İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

MULTI ROBOT COVERAGE METHODS

M.Sc. Thesis by
Mert TURANLI

Department : **Control and Automation Eng.**

Programme : **Control and Automation Eng.**

JUNE 2011

İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

MULTI ROBOT COVERAGE METHODS

M.Sc. Thesis by
Mert TURANLI
(504081119)

Date of submission : 27 June 2011
Date of defence examination: 08 June 2011

Supervisor (Chairman) : Prof. Dr. Hakan TEMELTAŞ (ITU)
Members of the Examining Committee : Prof. Dr. İbrahim EKSİN (ITU)
Assis. Prof. Dr. O. Kaan EROL (ITU)

JUNE 2011

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

ÇOKLU ROBOT SİSTEMLERİNDE HARİTA ÇEVRELEME YÖNTEMLERİ

YÜKSEK LİSANS TEZİ

Mert TURANLI
(504081119)

Tezin Enstitüye Verildiği Tarih : 27 Haziran 2011
Tezin Savunulduğu Tarih : 08 Haziran 2011

Tez Danışmanı : Prof. Dr. Hakan TEMELTAŞ (İTÜ)
Diğer Jüri Üyeleri : Prof. Dr. İbrahim EKSİN (İTÜ)
Yrd. Doç. Dr. O. Kaan EROL (İTÜ)

HAZİRAN 2011

FOREWORD

I would like to express my deep appreciation and thanks for my advisor. Also, I would like to thank to TUBITAK for supporting my thesis work during my Master's education. At last, I cannot forget the support of my family during every phase of my Master's work.

May 2011

Mert Turanlı

Control Engineer

TABLE OF CONTENTS

Page

ABBREVIATIONS

DISC	**:** Discoverage
VIG	**:** Variable Information Gains
VIGA	**:** Variable Information Gain Approach

LIST OF TABLES

Page

LIST OF FIGURES

Page

MULTI ROBOT COVERAGE METHODS

SUMMARY

Multi robot exploration and coordination problems has been being studied for more than one decade. The main reason of using multiple agents is because most of the problems can be solved by using more than one agent more efficiently. This is the case in coverage problem, too. The multi robot coverage problem is one of the extensions of the multi robot exploration and coordination topics. In multi robot coverage problem, the robots are trying to navigate in unexplored regions in coordination with each other in order to maximize the covered area in optimal time. In other words, the agents are running optimization algorithms to find an optimal control input under some constraints such as energy or collision distance constraints.

In this work, a number of selected coverage methods is studied with simulation results. The simulation environment is designed in MATLAB by using computer graphics and image processing functions. There are also some contributions to the literature in this thesis, such as a frontier based coverage approach with collision and obstacle avoidance. Apart from these contributions, the methods are compared and the results are discussed. Moreover, a method considering anisotropy of the sensors is presented.

ÇOKLU ROBOT SİSTEMLERİNDE HARİTA ÇEVRELEME YÖNTEMLERİ

ÖZET

Çoklu robot keşfetme ve koordinasyon problemleri son birkaç yıldır üzerinde çalışılan yeni konulardandır. Problem çözümlerinde birden fazla robotun kullanılmasının temel nedeni, bazı problemlerde bu yaklaşımın daha iyi sonuçlar vermesidir. Harita çevreleme probleminde de bu durum benzerdir. Çoklu robot harita çevreleme problemi, çoklu robot keşfetme ve koordinasyon problemlerinin bir uzantısı olup, robotların keşfedilmemiş alanlarda birbirleri ile etkileşim halinde hareket etmeleri amacını taşır. Birlikte çalışan bu robotların ortak hedefi en kısa zamanda haritayı tümüyle çevrelemektir. Diğer bir deyişle, her bir robot optimal kontrol işaretini hesaplamak için belli kısıtlar altında (enerji ya da çarpışma mesafesi gibi) birer optimizasyon algoritması koşturur.

Bu çalışmada, seçilen birkaç harita çevreleme yöntemi simulasyon sonuçları ile incelenmiştir. Simulasyon ortamı MATLAB üzerinde bilgisayar grafikleri ve görüntü işleme işlevleri kullanılarak gerçeklenmiştir. Ek olarak, bu çalışmanın literatüre bazı katkıları da mevcuttur. Bunlar, sınır yöntemine dayalı harita çevreleme yöntemine eklenen çarpışma önleme ve engelden kaçınma algoritmalarıdır. Bu katkıların yanında, seçilen yöntemler karşılaştırılmış ve sonuçlar tartışılmıştır. Ayrıca, sensörlerin yönden bağımsızlığını ele alan bir harita çevreleme yöntemi incelenmiştir.

1. INTRODUCTION

1.1. Motivation

Multi robot coordination problems are challenging and new research topics due to their essence in coordinated actions in some important applications. Most of the problems discussed in literature consider failure of vehicle control in multi robot group and communication losses among the robotic agents. In addition to these problems, some tasks related with the exploration of the environment, i.e. search and rescue, surveillance, recovery and environmental monitoring need more than one robotic agent. The other case in which it is impossible to use a single robotic agent is manipulation of objects with excessive geometric shapes in a constrained environment. In multi robot exploration problems, the solution is restricted by operation time, fuel or energy limitations and communication ranges among the robots. Hence, the multi robot exploration problem may be considered as an optimization problem with multiple constraints.

1.2. Background

A lot of work has been done on the multi robot problems such as coordination, localization, exploration, mapping, formation and path planning topics. To name a few from the robotics literature; a new approach to multi robot exploration under global uncertainty using particle filters [1], a cooperative path planning algorithm and communication protocol for multi agent sensor systems robust to failure of some agents [2], a multi agent coalition formation algorithm [3], a distributed algorithm for deploying multi robot systems consisting of wheeled mobile robots under nonholonomic constraints [4], a market-based approach Contract Net Protocol used for multi robot task coordination [5], an extended learning and imitation approach based on Evolving Societies of Learning Autonomous Systems method for multiple robots [6], a new approach for developing efficient methods for multi robot active and passive sensing applications [7], an improved Monte Carlo localization using self-adaptive samples [8], evolutionary robotics approach applied to a multi-robot system for autonomous task allocation [9], task distribution among multiple robot teams and coordination within the teams using coalition approach [10], leader-follower method with ad-hoc network to solve the real-time formation problem [11], artificial potential field method with new SWARM and SPREAD potential functions to solve distributed path planning problem for multi robots in real-time [12], cooperative simultaneous localization and mapping (SLAM) problem with Extended Kalman Filter using multiple robots [13] are some of the previous papers about multi robot problems.

Sensor networks and static sensor coverage problem has also found great interest in the last decade, as in the work related to optimal sensor allocation [14]. Multi robot coverage problem can be counted as a special case of sensor networks studies. In this case, each robotic agent or mobile sensor nodes relocate their positions and orientations to minimize or maximize a predefined cost function under some constraints.

Not only the sensor coverage problem, but also the mobile robot exploration problem is one of the significant open problems in the robotics area that is mainly discussed in this paper. In the literature, there are various number of works related to exploration topic such as frontier based approach used for exploration [15], Discoverage, multi robot exploration based on the Coverage problem [16], Voronoi based distributed sensing networks [14], mine detection using path planning [17], mobile sensor coverage problem [18], geometric, probabilistic and potential field approaches to multi-robot deployment [19] and coverage control over large scale task domains using multi-agent sensor network [20]. However, all of these studies do not consider dynamically updating the expected information gain function in coverage formulation which increases the performance of the optimization considerably.

Similarly, multi robot coverage problem has gained a lot of attention in the literature for the last decade. Biologically inspired neural network approach with obstacle avoidance for cleaning robots [21], optimal coverage for sensor allocation with Voronoi by using gradient descent algorithms [14], artificial potential fields based coverage for mobile sensor networks under given neighborhood constraints [22] are some of the works found in literature. Also, some other works related to coverage problem are; a cell decomposition approach to coverage problem with multi robot teams including allocation planning of these teams under communication restrictions [23], a new method for terrain coverage with multiple robots named multi robot forest coverage compared to another method called multi robot spanning tree coverage [24]. Therefore, one of the works considers the multi robot boundary inspection problem by using a coverage algorithm which is posed as k-Rural Postman Problem (kRPP) and uses the solution of kRPP to plan inspection routes which provide coverage of the boundary [25]. Additionally, another work shows the robustness and efficiency in multi-robot coverage algorithms based on spanning-tree coverage of approximate cell decomposition and examines the importance of the initial positions of the robots [26]. Another work is about a distributed coverage control for cooperating mobile sensor networks using a gradient-based algorithm and maximizes the joint detection probabilities of random events considering the communication costs [27]. Also, building efficient coverage paths for teams of robots by constructing spanning trees which minimizes the coverage time [28], a new robust multi robot online coverage algorithm based on approximate cell decomposition [29], another cell based multi robot complete coverage algorithm using Boustrophedon decomposition [30], dynamic coverage control which does not depend on gradient based methods [31], an extension of the multi robot forest coverage algorithm to terrains with non-uniform traversability [32], a multi robot coverage algorithm providing persistent coverage period considering the limitations in the communication and sensing ranges [33], a new strategy for the exploration of an unknown environment with a multi robot system with communication restrictions and maximum distance constraint among robots [34] are some of the related works. Moreover, a distributed control algorithm based on centroidal Voronoi diagram for covering specific area of heterogeneous groups of robots under sensory and communication range constraints [35], fast multi robot coverage under energy, velocity and working time constraints [36], a coverage algorithm for multi robot systems in an unknown environment searching for fixed or moving targets under communication range and sensory constraints [37], sensor based complete coverage using generalized Voronoi diagrams considering energy capacities of mobile robots [38], multi agent coverage approach as an extension of single robot BSA coverage algorithm [39], generalized Voronoi diagram based coverage planning with replanning approach for handling unknown environments considering energy constraints [40], a coalitional game theory named as weighted voting game technique based coverage [41], a cognitive based adaptive optimization algorithm for coverage in order to maximize the area monitored by a multi robot team [42] can be counted a group of works about the coverage problem.

In this thesis, in addition to the idea of autonomous navigation and exploration with the control law in DisCoverage [16], a new approach for updating and selecting the information gains in the objective function is given. In the Haumann's method, the information gain part of the objective function is considered a static function whereas in our approach the information gain function consists of two parts; a static and a dynamic one. Using this new method improves the performance by making the explored region ratio higher as the iteration of the optimization advances.

In addition to the information gain approach, there are some other assumptions in this work. The working space is considered as obstacle-free and the dynamics of the robot is taken as a simple differential kinematic model of a point mass with a sensing radius. Another assumption is the map and information gains are synchronized among all the robots.

1.3. Previous Works

There are a lot of experimental works about multi robot and Coverage algorithms in the literature. For example, Voronoi based coverage control with anisotropic sensors [43] is done in Fujita Laboratory, as shown in Figure 1.1.

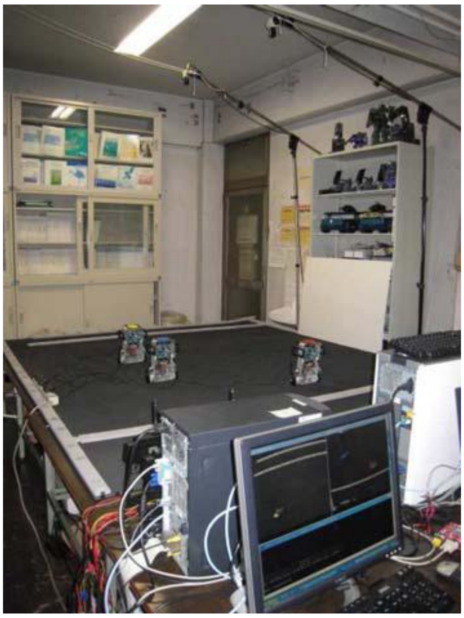

Figure 1.1 : Coverage Control in Fujita Laboratory at Tokyo Institute of Technology

Also, in multi agent coordination problem, there are a lot of experimental works in the robotics literature. One of them is Multi UAV Quadrotor Platform of MIT, as shown in Figure 1.2. Some of the works done on this platform are [44] and [45].

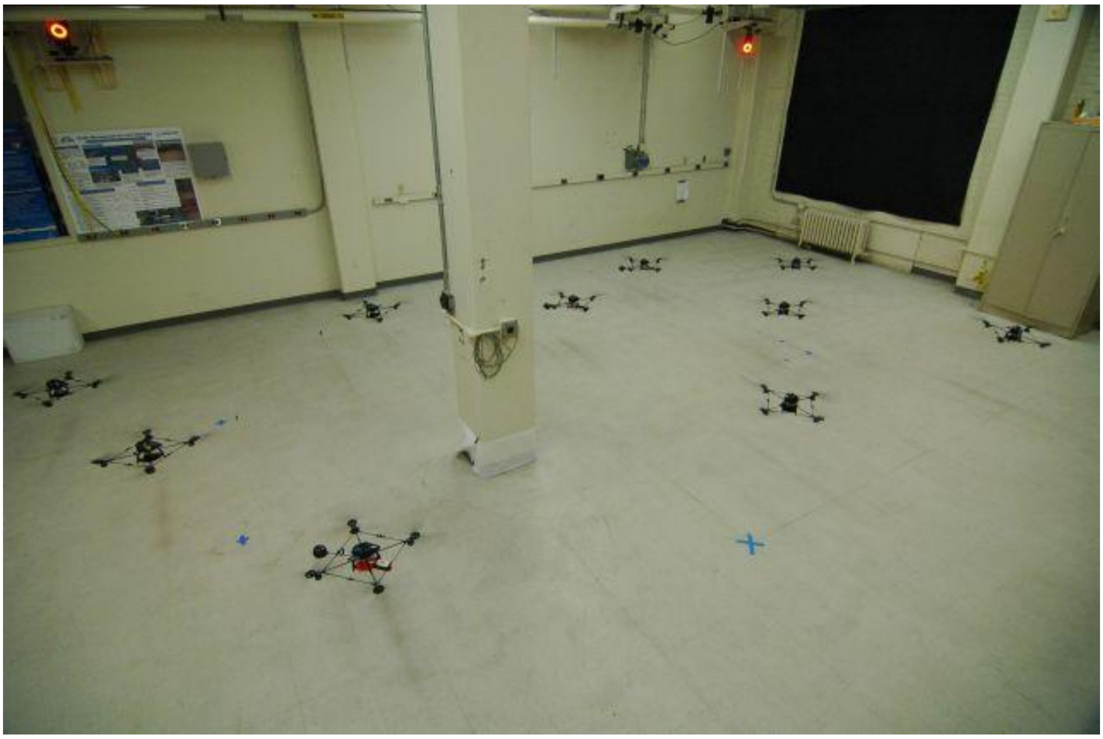

Figure 1.2 : Multi UAV Quadrotor Platform in MIT Aerospace Controls Laboratory

A research group studying mid-size soccer playing robots is CORAL Research Group at Carnegie Mellon University. Their robots called HammerHeads are shown in Figure 1.3. Some of the related publications about multi agent systems are [46] and [47].

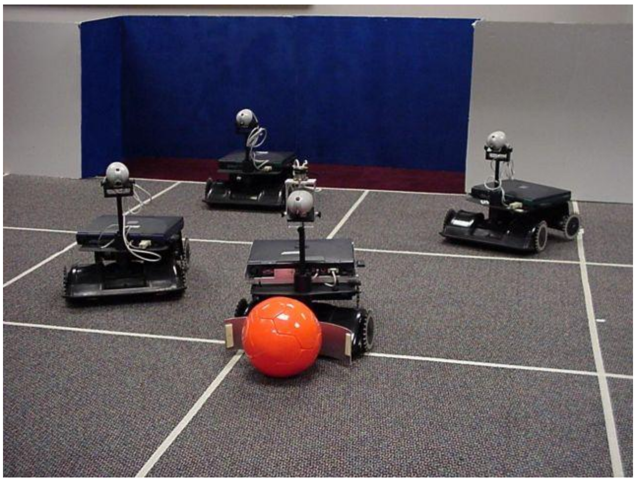

Figure 1.3 : HammerHeads Mid-Size Robots of CORAL Research Group at Carnegie Mellon University

Figure 1.4 : MuRoLa - Multi Robot Laboratory at Technische Universität München

In Figure 1.4, Multi Robot Laboratory at TU München can be seen. Some of the works about multi robot problems are about load sharing in human-robot cooperation [48] and a real time control architecture for multi robot systems [49].

Figure 1.5 : AI Robotics Laboratory at Eskisehir Osmangazi University

Some works related to multi robot coverage path planning [38] and sensor coverage [40] are done in AI Robotics Laboratory, as shown in Figure 1.5.

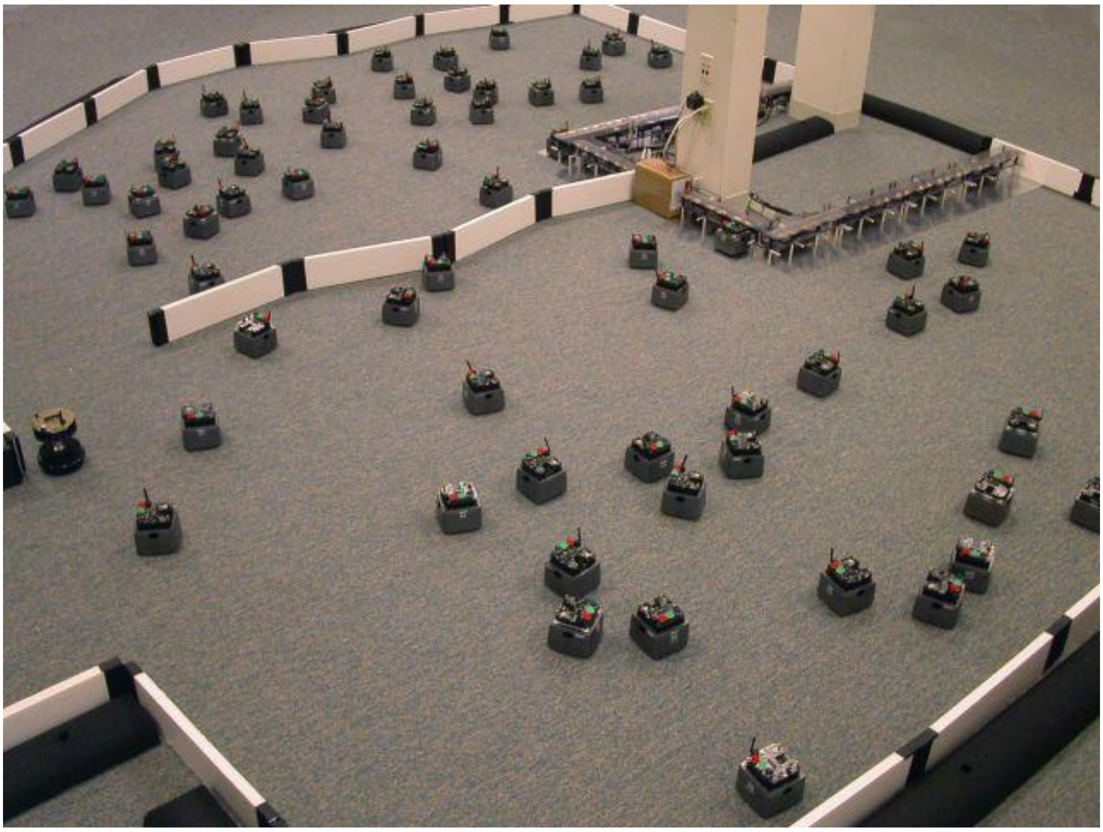

Figure 1.6 : SwarmBots in Multi Robot Systems Laboratory at Rice University

The Multi Robot Systems Laboratory has various works related to multi robot distributed coverage control such as [50] and [51], as shown in Figure 1.6.

