

GENERALIZED SENSOR DEPLOYMENT ALGORITHM
USING OPTIMAL CONTROL THEORY FOR TARGET
DETECTION

A thesis submitted to the Department of Electrical and Electronic Engineering of
Bangladesh University of Engineering and Technology in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL AND ELECTRONIC
ENGINEERING

By

SOUMYA KANTI DAS

B.SC. IN EEE, Bangladesh University of Engineering and Technology, 2016

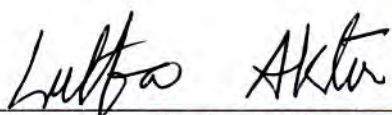
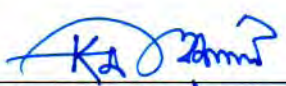
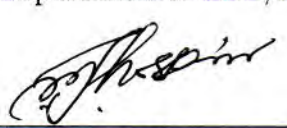
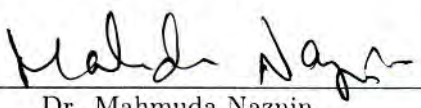


BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY

12 September, 2020

The thesis entitled “GENERALIZED SENSOR DEPLOYMENT ALGORITHM USING OPTIMAL CONTROL THEORY FOR TARGET DETECTION” submitted by Soumya Kanti Das, Student No.: 0416062201, Session: April, 2016, has been accepted as satisfactory in partial fulfillment of the requirements for the degree of MASTER OF SCIENCE IN ELECTRICAL AND ELECTRONIC ENGINEERING on 12th September, 2020.

BOARD OF EXAMINERS

 _____	Chairman (Supervisor)
Dr. Lutfu Akter Professor Department of EEE, BUET, Dhaka.	
 _____	Member (Ex-Officio)
Dr. Md. Kamrul Hasan Head and Professor Department of EEE, BUET, Dhaka.	
 _____	Member
Dr. Md. Farhad Hossain Professor Department of EEE, BUET, Dhaka.	
 _____	Member (External)
Dr. Mahmuda Naznin Professor Department of CSE, BUET, Dhaka.	

CANDIDATE'S DECLARATION

I, do hereby declare that neither this thesis nor any part of it has been submitted elsewhere for the award of any degree or diploma.



Soumya Kanti Das

Table of Contents

List of Tables	vi
List of Figures	vii
List of Symbols	ix
Acknowledgement	xi
Abstract	xii
1 Introduction, Background and Contributions	1
1.1 Introduction and Background	1
1.2 Motivation	4
1.3 Contributions	4
1.4 Organization of the Thesis	5
2 Background Study	7
2.1 Sensing Models	7
2.2 Optimal Control Theory	9
2.3 Linear Quadratic Regulator	12
2.4 LQR Solution Method	14
2.5 Sweep Method	14

3	Proposed Deployment Scheme Based On Optimal Control Theory	16
3.1	System Model, Optimal Control Formulation and Proposed Algorithms	16
3.2	System Model Alteration Due To Change of Parameters	28
3.2.1	Effects of Obstacles	28
3.2.2	Effects of Uncertainty of Deployment	30
4	Simulations and Results	32
4.1	Case 1: Simulation without Obstacles or Uncertainty	33
4.1.1	Simulation Parameters	33
4.1.2	Simulation Results	35
4.2	Case 2: Simulation with Obstacles	47
4.3	Case 3: Simulation with Uncertainty	52
5	Conclusion and Future Works	59
5.1	Conclusion	59
5.2	Future Works	60
	Bibliography	62

List of Tables

4.1	Simulation results for Case 1	35
4.2	Simulation results for Case 2	47
4.3	Simulation results for Case 3	52

List of Figures

2.1	A sequential discrete control system	11
3.1	Presence of obstacle in the operation grid	29
3.2	Block diagram of the proposed deployment process	31
4.1	Required Detection Probabilities	34
4.2	Heterogeneous sensor deployment for a $[20 \times 20]$ grid size with 30 input sensors	36
4.3	Heterogeneous sensor deployment for a $[20 \times 20]$ grid size with 50 input sensors	37
4.4	Difference between achieved and required detection probability for Heterogeneous sensor deployment for a $[20 \times 20]$ grid size	38
4.5	Heterogeneous sensor deployment for a $[40 \times 40]$ grid size	39
4.6	Difference between achieved and required detection probability for Heterogeneous sensor deployment for a $[40 \times 40]$ grid size	40
4.7	Sensors used only for injected target detection for a $[20 \times 20]$ grid size	41
4.8	Effective SE convergence graph of homogeneous sensor deployment for a $[20 \times 20]$ grid size with $\alpha=.15$	42
4.9	Effective SE convergence graph of heterogeneous sensor deployment for a $[20 \times 20]$ grid size with $\alpha=.15$	43
4.10	Effective SE convergence graph of homogeneous sensor deployment for a $[20 \times 20]$ grid size with $\alpha=.15$ at [34]	44

4.11 Line graph of grid dimensions vs sensor number for homogeneous deployment	45
4.12 Line graph of grid dimensions vs sensor number for heterogeneous deployment	46
4.13 Heterogeneous sensor deployment for a $[20 \times 20]$ grid size with obstacles	47
4.14 Difference between achieved and required detection probability for Heterogeneous sensor deployment for a $[20 \times 20]$ grid size with obstacles	48
4.15 Heterogeneous sensor deployment for a $[40 \times 40]$ grid size with obstacles	49
4.16 Difference between achieved and required detection probability for Heterogeneous sensor deployment of for a $[40 \times 40]$ grid size with obstacles	50
4.17 Sensors used only for injected target detection for a $[20 \times 20]$ grid size with obstacles	51
4.18 Heterogeneous sensor deployment for a $[20 \times 20]$ grid size with uncertainty	53
4.19 Difference between achieved and required detection probability for Heterogeneous sensor deployment for a $[20 \times 20]$ grid size with uncertainty	54
4.20 Heterogeneous sensor deployment for a $[40 \times 40]$ grid size with uncertainty	55
4.21 Difference between achieved and required detection probability for Heterogeneous sensor deployment for a $[40 \times 40]$ grid size with uncertainty	56
4.22 Sensors used only for injected target detection for a $[20 \times 20]$ grid size in the presence of uncertainty of deployment	57

