Coverage Control of Heterogeneous Multi-Agents

Submitted in partial fulfillment of the requirements
of the degree of

Master of Technology

by

Nijil George
(Roll No. 183079009)

Supervisors:
Prof. Dwaipayan Mukherjee

Department of Electrical Engineering
INDIAN INSTITUTE OF TECHNOLOGY BOMBAY
2021

Dedicated to my beloved parents.

Dissertation Approval

This dissertation entitled **Coverage Control of Heterogeneous Multi-Agents** by **Nijil George** is approved for the degree of **Master of Technology**.

Examiners:

Prof. Harish K. Pillai

7th July, 2021
....

Prof. Madhu N. Belur

Madhu N. Belur 7th July 2021
....

Supervisor:

Prof. Dwaipayan Mukherjee

....

Chairperson:

Prof. Madhu N. Belur

Madhu N. Belur 7th July 2021
....

Date: 7th July 2021

Place: Mumbai

Declaration

I declare that this written submission 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 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 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.

Nijil

Date: 7 th July 2021

Nijil George

Roll No. 183079009

Abstract

Coverage control deals with the deployment of a network of n agents into an area so that all points in that area is maintained above a particular level of coverage. Here coverage can mean that a point is in the sensing radius of a sensor. The objective of this study is to design a control law that will ensure adequate coverage by a group of heterogeneous agents in non-convex environment. The heterogeneous framework is also extended to the adaptive coverage control scenario where the relative importance of points inside the area is not known.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Coverage control for multi-agent systems have been studied widely in the past couple of decades owing to the advancements in networked systems. It finds application in search and rescue, surveillance and in distributed sensing. Coverage is said to be attained in a bounded environment when all points in the environment is serviced by at least one sensor. The sensors can vary depending on type of quantity being monitored but the algorithm that enables coverage remains the same. Coverage control can broadly be classified into static and dynamic coverage. In static coverage, the agents having locomotive capabilities are introduced into an environment, they have to autonomously position themselves in the environment such that coverage is maximized. It looks at the problem of optimal asymptotic placement of the agents in the environment and is comparable to the optimal facilities placement problem studied in operations research. In dynamic coverage, the mobile agents introduced in the environment have to keep moving to service the points in the environment. Here the problem is not about optimal placement but rather to maintain coverage of an environment where quantity being monitored is changing with time. In this case each point in the environment has an associated time dependent parameter which accumulates with time, the agents are supposed to periodically service them to keep aggregate coverage above a threshold. Both type of coverage has its own application.

Dynamic coverage is useful for cases where the quantity or event being monitored is time varying or if the number of agents is not enough to cover the entire area. This type of coverage is employed in cases where the relative importance of the points in the area changes with time. The agents have to service the points intermittently to avoid even one point from accumulate a time varying component over a threshold. [4] explores such a problem for persistent coverage for autonomous areal vehicles.

Static coverage is relevant for cases like the art gallery problem where the objective is to optimally place surveillance cameras in an art gallery such that minimum number of cameras can be used for covering all of the artworks. It also find application in placement of mobile transmission towers such that maximum number of people may get serviced by minimum number of towers. The solution for optimal placement of agents in a bounded area was first solved in a computationally efficient way in [5], here they have adapted a concept that was previously used in operations research for optimal placement of facilities. It was shown that the optimal placement is attained when the agents are positioned on the centroids of their respective voronoi cells, this optimal position is shown to be attainable through an iterative algorithm called the Lloyd's algorithm. This method is called centroidal voronoi tessellation and the control law is termed as move-to-centroid. However this method is only applicable for homogeneous agents in convex environment because the proof of asymptotic convergence only holds for convex environment. For introducing heterogeneity a more general form of voronoi diagrams called power diagrams [6] have been employed later on in [7], where the authors are considering heterogeneous agents that have variation in body size and sensor footprint size. The power diagram however lacks some desirable properties that the voronoi diagram possessed, this meant the move-to-centroid control law cannot be used directly and some heuristics had to be incorporated.

The literature on coverage control of multi-agents have two other approaches to the solution of this problem apart from the geometric (voronoi) approach. Firstly, there are literature that approach this problem from a probabilistic stand point as in [8]. This approach gives a better result but is computationally intensive. Secondly, there are methods that use artificial potential fields like in [9], here an artificial potential field is assumed over the environment.

The potential field is characterised by agents and obstacles having an artificial repulsive force among them, this will lead to the agents spreading across the environment. Static equilibrium is obtained when this repulsive force is balanced by an artificial viscous friction force which is dependant on the velocity of the agents. All these approaches however are considering homogeneous multi-agents.

Majority of the literature on coverage control caters to the homogeneous multi-agent systems. There are some literature where heterogeneity among the agents are considered, the heterogeneities considered include difference in type, footprint, directionality of the sensors and shape, speed, size of the agents. [10] looks into heterogeneous coverage control where the agents are equipped with sensor that sense different quantities. In [11] agents with different sensor footprints are dealt with. [12] deals with multi-agent team with different sensing capabilities and driftless control affine dynamics. [13] explores variation in size of the agent. [14][15][16] studies coverage control of anisotropic sensors. However, there were no literature on coverage control of heterogeneous multi-agent systems with agents having different kinematics among single integrator, double integrator and unicycle.

Another aspect of coverage control is that the knowledge of the relative importance of the points inside the region are assumed to be known. There are literature where this is assumed to be unknown to the agents. [17] gives one of the early record of trying to tackle this problem, it suggests and adaptive framework for the agents to learn the relative importance of points in the region.

This study extends the coverage control of multi-agent systems into more general cases. Firstly, it includes heterogeneity in the agents, this will be in the form of agents having different kinematics. Secondly, the heterogeneous agent frame work is extended to adaptive coverage control. Thirdly, a heuristic approach to the solution of coverage control for some non-convex environment is proposed.

1.2 Motivation

Extending the coverage control framework to a more real world case by incorporating heterogeneity in agent kinematics and relaxing the assumptions about prior knowledge of density function and convexity of area.

1.3 Research Objectives

- Find solution for coverage control of heterogeneous multi-agent systems
- Extend the heterogeneous framework to adaptive coverage control of multi-agent systems
- Find a solution for the coverage control of multi-agent systems in non-convex environment

1.4 Contribution

- The Coverage control of multi-agent systems was extended to include agents with heterogeneous agent kinematics
- The heterogeneous coverage control framework was extended to adaptive case where the assumption of relative importance of points being known to the agents is relaxed
- A heuristic way of overcoming certain type of non-convexity in the area is proposed

1.5 Outline of Dissertation

The subject matter of the dissertation is presented in the following five chapters,

- **Chapter-1** Gives an introduction to the coverage control problem and some of sub-problems relevant for this work
- **Chapter-2** Gives a literature review citing all the concepts used in the work ahead
- **Chapter-3** Formulates and solves the problem of coverage control of heterogeneous multi-agents. The results are verified through simulations
- **Chapter-4** Formulates and solves the problem of adaptive coverage control of heterogeneous multi-agents. The results are verified through simulations

- **Chapter-5** Formulates and proposes a heuristic solution to the problem of coverage control of heterogeneous multi-agents in non-convex environment. The results are verified through simulations
- **Chapter-6** Summarizes the study and points out the contributions. Also discuss the possible future work

This page was intentionally left blank.

Chapter 2

Review of Literature

This chapter will discuss the existing literature on decentralized coverage control of multi-agent systems. The chapter is divided into multiple sections based on the specific problem that is being tackled under the broader decentralized coverage control problem.

2.1 Decentralized Coverage Control

The decentralized coverage control of multi-agent systems have been a topic of research for many decades, however a computationally efficient and mathematically backed solution was proposed first by [5]. It achieved this by partitioning the area to be covered into cells which are assigned to individual agents to be serviced.

The problem is formulated as an optimal positioning problem. Consider an environment $Q \in \mathbb{R}^2$. Define a probability density function $\phi : Q \to \mathbb{R}_+$. ϕ represents the relative importance of coverage at a point in Q. Let $f : \mathbb{R}_+ \to \mathbb{R}_+$ represent the precision of the sensor reading as a function of distance from the agent. f will be a monotonically decreasing function of $|q - p_i|$, where $q \in Q$ and p_i is the position of i^{th} agent.

Consider a partition $\mathscr{F} = \{F_1, ..., F_2\}$ of Q which is based on the position of the agents p_i. The coverage cost function is defined as

$$\mathscr{H}(P, \mathscr{F}) = \frac{1}{2}\sum_{i=1}^{n}\int_{F_i} f(|q-p_i|)\,\phi(q)dq \tag{2.1}$$

The partition of Q can be based on Voronoi tessellation $V(P) = \{V_1, ..., V_n\}$ based on sensor positions $P = \{p_1, ..., p_n\}$. The i-th voronoi cell is defined as

$$V_i = \left\{q \in Q \,|\, |q-p_i| \leq |q-p_j|, \forall j \neq i\right\} \tag{2.2}$$

A more generalized version of the voronoi partition can be defined as

$$V_i = \left\{q \in Q \,|\, g(|q-p_i|) \leq g(|q-p_j|)\,|, \forall j \neq i\right\} \tag{2.3}$$

Where $g : \mathbb{R}_+ \to \mathbb{R}_+$ is a function of $|q-p_i|$ chosen for particular need.

By substituting F_i with V_i for the equation for the coverage cost function becomes

$$\mathscr{H}(P, V) = \frac{1}{2}\sum_{i=1}^{n}\int_{V_i} f(|q-p_i|)\,\phi(q)dq \tag{2.4}$$

Mass of the i^{th} voronoi cell is given by

$$M_{V_i} = \int_{V_i} \phi(q)dq \tag{2.5}$$

Centroid position of the i^{th} voronoi cell is given by

$$C_{V_i} = \frac{1}{M_{V_i}}\int_{V_i} q\phi(q)dq \tag{2.6}$$

The optimal coverage problem is now defined as a minimization problem to minimize equation (2.4) with respect to the agent positions p_i. The partial derivative of the cost function with respect to the positions is given by.

$$\frac{\partial \mathscr{H}(P)}{\partial p_i} = M_{V_i}(p_i - C_{V_i}) \tag{2.7}$$

It can be observed that the $p_i = C_{V_i}$ is an optimal solution for the above problem. It corresponds to positioning the agents at their respective voronoi centroids, this configuration is called *centroidal voronoi configuration*. It can be achieved iteratively using an algorithm called Lloyd's algorithm. The algorithm is based on the descent in the gradient direction of the cost function. It is given by

$$\dot{p}_i = -\frac{\partial \mathscr{H}(P)}{\partial p_i} \tag{2.8}$$

For a single integrator agent the following control law can be used to achieve the required configuration.

$$u_i = -k_{\text{prop}} M_{V_i}(p_i - C_V) \tag{2.9}$$

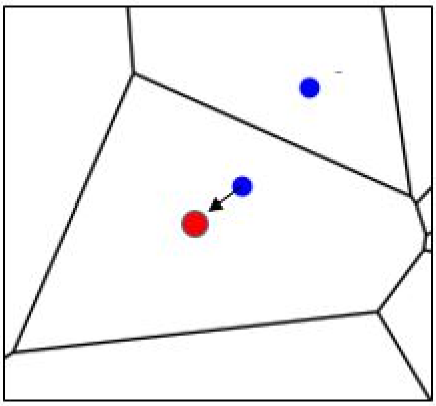

Figure 2.1: Gradient descent from p_i (blue) to centroid (red)

Over the years there have been many extension to the above literature

2.2 Heterogeneity Among Agents

Majority of the literature on coverage control pertains to a homogeneous group of agents. The few that consider heterogeneity among agents look into the variation in size, shape and type of sensors. This work will consider variation in the dynamics of the sensor. For that the three sensor dynamics will be introduced in this section.

2.2.1 Single Integrator

Single integrator agent is the simplest agent kinematics and most of the literature on coverage control pertains pertains to the same. The agent is able to take fast manoeuvres. The state of a single integrator is given by $p = [x\ y]^T$. Single integrator agents dynamics is modelled by

$$\dot{p}_i = u_i, \tag{2.10}$$

where u_i is the control input and it corresponds to the velocity of the agent. The control law as given by [5] is

$$u_i = -k(p_i - \hat{C}_{V_i}), \tag{2.11}$$

where $k \in \mathbb{R}_+$ is a proportional control gain.

2.2.2 Double Integrator

Double integrator agent is an agent kinematics with acceleration input, it resembles a physical mobile robot more when compared to a single integrator kinematics, since physical systems are driven by actuators which give force output. The states of a double integrator are $p_1 = [x\ y]^T$

and $p_2 = \dot{p}_1 = [\dot{x}\ \dot{y}]^T$. Double integrator agent dynamics are modelled by

$$\ddot{p}_i = u_i, \tag{2.12}$$

this can be written as two first order equations

$$\begin{aligned} \dot{p}_{i1} &= p_{i2} \\ \dot{p}_{i2} &= u_i, \end{aligned} \tag{2.13}$$

where u_i is the control input which corresponds to acceleration of the agent. The control law given in [18] is

$$\begin{aligned} u_i &= -k_1 \hat{M}_{V_i}(p_{i1} - \hat{C}_{V_i}) - k_2 p_{i_2} \\ &= -k_1 \hat{M}_{V_i}(p_i - \hat{C}_{V_i}) - k_2 \dot{p}_i, \end{aligned} \tag{2.14}$$

where k_1 and k_2 are the proportional and derivative control gains.

When compared to single integrators, the double integrator agent is more sluggish and may have oscillations before settling down.

2.2.3 Unicycle

Unicycle kinematics more accurately represents a physical robot than the previous two types. In this kinematical model in addition to its position the agent has a heading angle as part of its state, which is the case with most physical robots which are not omini-directional and have non-holonomic constrains. The non-holonomic constrains disallow the agent from motion orthogonal to its heading. The state of a unicycle model is given by $p = [x\ y\ \phi]^T$, where x and y are the position coordinates and ϕ is the heading angle. Heading angle is measured counter clock wise to the x-axis of the global frame of reference. Unicycle dynamics is given by

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{pmatrix} = \begin{pmatrix} \cos\phi & 0 \\ \sin\phi & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} u \\ \omega \end{pmatrix}, \tag{2.15}$$

where u and ω are the linear and angular velocity respectively.

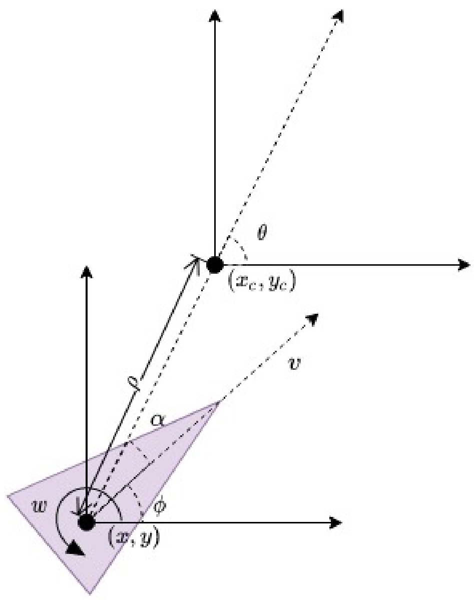

Figure 2.2: States of a unicycle in polar form

Unicycle model can be represented in the polar form as well, from a control perspective it is found to be more beneficial. The states of a unicycle when converted to the polar form is $p = [\rho\ \alpha\ \theta]^T$ and is related to the states in Cartesian coordinates by

$$\begin{aligned} \rho &= \sqrt{x^2 + y^2} \\ \alpha &= \text{atan2}(y, x) - \theta + \pi \\ \theta &= \phi + \theta \end{aligned} \tag{2.16}$$

Unicycle agent dynamics in polar form is given by

$$\begin{pmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} -u\cos\alpha \\ -\omega + u\frac{\sin\alpha}{\rho} \\ u\frac{\sin\alpha}{\rho}, \end{pmatrix} \tag{2.17}$$

with

$$
\begin{aligned}
\alpha &= \theta - \phi \\
\dot{\phi} &= \omega,
\end{aligned}
\tag{2.18}
$$

where the control inputs u and ω are velocity and angular velocity of the agent respectively.

ρ_i represents the position error between the agent and the target point, ϕ_i is the heading angle and α_i is the angle between the principle axis of the agent and the vector error ρ_i. The control law given in [19] and [20] is

$$
\begin{pmatrix} u \\ \omega \end{pmatrix} = \begin{pmatrix} (\gamma \cos\alpha)\rho \\ k\alpha + \gamma \frac{\cos\alpha \sin\alpha}{\alpha}(\alpha + h\theta) \end{pmatrix}, \tag{2.19}
$$

where k, γ, h are positive gain constants.

2.3 Decentralized Adaptive Coverage Control

The classical coverage control assumes that the distribution density function is known. In Adaptive coverage control it is assumed that the density function $\phi(q)$ is not known apriori, the agents try to learn the density function based on their sensor measurements. It is formulated in [17] and requires the following assumptions

Assumption 1 (Matching Conditions)[17]: The density function can be represented as

$$
\phi(q) = \mathscr{K}(q)^T a, \tag{2.20}
$$

where $\mathscr{K} : Q \to \mathbb{R}^m_+$ is a vector of basis function known to all agents and $a \in \mathbb{R}^m_+$ is constant parameter vector unknown to the agents. This assumption means that $\phi(q)$ can be represented as a weighted combination of set of basis functions $\mathscr{K}(q)^T = [\mathscr{K}_1(q), \mathscr{K}_2(q), ..., \mathscr{K}_m(q)]$.

Assumption 2 (Lower Bound)[17]:

$$
a(j) \geq \beta \; \forall j = 1, .., m, \tag{2.21}
$$

where $a(j)$ is the j^{th} element of the parameter vector a and $\beta > 0$ is a known real bound. This assumption implies a lower bound on $\phi(q)$ over Q. It ensures that $\phi(q)$ never becomes zero which may lead to C_{V_i} being undefined.

The agents are assumed to have the capability of measuring the density function $\phi(q)$ at its position. Let i^{th} agent's estimate of the parameter vector and the density function be $\hat{a}_i(t)$ and $\hat{\phi}(q) = \mathscr{K}(q)^T \hat{a}_i$ respectively. Similar to (2.5)(2.6) the estimated mass, first moment and centroid of the Voronoi cell V_i corresponding i^{th} agent are given by the following equations

$$\begin{aligned} \hat{M}_{V_i} &= \int_{V_i} \hat{\phi}_i(q)dq, \\ \hat{L}_{V_i} &= \int_{V_i} q\hat{\phi}_i(q)dq, \\ \hat{C}_{V_i} &= \frac{1}{\hat{M}_{V_i}} \int_{V_i} q\hat{\phi}_i(q)dq \end{aligned} \tag{2.22}$$

The parameter error is defined as

$$\tilde{a}_i = \hat{a}_i - a_i, \tag{2.23}$$

and the mass moment errors

$$\begin{aligned} \tilde{M}_{V_i} &= \int_{V_i} \mathscr{K}(q)^T \tilde{a}_i dq = \hat{M}_{V_i} - M_{V_i}, \\ \tilde{L}_{V_i} &= \int_{V_i} q\mathscr{K}(q)^T \tilde{a}_i dq = \hat{L}_{V_i} - L_{V_i}, \\ \tilde{C}_{V_i} &= \frac{\tilde{L}_{V_i}}{\tilde{M}_{V_i}}, \end{aligned} \tag{2.24}$$

note that $\tilde{C}_{V_i} \neq \hat{C}_{V_i} - C_{V_i}$. The actual and the estimated error vectors are defined as $e_i = C_{V_i} - p_i$ and $\hat{e}_i = \hat{C}_{V_i} - p_i$ respectively. The adaptation law [17] for $\hat{a}_i$ is defined as

$$\dot{\hat{a}}_i = \Gamma\left(\dot{\hat{a}}_{pre_i} - I_{proj_i}\dot{\hat{a}}_{pre_i}\right) \tag{2.25}$$

with

$$\dot{\hat{a}}_{pre_i} = -F_i\hat{a}_i - (\Lambda_i\hat{a}_i - \lambda_i) \tag{2.26}$$

where $\Gamma \in \mathbb{R}^{m\times m}$ is a gain matrix with non-zero diagonal elements representing the gains for each component. The variables F_i, Λ_i and λ_i are defined as follows,

$$F_i = \left[\int_{V_1} K(q)\left(q - \hat{C}_{V_i}\right)^T dq\right]\dot{p}_i \tag{2.27}$$

$$\Lambda_i = \int_0^t w(\tau)\mathscr{K}i(\tau)\mathscr{K}i(\tau)^T d\tau \tag{2.28}$$

$$\lambda_i = \int_0^t w(\tau)\mathscr{K}i(\tau)\phi i(\tau) d\tau \tag{2.29}$$

and matrix I_{proj_i} is defined as

$$I_{proj_i}(j) = \begin{cases} 0 & \text{for } \hat{a}_i(j) > \beta \\ 0 & \text{for } \hat{a}_i(j) = \beta \text{ and } \dot{\hat{a}}_{pre_i} \geq 0 \\ 1 & \text{otherwise} \end{cases} \tag{2.30}$$

This projection ensures that the parameter vector $\hat{a}$ is lower bounded by β. The function $w(t) \in \mathscr{L}^1$ is called a *weighing function* which simulates the parameter convergence of the adaptation law.

2.4 Coverage Control in Non-Convex Environment

In literature, non-convex environments are tackled in three ways: by manipulating the probability function that is related to the power diagrams, mapping from non-convex to an almost convex environment using diffeomorphism and by using different measure of distance such as geodesic used in [21]. [22] uses the method of manipulating the probability function to overcome non-convexity that is imposed by obstacles.

Here the various approaches used for solving coverage control problem for non-convex environment available in literature will be mentioned. The non-convexity of the environment may lead to two cases where the regular centroidal voronoi algorithm fails:

- Centroid of a voronoi cell lies outside the bounded area
- The straight line path to the centroid is blocked by an obstacle.

The following methods try to overcome the above instances of failure.