1.4. Organization of the Thesis

This thesis is organized as follows; in the second section, the multi robot coverage problem will be introduced by giving the necessary objective functions for optimization, sensor and robot models, Voronoi tessellations, information gains, collision and obstacle avoidance and numerical optimization algorithms used. Then, in the third section, selected coverage methods are investigated by introducing their theory. In the fourth section, the effect of sensor characteristics on coverage problem will be described with selected works in the literature. Also, in the fifth section simulation results of the coverage methods applied to a mine exploration problem will be given with single and multi vehicle cases. Finally, the results of the simulations and the thesis will be discussed in the conclusion section.

2. MULTI ROBOT COVERAGE PROBLEM

The Coverage problem is about covering and exploring the map with best possible performance. The performance criteria may be time or energy effort under some constraints. Some of these constraints can be energy (fuel or charge of battery), velocity or acceleration limits, kinematic constraints or obstacles. This Coverage problem can be formulated and solved by using constrained or unconstrained optimization. The special multi robot case involves groups of robots cooperating with each other in order to cover or explore the given map in finite time. The goal is to find optimal control inputs for distributed agents. As a result, an optimal trajectory for each agent can be found.

The basics of two important categorizations of coverage problems in the literature will be discussed in this section. For Voronoi based Coverage solution, preliminary information of Voronoi regions is given. Also, for the Coverage problem with isotropic/anisotropic sensor types, introductory information is mentioned.

2.1. Problem Definition

Let ∂S_i be the frontier curve obtained from the formulation of the borders between explored and unexplored space over the Voronoi region of vehicle i. The function f can be defined as the performance index of exploration process. In one of the previous works, namely DisCoverage, the used solution is to find optimal heading angles of vehicle moving with a constant linear velocity. The same approach is used for navigation and exploration in this work. However, the information gain part of the function f in DisCoverage is considered as a static function $\Phi(q)$. In this work, this approach is extended in order to use dynamically updated information gain function consisting of static and dynamic parts. The advantages of using dynamic gain function will be given with comparisons in the simulation section.

Figure 2.1 shows the visual explanation of the terms used in the coverage formulation. Frontiers are the curves separating the explored and unexplored regions on the map. Explored areas are defined as the areas that are sensed by one of the robots at least once. The mines are the information to be found in the mine clearance problem that will be discussed in the simulation studies.

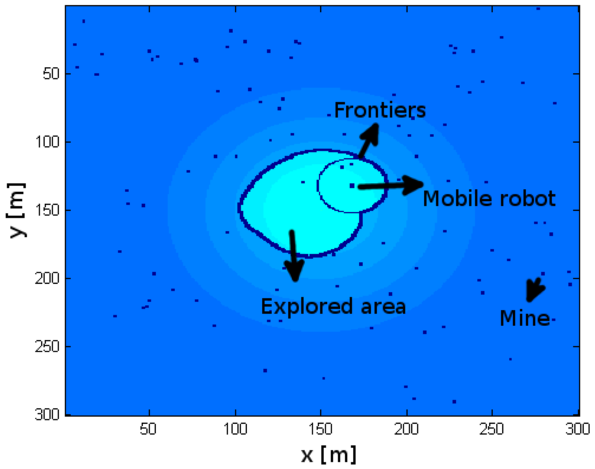

Figure 2.1 : Visual explanation of simulation terms

Let p_i be the position vector and θ_i be the orientation of the agent *i*. The objective function to be optimized for i^{th} agent becomes a curve integral expression over ∂S_i:

$$H_i(p_i, \theta_i) = \int_{\partial S_i} f(p_i, q, \theta_i)\Phi(q)dq \qquad \textbf{(2.1)}$$

We can define the angle α_i as follows:

$$\alpha_i = atan2(p_{i_y} - q_y, p_{i_x} - q_x) \qquad \textbf{(2.2)}$$

The function *f* consists of two parts, an angular and a distance component:

$$f(p_i, q, \theta_i) = \exp\left(-\frac{\alpha(p_i, q, \theta_i)^2}{2\alpha^2}\right)\exp\left(-\frac{\|q - p_i\|_2^2}{2\rho^2}\right) \qquad \textbf{(2.3)}$$

Individual agents will run the optimization in a distributed manner. Let *N* be the number of robots, $P = \{p_1, \ldots p_N\}$ denoting the position vectors of the agents and $\Theta = \{\theta_1, \ldots \theta_N\}$ representing the orientation angles. Maximizing H_i in parallel means also maximizing the total objective function *H* given below:

$$H(P,\Theta) = \sum_{i=1}^{N} H_i(p_i,\theta_i) \tag{2.4}$$

$$\{\theta_i^*, H_\mathrm{i}^*\} = \underset{\theta_i}{\operatorname{argmax}} H_i(p_i,\theta_i) \tag{2.5}$$

The optimization problem is converted into finding optimal orientation angles θ_i^* in parallel which maximize the objective function in order to cover the area.

2.2. Robot and Sensor Models

In this section, the mathematical model of vehicles used in this work will be given with sensor models. In the proposed method, the vehicle and sensor models play important roles because they enable the agent to navigate and percept its environment. The algorithms used in this work are dependent to these mathematical models.

2.2.1. Robot Model

The motion of the robots is considered as kinematic motions. This means that given angular and linear velocities, the robot will reach the desired speeds by using its wheels. This motion is modeled with a differential kinematic model since the type of robots considered is differential drive.

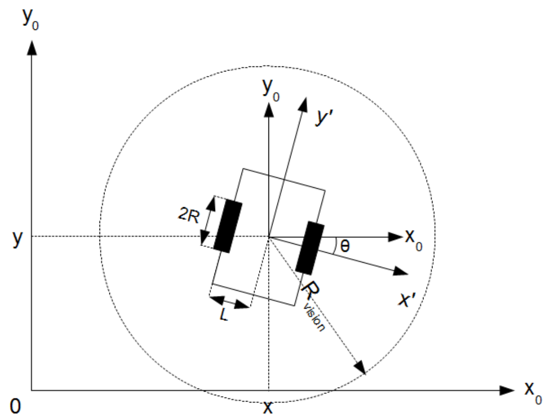

Figure 2.2 : Mobile Robot and Sensor Model

The kinematic model used in simulations is given as below:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} cos\theta & 0 \\ sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{2.6}$$

As shown in Figure 2.2, x, y and θ are the position and orientation states of the robot. The linear and angular velocity inputs are v and ω, respectively.

Additionally, a dynamic model of the differential drive mobile robot can be given as below, where q_R and q_L are the angular positions of the right and left wheels, M is the mass and inertia matrix, h is the matrix involving *Coriolis* and *Centrifugal* effects, F_{fr} is the friction vector and τ is the torque signal vector applied to the wheels [52].

$$q = [q_R \quad q_L]^T \tag{2.7}$$

$$M\ddot{q} + h(\dot{q})\dot{q} + F_{fr}(\dot{q}) = \tau \tag{2.8}$$

As mentioned in the paper, the terms in the matrices M and h are given below where m the mass of vehicle, m_w the mass of each wheel, x_c and y_c are the coordinates of the center of the gravity, I_0 is the inertia of the vehicle around the COG, I_w is the moment of inertia around the turning axis of each wheel.

$$I = I_0 + 2m_w L^2 + m(x_c^2 + y_c^2) \tag{2.9}$$

$$M_{11} = \frac{R^2}{4}m + \frac{R^2}{4L^2}I + R^2 m_w - \frac{R^2}{2L}my_c + I_w \tag{2.10}$$

$$M_{12} = M_{21} = \frac{R^2}{4}m - \frac{R^2}{4L^2}I + \frac{R^2}{2}m_w \tag{2.11}$$

$$M_{22} = \frac{R^2}{4}m + \frac{R^2}{4L^2}I + R^2 m_w + \frac{R^2}{2}my_c + I_w \tag{2.12}$$

$$h_{11} = h_{22} = 0 \tag{2.13}$$

$$h_{12} = -h_{21} = (\dot{q}_R - \dot{q}_L)\frac{R^3}{2L}my_c \tag{2.14}$$

Similarly, the friction torque vector has the coefficient matrices F_v and F_s, including the static and viscous friction coefficients, respectively. The friction model used in this work is given as below.

$$F_{fr}(\dot{q}) = F_v\dot{q} + F_s sgn(\dot{q}) \tag{2.15}$$

The kinematic constraints of the differential drive vehicle can be given with the velocity signals in the dynamic model are given as below. The equations are valid if there is no lateral and longitudinal slip in the wheels.

$$v = \frac{R}{2}(\dot{q}_R + \dot{q}_L) \tag{2.16}$$

$$\omega = \frac{R}{2L}(\dot{q}_R - \dot{q}_L) \tag{2.17}$$

The position vector of the vehicle including x, y and θ variables can be obtained by solving the kinematic state space equations by integration for the no-slip case.

2.2.2. Sensor Model

In the mobile robot coverage problem, the sensor model is also an important issue. The reason is that the performance of algorithm can be affected by the sensor performance which may depend on the distance and orientation to the target to be sensed [53]. These types of sensors are modeled as anisotropic sensors. For those sensor models, the area sensed by an agent can be affected from the sensor performance which is a function of the distance and orientation of the robot. As mentioned in the paper, camera, radar and directional microphone are various examples of anisotropic sensors.

In contrast, isotropic sensor models do not consider the performance degradation of the sensors with respect to distance or orientation. There are also various kinds of sensors in this category such as short range LASER scanners. As shown in Figure 2.2, the isotropic sensor model has a circular sensing region around the robot with a sensing radius R_{vision}. The performance of the sensor over this circular region is equal in all directions and distances.

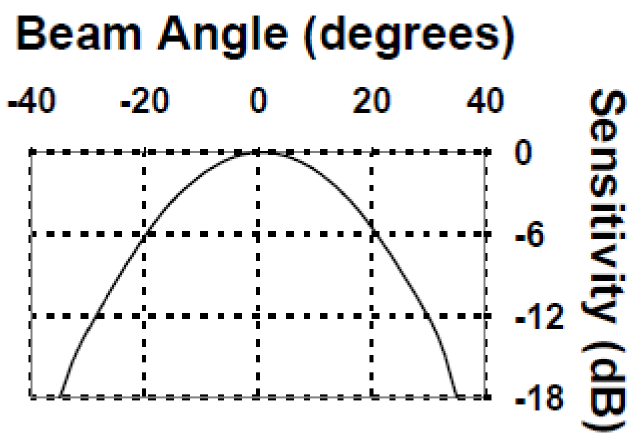

Figure 2.3 : Sensivity characteristic of an ultrasonic transducer

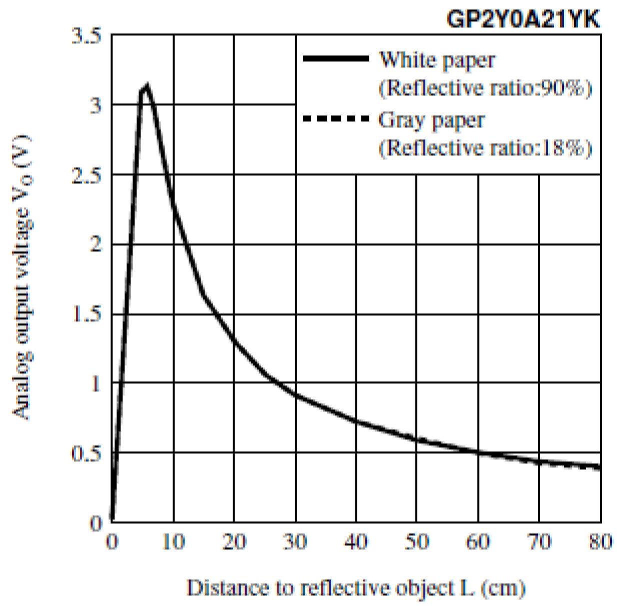

Figure 2.4 : Output characteristic of a Sharp infrared distance sensor

As it can be seen from Figure 2.3, ultrasonic transducers have performance characteristics changing with respect to the angle of coming sound wave. Similarly, in Figure 2.4, the output of an infrared ranger changes with respect to the measured distance of the object. These sensors can be considered and modeled as anisotropic sensors.

2.3. Voronoi Regions

A Voronoi diagram is a collection of regions that contain the nearest points to the points in the corresponding Voronoi region. It can give information about neighborhoods, closest points, communication costs etc. In this chapter, the Voronoi regions will be explained with examples. An example diagram can be seen from

Figure 2.5. Other examples are Figure 2.6 and Figure 2.7 with different number of points.

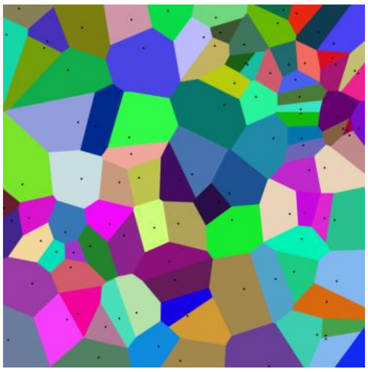

Figure 2.5 : Voronoi Diagram Example

For giving an example for Voronoi diagrams, consider planning of a supermarket chain. The aim is to open a new branch at a certain place on the map in order to maximize the profit. The expected profit depends on the number of customers attracted by the supermarket. For this purpose, we can define central places called sites around the living areas of the potential customers. The main assumption in this problem is that the people get their products at the nearest site.

A geometric model such as given in the supermarket chain example is called the Voronoi assignment model. In this model, every point is assigned to the nearest region. The regions created by this model are also called Voronoi regions and the resulting geometric interpretation is a Voronoi diagram.

It is possible to get much information from a Voronoi diagram. In supermarket example, it can be derived that two sites having a common boundary are likely to compete with each other because of the customers living around this area.

2.3.1. History of Voronoi Diagrams

The history of the Voronoi diagrams goes back to Descartes (1644). In addition, they were used by Dirichlet in 2- and 3- dimensional forms in his research of quadratic forms (1850). In 1854, the Voronoi diagrams were used in order to show the relationship between being closer to Broad Street pump and Soho cholera epidemic by British physician John Snow. This disease caused death of large amount of people living close to this pump.

The name of the Voronoi diagrams comes from the name of a Russian mathematician Georgy Fedoseevich Voronoi. The general n-dimensional case of Voronoi diagrams was defined by him in 1908.

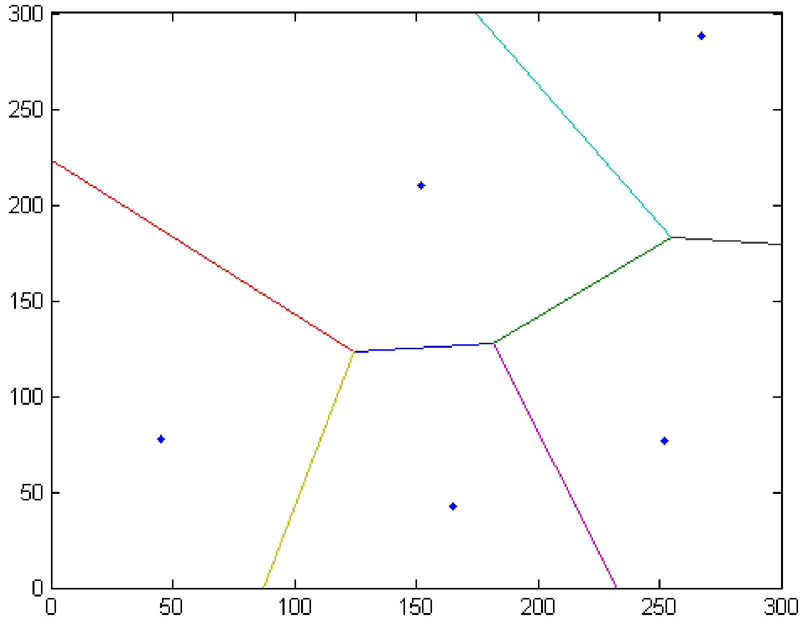

Figure 2.6 : Example Voronoi Diagram

Voronoi diagrams used in geophysics and meteorology areas to analyze experimental data such as rainfall measurements are called Thiessen polygons [54]. The name comes from the American meteorologist Alfred H. Thiessen. Also, Voronoi tessellations in condensed matter physics are known as Wigner-Seitz unit cells. Moreover, Brillouin zones is the special name of Voronoi tessellations of the reciprocal lattice of momenta.

2.3.2. Applications of Voronoi Diagrams

One application of Voronoi diagrams is nearest neighbor search. In order to get the nearest neighbor of a point, a point location data structure can be built based on a Voronoi diagram. Nearest neighbors search is used in numerous algorithms and methods such as Rapidly Exploring Random Trees path planner, vector quantization used in data compression, or simply finding nearest supermarket in a region.