List of Symbols

- η : Two dimensional operation grid.
- N_x/N_y : Grid point numbers along **X/Y**-axis.
- L : Types of sensors based on their sensing radii.
- N : Total number of sensors.
- $S(d)$: Signal energy as a function of distance d .
- N_o : Noise.
- r_d : Sensing radius matrix.
- α : Attenuation factor.
- $p_{detect(t)}(\mathbf{x}, \mathbf{y})$: Exponential individual achieved detection probability vector for t type of sensor at (\mathbf{x}, \mathbf{y}) point of size $N_x N_y \times \mathbf{1}$.
- $p_{miss(t)}(\mathbf{x}, \mathbf{y})$: Individual achieved miss probability vector for t type of sensor at (\mathbf{x}, \mathbf{y}) point of size $N_x N_y \times \mathbf{1}$.
- $v(i, j)$: Deployment vector of size $N_x N_y \times \mathbf{1}$ at (i, j) point.
- $PM_{overall(t)}(\mathbf{x}, \mathbf{y})$: Overall achieved miss probability vector for t type of sensor at (\mathbf{x}, \mathbf{y}) point of size $N_x N_y \times \mathbf{1}$.
- $pm_{(t)}(\mathbf{x}, \mathbf{y})$: Logarithmic overall achieved miss probability matrix for t type of sensor at (\mathbf{x}, \mathbf{y}) point of size $N_x N_y \times \mathbf{1}$.
- $a(t)$: user defined function for t type of sensor of size $N_x N_y \times N_x N_y$.
- A : best $a(t)$ matrix of size $N_x N_y \times N_x N_y$.
- $pd_{overall}^{req}$: Overall required detection probability vector of size $\mathbf{1} \times N_x N_y$.
- $pm_{overall}^{req}$: Overall required miss probability vector of size $\mathbf{1} \times N_x N_y$.

- \mathbf{X}_n : State of dynamic control system at n^{th} discrete step.
- \mathbf{I} : Cost function.
- n : Medium propagation constant.
- \mathbf{H}_n : Scalar Hamiltonian function at n^{th} discrete step.
- \mathbf{Q}, \mathbf{R} : Non negative definite diagonal weighting matrices also known as designer's parameters of size $N_x N_y \times N_x N_y$.
- \mathbf{Q}_f : Non negative definite diagonal weighting matrix at the final step of calculation of size $N_x N_y \times N_x N_y$.
- σ_i/σ_j : Standard deviations along \mathbf{X}/\mathbf{Y} -axis.
- \mathbf{P}_G : Joint probability density function.

Acknowledgement

Firstly, I thank almighty God for giving me sufficient wit to understand the problem correctly and to develop algorithms and codes accordingly to solve it.

Secondly, I thank my thesis supervisor Dr. Lutfu Akter. She has provided me with all the important documents needed to fully understand the theoretical background of the problem. Moreover, in the field of coding and formatting of thesis writing, she has been an immense help for me.

Abstract

There has been a plethora of experiments in the field of optimization of sensor deployment for target detection. Since most of those are heuristic in nature, they are optimal for a particular experiment but not universally optimal covering mathematical convergence. Moreover, none of those algorithms is applicable for heterogeneous sensor deployment. In this thesis, a generalized framework for sensor deployment based on optimal control theory for target detection is formulated and required algorithm is developed accordingly. Here, we consider the sensing radii of different sensors as the sole criterion for defining whether a deployment is homogeneous or heterogeneous. The algorithm satisfies for both homogeneous and heterogeneous sensor deployment optimization. Simulation results illustrate its effectiveness. Later we modify our algorithm to satisfy the effects of obstacles and uncertainty of deployment process and analyze the performances of the algorithm.

Chapter 1

Introduction, Background and Contributions

In this chapter, at first we provide a brief literature review on existing sensor deployment techniques. Then we state the motivation and the contributions of the thesis.

1.1 Introduction and Background

Sensor networks have emerged as a viable solution for many detection and surveillance applications. Sensor deployment is mainly about the idea of distributing sensors in the area of interest to detect and track identifiable targets and phenomena of interest with high accuracy. Numerous maverick and adjunct works have been done in the field of optimization of sensor deployment for target detection and tracking.

There are mainly two approaches of sensor deployment- deterministic and non deterministic. In deterministic sensor deployment, sensors are deployed in a particular strict geometric pattern to achieve the objectives [1-8]. In [1-3], **2D** deterministic model is used. In [1] and [2], it is shown with mathematical calculations that hexagonal deployment pattern would be the optimal coverage pattern. In [3], the operational area is rectangular and is sub-divided into two regions: (i) central region

and (ii) edge region. In [4-8], **3D** deterministic model is used. In [4], it is shown with mathematical calculations that truncated octahedron deployment pattern would be the optimal coverage pattern. In [5], it is shown that the deployment pattern would vary with the number of sensors. In [6], a special type of population-based swarm intelligence SI (Swarm Intelligence) algorithm is used named as BSO (Brain Swarm Optimization). Here a triangular tessellation pattern is proposed. In [7] and [8], the operation grid-points are modeled as Voronoi cells and the sensors are to be filled inside the centroid of each cell for optimization. However, the efforts in [1-8], are all regular geometric patterns and do not take into account of the effect of obstacles situated in the study gird.

In non deterministic approach [9-30], there is no fixed patterned sensor deployment but rather random deployment. To optimize the non deterministic deployment process, a number of mathematical models like PSO (Particle Swarm Optimization), Bio inspired and Enhanced Firework are used. In [9], a discrete PSO method called DPSO (Discrete Particle Swarm Optimization) algorithm is used in non convex region. In [10], a deployment method of optimal Underwater Acoustic Sensor Networks (UWASNs) for 3D environment to detect antisubmarine by using PSO (Particle Swarm Optimization) is formulated. In [11], IABPSO (Improved Adaptive Binary Particle Swarm Optimization) is used in the application of optimized sensor deployment in green buildings. In [12], a new variety of PSO named QBPSO (Quantum Behaved Particle Swarm Optimization) is proposed. In [13], a generic version of PSO is used for area coverage called GAPSO (Generic Particle Swarm Optimization). In [14], a bio inspired genetic algorithm is used. In [15], a sequential deployment method is used towards optimization in an unknown terrain. In [16], NP complete **2D** optimal coverage VANET sensor deployment is used. In [17], ABC (Artificial Bee Colony) method is used. In [18], ACO (Ant Colony Optimization) method is used in WSN with redundancy check. In [19], an enhanced firework algorithm is used which is a variant of SI (Swarm Intelligence) algorithm. In [20],

