}\right]$ is the Laplacian matrix for graph $\mathcal{G}^{(\alpha)}(t)$.

Theorem 6.5. *Using the modified adaptation law 6.13, and assuming that all the other conditions of Theorem 6.1 hold, all the statements of the Theorem 6.1 hold. In addition*

$$\lim_{t \to \infty} (\hat{a}_i(t) - \hat{a}_j(t)) = 0, \tag{6.14}$$

for all $i, j \in \{1, 2, \ldots, N\}$.

Proof. Proceeding the same way as in the proof of Theorem 6.1, we have an additional term in the derivative of the Lyapunov function $\dot{V}$, all other terms remaining exactly the same. The term is given by $-\sum_{i=1}^{N} \tilde{a}_i^\top \sum_{j=1}^{N} L_{ij}(\hat{a}_i - \hat{a}_j)$. Simplifying this,

$$\begin{aligned}
\sum_{i=1}^{N} \tilde{a}_i^\top \sum_{j=1}^{N} L_{ij}(\hat{a}_i - \hat{a}_j) &= \sum_{i=1}^{N} \left[\tilde{a}_i^{(1)} \tilde{a}_i^{(2)} \ldots \tilde{a}_i^{(p)}\right] \sum_{j=1}^{N} L_{ij} \begin{bmatrix} \hat{a}_i^{(1)} - \hat{a}_j^{(1)} \\ \hat{a}_i^{(2)} - \hat{a}_j^{(2)} \\ \vdots \\ \hat{a}_i^{(p)} - \hat{a}_j^{(p)} \end{bmatrix} \\
&= \sum_{\alpha=1}^{p} \sum_{i=1}^{N} \sum_{j=1}^{N} \tilde{a}_i^{(\alpha)} l_{ij}^{\alpha} (\hat{a}_i^{(\alpha)} - \hat{a}_j^{(\alpha)}) \\
&= \sum_{\alpha=1}^{p} \left\{ \sum_{i=1}^{N} \tilde{a}_i^{(\alpha)} \left[l_{i1}^{\alpha} (\hat{a}_i^{(\alpha)} - \hat{a}_1^{(\alpha)}) + l_{i2}^{\alpha} (\hat{a}_i^{(\alpha)} - \hat{a}_2^{(\alpha)}) + \cdots + l_{iN}^{\alpha} (\hat{a}_i^{(\alpha)} - \hat{a}_N^{(\alpha)}) \right] \right\} \\
&= \sum_{\alpha=1}^{p} \left[\tilde{a}_1^{(\alpha)} \tilde{a}_2^{(\alpha)} \ldots \tilde{a}_N^{(\alpha)}\right] \begin{bmatrix} l_{12}^{\alpha} + l_{13}^{\alpha} + \cdots + l_{1N}^{\alpha} & \ldots & \ldots & -l_{1N}^{\alpha} \\ -l_{21}^{\alpha} & \ldots & \ldots & -l_{2N}^{\alpha} \\ \vdots & \vdots & \vdots & \vdots \\ -l_{N1}^{\alpha} & \ldots & \ldots & \vdots \end{bmatrix} \begin{bmatrix} \hat{a}_1^{(\alpha)} \\ \hat{a}_2^{(\alpha)} \\ \vdots \\ \hat{a}_N^{(\alpha)} \end{bmatrix} \\
&= \sum_{\alpha=1}^{p} \tilde{a}^{\alpha\top} L^{\alpha} \hat{a}^{\alpha} = \sum_{\alpha=1}^{p} \hat{a}^{\alpha\top} L^{\alpha} \hat{a}^{\alpha},
\end{aligned}$$

where $\hat{a}^{\alpha} = \left[\hat{a}_1^{(\alpha)}\ \hat{a}_2^{(\alpha)}\ \ldots \hat{a}_N^{(\alpha)}\right]^\top$.

Thus the term contributed by the consensus term in $\dot{V}$ is non-positive. The term is also uniformly continuous. The other terms in $\dot{V}$ remain non-positive and uniformly continuous, as in the proof of Theorem 6.1. Thus using Barbalat's lemma, all the statements of the Theorem 6.1 holds. In addition, we have that $\lim_{t\to\infty} \hat{a}^{\alpha\top} L^{\alpha} \hat{a}^{\alpha} = 0$ for each α. Since L^{α} is the Laplacian matrix of the directed graph $\mathcal{G}^{(\alpha)}$, we have that $\lim_{t\to\infty} \hat{a}^{\alpha} = c_{\alpha}\mathbf{1}$ for each α (c_{α} is a positive constant), i.e., the agents achieve consensus on the parameter values and the theorem holds. □

6.3 Experiment Setup

6.3.1 Differential Drive Robots

The robots used are custom-built differential drive robots based on the Turtlebot3 Burger[1] platform. Each robot consists of two rear wheels with motor drivers, and a castor wheel in front. The controller board OpenCR 1.0 is used along with a Raspberry Pi 3 [2] module for controlling each robot. The robots communicate to each other and the host PC via WLAN.

6.3.2 Workspace and Localization System

The workspace where the robots move is a flat 4m×4m square region. For localization of robots, we use the motion-capture system from OptiTrack[3]. The system comprises of 16 cameras with infrared sensors that detect the markers fixed atop the robots. A proprietary software (Motive, by Optitrack) uses data in the form of images captured by the cameras, performs localization computations, and provides position data for all the robots in the workspace. This data is streamed over the local network using the Virtual Reality Peripheral Network [4] (VRPN) protocol. The localization system provides millimeter-level precision at high frequencies up to 200 Hz. See figure 6.3 for the overall setup.

6.3.3 Sensors

The density function is implemented as a light distribution using Xiaomi smart bulbs[5]. The Adafruit TCS34725 light sensors[6] are used for measuring the light intensity.

6.3.4 Host PC

The robots communicate with the host PC via WLAN. ROS (Robot Operating System) is used for the software implementation. Each robot runs multiple ROS nodes

[1] http://www.robotis.us/turtlebot-3/
[2] https://www.raspberrypi.org/products/raspberry-pi-3-model-b/
[3] https://optitrack.com
[4] https://github.com/vrpn/vrpn/wiki
[5] https://www.mi.com/in/mi-led-smart-bulb/
[6] https://www.adafruit.com/product/1334

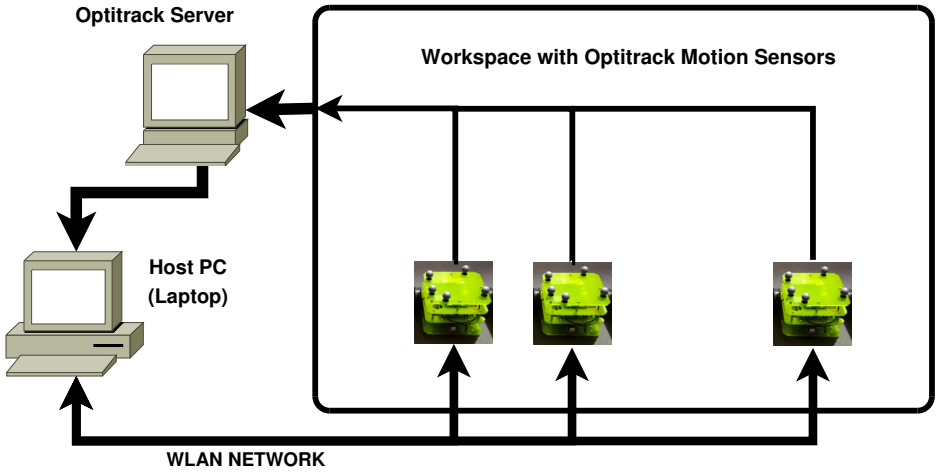


Figure 6.3: Experiment Setup

which implement the coverage algorithm, receives localization data from the Optitrack system as well as communicate with other robots. The host PC runs the ROS master node and subscribes to the position data from the localization system (using VRPN protocol), which are then distributed to the individual robots. The ROS nodes launched by the individual robots connects to the ROS master of the host PC. Figure 6.4 shows the overall software implementation using ROS.

6.4 Experiment Results

In this section, we discuss the experimental results obtained. We consider two types of coverage control laws for the robots: (1) based on the locational optimization cost function, and (2) based on the $\mathcal{L}^2$-distance cost function. In addition, we do two sets of experiments: (1) The density function is simulated, and (2) the density function is implemented using white light sources, and the agents measure the same using RGB sensors. In the case of simulated density function, the density function is implemented in the software, and this case allows us to study the performance of the coverage algorithm and parameter convergence in better detail since there is no sensor noise and associated issues. Using actual sources and sensors allows us to evaluate how well the algorithms behave in the real world with noisy measurements. The values of various constants used in the simulation are given in Table 6.1. The parameters to be estimated correspond to the strengths of the individual light sources (parameter a in table **??**). Video recordings of the experiments are available at: `https://www.youtube.com/playlist?list=`

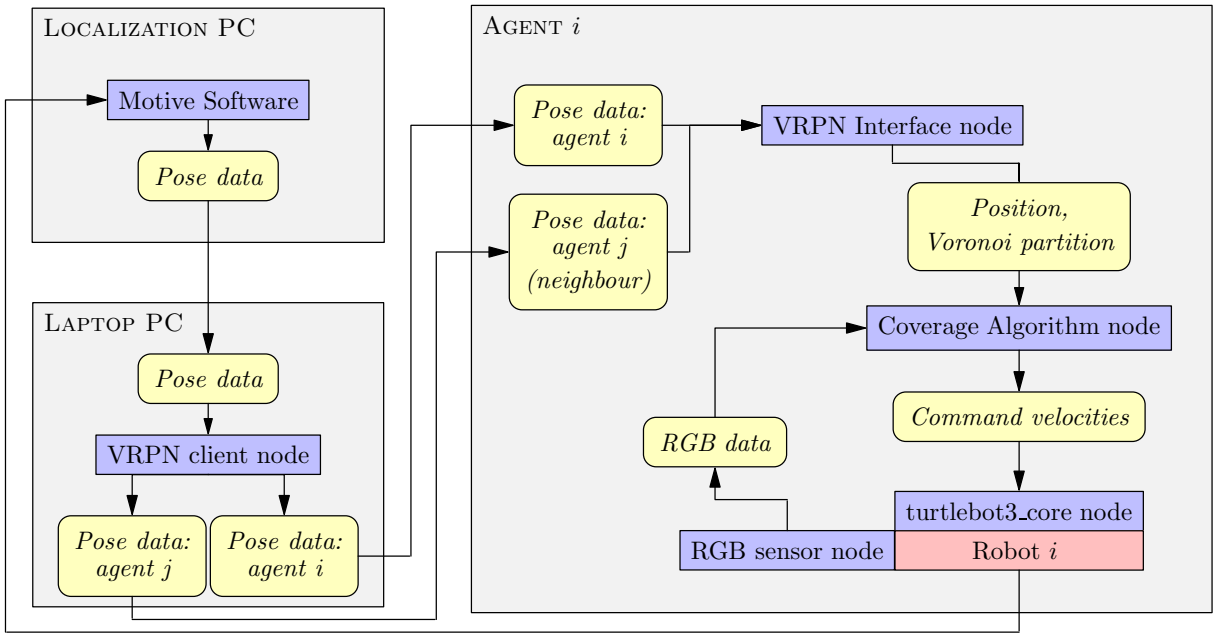


Figure 6.4: ROS Implementation: the blue blocks represent ROS nodes and the yellow blocks represent the data exchanged between nodes.

`PLH0udZHyB-e2A9gFZbvdLXygOIFZa6Ynv`.

6.4.1 Simulated Density Function

The true density function consists of two Gaussian components. The various constants related to the density function are given in Table 6.1. The trajectories, average position error, and the average velocity of the agents are shown in Figure 6.5. The average position and velocity errors are given by

$$e_p = \sum_{i=1}^{N} \|p_i - \hat{C}_{\mathcal{V}_i}\|, \qquad e_v = \sum_{i=1}^{N} \|v_i\|.$$

The corresponding plots for the locational optimization-based coverage are also shown in the figure for comparison. The initial position error and velocity are higher for the locational optimization case. The agent parameter estimation errors are compared in Figures 6.6 and 6.7 for the two parameters. It can be seen that for three of the agents, the estimation errors for Parameter 1 drops significantly from the initial value, with one agent able to estimate Parameter 1 accurately. The other two agents are not able to adapt for Parameter 1. Similarly, two of the agents are able to estimate Parameter 2 better while for the rest of the agents, Parameter 2 estimation error barely changes from the initial estimation error.

Parameter	Value	Description
Domain related		
borderx	[-2.0, 2.0, 2.0, -2.0]	x-coordinates of vertices of the domain
bordery	[-2.0, -2.0, 2.0, 2.0]	y-coordinates of vertices of the domain
Density function related		
centrex	[1.0, 1.0]	x-coordinates of centres of density fcn.
centrey	[0.98, -0.8]	y-coordinates of centres of density fcn.
σ	[0.6, 0.3]	std. deviation
a	[85, 30]	true strengths to be estimated
Control and Adaptation gains		
k_1	0.1	controller gain
Γ	$0.1I$	adaptation gain matrix
Loop rates		
Control loop	10 Hz.	rate at which control loop runs
Position update loop	20 Hz.	rate at which position data is available
Robot related		
d	0.05 m	distance of the centre from wheel axis
Adaptation law related		
paramInitValue	10	initial value of parameter estimates
α	1.0	filter parameter
γ	2	measurement update gain
ζ	1	consensus related gain

Table 6.1: Experiment related parameters

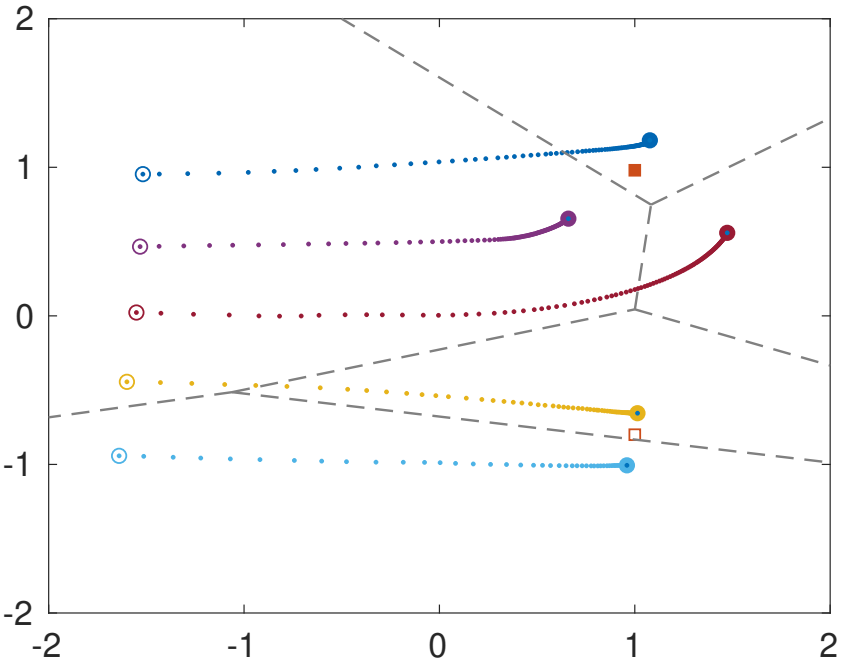
It also appears that the $\mathcal{L}^2$ algorithm performs slightly better from Figure 6.6. This could be due to the fact that the integral term in the $\mathcal{L}^2$ adaptation law (6.4) is much smaller (due to the presence of the exponential term) than for the locational optimization adaptation law (3.31). This term can be viewed as a coupling term between the coverage task (through the cost function $d_2(\cdot)$) and the estimation task. The term being smaller means that the estimate $\hat{a}_i$ is better able to adapt through the measured error in $\phi(\cdot)$ given by the second term in the adaptation law (6.4).

The parameter errors for adaptation with the consensus terms are shown in Figures 6.8 and 6.9. We show the average parameter estimation errors across all the agents for ease of comparison since the agent estimation errors closely follow each other due to the consensus term. From the plots, we see that overall, the parameter estimation errors start dropping faster in the locational optimization case, although towards the end, the drop in error becomes slower compared to the $\mathcal{L}^2$ case. This could be due to the fact that the initial velocity is higher for the locational optimization case, and thus, it is able to initially move faster to regions where significant measurements are available for adaptation.

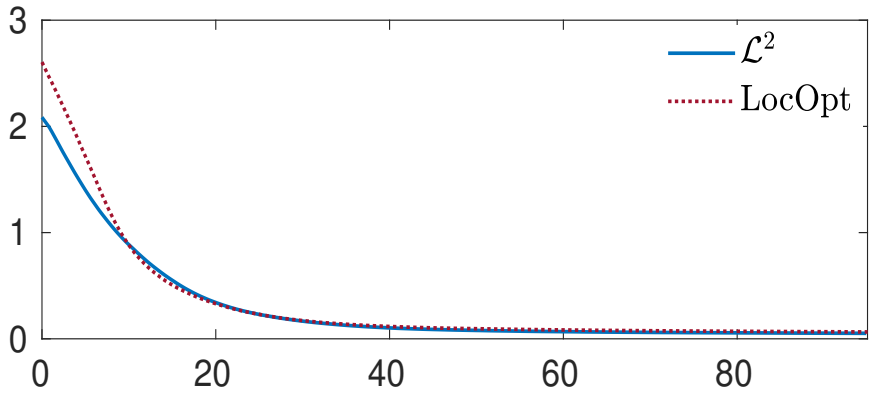
Overall, the final values of parameter estimates are slightly better for the $\mathcal{L}^2$ case. It can also be seen that the directional consensus algorithm leads to a significantly faster convergence as opposed to the undirected consensus algorithm.

6.4.2 Density Function implemented using Light Sources

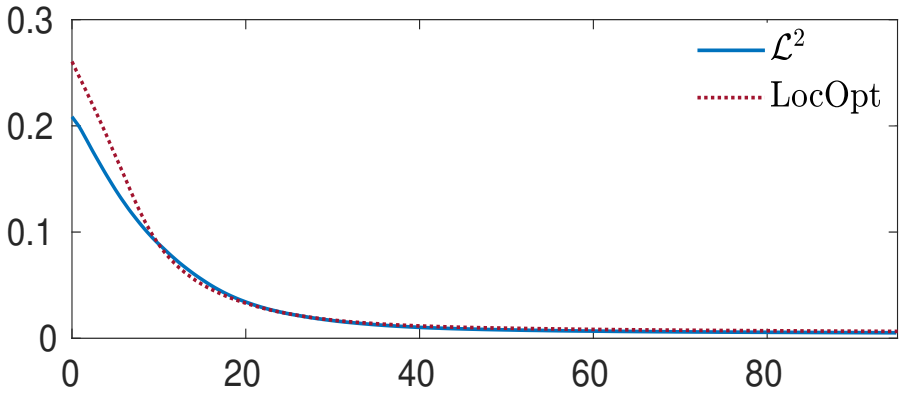
There are two light sources, one of high intensity and the other of lower intensity. Each agent is equipped with a TCS34725 RGB sensor to measure the light intensity. The measurements were observed to be quite noisy, as shown in figure 6.10.