2.4.1 Change in behaviour of the agent

In this method the coverage control for non-convex environment is implemented by switching between the regular centroidal voronoi algorithm and obstacle avoidance algorithm. One such solution is proposed in [1] where the agents follow the input from centroidal voronoi algorithm when they have free path between their current position and the centroid but will switch to tangent bug algorithm when there is obstacle in their path.

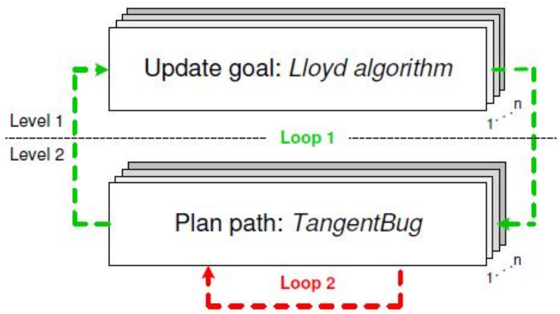

Figure 2.3: Two level algorithm used to avoid obstacles [1]

This method will give a sub-optimal solution for the non-convex environment. Here, if the centroid of a cell is within the allowed area but the path is blocked by an obstacle the tangent bug algorithm will be able to avoid the obstacle and reach the centroid. However, if the centroid of the voronoi cell lies inside an obstacle or outside the allowed area then the tangent bug algorithm will position the agent such that the position will be a projection of the target position which will contribute least to the cost function.

2.4.2 Diffeomorphism

In this approach a diffeomorphism is used to transform the original non-convex environment to an almost convex environment in which the centroidal voronoi algorithm can be used effectively. [2] and [23] uses such an algorithm to find a solution for coverage in non-convex environment. Here they have considered a simply connected area and relaxed the assumption of convexity. They are using Riemann Mapping Theorem to transform the non-convex area into a convex one and use inverse transform to get back the solution in the original space.

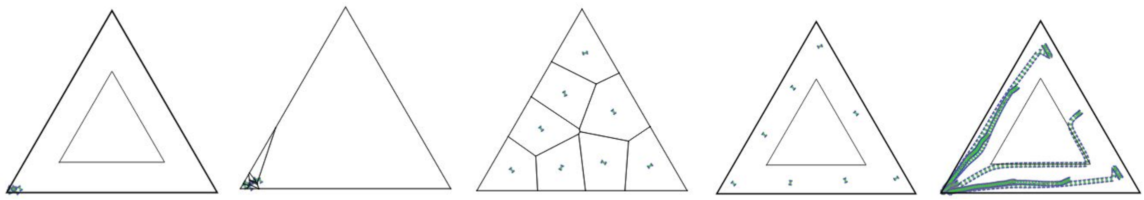

Figure 2.4: Original nonconvex area, transformed area, voronoi partition in transformed space, agent positions in original space [2]

This method has drawbacks as the the diffeomorphism for the non-convex region must be known beforehand, which means every time the algorithm has to be used in a different environment a new diffeomorphism has to be defined. Further more the concept of neighbours of an agent becomes hard to define in a transformed environment, ie some point that are far apart in the original environment may get mapped to nearby points. Also the cost function being solved is not the original one and the solution may not lead to the optimal for the actual function.

2.4.3 Geodesic distance metric

In [24][3] non-convexity is overcome using geodesic distance instead of the euclidean distance that is used in convex case. The geodesic distance, denoted as $d(x,y)$, is introduced between any two points in $x,y \in \Omega$ as the length of the shortest paths s(x,y), entirely contained in Ω.

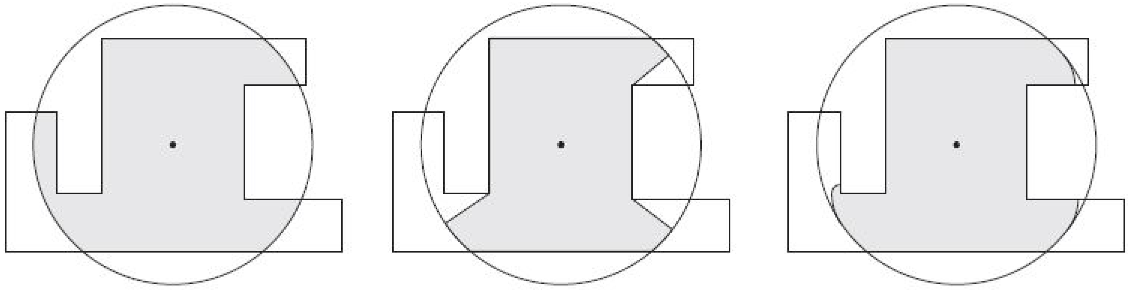

Figure 2.5: Comparison of sensor foot print when considering Euclidean distance (left), Visibility (middle), Geodesic distance (right). Figure taken from [3]

Here a different version of voronoi partition known as Geodesic Voronoi partition is introduced where instead of the euclidean distance geodesic distance is used, it is defined as $V_i^g = \{q \in \Omega \mid d(q,p_i) \leq d(q,p_j), j = 1,\dots n\}$ This partitioning gives a more realistic partitioning of the area, it considers the range of the sensor used for coverage.The voronoi partitioning based on Euclidean distance might lead to disjoint voronoi cells, this means some part of the voronoi cell assigned to an agent will not be under its sensing.

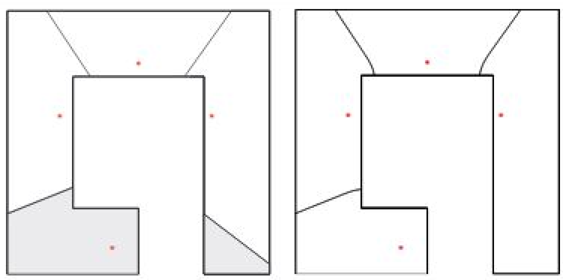

Figure 2.6: Portioning of non-convex region using Euclidean Voronoi (left) and Geodesic Voronoi (right) [3]

In the above figure it can be noticed that the voronoi partition based on Euclidean distance is giving a disjoint cell in the left figure. However the Geodesic voronoi partition is giving rise to connected voronoi cells.

2.4.4 Manipulating the weighing function

Another approach that is used to overcome the non-convex environment is to change the weighing function (density function). The density function is defined over the area to be covered, this function has traditionally been static. It was used to give relative importance of the points in the area, there are literature where this function is modified so that coverage can be attained in non-convex environment. In [25] the density function is made to be dependant on the positions p_i of the agents in addition to the points $q \in Q$. This would mean that the density function will not be static and will change during algorithm execution. Cost function becomes

$$\mathscr{H}(P,V) = \frac{1}{2}\sum_{i=1}^{n}\int_{V_i} f\left(|q-p_i|\right)\phi(q,p_i)dq$$

The density function also changes with the position of the agents in the environment, and so the agents will converge to the optimal position.

This approach can be explored more to solve the non-convex coverage problem. It gives a lot more flexibility compared to the previous approaches, the function used to define the density function needs to be designed carefully.

This page was intentionally left blank.

Chapter 3

Coverage Control of Heterogeneous Multi-Agents in Convex Environment

The coverage control of homogeneous multiagents have been introduced in section 2.1 and is similar to [5]. It was for a single integrator agent and a simple proportional feedback controller was used. When it comes to heterogeneous multi-agents, the controller needs to be different for different agents based on kinematics.

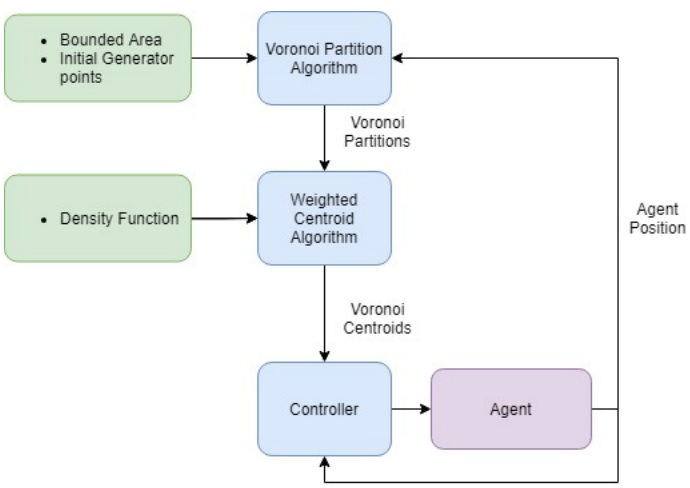

Figure 3.1: Controller topology for coverage control of homogeneous multi-agents

In this chapter agent heterogeneity is introduced into the the classical decentralized coverage control setting. The heterogeneity of the agents is in the form of kinematics the agents are following. It will find application in cases where coverage needs to be attained by a group of robots that follow different kinematics.

3.1 Problem Formulation

The problem of heterogeneous coverage control is formulated similar to [5]. Consider an convex bounded environment $Q \in \mathbb{R}^2$. Define a distribution density function $\phi : Q \to \mathbb{R}_+$. ϕ represents the relative importance of coverage at a point in Q. Let $f : \mathbb{R}_+ \to \mathbb{R}_+$ represent the precision of the sensor reading as a function of distance from the agent. f will be a monotonically decreasing function of $|q - p_i|$, where $q \in Q$ and p_i is the position of i^{th} agent. In this study $f = \|q - p_i\|^2$.
In addition to the setting it is assumed that the there are n agents that are introduced into the area, out of these n agents x are single integrators, y are double integrators and z are unicycles, such that $n = x + y + z$.
The optimal coverage problem can be formulated as a locational optimization problem which minimizes the cost function given by

$$\mathscr{H}(P,V) = \frac{1}{2}\sum_{i=1}^{n}\int_{V_i}\|q - p_i\|^2\phi(q)dq \tag{3.1}$$

The optimal coverage problem is now defined as a minimization problem to minimize equation (3.1) with respect to the agent positions p_i. The partial derivative of the cost function with respect to the positions is given by [5]

$$\frac{\partial \mathscr{H}(P)}{\partial p_i} = M_{V_i}(p_i - C_{V_i}), \tag{3.2}$$

where M_{V_i} and C_{V_i} are the mass and the centroid given by (2.5) and (2.6) respectively.

It can be observed that the $p_i = C_{V_i}$ is a local optimum for (3.1). It corresponds to positioning the agents at their respective voronoi centroids (centroidal voronoi configuration).

3.2 Methodology

In the previous section the problem of heterogeneous coverage control was formulated and the condition for optimal placement of agents have been stated. In this section the control laws required to drive the agents to their desired location are explored.

A two level nested control is proposed:

- **Outer Controller:** Computes the voronoi partition of the environment based on the position of the agents (generator points). It will also compute the centroid of each voronoi cell and give it as the set point for the inner controller.
- **Inner Controller:** Takes the set point given by the outer controller and generates the required control input to be given to the agents so as to minimize the error between the setpoint and the current position of the agent. This controller will be different for the different agent kinematics.

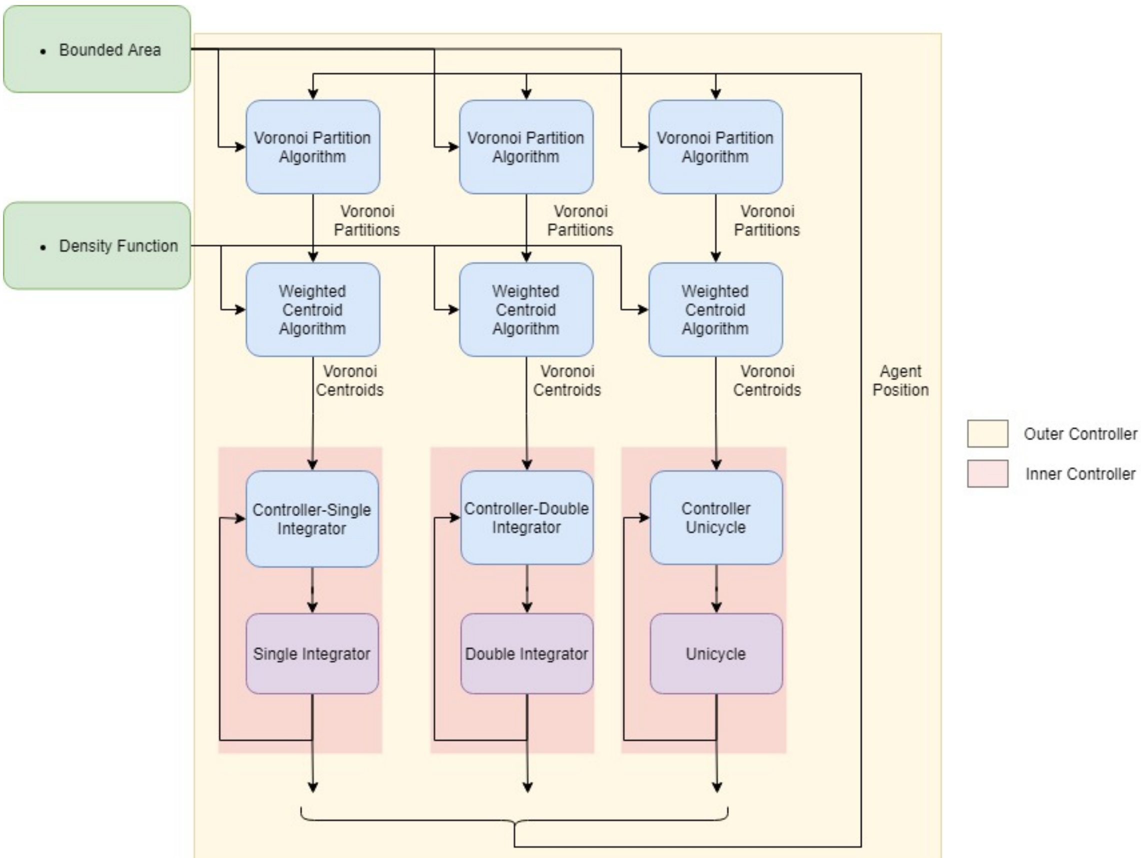

Figure 3.2: Controller topology for coverage control of homogeneous multi-agents

The outer controller is similar to the controller that is used in [5] and will estimate the voronoi cell of each agent in a decentralized manner. It does this by identifying the relative location (both distance and bearing) of each voronoi neighbour of an agent. Once the neighbours' locations are identified the agent will be able to estimate its own voronoi cell. The voronoi boundary between to agents is given by the perpendicular bisector of the line joining the position of the two agents.

The inner controller will differ based on the agent kinematics. The proposed controllers for the three types of kinematics considered in the study are

- **Single integrator:** For a single integrator agent given by the dynamics (2.10) the controller can be a simple proportional controller. It will be similar to the gradient descent in the direction of maximum gradient. The control law for the i^{th} agent is given by

$$u_i = -k(p_i - C_{V_i}), \tag{3.3}$$

  where k is the a positive proportional gain.

- **Double integrator:** For a double integrator agent with kinematics given by (2.12) the controller has to have a differential term to counter act the second order dynamics. The proportional and derivative control gains have to be tuned in order to reduce oscillations. The control law for the i^{th} agent is given by

$$\begin{aligned} u_i &= -k_1 M_{V_i}(p_{i_1} - C_{V_i}) - k_2 p_{i_2} \\ &= -k_1 M_{V_i}(p_i - C_{V_i}) - k_2 \dot{p}_i, \end{aligned} \tag{3.4}$$

  where k_1 is the positive proportional gain and k_2 is the positive derivative gain.

- **Unicycle:** For a unicycle agent given by the dynamics (2.15) the controller can be based on geometric projection of the heading vector onto the error vector. For the i^{th} agent, let centroid be $C_{V_i} = (x_c, y_c)$, current position $p_i = (x_i, y_i)$ and current heading ϕ_i. Now the control law can be stated as

$$\begin{aligned} v &= [h_i.f_d]h_i \\ \omega &= [h_i^{\perp}]^T.f_d, \end{aligned} \tag{3.5}$$

where

$$\begin{aligned} h_i &= [cos\phi \ sin\phi]^T \\ h_i^{\perp} &= [-sin\phi \ cos\phi]^T \\ f_d &= [(x_c - x_i) \ (y_c - y_i)]^T \end{aligned} \tag{3.6}$$

Unicycle represented by the polar form (2.17) can be controlled using polar controller given by

$$\begin{aligned} u_i &= (\gamma \cos \alpha_i)\, \rho_i \\ \omega_i &= k_3 \alpha_i + \gamma \frac{\cos \alpha_i \sin \alpha_i}{\alpha_i} (\alpha_i + h\theta_i) \end{aligned} \tag{3.7}$$

The two controllers for the unicycle will be compared using simulations in the next section.

3.3 Stability Analysis

In this section the stability analysis of the heterogeneous multi-agent system will be done using a Lyapunov like method. Considering x single integrators, y double integrators and z unicycles following dynamics in equations (2.10), (2.12) and (2.15) respectively and control laws according to equations (3.3), (3.4) and (3.7) respectively. Consider the Lyapunov function [5][19] candidate

$$\begin{aligned} \mathscr{V} = & \left(\frac{1}{2} \sum_{i=1}^{x} \int_{V_i} \|q - p_i\|^2 \phi(q) dq \right) + \\ & \left(\frac{k_1}{2} \sum_{i=x+1}^{x+y} \int_{V_i} \|q - p_i\|^2 \phi(q) dq + \frac{1}{2} \sum_{i=x+1}^{x+y} \dot{p}_i^2 \right) + \\ & \left(\frac{1}{2} \sum_{i=x+y+1}^{x+y+z} \int_{V_i} \|q - p_i\|^2 \phi(q) dq + \frac{1}{2} \sum_{i=x+y+1}^{x+y+z} (\alpha_i^2 + h\theta_i^2) \right) \\ = & (\mathscr{V}_1) + (\mathscr{V}_2) + (\mathscr{V}_3) \end{aligned} \tag{3.8}$$

In the above equation $\mathscr{V}$ is positive definite so are $\mathscr{V}_1$, $\mathscr{V}_2$, $\mathscr{V}_3$ separately, each of them correspond to an agent of particular kinematics. Since there are no coupling between the agents they can be separated.

$$\dot{\mathscr{V}} = \dot{\mathscr{V}_1} + \dot{\mathscr{V}_2} + \dot{\mathscr{V}_3} \tag{3.9}$$

Finding $\dot{\mathscr{V}_1}$ and substituting $\dot{p}_i$ with equation (3.3)

$$\begin{aligned}\dot{\mathscr{V}_1} &= \frac{1}{2}\sum_{i=1}^{x} \frac{\partial}{\partial p_i}\left(\int_{V_i} \|q-p_i\|^2 \phi(q)dq\right)\dot{p}_i \\ &= -k\sum_{i=1}^{x} M_{V_i}\|p_i - C_{V_i}\|^2 \leq 0\end{aligned} \tag{3.10}$$

Similarly finding $\dot{\mathscr{V}_2}$ and substituting $\ddot{p}_i$ with equation (3.4)

$$\begin{aligned}\dot{\mathscr{V}_2} &= \frac{1}{2}\sum_{i=x+1}^{x+y} \frac{\partial}{\partial p_i}\left(k_1\int_{V_i} \|q-p_i\|^2 \phi(q)dq + \dot{p}_i^2\right)\dot{p}_i \\ &= \sum_{i=x+1}^{x+y} k_1 M_{V_i}(p_i - C_{V_i})\dot{p}_i + \ddot{p}_i\dot{p}_i \\ &= \sum_{i=x+1}^{x+y} k_1 M_{V_i}(p_i - C_{V_i})\dot{p}_i + (-k_1 M_{V_i}(p_i - C_{V_i}) - k_2)\dot{p}_i \\ &= -k_2\sum_{i=x+1}^{x+y} \dot{p}_i^2 \leq 0\end{aligned} \tag{3.11}$$

For unicycle agents from Figure 2.2

$$\begin{aligned}C_{V_i} - p_i &= \begin{pmatrix} x_c - x_i \\ y_c - y_i \end{pmatrix} = \begin{pmatrix} \rho_i cos\theta_i \\ \rho_i sin\theta_i \end{pmatrix} \\ &= \begin{pmatrix} \rho_i cos(\phi_i + \alpha_i) \\ \rho_i sin(\phi_i + \alpha_i) \end{pmatrix}\end{aligned} \tag{3.12}$$

Finding $\dot{\mathscr{V}}_3$ and substituting $\dot{p}_i = u_i$, $\dot{\alpha}_i$ and $\dot{\theta}_i$ with equation (3.7) and $(C_{V_i} - p_i)$ with equation (3.12)

$$\begin{aligned}
\dot{\mathscr{V}}_3 &= \frac{1}{2} \sum_{i=x+y+1}^{x+y+z} \frac{\partial}{\partial p_i} \left(\int_{V_i} \|q - p_i\|^2 \phi(q) dq \right) \dot{p}_i + (\alpha_i \dot{\alpha}_i + h\theta_i \dot{\theta}_i) \\
&= \sum_{i=x+y+1}^{x+y+z} M_{V_i} (p_i - C_{V_i})^T \dot{p}_i + (\alpha_i \dot{\alpha}_i + h\theta_i \dot{\theta}_i) \\
&= \sum_{i=x+y+1}^{x+y+z} -M_{V_i} \begin{pmatrix} \rho_i cos(\phi_i + \alpha_i) \\ \rho_i sin(\phi_i + \alpha_i) \end{pmatrix} \cdot \begin{pmatrix} (\rho_i cos\alpha_i) cos(\phi_i) \\ (\rho_i cos\alpha_i) sin(\phi_i) \end{pmatrix} \\
&\qquad + \left(\alpha_i \left(-\omega_i + u_i \frac{sin\alpha_i}{\rho_i} \right) + h\theta_i u_i \frac{sin\alpha_i}{\rho_i} \right) \\
&= - \sum_{i=x+y+1}^{x+y+z} \left(M_{V_i} \rho_i^2 \gamma cos^2 \alpha_i \right) + k_3 \alpha_i^2 \leq 0
\end{aligned} \tag{3.13}$$

And so from equations (3.9), (3.10), (3.11), (3.13)

$$\dot{\mathscr{V}} = \dot{\mathscr{V}}_1 + \dot{\mathscr{V}}_2 + \dot{\mathscr{V}}_3 \leq 0 \tag{3.14}$$

Since $\mathscr{V}$ is lower bounded and $\dot{\mathscr{V}}$ is negative semidefinite it can be concluded that $\dot{\mathscr{V}} \to 0$ as $t \to \infty$ by Lyapunov-like lemma.