a CPM (Coverage Probability Model) is used for sensor deployment optimization. In [21], two linear programming formulations based on real pollutant's dispersion model is used for optimization. In [22], a GSA (Greedy Strategy Approximation) algorithm is used to model a composite event and this is a heterogeneous sensor deployment. In [23], RL (Restrained Lloyd) and DA (Deterministic Annealing) algorithms are used for sensor optimizing. In [24], railway environment monitoring system is modeled and WSN is deployed and optimized. In [25], GTF (Graph Theoretical Framework) is used to connect mobile sensors for optimization. In [26], NP complete GBPT (Greedy Based Polynomial Time) algorithm is used to optimize sensor deployment. In [27], node deployment optimization on industrial environment is achieved in a heuristic manner. In [28], SA (Simulated Annealing) method is used for optimization of sensors for lunar surveys. In [29], a non linear programming algorithm is used on IoT (Internet of Things) for self localization. In [30], Poisson random distribution is used to simulate the deployment process and a hierarchical algorithm and disk modeling are used to optimize it.

From [1-30], all the processes irrespective of deterministic or random in nature, use static sensor deployment. Whereas, in [31-33], different mobile sensor random deployment optimization methods are used. In [31], hybrid nodes consisting of sensors and mobile phones are nearly optimized. In [32], a blanket coverage of mobile sensor network is achieved. In [33], a simple peer-to-peer communication is used for mobile sensor network for optimization while keeping the sensors at the centroid of the Voronoi cells.

However, None of the methods discussed in [1-33], are mathematically proven optimal but optimal for a situation rather. There is a strong need for a mathematically optimal model. In [34], a novel approach on sensor deployment is effectuated hinged on optimal control theory for homogeneous sensors. Here, the term homogeneous means that by definition all the sensors are of same type and radius. Using dynamic linear equation for the system and formulating a quadratic cost function,

the deployment problem is maneuvered as a LQR (Linear Quadratic Regulator) problem in [35]. Here, the control vectors at each discrete step are the position vectors of the sensors. Hence, the LQR optimality conditions are formed by applying KKT (Karush-Kuhn-Tucker) conditions [35], [36], [37] on the system model. Finally, sweep method [36] is used to solve those KKT conditions to obtain the coordinates of the sensors for optimal deployment process. The proposed algorithm in [34] is not only based on a mathematical framework but also uses a minimum number of sensors to satisfy given detection requirements. This is very desirable in the field of target detection.

1.2 Motivation

From the previous section, it is ratified that all the previous works in [1-33], are of heuristic in nature. Only in [34], we can find an algorithm based on a mathematical model (optimal control theory). Yet it is only designed for homogeneous sensors. There can be a situation where a limited number of sensors are available of a type which is less than the required number of sensors for an application. In this case, we can use heterogeneous sensors that are similar in type but different in radius. In this thesis our main motivation is to develop an algorithm based on optimal control theory to optimize the heterogeneous sensor deployment process. Then it is also our motive to modify the proposed algorithm for two practical cases: (i) obstacles and (ii) uncertainty of deployment to make it more realistic.

1.3 Contributions

Most of the prior works on sensor deployment are heuristic in nature and none of those are generalized for the optimization of both homogeneous and heterogeneous sensor deployment. Hence, in this thesis, we attempt to develop a generalized algorithm which is non-heuristic nature and pivoted on a mathematical model. We

use optimal control theory for our benefit. In this thesis, mainly we have two contributions:

(i) First, we declare the term heterogeneous on the pedestal of the difference of sensing radii of various sensors. Our primary contribution is to develop a generalized framework capable of both homogeneous and heterogeneous sensor deployment for target detection based on optimal control theory. In other words, a generalized sensor deployment problem is first considered as an optimal control theory based problem by proposing an algorithm to select a unique overall miss probability matrix from the overall miss probability matrices of each type of sensor on the basis of minimum value of miss probabilities at particular detection points. That is how this method is sufficient for both homogeneous and heterogeneous sensor deployment simultaneously since both processes use a singular selection matrix rather than heterogeneous process using multiple selection matrices and homogeneous process using a singular selection matrix. Then, it is modeled as a discrete LQR problem to find out a set of optimal conditions by minimizing its cost function. Next, sweep method [36] is used to solve the LQR problem and from the solution, the co-ordinates of the sensors are found. Finally, a distance dependent algorithm is proposed to identify the type of each deployed sensor. Simulation results show the effectiveness of our proposed algorithms.

(ii) Our secondary contribution is to consider two natural consequences of sensor deployment: (i) presence of obstacles and (ii) uncertainty of the deployment process. We then modify the algorithm and observe the performances accordingly. In [38] and [39], the theoretical background of obstacle modeling is described and from [40] the modeling of uncertainty of deployment is formulated.

1.4 Organization of the Thesis

The thesis is organized as follows: Firstly, Chapter 2 discusses the theoretical background. Secondly, Chapter 3 describes the system model, the problem formulation

along with proposed algorithms. Next, Chapter 4 delineates the required alterations of the original model for obstacles in the operation-grid and augmentation of Gaussian probability theory of distribution into the original algorithms for the case of uncertainty [40]. Then, Chapter 5 presents the simulations and results of the proposed algorithms. Finally, Chapter 6 concludes the paper and discusses our intended future works.

Chapter 2

Background Study

In this chapter, we discuss the necessary theoretical backgrounds that we need for our system model. First, we talk about the different mathematical sensing models to attune the sensing behavior of the sensors and our pick of the model among those. Here, we have to establish a discrete dynamic system model and a cost function for our problem. Our goal is to optimize this discrete dynamic system model by minimizing the cost function and we need optimal control theory based approach to do that. So we describe how to construct these discrete dynamic system model and cost function based on optimal control theory and apply KKT conditions [35], [36], [37] to extract optimality conditions. Next, we discuss about how to model a discrete dynamic system as a LQR (Linear Quadratic Regulator) problem. Finally, we describe the sweep method [36] for solving a LQR problem.