(a) Trajectories: the Voronoi partition corresponding to the final positions is shown in dashed grey.

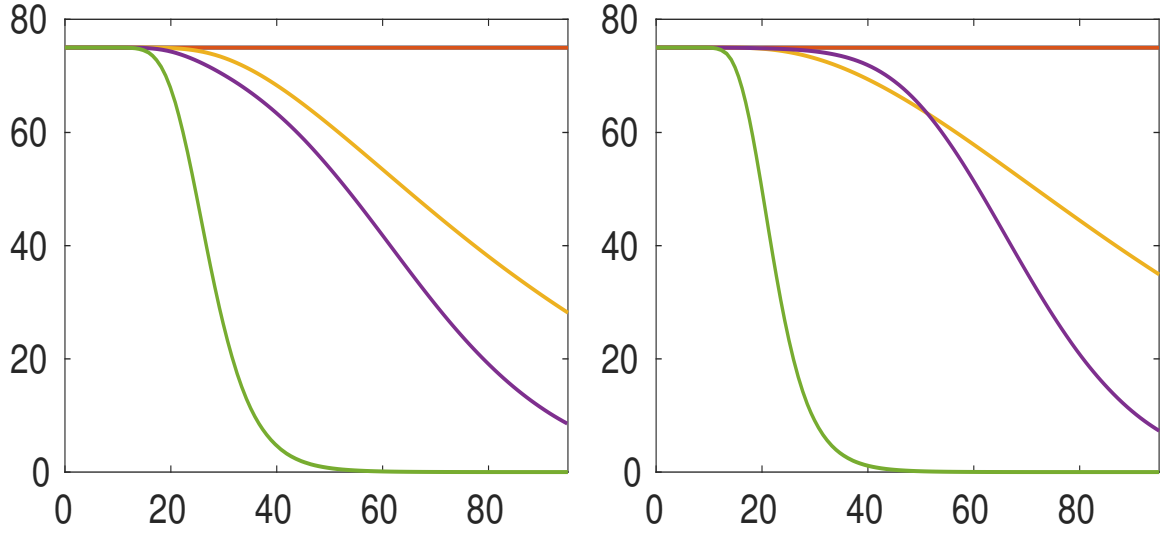


(b) Avg. Position error vs Time (sec).



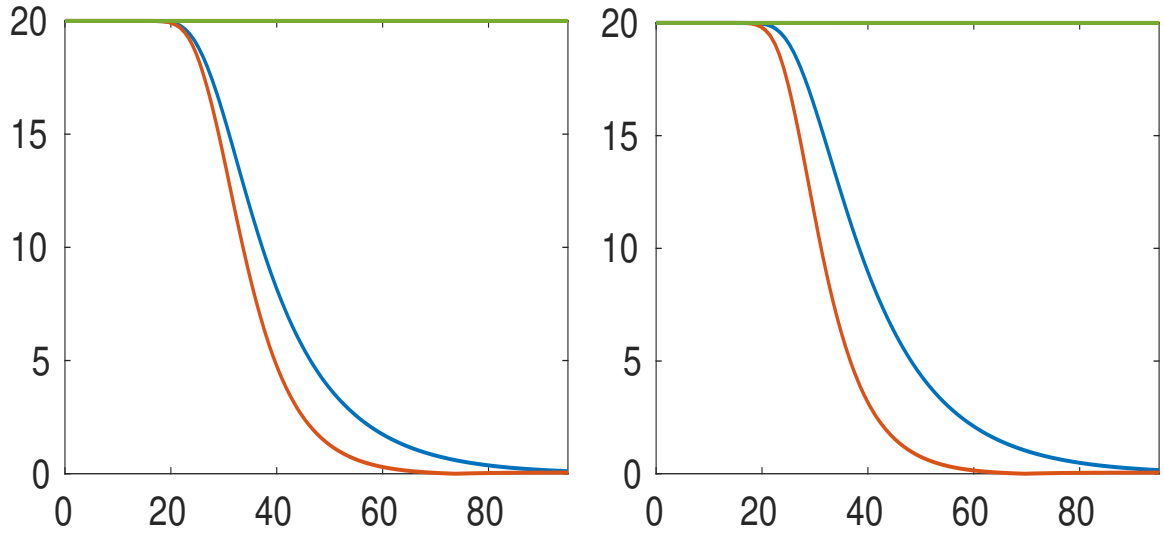
(c) Avg. Velocity vs Time (sec).

Figure 6.5: Results for $\mathcal{L}^2$ coverage: simulated $\phi(\cdot)$.



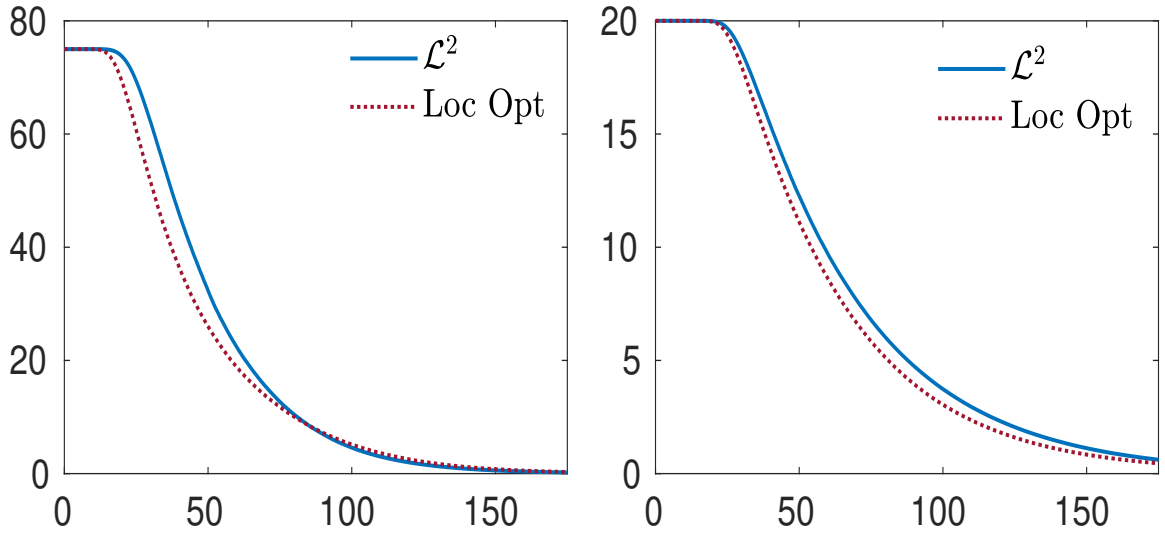
(a) $\mathcal{L}^2$ coverage (b) LocOpt coverage

Figure 6.6: Simulated $\phi(\cdot)$: Parameter 1 estimation errors with time - No consensus.



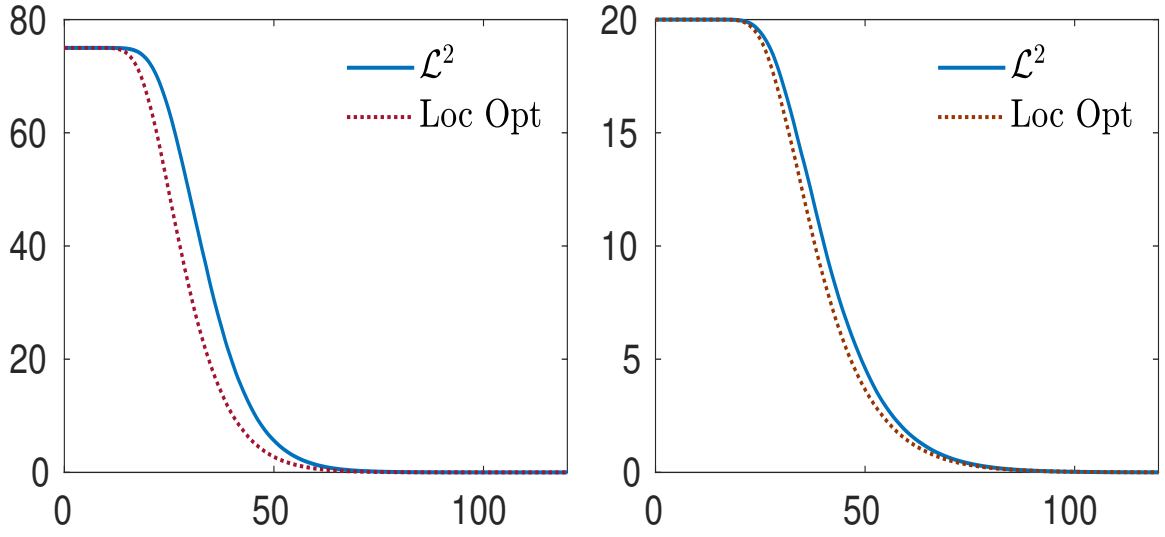
(a) $\mathcal{L}^2$ coverage (b) LocOpt coverage

Figure 6.7: Simulated $\phi(\cdot)$: Parameter 2 estimation errors with time - No consensus.



(a) Parameter 1 (b) Parameter 2

Figure 6.8: Simulated $\phi(\cdot)$: Avg. parameter estimation errors with time - with consensus.



(a) Parameter 1 (b) Parameter 2

Figure 6.9: Simulated $\phi(\cdot)$: Avg. parameter estimation errors with time - with directed consensus.

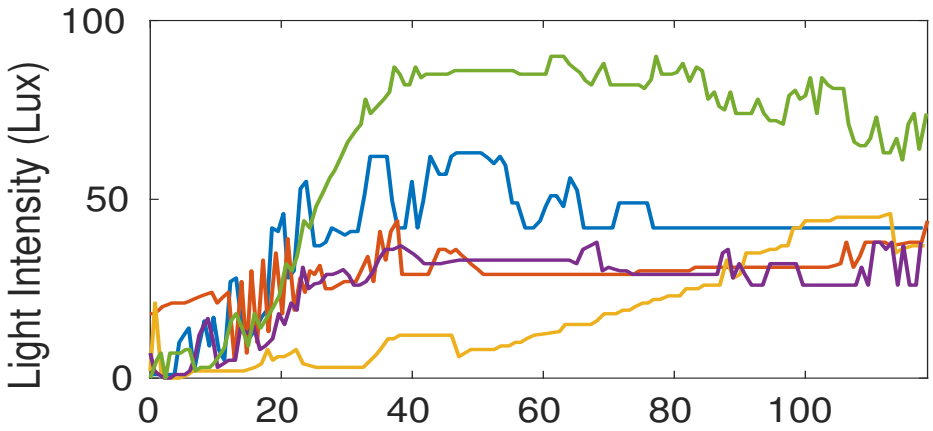
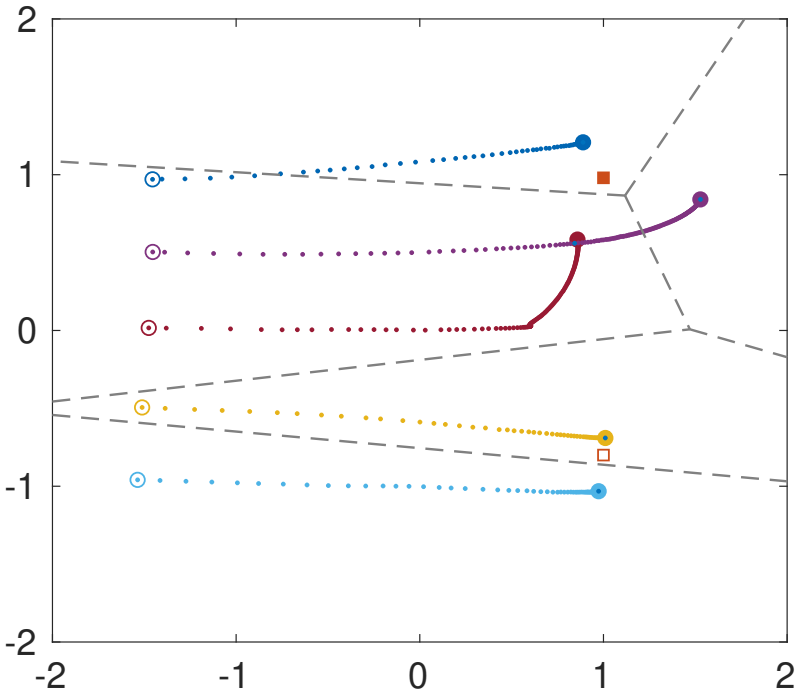
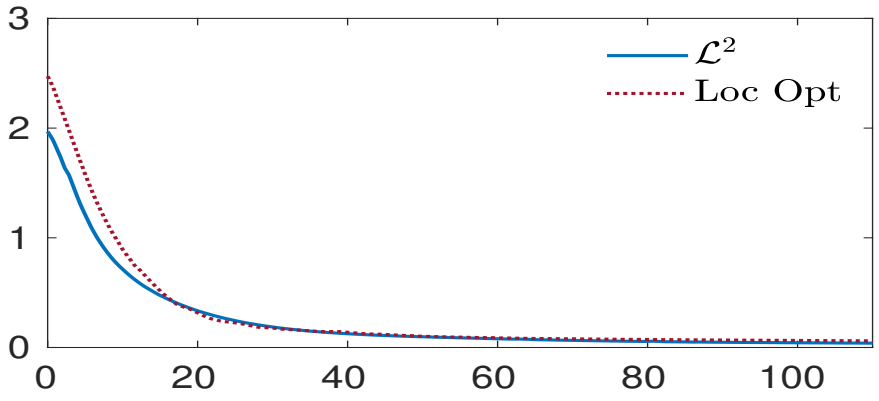


Figure 6.10: Measured signal from various sensors with time

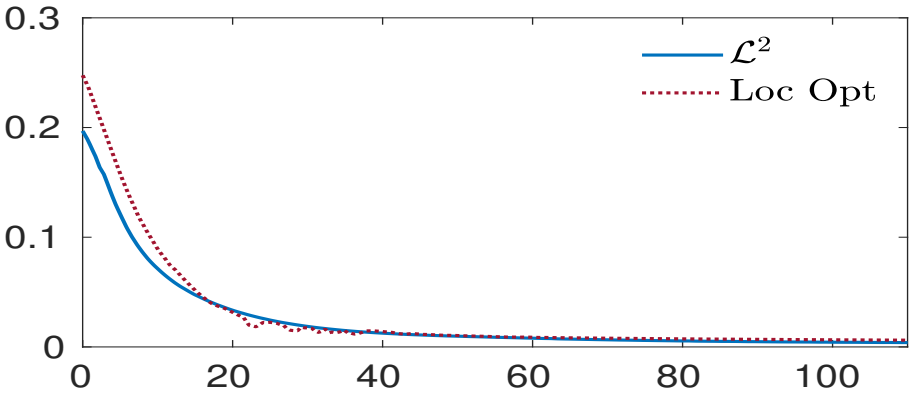
Therefore, the error plots shown in this section were obtained by averaging across 2-3 runs. It can be observed that the plots are less smooth as compared to the simulated density function case due to the noisy sensors. The trajectories, average position error, and the average velocity of the agents for $\mathcal{L}^2$ coverage are shown in Figure 6.11. The plots also show the results of locational optimization-based coverage for comparison. As with the results for simulated density function, we see that the initial position error and velocity are larger for the locational optimization case. The agent parameter estimation errors are compared in Figures 6.12 and 6.13 for the two parameters with no consensus term in the adaptation law. The parameter errors for adaptation with the consensus terms are shown in Figures 6.14 and 6.15. Overall the parameter estimates using the $\mathcal{L}^2$ coverage framework seems to be more accurate. It can be seen in Figure 6.14 that the parameter error drops to zero and then increases. This is because the algorithm does not guarantee that parameter errors go to zero, and the sensor noise may perturb the parameter estimate and make it settle at a higher value, though this value of estimate guarantees statement (3) of Theorem 6.1 to hold. For the noise-free simulated density function case in Figure 6.8, it is seen that the errors decrease monotonically. It can also be seen from the plots that the directed consensus leads to faster convergence of the parameter errors as expected.



(a) Trajectories: the Voronoi partition corresponding to the final positions is shown in dashed grey.

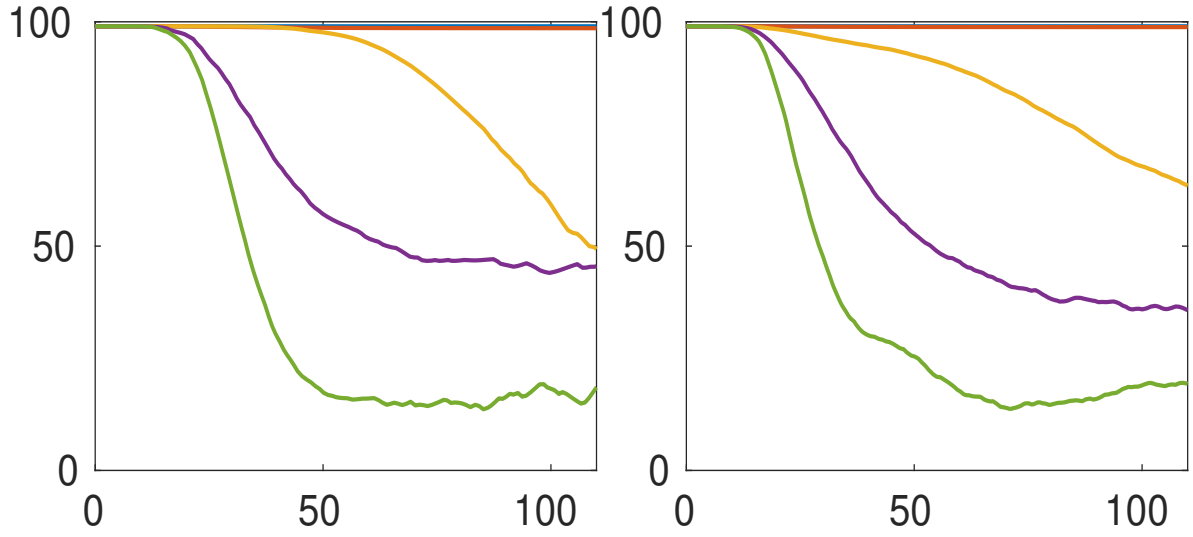


(b) Avg. Position error vs Time (sec).