3.4 Simulations

Simulations were done on Matlab and Simulink, the agent dynamics and the inner controllers were modelled in Simulink and they were invoked by a Matlab script which produced the control signal from the outer controller. More details regarding the implementation of simulations are given in the Appendix B. Here the simulation results will be given.

Table 3.1: Simulation parameters used for heterogeneous coverage control

Parameter	Value	Parameter	Value
k	6	k_1	3
k_2	2	k_3	1
γ	3	h	1
ω_{max}	$\pm\pi$	n	15

A 10 unit square area was considered as the region to be covered, $n = 15$ agents were introduced at random positions in the area. A distribution density function was defined over the area and each agent has the knowledge of the bounds of the region and the distribution density function. It is assumed that the agents know their own positions in the area (perfectly localized) and have the capability to find the location of their neighbours, using which each agent has the capability to compute its voronoi partition. Additionally, $n = 15$ agents include: $x = 5$ single integrator agents, $y = 5$ double integrator agents and $z = 5$ unicycle agents. Specifications regarding the initial position and the distribution density function used vary with experiment and they are mentioned under each subheading.

3.4.1 Experiment 1: Uniform distribution with agents randomly placed

In this experiment the distribution density function is assumed to be uniform distribution, i.e. all points inside the are are equally important. The initial location of the agents can be anywhere in the region considered. Agent numbered 1 through 5 are single integrators, numbered 6 through

10 are double integrators and numbered 11 through 15 are unicycles

$$\phi(q) = \frac{1}{\text{area of Q}} \, \forall \, q \in Q \tag{3.15}$$

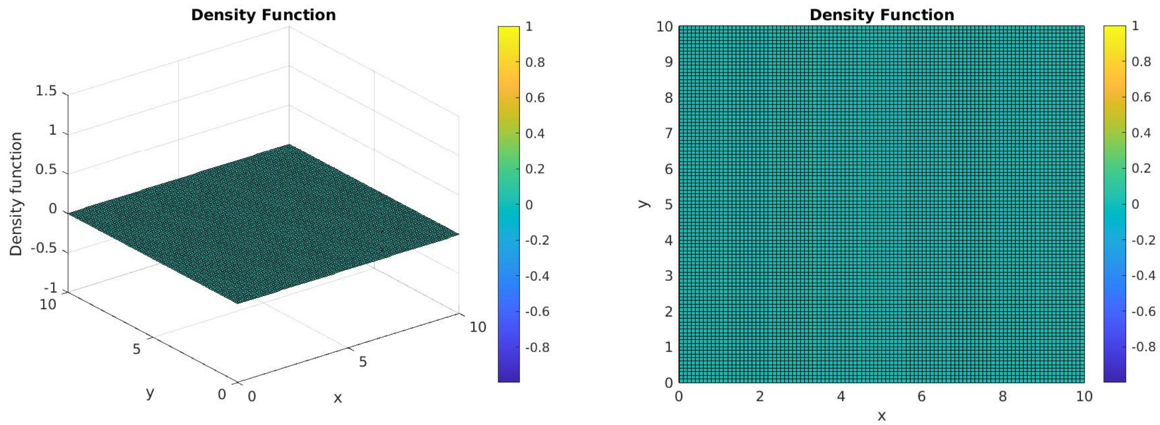

Figure 3.3: Uniform density distribution function 3d and 2d views

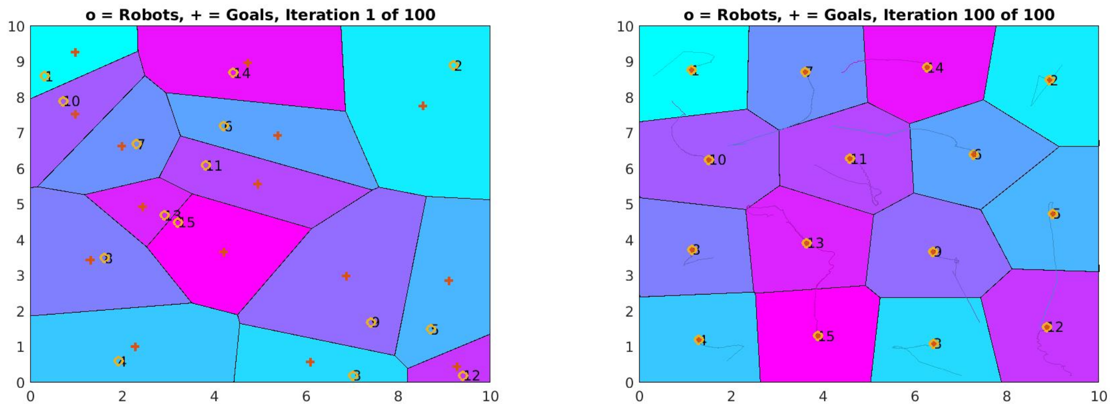

Figure 3.4: $n = 15$ agents initial position randomly (left) and final position with path (right)

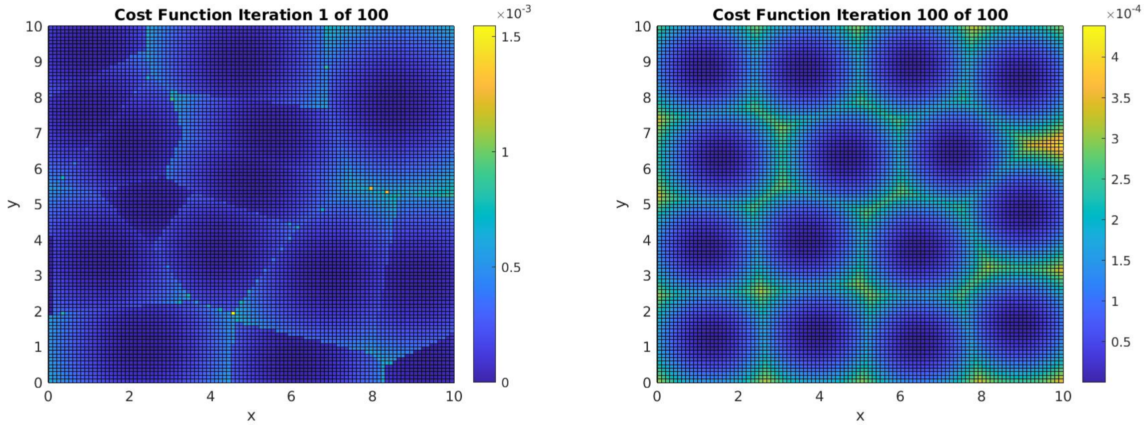

Figure 3.5: Distribution of cost function in the region initial (left) and final (right)

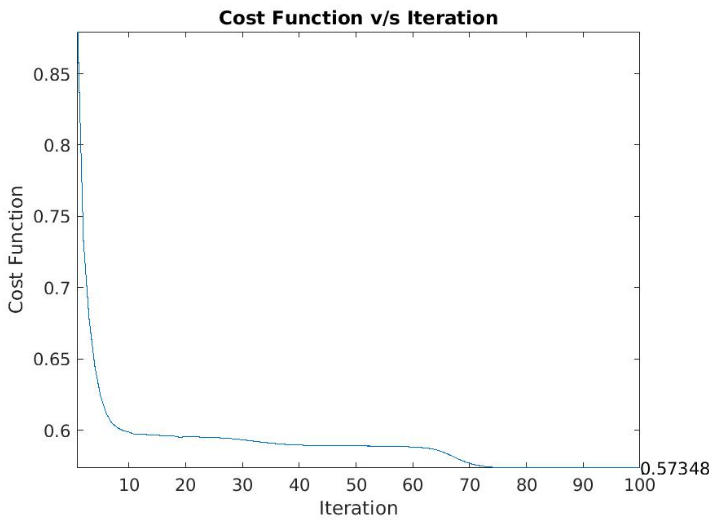

Figure 3.6: Cost function variation with iteration

Observations: The agents that are initially randomly positioned spread out to reduce the coverage cost function. As can be seen from Figure 3.4 the agents are positioned more or less uniformly in the region. The cost function reduced from 0.87974 to 0.57348 (35% reduction in the initial cost).

3.4.2 Experiment 2: Uniform distribution with agents starting from unit square

In this experiment the distribution density function is assumed to be uniform distribution, i.e. all points inside the area are equally important. The distribution is same as that given by equation (3.15) and Fig.3.3. The initial location of all the agents are inside a unit square in the region considered.

3.4.2.1 Setting A: Unicycles initially near the boundary

In this setting the unicycle agents are intentionally given initial positions near the boundary of the region.

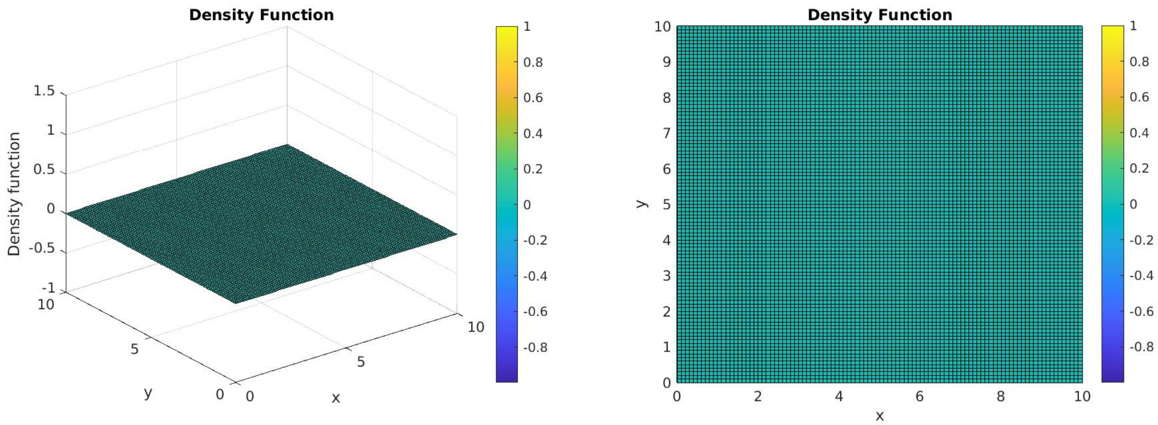

Figure 3.7: Uniform density distribution function 3d and 2d views

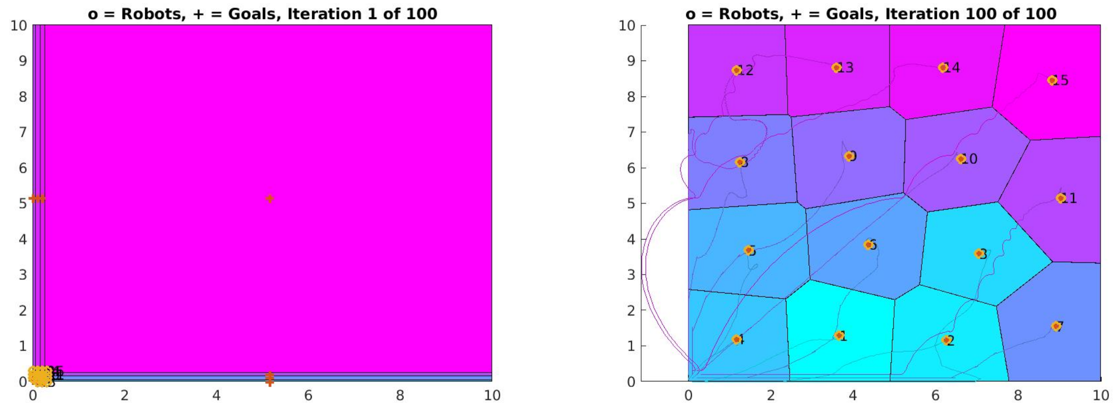

Figure 3.8: $n = 15$ agents initial position inside a unit square (left) and final position with path (right)

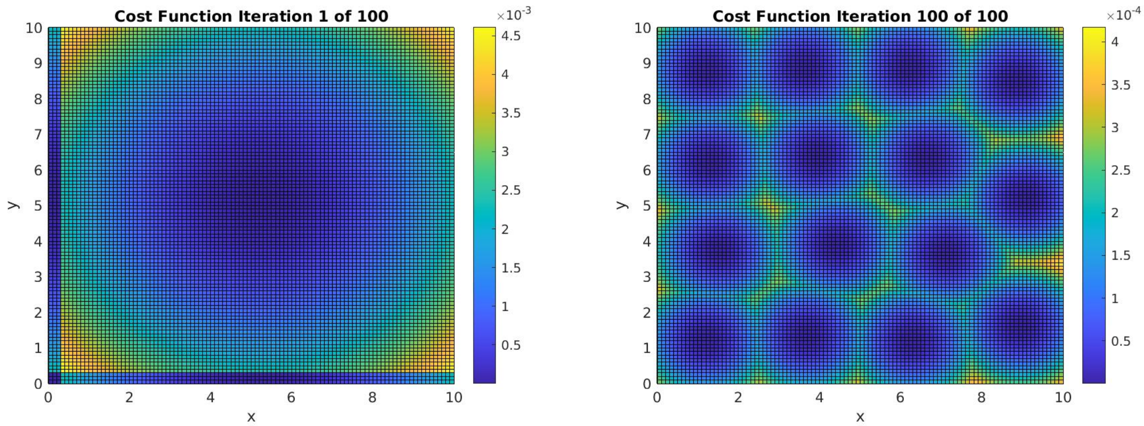

Figure 3.9: Distribution of cost function in the region initial (left) and final (right)

Observations: The agents that are initially positioned inside a unit square spread out to reduce the coverage cost function. As can be seen from Figure 3.8 the agents are positioned more or less uniformly in the region. The cost function reduced from 7.7648 to 0.57366 (92.6% reduction in the initial cost) without considering extra penalty for agents moving outside the valid region. Ideally the agents should lie inside the valid region, but because of non-straight line path followed by the unicycle agents the agents go outside the valid region.

3.4.2.2 Setting B: Unicycles initially away from the boundary

In this setting the unicycle agents are intentionally given initial positions away from the boundary of the region.

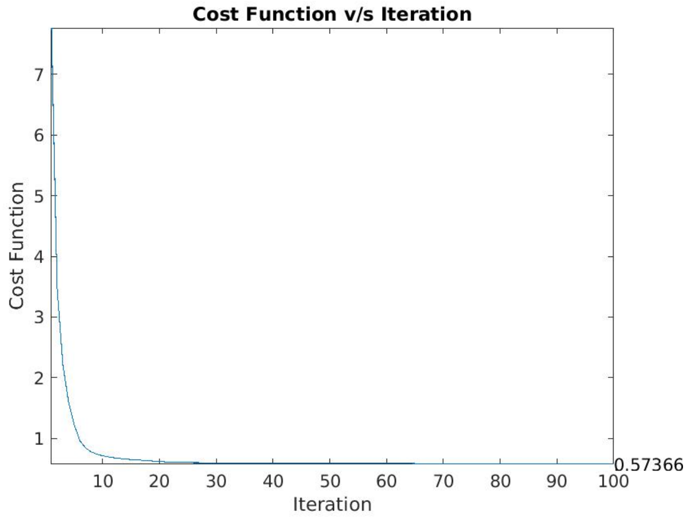

Figure 3.10: Cost function variation with iteration

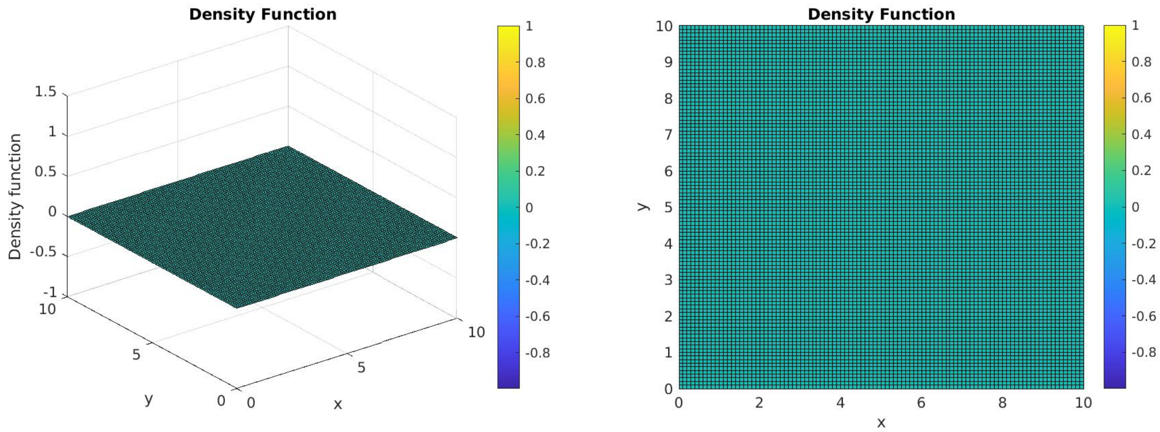

Figure 3.11: Uniform density distribution function 3d and 2d views

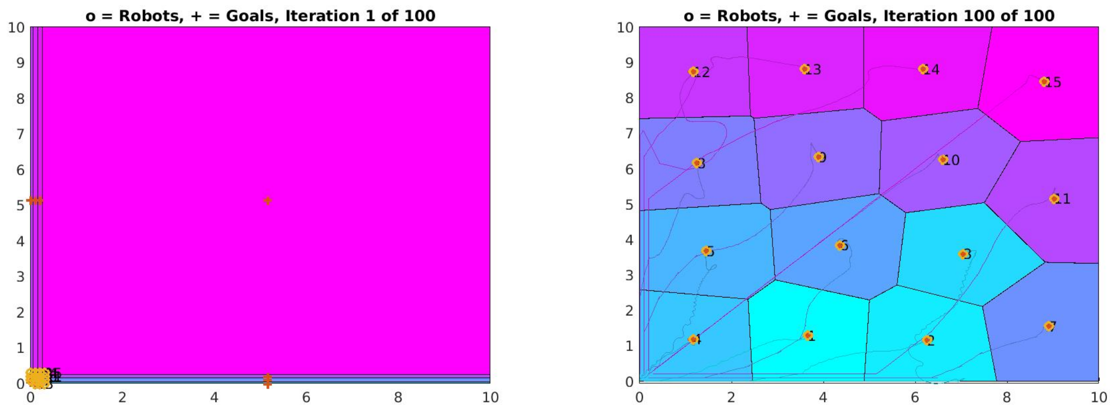

Figure 3.12: $n = 15$ agents initial position inside a unit square (left) and final position with path (right)

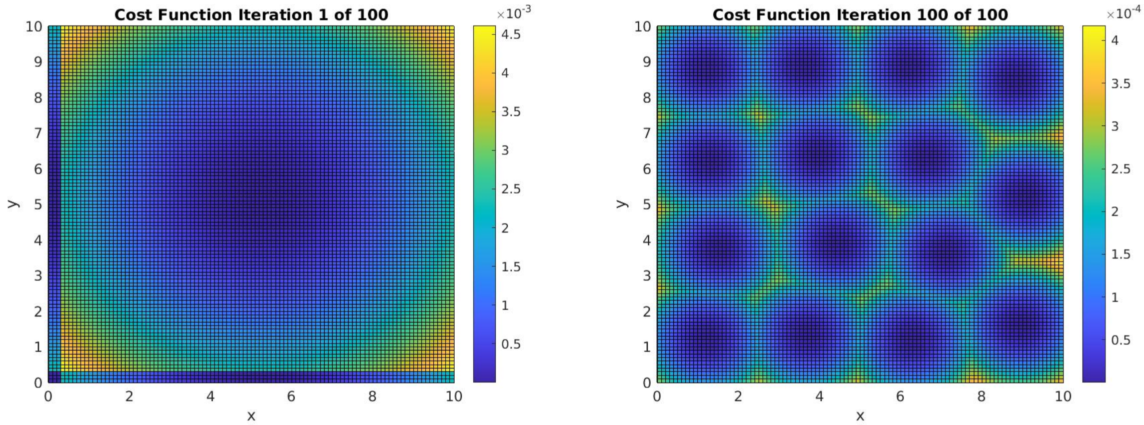

Figure 3.13: Distribution of cost function in the region initial (left) and final (right)

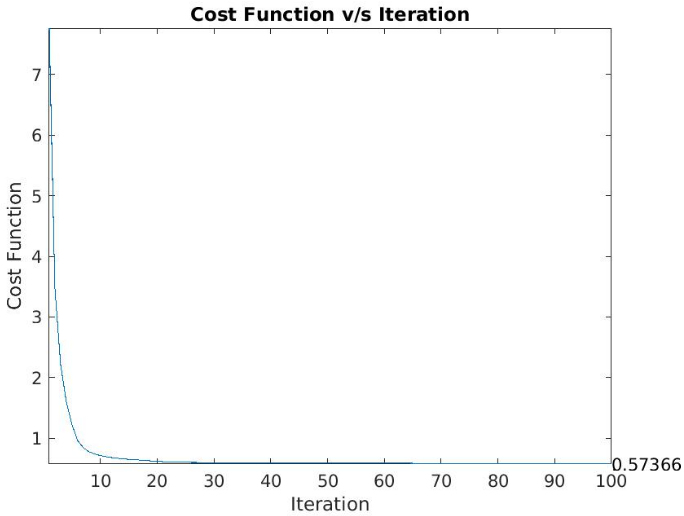

Figure 3.14: Cost function variation with iteration

Observations: The agents that are initially positioned inside a unit square spread out to reduce the coverage cost function. As can be seen from Figure 3.12 the agents are positioned more or less uniformly in the region. The cost function reduced from 7.7648 to 0.57366 (92.6% reduction in the initial cost) without considering extra penalty for agents moving outside the valid region. Comparing the observations of setting A and B it can be inferred that it is better to initially place unicycle agents towards the interior of the region rather than near to the boundary. This will prevent agents from moving outside the valid region.