2.1 Sensing Models

In this section, we discuss several sensing models that have been used to delineate sensor detection process. Two points are required to be noted here. First one is the sensor's effective coverage area and second one is the sensors robustness to different degradation from ideal model such as obstacles in the operational grid and uncertainty in the deployment process. We mathematically model the sensing behavior

of the sensors on a rectangular grid having a dimension of $N_x \times N_y$ although grid of any shape is applicable. That is what we call a ROI (Region of Interest). There are mainly four sensing models that we discuss here.

1. Point Model:

In this model, a sensor does not require an effective coverage area rather the phenomenon has to go through the sensor itself so that the sensor can sense it. For example, gas sensors sense chemical gases when those gases go through the sensors.

2. Disc Model:

In this model, a sensor do have an effective coverage area to work with. When the target reaches inside the coverage area the sensor detects the target. The effective coverage can be a circle or a sphere depending on the nature of the model (2D/3D). So in effect, this is a binary decision model. When the target is inside the effective coverage area or sensing radius r_d , the detection probability is 1 otherwise 0.

If the sensor is in (i, j) point and the target is in (x, y) point on the grid, distance is

$$d(x, y) = \sqrt{(x - i)^2 + (y - j)^2} \quad (2.1)$$

Hence, the detection probability $p_{detect}(x, y)$ with target at (x, y) point is

$$p_{detect}(x, y) = \begin{cases} 1, & \text{if } d(x, y) \leq r_d \\ 0, & \text{if } d(x, y) > r_d \end{cases} \quad (2.2)$$

Disc model is widely used in acoustic and seismic sensors.

3. Distance Dependent Model:

In this model, sensors detection probability is similar as disc model but the difference is that inside the effective coverage area, its detection probability doesn't stay fixed at 1 but varies depending on the distance between sensor and target. The

detection probability $p_{detect}(\mathbf{x}, \mathbf{y})$ with target at (\mathbf{x}, \mathbf{y}) point is

$$p_{detect(t)}(\mathbf{x}, \mathbf{y}) = \begin{cases} e^{-\alpha d(\mathbf{x}, \mathbf{y})}, & \text{if } d(\mathbf{x}, \mathbf{y}) \leq r_d \\ 0, & \text{if } d(\mathbf{x}, \mathbf{y}) > r_d \end{cases} \quad (2.3)$$

where, α is the decay parameter depending on sensor's design and environment. This model is more realistic than disc model since magnitude of detection signal decays as it travels through a particular medium.

This is the model which we will use for our thesis.

4. Energy Detector Model:

In this model, the sensor measures the energy of the signal that target emits. This depends on distance and noise. The measurement of energy is

$$\mathbf{y} = \mathbf{S}(d) + \mathbf{N}_o^2 \quad (2.4)$$

where, $\mathbf{S}(d)$ is the energy of the signal and \mathbf{N}_o^2 is the energy of the noise.

2.2 Optimal Control Theory

Optimal control theory ensures that a dynamic system can be used to get prospective results. It also provides an analytical solution to a problem and handles multi-variables. Let us assume a discrete dynamic system:

$$\mathbf{X}_{n+1} = \mathbf{f}(\mathbf{X}_n, \mathbf{v}_n), \quad n = 0, 1, \dots, N - 1 \quad (2.5)$$

Here n is the step number, N is the total number of time stamps, \mathbf{X}_n is the state vector and \mathbf{v}_n is the control vector. The function $\mathbf{f}(\mathbf{X}_n, \mathbf{v}_n)$ depends on both \mathbf{X}_n and \mathbf{v}_n .

Then we assume the initial condition as

$$\mathbf{X}_0 = \mathbf{X}_{in} \quad (2.6)$$

The cost function I is formulated as

$$I = L(\mathbf{X}_N) + \sum_{n=0}^{N-1} V_n(\mathbf{X}_n, \mathbf{v}_n) \quad (2.7)$$

where, $L(\mathbf{X}_N)$ is the final state and $V_n(\mathbf{X}_n, \mathbf{v}_n)$ is the dynamic portion of the function. In optimal control theory, the aim is to maneuver a set of control vectors \mathbf{v}_n to minimize the cost function I .

The optimal control problem can be stated as a constrained optimization problem and so we can use Lagrange multipliers and variational calculus to find out optimal conditions to minimize the cost function. We define the scalar Hamiltonian function [35] H_n as

$$H_n = V_n(\mathbf{X}_n, \mathbf{v}_n) + \lambda_{n+1}^T f_n(\mathbf{X}_n, \mathbf{v}_n) \quad (2.8)$$

where, λ_n is the n^{th} Lagrangian multiplier.

Using the Hamiltonian [35] derive the discrete KKT (Karush-Kuhn-Tucker) conditions [35], [36], [37] given below:

$$\mathbf{X}_{n+1} = f(\mathbf{X}_n, \mathbf{v}_n) \quad (2.9)$$

$$\mathbf{X}_0 = \mathbf{X}_{in} \quad (2.10)$$

$$\lambda_n = \nabla_{\mathbf{X}_n}^T f(\mathbf{X}_n, \mathbf{v}_n) \lambda_{n+1} + \nabla_{\mathbf{X}_n} V_n \quad (2.11)$$

$$\lambda_N = \nabla_{\mathbf{X}_N} L \quad (2.12)$$

$$\mathbf{0} = \nabla_{\mathbf{v}_n} V_n + \lambda_{n+1}^T \nabla_{\mathbf{v}_n} f(\mathbf{X}_n, \mathbf{v}_n) \quad (2.13)$$

where, $\nabla_{\mathbf{X}} L$ is the differential of L with respect to \mathbf{X} .

It is notable that the boundary conditions for the set of equations (2.9-2.13) are at $n = 0$, since $\mathbf{X}_0 = \mathbf{X}_{in}$ and at $n = N$, since $\lambda_N = \nabla_{\mathbf{X}_N} L$. So this can be called as a two-point boundary problem.