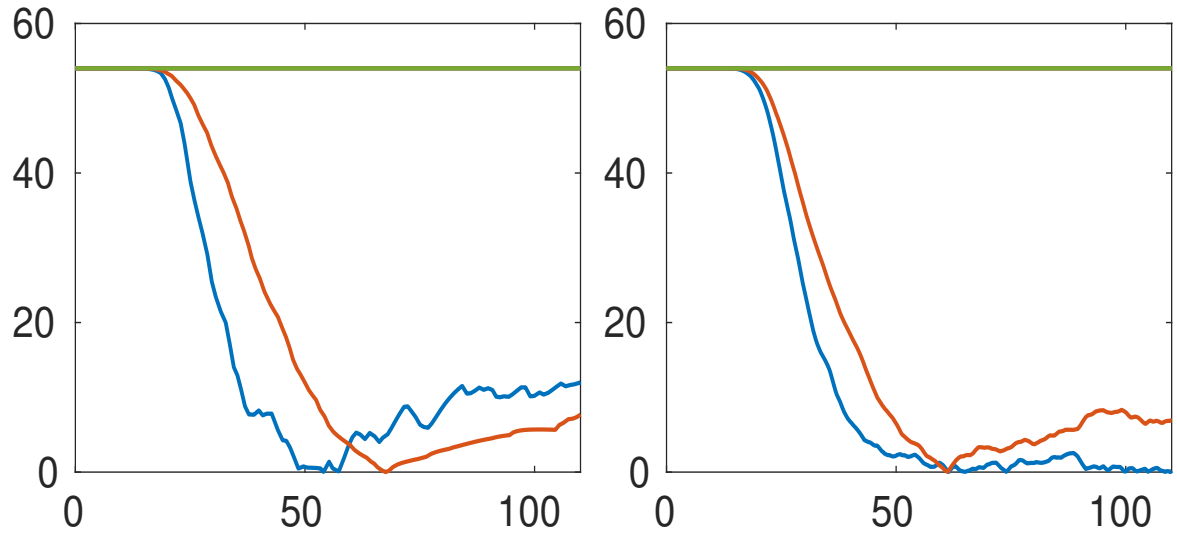
(c) Avg. Velocity vs Time (sec).

Figure 6.11: Results for $\mathcal{L}^2$ coverage: RGB sensors.



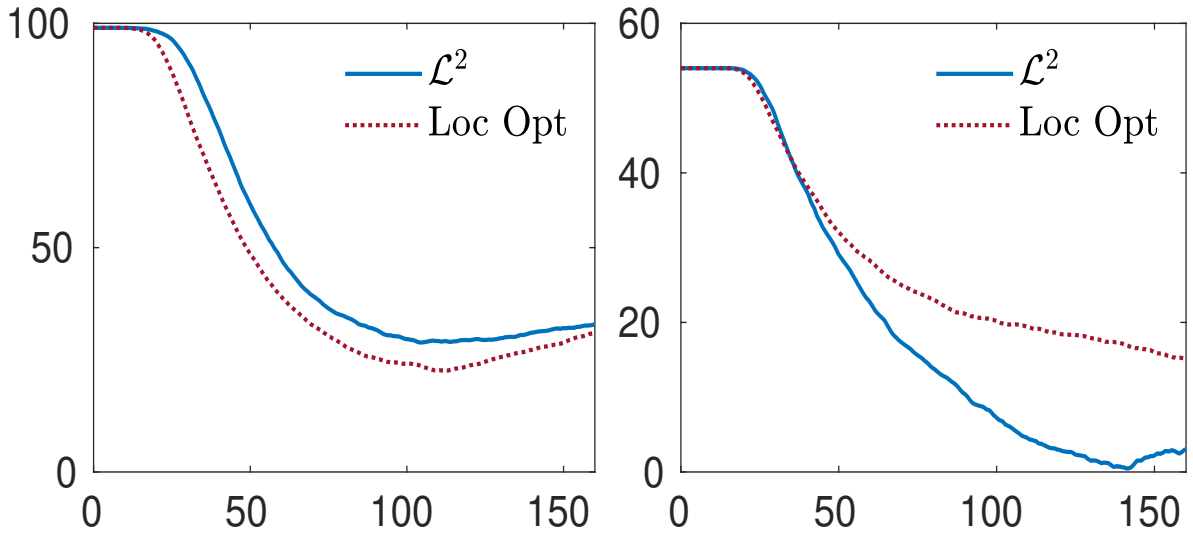
(a) $\mathcal{L}^2$ coverage (b) LocOpt coverage

Figure 6.12: RGB sensors: Parameter 1 estimation errors with time - No consensus.



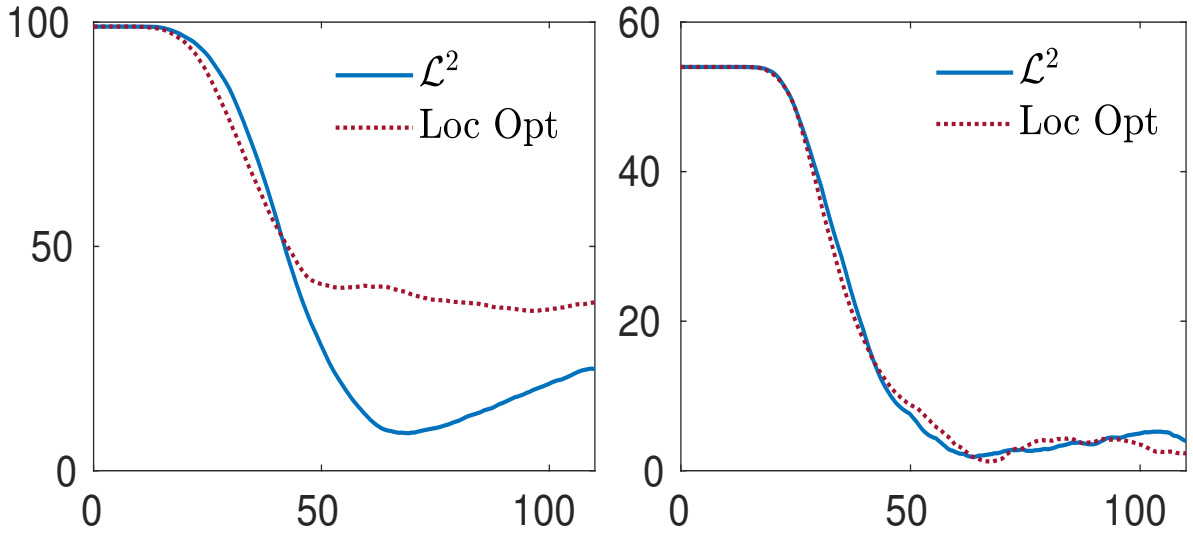
(a) $\mathcal{L}^2$ coverage (b) LocOpt coverage

Figure 6.13: RGB sensors: Parameter 2 estimation errors with time - No consensus.



(a) Parameter 1 (b) Parameter 2

Figure 6.14: RGB sensors: Avg. parameter estimation errors with time - with consensus.



(a) Parameter 1 (b) Parameter 2

Figure 6.15: RGB sensors: Avg. parameter estimation errors with time - with directed consensus.

Chapter 7

Estimating the Density Function

In this chapter, we look at a slightly different problem closely related to the coverage problem we looked at in the previous chapters. The problem is the accurate estimation of the density function. In the coverage problem, we were interested in controlling the robots so that the robots attain an optimal configuration (or a near-optimal configuration) with respect to the density function. In order to achieve this goal, the robot needs to adapt the density function parameters so that the density function is close to the actual density function. As discussed in section 6.2, the exact estimation (asymptotically) of the density function parameters require a time integral quantity (given by equation (6.8)) to be positive definite, which is a sufficient richness condition for the robot trajectories. See [30] for more details. In general, the robot trajectories need not meet this condition since the trajectories of the robots are decided based primarily on the gradient of the coverage cost function and not on estimating the density function parameters. However, it is crucial to estimate the true values of those parameters since the estimation of the unknown scalar field is often the primary objective for a robotic sensor network, and it may lead to more efficient deployment of robots. For example, in the case of radiation spill, if we have a good estimate of the radiation concentration, we may directly deploy agents to regions of high concentration. Thus in this chapter, we look at a slightly different problem closely related to and motivated by the coverage problem discussed above. Our primary aim in this chapter is to accurately estimate the scalar field, not coverage. The unknown scalar field is approximated using positive definite radial basis functions and a similar adaptive approach as that in chapter 3 is adopted for parameter estimation while ensuring the positive-definiteness of the time integral (6.8). Compared to previous works

in literature on scalar field estimation (see section 2.3), we provide theoretical guarantees for the convergence and boundedness of parameter estimates when the scalar field can be exactly parameterized using RBFs.

7.1 Problem Statement

We consider a compact region $\mathcal{Q} \subset \mathbb{R}^n$ with N mobile sensors. The position of the sensors is denoted by $p_i;\ i = 1, 2, \ldots, N$. There also exists a continuous scalar field $\phi : \mathcal{Q} \to \mathbb{R}_+$ over $\mathcal{Q}$, which is unknown. The objective is to estimate the unknown scalar field using N mobile sensors assuming the sensors can measure the value of the scalar field at their respective positions. We assume that the unknown scalar field can be represented by positive definite radial basis functions (RBFs). In other words, we assume the density function can be parameterized as

$$\phi(q) = \mathcal{K}(q)^\top a \tag{7.1a}$$

$$= \sum_{i=1}^{m} \mathcal{K}^i(q) a^i, \tag{7.1b}$$

where $a \in \mathbb{R}^m$ is a constant vector, and $\mathcal{K}(q) = [\mathcal{K}^1(q)\ \mathcal{K}^2(q)\ \ldots\ \mathcal{K}^m(q)]^\top$ with $\mathcal{K}^i : \mathcal{Q} \to \mathbb{R}_+$ given by $\mathcal{K}^i(q) = \varphi(\|c_i - q\|)$ are radial basis functions for a set of points c_i. This assumption is common in neural networks and justified as follows:

Theorem 7.1 ([55, 56])**.** *For any continuous function $f(q)$ and any $\epsilon > 0$, there is an RBF network with p elements, a set of centers $\{c_i\}_{i=1}^m$, such that we can define*

$$\hat{f}(q) = \sum_{i=1}^{m} a^i \mathcal{K}^i(q) = a^\top \mathcal{K}(q),$$

with $\|f - \hat{f}\|_{L_2}^2 \leq \epsilon = \mathcal{O}\left(p^{-\frac{1}{n}}\right)$.

The theorem tells us that we can approximate a continuous function to an arbitrary accuracy by using a network of RBF elements. An example of positive definite radial kernel is the Gaussian kernel,

$$\mathcal{K}^i(q) = \varphi(\|c_i - q\|) = \exp\left\{-\frac{\|c_i - q\|^2}{\sigma_i^2}\right\}, \tag{7.2}$$

where c_i are the centres of the Gaussian kernels. The main problem studied in this work is to accurately determine the parameters a^i so that the scalar field $\phi(.)$ may be accurately reconstructed. We make the following assumption:

Assumption 7.2. The centers c_i of the radial functions are known to all the mobile agents.

The strengths a_i of individual radial functions are unknown and need to be estimated. To proceed, we require the following theorem:

Theorem 7.3 (Micchelli's Theorem [57])**.** *Given p distinct points $c_1, c_2, \ldots, c_p$ in $\mathbb{R}^n$, the $p \times p$ matrix* K*, whose elements are $\mathrm{K}_{ij} = \mathcal{K}^i(c_j) = \varphi(\|c_i - c_j\|)$ is non-singular.*

The theorem says that for positive definite radial kernels, the $p \times p$ matrix formed by evaluating the radial functions at each of the centers is non-singular. In what follows, we assume that $\phi(.)$ can be *exactly parameterized* by the RBF kernels. A consequence of Theorem 7.3 is given below:

Lemma 7.1. *Given that ϕ is parameterized as in (7.1a), the matrix S given by*

$$S := \int_{\mathcal{Q}} \mathcal{K}(q)\mathcal{K}(q)^\top dq, \tag{7.3}$$

where $\mathcal{K}(q) = [\mathcal{K}^1(q) \;\; \mathcal{K}^2(q) \;\; \ldots \;\; \mathcal{K}^p(q)]^\top$ is positive definite.

Proof. From the definition of S, we know it is at least positive semi-definite. Therefore for any $v \neq 0$, $v^\top S v \geq 0$ or

$$\int_{\mathcal{Q}} |\mathcal{K}(q)^\top v|^2 dq \geq 0.$$

Now, since $\mathcal{K}(q)$ consists of positive definite radial kernels, we have from Theorem 7.3 that

$$\begin{pmatrix} \mathcal{K}^1(c_1) & \mathcal{K}^1(c_2) & \ldots & \mathcal{K}^1(c_p) \\ \mathcal{K}^2(c_1) & \mathcal{K}^2(c_2) & \ldots & \mathcal{K}^2(c_p) \\ \ldots & \ddots & \ldots & \vdots \\ \mathcal{K}^p(c_1) & \mathcal{K}^p(c_2) & \ldots & \mathcal{K}^p(c_p) \end{pmatrix}$$

is positive definite. This implies that the vectors $\mathcal{K}(c_j);\ j = 1, 2, \ldots, p$ are linearly independent. Thus, given any $v \neq 0, v \in \mathbb{R}^p$, there exists some $j \in \{1, 2, \ldots, p\}$ such that $\mathcal{K}(c_j)^\top v$ is non-zero. This along with the fact that $\mathcal{K}(\cdot)$ is continuous allows us to conclude that

$$\int_{Q} |\mathcal{K}(q)^\top v|^2 dq > 0, \quad \text{for any } v \neq 0.$$

Hence, S is positive definite. □

7.2 Single Mobile Robot Sensor

In this section, we consider the case of a single mobile sensor ($N = 1$) with position $p(t)$ at time t deployed in the region $\mathcal{Q}$ to estimate the scalar field parameter a. Then we can state the following corollary to Lemma 7.1.

Corollary 7.4. *Suppose the mobile sensor moves continuously within the domain Q, such that in time T, it passes through each of the RBF centers c_i ; $i = 1, 2, \ldots, m$, then*

$$\mathcal{S}_T := \int_0^T \mathcal{K}(p(t))\mathcal{K}(p(t))^\top dt \tag{7.4}$$

is positive definite.

Proof. The proof is essentially the same and follows from Lemma 7.1. □

Now consider the following integrators running on the mobile sensor:

$$\begin{aligned} \dot{\Lambda} &= \mathcal{K}(t)\mathcal{K}(t)^\top, \\ \dot{\lambda} &= \mathcal{K}(t)\phi(t), \end{aligned} \tag{7.5}$$

where $\mathcal{K}(t) := \mathcal{K}(x(t))$ denotes the value of function $\mathcal{K}(\cdot)$ at the point where the robot is at time t and $\phi(t)$ is the measured value of the density function $\phi(\cdot)$ by the robot at time t.

Proposition 7.1. *Suppose the mobile sensor moves such that it passes through each of the centres c_i; $i = 1, 2, \ldots, m$ in some finite time $T > 0$, and during this motion updates its estimate $\hat{a}$ of a by*

$$\dot{\hat{a}} = -\Gamma \left(\Lambda \hat{a} - \lambda\right), \tag{7.6}$$

where Γ is a positive definite gain matrix, then the estimate $\hat{a}$ is bounded and converges asymptotically to the true value a.

Proof. Under the given assumptions and Corollary 7.4,

$$S(T) := \int_0^T \mathcal{K}(\tau)\mathcal{K}(\tau)^\top d\tau$$

is positive definite. This implies that

$$S(t) = \int_0^t \mathcal{K}(\tau)\mathcal{K}(\tau)^\top d\tau$$

is positive definite for all $t \geq T$.

Now consider the positive definite candidate Lyapunov function,

$$V = \frac{1}{2}\tilde{a}^\top \Gamma^{-1}\tilde{a}, \tag{7.7}$$

where $\tilde{a} = \hat{a} - a$ is the estimation error. Taking the derivative of V, we obtain

$$\dot{V} = \tilde{a}^\top \Gamma^{-1}\dot{\hat{a}}.$$

Substituting the update law from (7.6) and simplifying, we get

$$\begin{aligned}\dot{V} &= -\tilde{a}^\top S(t)\tilde{a} \\ &\leq \begin{cases} 0 & \text{for } 0 \leq t < T, \\ -\alpha V & \text{for } t \geq T, \end{cases}\end{aligned}$$

where $\alpha = \frac{\lambda_{\min}(S(T))}{\lambda_{\max}(\Gamma^{-1})} > 0$, $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ denoting the minimum and maximum eigenvalues of their argument. Since V is always non-increasing and bounded from below, $\tilde{a}(t)$ is bounded for all $t > 0$. Since $\dot{V} < 0$ for all $t \geq T$, then we have $V(t) \to 0$ as $t \to \infty$. This implies that $\tilde{a} \to 0$ as $t \to \infty$. □

Remark 7.5. The matrix $S(t)$ being positive definite for all $t \geq T$ is a *sufficient excitation* condition, similar to (but weaker than) the persistency of excitation condition, on the robot trajectories which ensures parameter convergence. See [30] for more information.