3.4.3 Experiment 3: Bimodal gaussian distribution with agents randomly placed

In this experiment the distribution density function is assumed to be bimodal gaussian distribution, i.e. all points inside the area are not equally important. Here the denisty function is a sum of two gaussian functions both with $\sigma = 1.8$ and centered at $\mu_1 = (1.67, 1.67)$ and $\mu_2 = (8.33, 8.33)$ .The initial location of the agents can be anywhere in the region considered.

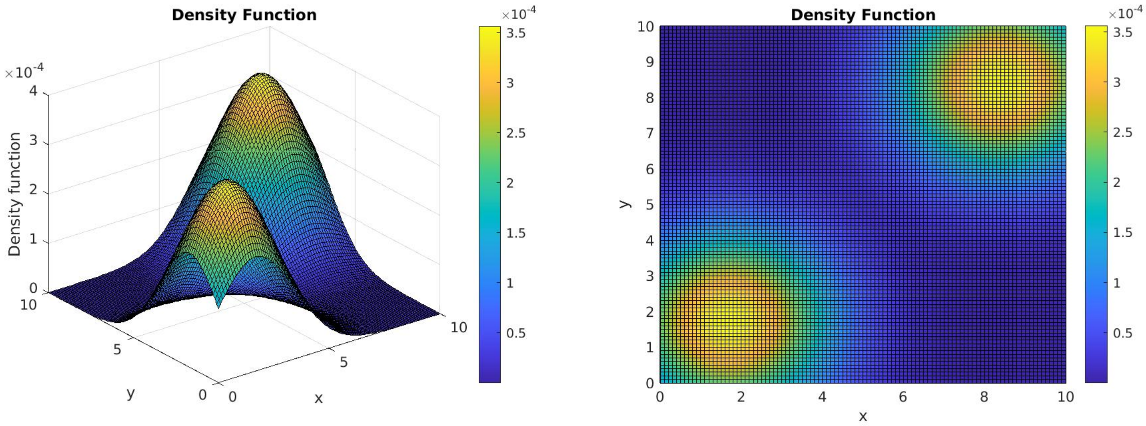

Figure 3.15: Bimodal density distribution function with $\sigma = 1.8$ and means $\mu_1 = (1.67, 1.67)$ and $\mu_2 = (8.33, 8.33)$ 3d and 2d views

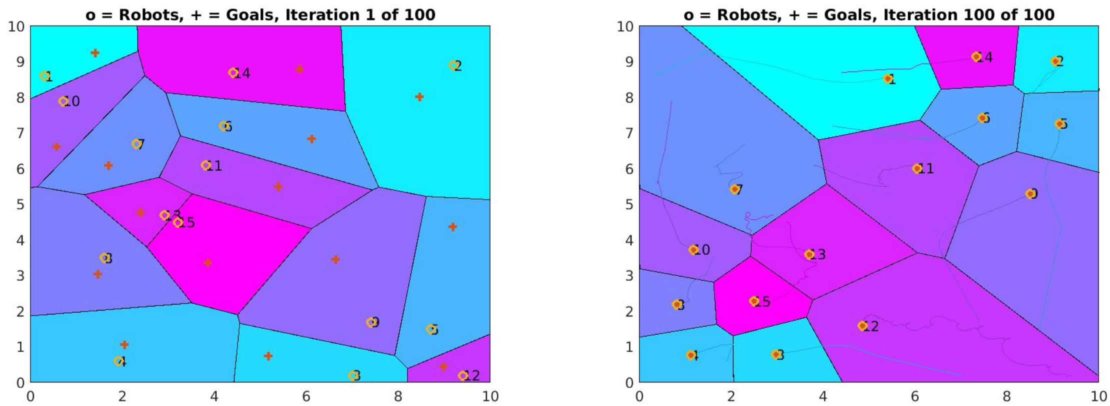

Figure 3.16: $n = 15$ agents initial position randomly (left) and final position with path (right)

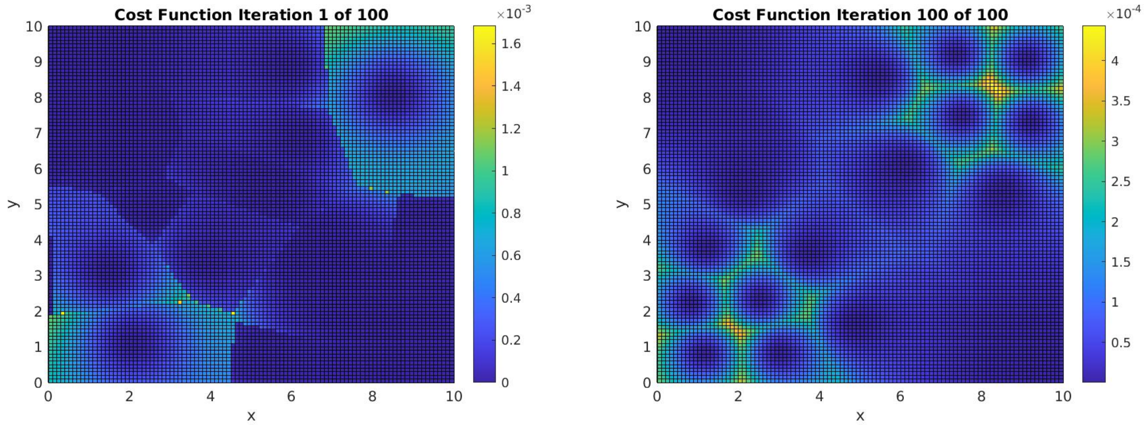

Figure 3.17: Distribution of cost function in the region initial (left) and final (right)

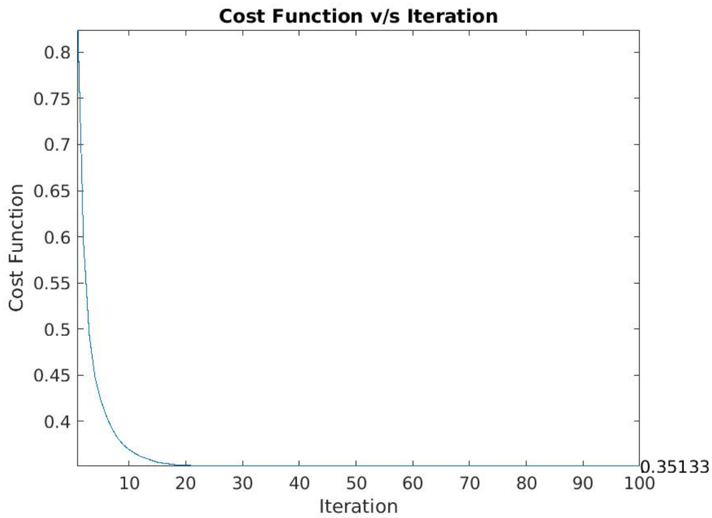

Figure 3.18: Cost function variation with iteration

Observations: It can be seen that there is a higher concentration of agents near the points in the region where the density function is higher. The cost function reduced from the initial 0.82409 to 0.35133 (57% reduction).

3.4.4 Experiment 4: Bimodal gaussian distribution with agents starting from unit square

In this experiment the distribution density function is assumed to be bimodal gaussian distribution, i.e. all points inside the are are not equally important. The initial location of the agents is inside a unit square in the region considered.

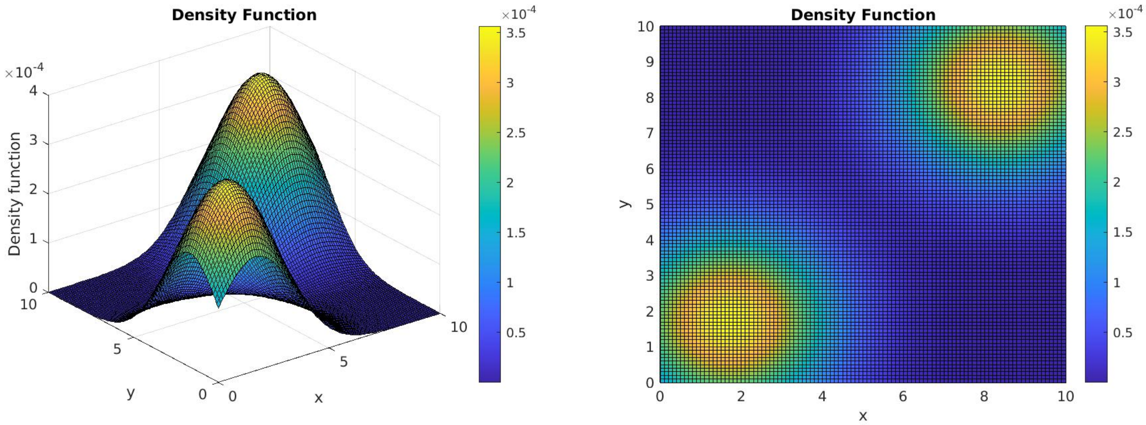

Figure 3.19: Bimodal density distribution function with $\sigma = 1.8$ and means $\mu_1 = (1.67, 1.67)$ and $\mu_2 = (8.33, 8.33)$ 3d and 2d views

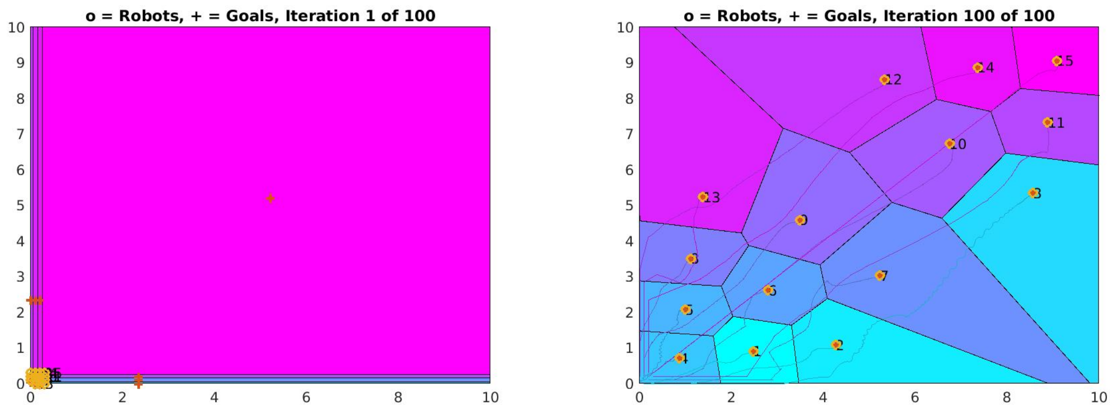

Figure 3.20: $n = 15$ agents initial position inside a unit square(left) and final position with path (right)

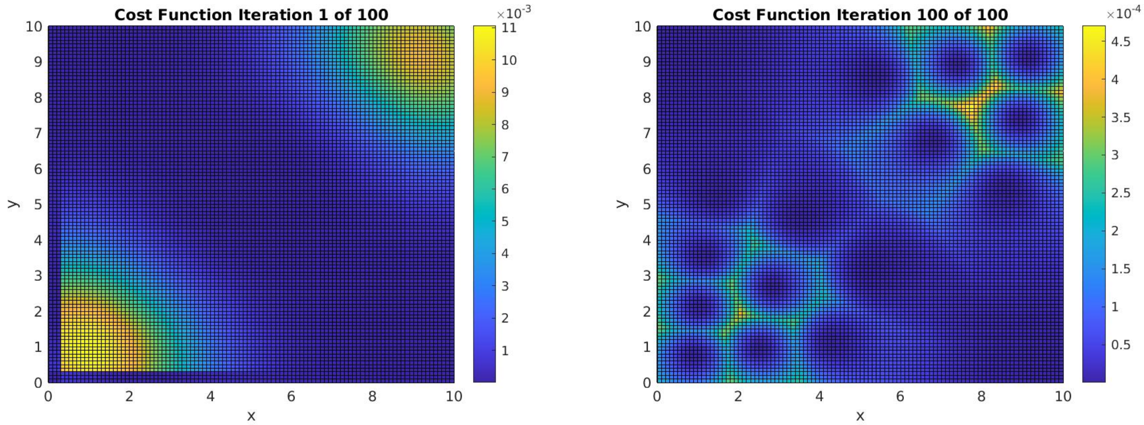

Figure 3.21: Distribution of cost function in the region initial (left) and final (right)

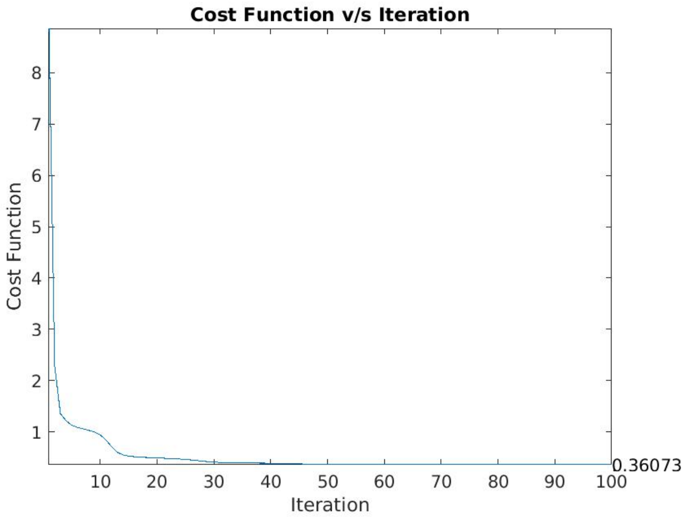

Figure 3.22: Cost function variation with iteration

Observations: It can be seen that there is a higher concentration of agents near the points in the region where the density function is higher. The cost function reduced from the initial 8.8616 to 0.36073 (96% reduction).

3.5 Results

- The proposed controllers worked fine for coverage control of a heterogeneous group of agents when the agents are initially placed towards the interior of the region. However, when the agents are initially placed near to the boundaries some of the unicycle and double integrator agents tend to move outside the boundaries of the region. This can be attributed to the non-straight line paths followed by the unicycle and the oscillations by the double-integrator agent. These agents are not following the move-to-centroid control law that was designed for the single-integrator agent.

- Out of the two controllers proposed for unicycle agents, the polar controller performed superior to the projection based controller in terms of less curvier path followed. This is in line with the assumption that the agents follow straight line path and are always within their voronoi cell.

This page was intentionally left blank.

Chapter 4

Adaptive Coverage Control of Heterogeneous Multi-Agents in Convex Environment

The adaptive coverage control of homogeneous multi-agents have been introduced in section 2.3 and is similar to [17]. It was for a single integrator agent and a simple proportional feedback controller was used. When it comes to heterogeneous multi-agents, the controller needs to be different for different agents based on kinematics.

In this chapter, agent heterogeneity is introduced into the the adaptive decentralized coverage control setting. The heterogeneity of the agents is in the form of kinematics the agents are following. A Lyapunov based stability proof is provided for the proposed solution and verified using simulations.

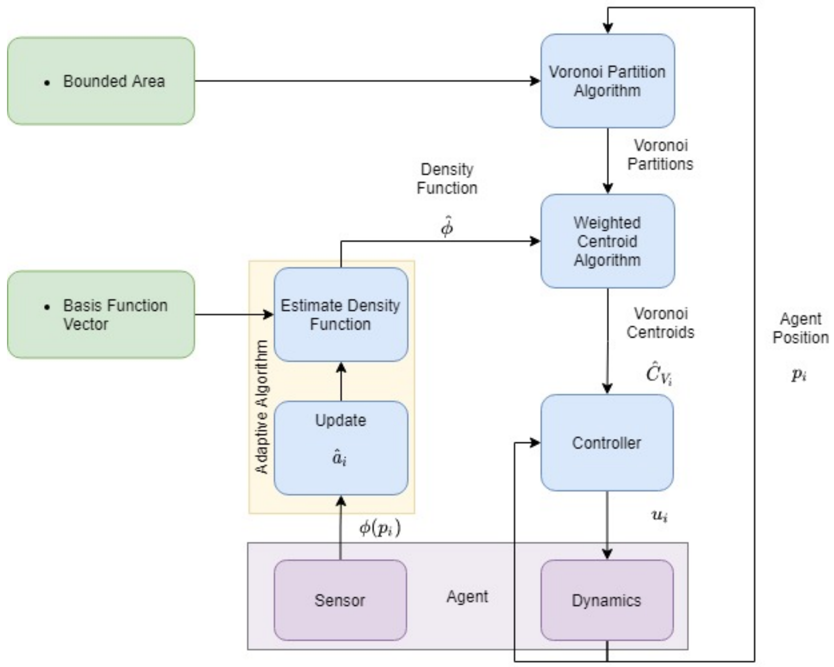

Figure 4.1: Controller topology for adaptive coverage control of homogeneous multi-agents

4.1 Problem Formulation

The problem of heterogeneous adaptive coverage control is formulated similar to [17] [19] [18]. Consider an environment $Q \in \mathbb{R}^2$. Define a distribution density function $\phi : Q \to \mathbb{R}_+$. ϕ represents the relative importance of coverage at a point in Q, this density function is unknown to the agents. Let $f : \mathbb{R}_+ \to \mathbb{R}_+$ represent the precision of the sensor reading as a function of distance from the agent. f will be a monotonically decreasing function of $|q - p_i|$, where $q \in Q$ and p_i is the position of i^{th} agent, in this study $f = \|q - p_i\|^2$. In addition to the setting it is assumed that the there are n agents that are introduced into the area, out of these n agents x are single integrators, y are double integrators and z are unicycles, such that $n = x + y + z$. Here it is assumed that the agents have the ability to measure the density function using its sensor. The agent in-turn have to *learn* or estimate the density function $\phi(q)$ at location q from their sensor measurements. For this setting the following assumptions are required.

Assumption 1 (Matching Conditions)[17]: The density function can be represented as

$$\phi(q) = \mathscr{K}(q)^T a, \tag{4.1}$$

where $\mathscr{K} : Q \rightarrow \mathbb{R}_+^m$ is a vector of basis function known to all agents and $a \in \mathbb{R}_+^m$ is constant parameter vector unknown to the agents. This assumption means that $\phi(q)$ can be represented as a weighted combination of set of basis functions $\mathscr{K}(q)^T = [\mathscr{K}_1(q), \mathscr{K}_2(q), ..., \mathscr{K}_m(q)]$.
Assumption 2 (Lower Bound)[17]:

$$a(j) \geq \beta \ \forall j = 1,..,m, \tag{4.2}$$

where $a(j)$ is the j^{th} element of the parameter vector a and $\beta > 0$ is a known real bound. This assumption implies a lower bound on $\phi(q)$ over Q. It ensures that $\phi(q)$ never becomes zero which may lead to C_{V_i} being undefined.

Let the i^{th} agent's estimate of the parameter vector and the density function be $\hat{a}_i(t)$ and $\hat{\phi}_i(q) = \mathscr{K}(q)^T \hat{a}_i$ respectively. Similar to (2.5)(2.6) the estimated mass, first moment and centroid of the voronoi cell V_i corresponding i^{th} agent are given by the following equations

$$\begin{aligned}
\hat{M}_{V_i} &= \int_{V_i} \hat{\phi}_i(q) dq, \\
\hat{L}_{V_i} &= \int_{V_i} q \hat{\phi}_i(q) dq, \\
\hat{C}_{V_i} &= \frac{1}{\hat{M}_{V_i}} \int_{V_i} q \hat{\phi}_i(q) dq
\end{aligned} \tag{4.3}$$

The parameter error is defined as

$$\tilde{a}_i = \hat{a}_i - a_i, \tag{4.4}$$

and the errors corresponding to the estimated mass, first moment and centroid of the voronoi cell V_i corresponding to the i^{th} agent are given by the following equations

$$\begin{aligned}
\tilde{M}_{V_i} &= \int_{V_i} \mathscr{K}(q)^T \tilde{a}_i dq = \hat{M}_{V_i} - M_{V_i}, \\
\tilde{L}_{V_i} &= \int_{V_i} q \mathscr{K}(q)^T \tilde{a}_i dq = \hat{L}_{V_i} - L_{V_i}, \\
\tilde{C}_{V_i} &= \frac{\tilde{L}_{V_i}}{\tilde{M}_{V_i}},
\end{aligned} \tag{4.5}$$

note that $\tilde{C}_{V_i} \neq \hat{C}_{V_i} - C_{V_i}$. The actual and the estimated error vectors are defined as $e_i = C_{V_i} - p_i$ and $\hat{e}_i = \hat{C}_{V_i} - p_i$ respectively. The adaptation law [17] for $\hat{a}_i$ is defined as

$$\dot{\hat{a}}_i = \Gamma \left(\dot{\hat{a}}_{pre_i} - I_{proj_i} \dot{\hat{a}}_{pre_i} \right), \tag{4.6}$$

where

$$\dot{\hat{a}}_{pre_i} = -F_i \hat{a}_i - (\Lambda_i \hat{a}_i - \lambda_i), \tag{4.7}$$

where $\Gamma \in \mathbb{R}^{m \times m}$ is a gain matrix with non-zero diagonal elements representing the gains for each component. The variables F_i, Λ_i and λ_i are defined as follows,

$$F_i = \left[\int_{V_1} K(q) \left(q - \hat{C}_{V_i} \right)^T dq \right] \dot{p}_i \tag{4.8}$$

$$\Lambda_i = \int_0^t w(\tau) \mathscr{K}_i(\tau) \mathscr{K}_i(\tau)^T d\tau \tag{4.9}$$

$$\lambda_i = \int_0^t w(\tau) \mathscr{K}_i(\tau) \phi_i(\tau) d\tau \tag{4.10}$$

and matrix I_{proj_i} is defined as

$$I_{proj_i}(j) = \begin{cases} 0 & \text{for } \hat{a}_i(j) > \beta \\ 0 & \text{for } \hat{a}_i(j) = \beta \text{ and } \dot{\hat{a}}_{pre_i} \geq 0 \\ 1 & \text{otherwise} \end{cases} \tag{4.11}$$

The function $w(t) \in \mathscr{L}^1$ is called a *weighing function* which simulates the parameter convergence of the adaptation law. The optimal coverage problem can be formulated as a locational optimization problem which minimizes the cost function given by

$$\mathscr{H}(P, V) = \frac{1}{2} \sum_{i=1}^{n} \int_{V_i} \| q - p_i \|^2 \hat{\phi}(q) dq \tag{4.12}$$

The optimal coverage problem is now defined as a minimization problem to minimize equation (4.12) with respect to the agent positions p_i. The partial derivative of the cost function with respect to the positions is given by [5]

$$\frac{\partial \mathscr{H}(P)}{\partial p_i} = \hat{M}_{V_i}\left(p_i - \hat{C}_{V_i}\right), \tag{4.13}$$

where $\hat{M}_{V_i}$ and $\hat{C}_{V_i}$ are the mass and the centroid given by (4.3).