With regards to the deployment problem, this can be mapped as an optimal control problem. The detection problem can be viewed as a system. The state

of the system (\mathbf{X}_n) corresponds to the network's overall detection performance when n number of sensors are deployed. As the system's performance depends on the number of sensors deployed, the additional sensor deployment ($n \rightarrow n + 1$) is equivalent to the effect of deploying a control vector in the system. Hence, sequential deployment of sensors corresponds to a series of control vectors deployed in the system. The system's evolution function \mathbf{f}_n depends on the change in the system when a control vector (i.e., deployment of an additional sensor) is applied on the system. This change depends on network's design, characteristics of the deployed sensors and detection requirements. The final element is the cost function \mathbf{I} . The aim is to minimize the the squared difference between achieved and required detection performance and this can easily be seen as equivalent to minimizing \mathbf{I} . This can be done by solving the KKT [35], [36], [37] based LQR optimality conditions through sweep method [36]. Hence total problem can be viewed as a sequential discrete control system shown in Figure 2.1.

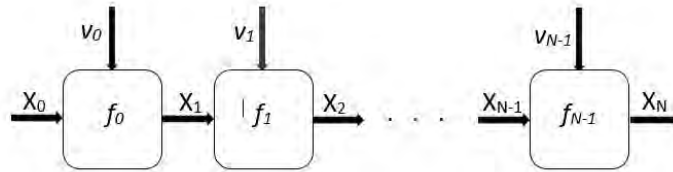


Figure 2.1: A sequential discrete control system

2.3 Linear Quadratic Regulator

The linear quadratic regulator (LQR) problem is very well studied. One advantage of LQR method is that we can use KKT (Karush-Kuhn-Tucker) conditions [35], [36], [37] to set up specific optimality conditions (2.9-2.13) which are sufficient enough to solve the problem. There are well defined approaches such as sweep method [36] to solve a set of optimality conditions resulting from the application of discrete KKT conditions [35], [36], [37]. In this section, we discuss the LQR formulation and modeling of the deployment problem as a LQR problem.

In a LQR problem, the system function \mathbf{f} is linear and can be stated as

$$\mathbf{X}_{n+1} = \mathbf{C}_n \mathbf{X}_n + \mathbf{D}_n \mathbf{v}_n, \quad n = 0, 1, 2, \dots, N-1 \quad (2.14)$$

where, \mathbf{C}_n and \mathbf{D}_n are matrices of appropriate dimensions. Also the cost function \mathbf{I} is quadratic and stated as

$$\mathbf{I} = \frac{1}{2} \mathbf{X}_N^T \mathbf{Q}_f \mathbf{X}_N + \frac{1}{2} \sum_{n=0}^{N-1} (\mathbf{X}_n^T \mathbf{Q}_n \mathbf{X}_n + \mathbf{v}_n^T \mathbf{R}_n \mathbf{v}_n) \quad (2.15)$$

Here, \mathbf{R}_n , \mathbf{Q}_n and \mathbf{Q}_f are non negative definite diagonal weighting matrices with appropriate dimensions. \mathbf{R}_n and \mathbf{Q}_n are also called as designer's parameters. In order to have a positive value of the cost function \mathbf{I} , the possible combination of \mathbf{X}_n and \mathbf{v}_n and the weighting matrices have to be non negative.

So we can use Hamiltonian function [35] with regards to LQR as

$$\mathbf{H}_n = \frac{1}{2} \mathbf{X}_n^T \mathbf{Q}_n \mathbf{X}_n + \frac{1}{2} \mathbf{v}_n^T \mathbf{R}_n \mathbf{v}_n + \lambda_{n+1}^T (\mathbf{C}_n \mathbf{X}_n + \mathbf{D}_n \mathbf{v}_n) \quad (2.16)$$

Applying KKT conditions [35], [36], [37] at equations (2.14-2.16), we can achieve the optimality conditions form equations (2.9-2.13) as,

$$\mathbf{X}_{n+1} = \mathbf{C}_n \mathbf{X}_n + \mathbf{D}_n \mathbf{v}_n, \quad n = 0, 1, \dots, N-1 \quad (2.17)$$

$$\mathbf{X}_0 = \mathbf{X}_{in} \quad (2.18)$$

$$\lambda_n = C_n^T \lambda_{n+1} + Q_n X_n \quad (2.19)$$

$$\lambda_N = Q_f X_N \quad (2.20)$$

$$v_n = -R_n^{-1} D_n^T \lambda_{n+1} \quad (2.21)$$

The LQR problem is a very well behaved problem and can be analytically solved by solving equations (2.17-2.21). The linearity of f_n and convexity of cost function I means that LQR optimality conditions, derived from the KKT conditions [35], [36], [37] are sufficient. For above reason, the modeling of an optimal control problem as a LQR problem is desirable.

2.4 LQR Solution Method

In this section, we discuss sweep method [36] to solve the LQR problem. LQR problem can be classified as static or dynamic depending on the system evolution function \mathbf{f}_n remaining same or changing with the inclusion of a control vector. In other words, \mathbf{C}_n and \mathbf{D}_n are functions of $\mathbf{n} = \mathbf{0}, \mathbf{1}, \dots, \mathbf{N} - \mathbf{1}$ in a dynamic problem and fixed in a static problem. So in a dynamic LQR problem the optimal control vector is calculated on the basis of current system evolution function. Also the designer's parameters remain same for static LQR problem and evolve for dynamic LQR problem. Therefore, solving a dynamic LQR problem is stated as 1-step horizon optimization problem and static LQR problem as \mathbf{N} -step optimization problem.

2.5 Sweep Method

The sweep method [36] is structured to solve static LQR problem or in other words, \mathbf{N} -step optimization problems. Since this is a case of static LQR problem, the designer's parameters \mathbf{R}_n and \mathbf{Q}_n remain fixed irrespective of state changing. So we can rename them as \mathbf{R} and \mathbf{Q} . It is based on the observation that the boundary condition on the Lagrange multiplier is stated as

$$\lambda_N = \mathbf{Q}_f \mathbf{X}_N. \quad (2.22)$$

Based on sweep method [36], equation (2.22) can be re-written as

$$\lambda_n = \mathbf{P}_n \mathbf{X}_n, \quad \mathbf{n} = \mathbf{0}, \mathbf{1}, \dots, \mathbf{N} - \mathbf{1} \quad (2.23)$$

where, \mathbf{P}_n is a square matrix with proper dimensions.