7.2.1 Relaxing the condition in Corollary 7.4

In Corollary 7.4, it was required that the mobile sensor passes through the centers c_i of the radial kernels. This can be relaxed so that the mobile sensor need only move through a sufficiently small neighbourhood of each of the centers c_i, as described in [58]. Consider the vector $\mathcal{X}(q) := \mathrm{K}^{-1}\mathcal{K}(q)$, where K is the matrix specified in Theorem 7.3. Then $\mathcal{X}(q)$ has the property that $\mathcal{X}^j(c_k) = \delta_{jk}$ where δ_{jk} is the Kronecker delta function and $\mathcal{X}^j(c_k)$ is the j-th component of $\mathcal{X}(c_k)$. Now consider the diagonal dominance sets defined by $(0 < \varepsilon < 1)$

$$\mathcal{A}_j^\varepsilon := \left\{ q \in \mathcal{Q} \,:\, |\mathcal{X}^j(q)| - \sum_{i=1, i\neq j}^{m} |\mathcal{X}^i(q)| > \varepsilon \right\}.$$

It can be easily seen that $\mathcal{A}_j^\varepsilon$ contains the center c_j, and thus, $\mathcal{A}_j^\varepsilon$ is an open subset containing c_j. The following lemma is an adaptation of Theorem 1 in [58]:

Lemma 7.2. *Suppose that the mobile sensor moves continuously throughout the domain $\mathcal{Q}$ such that in time T, the trajectory traverses through each of the neighbourhoods $\mathcal{A}_j^\varepsilon,\ j = 1, 2, \ldots, m$, then the matrix $\mathcal{S}_T$ given by equation (7.4) is positive definite.*

Proof. $\mathcal{S}_T$ can be written as $\mathcal{S}_T = \mathrm{K}\bar{\mathcal{S}}_T\mathrm{K}^\top$ where

$$\bar{\mathcal{S}}_T = \int_0^T \mathcal{X}(p(t))\mathcal{X}(p(t))^\top dt.$$

Since K is invertible, $\mathcal{S}_T$ is positive definite iff $\bar{\mathcal{S}}_T$ is positive definite. $\bar{\mathcal{S}}_T$ is positive definite iff there exists some $\delta > 0$ such that $\underline{\sigma}(\bar{\mathcal{S}}_T) \geq \delta$ where $\underline{\sigma}(A)$ denotes the minimum singular value of A. Suppose $\bar{\mathcal{S}}_T$ is not positive definite under the conditions of the theorem. Then there exists no $\delta > 0$ such that $\underline{\sigma}(\bar{\mathcal{S}}_T) \geq \delta$. This implies that for any $\delta > 0$, there exists $u \neq 0, \|u\| = 1$ such that $u^\top \bar{\mathcal{S}}_T u < \delta$, i.e.,

$$\int_0^T u^\top \mathcal{X}(p(t))\mathcal{X}(p(t))^\top u\, dt < \delta.$$

Let i be the index of the components of u which has the largest absolute value. i.e., $|u^i| \geq |u^j|\ \forall j$. Also let $[t_{i1}, t_{i2}] \subset [0, T]$ be the subinterval during which the mobile sensor trajectory is contained in the set $\mathcal{A}_i^\varepsilon$. Clearly since the set $\mathcal{A}_i^\varepsilon$ is open and the trajectory is continuous, $[t_{i1}, t_{i2}]$ has finite positive length. Then,

$$\int_0^T u^\top \mathcal{X}(p(t))\mathcal{X}(p(t))^\top u\, dt = \int_0^T |\mathcal{X}^\top u|^2\, dt \tag{7.8}$$

$$\geq \int_{t_{i1}}^{t_{i2}} |\mathcal{X}^\top u|^2\, dt = \int_{t_{i1}}^{t_{i2}} |\sum_{j=1}^m \mathcal{X}^j u^j|^2\, dt \tag{7.9}$$

$$\geq \int_{t_{i1}}^{t_{i2}} (|\mathcal{X}^i u^i| - |\sum_{j=1, j\neq i}^m \mathcal{X}^j u^j|)^2\, dt \tag{7.10}$$

$$\geq \int_{t_{i1}}^{t_{i2}} (|\mathcal{X}^i u^i| - \sum_{j=1, j\neq i}^m |\mathcal{X}^j u^j|)^2\, dt \tag{7.11}$$

$$\geq \int_{t_{i1}}^{t_{i2}} ((|\mathcal{X}^i| - \sum_{j=1, j\neq i}^m |\mathcal{X}^j|)|u^i|)^2\, dt \tag{7.12}$$

$$\geq \int_{t_{i1}}^{t_{i2}} \varepsilon^2 |u^i|^2\, dt = (t_{i2} - t_{i1})\varepsilon^2 |u^i|^2. \tag{7.13}$$

Choosing $\delta < (t_{i2} - t_{i1})\varepsilon^2 |u^i|^2$ leads to a contradiction. Therefore, $\bar{\mathcal{S}}_T$ is positive definite and hence, $\mathcal{S}_T$ is positive definite. □

A sufficient condition for satisfaction of Lemma 7.2's assumptions:

Since checking the condition of the mobile sensor traversing through the sets $\mathcal{A}_j^\varepsilon$ in Lemma 7.2 involves transforming the vector $\mathcal{K}(q)$ at each instant which can be cumbersome if the number of parameters are large, we, therefore, present a simpler sufficient condition which ensures that a given point q is inside the set $\mathcal{A}_j^\varepsilon$. Note that the conditions derived are not equivalent to the conditions of the lemma, but only sufficient and thus can be conservative. However, it may be useful during implementations.

Lemma 7.3. *Given the mobile sensor position p, if*

$$\|\mathcal{K}(p) - \mathcal{K}(c_j)\|_\infty < \frac{(1-\epsilon)}{2(m-1)\|\mathrm{K}^{-1}\|_\infty}, \tag{7.14}$$

then $p \in \mathcal{A}_j^\varepsilon$.

Proof. We have the i-th component of $\mathcal{X}(p)$, $\mathcal{X}^i(p) = [\mathrm{K}^{-1}\mathcal{K}(p)]^i$. Then

$$\mathcal{X}^i(p) - \mathcal{X}^i(c_j) = \left[\mathrm{K}^{-1}(\mathcal{K}(p) - \mathcal{K}(c_j))\right]^i. \tag{7.15}$$

Now consider the mapping

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = B_j \left(\mathcal{X}(p) - \mathcal{X}(c_j)\right), \tag{7.16}$$

where

$$B_j = \begin{bmatrix} 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ 1 & \dots & 1 & 0 & 1 & \dots & 1 \end{bmatrix}. \tag{7.17}$$

The 1 in the first row and the 0 in the second row occurs at the j-th column. If the infinity-norm of $y = [y_1, y_2]^\top$, $\|y\|_\infty < (1-\varepsilon)/2$, then it is guaranteed that $p \in \mathcal{A}_j^\varepsilon$. We also have

$$\|y\|_\infty \leq \|B\|_\infty \|\mathcal{X}(p) - \mathcal{X}(c_j)\|_\infty \tag{7.18}$$

$$\leq \|B\|_\infty \|\mathrm{K}^{-1}\|_\infty \|\mathcal{K}(p) - \mathcal{K}(c_j)\|_\infty. \tag{7.19}$$

Requiring the above bound to be less than $\frac{(1-\epsilon)}{2}$ and noting that $\|B\|_\infty = (m-1)$ we have

$$\|\mathcal{K}(p) - \mathcal{K}(c_j)\|_\infty < \frac{(1-\epsilon)}{2(m-1)\|\mathrm{K}^{-1}\|_\infty}. \tag{7.20}$$

□

Any point p which satisfies the above condition will lie in the set $\mathcal{A}_j^\varepsilon$ although all points in $\mathcal{A}_j^\varepsilon$ are not characterized by the above condition.

7.3 Mobile Sensor Network

Suppose that we have N mobile sensors deployed in the region $\mathcal{Q}$, with the position of mobile sensor i denoted by p_i. We want to estimate the function $\phi : \mathcal{Q} \to \mathbb{R}_+$ collectively. We assume that equation (7.1a) holds so that we can linearly parameterize $\phi(\cdot)$ in terms of radial basis functions. We partition the region into N components $\mathcal{Q}_i$ $(i = 1, 2, \ldots, N)$. Correspondingly we partition the basis function vector $\mathcal{K}(q)$ and the parameter vector a as

$$\mathcal{K}(q) = \begin{bmatrix} \mathcal{K}^{(1)}(q) \\ \mathcal{K}^{(2)}(q) \\ \vdots \\ \mathcal{K}^{(N)}(q) \end{bmatrix}, \qquad a = \begin{bmatrix} a^{(1)} \\ a^{(2)} \\ \vdots \\ a^{(N)} \end{bmatrix}. \tag{7.21}$$

Each region $\mathcal{Q}_i$ contains the centers of the basis functions in the sub-vector $\mathcal{K}^{(i)}$. We assign each region $\mathcal{Q}_i$ to one of the mobile sensors where the sensor operates. This assignment is permanent and each mobile sensor starts within its region $\mathcal{Q}_i$ and moves in $\mathcal{Q}_i$. The algorithms presented below do not depend on any particular partition or assignment of mobile sensors, and this can be done arbitrarily. One particular method to divide the region and assign the sensors will be discussed in section 7.5. Assuming the region $\mathcal{Q}$ is partitioned and the mobile sensors are assigned to each partition, we consider the graph $\mathcal{G}$ with the vertices representing the mobile sensors and an edge existing between two sensors if they belong to adjacent partitions. See figure 7.1 for an illustration. Now we consider two cases: (1) each mobile sensor estimates the entire parameter vector, and (2) each mobile sensor estimates only part of the parameter vector.

7.3.1 Each mobile sensor estimates the full parameter vector

In this subsection, we consider the case where each mobile sensor estimates the entire parameter vector, the estimate of sensor i being denoted by $\hat{a}_i$. To proceed, we consider the following integrators running on mobile sensor i:

$$\dot{\Lambda}_i = \mathcal{K}_i(t)\mathcal{K}_i(t)^\top, \tag{7.22}$$

$$\dot{\lambda}_i = \mathcal{K}_i(t)\phi_i(t), \tag{7.23}$$

where $\mathcal{K}_i(t) = \mathcal{K}(p_i(t))$ and $\phi_i(t) = \phi(p_i(t))$ is the measurement of $\phi(\cdot)$ obtained by sensor i at its location at time t.

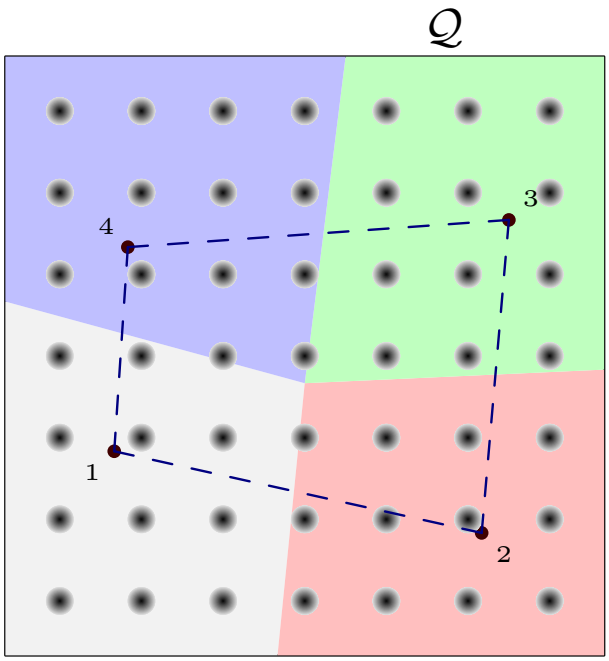


Figure 7.1: Illustration of four mobile sensors with a partition of domain $\mathcal{Q}$: A graph with mobile sensors as root nodes and edge between neighbouring sensors is depicted in the figure. Gaussian RBFs are also indicated in the figure using dark circles.

We consider the following update law for the parameter estimate of mobile sensor i:

$$\dot{\hat{a}}_i = -\Gamma \left(\Lambda_i \hat{a}_i - \lambda_i\right) - \Gamma\zeta \sum_{j=1}^{N} l_{ij} \left(\hat{a}_i - \hat{a}_j\right), \tag{7.24}$$

where ζ is a positive constant, l_{ij} is the weight of the edge between sensors i and j. The weight l_{ij} is zero if there is no edge between sensor i and j and positive otherwise. The first term corresponds to the measurement update of mobile sensor i and the second term is a consensus term to ensure that the estimates of all the mobile sensors asymptotically agree or come close to each other. This is critical in establishing the convergence of the estimation error, as will be shown below.

Lemma 7.4. *Suppose the mobile sensors translate continuously such that in some time T, each sensor i passes through each of the centers in the region $\mathcal{Q}_i$ so that*

$$\int_0^T \mathcal{K}_i^{(i)}(t)\mathcal{K}_i^{(i)}(t)^\top dt > 0, \quad \text{for } i = 1, 2, \ldots, N,$$

where $\mathcal{K}_i^{(i)}(t)$ denotes part of the vector $\mathcal{K}_i(t)$ corresponding to the partition (7.21). Then,

we have

$$\sum_{i=1}^{N} \int_{0}^{T} \mathcal{K}_i(t)\mathcal{K}_i(t)^\top dt > 0.$$

Proof. Since each mobile sensor i passes through the centers in the region $\mathcal{Q}_i$, the union of the trajectories of all mobile sensors cover all the centers, which implies that the matrix

$$\sum_{i=1}^{N} \int_{0}^{T} \mathcal{K}_i(t)\mathcal{K}_i(t)^\top dt \tag{7.25}$$

is positive definite using the same arguments as in the proof of Corollary 7.4 and Lemma 7.1. □

Remark 7.6. Lemma 7.4 states that each agent passing through the centers in its partition $\mathcal{Q}_i$ is sufficient to ensure that the total sum matrix (7.25) is positive definite.

Now we have the following result:

Theorem 7.7. *Suppose the N mobile sensors adopt the parameter adaptation law (7.24). Further, assume that each mobile sensor i produces a trajectory going through all the basis function centers in $\mathcal{Q}_i$. Then*

$$\lim_{t\to\infty} (\hat{a}_i - a) = 0, \tag{7.26}$$

for each $i \in \{1, 2, \ldots, N\}$, i.e., the mobile sensors arrive at a common value for the parameters, the common value being the true parameter value.

Proof. Consider the function

$$V = \frac{1}{2} \sum_{i=1}^{N} \tilde{a}_i^\top \Gamma^{-1} \tilde{a}_i. \tag{7.27}$$

Taking the derivative of V,

$$\begin{aligned} \dot{V} &= \sum_{i=1}^{N} \tilde{a}_i^\top \Gamma^{-1} \dot{\tilde{a}}_i \\ &= -\sum_{i=1}^{N} \tilde{a}_i^\top \left(\Lambda_i \hat{a}_i - \lambda_i\right) - \zeta \sum_{i=1}^{N} \tilde{a}_i^\top l_{ij} \left(\hat{a}_i - \hat{a}_j\right). \end{aligned}$$

Substituting for the variables Λ_i, λ_i and rearranging the second term,

$$\dot{V} = -\sum_{i=1}^{N} \tilde{a}_i^\top \int_{0}^{t} \mathcal{K}_i(\tau)\mathcal{K}_i^\top(\tau) d\tau \tilde{a}_i - \zeta \sum_{\alpha=1}^{m} \hat{a}^{\alpha^\top} L \hat{a}^\alpha \tag{7.28}$$

$$\leq 0, \tag{7.29}$$

where $\hat{a}^\alpha = [a_1^\alpha\, a_2^\alpha\, \ldots\, a_N^\alpha]^\top$ is the vector of the estimate of parameter α of all the sensors stacked together. The function V is lower bounded and non-increasing, and therefore tends to a limit. This implies that $\dot{V}$ is integrable and also that the estimates $\hat{a}_i$ are bounded. $\dot{V}$ is also uniformly continuous since the derivative of each term in $\dot{V}$ is bounded. Using Barbalat's lemma, we conclude that $\dot{V}$ tends to zero as $t \to \infty$. From the second term in $\dot{V}$, noting that L is the Laplacian matrix of the connected graph $\mathcal{G}$ with nullspace $k\mathbf{1}$ where $\mathbf{1}$ is the vector of ones and $k \in \mathbb{R}$, we see that as $t \to \infty$, $\hat{a}^\alpha \to k_\alpha \mathbf{1}$ for some k_α. Then,

$$\lim_{t\to\infty} (\hat{a}_i - \hat{a}_j) = 0$$

since $\hat{a}_i = [a_i^1\, a_i^2\, \ldots\, a_i^m]^\top$. Now from the first term of $\dot{V}$ we have, as $t \to \infty$,

$$-\tilde{a}^\top \sum_{i=1}^{N} \int_0^t \mathcal{K}_i(\tau)\mathcal{K}_i^\top(\tau) d\tau \tilde{a} = 0,$$

where $\tilde{a}$ is the common value to which the mobile sensor parameter estimation errors $\tilde{a}_i$ converge. Then using Lemma 7.4, it follows that $\lim_{t\to\infty} \tilde{a} = 0$ and the parameter estimates converge to the true parameter values. □

Remark 7.8. Although Lemma 7.4 and Theorem 7.7 requires that the mobile sensors move through the centers, the relaxation given in section 7.2.1 (requiring that the mobile sensors move only through the neighbourhoods $\mathcal{A}_j^\varepsilon$ of the centers) also applies here, as well as in all the following results, which requires the sensors to move through the centers.

7.3.2 Each mobile sensor estimates only part of the parameter vector

If the number of parameters m is large as could be the case when the density function is completely unknown, each mobile sensor estimating the entire parameter vector could be computationally intensive, as it would require computing $\left(\frac{m(m+1)}{2} + m\right)$ filter variables in addition to the m parameter estimates. In such cases, it would be beneficial to have each mobile sensor estimate only part of the parameters. Suppose each mobile sensor i is to estimate only part of the a-vector $a^{(i)}$ given by (7.21). Now we use $\hat{a}_i$ to denote the estimate of $a^{(i)}$ by sensor i. We write

$$\phi(q) = \mathcal{K}(q)^\top a \tag{7.30}$$

$$= \mathcal{K}^{(i)}(q)^\top a^{(i)} + \bar{\mathcal{K}}^{(i)^\top} \bar{a}^{(i)}, \tag{7.31}$$

where $\mathcal{K}(q)$ and the parameter a are partitioned appropriately. Since the mobile sensor i's measurement is denoted by $\phi_i(t) := \phi(p_i(t))$, we have

$$\phi_i(t) = \mathcal{K}_i^{(i)}(t)^\top a^{(i)} + \bar{\mathcal{K}}_i^{(i)}(t)^\top \bar{a}^{(i)} \tag{7.32}$$

$$= \mathcal{K}_i^{(i)}(t)^\top a^{(i)} + \Delta\phi_i(t), \tag{7.33}$$

where $\mathcal{K}_i(t) := \mathcal{K}(p_i(t))$ and $\Delta\phi_i(t) := \bar{\mathcal{K}}_i^{(i)}(t)^\top \bar{a}^{(i)}$. The basis functions in $\bar{\mathcal{K}}_i^{(i)}(t)$ are centered outside the region $\mathcal{Q}_i$ and thus their values at the points $p_i(t)$ are assumed to be small. Under this condition, we consider the contribution to $\phi(.)$ from these terms as a disturbance $\Delta\phi_i(t)$.

Let $C = \{c_1, c_2, \ldots, c_m\}$ be the set of centers of the basis functions, $C_i \subset C$ be its subset which belongs to $\mathcal{Q}_i$. We can then bound $\Delta\phi_i(t)$ as follows:

Lemma 7.5. *For each mobile sensor i, $i \in \{1, 2, \ldots, N\}$,*

$$|\Delta\phi_i(t)| \leq m\delta_i a_{\max}, \tag{7.34}$$

where $\delta_i := \max_{j\in\{1,\ldots,m\}} \exp\left\{-\frac{d_i^2}{\sigma_j^2}\right\}$, $d_i := dist(C_i, C \setminus C_i)$, $dist(A, B) = \min_{a\in A, b\in B} \|a - b\|$, and $a_{\max}$ is an upper bound for the parameters, i.e., $|a^i| \leq a_{\max}\ \forall i \in \{1, 2, \ldots, m\}$.

To make the bound independent of i, we have

$$|\Delta\phi_i(t)| \leq m\delta a_{\max}, \tag{7.35}$$

where $\delta = \max_{j\in\{1,\ldots,N\}} \delta_i$.