Another application of Voronoi diagram can be the derivation of the capacity of a wireless network. In addition to wireless networks, in climatology, the Voronoi tessellations can be used to calculate the rainfall of an area (Thiessen polygons). Also, these diagrams are used in computer graphics in order to create organic looking graphics. Additionally, computational chemistry uses the Voronoi cells in Voronoi deformation density method which computes atomic charges according to the positions of the nuclei in a molecule. Moreover, in mining, the reserves of various resources such as minerals, valuable materials, can be estimated by using Voronoi polygons.

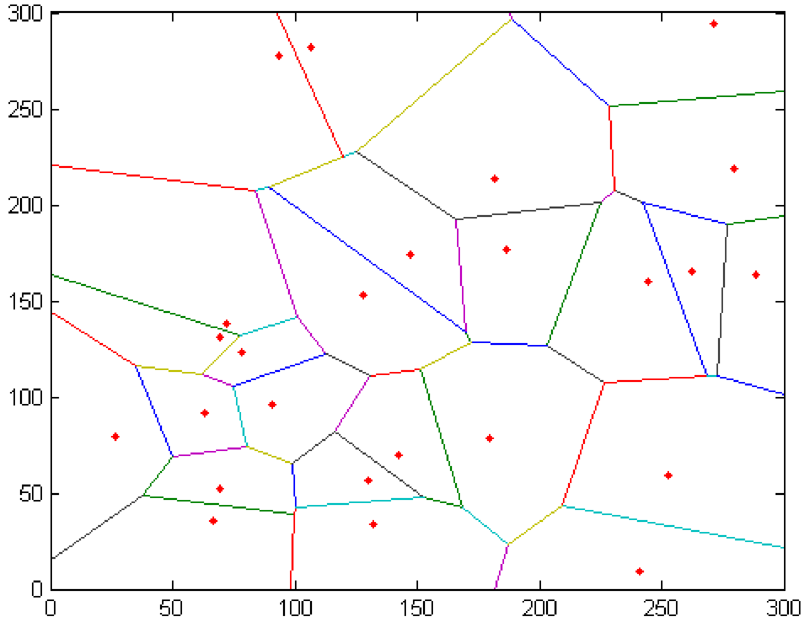

Figure 2.7 : Example Voronoi Regions

The most important application of the Voronoi tessellations is autonomous robot navigation because of their wide usage in robotics. They can be used in order to avoid from obstacles or partitioning the map in multi robot exploration problems in order to enable the robotic agents to cooperate, as discussed in this work.

2.3.3. Formal Definition of Voronoi Tessellations

For the point q in a Voronoi region of point p_i, the distances to p_i are minimal among the other points p_j. Defining it more formally, the Voronoi region of the point i is:

$$V_i = \{q \in Q, \|q - p_i\| \leq \|q - p_j\|, \forall j\} \quad \textbf{(2.18)}$$

Voronoi regions play an important role in partitioning a robot's area in the multi robot coordination problems. One of the reasons is the collision avoidance is guaranteed for the multi-robot coordination by using the Voronoi regions as mentioned in DisCoverage [16].

2.3.4. Computation of the Voronoi Diagrams

There are several algorithms in literature to compute the Voronoi tessellations [55]. Some of them are Bowyer–Watson algorithm, Fortune's algorithm and Lloyd's algorithm. In this chapter, the Fortune's algorithm will be explained.

Fortune's algorithm, also known as plane sweep algorithm, can compute the Voronoi diagram with an algorithm complexity *O(n logn)* [56]. Also, it is proved that Fortune's algorithm is optimal.

Some definitions and explanations are necessary before introducing the algorithm. The main strategy in the plane sweep algorithm is to sweep a horizontal line from top to bottom over the plane. Let the set $X = \{x_1, x_2, \dots, x_N\}$ be the point sites in the plane. The sweep line *l* moves from top to bottom over the plane. The closed half-plane above l is represented by l^+. Also, a beach line is defined as the sequence of parabolic arcs which bounds the locus of points that are closer to some site $x_i \in l^+$ than to *l*. Additionally, S is a balanced binary search tree taken as the beach line. Also, L is the event queue and its type is priority queue. The priority of an event changes with respect to its y-coordinate.

The Fortune's algorithm is given in the Table 2.1, Table 2.2 and Table 2.3 [56].

Table 2.1 shows the steps of the Fortune's algorithm including two subroutines; HandleCircleEvent and HandleSiteEvent. These subroutines are given in Table 2.2 and Table 2.3, respectively.

Table 2.1 : Fortune's algorithm

Input: X Output: Voronoi diagram *Vor(X)* given inside a bounding box in a doubly-connected edge list data structure
Initialize the event queue L with all site events **While** L is not empty **Do** Consider the event with largest y-coordinate in L **If** the event is a site event, occurring at site x_i **Then** HandleSiteEvent(x_i) **Else** HandleCircleEvent(x_i), where p_l is the lowest point of the circle causing the event Remove the event from L Compute a bounding box containing all vertices of the Voronoi diagram in its interior and attach the half-infinite edges to the bounding box by updating the doubly-connected edge list appropriately. Traverse the half-edges of the doubly-connected edge list to add the cell records and the pointers to and from them.

Table 2.2 : HandleCircleEvent function of Fortune's algorithm

HandleCircleEvent(x_i)
Search in S for the arc α vertically above x_l and delete all circle events involving α from L. Delete the leaf representing α from S. Update the tuples representing the breakpoints at the internal nodes. Perform rebalancing operations on S if necessary. Add the center of the circle causing the event as a vertex record in the Voronoi diagram structure and create two half-edge records corresponding to the new breakpoint of the Voronoi diagram. Set the pointers between them appropriately. Check the new triples of consecutive arcs that arise because of the disappearance of α. Insert the corresponding circle event into L only if the circle intersects the sweep line and the circle event is not present yet in L.

Table 2.3 : HandleSiteEvent function of Fortune's algorithm

HandleSiteEvent(x_i)
Search in S for the arc α vertically above x_i and delete all circle events involving α from L.
Replace the leaf of S representing α with a subtree having three leaves. The middle leave stores the new site x_i and the other two leaves store the site x_j that was originally stored with α. Store the tuples $\langle x_j , x_i \rangle$ and $\langle x_i , x_j \rangle$ representing the new breakpoints at the two new internal nodes. Perform rebalancing operations on T if necessary.
Create new records in the Voronoi diagram structure for the two half-edges separating $V(x_i)$ and $V(x_j)$ which will be traced out by the two new breakpoints.
Check the triples of consecutive arcs involving one of three new arcs. Insert corresponding circle event only if the circle intersects the sweep line and the circle event is not present yet in L.

2.4. Collision and Obstacle Avoidance

Although it is claimed that collision avoidance is guaranteed for multi robot exploration problem by only using the Voronoi regions [16], it is only true for point mass robot model. In order to use a more realistic robot models, such as given in Figure 2.2, a more advanced collision avoidance method is necessary. This method will be explained in detail in Coverage Methods section, similar to the method in the work [18].

Obstacle avoidance can be performed similar to collision avoidance. The optimization objective function is modified in order to create repulsive fields around obstacle regions similar to potential fields method. The repulsive fields behave opposite to fields having non-zero information gains. Also, like in collision avoidance, in the optimization loop, a forward simulation can be done and the resulting point can be checked against obstacles. If the point is on obstacle, the cost of this control input will be saturated to the lowest value. Hence, the optimization will not select the control inputs that make the agent to collide with obstacles.

2.5. Information Gains

In the Coverage formulation, as given in Equation (**2.19**, the function $\Phi(q)$ is defined as information gain function. It weights the areas that are expected to be denser with respect to information.

$$H_i(p_i, \theta_i) = \int_{\partial s_i} f(p_i, q, \theta_i)\Phi(q)dq \qquad \textbf{(2.19)}$$

In the Discoverage and Anisotropic Coverage approaches, the information gains are taken as expected static functions. However, a new method is presented that allows this function to be dynamic and updated at each optimization step.

2.6. Numerical Optimization Algorithm

Since the Coverage formulation includes functions with nonlinear terms such as numerical integration over frontier regions, it is hard to derive the solution by analytically. So, an appropriate numerical optimization method is used in order to compute the solution of the optimization at each optimization loop.

Two approaches are tested in simulation; gradient descent and global optimal search methods.

2.6.1. Gradient Descent

Gradient descent algorithm is a first order numerical optimization algorithm and finds local minimum of a function. It finds the minimum point by changing step size according to the negative value of the gradient of the function at a point.

$$x_{k+1} = x_k - \beta\nabla f(x_k) \qquad \textbf{(2.20)}$$

The algorithm ends with the error condition. Let ε be a small number close to zero.

$$|\nabla f(x_k)| < \varepsilon \qquad \textbf{(2.21)}$$

The derivative of function f in all directions becomes nearly zero, so the function does not change at this local minimum point.

2.6.2. Global Optimum Search

The method used for global optimum search is based on exhaustive search [57]. The objective function is evaluated on the search space and the global maximum or minimum point is found. This method gives the best results and avoids the local minimum points. Also, it is fast because the search space is small.

Another method which can be categorized into global search is genetic algorithm. Although it can avoid local extremum points, it is slow compared to the exhaustive search method.

3. COVERAGE METHODS

In this chapter, two different coverage methods are explained. One of them is the Discoverage approach which uses a control law based on frontiers of the regions and Voronoi partitioning for multiple agents. On the other hand, the other method is based on optimization without partitioning the map and without using frontier integrals. However, the latter method uses an anisotropic sensor model.

3.1. A Frontier and Voronoi Based Coverage Approach: Discoverage

Let S_i is the frontier curve obtained from the formulation of the borders between explored and unexplored space over the Voronoi region of vehicle i. The function f can be defined as the performance index of exploration process. In the previous work DisCoverage, the used solution is to find optimal heading angles of vehicle moving with a constant linear velocity. The same approach is used for navigation and exploration.

Letting p_i the position vector and θ_i is the orientation of the agent i, the objective function to be optimized for i^{th} agent becomes a curve integral expression over ∂S_i:

$$H_i(p_i, \theta_i) = \int_{\partial S_i} f(p_i, q, \theta_i)\Phi(q)dq \tag{3.1}$$

We can define the angle δ_i as follows:

$$\delta_i = atan2(p_{i_y} - q_y, p_{i_x} - q_x) \tag{3.2}$$

The function f consists of two parts, an angular and a distance component:

$$f(p_i, q, \theta_i) = \exp\left(-\frac{\delta(p_i, q, \theta_i)^2}{2\alpha^2}\right)\exp\left(-\frac{\|q - p_i\|_2^2}{2\rho^2}\right) \tag{3.3}$$

Individual agents will run the optimization in a distributed manner. Let $P = \{p_1, \dots p_N\}$ denoting the position vectors of the agents and $\Theta = \{\theta_1, \dots \theta_N\}$ representing the orientation angles. Maximizing H_i in parallel means also maximizing the total objective function H given below:

$$H(P, \Theta) = \sum_{i=1}^{N} H_i(p_i, \theta_i) \quad \textbf{(3.4)}$$

$$\{\theta_i^*, H_i^*\} = \underset{\theta_i}{\operatorname{argmax}}\, H_i(p_i, \theta_i) \quad \textbf{(3.5)}$$

The optimization problem is converted into finding optimal orientation angles θ_i^* in parallel which maximize the objective function in order to cover the area.

3.2. Coverage with Variable Information Gains

In previous works, the expected information gain part of the objective function is taken as exponential functions with constant coefficients. This function is also known as density function. In our approach, the information gain function consists of two parts; a static (expected) and a dynamic part, as given below.

$$\Phi(q) = \Phi_{static}(q) + \Phi_{dynamic}(q) \quad \textbf{(3.6)}$$

And of course, an update rule for the dynamic part is necessary. This rule depends on the location of the explored information in the area. In our case studies, this information is taken as explored mines. The location of the explored mines weights the area within an exploration radius which determines standard deviation of the Gaussian function. The amplitude of the Gaussian is selected by using the density of information in the area.

A simple algorithm is proposed in order to update the information gain function. Algorithm steps are given in Figure 3.1.

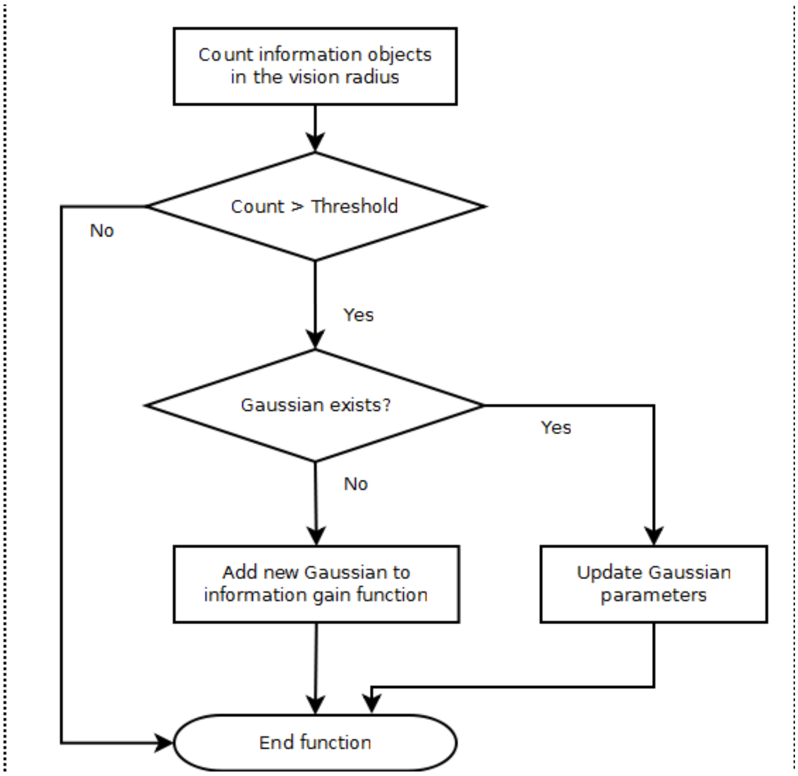

Figure 3.1 : VIG Algorithm. This subroutine is called at every step of the optimization.

The update algorithm enables the robots to navigate around the dense information areas first. In brief, the priorities of the areas are dynamically changed. The effect of the method will be given in a case study.

3.3. Coverage with Collision Avoidance

Most of the works about multi robot coverage problem consider the agents as a point mass or particle. This assumption can be true in most cases, such as the robot has negligible physical dimensions comparing to the area to be covered. Additionally, some of the works (Discoverage) claim that Voronoi partitioning can automatically enable collision avoidance among the robots. However, the robots can collide even if they are in separate Voronoi regions and calculate their own optimal paths in these regions. A similar work about collision avoidance for coverage problem [18] is done in the literature. However, the suggested method in this thesis is based on the work Discoverage [16].

In order to get realistic results and for a good implementation, an improved collision avoidance mechanism is necessary. This is accomplished by performing a forward simulation in each optimization step. p_i^* is the position vector of the agent i obtained from forward simulation. If a collision occurs in the forward simulation, the cost is set to the lowest value. Also, the collision is detected by calculating distance between each robot and checking it against a distance threshold.

The analytical expression of this method can be given by subtracting a penalty term from the cost function. Here, c is a constant with a sufficiently large value.

$$\begin{aligned} H_i(p_i, \theta_i^*) = & \int_{\partial S_i} f(p_i, q, \theta_i)\Phi(q)dq \\ & - \sum_{\mathrm{j \neq i}} \mathrm{c}\left(\mathrm{R_{collision}} - \|p_i^* - p_j^*\|\right) \end{aligned} \quad \textbf{(3.7)}$$

If the agents i and j are closer to each other than a collision distance $R_{collision}$, the penalty term becomes positive values. So, the value cost function will be very low. Since the optimization aims to maximize H_i, this value will not be the optimal solution.

3.4. Coverage with Obstacle Avoidance

An obstacle avoidance method can be applied to the optimization problem. The coverage objective function is modified, as in the collision avoidance case. This suggested method is similar to the collision avoidance works. However, it is a new approach.

Similarly, the analytical expression of the obstacle avoidance method can be given by subtracting a penalty term from the cost function. An obstacle j is defined as a circular region having a radius R_j and center point o_j. Also, c is a constant with a sufficiently large value and N_o is the number of obstacles.

$$H_i(p_i, \theta_i^*) = \int_{\partial S_i} f(p_i, q, \theta_i)\Phi(q)dq - \sum_{\mathrm{j=1}}^{\mathrm{N_o}} \mathrm{c}\left(\mathrm{R_j} - \|p_i^* - o_j\|\right) \quad \textbf{(3.8)}$$

Each agent computes its objective function H_i and calculates optimal control variables. In case of reaching an obstacle, the penalty term will cause the value of the objective function to decrease rapidly. Hence, the agent will avoid the obstacle region.

4. SENSOR CHARACTERISTICS ON MULTI ROBOT COVERAGE PROBLEM

Sensor types and characteristics are important issues in multi robot coverage problems. Localization with anisotropic sensors by using a special transformation of measurements [58] and a hybrid localization method in anisotropic sensor networks by analyzing information of nodes [59] are some of the works in the literature about localization with anisotropic sensors. About the coverage problem, there are also some works related to anisotropic type sensors such as, coverage control for mobile sensor networks using anisotropic sensor models and Lloyd algorithm with collision avoidance [60], coverage control of robotic agents with limited range communication and anisotropic sensing [61] and gradient based coverage control of multiple robots with limited anisotropic sensors [53].

In this section, the effect of sensor characteristics on coverage problem will be investigated. The sensors are categorized according to their anisotropy and the coverage problem is discussed with isotropic and anisotropic type sensors.

4.1. Coverage with Anisotropic Sensors

In the proposed method in the literature [53], a coverage control method is proposed by using multiple agents with anisotropic sensors. Also, it is assumed that *N* independent robots or agents are moving in the region Q which is defined as a polyhedron in R^2. The location and orientation of the robots are $(x,y)=\{(x_1,y_1),\ldots,(x_N,y_N)\}$ and $\theta=\{\theta_1,\ldots,\theta_N\}$, respectively. The kinematic model used in the method is as given below:

$$x_i(k+1) = x_i(k) + u_{x,i}(k) \quad \textbf{(4.1)}$$

$$y_i(k+1) = y_i(k) + u_{y,i}(k) \quad \textbf{(4.2)}$$

$$\theta_i(k+1) = \theta_i(k) + v_i(k) \quad \textbf{(4.3)}$$

In this notation, k is the sample number or iteration count, $u_{x,i}(k)$, $u_{y,i}(k)$, $v_i(k)$ are the control inputs for position and orientation. Let q be a point in Q. If an event occurs at q, the sensor at location (x_i,y_i) will detect the signal emitted from point q. According to the anisotropic sensor model, the received signal quality changes with respect to the distance from the sensor and orientation of the target. This change is modeled by using a performance function $p_i(q)$. If the value of this function is low, this means the point q is sensed by the sensor i poorly.