It can be observed that the $p_i = \hat{C}_{V_i}$ is a local optimum for (4.12). It corresponds to positioning the agents at their respective voronoi centroids (centroidal voronoi configuration).

4.2 Methodology

In the previous section the problem of heterogeneous adaptive coverage control was formulated and the condition for optimal placement of agents have been stated. In this section the control laws required to drive the agents to their desired location are explored. The controllers are similar to the controllers used in section 3.2

A two level nested control is proposed:

- **Outer Controller:** Computes the voronoi partition of the environment based on the position of the agents (generator points). It will also compute the centroid of each voronoi cell and give it as the set point for the inner controller.
- **Inner Controller:** Takes the set point given by the outer controller and generates the required control input to be given to the agents so as to minimize the error between the setpoint and the current position of the agent. This controller will be different for the different agent kinematics.

The outer controller is similar to the controller that is used in [5] and will estimate the voronoi cell of each agent in a decentralized manner. It does this by identifying the relative location (both distance and bearing) of each voronoi neighbour of an agent. Once the neighbours' locations are identified the agent will be able to estimate its own voronoi cell. The voronoi boundary between to agents is given by the perpendicular bisector of the line joining the position of the two agents.

The inner controller will differ based on the agent kinematics. The proposed controllers

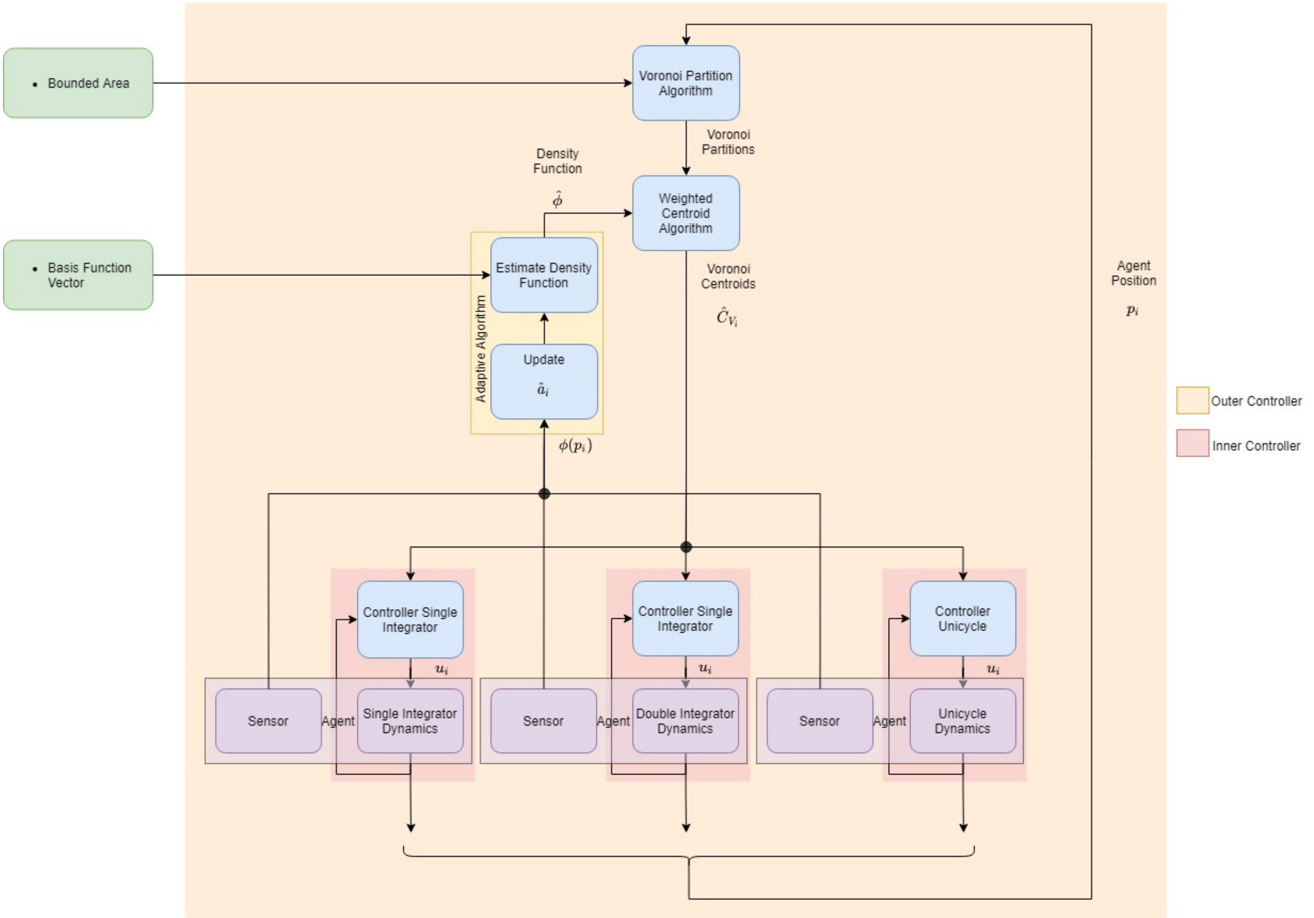

Figure 4.2: Controller topology for adaptive coverage control of heterogeneous multi-agents

for the three types of kinematics considered in the study are

- **Single integrator:** For a single integrator agent given by the dynamics (2.10) the controller can be a simple proportional controller. It will be similar to the gradient descent in the direction of maximum gradient. The control law for the i^{th} agent is given by

$$u_i = -k(p_i - C_{V_i}), \tag{4.14}$$

  where k is the a positive proportional gain.

- **Double integrator:** For a double integrator agent with kinematics given by (2.12) the controller has to have a differential term to counter act the second order dynamics. The proportional and derivative control gains have to be tuned in order to reduce oscillations.

The control law for the i^{th} agent is given by

$$\begin{aligned} u_i &= -k_1 M_{V_i}(p_{i1} - C_{V_i}) - k_2 p_{i_2} \\ &= -k_1 M_{V_i}(p_i - C_{V_i}) - k_2 \dot{p}_i, \end{aligned} \tag{4.15}$$

where k_1 is the positive proportional gain and k_2 is the positive derivative gain.

- **Unicycle:** Unicycle represented by the polar form (2.17) can be controlled using polar controller given by

$$\begin{aligned} u_i &= (\gamma \cos \alpha_i)\, \rho_i \\ \omega_i &= k_3 \alpha_i + \gamma \frac{\cos \alpha_i \sin \alpha_i}{\alpha_i} (\alpha_i + h\theta_i) \end{aligned} \tag{4.16}$$

4.3 Stability Analysis

In this section the stability analysis of the heterogeneous multi-agent system will be done using a Lyapunov like method. Considering x single integrators, y double integrators and z unicycles following dynamics in equations (2.10), (2.12) and (2.15) respectively and control laws according to equations (4.14), (4.15) and (4.16) respectively. Consider the Lyapunov function [5][19] candidate

$$\begin{aligned} \mathscr{V} =& \left(\frac{1}{2} \sum_{i=1}^{x} \int_{V_i} \|q - p_i\|^2 \phi(q) dq + \frac{1}{2} \sum_{i=1}^{x} \tilde{a}_i^T \Gamma^{-1} \tilde{a}_i \right) \\ & \left(\frac{k_1}{2} \sum_{i=x+1}^{x+y} \int_{V_i} \|q - p_i\|^2 \phi(q) dq + \frac{1}{2} \sum_{i=x+1}^{x+y} \dot{p}_i^2 + \frac{1}{2} \sum_{i=x+1}^{x+y} \tilde{a}_i^T \Gamma^{-1} \tilde{a}_i \right) + \\ & \left(\frac{1}{2} \sum_{i=x+y+1}^{x+y+z} \int_{V_i} \|q - p_i\|^2 \phi(q) dq + \frac{1}{2} \sum_{i=x+y+1}^{x+y+z} (\alpha_i^2 + h\theta_i^2) + \frac{1}{2} \sum_{i=x+y+1}^{x+y+z} \tilde{a}_i^T \Gamma^{-1} \tilde{a}_i \right) \\ =& (\mathscr{V}_1) + (\mathscr{V}_2) + (\mathscr{V}_3) \end{aligned} \tag{4.17}$$

In the above equation $\mathscr{V}$ is positive definite so are $\mathscr{V}_1$, $\mathscr{V}_2$, $\mathscr{V}_3$ separately, each of them correspond to an agent of particular kinematics. Since there are no coupling between the agents they can be separated.

$$\dot{\mathscr{V}} = \dot{\mathscr{V}}_1 + \dot{\mathscr{V}}_2 + \dot{\mathscr{V}}_3 \tag{4.18}$$

Finding $\dot{\mathscr{V}}_1$ and substituting $\dot{p}_i$ with equation (3.3)

$$\begin{aligned}
\dot{\mathscr{V}}_1 &= \sum_{i=1}^{x}\left[\frac{1}{2}\frac{\partial}{\partial p_i}\left(\int_{V_i}\|q-p_i\|^2\phi(q)dq\right)\dot{p}_i + \tilde{a}_i^T\Gamma^{-1}\dot{\hat{a}}_i\right] \\
&= \sum_{i=1}^{x}\left[\left(-\int_{V_i}(q-p_i)^T\phi(q)dq\right)\dot{p}_i + \tilde{a}_i^T\Gamma^{-1}\dot{\hat{a}}_i\right] \\
&= \sum_{i=1}^{x}\left[\left(-\int_{V_i}(q-p_i)^T\left(\hat{\phi}_i(q)+\mathscr{K}(q)^T\tilde{a}_i(q)\right)dq\right)\dot{p}_i + \tilde{a}_i^T\Gamma^{-1}\dot{\hat{a}}_i\right] \\
&= \sum_{i=1}^{x}\left[-\int_{V_i}(q-p_i)^T\hat{\phi}_i dq\dot{p}_i + \int_{V_i}\tilde{a}_i^T\mathscr{K}(q)(q-p_i)^T dq\dot{p}_i + \tilde{a}_i^T\Gamma^{-1}\dot{a}_i\right] \\
&= \sum_{i=1}^{x}\left[-\hat{M}_{V_i}\left(\hat{C}_{V_i}-p_i\right)^T K\left(\hat{C}_{V_i}-p_i\right) + \tilde{a}_i^T\int_{V_i}\mathscr{K}(q)(q-p_i)^T dq\left(\hat{C}_{V_i}-p_i\right) + \tilde{a}_i^T\Gamma^{-1}\dot{a}_i\right] \\
&= \sum_{i=1}^{x}\left[-\hat{M}_{V_i}\left(\hat{C}_{V_i}-p_i\right)^{\mathrm{T}} K\left(\hat{C}_{V_i}-p_i\right) + \tilde{a}_i^{\mathrm{T}}F_i\hat{a}_i - \tilde{a}_i^{\mathrm{T}}F_i\hat{a}_i - \tilde{a}_i^{\mathrm{T}}\gamma(\Lambda_i\hat{a}_i-\lambda_i) - \tilde{a}_i^{\mathrm{T}}I\mathrm{proj}_i\dot{\hat{a}}\mathrm{pre}_i\right] \\
&= -\sum_{i=1}^{x}\left[\hat{M}_{V_i}\left(\hat{C}_{V_i}-p_i\right)^{\mathrm{T}} K\left(\hat{C}_{V_i}-p_i\right) + \tilde{a}_i^{\mathrm{T}}\gamma\int_0^t w(\tau)\mathscr{K}_i\mathscr{K}_i^{\mathrm{T}}\tilde{a}_i(t)d\tau + \tilde{a}_i^{\mathrm{T}}I_{\mathrm{proj}_i}\dot{\hat{a}}\mathrm{pre}_i\right] \\
&= -\sum_{i=1}^{x}\left[\hat{M}_{V_i}\left(\hat{C}_{V_i}-p_i\right)^{\mathrm{T}} K\left(\hat{C}_{V_i}-p_i\right) + \gamma\int_0^t w(\tau)\left(\mathscr{K}_i(\tau)^{\mathrm{T}}\tilde{a}_i(t)\right)^2 d\tau + \tilde{a}_i^{\mathrm{T}}I_{\mathrm{proj}_i}\dot{\hat{a}}_i\mathrm{pre}_i\right],
\end{aligned} \tag{4.19}$$

here the first two terms are non-negative. Focusing on the third term

$$\tilde{a}_i(j)I_{\mathrm{proj}_i}(j)\dot{\hat{a}}_{\mathrm{pre}_i}(j) \tag{4.20}$$

from equation (4.11) if

- $\hat{a}_i(j) > \beta$, or $\hat{a}_i(j) = \beta$ and $\dot{\hat{a}}_{\mathrm{pre}_i}(j) \geq 0$, then $I_{\mathrm{proj}_i}(j) = 0$ and the term vanishes.

- $\hat{a}_i(j) = \beta$ and $\dot{\hat{a}}_{\mathrm{pre}_i}(j) < 0$, then $\tilde{a}_i(j) = \hat{a}_i(j) - a_i(j) \leq 0$ (from Assumption 1). Furthermore, $I_{\mathrm{proj}_i}(j) = 1$ and $\hat{a}_{\mathrm{pre}_i}(j) < 0$ implies that the term is non-negative.

In all the cases each term of the equation (4.19) is non-negative, so $\dot{\mathscr{V}}_1 \leq 0$

Similarly finding $\dot{\mathscr{V}}_2$ and substituting $\ddot{p}_i$ with equation (3.4)

$$\begin{aligned}
\dot{\mathscr{V}}_2 &= \sum_{i=x+1}^{x+y} \left[\frac{\partial}{\partial p_i} \left(\frac{k_1}{2} \int_{V_i} \|q-p_i\|^2 \phi(q)dq + \ddot{p}_i \right) \dot{p}_i + \tilde{a}_i^T \Gamma^{-1} \dot{\tilde{a}}_i \right] \\
&= \sum_{i=x+1}^{x+y} \left[k_1 M_{V_i} \left(p_i - C_{V_i}\right) \dot{p}_i + \ddot{p}_i \dot{p}_i + \tilde{a}_i^T \Gamma^{-1} \dot{\tilde{a}}_i \right] \\
&= \sum_{i=x+1}^{x+y} \left[k_1 M_{V_i} \left(p_i - C_{V_i}\right) \dot{p}_i + \left(-k_1 M_{V_i}(p_i - C_{V_i}) - k_2\right) \dot{p}_i + \tilde{a}_i^T \Gamma^{-1} \dot{\tilde{a}}_i \right] \\
&= -\sum_{i=x+1}^{x+y} \left[k_2 \dot{p}_i^2 + \tilde{a}_i^T \Gamma^{-1} \dot{\tilde{a}}_i \right],
\end{aligned} \tag{4.21}$$

here the first term is non-negative and the second term is shown to be non-negative from the discussion following the equation (4.21). So $\dot{\mathscr{V}}_2 \leq 0$

For unicycle agents from Figure 2.2

$$\begin{aligned}
C_{V_i} - p_i = \begin{pmatrix} x_c - x_i \\ y_c - y_i \end{pmatrix} &= \begin{pmatrix} \rho_i cos\theta_i \\ \rho_i sin\theta_i \end{pmatrix} \\
&= \begin{pmatrix} \rho_i cos(\phi_i + \alpha_i) \\ \rho_i sin(\phi_i + \alpha_i) \end{pmatrix}
\end{aligned} \tag{4.22}$$

Finding $\dot{\mathscr{V}}_3$ and substituting $\dot{p}_i = u_i$, $\dot{\alpha}_i$ and $\dot{\theta}_i$ with equation (3.7) and $(C_{V_i} - p_i)$ with equation (3.12)

$$\begin{aligned}
\dot{\mathscr{V}}_3 &= \frac{1}{2} \sum_{i=x+y+1}^{x+y+z} \left[\frac{\partial}{\partial p_i} \left(\int_{V_i} \|q-p_i\|^2 \phi(q)dq \right) \dot{p}_i + \left(\alpha_i \dot{\alpha}_i + h\theta_i \dot{\theta}_i\right) + \tilde{a}_i^T \Gamma^{-1} \dot{\tilde{a}}_i \right] \\
&= \sum_{i=x+y+1}^{x+y+z} \left[M_{V_i} \left(p_i - C_{V_i}\right)^T \dot{p}_i + \left(\alpha_i \dot{\alpha}_i + h\theta_i \dot{\theta}_i\right) + \tilde{a}_i^T \Gamma^{-1} \dot{\tilde{a}}_i \right] \\
&= \sum_{i=x+y+1}^{x+y+z} \Bigg[-M_{V_i} \begin{pmatrix} \rho_i cos(\phi_i + \alpha_i) \\ \rho_i sin(\phi_i + \alpha_i) \end{pmatrix} \cdot \begin{pmatrix} (\rho_i cos\alpha_i) cos(\phi_i) \\ (\rho_i cos\alpha_i) sin(\phi_i) \end{pmatrix} \\
&\qquad + \left(\alpha_i \left(-\omega_i + u_i \frac{sin\alpha_i}{\rho_i} \right) + h\theta_i u_i \frac{sin\alpha_i}{\rho_i} \right) + \tilde{a}_i^T \Gamma^{-1} \dot{\tilde{a}}_i \Bigg] \\
&= -\sum_{i=x+y+1}^{x+y+z} \left[\left(M_{V_i} \rho_i^2 \gamma cos^2 \alpha_i \right) + k_3 \alpha_i^2 + \tilde{a}_i^T \Gamma^{-1} \dot{\tilde{a}}_i \right],
\end{aligned} \tag{4.23}$$

here also all terms are non-negative, so $\dot{\mathscr{V}}_3 \leq 0$ And so from equations (3.9), (3.10), (3.11), (3.13)

$$\dot{\mathscr{V}} = \dot{\mathscr{V}}_1 + \dot{\mathscr{V}}_2 + \dot{\mathscr{V}}_3 \leq 0 \tag{4.24}$$

Since $\mathscr{V}$ is lower bounded and $\dot{\mathscr{V}}$ is negative semidefinite it can be concluded that $\dot{\mathscr{V}} \to 0$ as $t \to \infty$ by Lyapunov-like lemma.

4.4 Simulations

Simulations were done on Matlab and Simulink, the agent dynamics and the inner controllers were modelled in Simulink and they were invoked by a Matlab script which produced the control signal from the outer controller. More details regarding the implementation of simulations are given in the Appendix B. Here the simulation results will be given.

Table 4.1: Simulation parameters used for heterogeneous adaptive coverage control

Parameter	Value	Parameter	Value
k	6	k_1	3
k_2	2	k_3	1
γ	3	h	1
ω_{max}	$\pm\pi$	n	15
α	1	a	$[1000, 0.1, ..., 1000, 0.1]^T$
μ_1	$[1.67, 1.67]$	μ_2	$[8.33, 8.33]$
m	9	a_{min}	0.1
$\gamma_{adaptive}$	20	Γ	$1 \times 10^{-2} I$

A 10 unit square area was considered as the region to be covered, $n = 15$ agents were introduced at random positions in the area. A distribution density function was defined over the area and each agent has the knowledge of the bounds of the region and the distribution density function. It is assumed that the agents know their own positions in the area (perfectly localized) and have the capability to find the location of their neighbours, using which each agent has the capability to compute its voronoi partition. Additionally, $n = 15$ agents include: $x = 5$ single integrator agents, $y = 5$ double integrator agents and $z = 5$ unicycle agents. Specifications regarding the initial position and the distribution density function used vary with experiment and they are mentioned under each subheading.

4.4.1 Experiment 1: Bimodal gaussian distribution with agents randomly placed

In this experiment the distribution density function is assumed to be bimodal gaussian distribution, i.e. all points inside the are are not equally important. The initial location of the agents can be anywhere in the region considered.