Proof. The lemma follows from the definition of $\Delta\phi_i(t)$ using Cauchy-Schwartz inequality. □

We now define the following integrators:

$$\begin{aligned} \dot{\Lambda}_i &= s\mathcal{K}_i^{(i)}\mathcal{K}_i^{(i)^\top}, \\ \dot{\lambda}_i &= s\mathcal{K}_i^{(i)}\phi_i, \end{aligned} \tag{7.36}$$

where s is a switching signal which takes values in the set $\{0, 1\}$. Consider the following adaptation law:

$$\dot{\hat{a}}_i = -\Gamma\left(\Lambda_i \hat{a}_i - \lambda_i\right). \tag{7.37}$$

Then we have the following result:

Theorem 7.9. *Suppose the N mobile sensors implement the parameter adaptation law (7.37) with each sensor i only estimating part of the full parameter vector $a^{(i)}$. Further, assume that each mobile sensor i produces a trajectory going through all the basis function centers in $\mathcal{Q}_i$ in time T. Then setting $s = 1$ for $t \leq T$ and $s = 0$ for $t > T$ in (7.36), we have*

$$\lim_{t\to\infty} \|\hat{a}_i(t) - a^{(i)}\| \leq r_i,$$

where

$$r_i = \frac{Tm\delta_i a_{\max}}{\alpha\eta_i},$$

$a_{\max}$ is the upper bound on the parameter values in $a^{(i)}$, $\alpha \in (0,1)$ and η_i is the smallest eigen-value of the matrix $\int_0^T \mathcal{K}_i^{(i)} \mathcal{K}_i^{(i)^\top} d\tau$.

Proof. Consider

$$V = \frac{1}{2} \sum_{i=1}^{N} \tilde{a}_i^\top \Gamma^{-1} \tilde{a}_i. \tag{7.38}$$

Taking the derivative,

$$\dot{V} = -\sum_{i=1}^{N} \tilde{a}_i^\top \left(\Lambda_i \hat{a}_i - \lambda_i\right) \tag{7.39}$$

$$= -\sum_{i=1}^{N} \tilde{a}_i^\top \int_0^t s\mathcal{K}_i^{(i)} \left(\mathcal{K}_i^{(i)^\top} \hat{a}_i - \mathcal{K}_i^{(i)^\top} a^{(i)} - \Delta\phi_i\right) d\tau \tag{7.40}$$

$$= -\sum_{i=1}^{N} \tilde{a}_i^\top \int_0^t s\mathcal{K}_i^{(i)} \left(\mathcal{K}_i^{(i)^\top} \tilde{a}_i - \Delta\phi_i\right) d\tau \tag{7.41}$$

$$= -\sum_{i=1}^{N} \tilde{a}_i^\top \int_0^t s\mathcal{K}_i^{(i)} \mathcal{K}_i^{(i)^\top} d\tau\, \tilde{a}_i + \sum_{i=1}^{N} \tilde{a}_i^\top \int_0^t s\mathcal{K}_i^{(i)} \Delta\phi_i d\tau. \tag{7.42}$$

For $t \geq T$, the first term becomes negative definite. We then have

$$\dot{V} = -\sum_{i=1}^{N} \tilde{a}_i^\top \int_0^T \mathcal{K}_i^{(i)} \mathcal{K}_i^{(i)^\top} d\tau\, \tilde{a}_i + \sum_{i=1}^{N} \tilde{a}_i^\top \int_0^T \mathcal{K}_i^{(i)} \Delta\phi_i d\tau, \tag{7.43}$$

for $t > T$. Then

$$\dot{V} \leq -\sum_{i=1}^{N} \eta_i \|\tilde{a}_i\|^2 + \sum_{i=1}^{N} \|\tilde{a}_i\| Tm\delta_i a_{\max} \tag{7.44}$$

$$\leq -\kappa V - \sum_{i=1}^{N} \|\tilde{a}_i\| \left(\alpha\eta_i \|\tilde{a}_i\| - Tm\delta_i a_{\max}\right), \tag{7.45}$$

where $\kappa = \frac{\eta_{\min}}{\lambda_{\max}(\Gamma^{-1})}$, $\alpha \in (0,1)$ and $\eta_{\min} = \min_{i\in\{1,2,\ldots,N\}} \eta_i$. Thus for $\|\tilde{a}_i\| > r_i$, we have $\dot{V} \leq -\kappa V$ and V decays exponentially. Therefore the statement of the theorem holds. □

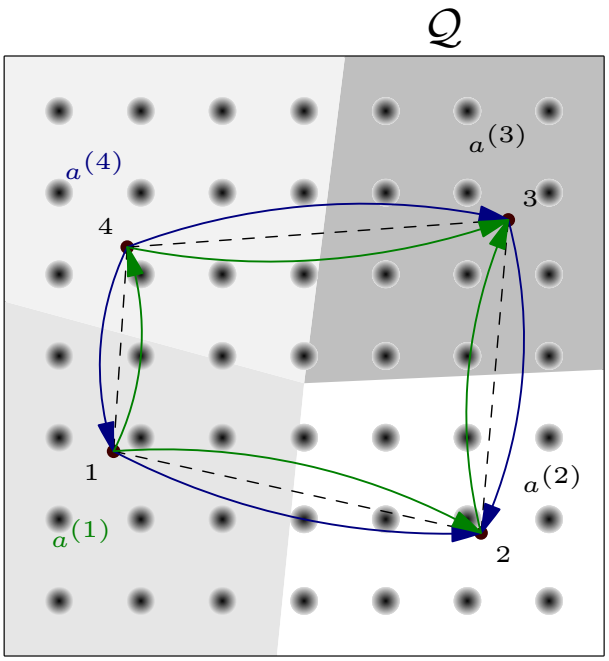


Figure 7.2: Illustration of four mobile sensors with the directed graphs corresponding to $a^{(1)}$ and $a^{(4)}$. The Gaussian RBFs are also indicated in the figure using dark circles.

7.3.3 Improving the steady state error

In this section, we propose a strategy to improve the steady-state error with the strategy in Theorem 7.9. Note that the strategy in Theorem 7.9 is completely decentralized in that there is no real-time communication required between the mobile sensors to implement the estimation strategy. On the other hand, we may be able to get better parameter estimates at the cost of exchanging information about parameter estimates with other mobile sensors.

The term $\Delta\phi_i(t)$ depends on the true value of parameters corresponding to the other mobile sensors (denoted $\bar{a}^{(i)}$). Since we do not know the true values, we cannot cancel this term and treat it as a disturbance. But we may do better since we know that the other mobile sensors have estimates for the true values of $\bar{a}^{(i)}$. We can use these parameter estimates to reduce the effect of the $\Delta\phi_i(t)$ term on the estimation algorithm. Note that the vector $\bar{a}^{(i)}$ consists of the sub-vectors $a^{(j)}$ for all $j \neq i$. Now, corresponding to each $a^{(i)}$, we construct a directed graph with a rooted out-branching (see [59]), denoted $\mathcal{G}_i$ which is a sub-graph of the undirected graph $\mathcal{G}$ with mobile sensor i as the root node. An illustration is shown in figure 7.2.

For each mobile sensor i, we introduce additional states b_i^j for each $j \in \{1, 2, \ldots, N\}$

and $j \neq i$, which evolves according to the equation

$$\dot{b}_i^j = -\sum_{k=1}^{N} l_{ik}^d \left(\hat{b}_i^j - \hat{b}_k^j\right), \tag{7.46}$$

where we define $b_i^i := \hat{a}_i$ for ease of notation and l_{ik}^d is zero if there is no directed path from node i to k in graph $\mathcal{G}_j$, and non-zero constant value otherwise. This implements a directed consensus protocol on the variables b_i^j with $i = 1, 2, \ldots, N$ (see [59]) converging to the root value $b_j^j = \hat{a}_j$ for each j. Thus b_i^j is an estimator of $\hat{a}_j$ by mobile sensor i. We now use the modified integrators:

$$\dot{\Lambda}_i = s\mathcal{K}_i^{(i)}\mathcal{K}_i^{(i)^\top}, \tag{7.47}$$

$$\dot{\lambda}_i = s\mathcal{K}_i^{(i)}\left(\phi_i - \bar{\mathcal{K}}_i^{(i)^\top} b_i\right), \tag{7.48}$$

where b_i is the concatenated vector given by $b_i = \left[b_i^{1^\top} \ldots b_i^{j^\top} \ldots b_i^{N^\top}\right]^\top$ ($j = i$ not included). Using the adaptation law (7.37) we can see that the disturbance term now becomes

$$\Delta\phi_i'(t) := \bar{\mathcal{K}}_i^{(i)}(t)^\top(\bar{a}^{(i)} - b_i), \tag{7.49}$$

which is expected to be smaller than $\Delta\phi_i(t)$, although we cannot put a theoretical bound better than r_i in Theorem 7.9. The stability and convergence in case of the above modification is not proved here as it is essentially a similar exercise to that in the previous section. We will investigate the effect of the above modification in section 7.5.

7.4 Unknown Centres

In this section, we assume as before that the scalar field is a finite linear combination of radial basis functions. We further assume that the centers are not exactly known, but known to within an accuracy of ϵ_c, i.e., $\|\hat{c}_i - c_i\| \leq \epsilon_c$. We will evaluate the quality of parameter estimates in this case. Define

$$\tilde{\mathcal{K}}(q) = \hat{\mathcal{K}}(q) - \mathcal{K}(q),$$

where $\hat{\mathcal{K}}(q)$ is the RBF evaluated at the known values of the centers and $\mathcal{K}(q)$ corresponds to the true values of the centers.

7.4.1 Each mobile sensor estimates only a part of the parameter vector

As in section 7.3.2, we assume that each mobile sensor estimates part of the parameter vector $a^{(i)}$ corresponding to the partition $\mathcal{Q}_i$. In this case we propose the following modified filters,

$$\dot{\Lambda}_i = s\hat{\mathcal{K}}_i^{(i)}\hat{\mathcal{K}}_i^{(i)^\top}, \tag{7.50}$$

$$\dot{\lambda}_i = s\hat{\mathcal{K}}_i^{(i)}\phi_i, \tag{7.51}$$

with equation (7.37) as the adaptation law. Then we have the following result.

Proposition 7.2. *Assuming the centers are only known to within an accuracy of ϵ_c, let each mobile sensor pass through the set of known (inaccurate) centers in $\mathcal{Q}_i$ in time T. If each mobile sensor implements the adaptation law* (7.37) *along with* (7.51) *setting $s = 1$ for $t \leq T$ and $s = 0$ for $t > T$, the estimation error $\tilde{a}_i$ converges to within a bound r_i of the origin, where*

$$r_i = \frac{Tma_{\max}(\sqrt{m}k\epsilon_c + \delta_i)}{\alpha\eta_i},$$

with $k > 0$ is the Lipschitz constant such that $|\mathcal{K}^i(q) - \hat{\mathcal{K}}^i(q)| \leq k\epsilon_c$, η_i is the smallest eigen-value of $\int_0^T \hat{\mathcal{K}}_i^{(i)}\hat{\mathcal{K}}_i^{(i)^\top} d\tau$ and $\alpha \in (0, 1)$.

Proof. Consider the same Lyapunov function as before,

$$V = \sum_{i=1}^{N} \tilde{a}_i^\top \Gamma^{-1} \tilde{a}_i.$$

Taking the time derivative,

$$\begin{aligned}
\dot{V} &= -\sum_{i=1}^{N} \tilde{a}_i^\top \left(\Lambda_i \hat{a}_i - \lambda_i\right) \\
&= -\sum_{i=1}^{N} \tilde{a}_i^\top \int_0^t s\hat{\mathcal{K}}_i^{(i)} \left(\hat{\mathcal{K}}_i^{(i)^\top}\hat{a}_i - \mathcal{K}_i^{(i)^\top} a^{(i)} - \Delta\phi_i\right) d\tau \\
&= -\sum_{i=1}^{N} \tilde{a}_i^\top \int_0^t s\hat{\mathcal{K}}_i^{(i)}\hat{\mathcal{K}}_i^{(i)^\top} d\tau\, \tilde{a}_i - \sum_{i=1}^{N} \tilde{a}_i^\top \int_0^t s\hat{\mathcal{K}}_i^{(i)}\tilde{\mathcal{K}}_i^{(i)^\top} d\tau\, a^{(i)} + \sum_{i=1}^{N} \tilde{a}_i^\top \int_0^t s\hat{\mathcal{K}}_i^{(i)} \Delta\phi_i d\tau.
\end{aligned}$$

Setting $s = 1$ for $t \leq T$ and $s = 0$ for $t > T$ as before and, assuming the first term becomes negative definite at time T, while noting that $|\hat{\mathcal{K}}^i(q)| \leq 1 \implies \|\hat{\mathcal{K}}(q)\| \leq \sqrt{m}$,

and $|\tilde{\mathcal{K}}^i(q)| \leq k\epsilon_c \implies \|\tilde{\mathcal{K}}(q)\| \leq \sqrt{m}k\epsilon_c$ for some k (Lipschitz constant), we now have

$$\begin{aligned}\dot{V} &\leq -\sum_{i=1}^{N} \tilde{a}_i^\top \int_0^T \hat{\mathcal{K}}_i^{(i)} \hat{\mathcal{K}}_i^{(i)\top} d\tau\, \tilde{a}_i + \sum_{i=1}^{N} \|\tilde{a}_i\| Tma_{\max}(\sqrt{m}k\epsilon_c + \delta_i) \\ &\leq -\kappa V - \sum_{i=1}^{N} \|\tilde{a}_i\| \left(\alpha\eta_i \|\tilde{a}_i\| - Tma_{\max}(\sqrt{m}k\epsilon_c + \delta_i)\right),\end{aligned}$$

for $t \geq T$. Therefore, the statement of the proposition follows. □

7.4.2 Each mobile sensor estimates the entire parameter vector

We define the following filter equations,

$$\dot{\Lambda}_i = s\hat{\mathcal{K}}_i \hat{\mathcal{K}}_i^\top, \tag{7.52}$$

$$\dot{\lambda}_i = s\hat{\mathcal{K}}_i \phi_i. \tag{7.53}$$

The adaptation law is given by equation (7.24). In this case, we have

Proposition 7.3. *Suppose the N mobile sensors adopt the parameter adaptation law (7.24) with the integrators* (7.53)*, other conditions remaining the same as in Proposition 7.2. Then the parameter estimation errors of the mobile sensors converge to within a bound r_i of origin, where*

$$r_i = \frac{Tm\sqrt{m}k\epsilon_c a_{\max}}{\alpha\eta_{\min}},$$

where k is the Lipschitz constant and $\eta_{\min}$ is the smallest eigen-value of $\left(\underline{Q} + P^\top \underline{L} P\right)$ with $\underline{Q}$, $\underline{L}$, and P given by (7.54)*,* (7.55)*, and* (7.57)*.*

Proof. Consider the Lyapunov function

$$V = \sum_{i=1}^{N} \tilde{a}_i^\top \Gamma^{-1} \tilde{a}_i.$$

Taking the derivative of V,

$$\begin{aligned}\dot{V} &= \sum_{i=1}^{N} \tilde{a}_i^\top \Gamma^{-1} \dot{\tilde{a}}_i \\ &= -\sum_{i=1}^{N} \tilde{a}_i^\top \left(\Lambda_i \hat{a}_i - \lambda_i\right) - \zeta \sum_{i=1}^{N} \tilde{a}_i^\top l_{ij} \left(\hat{a}_i - \hat{a}_j\right).\end{aligned}$$

Substituting for the variables Λ_i, λ_i and rearranging the second term,

$$\dot{V} = -\sum_{i=1}^{N} \tilde{a}_i^\top \int_0^t s\hat{\mathcal{K}}_i \hat{\mathcal{K}}_i^\top d\tau \tilde{a}_i - \sum_{i=1}^{N} \tilde{a}_i^\top \int_0^t s\hat{\mathcal{K}}_i \tilde{\mathcal{K}}_i{}^\top d\tau\, a^{(i)} - \zeta \sum_{\alpha=1}^{m} \hat{a}^{\alpha\top} L \hat{a}^\alpha.$$

Simplifying,

$$\dot{V} = -\sum_{i=1}^{N} \tilde{a}_i^\top \int_0^T \hat{\mathcal{K}}_i \hat{\mathcal{K}}_i^\top d\tau \tilde{a}_i - \sum_{i=1}^{N} \tilde{a}_i^\top \int_0^T \hat{\mathcal{K}}_i \tilde{\mathcal{K}}_i{}^\top d\tau \, a^{(i)} - \zeta \sum_{\alpha=1}^{m} \tilde{a}^{\alpha\top} L \tilde{a}^\alpha,$$

for $t \geq T$. We can write the first and last terms in the above equation in terms of stacked vectors as

$$\begin{aligned} \dot{V} &= -\underline{\tilde{a}}^\top \underline{Q} \underline{\tilde{a}} - \zeta \, \underline{\tilde{a}}^\top P^\top \underline{L} P \underline{\tilde{a}} - \underline{\tilde{a}}^\top E \underline{a} \\ &= -\underline{\tilde{a}}^\top \left(\underline{Q} + \zeta \, P^\top \underline{L} P \right) \underline{\tilde{a}} - \underline{\tilde{a}}^\top E \underline{a}, \end{aligned}$$

where $\underline{\tilde{a}} = [\tilde{a}_1^\top \, \tilde{a}_2^\top \, \ldots \, \tilde{a}_N^\top]^\top$,

$$\underline{Q} = \begin{bmatrix} \int_0^T \hat{\mathcal{K}}_1 \hat{\mathcal{K}}_1^\top d\tau & \ldots & 0 \\ 0 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & \int_0^T \hat{\mathcal{K}}_N \hat{\mathcal{K}}_N^\top d\tau \end{bmatrix}, \tag{7.54}$$

$$\underline{L} = \begin{bmatrix} L & 0 & \ldots & 0 \\ 0 & L & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & L \end{bmatrix}, \tag{7.55}$$

$$E = \begin{bmatrix} \int_0^T \hat{\mathcal{K}}_1 \tilde{\mathcal{K}}_1^\top d\tau & \ldots & 0 \\ 0 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & \int_0^T \hat{\mathcal{K}}_N \tilde{\mathcal{K}}_N^\top d\tau \end{bmatrix} \tag{7.56}$$

and P is the permutation matrix

$$P = \begin{bmatrix} 1 & 0 & \ldots & 0 & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 1 & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & \ldots & 0 & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \tag{7.57}$$

of dimension $mN \times mN$. We show that the matrix $\left(\underline{Q} + P^\top \underline{L} P \right)$ is positive definite. Each of the terms are positive semi-definite. The nullspace of matrix $\underline{L}$ contains elements of

the form

$$c_1 \begin{bmatrix} \mathbf{1}_m \\ 0 \\ \vdots \\ 0 \end{bmatrix} + c_2 \begin{bmatrix} 0 \\ \mathbf{1}_m \\ \vdots \\ 0 \end{bmatrix} + \cdots + c_N \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \mathbf{1}_m \end{bmatrix}.$$