Substituting λ_n in equation (2.21) we get

$$\mathbf{v}_n = -\mathbf{R}^{-1} \mathbf{D}^T \mathbf{P}_{n+1} \mathbf{X}_{n+1}, \quad \mathbf{n} = \mathbf{0}, \mathbf{1}, \dots, \mathbf{N} - \mathbf{1}. \quad (2.24)$$

Since $\mathbf{X}_{n+1} = \mathbf{C}_n \mathbf{X}_n + \mathbf{D}_n \mathbf{v}_n$ we can express \mathbf{v}_n as

$$\mathbf{v}_n = -\mathbf{G}_n \mathbf{X}_n, n = 0, 1, \dots, N - 1 \quad (2.25)$$

where, \mathbf{G}_n is a square matrix with proper dimensions and given as

$$\mathbf{G}_n = (\mathbf{R} + \mathbf{D}^T \mathbf{P}_{n+1} \mathbf{D})^{-1} \mathbf{D}^T \mathbf{P}_{n+1} \mathbf{C}. \quad (2.26)$$

From above equations we can define \mathbf{P}_n as

$$\mathbf{P}_n = \mathbf{C}^T (\mathbf{P}_{n+1} - \mathbf{P}_{n+1} \mathbf{D} \mathbf{S}^{-1} \mathbf{D}^T \mathbf{P}_{n+1}) \mathbf{C} + \mathbf{Q}_n \quad (2.27)$$

where, \mathbf{S}_n is a square matrix with proper dimensions and given as

$$\mathbf{S}_n = \mathbf{R} + \mathbf{D}^T \mathbf{P}_{n+1} \mathbf{D}. \quad (2.28)$$

The boundary condition of equation (2.27) can be acquired from equation (2.22) and (2.23) as

$$\mathbf{P}_N = \mathbf{Q}_f. \quad (2.29)$$

Equation (2.27) can be described as a discrete time algebraic Riccati equation.

So in this chapter, we have discussed the necessary theoretical backgrounds to understand the mathematical modeling of the system, considered in this thesis. In the next chapter, we will formulate our system model and describe the proposed algorithms.

Chapter 3

Proposed Deployment Scheme Based On Optimal Control Theory

In this chapter, at first we state our system model followed by sensor deployment problem formulation. Later on we describe the solution of deployment problem based on optimal control theory. Finally, we discuss two other features on the initial system model to make it more realistic and modify the original algorithms accordingly.

3.1 System Model, Optimal Control Formulation and Proposed Algorithms

Before describing our system model, we first consider some assumptions for the convergence of the model which are given below:

- (1) Individual sensors are the sources and fusion center is the sink and network architecture is parallel architecture with fusion center.
- (2) Required detection probability is predetermined by end users and works as a threshold value for each point in case of both deployment and decision making.
- (3) It is a non collaborative detection networks so 'OR' rule is used at FC which is overall achieved miss probability.
- (4) No false alarm rate is considered.

- (5) Noiseless communication is considered.
- (6) The performance degradation of sensors with time is ignored.
- (7) Each of the deployed sensors will be on for all the time until it is drained off.
- (8) In active hours, energy of the sensors are not depleted.
- (9) Only detection process before actual deployment is considered. Meanwhile, parameter like communication radius which is significant in communication after deployment process is not in our focus and is not considered.

Here, in our system model, we use a $\mathbf{N}_x \times \mathbf{N}_y$ sample-grid ($\boldsymbol{\eta}$) for problem formulation. The types of sensors we used here are heterogeneous based on their sensing radii values. Let us assume that there are \mathbf{L} class of sensors and \mathbf{N} number of sensors and the sensing range matrix is

$$\mathbf{r}_d = \begin{bmatrix} r_{d1} & r_{d2} & r_{d3} & \dots & r_{dL} \end{bmatrix} \quad (3.1)$$

If the sensor is in (i, j) point and the target is in (\mathbf{x}, \mathbf{y}) point on the grid, distance is

$$\mathbf{d}(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - i)^2 + (\mathbf{y} - j)^2}. \quad (3.2)$$

Henceforth, the detection probability equation of any sensor in any class in the set becomes

$$\mathbf{p}_{detect(t)}(\mathbf{x}, \mathbf{y}) = \begin{cases} e^{-\alpha \mathbf{d}(\mathbf{x}, \mathbf{y})}, & \text{if } \mathbf{d}(\mathbf{x}, \mathbf{y}) \leq r_d(t) \\ 0, & \text{if } \mathbf{d}(\mathbf{x}, \mathbf{y}) > r_d(t) \end{cases} \quad (3.3)$$

where, $\mathbf{t} = 1$ to \mathbf{L} and $\mathbf{d}(\mathbf{x}, \mathbf{y})$, $r_d(t)$ and α are Euclidean distance between target and sensor, sensor range, and attenuation factor respectively.

‘OR’ fusion rule is adopted. Under this rule, if a target is detected by one or more sensors, we assume that the target is detected. The opposite of this is the miss probability where it is assumed that the target point is not covered even by a single sensor. According to this rule, we would like to use miss probability instead

of detection probability in our thesis which is

$$\mathbf{p}_{miss(t)}(\mathbf{x}, \mathbf{y}) = \mathbf{1} - \mathbf{p}_{detect(t)}(\mathbf{x}, \mathbf{y}). \quad (3.4)$$

where, $t = 1$ to L and (\mathbf{x}, \mathbf{y}) is the detection point.

The deployment vector equation is

$$\mathbf{v}(i, j) = \begin{cases} \mathbf{1}, & \text{if there is presence of a sensor at } (i, j) \text{ point} \\ \mathbf{0}, & \text{if there is absence of a sensor at } (i, j) \text{ point} \end{cases} \quad (3.5)$$

It determines whether or not there is presence of a sensor in a grid point.

By taking ‘OR’ fusion rule and combining equation (3.4) and (3.5) we can extract the overall miss probability as

$$\mathbf{PM}_{overall(t)}(\mathbf{x}, \mathbf{y}) = \prod_{(i,j) \in \eta} \mathbf{p}_{miss(t)}((\mathbf{x}, \mathbf{y}), (i, j))^{v(i,j)} \quad (3.6)$$

where, $t = 1$ to L , (\mathbf{x}, \mathbf{y}) is the detection point and (i, j) is the sensor point. Equation (3.6) is a non-linear one and it will increase the calculative complexities if we use this for further mathematical derivatives. Hence, we use natural logarithm to convert equation (3.6) into a linear one which is

$$\mathbf{pm}_{overall(t)}(\mathbf{x}, \mathbf{y}) = \sum_{(i,j) \in \eta} v(i, j) \ln \mathbf{p}_{miss(t)}((\mathbf{x}, \mathbf{y}), (i, j)). \quad (3.7)$$

where, $t = 1$ to L , (\mathbf{x}, \mathbf{y}) is the detection point and (i, j) is the sensor point.