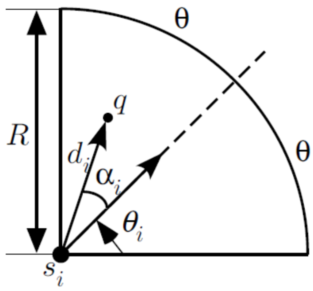

Figure 4.1 : The geometric explanation of the variables

Let R be the maximum sense range and Θ be the maximum sensing direction of an anisotropic sensor as shown in Figure 4.1.

$$d_i = \|q - s_i\| \quad \textbf{(4.4)}$$

$$\alpha_i = \measuredangle(q - s_i) - \theta_i \quad \textbf{(4.5)}$$

$$\Theta \epsilon (0, \frac{\pi}{2}] \quad \textbf{(4.6)}$$

A sensor model created in the paper is as below where Q_i is the region of sensing of sensor i:

$$p_i(q) = \begin{cases} \dfrac{(d_i - R)^2(\alpha_i - \Theta)^2}{R^2\Theta^2} & , \quad if\ q \epsilon Q_i \\ 0 & , otherwise \end{cases} \quad \textbf{(4.7)}$$

$$P(q, s, \theta) = 1 - \prod_{i=1}^{N}(1 - p_i(q)) \quad \textbf{(4.8)}$$

$$F(s, \theta) = \int_Q \phi(q) P(q, s, \theta) dq \quad \textbf{(4.9)}$$

The gradient flows approach is used to design the control laws $u_i(k)$ and $v_i(k)$. In order to achieve optimal coverage, objective function F(s,θ) should be maximized. It is necessary to find the optimal control laws $u_{xi}(k)$, $u_{yi}(k)$ and $v_i(k)$ by taking the partial derivatives with respect to x, y and θ.

$$\frac{\partial F}{\partial x_i} = \int_Q \emptyset(q) \frac{\partial P(q,s,\theta)}{\partial x_i} dq \tag{4.10}$$

$$\frac{\partial F}{\partial y_i} = \int_Q \emptyset(q) \frac{\partial P(q,s,\theta)}{\partial y_i} dq \tag{4.11}$$

$$\frac{\partial F}{\partial \theta_i} = \int_Q \emptyset(q) \frac{\partial P(q,s,\theta)}{\partial \theta_i} dq \tag{4.12}$$

By using the equation of $P(q,s,\theta)$, the partial derivatives can be rewritten as follows:

$$\frac{\partial F}{\partial x_i} = \int_Q \emptyset(q) \prod_{k=1,k\neq i}^{N} \big(1 - p_k(q)\big) \left(\frac{\partial p_i(q)}{\partial d_i} \frac{\partial d_i}{\partial x_i} + \frac{\partial p_i(q)}{\partial \alpha_i} \frac{\partial \alpha_i}{\partial x_i} \right) dq \tag{4.13}$$

$$\frac{\partial F}{\partial y_i} = \int_Q \emptyset(q) \prod_{k=1,k\neq i}^{N} \big(1 - p_k(q)\big) \left(\frac{\partial p_i(q)}{\partial d_i} \frac{\partial d_i}{\partial y_i} + \frac{\partial p_i(q)}{\partial \alpha_i} \frac{\partial \alpha_i}{\partial y_i} \right) dq \tag{4.14}$$

$$\frac{\partial F}{\partial \theta_i} = \int_Q \emptyset(q) \prod_{k=1,k\neq i}^{N} (1 - p_k(q)) \left(\frac{\partial p_i(q)}{\partial \alpha_i} \frac{\partial \alpha_i}{\partial \theta_i} \right) dq \tag{4.15}$$

The control laws can be given as below:

$$u_{x,i} = \beta_{x,k} \frac{\partial F}{\partial x_i(k)} \tag{4.16}$$

$$u_{y,i} = \beta_{y,k} \frac{\partial F}{\partial y_i(k)} \tag{4.17}$$

$$v_{x,i} = \gamma_k \frac{\partial F}{\partial \theta_i(k)} \tag{4.18}$$

The step sizes $\beta_{x,k}$, $\beta_{y,k}$ and $\gamma_k$is selected in order to guarantee the convergence. Also, it is necessary to give the following assumption:

If $q \notin Q_i$then, $p_i(q) = 0$, $\frac{\partial p_i(q)}{\partial d_i(q)} = 0$, $\frac{\partial p_i(q)}{\partial \alpha_i(q)} = 0$.

This assumption says that the sensor *i* can only sense the point inside its region of sensing Q_i.

$$\frac{\partial F}{\partial x_i} = \int_{Q_i} \emptyset(q) \prod_{k \in N_i}^{N} \big(1 - p_k(q)\big) \left(\frac{\partial p_i(q)}{\partial d_i} \frac{\partial d_i}{\partial x_i} + \frac{\partial p_i(q)}{\partial \alpha_i} \frac{\partial \alpha_i}{\partial x_i} \right) dq \quad \textbf{(4.19)}$$

$$\frac{\partial F}{\partial y_i} = \int_{Q_i} \emptyset(q) \prod_{k \in N_i}^{N} \big(1 - p_k(q)\big) \left(\frac{\partial p_i(q)}{\partial d_i} \frac{\partial d_i}{\partial y_i} + \frac{\partial p_i(q)}{\partial \alpha_i} \frac{\partial \alpha_i}{\partial y_i} \right) dq \quad \textbf{(4.20)}$$

$$\frac{\partial F}{\partial \theta_i} = \int_{Q_i} \emptyset(q) \prod_{k \in N_i}^{N} (1 - p_k(q)) \left(\frac{\partial p_i(q)}{\partial \alpha_i} \frac{\partial \alpha_i}{\partial \theta_i} \right) dq \quad \textbf{(4.21)}$$

The N_i is the neighbor set of the sensor *i* defined by:

$$N_i = \{r \colon \|s_i - s_r\| < 2R \, , r = 1, \dots, N, \;\; r \neq i\} \quad \textbf{(4.22)}$$

The final form of the derivatives is as follows:

$$\begin{aligned} \frac{\partial F}{\partial x_i} = \int_{Q_i} \emptyset(q) \prod_{k \in N_i}^{N} \big(1 - p_k(q)\big) \bigg(& \frac{\partial p_i(q)}{\partial d_i} \frac{x_i - q_x}{\|s_i - q\|} \\ & + \frac{\partial p_i(q)}{\partial \alpha_i} \frac{a + b}{\sqrt{m^2 - l^2}} \bigg) dq \end{aligned} \quad \textbf{(4.23)}$$

$$\begin{aligned} \frac{\partial F}{\partial y_i} = \int_{Q_i} \emptyset(q) \prod_{k \in N_i}^{N} \big(1 - p_k(q)\big) \bigg(& \frac{\partial p_i(q)}{\partial d_i} \frac{y_i - q_y}{\|s_i - q\|} \\ & + \frac{\partial p_i(q)}{\partial \alpha_i} \frac{a + b}{\sqrt{m^2 - l^2}} \bigg) dq \end{aligned} \quad \textbf{(4.24)}$$

$$\frac{\partial F}{\partial \theta_i} = \int_{Q_i} \emptyset(q) \prod_{k \in N_i}^{N} (1 - p_k(q)) \left(\frac{\partial p_i(q)}{\partial \alpha_i} \frac{-z}{\sqrt{m^2 - l^2}} \right) dq \quad \textbf{(4.25)}$$

The constants are:

$$a = (cos\theta_i, sin\theta_i) \tag{4.26}$$

$$b = \frac{(q - s_i)(cos\theta_i, sin\theta_i)(s_i - q)}{y^2} \tag{4.27}$$

$$l = (q - s_i)(cos\theta_i, sin\theta_i) \tag{4.28}$$

$$m = \|s_i - q\| \tag{4.29}$$

$$z = (q - s_i)(-sin\theta_i, cos\theta_i) \tag{4.30}$$

In brief, the robots can calculate their control signals by using the sensor data from their neighbors in the sensing regions.

4.2. Coverage with Isotropic Sensors

The same coverage method can be applied to isotropic sensor case by simply changing the sensor model.

An isotropic sensor model can be selected as p_i given below. Q_i is the region of sensing of sensor i, q is a point in the space.

$$p_i(q) = \begin{cases} 1 & , \quad if\ q \epsilon Q_i \\ 0 & , otherwise \end{cases} \tag{4.31}$$

$$P(q, s, \theta) = 1 - \prod_{i=1}^{N} (1 - p_i(q)) \tag{4.32}$$

$$F(s, \theta) = \int_Q \phi(q) P(q, s, \theta) dq \tag{4.33}$$

The same coverage formulation can be used in order to produce a control input for the agent i. Similarly, the method is based on gradients of the performance function.

5. SIMULATION STUDIES

5.1. Simulation Environment

Simulations of the proposed approaches are done in MATLAB environment. The map and all information regarding to the coverage algorithm and robot kinematics are stored in arrays and structures. The detailed explanation of the algorithm steps can be found in Figure 5.4. Also, the axes of all simulation plots are compliant to the explanation frame in Figure 2.1.

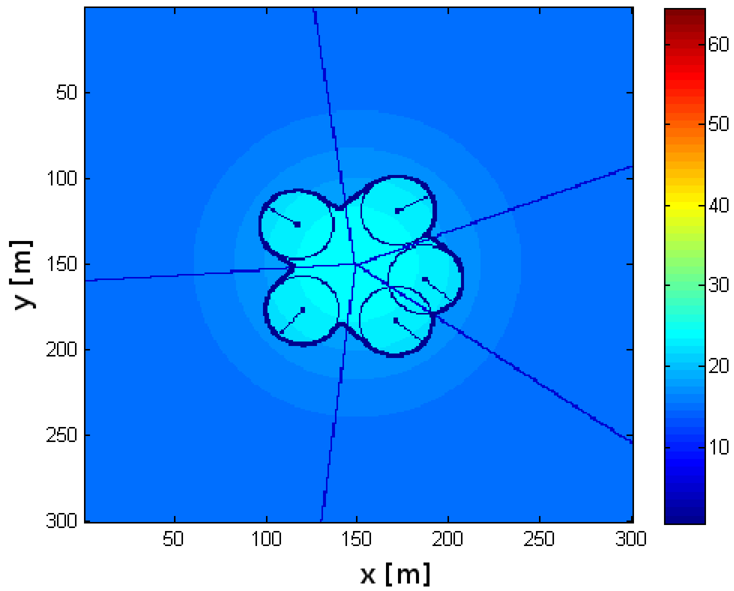

Figure 5.1 : Sample simulation frame

In the first step, the explored regions are encoded in the map image matrix $I(x,y)$. An explored region is defined as the area seen by an agent within the vision radius R_{vision}. Then, the frontiers of explored regions are found by filtering the map matrix with the Laplacian filter. The map image $I(x,y)$ consisting of explored and unexplored regions are convolved with the Laplacian kernel given below.

$$\nabla^2 = D_{xy}^2 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \textbf{(5.1)}$$

$$F(x,y) = I(x,y) * D_{xy}^2 \quad \textbf{(5.2)}$$

The resulting image *F(x,y)* contains frontier curves, as shown in Figure 5.2.

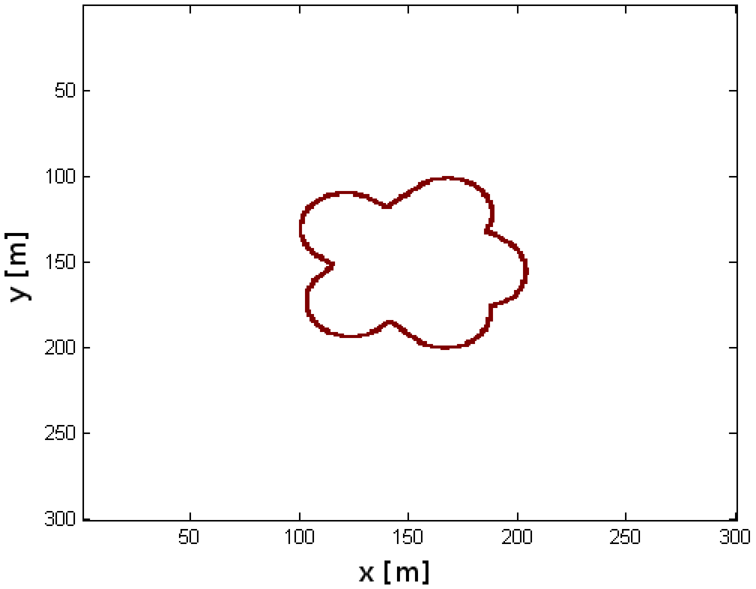

Figure 5.2 : Frontier curve image *F(x,y)* after applying Laplacian filter

Moreover, the Voronoi regions of individual agents are found by using the modified version built-in MATLAB function which uses Delaunay triangulation to find Voronoi lines. In order to draw the lines, Bresenham's line algorithm is used.

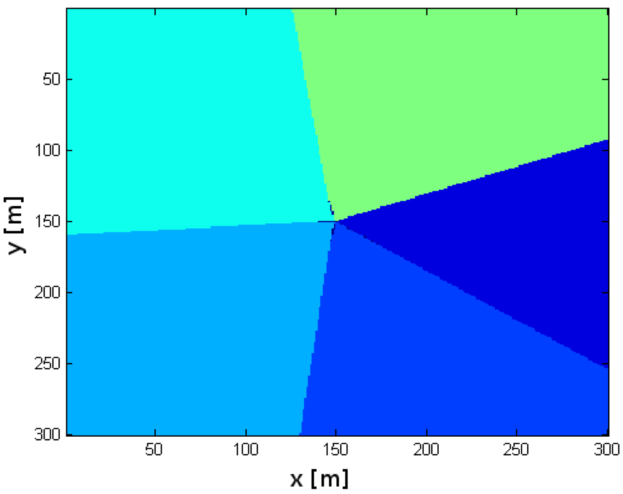

Figure 5.3 : Voronoi regions obtained with Flood Fill algorithm

In order to associate the Voronoi regions with robotic agents, the flood fill algorithm is used with connected component labeling. First, the Voronoi regions are filled as shown in Figure 5.3 and the filled regions are labeled by using connected component labeling algorithm. Then, the labels obtained are associated with each robot. So, if a point coordinate is known in the map, the corresponding agent ID can be obtained. This provides taking the integral over the Voronoi region for a specific robot.

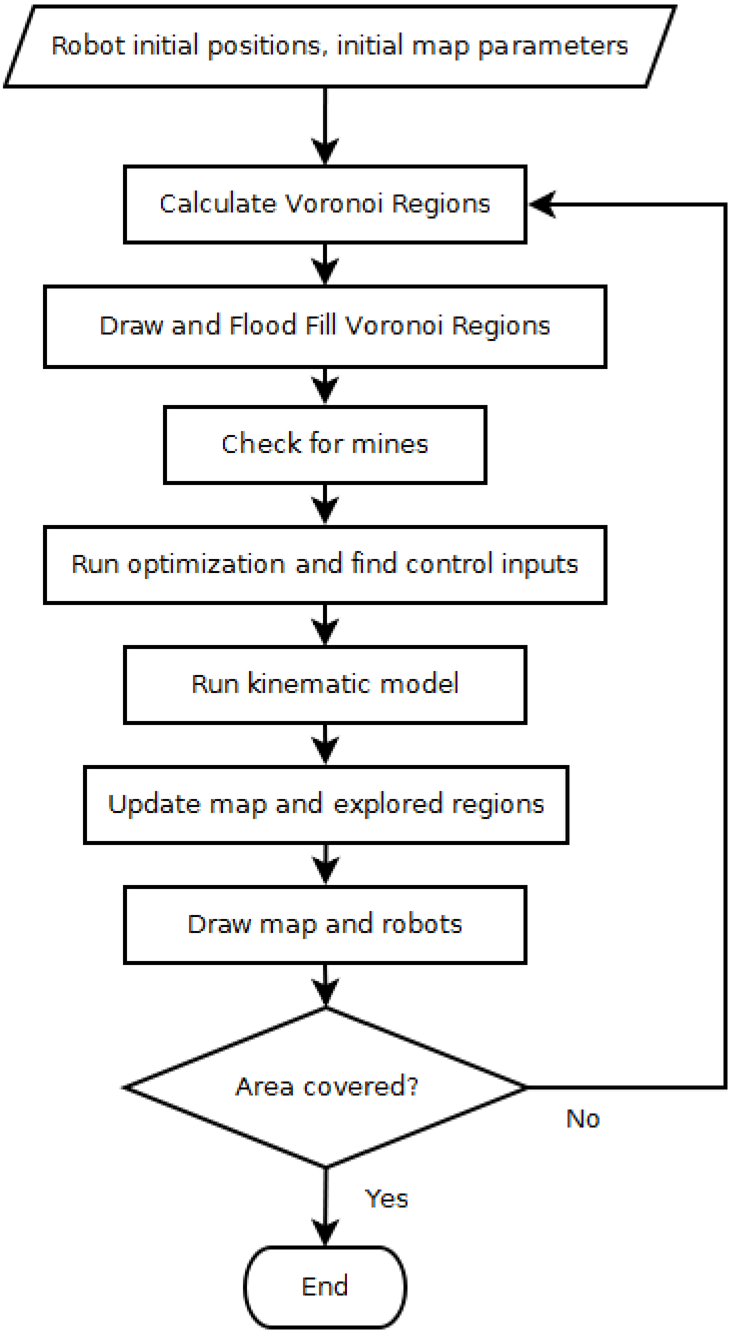

Figure 5.4 : Simulation algorithm flowchart for mine clearance

5.2. Discoverage with Single Agent

In the simulation, 100 of mines are distributed randomly over a 300x300 map. The initial position of the robot is (150, 150) and its initial orientation is zero. α and ρ are chosen as 0.5 and 10, respectively.

In Figure 5.5, Figure 5.6 and Figure 5.7 the output of the algorithm is given. The blue points represent mines to be explored and the robot has a vision radius of 20.