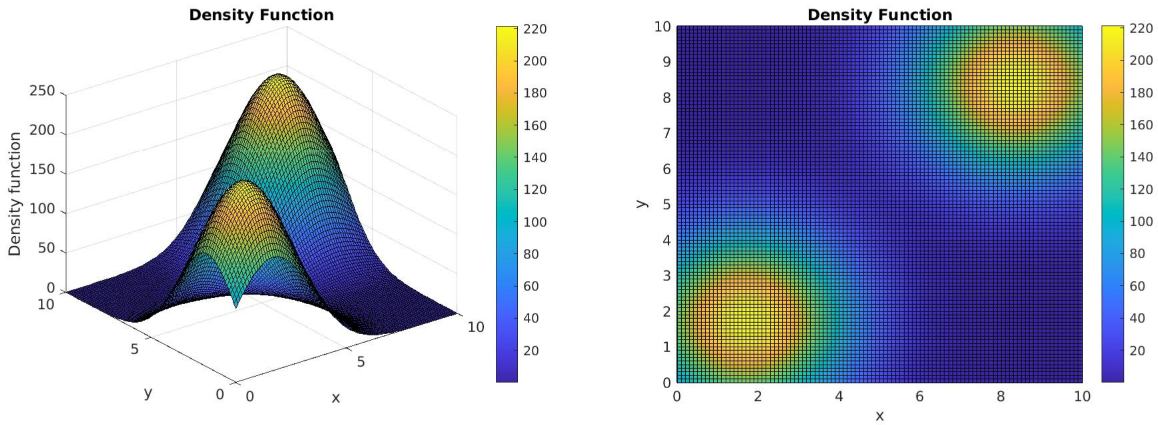

Figure 4.3: Bimodal density distribution function with $\sigma = 1.8$ and means $\mu_1 = (1.67, 1.67)$ and $\mu_2 = (8.33, 8.33)$ 3d and 2d views

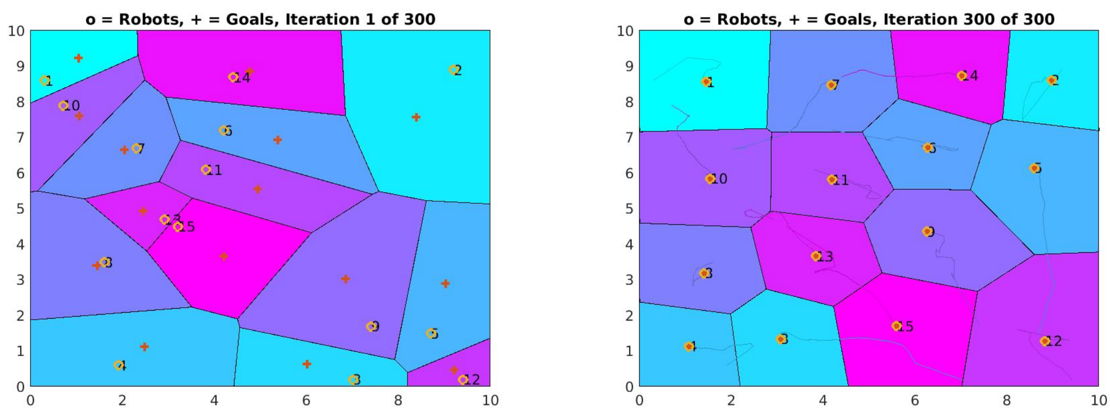

Figure 4.4: $n = 15$ agents initial position randomly (left) and final position with path (right)

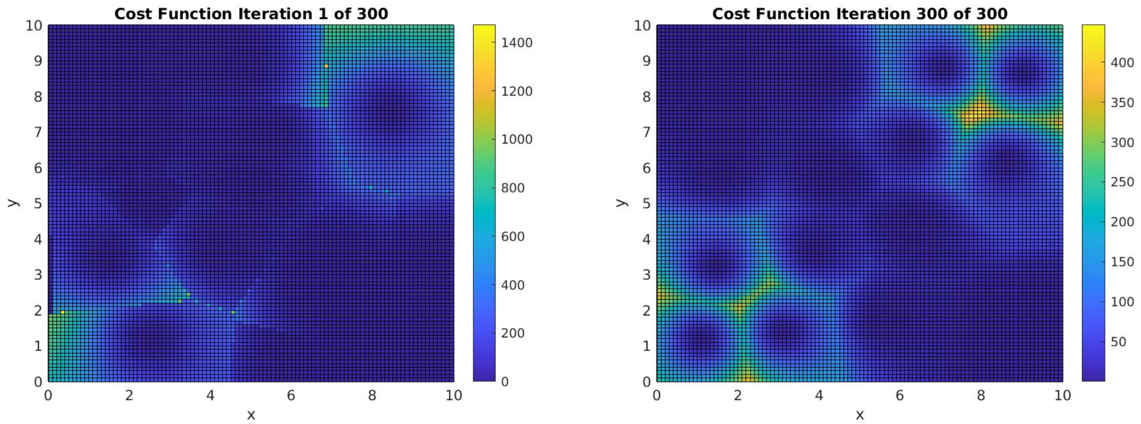

Figure 4.5: Distribution of cost function in the region initial (left) and final (right)

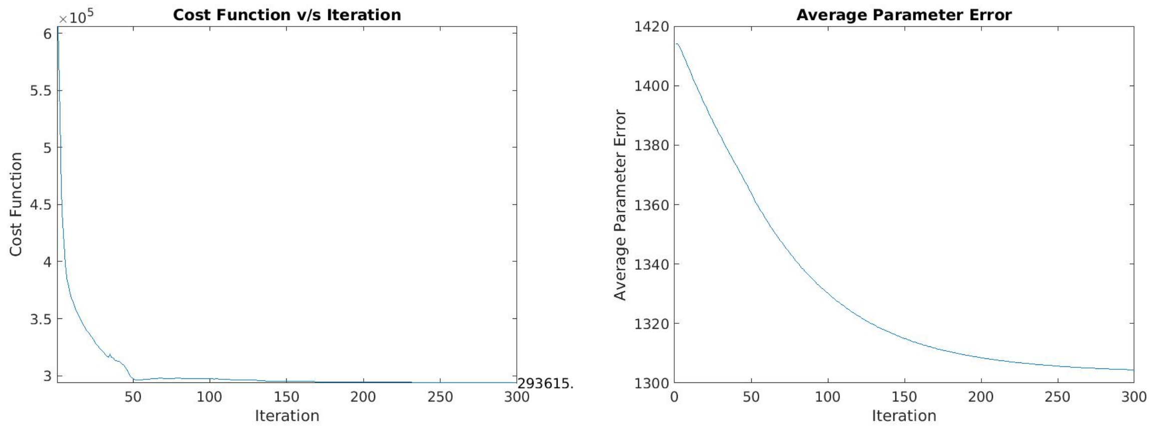

Figure 4.6: Cost function variation with iteration (left) and average parameter error with iteration (right)

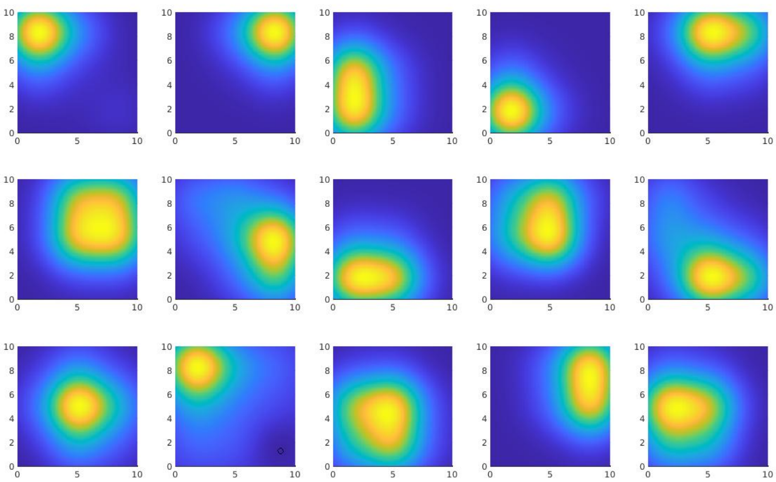

Figure 4.7: Estimated density function by the 15 agents

Observations: The average parameter error is decreasing with iteration, which suggests that the agents are getting better at estimating the density function. The reduction in cost function is comparable to the non-adaptive case (51.6% reduction). Fig 4.7 shows the density function estimated by the different agents. It can be observed that the estimated density function is not the same as the original density function, but it will depend on the path that the agent has taken to reach its final position.

4.4.2 Experiment 2: Bimodal gaussian distribution with agents starting from unit square

In this experiment the distribution density function is assumed to be bimodal gaussian distribution, i.e. all points inside the are are not equally important. The initial location of the agents is inside a unit square in the region considered.

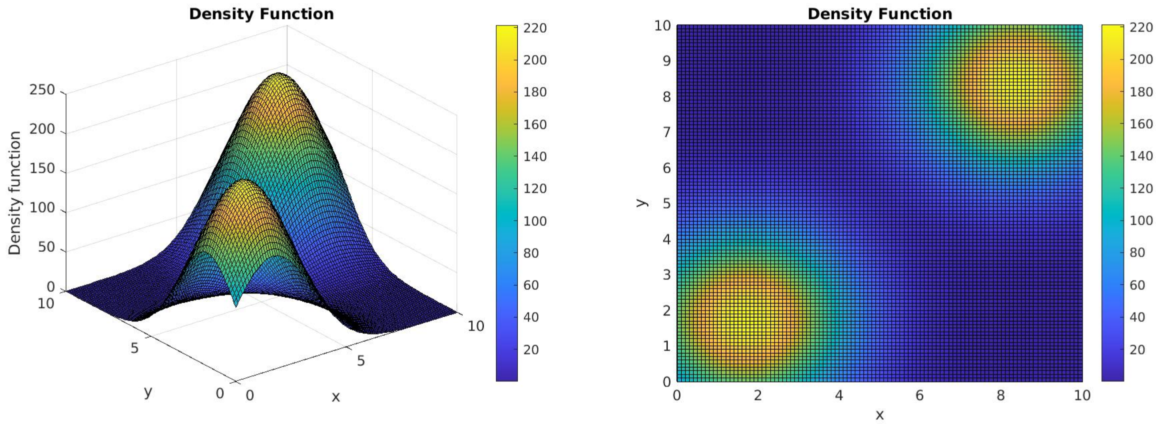

Figure 4.8: Bimodal density distribution function with $\sigma = 1.8$ and means $\mu_1 = (1.67, 1.67)$ and $\mu_2 = (8.33, 8.33)$ 3d and 2d views

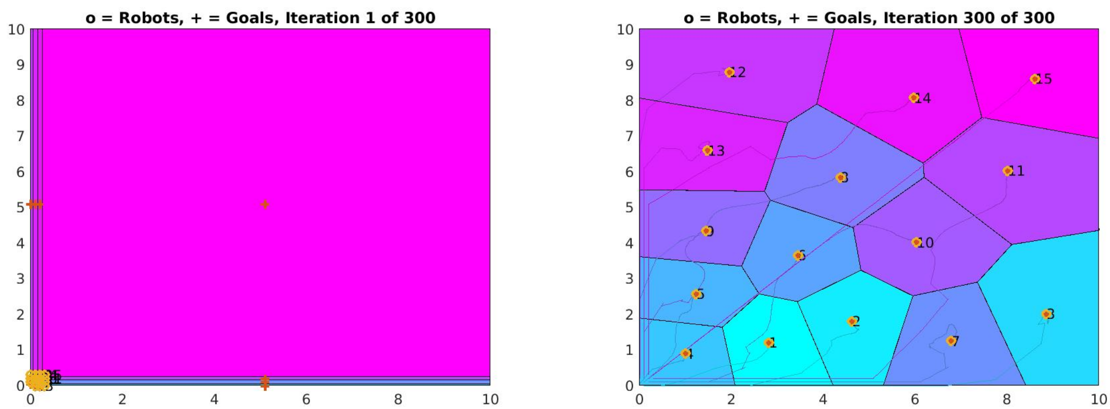

Figure 4.9: $n = 15$ agents initial position inside unit square (left) and final position with path (right)

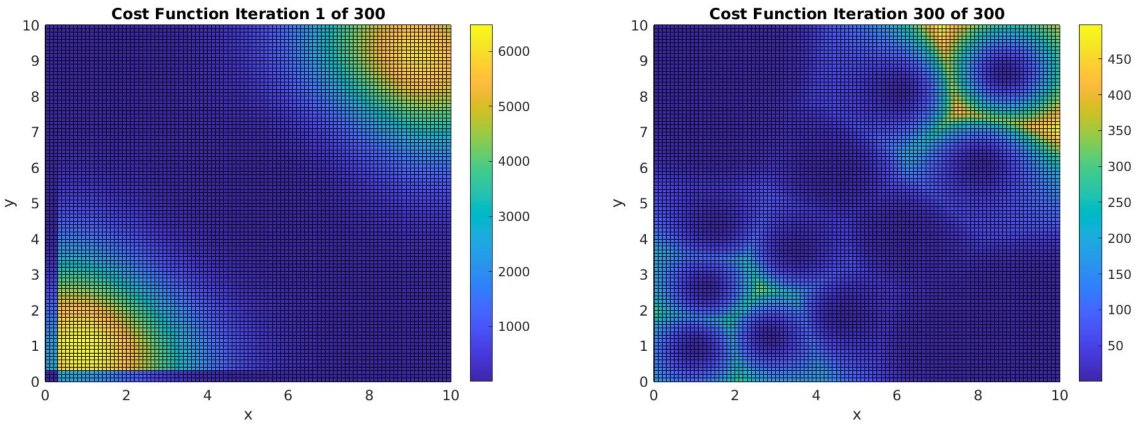

Figure 4.10: Distribution of cost function in the region initial (left) and final (right)

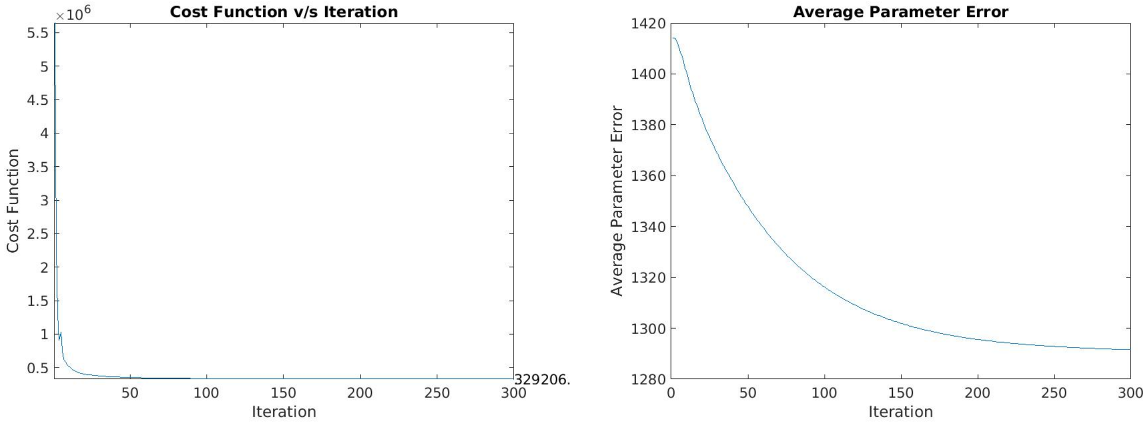

Figure 4.11: Cost function variation with iteration (left) and average parameter error with iteration (right)

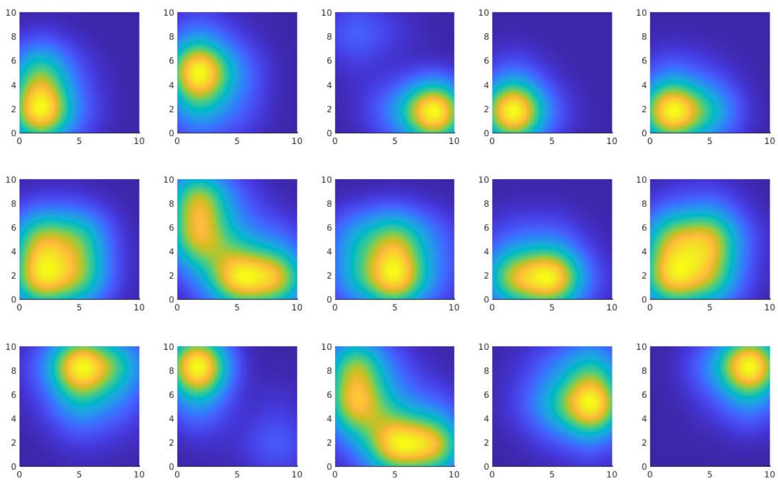

Figure 4.12: Estimated density function by the 15 agents

Observations: The average parameter error is decreasing with iteration, which suggests that the agents are getting better at estimating the density function. The reduction in cost function is comparable to the non-adaptive case (94.2% reduction). Fig 4.12 shows the density function estimated by the different agents. It can be observed that the estimated density function is not the same as the original density function, but it will depend on the path that the agent has taken to reach its final position.

4.5 Results

- The adaptive coverage control law for heterogeneous multi-agents that was proposed is giving results that are comparable to the non-adaptive case that was simulated in the previous chapter.
- As expected the estimated density function learned by the agents are not exactly matching with the original density function and also they do not converge to the same for all agents. A consensus law can be applied to the parameter estimation part of the proposed algorithm to make the estimated parameter same among the agents.

This page was intentionally left blank.

Chapter 5

Coverage Control of Heterogeneous Multi-Agents in Non-Convex Environment

In the chapters till now the approaches that were taken for solving the coverage control problem all had an assumption that the region under consideration is convex. Convexity of the region ensures that the voronoi partitions of the region are also convex [26]. This in-turn ensures that the centroids of the voronoi partitions remain inside the respective partitions itself, allowing for a straight line path between the corresponding agent's current position and the centroid. But once the non-convexity assumption is removed or relaxed these beneficial properties may not be available. The various approaches to solving non-convexity have been discussed in section 2.4.

This section proposes a heuristic solution to a form of non-convexity which is similar to the approach taken section 2.4.4, i.e. manipulation of the distribution density function. The effectiveness of the proposed method is then verified using simulations.

5.1 Problem Formulation

The problem is formulated similar to the problem formulation in [5] with a relaxed assumption on convexity of the region. Consider a bounded environment $Q \in \mathbb{R}^2$. Define a distribution density function $\phi : Q \to \mathbb{R}_+$. ϕ represents the relative importance of coverage at a point in Q. Let $f : \mathbb{R}_+ \to \mathbb{R}_+$ represent the precision of the sensor reading as a function of distance from the agent. f will be a monotonically decreasing function of $|q - p_i|$, where $q \in Q$ and p_i is the position of i^{th} agent. In this study $f = \|q - p_i\|^2$. In addition to the setting it is assumed that the there are n agents that are introduced into the area, out of these n agents x are single integrators, y are double integrators and z are unicycles, such that $n = x + y + z$.
The optimal coverage problem can be formulated as a locational optimization problem which minimizes the cost function given by

$$\mathscr{H}(P,V) = \frac{1}{2}\sum_{i=1}^{n}\int_{V_i}\|q - p_i\|^2\phi(q)dq \tag{5.1}$$

The optimal coverage problem is now defined as a minimization problem to minimize equation (3.1) with respect to the agent positions p_i. The partial derivative of the cost function with respect to the positions is given by [5]

$$\frac{\partial\mathscr{H}(P)}{\partial p_i} = M_{V_i}\left(p_i - C_{V_i}\right), \tag{5.2}$$

where M_{V_i} and C_{V_i} are the mass and the centroid given by (2.5) and (2.6) respectively.

It can be observed that the $p_i = C_{V_i}$ is a local optimum for (5.1). It corresponds to positioning the agents at their respective voronoi centroids (centroidal voronoi configuration).

5.2 Methodology

In the previous section the problem of heterogeneous coverage control was formulated and the condition for optimal placement of agents have been stated. In this section the control laws required to drive the agents to their desired location are explored.

A two level nested control is proposed:

- **Outer Controller:** Computes the voronoi partition of the environment based on the position of the agents (generator points). It will also compute the centroid of each voronoi cell and give it as the set point for the inner controller.

- **Inner Controller:** Takes the set point given by the outer controller and generates the required control input to be given to the agents so as to minimize the error between the setpoint and the current position of the agent. This controller will be different for the different agent kinematics.

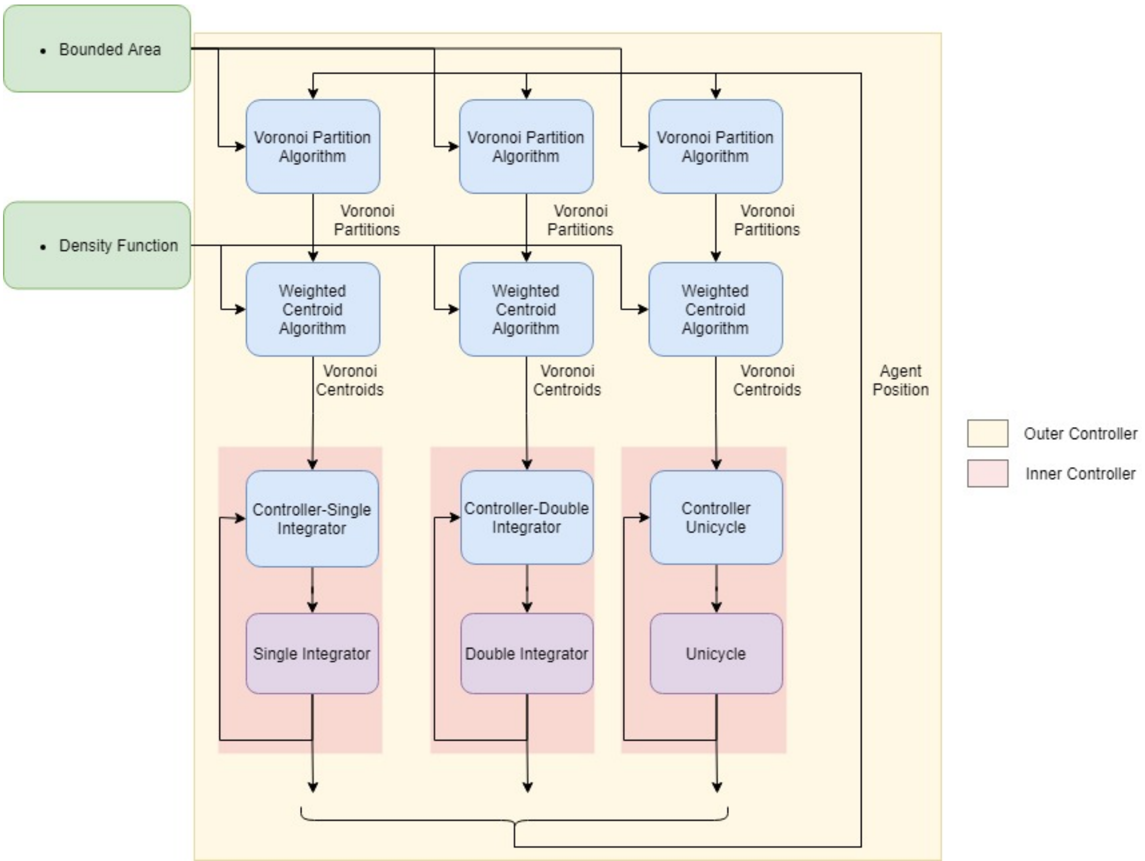

Figure 5.1: Controller topology for coverage control of homogeneous multi-agents

The outer controller is similar to the controller that is used in [5] and will estimate the voronoi cell of each agent in a decentralized manner. It does this by identifying the relative location (both distance and bearing) of each voronoi neighbour of an agent. Once the neighbours' locations are identified the agent will be able to estimate its own voronoi cell. The voronoi boundary between to agents is given by the perpendicular bisector of the line joining the position of the two agents.