Therefore $P^\top \underline{L} P$ has nullspace elements of the form

$$c_1 \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + c_2 \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} + \cdots + c_N \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix},$$

i.e., elements of the form $[\, c_1\, c_2\, \ldots\, c_N\, c_1\, c_2\, \ldots\, c_N]^\top$. Correspondingly the $\underline{Q}$ term can be written as

$$c^\top \sum_{i=1}^{N} \int_0^T \hat{\mathcal{K}}_i \hat{\mathcal{K}}_i^\top dt\, c,$$

where $c = [\, c_1\, c_2\, \ldots\, c_N]^\top$. Under the assumptions of the proposition, and Lemma 7.4, the above term is strictly positive. Hence $\left(\underline{Q} + P^\top \underline{L} P\right)$ is positive definite. Let $\eta_{\min}$ be the smallest eigen-value of $\left(\underline{Q} + P^\top \underline{L} P\right)$. Then we have

$$\begin{aligned} \dot{V} &\leq -\kappa V - \alpha \eta_{\min} \|\underline{\tilde{a}}\|^2 + \sum_{i=1}^{N} \|\tilde{a}_i\| T m \sqrt{m} k \epsilon_c a_{\max} \\ &= -\kappa V - \alpha \eta_{\min} \sum_{i=1}^{N} \|\tilde{a}_i\| \left(\|\tilde{a}_i\| - \frac{T m \sqrt{m} k \epsilon_c a_{\max}}{\alpha \eta_{\min}} \right), \end{aligned}$$

for some $\kappa > 0$. Thus for $\|\tilde{a}_i\| > \frac{T m \sqrt{m} k \epsilon_c a_{\max}}{\alpha \eta_{\min}}$, V decreases exponentially and the result holds. □

7.5 Simulations

In this section, we verify the algorithms presented using simulations. First, we consider the exact parameterization case where the true scalar field is a linear combination

$c_{i,x}$	0.20	0.35	0.60	0.85	0.70	0.75	0.15	0.35
$c_{i,y}$	0.25	0.26	0.18	0.30	0.75	0.90	0.75	0.60
a^i	2.0	1.0	1.5	1.8	1.2	1.6	2.5	1.1

Table 7.1: Parameters of the simulated scalar field

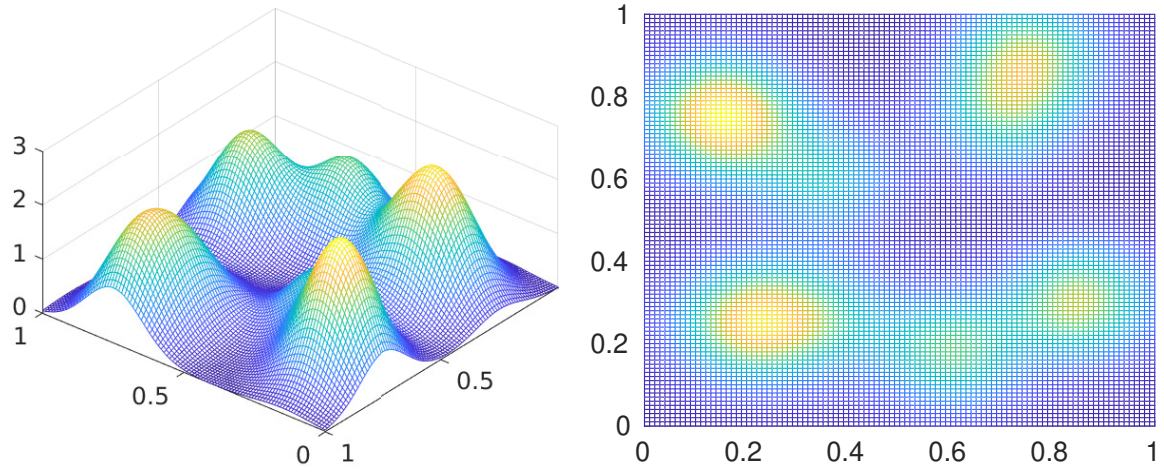
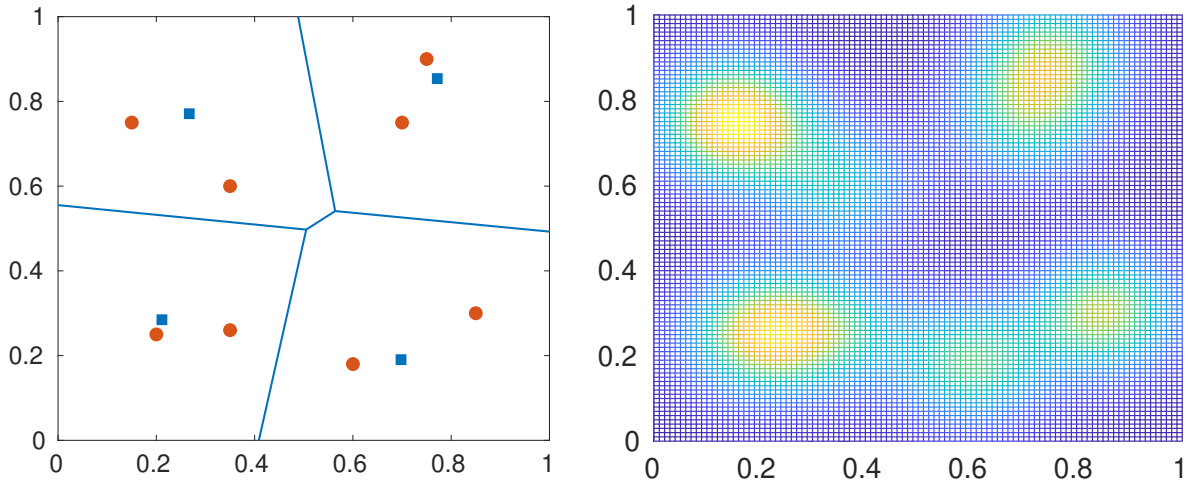


Figure 7.3: The scalar field used for verifying the algorithms

of RBFs where the centers of RBFs are known. This case allows us to verify the correctness of the algorithms presented in the chapter. Next, we consider a scalar field that is completely unknown and use the algorithms presented to reconstruct the scalar field. The mobile sensors in the simulations are assumed to be single integrators with dynamics given by $\dot{p}_i = u_i$ where p_i is the position of sensor i and u_i is its control input. For ease of comparing various algorithms, we refer to the algorithm in section 7.3.1 as *Algorithm S*1, the algorithm presented in section 7.3.2 as *Algorithm S*2, and the modified version of algorithm *S*2 in section 7.3.3 as *Algorithm S*3.

7.5.1 Exact parameterization

We consider the unit square region $\mathcal{Q}$ with four mobile sensors. The scalar field to be estimated is exactly parameterized in terms of Gaussian RBFs (given by equation (7.2)), the x and y coordinates of the RBF centers c_i being given in table 7.1. The standard deviation of each of the Gaussians σ_i is chosen to be 0.1. The true parameter values a^i are given in table 7.1. The scalar field is shown in figure 7.3. The initial positions of



(a) Partitions

(b) Algorithm 1.

Figure 7.4: Left: Initial positions (blue squares), corresponding partitions and centres of RBFs (red circles); Right: Reconstructed field using algorithm $S1$.

the mobile sensors were chosen randomly and shown in figure 7.4. The partition of the region was done by constructing the Voronoi cells for each mobile sensor. The Voronoi cell of mobile sensor i (denoted $\mathcal{Q}_i$) consists of those points which are closer to sensor i as compared to all other sensors:

$$\mathcal{Q}_i = \{q \in \mathcal{Q} : \|q - p_i\| \leq \|q - p_j\|, j = 1, 2, \ldots, N; j \neq i\} \tag{7.58}$$

For motion control of the sensors, we use a proportional control law $u_i = k(p_i - p_{gi})$ where p_{gi} is made to switch between all the centers in the region $\mathcal{Q}_i$ making sure the condition in Lemma 7.2 is satisfied. The control gain k was chosen to be 5. The simulation ran for 16.5 seconds. The excitation condition was achieved in $T = 1.5$ seconds. The reconstructed scalar field with algorithm $S1$ is shown in figure 7.4 on the right and the average (across all the mobile sensors) parameter estimation error is shown in figure 7.5. It can be seen that the parameters converge exactly to the true values and exact reconstruction is achieved.

The reconstructed field with algorithm $S2$ and algorithm $S3$ is shown in figure 7.6. The corresponding estimation errors are shown in figures 7.7 and 7.8, respectively. The maximum parameter estimation error using algorithm $S2$ was found to be 0.030, and using the algorithm $S3$ was found to be 0.017. Thus the algorithm $S3$ is seen to give better parameter estimates in this case.

We also present simulation results where we do not know the exact value of the

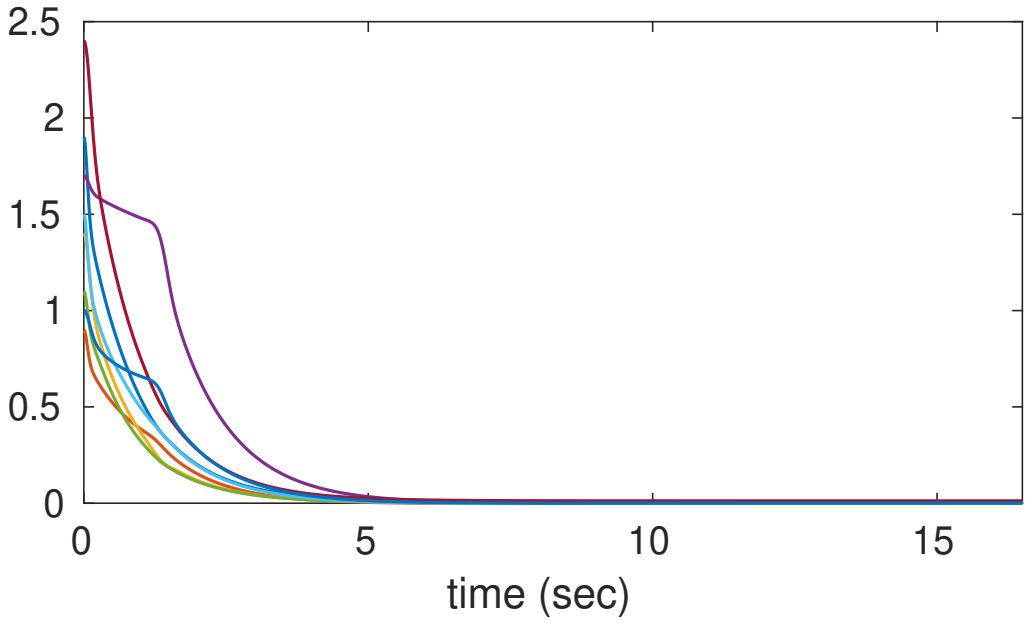
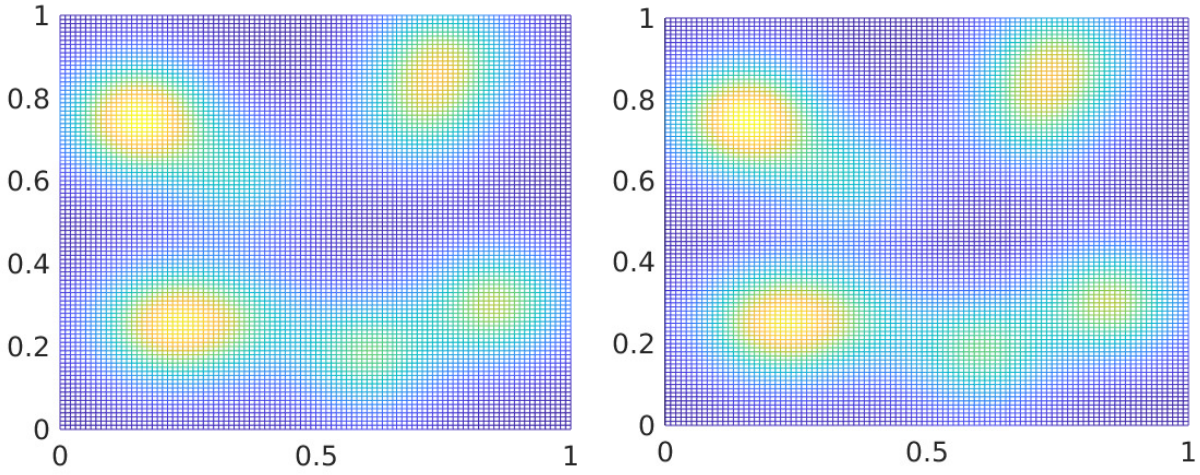


Figure 7.5: Algorithm $S1$: Average parameter estimation error with time



(a) Algorithm $S2$.

(b) Algorithm $S3$.

Figure 7.6: The reconstructed field using algorithms $S2$ and $S3$.

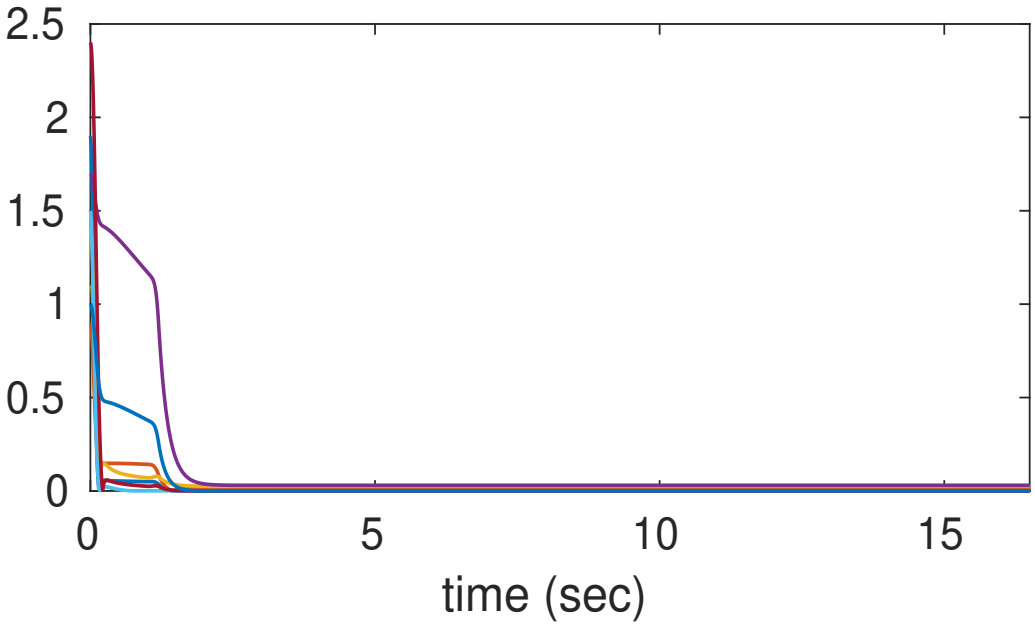


Figure 7.7: Algorithm $S2$: Average parameter estimation error with time

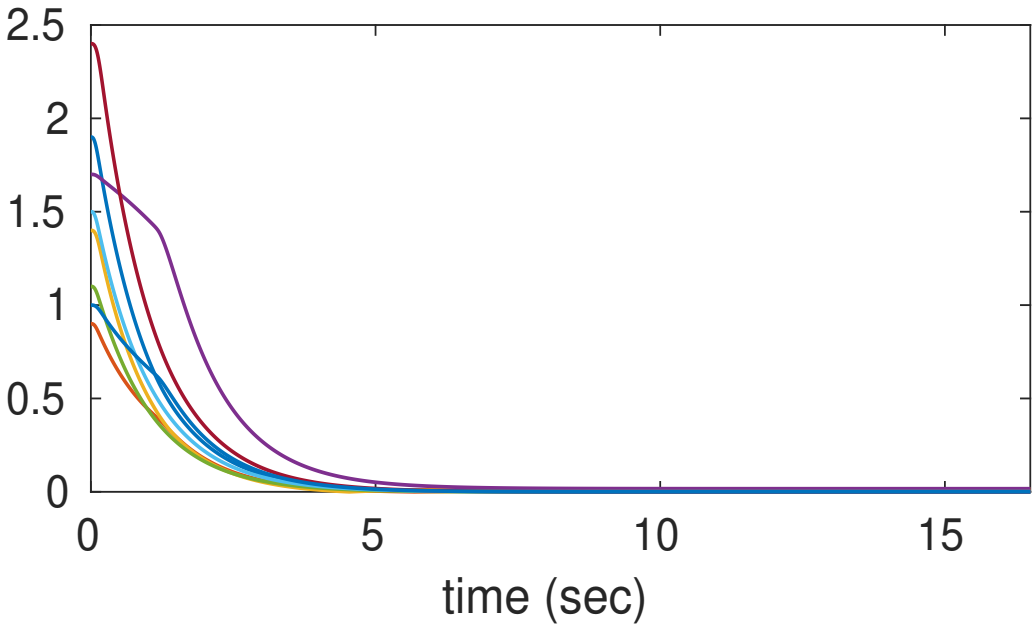
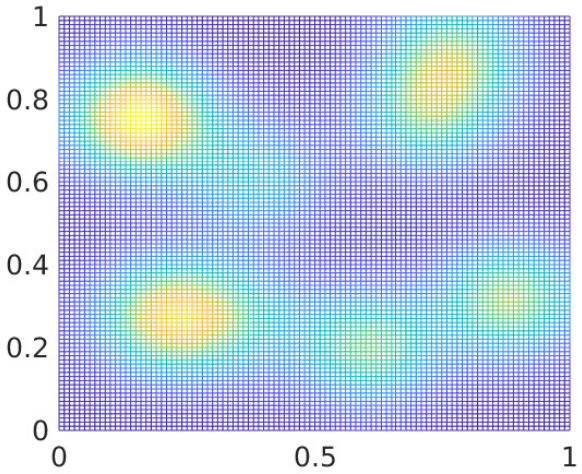


Figure 7.8: Algorithm $S3$: Average parameter estimation error with time

	Max. est. error
*Algorithm S*1	0.16
*Algorithm S*2	0.62
*Algorithm S*3	0.44

(a) Max. parameter estimation errors.



(b) Algorithm $S1$.

Figure 7.9: Unknown Centres: Max. parameter estimation errors (left) and the reconstructed field using algorithm $S1$ (right).