The information gain function is selected as below. In this simulation, the variable information gain approach is used with a single agent.

$$\Phi(q) = \Phi_{static}(q) + \Phi_{dynamic}(q) \tag{5.3}$$

$$\Phi_{static}(q) = \exp\left(-\frac{\|q-\mu\|^2}{2\sigma^2}\right) \tag{5.4}$$

$$\Phi_{dynamic}(q) = \sum_{\mathrm{i}} \exp\left(-\frac{\|q-\mu_i\|^2}{2\sigma_{\mathrm{i}}^2}\right) \tag{5.5}$$

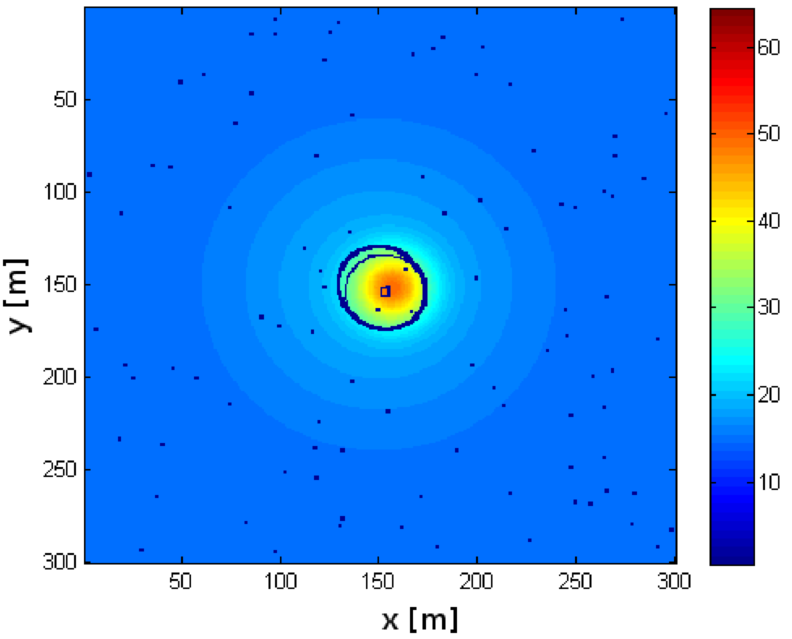

Figure 5.5 : A simulation frame with single agent

In Figure 5.5, starting from initial state, the vehicle explores a dense mine region and updates the information gain function which affects the overall cost function. So, it is possible to say that after finding a dense information region, the robot gives the priority of exploring to this area.

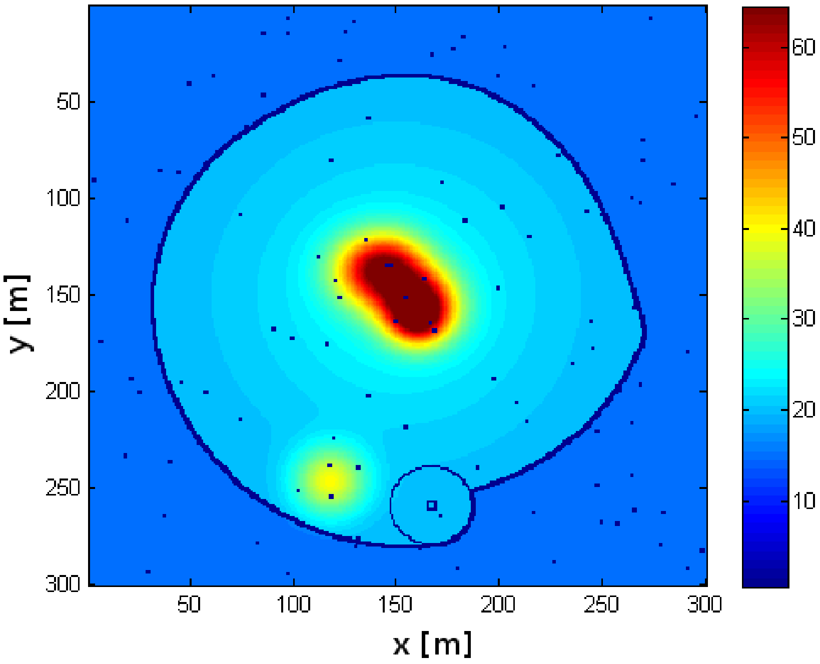

Figure 5.6 : Covered regions by the agent

In Figure 5.6, the explored mines create new Gaussian functions on the information gain function. This enables the robot to navigate around these areas.

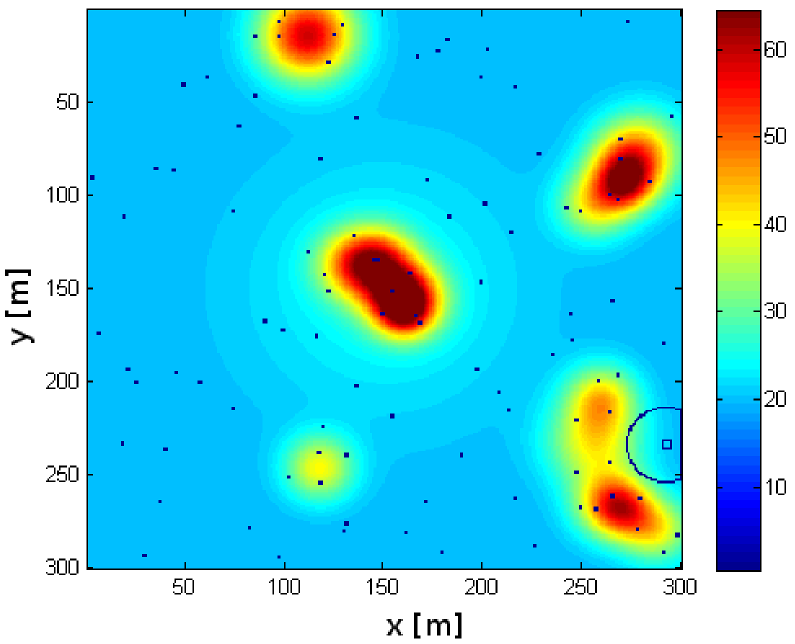

Figure 5.7 : The whole map is covered by the agent

At last, in Figure 5.7, the exploration is completed after 910 iterations. The mine density map can be seen from the figure where colors denote dense mine regions.

5.3. Discoverage with Multi Agents

In this section, the simulation results of the coverage method with multiple agents will be shown. In Figure 5.8, 5 robots are at their initial positions with zero initial orientation angles. The width and height of the map are 300 meters. Voronoi lines are shown and they separate the regions of robots. The coverage algorithm runs over these Voronoi regions.

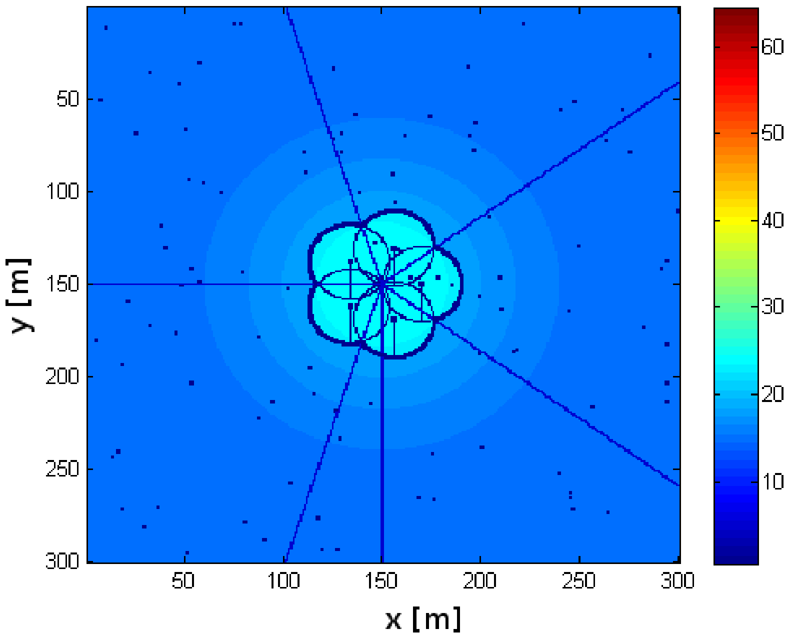

Figure 5.8 : One of the initial frames of the multi agent coverage simulation

Robots are seeking the 150 mines distributed randomly over the map and their vision radiuses are 20 meters. Also, the coefficients α and ρ are chosen as 0.5 and 10, respectively. The initial information gain function is taken as follows:

$$\Phi_{static}(q) = \exp\left(-\frac{\|q-\mu\|^2}{2\sigma^2}\right) \quad \textbf{(5.6)}$$

Here, μ determines the center coordinates of the Gaussian curve and σ is the standard deviation. The width or radius of the Gaussian can be changed by selecting the parameter σ. The values for μ and σ are selected for the simulation as (150, 150) and 50, respectively.

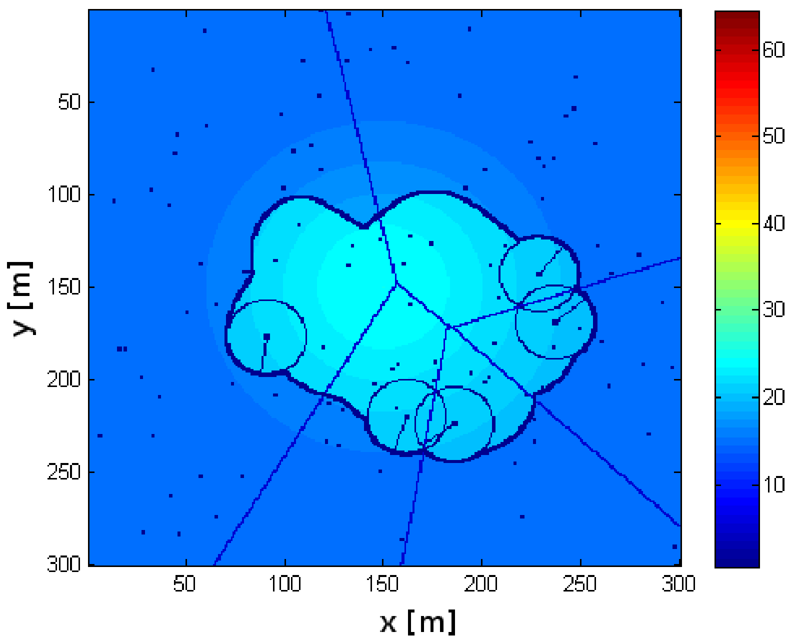

Figure 5.9 : 20th iteration of the coverage simulation

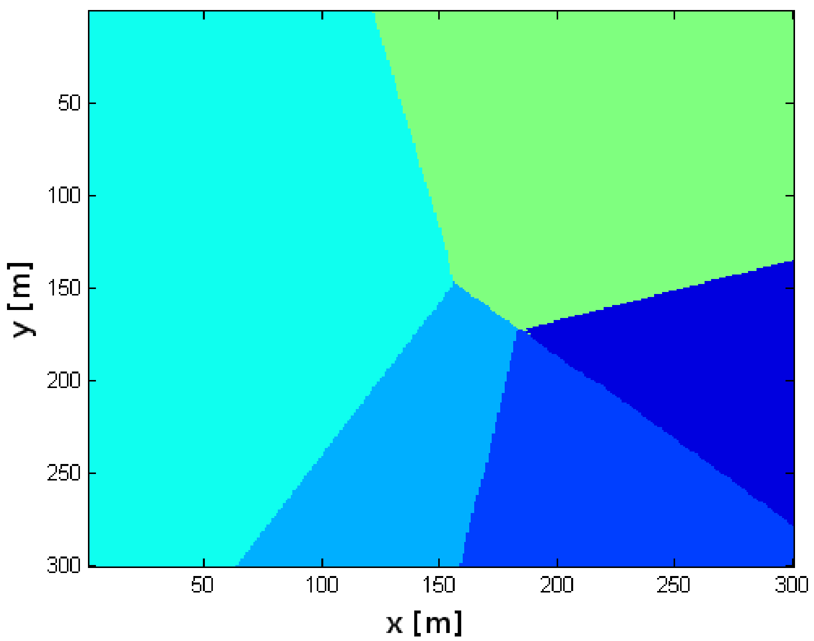

Figure 5.10 : Voronoi regions of the agents at 20th iteration

In Figure 5.9 and Figure 5.11, 20th and 80th iterations of the simulation are shown, respectively. Figure 5.10 represents the Voronoi regions of the robots in Figure 5.9. Since there is no additional collision avoidance, the robots are coming pretty close to each other, closer than a distance equal to their vision radiuses. However, if the robots are taken as point masses, the Voronoi regions prevent the robots from collision. A simulation with improved collision avoidance will be presented in one of the next sections. Moreover, it can be seen that the covered area increases with the iteration number. The robotic agents are discovering the region in distributed manner. Also, it should be noted that a robot having a fully discovered region will stop discovering until its Voronoi region include undiscovered regions of the map.

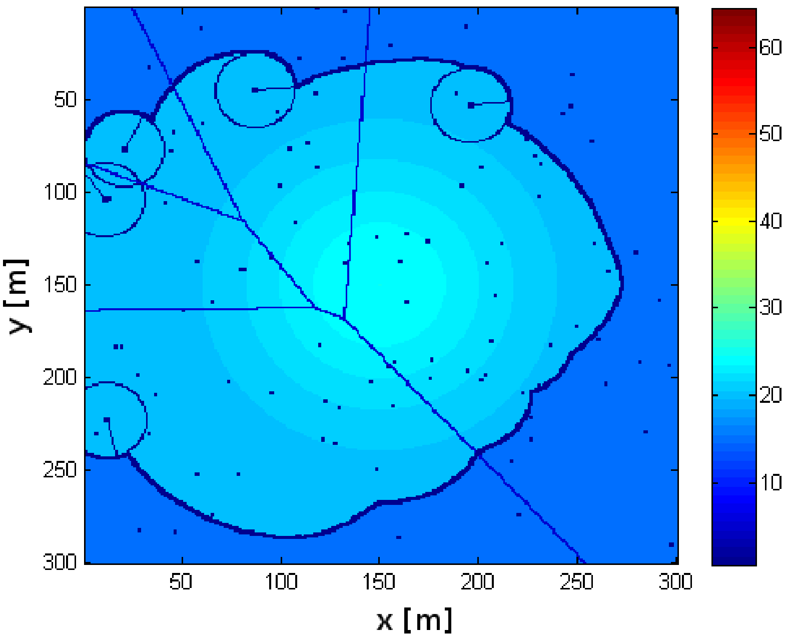

Figure 5.11 : 80th iteration of the multi agent coverage simulation

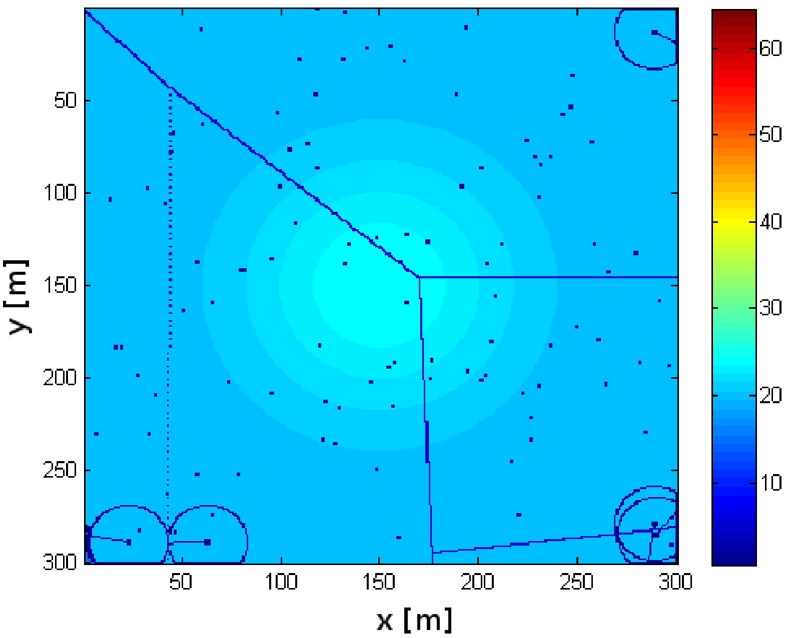

Figure 5.12 : End frame of the simulation (267th iteration)

In Figure 5.12, the last frame of the simulation at iteration 267 is shown. The area is completely covered by the agents and the agents are stopped because their Voronoi regions are discovered.

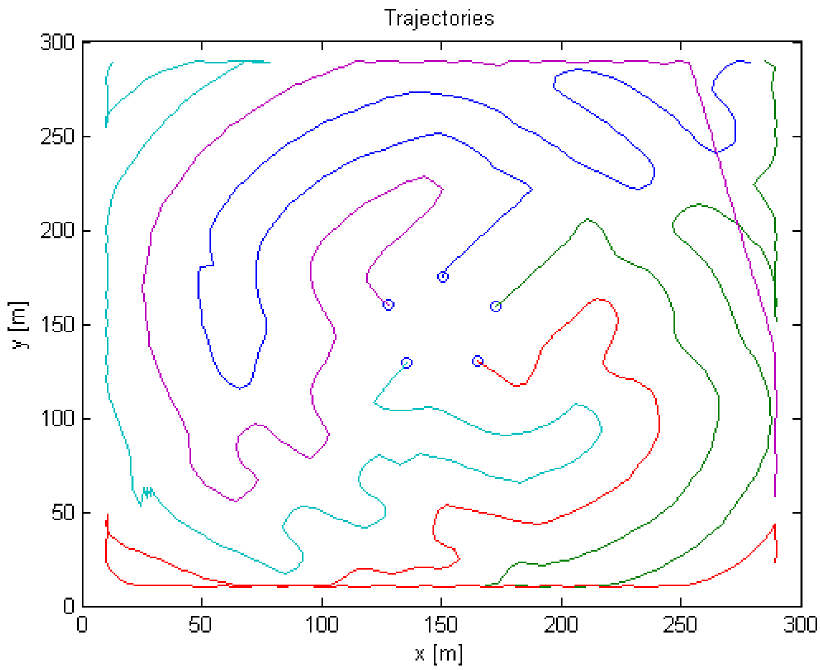

Figure 5.13 : Trajectories of the robots in the multi agent coverage simulation

In Figure 5.13, the trajectories of the robotic agents are given. The circles are showing the initial positions. The path of each robot has a different colour. It can be concluded that the robots follow distinct paths by going over the uncovered regions.