The inner controller will differ based on the agent kinematics. The proposed controllers for the three types of kinematics considered in the study are

- **Single integrator:** For a single integrator agent given by the dynamics (2.10) the controller can be a simple proportional controller. It will be similar to the gradient descent in the direction of maximum gradient. The control law for the i^{th} agent is given by

$$u_i = -k(p_i - C_{V_i}), \tag{5.3}$$

  where k is the a positive proportional gain.

- **Double integrator:** For a double integrator agent with kinematics given by (2.12) the controller has to have a differential term to counter act the second order dynamics. The proportional and derivative control gains have to be tuned in order to reduce oscillations. The control law for the i^{th} agent is given by

$$\begin{aligned} u_i &= -k_1 M_{V_i}(p_{i_1} - C_{V_i}) - k_2 p_{i_2} \\ &= -k_1 M_{V_i}(p_i - C_{V_i}) - k_2 \dot{p}_i, \end{aligned} \tag{5.4}$$

  where k_1 is the positive proportional gain and k_2 is the positive derivative gain.

- **Unicycle:** For a unicycle agent given by the dynamics (2.15) the controller can be based on geometric projection of the heading vector onto the error vector. For the i^{th} agent, let centroid be $C_{V_i} = (x_c, y_c)$, current position $p_i = (x_i, y_i)$ and current heading ϕ_i. Unicycle represented by the polar form (2.17) can be controlled using polar controller given by

$$\begin{aligned} u_i &= (\gamma \cos \alpha_i)\, \rho_i \\ \omega_i &= k_3 \alpha_i + \gamma \frac{\cos \alpha_i \sin \alpha_i}{\alpha_i} (\alpha_i + h\theta_i) \end{aligned} \tag{5.5}$$

Extension to non-convex regions

For extending the above framework to non-convex region a modified density function is used for calculating the centroid of the voronoi partitions. The proposed modified density function is

$$\phi' = \phi + \mathrm{d} \times (e^{-\mathrm{ct}} D_{dist}), \tag{5.6}$$

where d and c are positive constants, D_{dist} is the distance transform of the region.

$$D_{dist}(q) = min \, \|q - \delta Q\| \ \ \forall q \in Q, \tag{5.7}$$

where δQ is the set of boundary points of Q

Intuition behind the heuristic

During simulation study it was seen that for coverage control of non-convex environment, agents tend to move outside the valid region during the initial few iteration after commencement, this is due to a large pull towards unexplored extremities of the region. By modifying the density function as given by equation (5.6) this initial pull towards extremities is reduced initially. There will be a larger (depends on d) pull towards the interior of the region initially making all the agents to move towards the interior and then depending on c the modification will gradually reduce to converge to the original density function.

5.3 Simulations

Simulations were done on Matlab and Simulink, the agent dynamics and the inner controllers were modelled in Simulink and they were invoked by a Matlab script which produced the control signal from the outer controller. More details regarding the implementation of simulations are given in the Appendix B. Here the simulation results will be given.

Table 5.1: Simulation parameters used for heterogeneous coverage control in non-convex environment

Parameter	Value	Parameter	Value
k	6	k_1	3
k_2	2	k_3	1
γ	3	h	1
d	2	c	0.5
ω_{max}	$\pm\pi$	n	15

A 10 unit square area was considered as the region to be covered, $n = 15$ agents were introduced at random positions in the area. A distribution density function was defined over the area and each agent has the knowledge of the bounds of the region and the distribution density function. It is assumed that the agents know their own positions in the area (perfectly localized) and have the capability to find the location of their neighbours, using which each agent is able to compute its voronoi partition. Additionally, $n = 15$ agents include: $x = 5$ single integrator agents, $y = 5$ double integrator agents and $z = 5$ unicycle agents. Specifications regarding the initial position and the distribution density function used vary with experiment and they are mentioned under each subheading.

5.3.1 Experiment 1: Uniformly distributed non-convex region

In this experiment the distribution density function is assumed to be uniformly distributed, i.e. all points inside the are equally important. The initial location of the agents are in the unit square.

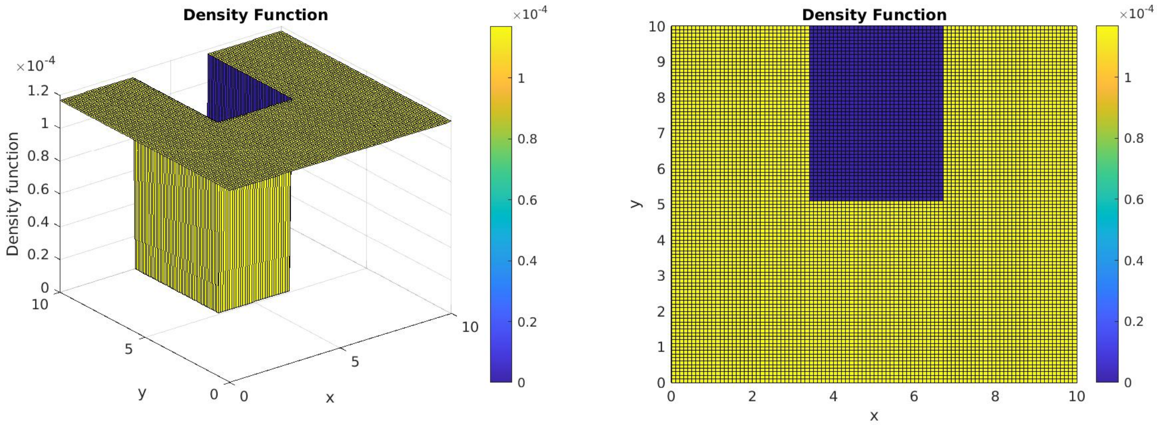

Figure 5.2: Uniform density distribution function 3d and 2d views

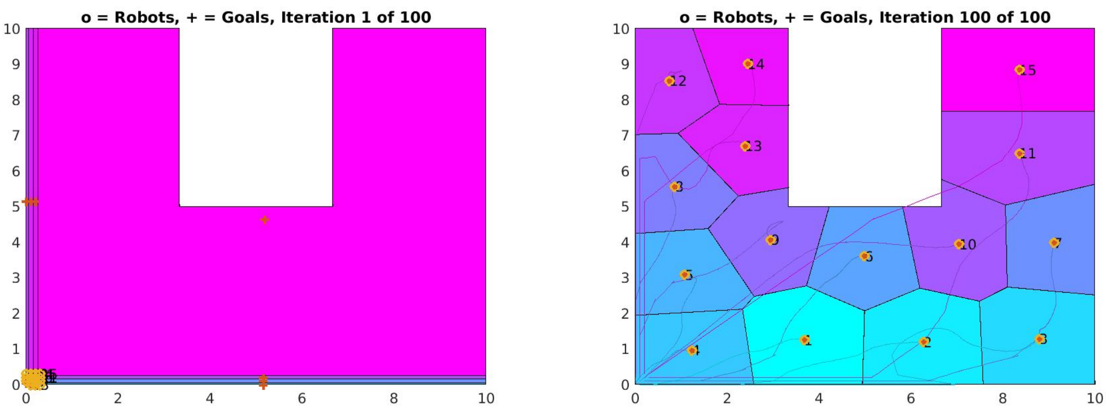

Figure 5.3: $n = 15$ agents initial position inside a unit square(left) and final position with path (right)

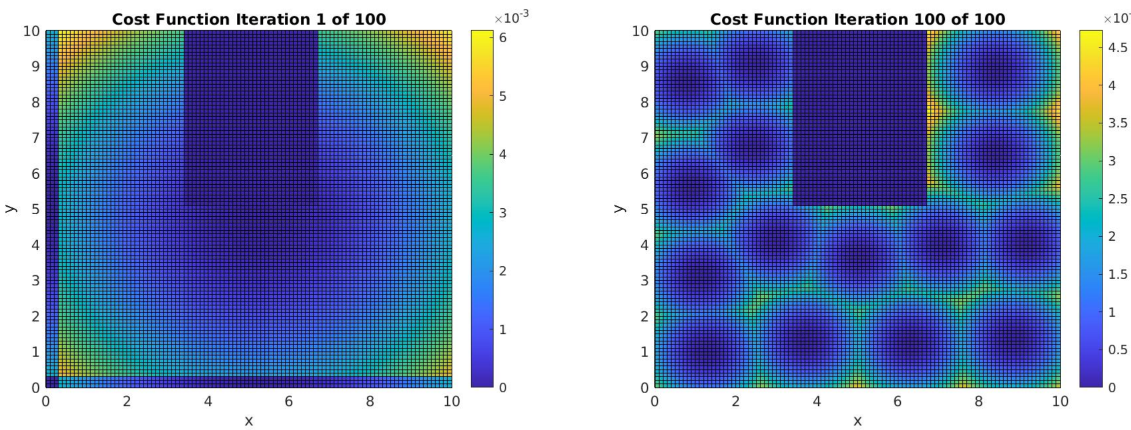

Figure 5.4: Distribution of cost function in the region initial (left) and final (right)

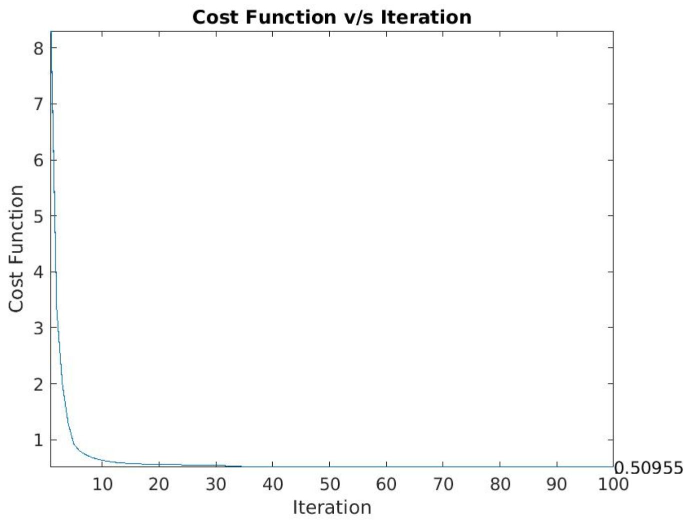

Figure 5.5: Cost function variation with iteration

Observations: It is seen that for a non-convex region some of the agents are moving outside the valid region in Fig.5.3. In a real world situation, outside of the boundaries of the region may not be accessible to the agents or the boundaries themselves may be walls which will result in collision of the agents with the walls.

5.3.2 Experiment 2: Uniformly distributed non-convex region with modified density function

In this experiment the distribution density function is assumed to be uniformly distributed, i.e. all points inside the are equally important. The initial location of the agents are in the unit square. In addition the centroids are calculated based on the modified density function proposed in the previous section.

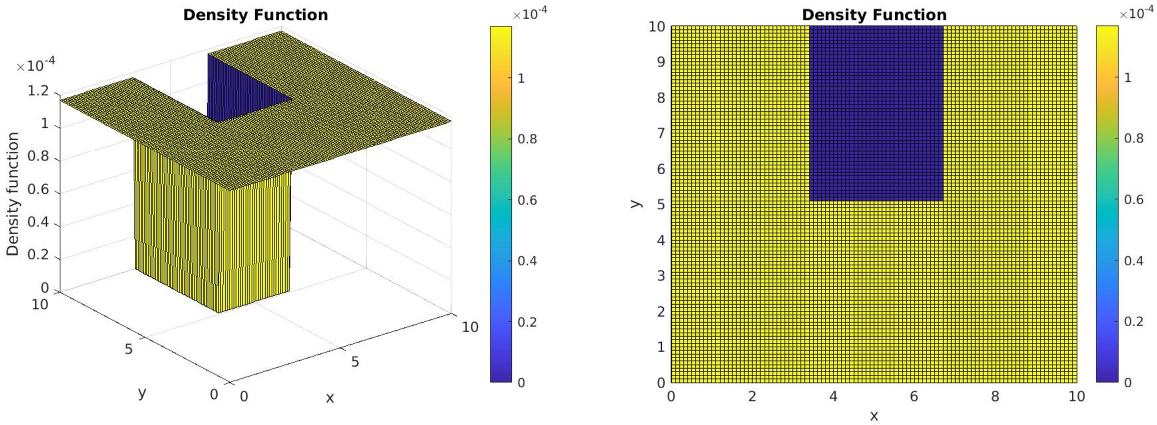

Figure 5.6: Uniform density distribution function 3d and 2d views

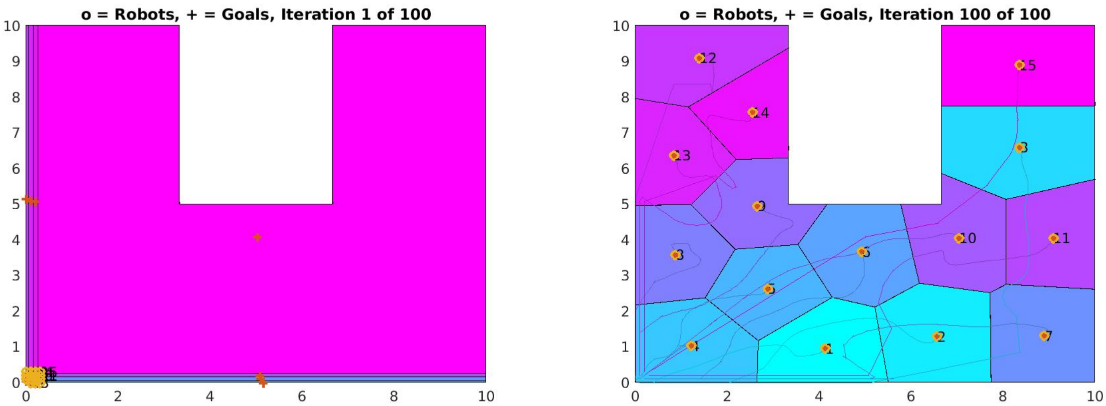

Figure 5.7: $n = 15$ agents initial position inside a unit square (left) and final position with path (right)

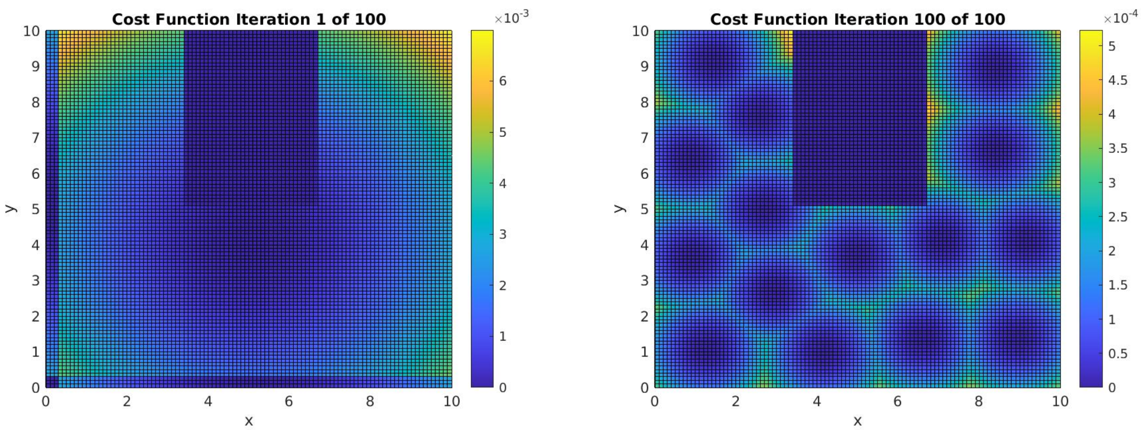

Figure 5.8: Distribution of cost function in the region initial (left) and final (right)

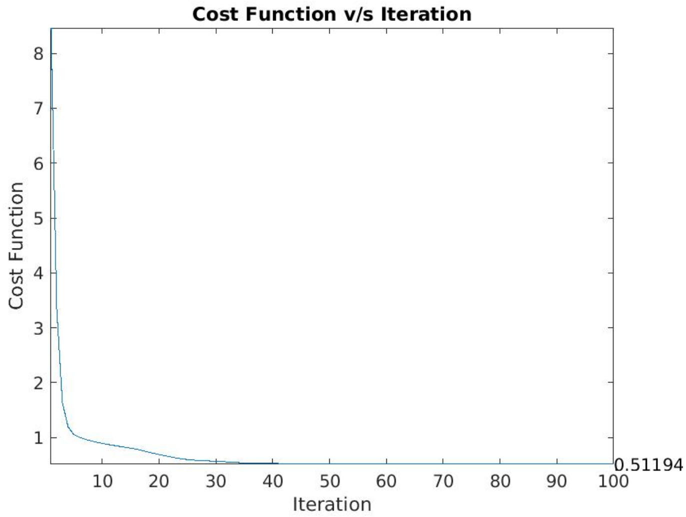

Figure 5.9: Cost function variation with iteration

Observations: It can be seen that for everything else remaining the same, the agents are staying within the valid region for this experiment Fig 5.7. And the final cost function value is almost same as the previous case.

5.4 Results

- It is verified through simulations that the proposed modified density function helps in keeping the agents in the valid region.
- The value of the cost function seems to be larger for the same number of iterations. But if an additional penalty is given the agents moving outside the valid region the cost function value in the second case would be an improvement.

Chapter 6

Summary and Conclusions

This section will summarize the entire study in terms of the contribution and also give the concluding remarks

6.1 Contributions

- The Coverage control of multi-agent systems was extended to include agents with heterogeneous agent kinematics. A theoretical based stability proof is provided for the same and verified through simulations.

- The heterogeneous coverage control framework was extended to adaptive case where the assumption of relative importance of points being known to the agents is relaxed. A theoretical stability proof is provided for the same and verified through simulations.

- A heuristic way of overcoming certain type of non-convexity in the area is proposed and verified through simulations. The intuition behind the same is discussed.

6.2 Scope for future research

- Automatic partitioning the area based on the capabilities of the different type of agents
- Determining the class of non-convexity that can be solved by the heuristic method proposed
- Developing the theory for consensus among the estimated densities learned by the agents

Appendix A

Background of Coverage Control

Some background theory regarding concepts used in coverage control will be explained in this appendix.

A.1 Voronoi Partitions

Voronoi partition is a scheme for partitioning a given space based on the concept of 'nearness' of points in a set of some finite number of predefined locations in the set. The concept finds application in many fields including image processing and sensor coverage.

Partition of a set **X** means a collection of subsets $\mathbf{W_i}$ of **X** with disjoint interiors such that their union is **X** itself. Let $\mathbf{Q} \in \mathbb{R}^d$ be a convex polytope. Let $\mathscr{P}$=$\{p_1, p_2, ..., p_N\}$, be a finite set of nodes (generators) $p_i \in \mathbf{Q}$ The Voronoi partition generated by $\mathscr{P}$ with respect to the Euclidean norm is the collection $\{Vi(\mathscr{P})\}_{\{1,2,...,N\}}$ defined as

$$V_i(\mathscr{P}) = \{q \in Q | \|q - p_i\| \leq \|q - p_j\|, _j \in \mathscr{P}\}$$

The Voronoi cell V_i is the collection of those points which are closer to p_i compared to any other point in P. In $\mathbb{R}^2$, the intersection of any two Voronoi partitions is either null, a line segment (boundary), or a point (vertex). The boundaries of the Voronoi cells are unions of convex subset of at most $d-1$ dimensional hyperplanes in $\mathbb{R}^d$ and the intersection of two Voronoi cells is either a convex subset of a hyperplane or a null set [26].

The Voronoi partition is generalized in a variety of ways. One of the standard generalizations is the generalization of the metric being used. Other widely used generalizations are multiplicatively and/or additively weighted Voronoi partitioning schemes which are defined as,

$$V_i(\mathscr{P}) = \{q \in Q | \alpha_i r_i + d_i \leq \alpha_j r_j + d_j, \forall j \neq i, j \in \{1,2,...,N\}\}$$

where, usually $\alpha_i > 0$ and $d_i \geq 0$, and are multiplicative and additive weights, respectively, and $r_i = \|p_i - q\|$. In the case of multiplicatively weighted Voronoi decomposition, the Voronoi cells no longer possess nice properties such as topological connectedness and non-emptiness. A Voronoi cell may be made up of a union of disjoint sets and may con- tain other Voronoi cells embedded inside it. Some of the cells could be empty sets. Some of the other generalizations are use of a pseudo-metric and using general objects like lines and polytopes as sites instead of points.

The Voronoi partition based approaches have several advantages but at the cost of increased computational overhead. Efficient and distributed algorithms/methods are available in the literature for Voronoi related computations reducing these overheads

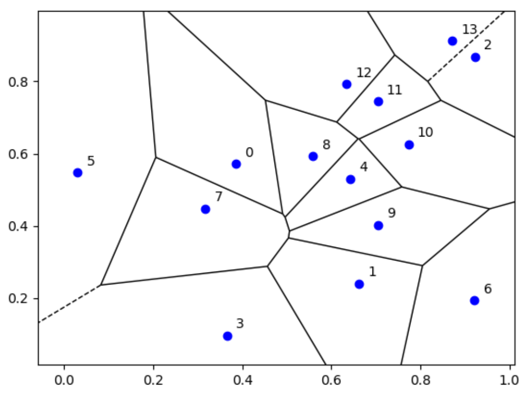

Figure A.1: Voronoi partition of a simple area by randomly placed 14 generators

The Voronoi partition has a dual problem known as Delaunay triangulation or the Delaunay graph. In a Delaunay graph two nodes are neighbors if the corresponding voronoi cells share a common edge.