Now we define a function,

$$\mathbf{a}_{(t)}(\mathbf{x}, \mathbf{y}) = \begin{cases} \ln \mathbf{p}_{miss(t)}((\mathbf{x}, \mathbf{y}), (i, j)), & \text{if } d((\mathbf{x}, \mathbf{y}), (i, j)) \leq r_d(t) \\ \mathbf{0}, & \text{if } d((\mathbf{x}, \mathbf{y}), (i, j)) > r_d(t) \end{cases}.$$

The deployment vector \mathbf{v} is of dimension $(N_x N_y \times 1)$. The $((i - 1) \times N_y + j)^{\text{th}}$ point of \mathbf{v} is the relative to (i, j) point in the operational grid. Whereas, the dimension of $\mathbf{a}_{(t)}$ matrix is $(N_x N_y \times N_x N_y)$. $\mathbf{a}_{(t)}$ matrix can be expressed as

$\mathbf{a}_{(t)}(\mathbf{row}, \mathbf{column})$, where $\mathbf{row} = (x-1)N_y + y$ and $\mathbf{column} = (i-1)N_y + j$. There can be as many types of \mathbf{a} matrices as there are types of sensors. We propose an algorithm to choose the best possible \mathbf{A} matrix from $\mathbf{a}_{(t)}$ matrices and that is shown in *Algorithm 1*.

Algorithm 1 : Proposed algorithm for Choosing best \mathbf{A} matrix

1. ***Input*** : $\mathbf{a}\{t\}$
- where, $t = 1$ to L
2. ***Output*** : \mathbf{A}
3. ***for*** $x = 1$ to length $(N_x \times N_y)$ do
4. ***for*** $y = 1$ to length $(N_x \times N_y)$ do
5. $\mathbf{A}(x, y) = 1000$.
6. ***for*** $i = L : -1 : 2$ do
7. $\mathbf{c}(x, y) = \text{MIN}(\mathbf{a}\{i\}(x, y), \mathbf{a}\{i - 1\}(x, y))$
8. $\mathbf{A}(x, y) = \text{MIN}(\mathbf{A}(x, y), \mathbf{c}(x, y))$
9. ***end for***
10. ***end for***
11. ***end for***

The overall logarithmic miss probability can be written in terms of \mathbf{A} and \mathbf{v} matrices which is

$$\mathbf{pm}_{overall} = \mathbf{A}\mathbf{v} \quad (3.8)$$

Now, for deploying optimal number of sensors in the grid, we use minimum squared error (MSE) criterion. For that, we use some empirical values as overall required detection probabilities ($\mathbf{pd}_{overall}^{req}$) of dimension $(\mathbf{1} \times N_x N_y)$ at different points of the grid depending on distance from center point on the grid which are

$$\mathbf{pd}_{overall}^{req} = \mathbf{nm}$$

where, \mathbf{nm} depends on $\mathbf{d}(\mathbf{x}, \mathbf{y})$.

Hence, overall required miss probability is

$$\mathbf{pm}_{overall}^{req} = (\ln(1 - \mathbf{pd}_{overall}^{req}))^T$$

and overall required detection probability is

$$\mathbf{pd}_{overall}^{req} = (\mathbf{pd}_{overall}^{req})^T$$

So we define MSE for deploying N sensors as

$$MSE = MIN \sum_{n=1}^N (\mathbf{pm}_{overall}^n(\mathbf{k}) - \mathbf{pm}_{overall}^{req}(\mathbf{k}))^2 \quad (3.9)$$

where, $\mathbf{p}_{detect}^n(\mathbf{k}) < \mathbf{p}_{detect}^{req}(\mathbf{k})$ at \mathbf{k}^{th} point in the grid.

By defining, $\mathbf{X}_n = \mathbf{pm}_{overall}^n - \mathbf{pm}_{overall}^{req}$, equation (3.8) can be expressed as

$$\mathbf{X}_{n+1} = \mathbf{X}_n + \mathbf{A}\mathbf{v}_n \quad (3.10)$$

where, \mathbf{v}_n is deployed at n^{th} step. Now to optimize the dynamic deployment process of sensors, we assign a cost function \mathbf{I} which itself is weighted squared error (SE) function of \mathbf{X}_n state variable. \mathbf{I} is defined as

$$\mathbf{I} = \frac{1}{2} \mathbf{X}_N^T \mathbf{Q}_f \mathbf{X}_N + \frac{1}{2} \sum_{n=0}^{N-1} (\mathbf{X}_n^T \mathbf{Q} \mathbf{X}_n + \mathbf{v}_n^T \mathbf{R} \mathbf{v}_n) \quad (3.11)$$

Here, \mathbf{R} , \mathbf{Q} and \mathbf{Q}_f are non negative definite diagonal matrices and their dimensions are $N_x N_y \times N_x N_y$. \mathbf{R} and \mathbf{Q} are called designer's parameters.

We use,

$$\mathbf{R}(j, j) = \left(\frac{\mathbf{pm}_{overall}^{req}(j)}{\sum_{j=1 \text{ to } N_x N_y} (\mathbf{pm}_{overall}^{req}(j))} \right)^{-1}$$

and

$$\mathbf{Q}(j, j) = (\mathbf{R}(j, j))^{-1}$$

and $\mathbf{Q}_f = \mathbf{Q}_n$ to simplify the model because those values satisfy the requirement of being non negative definite diagonal weighting matrix. Now, by using above \mathbf{R} , \mathbf{Q} , \mathbf{Q}_f and MSE function (equation (3.9)), we can minimize cost function \mathbf{I} . Hence, the dynamic transformation of MSE function from equation (3.9) is

$$ESE = MIN \sum_{n=1}^N \mathbf{X}_n(\mathbf{k})^2. \quad (3.12)$$

where, $\mathbf{p}_{detect}^n(\mathbf{k}) < \mathbf{p}_{detect}^{req}(\mathbf{k})$ at \mathbf{k}^{th} point in the grid.