5.4. Multi Robot Discoverage with Collision Avoidance

In this section, the simulation results of the coverage method with multiple agents and collision avoidance will be shown. The width and height of the map are 300 meters. In Figure 5.14 and Figure 5.15, Voronoi lines are shown and they separate the regions of robots, as in the previous multi agent simulations.

The information gain function is taken as follows:

$$\Phi(q) = \exp\left(-\frac{\|q-\mu\|^2}{2\sigma^2}\right) \tag{5.7}$$

The values for μ and σ are selected for the simulation as (150, 150) and 50, respectively. The vision radius of robots is 20 meters. The coefficients of α and ρ are chosen as 0.5 and 10, respectively.

The distance for collision is taken as 20 meters which is the same distance with the vision radius of the robots. If the distance of two robots is less than this value, the safe region against collision is violated.

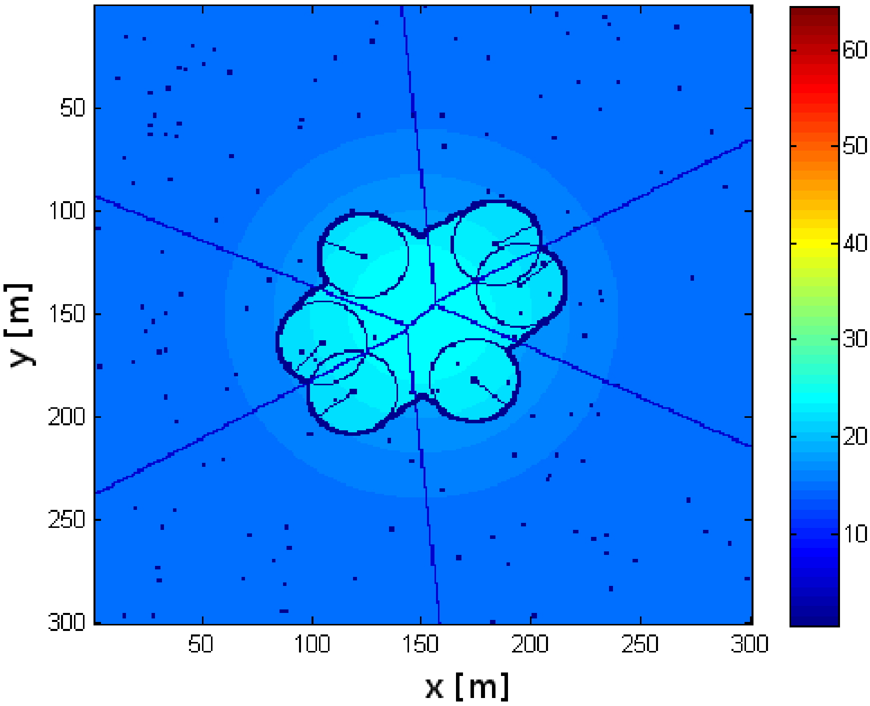

Figure 5.14 : Simulation without collision avoidance (6 robots)

In Figure 5.14, a frame of the simulation without collision avoidance is shown. As it can be seen from the figure, some of the robots are moving inside the vision region of the robots. Since, the collision distance and vision radius is taken as same values, it can be said that some of the robots are inside the regions with collision risk. The optimization algorithm should give appropriate control inputs to avoid these regions.

In Figure 5.15, the same frame of the simulation with collision avoidance is given. It can be said that the robots can avoid the risky regions defined by the collision distance which is taken as 20 meters. The optimization algorithm gives the necessary control inputs considering the collision distance.

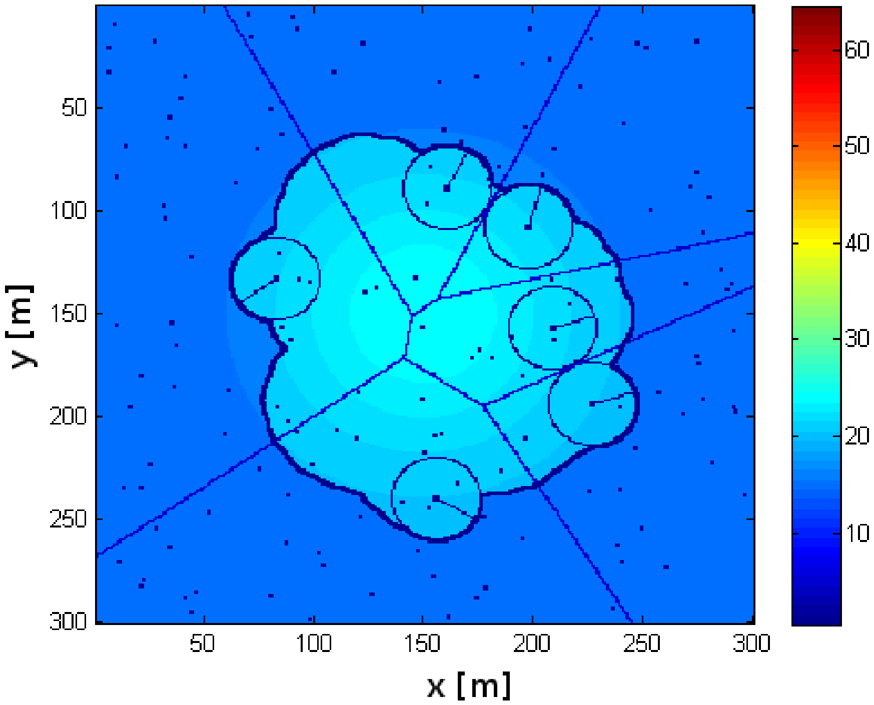

Figure 5.15 : Simulation with collision avoidance (6 robots)

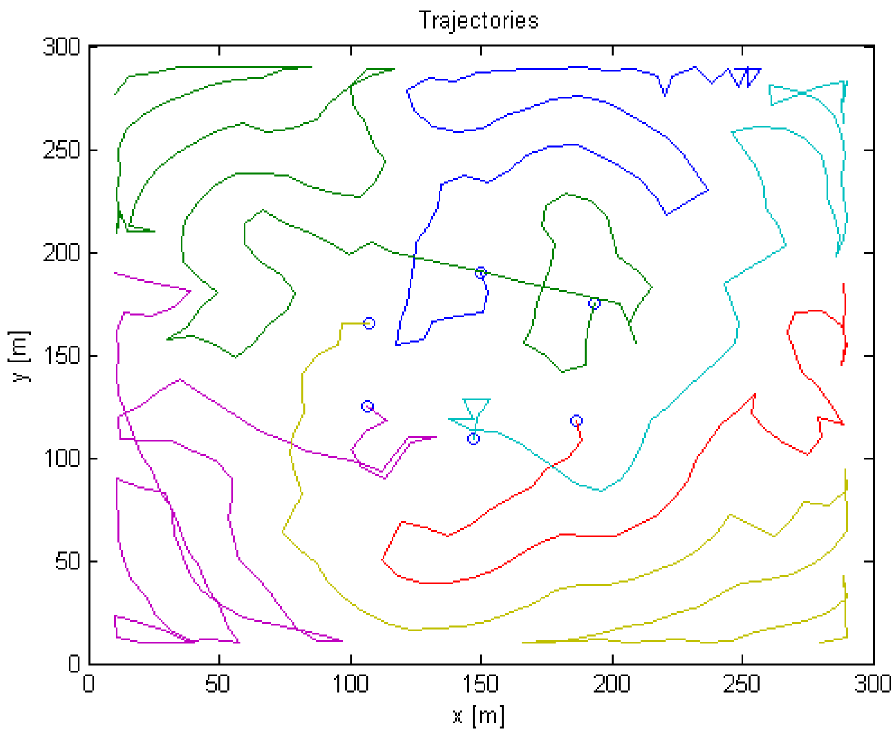

Figure 5.16 : Trajectories of the robots in simulation with collision avoidance

Figure 5.16 shows the paths that the robots follow in simulation with collision avoidance. It can be seen that the distance between the robots does not become less than the safety distance for collision avoidance. In conclusion, the collision avoidance method gives successful results.

5.5. Multi Robot Discoverage with Variable Information Gains

In this section, the simulation results of the coverage method with multiple agents and variable information gain approach will be shown. In Figure 5.17, 4 robots are at their initial positions with zero initial orientation angles. The width and height of the map are 300 meters.

The information gain function is taken as follows:

$$\Phi(q) = \Phi_{static}(q) + \Phi_{dynamic}(q) \tag{5.8}$$

$$\Phi_{static}(q) = \exp\left(-\frac{\|q-\mu\|^2}{2\sigma^2}\right) \tag{5.9}$$

$$\Phi_{dynamic}(q) = \sum_{\mathrm{i}} \exp\left(-\frac{\|q-\mu_i\|^2}{2\sigma_{\mathrm{i}}^2}\right) \tag{5.10}$$

Also, Voronoi lines are shown. In Figure 5.17 and they separate the regions of robots, as in the previous multi agent simulations.

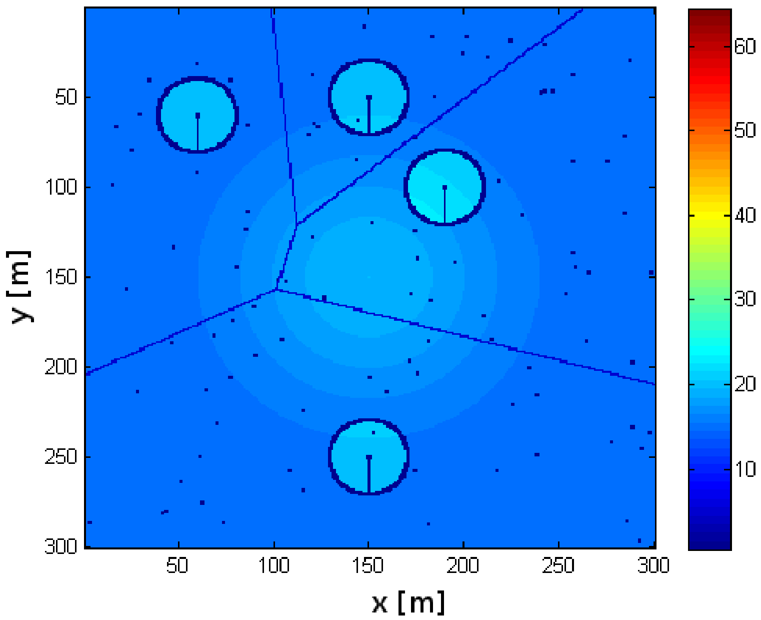

Figure 5.17 : Initial frame of the simulation with Variable Information Gains

In Figure 5.17, the first frame of the simulation is given. It can be seen that there are four mobile robots located at their initial positions and orientations. Also, the dark blue lines separating Voronoi partitions enable the robots to have their own exploration regions. The Voronoi partitions are changing dynamically according to the positions of the robots. Also, the size of map is taken as 300x300 meters.

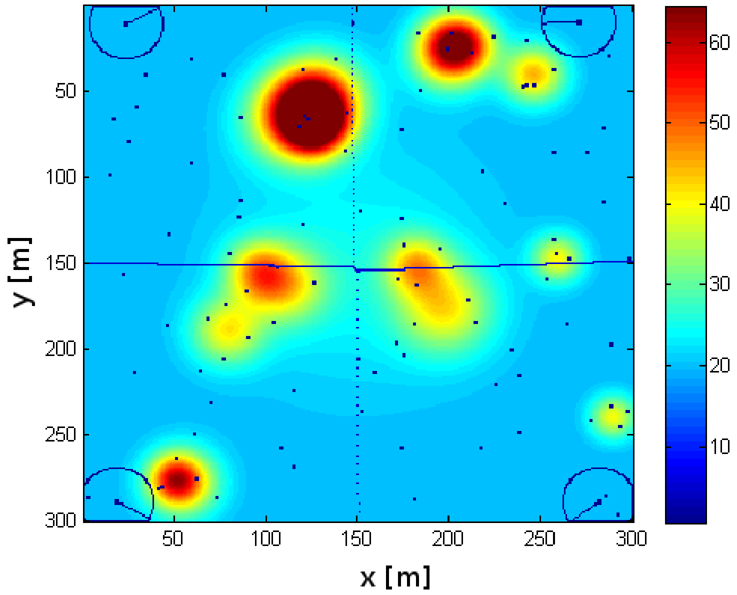

Figure 5.18 : The end frame of the simulation with information gains

In Figure 5.18, the last simulation frame of the proposed approach (VIGA) is shown. It can be seen that the regions with densely distributed mines are marked with Gaussian functions. The four robots are stopped after exploring the whole map, after 161 iterations.

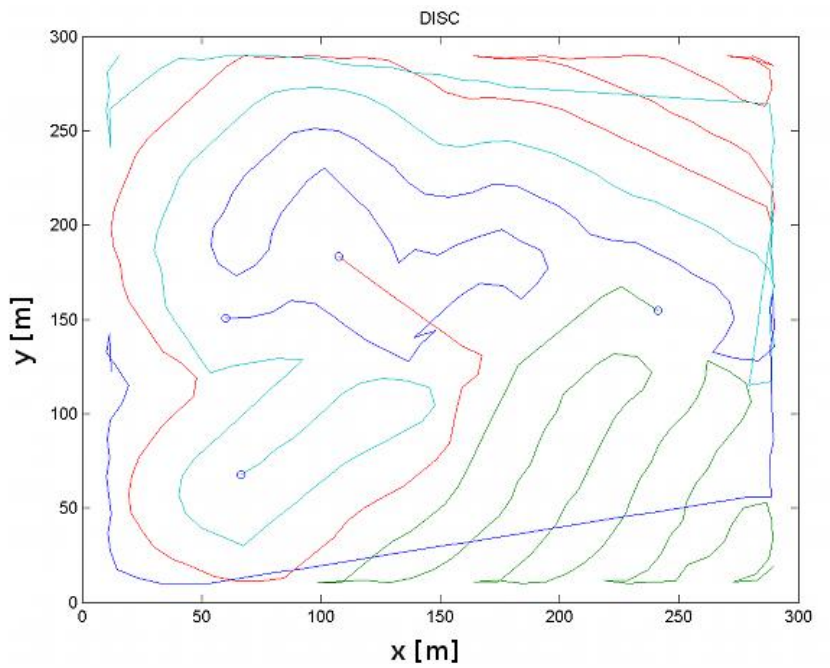

Figure 5.19 : Trajectories of the robots with original method

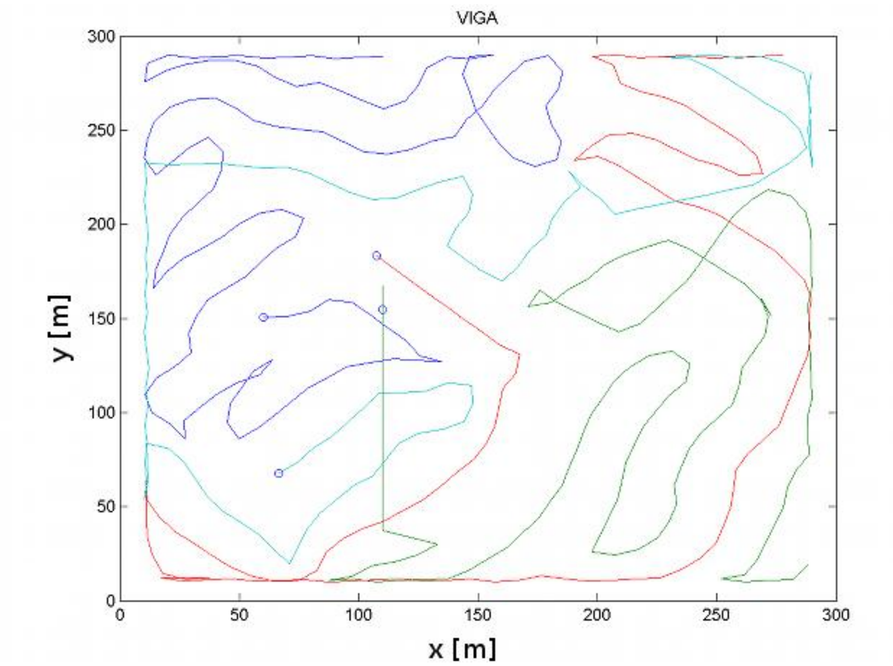

Figure 5.20 : Trajectories of the robots with the VIG approach

In the two simulations, 150 and 300 of mines are randomly distributed over the map. In Figure 5.19 and Figure 5.20, the trajectories of four agents in two simulations are given. First simulation uses the original method Discoverage (DISC) and the other one uses the proposed approach, variable information gains (VIGA).