A.2 Convex and Non-convex Area

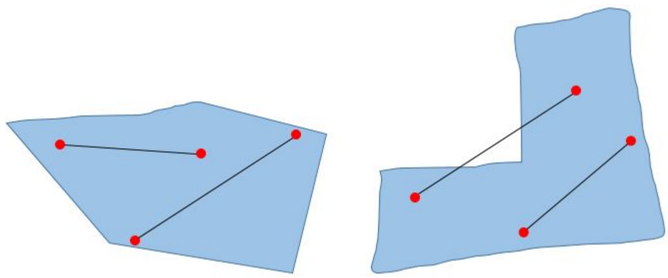

Figure A.2: Convex and non-convex region

Convex area: if any pair of points in the area can be connected by a line that lies entirely in the area
Non-convex area - at least a pair of points in the area cannot be connected using a line that lies entirely in the area
Lloyd's algorithm (move to centroid) that is used in the traditional coverage control will not give optimal solution when the region is non-convex. Limitation is caused by the following three cases:

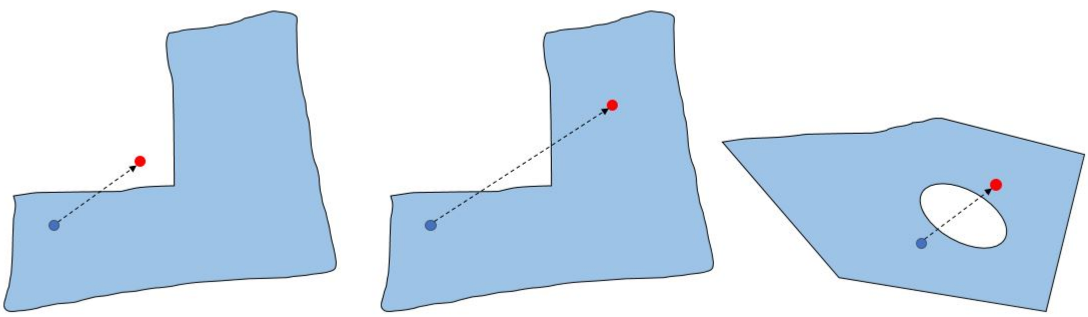

Figure A.3: Centroid outside area (left) Path to centroid outside area (middle) Path to centroid blocked by obstacle (right)

1. Centroid lying outside the area

2. Straight line path between centroid and and current position is going outside the area
3. Straight line path between centroid and and current position is blocked by an obstacle

A.3 Sensor Effectiveness

The variation of effectiveness of a sensor with distance depends on many factors including the type of sensor, manufacturer, age of the sensor etc. In this study it was assumed that the effectiveness of the sensor is proportional to

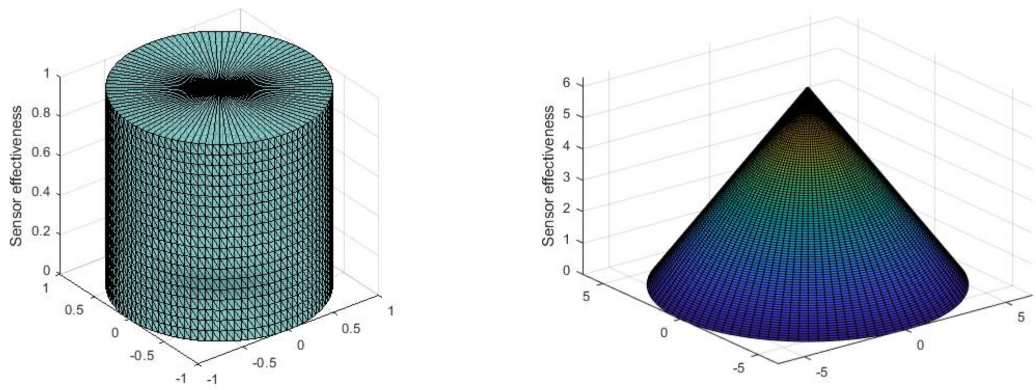

Figure A.4: Sensor effectiveness remaining constant inside sensing radius (left) and varying linearly with distance from the sensor (right)

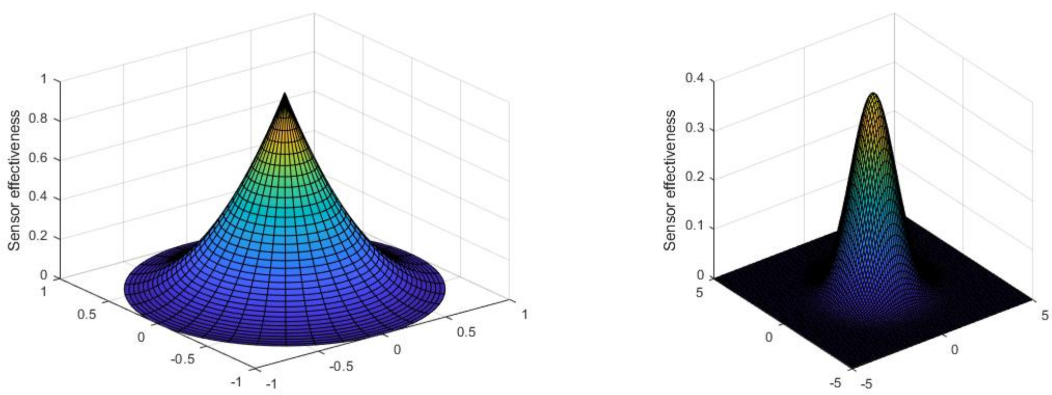

Figure A.5: Sensor effectiveness decaying with euclidean distance (left) and exponential to distance (right)

A.4 Assumption 1: Sum of gaussians

In Chapter 4 it was assumed that the distribution density function can be approximated as a sum of gaussian functions with fixed σ and μs distributed at equally spaced grid points on the area. The following figure is an example of such an assumption.

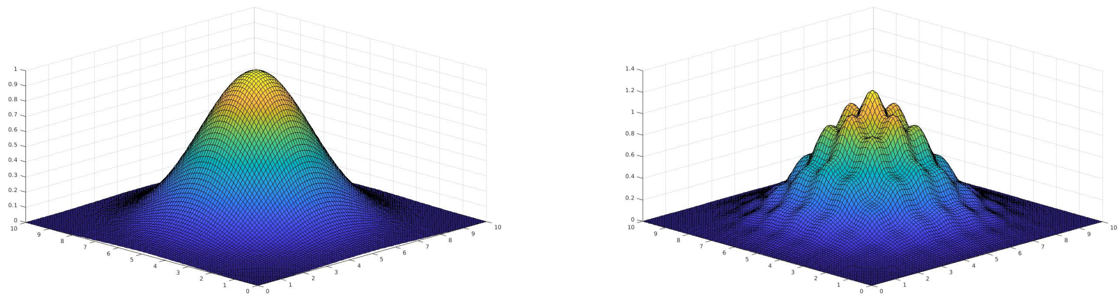

Figure A.6: Gaussian distribution (left) and gaussian distribution approximated by sum of gaussians (right)

This page was intentionally left blank.

Appendix B

Simulation Details

The details regarding simulation are given in this appendix.

B.1 Matlab Code for Heterogeneous Adaptive Coverage Control

All the relevant Matlab codes and simulink models have been uploaded to the following repository: `https://github.com/NijilGeorge/multiagent_systems_coverage_control`

B.2 Simulink Models for Agent Kinematics and Control

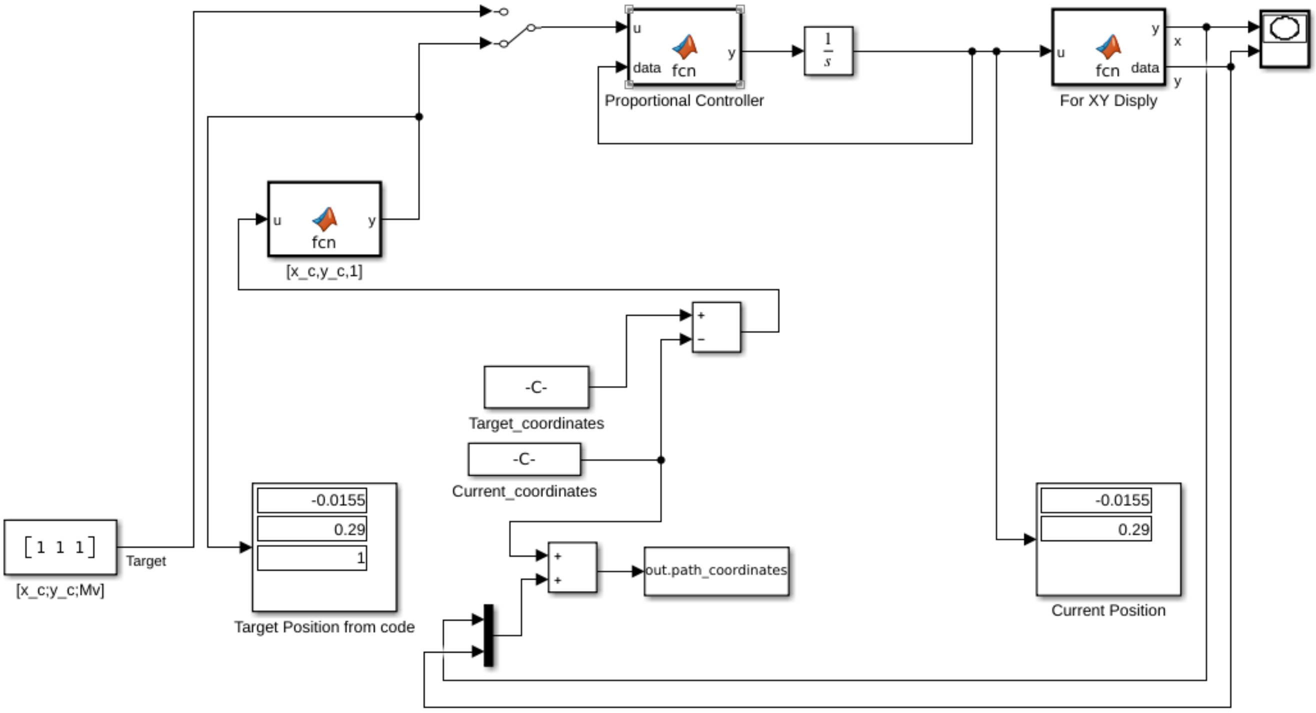

Figure B.1: Simulink model for single integrator with controller

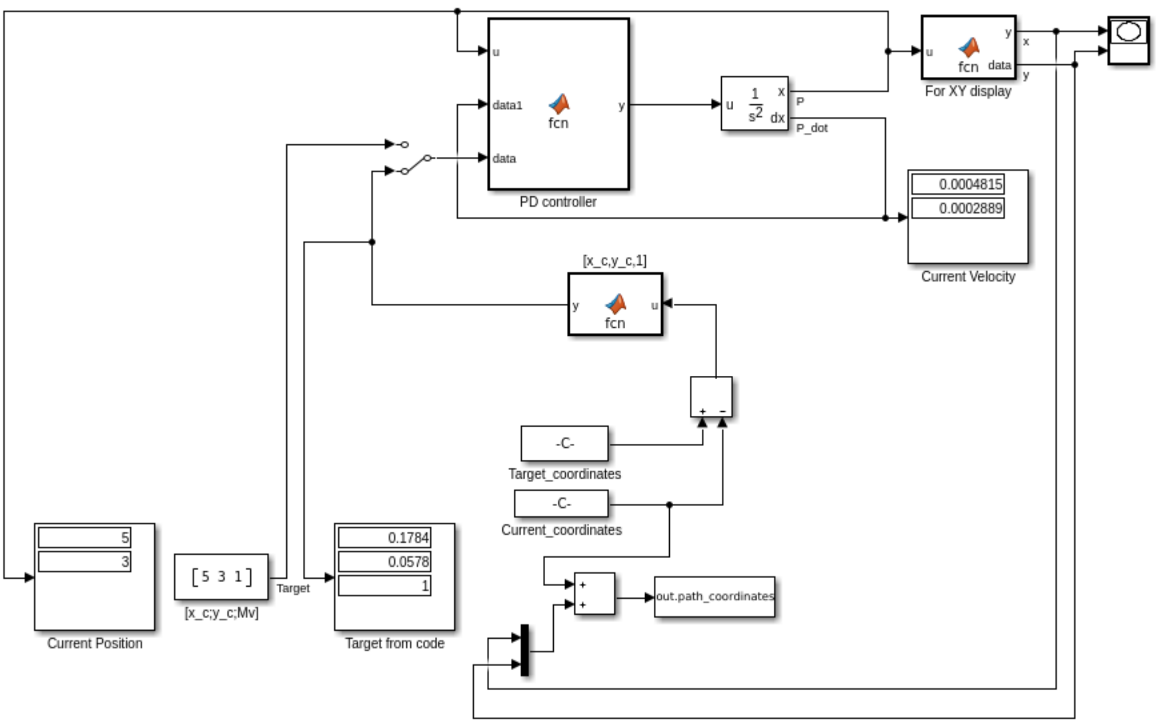

Figure B.2: Simulink model for double integrator with controller

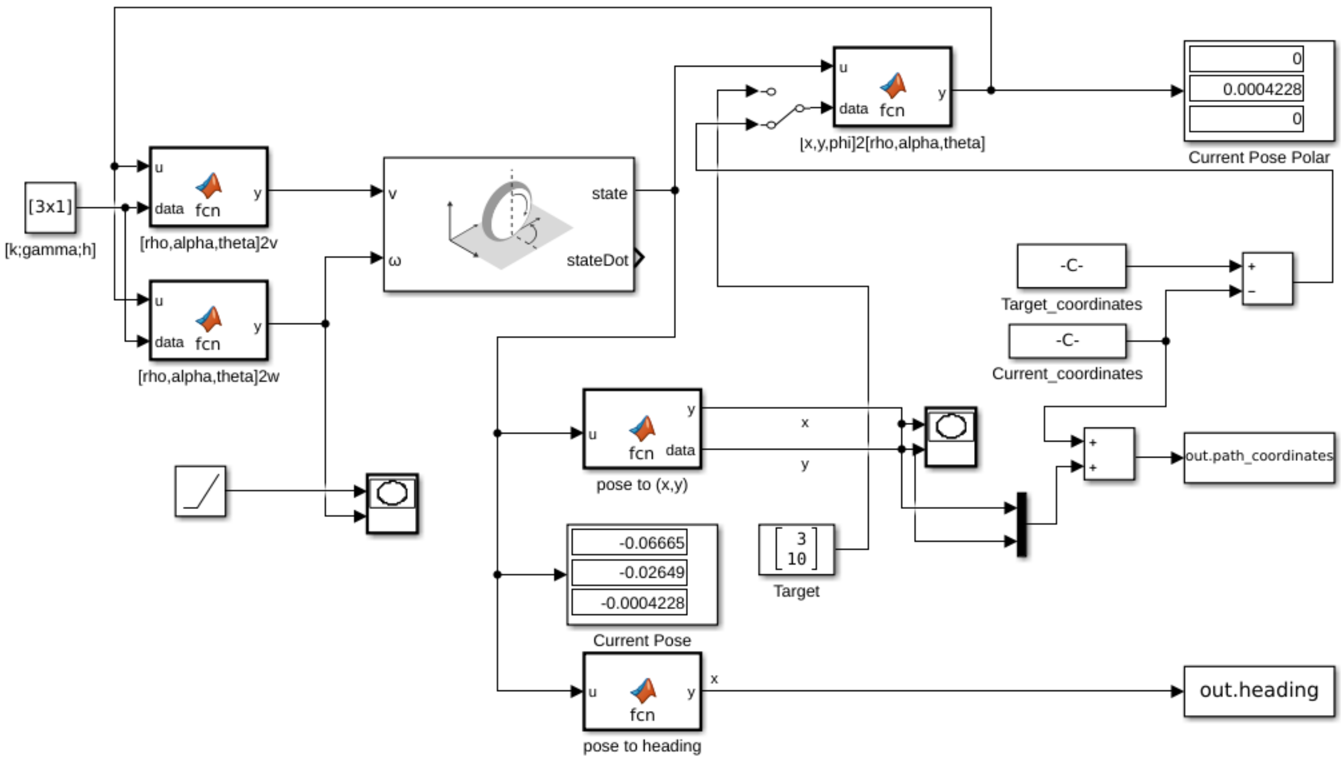

Figure B.3: Simulink model for unicycle with controller

This page was intentionally left blank.

References

[1] Andreas Breitenmoser, Mac Schwager, Jean-Claude Metzger, Roland Siegwart, and Daniela Rus. Voronoi coverage of non-convex environments with a group of networked robots. In *2010 IEEE international conference on robotics and automation*, pages 4982–4989. IEEE, 2010.

[2] Carlos H Caicedo-Nunez and Milos Zefran. Performing coverage on nonconvex domains. In *2008 IEEE International Conference on Control Applications*, pages 1019–1024. IEEE, 2008.

[3] Michalis Thanou, Yiannis Stergiopoulos, and Anthony Tzes. Distributed coverage using geodesic metric for non-convex environments. In *2013 IEEE international conference on robotics and automation*, pages 933–938. IEEE, 2013.

[4] Nikhil Nigam, Stefan Bieniawski, Ilan Kroo, and John Vian. Control of multiple uavs for persistent surveillance: Algorithm and flight test results. *IEEE Transactions on Control Systems Technology*, 20(5):1236–1251, 2011.

[5] J. Cortes, S. Martinez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255, 2004.

[6] Franz Aurenhammer. Power diagrams: properties, algorithms and applications. *SIAM Journal on Computing*, 16(1):78–96, 1987.

[7] O. Arslan and D. E. Koditschek. Voronoi-based coverage control of heterogeneous disk-shaped robots. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4259–4266, 2016.

[8] Wei Li and Christos G Cassandras. Distributed cooperative coverage control of sensor networks. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 2542–2547. IEEE, 2005.

[9] Andrew Howard, Maja J Matarić, and Gaurav S Sukhatme. Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. In *Distributed Autonomous Robotic Systems 5*, pages 299–308. Springer, 2002.

[10] María Santos, Yancy Diaz-Mercado, and Magnus Egerstedt. Coverage control for multirobot teams with heterogeneous sensing capabilities. *IEEE Robotics and Automation Letters*, 3(2):919–925, 2018.

[11] Luciano CA Pimenta, Vijay Kumar, Renato C Mesquita, and Guilherme AS Pereira. Sensing and coverage for a network of heterogeneous robots. In *2008 47th IEEE conference on decision and control*, pages 3947–3952. IEEE, 2008.

[12] Giovanni Franzini and Mario Innocenti. Effective coverage control for teams of heterogeneous agents. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 2836–2841. IEEE, 2017.

[13] Omur Arslan and Daniel E Koditschek. Voronoi-based coverage control of heterogeneous disk-shaped robots. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4259–4266. IEEE, 2016.

[14] Azwirman Gusrialdi, Sandra Hirche, Takeshi Hatanaka, and Masayuki Fujita. Voronoi based coverage control with anisotropic sensors. In *2008 American control conference*, pages 736–741. IEEE, 2008.

[15] Azwirman Gusrialdi, Takeshi Hatanaka, and Masayuki Fujita. Coverage control for mobile networks with limited-range anisotropic sensors. In *2008 47th IEEE Conference on Decision and Control*, pages 4263–4268. IEEE, 2008.

[16] Katie Laventall and Jorge Cortés. Coverage control by multi-robot networks with limited-range anisotropic sensory. *International Journal of Control*, 82(6):1113–1121, 2009.

[17] Mac Schwager, Daniela Rus, and Jean-Jacques Slotine. Decentralized, adaptive coverage control for networked robots. *The International Journal of Robotics Research*, 28(3):357–375, 2009.

[18] Rihab A Razak, Srikant Sukumar, and Hoam Chung. Decentralized adaptive coverage control of nonholonomic mobile robots. *IFAC-PapersOnLine*, 49(18):410–415, 2016.

[19] Jose-Marcio Luna, Rafael Fierro, Chaouki Abdallah, and John Wood. An adaptive coverage control algorithm for deployment of nonholonomic mobile sensors. In *49th IEEE Conference on Decision and Control (CDC)*, pages 1250–1256. IEEE, 2010.

[20] Michele Aicardi, Giuseppe Casalino, Antonio Bicchi, and Aldo Balestrino. Closed loop steering of unicycle like vehicles via lyapunov techniques. *IEEE Robotics & Automation Magazine*, 2(1):27–35, 1995.

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

[22] Manuel Boldrer, Daniele Fontanelli, and Luigi Palopoli. Coverage control and distributed consensus-based estimation for mobile sensing networks in complex environments. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 7838–7843. IEEE, 2019.

[23] Carlos Humberto Caicedo-Núnez and Milos Zefran. A coverage algorithm for a class of non-convex regions. In *2008 47th IEEE Conference on Decision and Control*, pages 4244–4249. IEEE, 2008.

[24] Joseph W Durham, Ruggero Carli, Paolo Frasca, and Francesco Bullo. Discrete partitioning and coverage control with gossip communication. In *Dynamic Systems and Control Conference*, volume 48937, pages 225–232, 2009.

[25] Hamed Fathalizadeh Parapari, Farzaneh Abdollahi, and Mohammad Bagher Menhaj. Coverage control in non-convex environment considering unknown non-convex obstacles. In *2014 Second RSI/ISM International Conference on Robotics and Mechatronics (ICRoM)*, pages 119–124. IEEE, 2014.

[26] Qiang Du, Vance Faber, and Max Gunzburger. Centroidal voronoi tessellations: Applications and algorithms. *SIAM review*, 41(4):637–676, 1999.

This page was intentionally left blank.

Acknowledgments

I wish to record a deep sense of gratitude to **Prof. Dwaipayan Mukherjee**, my supervisor for his valuable guidance and constant support at all stages of my M.Tech study and related research. I would also like to thank my family and friends for being there for me as a support structure.