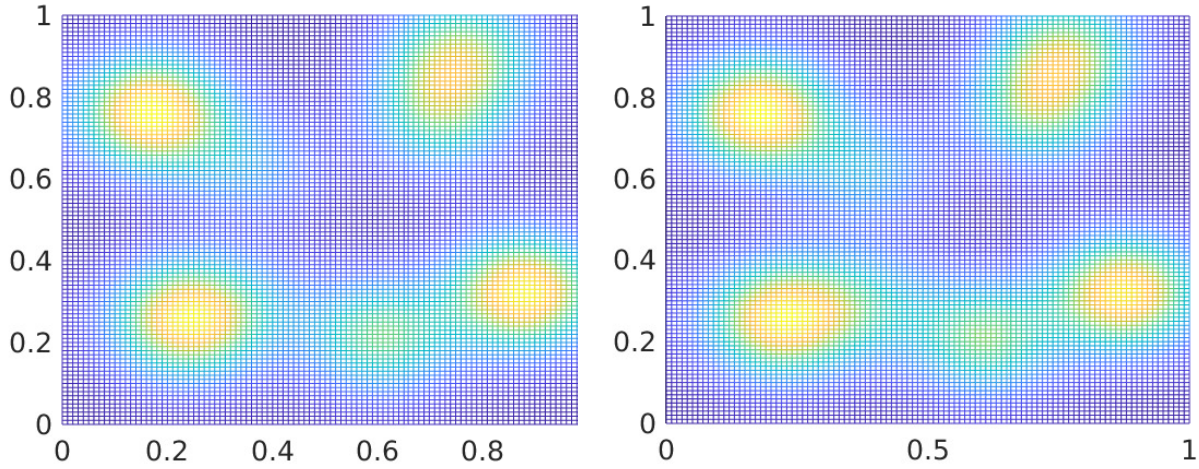
centers of the RBFs (as in section 7.4). We assume we know the centers within an accuracy of $\epsilon_c = 0.05$. For this, we add a random perturbation (bounded by ϵ_c) to the true center coordinates and use the perturbed centers in the estimation algorithm. The reconstructed fields with algorithm $S1$, algorithm $S2$, and algorithm $S3$ are shown in figures 7.9 and 7.10, respectively. Table 7.9a also compares the maximum steady-state parameter errors in the three cases. As expected, algorithm $S1$ has much lower steady state error compared algorithm $S2$ and the algorithm $S3$ performs better than algorithm $S2$. It should be noted that all the algorithms identify the main features of the true field, as seen from the reconstructed field plots.

7.5.2 Fully unknown scalar field

Now we test the estimation algorithm on a more general scalar field which is not a linear combination of RBFs. For this we consider the continuous scalar field given by

$$\phi(x,y) = 3x^2 e^{\frac{-(x-0.7)^2-(y-0.7)^2}{0.05}} + e^{\frac{-(x-0.4)^2-(y-0.4)^2}{0.06}} + \frac{1}{3}e^{\frac{-(x-0.2)^2-(y-0.2)^2}{0.08}}$$

over the unit square region $\mathcal{Q}$. A plot of $\phi(\cdot)$ is shown in figure 7.11. We use $N = 5$ mobile sensors with the partitions $\mathcal{Q}_i$ determined as follows: We first run a uniform coverage algorithm (coverage algorithm presented in [1] with a uniform density function $\phi(q) \equiv 1$). This makes the mobile sensors uniformly spread out in the region $\mathcal{Q}$. We then



(a) Algorithm $S2$. (b) Algorithm $S3$.

Figure 7.10: Unknown Centres: Reconstructed field.

$\sigma_i = 0.04$	T (sec)	$\|e\|_2$	$\sigma_i = 0.05$	T (sec)	$\|e\|_2$
Algorithm $S1$	3.1	0.045	*Algorithm* $S1$	3.9	0.012
Algorithm $S2$	3.1	0.054	*Algorithm* $S2$	3.7	0.053
Algorithm $S3$	3.1	0.048	*Algorithm* $S3$	3.7	0.028

Table 7.2: Comparison of algorithms for $m = 100$ parameters.

compute the Voronoi partition (7.58) of the sensors and use it as the required partition $\mathcal{Q}_i$.

We first show the results for approximating the field $\phi(\cdot)$ with $m = 100$ Gaussian RBFs. The centers of the Gaussian are arranged in a uniform grid over the region $\mathcal{Q}$. The reconstructed field plots for two values of σ_i (standard deviation of the Gaussian RBFs) are shown in figures 7.12, 7.13, and 7.14 with the three algorithms. To compare the various algorithms, we use the integral error (see Theorem 7.1)

$$\|e\|_2 = \int_{\mathcal{Q}} |\phi(q) - \mathcal{K}(q)^\top \hat{a}| dq,$$

where $\hat{a}$ is the final parameter estimate obtained from the given algorithm. The integral error for approximation of $\phi(\cdot)$ using $m = 100$ parameters is shown in table 7.2. The table also shows the time T in seconds at which the excitation (positive definiteness) condition is achieved. The total runtime of the estimation algorithms was $T + 20$ seconds.

The reconstructed field plots for $m = 196$ parameters is shown in figures 7.15, 7.16

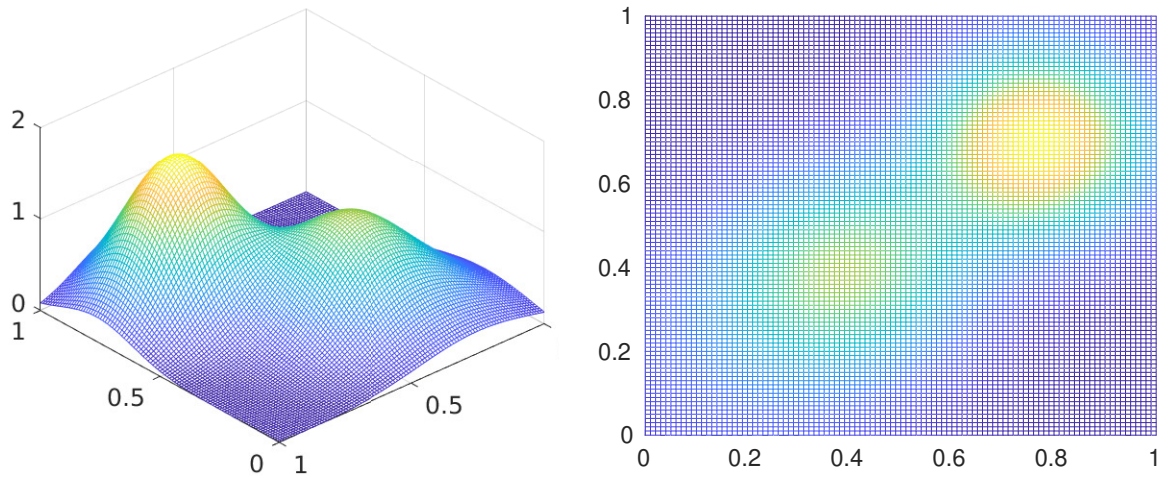
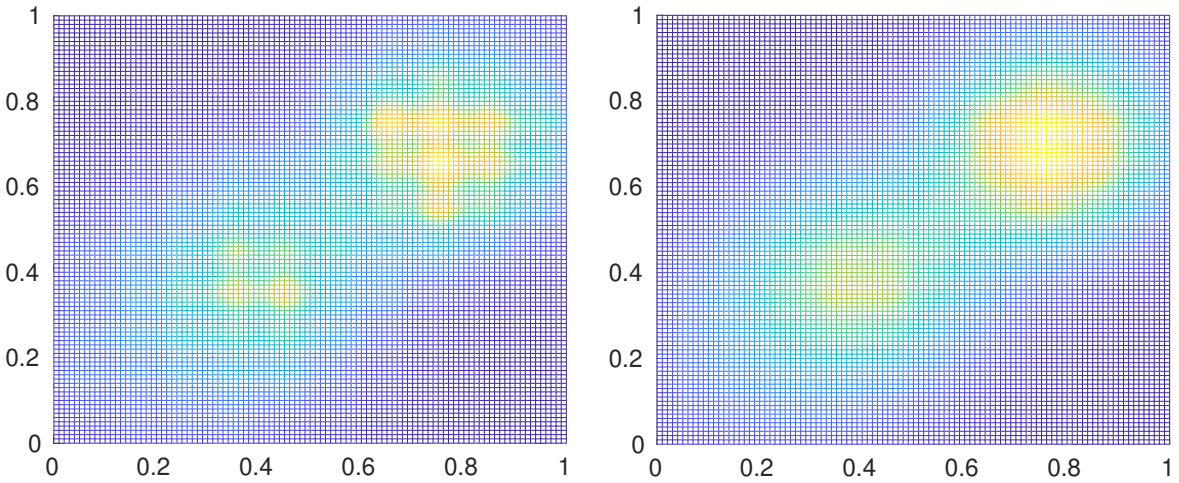


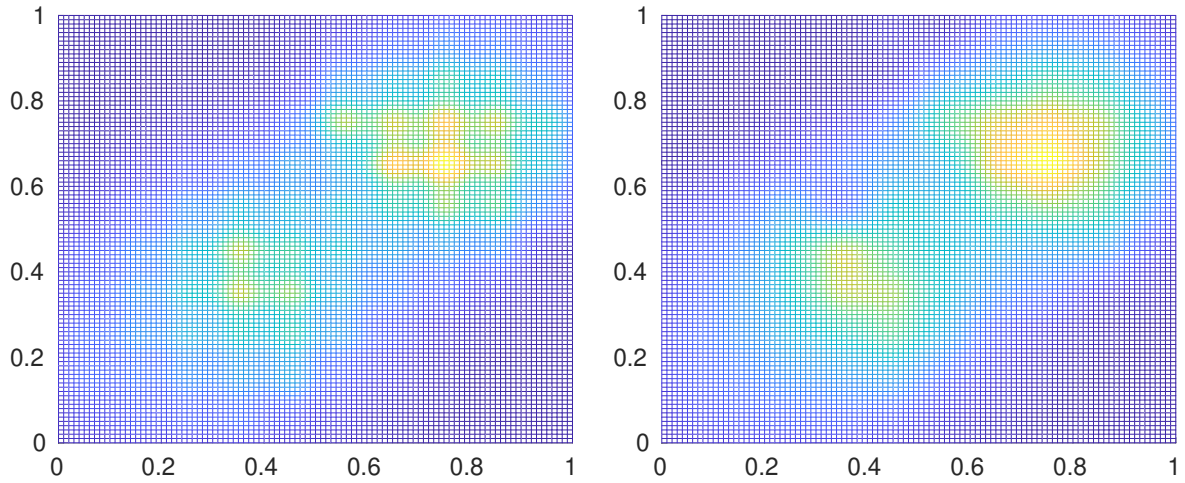
Figure 7.11: The scalar field $\phi(x, y)$ used in the simulation



(a) $\sigma_i = 0.04$.

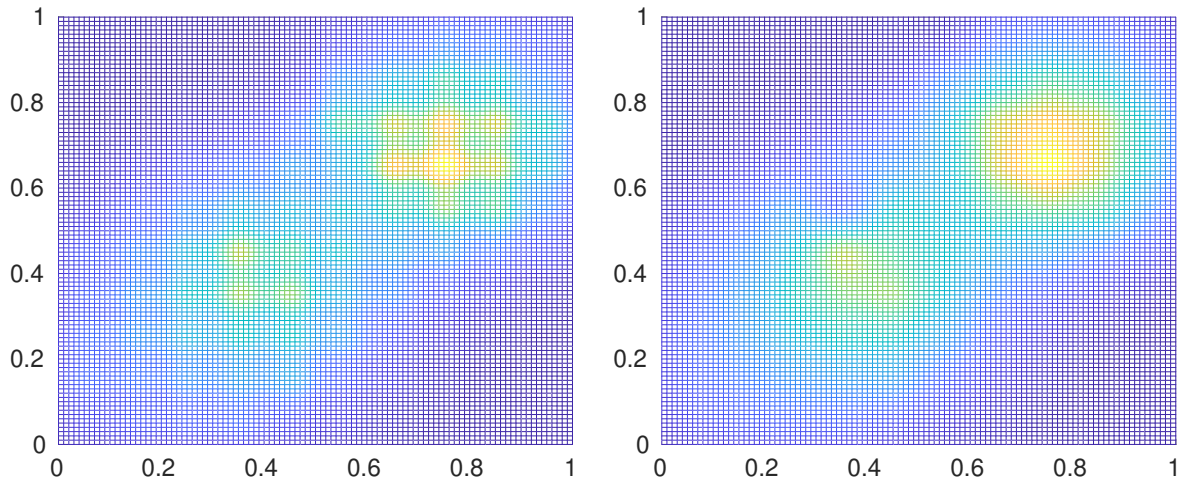
(b) $\sigma_i = 0.05$.

Figure 7.12: Reconstructed field ($m = 100$) with algorithm $S1$.



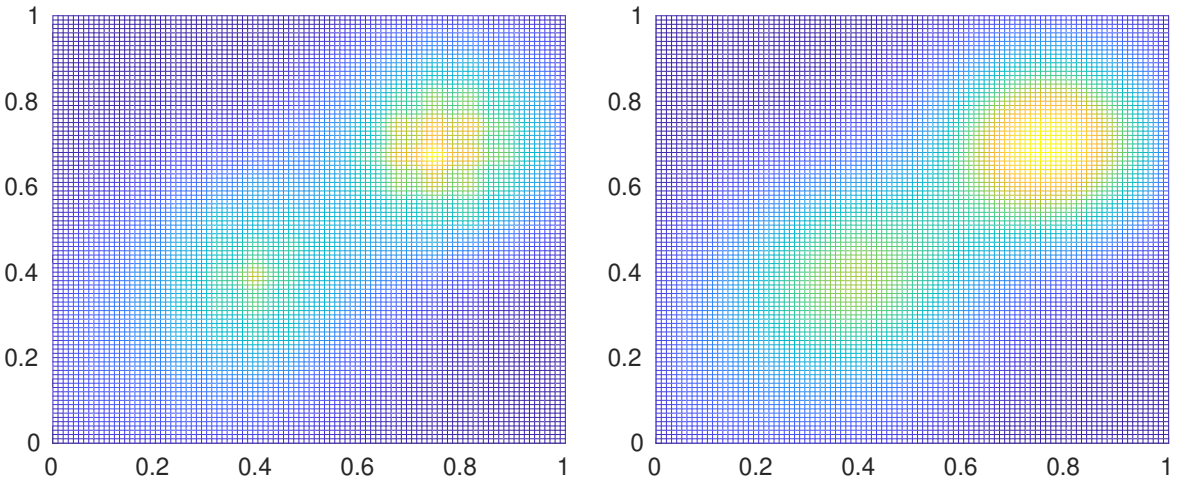
(a) $\sigma_i = 0.04$. (b) $\sigma_i = 0.05$.

Figure 7.13: Reconstructed field ($m = 100$) with algorithm $S2$.



(a) $\sigma_i = 0.04$. (b) $\sigma_i = 0.05$.

Figure 7.14: Reconstructed field ($m = 100$) with algorithm $S3$.



(a) $\sigma_i = 0.03$. (b) $\sigma_i = 0.04$.

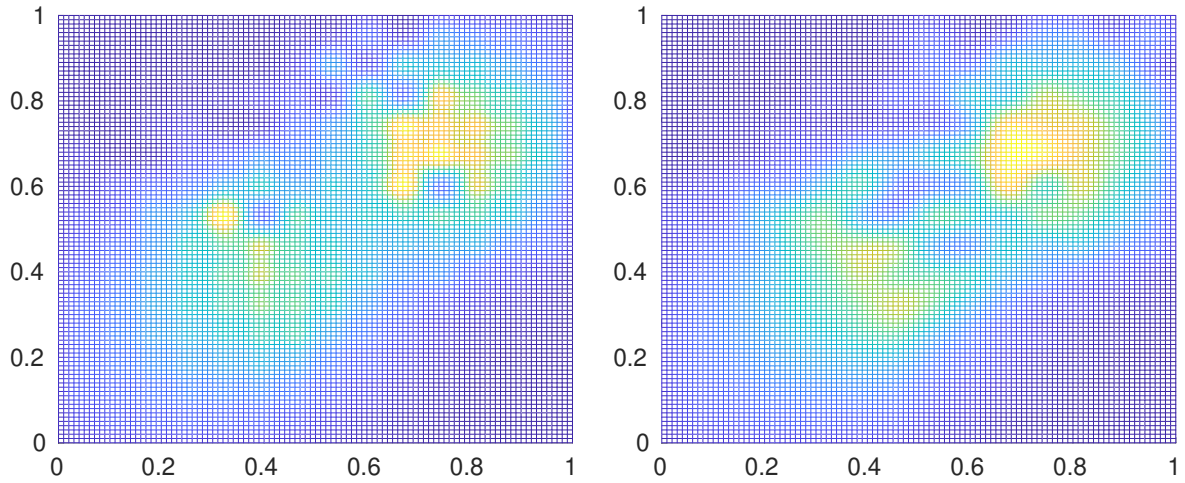
Figure 7.15: Reconstructed field ($m = 196$) with algorithm $S1$.

$\sigma_i = 0.03$	T (sec)	$\|e\|_2$	$\sigma_i = 0.04$	T (sec)	$\|e\|_2$
Algorithm $S1$	6.6	0.031	*Algorithm* $S1$	8.9	0.008
Algorithm $S2$	6.6	0.059	*Algorithm* $S2$	8.8	0.073
Algorithm $S3$	6.6	0.053	*Algorithm* $S3$	8.8	0.039

Table 7.3: Comparison of algorithms for $m = 196$ parameters.

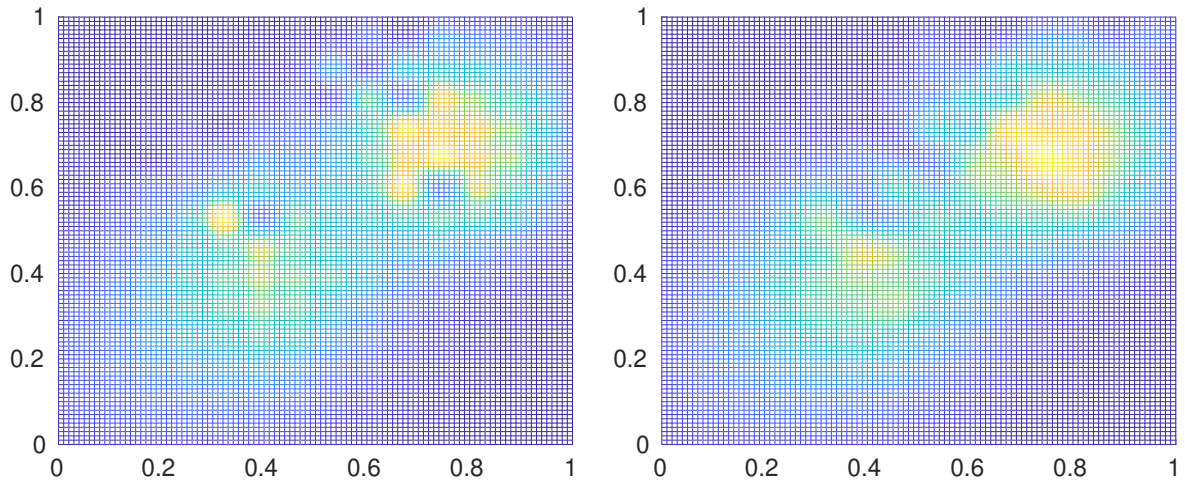
and 7.17 with the three algorithms. The comparison of various algorithms is given in table 7.3.