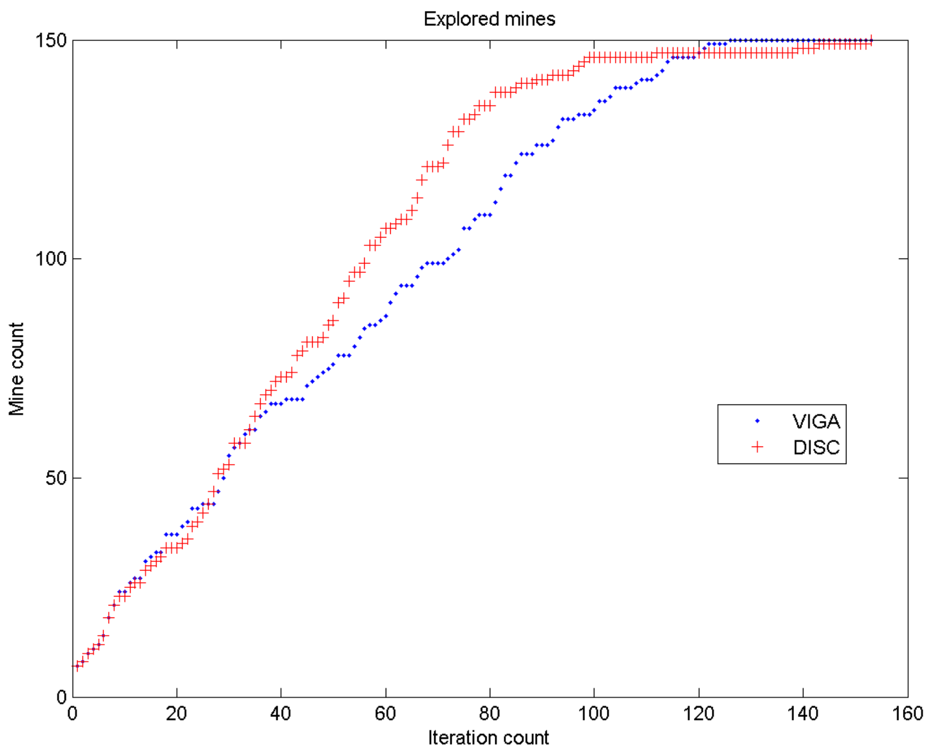

Figure 5.21 : Explored mine count versus iteration count (150 mines)

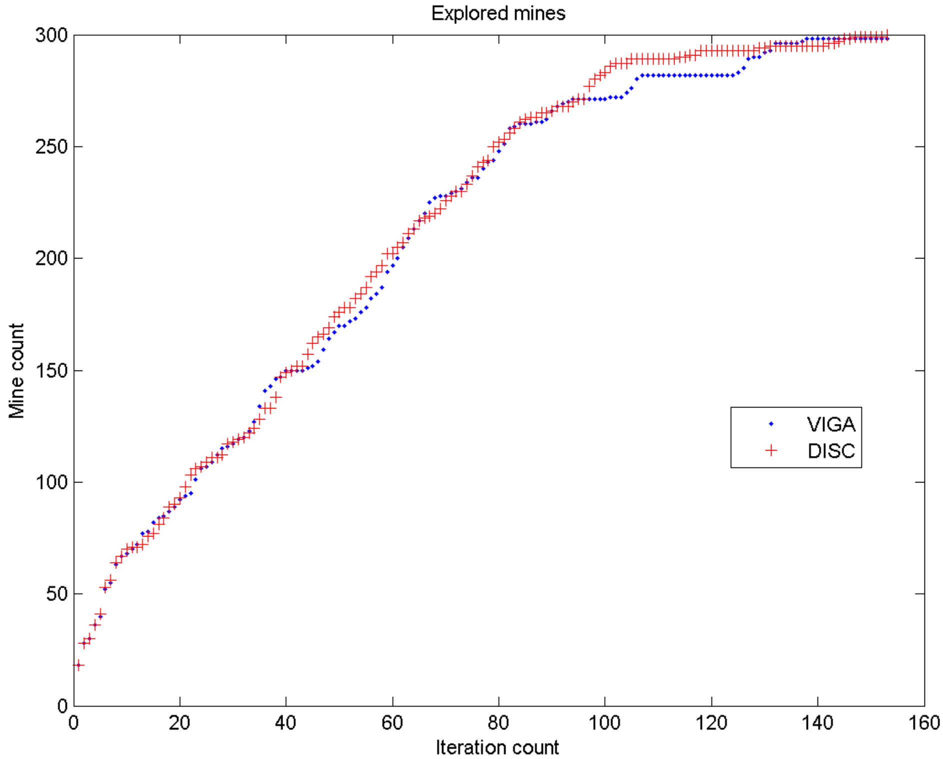

Figure 5.22 : Explored mine count with respect to iteration count (300 mines)

Figure 5.21 and Figure 5.22 show explored mine counts with respect to iteration count, with 150 and 300 initial mine count, respectively. From these results, it can be seen that there is a minor difference between two algorithms.

However, the performances of the two algorithms are different according to explored region ratio shown in Figure 5.23 and Figure 5.24. With the proposed approach, the reach time of explored region ratio and iteration count is less than the original method. So, the proposed approach gives better results. It can be concluded that the proposed variable information approach has a better result in both explored region ratio and explored mine counts compared to the original method.

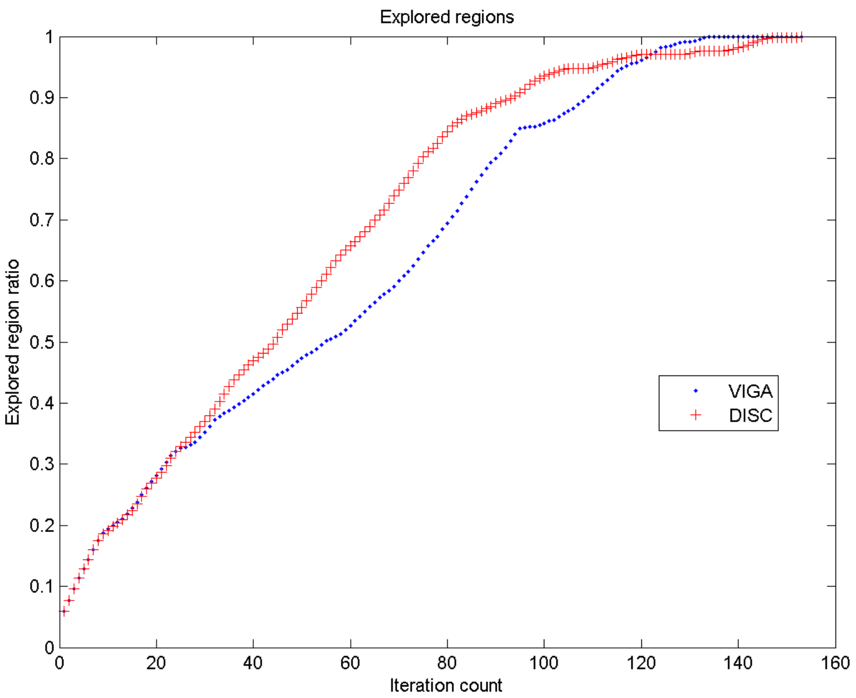

Figure 5.23 : Explored region ratio versus iteration count (150 mines)

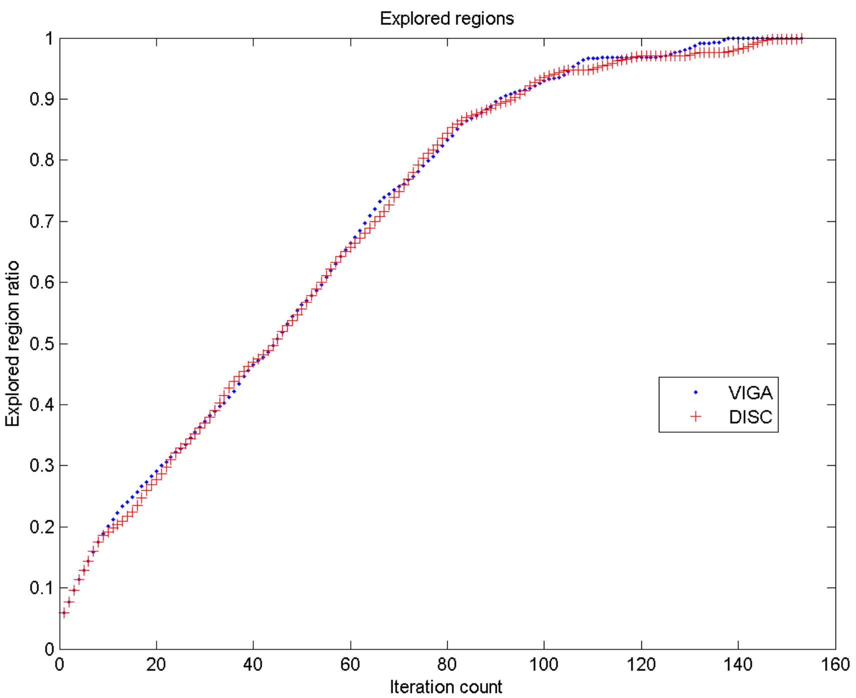

Figure 5.24 : Explored region ratio versus iteration count (300 mines)

The performances of the two methods are compared by changing the robot and mine counts. Iteration count is taken as the main performance criterion. In Figure 5.21, Figure 5.22, Figure 5.23 and Figure 5.24, it can be seen that VIGA method gives better results with mine counts 150 and 300 when the robot count is 4.

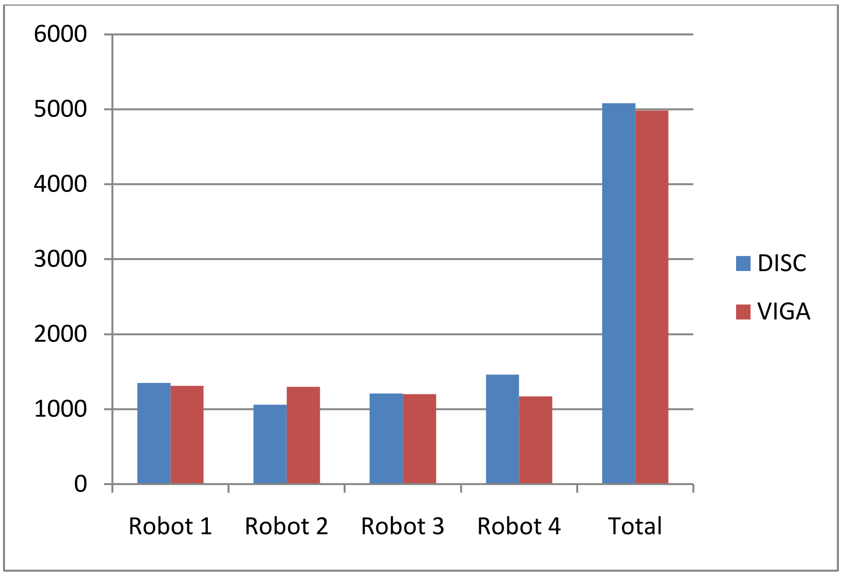

Figure 5.25 : The distances taken by robots in meters (150 mines)

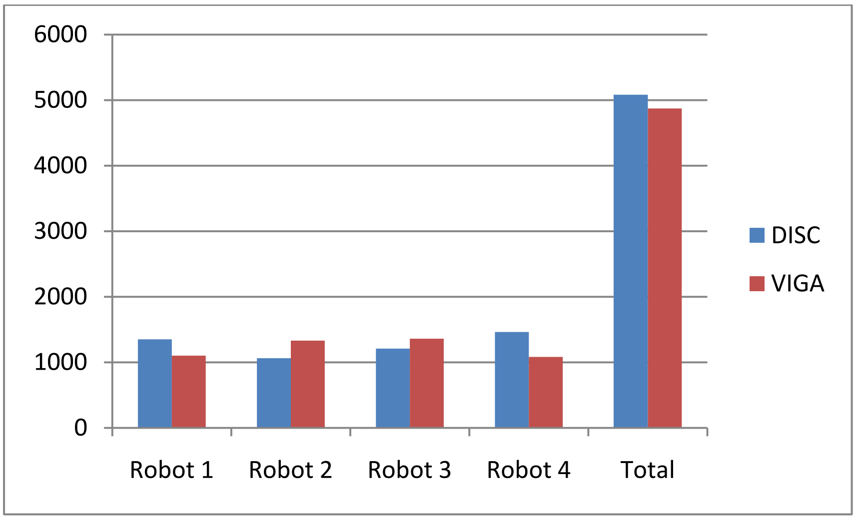

Figure 5.26 : The distances taken by robots in meters (300 mines)

In Figure 5.25, the distances taken by four robots in the simulation with different mine counts (150 and 300) are given. It can be seen that the total distance taken by the four robots is smaller in the VIGA than in the original Discoverage method. So, it can be said that Variable Information Gain Approach requires less total energy effort than the original method for 150 and 300 mines.

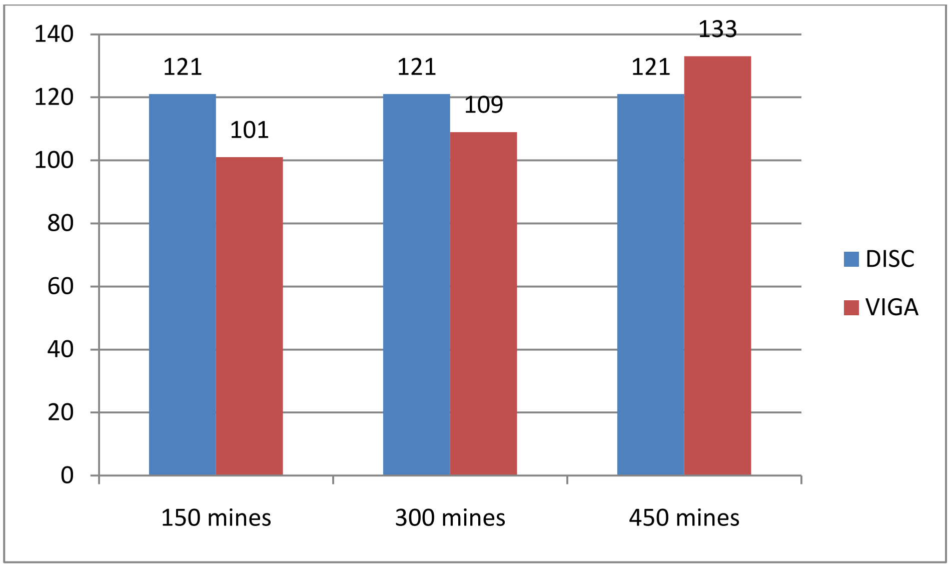

Figure 5.27 : Iteration counts with 5 robots

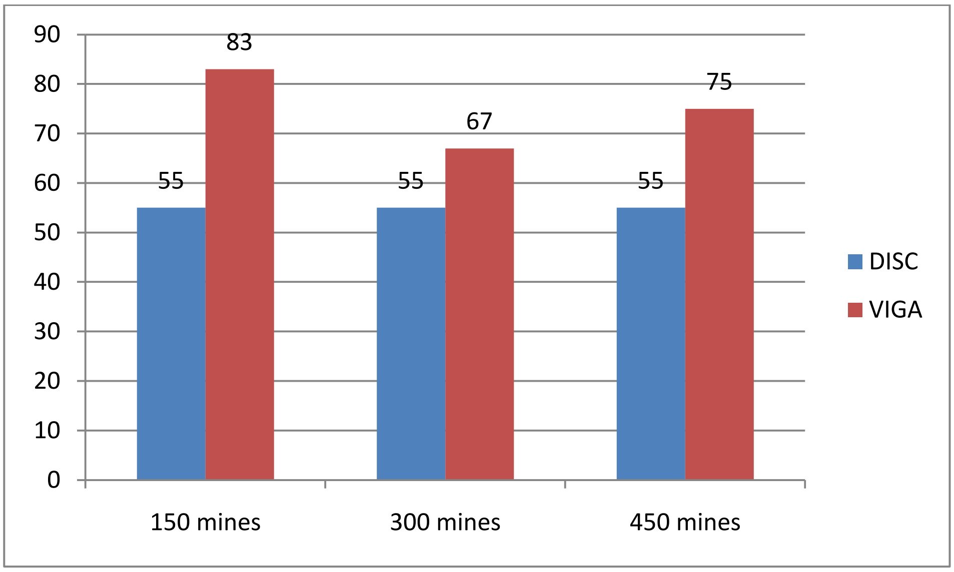

Figure 5.28 : Iteration counts with 10 robots

In Figure 5.27, the results of simulation with different mine counts (150, 300 and 450) with 5 agents are given. The iteration counts of the coverage with variable information gain approach for 150 and 300 mines are less than the original Discoverage method. However, above 300 mines, the Discoverage method gives better results.

In Figure 5.28, the robot count is increased to 10. In conclusion, the Discoverage algorithm gives better results compared to coverage with variable information gains for 150, 300 and 450 mines.

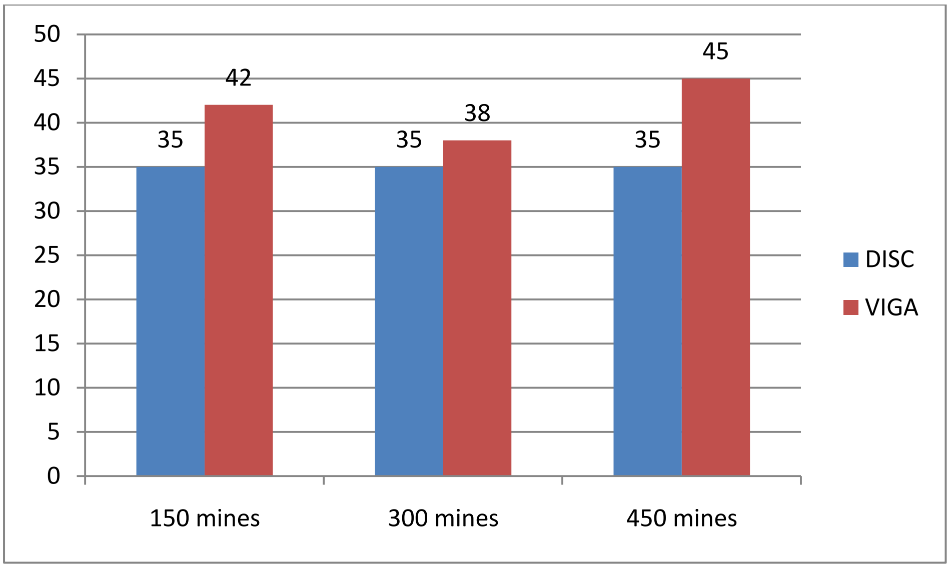

Figure 5.29 : Iteration counts with 20 robots

Also, in Figure 5.29, the same simulation is done with 20 robots. The iteration counts are less than the original method Discoverage in 10 and 20 robots.

From Figure 5.27, Figure 5.28 and Figure 5.29, it can be concluded that the coverage with variable information gains gives better results when the robot count is low (five robots) and the mine count is not greater than 150.

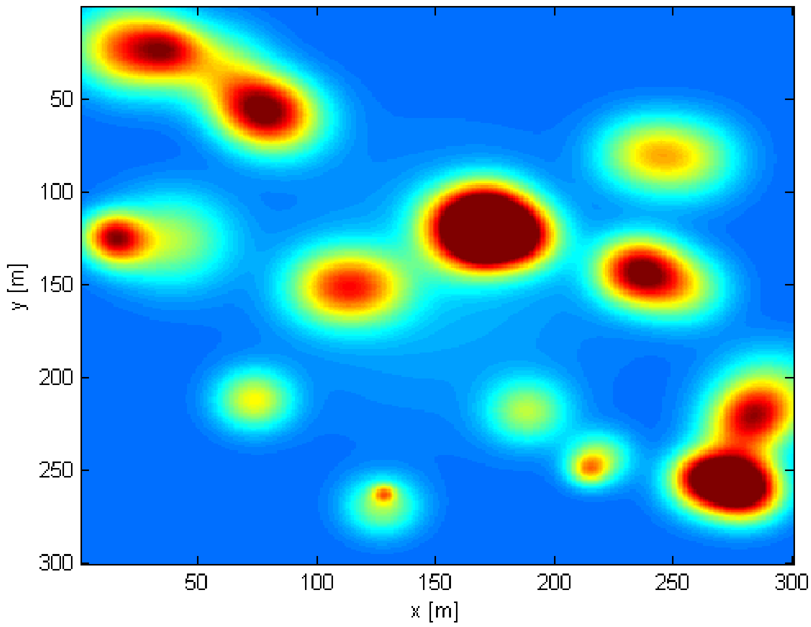

Figure 5.30 : Information gains at the end of the simulation (2D)

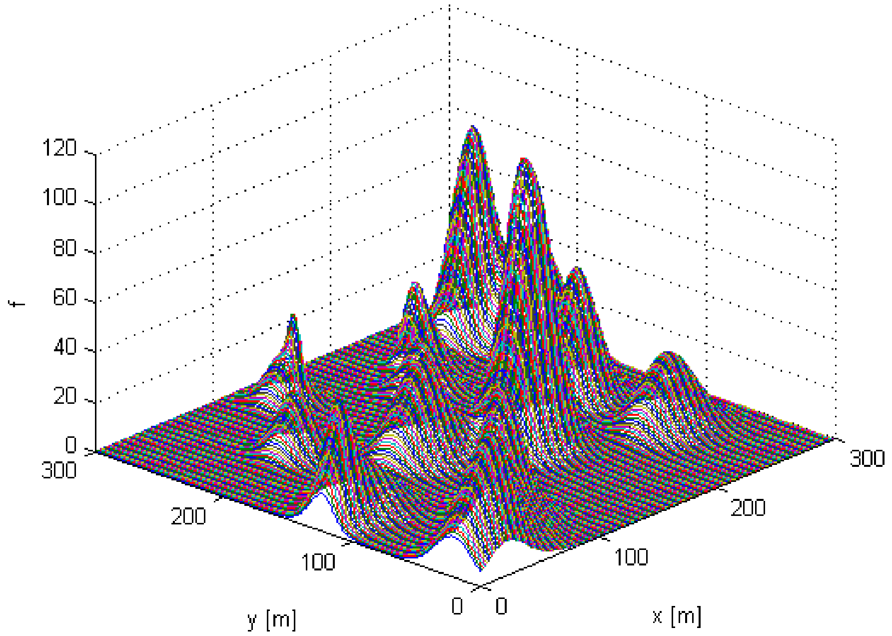

Figure 5.31 : Information gains at the end of the simulation (3D)

In Figure 5.30, the last frame of the simulation can be seen. The colored regions are showing the information gains. Since the variable information gain algorithm inserts Gaussian functions to Φ at the regions with high mine count, some regions have greater information gains than other regions. In Figure 5.31, the 3 dimensional plot of the information gain function Φ is given. Large values of this function show that there are a lot of mines around this region.

6. CONCLUSION

In this thesis, a number of selected Coverage methods are studied with simulation results. The method Discoverage with single and multi robot cases are applied to simulation as an application of mine clearance. In this mine clearance application, the robots are supposed to clear the map including a number of randomly distributed mines by using one of the proposed coverage algorithm approaches. In the simulation studies, the Discoverage approach is extended with two new approaches including Variable Information Gains and collision avoidance. The Discoverage algorithm with Variable Information Gains gives better results compared to the original method when the robot count and mine count is low. In addition to the VIG approach, the Discoverage with collision avoidance algorithm prevents the robots from getting closer than a safety distance in order to avoid collisions. Successful results are obtained in simulation. This approach is important for multi robot real time implementations because of the physical dimensions of the robots. Also, an existing method for coverage with anisotropic sensors is studied under a separate section.

In future, the proposed obstacle avoidance methods for coverage may be implemented for multi robot case. This approach can be extended to collision and obstacle avoidance implementation. Moreover, a real time implementation of the suggested algorithms with multiple agents may be done by using appropriate localization methods, as done in the implementations given in previous works section.

REFERENCES

[1] **Jonathan Ko, Benjamin Stewart, Dieter Fox, Kurt Konolige, and Benson Limketkai**, 2003: A Practical, Decision-theoretic Approach to Multi-robot Mapping and Exploration. *International Conference on Intelligent Robots and Systems*, Las Vegas.

[2] **Jenhui Chen and Li-Ren Li**, 2005: Path Planning Protocol for Collaborative Multi-Robot Systems. In *International Symposium on Computational Intelligence in Robotics and Automation*, Espoo.

[3] **Lovekesh Vig and Juile A. Adams**, 2006: Multi-Robot Coalition Formation. *IEEE Transactions on Robotics*, vol. **22**, no. 4, pp. 637-649, Aug..

[4] **Yu Zhou and Jindong Tan**, 2008: Deployment of Multi-robot Systems under the Nonholonomic Constraint. *International Conference on Advanced Intelligent Mechatronics*, Xi'an.

[5] **C. S. Lim, R. Mamat, and T. Bräunl**, 2009: Market-based Approach for Multi-Team Robot Cooperation. *International Conference on Autonomous Robots and Agents*, Wellington.

[6] **Willi Richert, Ulrich Scheller, Markus Koch, Bernd Kleinjohann, and Claudius Stern**, 2009: Increasing the autonomy of mobile robots by imitation in multi-robot scenarios. *Fifth International Conference on Autonomic and Autonomous Systems*.

[7] **Elias B. Kosmatopoulos, Lefteris Doitsidis, and Kostas Aboudolas**, 2009: A Unified Methodology for Multi-Robot Passive & Active Sensing. *17th Mediterranean Conference on Control & Automation*, Thessaloniki.

[8] **Lei Zhang, René Zapata, and Pascal Lépinay**, 2009: Self-Adaptive Monte Carlo for Single-Robot and Multi-Robot Localization. *International Conference on Automation and Logistics*, Shenyang, 2009.

[9] **Saori Iwanaga, Kazuhiro Ohkura, and Tomoya Matsuda**, 2009: Evolving Adaptive Group Behavior in a Multi-Robot System. *ICROS-SICE International Joint Conference*, Fukuoka.

[10] **Pedro M. Shiroma and Mario F. M. Campos**, 2009: CoMutaR: A framework for multi-robot coordination and task allocation. *International Conference on Intelligent Robots and Systems*, St. Louis.

[11] **Yi Zhang, Li Zeng, Yanhua Li, and Quanjie Liu**, 2009: Multi-robot Formation Control Using Leader-follower for MANET. *International Conference on Robotics and Biomimetics*, Guilin.

[12] **Sung-hwan Kim, Gyungtae Lee, Inpyo Hong, Young-Joo Kim, and Daeyoung Kim**, 2010: New Potential Functions for Multi robot path planning : SWARM or SPREAD. *International Conference on Computer and Automation Engineering*, Chongqing.

[13] **Tong Tao, Yalou Huang, Jing Yuan, Fengchi Sun, and Xiaolin Wu**, 2010: Cooperative Simultaneous Localization and Mapping for Multi-robot: Approach & Experimental Validation. *World Congress on Intelligent Control and Automation*, Jinan.

[14] **Jorge Cortés, Sonia Martinez, Timur Karatas, and Francesco Bullo**, 2004: Coverage Control for Mobile Sensing Networks. *IEEE Transactions on Robotics and Automation*, vol. **20**, no. 2, pp. 243-255.

[15] **Brian Yamauchi**, 1997: A Frontier-Based Approach for Autonomous Exploration. *Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp. 146–151.

[16] **Dominik A. Haumann, Kim D. Listmann, and Volker Willert**, 2010: DisCoverage: A new Paradigm for Multi-Robot Exploration. *2010 IEEE International Conference on Robotics and Automation*, Anchorage, Alaska.

[17] **Ercan U. Acar, Howie Choset, Zhang Yangang, and Mark Schervish**, 2003: Path Planning for Robotic Demining: Robust Sensor-based Coverage of Unstructured Environments and Probabilistic Methods. *The International Journal of Robotics Research*, vol. **22**, no. 7-8.

[18] **Islam I. Hussein and Dusan M. Stipanovic**, 2007: Effective Coverage Control for Mobile Sensor Networks With Guaranteed Collision Avoidance. *IEEE Transactions on Control Systems Technology*, pp. 642-657.

[19] **Mac Schwager, Daniela Rus, and Jean-Jacques Slotine**, 2010: Unifying Geometric, Probabilistic, and Potential Field Approaches to Multi-robot Deployment. *The International Journal of Robotics Research*.

[20] **Islam I. Hussein and Yue Wang**, 2010: Awareness Coverage Control Over Large-Scale Domains With Intermittent Communications. *IEEE Transactions on Automatic Control*, pp. 1850-1859.

[21] **Simon X. Yang, Xiaobu Yuan Chaomin Luo**, 2002: Real-time Area-Covering Operations with Obstacle Avoidance for Cleaning Robots. *Intelligent Robots and Systems Conference*, Switzerland.

[22] **Sameera Poduri and Gaurav S. Sukhatme**, 2004: Constrained Coverage for Mobile Sensor Networks. *International Conference on Robotics and Automation*, New Orleans, LA.

[23] **Ioannis Rekleitis, Vincent Lee-Shue, Ai Peng New, and Howie Choset**, 2004: Limited Communication, Multi-Robot Team Based Coverage. *International Conference on Robotics and Automation*, New Orleans, LA.

[24] **Xiaoming Zheng, Sonal Jain, Sven Koenig, and David Kempe**, 2005: Multi-Robot Forest Coverage. *Intelligent Robots and Systems*, Los Angeles.

[25] **Kjerstin Easton and Joel Burdick**, 2005: A Coverage Algorithm for Multi-robot Boundary Inspection. *International Conference on Robotics and Automation*, Barcelona.

[26] **Noam Hazon and Gal A. Kaminka**, 2005: Redundancy, Efficiency and Robustness in Multi-Robot Coverage. *International Conference on Robotics and Automation*, Barcelona, 2005.

[27] **Wei Li and Christos G. Cassandras**, 2005: Distributed Cooperative Coverage Control of Sensor Networks. *IEEE Conference on Decision and Control*, Seville.

[28] **Noa Agmon, Noam Hazon, and Gal A. Kaminka**, 2006: Constructing Spanning Trees for Efficient Multi-Robot Coverage. *IEEE International Conference on Robotics and Automation*, Orlando.

[29] **Noam Hazon, Fabrizio Mieli, and Gal A. Kaminka**, 2006: Towards Robust On-line Multi-Robot Coverage. *IEEE International Conference on Robotics and Automation*, Orlando.

[30] **Chan Kong Sze, New Ai Peng, and Ioannis Rekleitis**, 2006: Distributed Coverage with Multi-Robot System. *IEEE International Conference on Robotics and Automation*, Orlando.

[31] **Peter F. Hokayem, Dusan Stipanovic, and Mark W. Spong**, 2007: Dynamic Coverage Control with Limited Communication. *American Control Conference*, New York City.

[32] **Xiaoming Zheng and Sven Koenig**, 2007: Robot Coverage of Terrain with Non-Uniform Traversability. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego.

[33] **Peter F. Hokayem, Dusan Stipanovic, and Mark W. Spong**, 2007: On Persistent Coverage Control. *IEEE Conference on Decision and Control*, New Orleans.

[34] **Jonathan Rogge and Dirk Aeyels**, 2007: Sensor coverage with a multi-robot system. *International Symposium on Intelligent Control*, Singapore.

[35] **Nuttapon Boonpinon and Attawith Sudsang**, 2007: Constrained Coverage for Heterogeneous Multi-Robot Team. *International Conference on Robotics and Biomimetics*, Sanya.

[36] **Hao Wu, Guohui Tian, and Xiaolei Li**, 2008: Research on Optimized Partition Algorithm for Multi Robots Collaboration Coverage. *World Congress on Intelligent Control and Automation*, Chongqing.

[37] **Jonathan A. Rogge and Dirk Aeyels**, 2009: Multi-Robot Coverage to locate fixed and moving targets. *IEEE Multi-conference on Systems and Control*, Saint Petersburg.

[38] **Osman Parlaktuna, Aydin Sipahioglu, Gokhan Kirlik, and Ahmet Yazici**, 2009: Multi-robot sensor-based coverage path planning using capacitated arc routing approach. *IEEE Multi-conference on Systems and Control*, Saint Petersburg.

[39] **E. Gonzalez and E. Gerlein**, 2009: BSA-CM: A Multi-Robot Coverage Algorithm. *IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*.

[40] **Ahmet Yazici, Gokhan Kirlik, Osman Parlaktuna, and Aydin Sipahioglu**, 2009: A Dynamic Path Planning Approach for Multi-Robot Sensor-Based Coverage Considering Energy Constraints. *International Conference on Intelligent Robots and Systems*, St. Louis.

[41] **Ke Cheng and Prithviraj Dasgupta**, 2010: Multi-Agent Coalition Formation for Distributed Area Coverage: Analysis and Evaluation. *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*.

[42] **Alessandro Renzaglia, Lefteris Doitsidis, Agostino Martinelli, and Elias B. Kosmatopoulos**, 2010: Cognitive-based Adaptive Control for Cooperative Multi-Robot Coverage. *International Conference on Intelligent Robots and Systems*, Taipei.

[43] **Azwirman Gusrialdi, Sandra Hirche, David Asikin, Takeshi Hatanaka, and Masayuki Fujita**, 2009: Voronoi-based coverage control with anisotropic sensors and experimental case study. *Intel Serv Robotics*, pp. 195–204.

[44] **B. Bethke, M. Valenti, and How J**., 2008: Experimental Demonstration of UAV Task Assignment with Integrated Health Monitoring. *IEEE Robotics and Automation Magazine*, Mar..

[45] **J. How, B. Bethke, A. Frank, D. Dale, and J. Vian**, 2008: Real-time Indoor Autonomous Vehicle Test Environment. *IEEE Control Systems Magazine*, Mar.

[46] **Douglas Vail and Manuela Veloso**, 2003: Dynamic Multi-Robot Coordination. *Multi-Robot Systems: From Swarms to Intelligent Automata,* vol. **2**, pp. 87-100.

[47] **Thuc Vu, Jared Go, Gal Kaminka, and Manuela Veloso**, 2003: MONAD: A Flexible Architecture for Multi-Agent Control. *AAMAS'03*.

[48] **M. Lawitzky, A. Mörtl, and S. Hirche**, 2010: Load Sharing in Human-Robot Cooperative Manipulation. *19th IEEE International Symposium on Robot and Human Interactive Communication*.

[49] **D. Althoff et al.**, 2009: A Flexible Architecture for Real-time Control in Multi-robot Systems. *Proceedings of the Third International Workshop on Human Centered Robotic Systems '09*.

[50] **Mac Schwager, James McLurkin, Jean-Jacques E. Slotine, and Daniela Rus**, 2009: From Theory to Practice: Distributed Coverage Control

Experiments with Groups of Robots. *Springer Tracts in Advanced Robotics*, vol. **54**, pp. 127-136.

[51] **Mac Schwager, James Mclurkin, and Daniela Rus**, 2006: Distributed Coverage Control with Sensory Feedback for Networked Robots. *Proceedings of Robotics: Science and Systems*.

[52] **Mert Turanlı and Hakan Temeltaş**, 2009: Bir Mobil Robot Yörünge İzleyici Sisteminin Tasarımı. *TOK 2009*.

[53] **Azwirman Gusrialdi, Takeshi Hatanaka, and Masayuki Fuj**, 2008: Coverage Control for Mobile Networks with Limited-Range Anisotropic Sensors. *Proceedings of the 47th IEEE Conference on Decision and Control*, Cancun, Mexico.

[54] Wikipedia. [Online]. "http://en.wikipedia.org/wiki/Voronoi_diagram". Retrieved in March 2011.

[55] Wikipedia. [Online]. "http://en.wikipedia.org/wiki/Fortune%27s_algorithm". Retrieved in March 2011.

[56] **Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars**, 2010: *Computational Geometry: Algorithms and Applications*.: Springer.

[57] **Jürg Nievergelt**, 2000: Exhaustive search, combinatorial optimization and enumeration: Exploring the potential of raw computing power. *SOFSEM 2000*.

[58] **Hyuk Lim and C. Jennifer Hou**, 2005: Localization for Anisotropic Sensor Networks. *24th Annual Joint Conference of the IEEE Computer and Communications Societies*.

[59] **King-Yip Cheng, King-Shan Lui, and Vincent Tam**, 2006: Hybrid Approach for Localization in Anisotropic Sensor Networks. *Vehicular Technology Conference*, Melbourne.

[60] **Azwirman Gusrialdi, Sandra Hirche, Takeshi Hatanaka, and Masayuki Fujita**, 2008: Voronoi Based Coverage Control with Anisotropic Sensors. *American Control Conference*, Seattle.

[61] **Katie Laventall and Jorge Cortés**, 2008: Coverage control by robotic networks with limited-range anisotropic sensory. *American Control Conference*, Seattle.

CURRICULUM VITAE

Candidate's full name: **Mert Turanlı**

Place and date of birth: **Istanbul, 20.01.1986**

Permanent Address: **Uphill Court B2 D: 19 Batı Ataşehir/İstanbul, 34746**

Universities and Colleges attended:

Istanbul Technical University, Control Engineering (2008)

Publications:

- Sencan O., **Turanli M.**, Sariyanidi E., Temeltas H., Kurnaz S., Bogosyan S., 2010: A Real-time SLAM Algorithm with Optical Flow based Motion Extraction for Autonomous Robot Navigation. *International Unmanned Vehicles Workshop,* July, 2010 İstanbul, Turkey.
- **Turanli M.**, Temeltas H., 2009: Bir Mobil Robot Yörünge İzleyici Sisteminin Tasarımı. *Türk Otomatik Kontrol Konferansı*, September, 2009 İstanbul, Turkey.
- **Turanli M.**, Koyuncu E., İnalhan G., 2008: Otonom Mobil Robotlar için Genel Amaçlı bir Test Ortamının Tasarımı: İTÜKAL Robotik Test Ortamı. *Türk Otomatik Kontrol Konferansı*, September, 2008 İstanbul, Turkey.