We see that algorithm $S1$ gives better approximation compared to the others as expected. Also, the algorithm $S3$ performs significantly better compared to algorithm $S2$. Increasing the number of parameters gives a better approximation as expected for algorithm $S1$, though for the other algorithms this is not the case. This is because for Algorithm $S1$, the error incurred in estimation is only due to the error in approximation of $\phi(\cdot)$ using the m RBFs. Using a larger number of RBFs improves the approximation (reduces the approximation error) and thus results in lower approximation error. This is not the case with algorithms $S2$ and $S3$ since for these algorithms the error incurred in estimation is not just due to the RBF approximation, but also due to the $\Delta\phi_i(\cdot)$ term, which may increase with an increase in the number of RBFs (see also Theorem 7.9). σ_i also plays an important role in the reconstruction of the original field. For $m = 100$, $\sigma_i = 0.05$



(a) $\sigma_i = 0.03$. (b) $\sigma_i = 0.04$.

Figure 7.16: Reconstructed field ($m = 196$) with algorithm $S2$.



(a) $\sigma_i = 0.03$. (b) $\sigma_i = 0.04$.

Figure 7.17: Reconstructed field ($m = 196$) with algorithm $S3$.

seems to provide a better approximation compared to $\sigma_i = 0.04$, and for $m = 196$, $\sigma_i = 0.04$ seems to provide a better approximation compared to $\sigma_i = 0.03$. Regarding computation and memory requirements, Algorithm $S1$ requires each agent to maintain 5150 integrator variables for $m = 100$ while algorithms $S2$ and $S3$ requires 405 and 478 variables, respectively (approximately $13X$ lower than $S1$). For $m = 196$, Algorithm $S1$ requires each agent to maintain 19306 integrator variables whereas Algorithm $S2$ and $S3$ requires each agent to maintain a maximum of 1539 and 1681 integrator variables, respectively. Correspondingly, one loop of updates required average computation time of 0.0024 sec, 0.0010 sec, and 0.0012 sec for algorithms $S1$, $S2$ and $S3$ respectively[1]. The corresponding computation times for $m = 196$ are 0.0055 sec, 0.0019 sec, and 0.0020 sec. To summarize, algorithm $S1$ gives better approximation compared to the others though it is more compute and memory intensive. The algorithm $S3$ also gives a good approximation requiring much less memory. It may also be noted that in many applications, we may only be interested in identifying the main features of the original field, which was successfully done in most of the cases discussed.

[1]The computation times were obtained on MATLAB® 9.4 running on an Intel core-i5 processor (7th gen) with 8GB of RAM.

Chapter 8

Conclusions and Future Work

This thesis discusses distributed adaptive control strategies for coverage control and scalar field estimation using mobile sensor networks. First, we consider the coverage control problem where the density function is unknown but can be linearly parameterized in terms of constant parameters. We develop coverage control algorithms for nonholonomic mobile robots using the locational optimization framework. We derive control and adaptation laws such that the robots converge to the estimated centroids of their Voronoi region. We also look at the case where the robot dynamics parameters are unknown. In this case, using an additional adaptation law for the robot parameters, convergence to estimated centroids is guaranteed. We verify the above algorithms using simulations. We also compare the dynamics based coverage control with the coverage control law based on kinematics alone and conclude that the dynamics based algorithm performs better.

Next, we propose a new framework for the formulation of the coverage control problem in terms of a distance function between density functions. We quantify the sensing performance of the individual mobile sensors using a *sensing function*, and the coverage problem is posed in terms of minimizing some distance function between an aggregated sensing function and the target density function. We show that the locational optimization can be formulated as the minimization of the K-L divergence with the agent sensing functions being Gaussian functions and the aggregate function formed by taking the max-operator among the agents. We also consider the notion of $\mathcal{L}^2$ coverage where coverage is obtained by minimizing the $\mathcal{L}^2$ distance between the aggregate sensing function and the target density function. We also propose a modified form of consensus-based adaptation law for the parameters, which results in faster convergence of parameters. We implement

the coverage algorithm on actual differential drive robots and present the hardware experiment results and also compare the performance of the $\mathcal{L}^2$ coverage with the locational optimization-based coverage, concentrating primarily on the performance for parameter estimation. We find that $\mathcal{L}^2$ coverage gives better estimates for the parameters overall. The locational optimization seems to start the parameter adaptation faster than the $\mathcal{L}^2$ optimization in some cases.

Finally, we discuss the problem of estimation of scalar field using mobile sensor networks. We assume that the scalar field to be estimated is a linear combination of positive definite radial basis functions. Parameter estimation algorithms are derived for the case where each mobile sensor estimates all the parameters (Algorithm $S1$), as well as the case where each mobile sensor estimates only a subset of parameters (Algorithms $S2$ and $S3$). We test the algorithms using simulations and verify that $S1$ performs better compared to $S2$ and $S3$ although computational requirements are higher. Algorithm $S3$ also performs well and is found to give a good balance between estimation performance and computational requirements.

One of the issues with the mobile sensor deployment algorithms for tasks like coverage is that they are only guaranteed to approach local minima, which can be many since the objective functions are generally non-convex, and the algorithms are generally sensitive to initial conditions. Methods for avoiding poor local minima still needs investigation. Another direction for future research is incorporating the orientation of the mobile sensors into the coverage problem. This can be much more challenging because for this case, the analysis and control law design proposed in this thesis may not extend in a straight forward manner. We also argue that such problems can be posed much more elegantly using the distance-function based approach. Another line of investigation is to try to generalize the distance function optimization framework for other multi-agent problems such as consensus/rendezvous problems, formation problems, etc.

An interesting extension to the scalar field estimation problem is to consider noisy sensors with measurement accuracy decreasing with distance from the sensors. Also, the current work on scalar field estimation assumes the scalar field to be fixed. One direction for further work in this line is to consider time-varying scalar fields. This is in general difficult since then the parameters would be time-varying and the adaptive control methods discussed in this thesis may not work. One way to tackle this issue would be

to model the time-variation using models such as partial differential equations or the like. We have also assumed in the current work that the radial basis functions used for approximating the scalar field are fixed. It would be interesting to consider the case where the centers of basis functions need not be fixed but may be adapted to better approximate the given scalar field. This could possibly lead to better approximation using the same number of parameters.

Bibliography

[1] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," vol. 20, no. 2, pp. 243–255, Apr. 2004.

[2] F. Bullo, J. Cortés, and S. Martínez, *Distributed Control of Robotic Networks*, ser. Applied Mathematics Series. Princeton University Press, 2009, electronically available at http://coordinationbook.info.

[3] Q. Du, V. Faber, and M. Gunzburger, "Centroidal Voronoi Tessellations: Applications and Algorithms," *SIAM Rev.*, vol. 41, no. 4, pp. 637–676, 1999.

[4] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," in *Proc. IEEE Int. Conf. Robot. Autom.*, May, 2002, pp. 1327–1332.

[5] L. C. a. Pimenta, V. Kumar, R. C. Mesquita, and G. a. S. Pereira, "Sensing and coverage for a network of heterogeneous robots," *Proceedings of the IEEE Conference on Decision and Control*, no. 2, pp. 3947–3952, 2008.

[6] A. Deshpande, S. Poduri, D. Rus, and G. S. Sukhatme, "Distributed coverage control for mobile sensors with location-dependent sensing models," in *Proc. IEEE Int. Conf. Robot. Autom.*, Kobe, Japan, May, 2009, pp. 2344–2349.

[7] M. Schwager, B. J. Julian, M. Angermann, and D. Rus, "Eyes in the sky: Decentralized control for the deployment of robotic camera networks," *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1541–1561, 2011.

[8] S. Bhattacharya, N. Michael, and V. Kumar, "Distributed coverage and exploration in unknown non-convex environments," *Springer Tracts in Advanced Robotics*, vol. 83 STAR, pp. 61–75, 2012.

[9] A. Deshpande, S. E. Sarma, K. Youcef-Toumi, and S. Mekid, "Optimal coverage of an infrastructure network using sensors with distance-decaying sensing quality," *Automatica*, vol. 49, no. 11, pp. 3351–3358, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.automatica.2013.07.029

[10] W. Li and C. Cassandras, "Distributed Cooperative coverage Control of Sensor Networks," in *Proceedings of the 44th IEEE Conference on Decision and Control*, 2005, pp. 2542–2547.

[11] J. Wagenpfeil, A. Trachte, T. Hatanaka, M. Fujita, and O. Sawodny, "Distributed decision making for task switching via a consensus-like algorithm," in *American Control Conference, 2009. ACC '09.*, June 2009, pp. 5761–5766.

[12] J. Choi and R. Horowitz, "Learning coverage control of mobile sensing agents in one-dimensional stochastic environments," *IEEE Transactions on Automatic Control*, vol. 55, no. 3, pp. 804–809, 2010.

[13] J. O. Oyekan, H. Hu, and D. Gu, "Bio-inspired Coverage Of Invisible Hazardous Substances in the Environment," *International Journal of Information Acquisition*, vol. 07, no. 03, pp. 193–204, Sep. 2010. [Online]. Available: http://www.worldscientific.com/doi/abs/10.1142/S0219878910002154?journalCode=ijia

[14] J. Oyekan, H. Hu, and D. Gu, "A novel bio-inspired distributed coverage controller for pollution monitoring," in *2011 IEEE International Conference on Mechatronics and Automation, ICMA 2011*, 2011, pp. 1651–1656.

[15] Z. Mi, Y. Yang, and G. Liu, "Coverage enhancement of mobile multi-agent networks while preserving global connectivity," in *Proceedings - IEEE International Conference on Robotics and Automation*, no. 1, 2011, pp. 5381–5386.

[16] K. Derr and M. Manic, "Adaptive control parameters for dispersal of multi-agent mobile Ad Hoc network (MANET) swarms," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 1900–1911, 2013.

[17] S. Rahili and W. Ren, "Game Theory Control Solution for Sensor Coverage Problem in Unknown Environment," in *IEEE Conference on Decision and Control*, 2014, pp. 1173–1178.

[18] F. Mohseni, A. Doustmohammadi, and M. B. Menhaj, "Distributed receding horizon coverage control for multiple mobile robots," *IEEE Systems Journal*, vol. pp, no. 99, pp. 1–10, 2014.

[19] C. Song, L. Liu, G. Feng, Y. Wang, and Q. Gao, "Persistent awareness coverage control for mobile sensor networks," *Automatica*, vol. 49, no. 6, pp. 1867–1873, 2013.

[20] J. M. Palacios-Gasos, E. Montijano, C. Sagues, and S. Llorente, "Distributed coverage estimation and control for multirobot persistent tasks," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1444–1460, Dec 2016.

[21] ——, "Cooperative periodic coverage with collision avoidance," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 4, pp. 1411–1422, July 2019.

[22] B. Hexsel, N. Chakraborty, and K. Sycara, "Coverage control for mobile anisotropic sensor networks," in *2011 IEEE Int. Conf. Robot. and Autom.*, May 2011, pp. 2878–2885.

[23] J. L. Ny and G. J. Pappas, "Adaptive deployment of mobile robotic networks," vol. 58, no. 3, pp. 654–666, March 2013.

[24] K. R. Guruprasad and D. Ghose, "Heterogeneous locational optimisation using a generalised Voronoi partition," *Int. J. Control*, vol. 86, no. 6, pp. 977–993, 2013.

[25] S. Miah, A. Y. Panah, M. M. H. Fallah, and D. Spinello, "Generalized non-autonomous metric optimization for area coverage problems with mobile autonomous agents," *Automatica*, vol. 80, pp. 295–299, 2017.

[26] S. D. Bopardikar, D. Mehta, and J. D. Hauenstein, "Optimal Configurations in Coverage Control with Polynomial Costs," *ArXiv e-prints*, Jan. 2018.

[27] M. Santos, Y. Diaz-Mercado, and M. Egerstedt, "Coverage control for multirobot teams with heterogeneous sensing capabilities," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 919–925, April 2018.

[28] A. Sadeghi and S. L. Smith, "Coverage control for multiple event types with heterogeneous robots," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 3377–3383.

[29] M. Schwager, J. E. Slotine, and D. Rus, "Decentralized, adaptive control for coverage with networked robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 10-14, 2007, pp. 3289–3294.

[30] M. Schwager, D. Rus, and J.-J. Slotine, "Decentralized, adaptive coverage control for networked robots," *Int. J. Rob. Res.*, vol. 28, no. 3, pp. 357–375, Mar. 2009.

[31] J.-M. Luna, R. Fierro, C. Abdallah, and J. Wood, "An adaptive coverage control algorithm for deployment of nonholonomic mobile sensors," *Decision and Control (CDC), 2010 49th IEEE Conference on*, pp. 1250–1256, 2010.

[32] V. Krishnan and S. MartÃnez, "Distributed optimal transport for the deployment of swarms," in *2018 IEEE Conference on Decision and Control (CDC)*, Dec 2018, pp. 4583–4588.

[33] Ã. Arslan, "Statistical coverage control of mobile sensor networks," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 889–908, Aug 2019.

[34] R. Nowak, U. Mitra, and R. Willett, "Estimating inhomogeneous fields using wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 6, pp. 999–1006, Aug 2004.

[35] W. Bajwa, A. Sayeed, and R. Nowak, "Matched source-channel communication for field estimation in wireless sensor network," in *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.*, April 2005, pp. 332–339.

[36] T. van Waterschoot and G. Leus, "Static field estimation using a wireless sensor network based on the finite element method," in *2011 4th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Dec 2011, pp. 369–372.

[37] M. C. Vuran and O. B. Akan, "Spatio-temporal characteristics of point and field sources in wireless sensor networks," in *2006 IEEE International Conference on Communications*, vol. 1, June 2006, pp. 234–239.

[38] H. Zhang, J. M. F. Moura, and B. Krogh, "Dynamic field estimation using wireless sensor networks: Tradeoffs between estimation error and communication cost," *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2383–2395, June 2009.

[39] D. Dardari, A. Conti, C. Buratti, and R. Verdone, "Mathematical evaluation of environmental monitoring estimation error through energy-efficient wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 7, pp. 790–802, July 2007.

[40] A. Dogandzic and B. Zhang, "Distributed estimation and detection for sensor networks using hidden markov random field models," *IEEE Transactions on Signal Processing*, vol. 54, no. 8, pp. 3200–3215, Aug 2006.

[41] R. Graham and J. Cortes, "Adaptive information collection by robotic sensor networks for spatial estimation," *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1404–1419, June 2012.

[42] I. Nevat, G. W. Peters, and I. B. Collings, "Random field reconstruction with quantization in wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 61, no. 23, pp. 6020–6033, Dec 2013.

[43] R. K. Ramachandran and S. Berman, "The effect of communication topology on scalar field estimation by large networks with partially accessible measurements," in *2017 American Control Conference (ACC)*, May 2017, pp. 3886–3893.

[44] Y. P. Bergamo and C. G. Lopes, "Scalar field estimation using adaptive networks," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2012, pp. 3565–3568.

[45] H. M. La and W. Sheng, "Distributed sensor fusion for scalar field mapping using mobile sensor networks," *IEEE Transactions on Cybernetics*, vol. 43, no. 2, pp. 766–778, April 2013.

[46] H. M. La, W. Sheng, and J. Chen, "Cooperative and active sensing in mobile sensor networks for scalar field mapping," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 1, pp. 1–12, Jan 2015.

[47] W. Wu and F. Zhang, "Cooperative exploration of level surfaces of three dimensional scalar fields," *Automatica*, vol. 47, no. 9, pp. 2044–2051, Sep. 2011. [Online]. Available: http://dx.doi.org/10.1016/j.automatica.2011.06.001

[48] B. Zhang and G. S. Sukhatme, "Adaptive sampling for estimating a scalar field using a robotic boat and a sensor network," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, April 2007, pp. 3673–3680.

[49] R. Fierro and F. Lewis, "Control of a nonholonomic mobile robot: backstepping kinematics into dynamics," *J. Robot. Syst.*, vol. 14, no. 3, pp. 149–163, 1997.

[50] H. Flanders, "Differentiation Under the Integral Sign," *Amer. Math. Monthly*, vol. 80, no. 6, pp. 615–627, 1973.

[51] Q. Du and M. Emelianenko, "Acceleration schemes for computing centroidal voronoi tessellations," *Numer. Linear Algebra Appl.*, vol. 13, no. 2-3, pp. 173–192, 2006.

[52] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 03 1951.

[53] W. Press, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 2007.

[54] H. Khalil, *Nonlinear Systems*, ser. Pearson Education. Prentice Hall, 2002.

[55] J. Park and I. W. Sandberg, "Universal approximation using radial-basis-function networks," *Neural Computation*, vol. 3, no. 2, pp. 246–257, June 1991.

[56] E. Lavretsky and K. A. Wise, *Robust Adaptive Control*. Springer, 2013.

[57] C. Micchelli, "Interpolation of scattered data: distance matrices and conditionally positive definite functions," *Constr. Approx.*, vol. 2, pp. 11–22, 1986.

[58] D. Gorinevsky, "On the persistency of excitation in radial basis function network identification of nonlinear systems," *IEEE Transactions on Neural Networks*, vol. 6, no. 5, pp. 1237–1244, 1995.

[59] M. Mesbahi and M. Egerstedt, *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010, vol. 33.

List of Publications

The work presented in this thesis appears in the following publications/pre-prints:

- Rihab A Razak, Srikant Sukumar, Hoam Chung, *Decentralized adaptive coverage control of nonholonomic mobile robots*, IFAC-PapersOnLine, vol. 49, 18, 410-415.
- Rihab A Razak, Srikant Sukumar, Hoam Chung, *Distributed Coverage Control of Mobile Sensors: Generalized Approach using Distance Functions*, 2018 IEEE Conference on Decision and Control (CDC), 3323-3328.
- Rihab A Razak, Srikant Sukumar, Hoam Chung, *Estimating Scalar Fields with Mobile Sensor Networks*, accepted for presentation at the Indian Control Conference 2019.
- Rihab A Razak, Srikant Sukumar, Hoam Chung, *Decentralized and adaptive control of multiple nonholonomic mobile robots for sensing coverage*, Internataional Journal of Robust and Nonlinear Control, vol. 28, 6, 2636-2650.
- Rihab A Razak, Srikant Sukumar, Hoam Chung, *Scalar Field Estimation with Mobile Sensor Networks*, under review, arxiv pre-print available at http://arxiv.org/abs/1907.01309.
- Rihab A Razak, Srikant Sukumar, Hoam Chung, *Distributed Adaptive Coverage Control over Planar Region using Differential Drive Robots*, arxiv pre-print available at http://arxiv.org/abs/1908.01161.