We call it as the effective SE or ESE function. Since we are using a squared function, we can nullify the problem of the function being positive or negative. The result is always positive and this function detects the points where detection probability is under-satisfactory (i.e., negative \mathbf{X}_n value) or exceeding (i.e., positive \mathbf{X}_n value) the satisfaction levels. However, there is a problem. Since this function punishes both positive and negative deviations, to avoid getting chastised for positive deviations(i.e, exceeding required detection probability at a particular grid point), we set zero value for ESE at those points. So the cost function \mathbf{I} is minimized at satisfied points. Except those points, after each sensor deployment, calculating ESE and minimizing it subsequently is equivalent to minimizing whole the cost function \mathbf{I} . This is a linear-quadratic regulator problem. We use dynamic optimization technique to solve the linear quadratic regulator problem. The dynamic equation (3.10) can be expressed as

$$\mathbf{X}_{n+1} = \mathbf{f}(\mathbf{x}_n, \mathbf{v}_n), \quad n = 0, 1, \dots, N - 1. \quad (3.13)$$

With an initial condition $\mathbf{X}_0 = \mathbf{X}_{in}$, we can write the cost function as

$$\mathbf{I} = L(\mathbf{X}_N) + \sum_{n=1}^{N-1} V_n(\mathbf{X}_n, \mathbf{v}_n). \quad (3.14)$$

Now, we can use Hamiltonian function from equation (2.8) to derive discrete KKT conditions [35], [36], [37] given below:

$$\mathbf{X}_{n+1} = \mathbf{f}(\mathbf{X}_n, \mathbf{v}_n) \quad (3.15)$$

$$\mathbf{X}_0 = \mathbf{X}_{in} \quad (3.16)$$

$$\lambda_n = \nabla_{\mathbf{X}_n}^T f(\mathbf{X}_n, \mathbf{v}_n) \lambda_{n+1} + \nabla_{\mathbf{X}_n} V_n \quad (3.17)$$

$$\lambda_N = \nabla_{\mathbf{X}_N} L \quad (3.18)$$

$$\mathbf{0} = \nabla_{\mathbf{v}_n} V_n + \lambda_{n+1}^T \nabla_{\mathbf{v}_n} f(\mathbf{X}_n, \mathbf{v}_n). \quad (3.19)$$

We re-write Hamiltonian function [35] with regards to LQR as

$$H_n = \frac{1}{2} \mathbf{X}_n^T \mathbf{Q} \mathbf{X}_n + \frac{1}{2} \mathbf{v}_n^T \mathbf{R} \mathbf{v}_n + \lambda_{n+1}^T (\mathbf{X}_n + \mathbf{A} \mathbf{v}_n)$$

Now applying conditions from equations (3.15-3.19) on Hamiltonian function, we can find the following optimality conditions

$$\mathbf{X}_{n+1} = \mathbf{X}_n + \mathbf{A} \mathbf{v}_n \quad (3.20)$$

$$\mathbf{X}_0 = \mathbf{X}_{in} \quad (3.21)$$

$$\lambda_n = \lambda_{n+1} + \mathbf{Q} \mathbf{X}_n \quad (3.22)$$

$$\lambda_N = \mathbf{Q}_f \mathbf{X}_N \quad (3.23)$$

$$\mathbf{v}_n = -\mathbf{R}^{-1} \mathbf{A}^T \lambda_{n+1} \quad (3.24)$$

where, \mathbf{Q} and \mathbf{R} are design parameters.

Comparing equation (3.21) to equation (2.18), we get the values of parameters, $\mathbf{C} = \mathbf{1}$ and $\mathbf{D} = \mathbf{A}$. Now, to find optimal \mathbf{v}_n , we have to use sweep method [36].

Substituting the values of \mathbf{C} and \mathbf{D} into equations (2.22-2.29), we get

$$\lambda_n = P_n \mathbf{X}_n \quad (3.25)$$

Where,

$$P_n = P_{n+1} - P_{n+1} \mathbf{A} S_n^{-1} \mathbf{A}^T P_{n+1} + \mathbf{Q} \quad (3.26)$$

$$S_n = \mathbf{R} + \mathbf{A}^T P_{n+1} \mathbf{A} \quad (3.27)$$

$$\mathbf{G}_n = S_n^{-1} \mathbf{A}^T P_{n+1} \quad (3.28)$$

So, \mathbf{v}_n can be solved from \mathbf{G}_n and \mathbf{X}_n as

$$\mathbf{v}_n = -\mathbf{G}_n \mathbf{X}_n \quad (3.29)$$

We calculate \mathbf{S}_k , \mathbf{G}_k and \mathbf{P}_k in reverse order from $n = N - 1$ to $n = 0$. The initial steps of sweep method [36] include the final conditions for KKT [35], [36], [37] equations as $\mathbf{P}_n = \mathbf{Q}_f$, $\mathbf{G}_n = \mathbf{0}$ and $\mathbf{S}_n = \mathbf{0}$. The dimension of matrices \mathbf{P}_n , \mathbf{S}_n and \mathbf{G}_n is equal and it is $N_x N_y \times N_x N_y$. The steps of sweep method [36] are given in *Algorithm 2*.

Algorithm 2 : Algorithm for Sweep method [36]

1. **Inputs** : $\mathbf{A}, \mathbf{Q}, \mathbf{R}$
 2. **Output** : *Gain Matrix* \mathbf{G}_n
 3. **Initialize** $\mathbf{P}_n = \mathbf{Q}_f, \mathbf{G}_n = \mathbf{0}$ and $\mathbf{S}_n = \mathbf{0}$
 4. **for** $n = N - 1$ to 0 do
 5. **Calculate** \mathbf{S}_n from equation (3.27)
 6. **Calculate** \mathbf{G}_n from equation (3.28)
 7. **Save** \mathbf{G}_n .
 8. **Calculate** \mathbf{P}_n from equation (3.26)
 9. **end for**
 10. **for** $n = 0$ to $N - 1$ do
 11. **Calculate** \mathbf{v}_n from equation (3.29)
 12. **end for**
-

Though LQR method is used for minimizing the ESE function which is equivalent to minimizing cost function \mathbf{I} , we can actually go a step forward by stating an

algorithm to calculate the number of sensors and their corresponding coordinates needed for each step of deployment to minimize the ESE from equation (3.12) at unsatisfied points. The deployment method is given in **Algorithm 3**.

Algorithm 3 : Algorithm for deployment method

1. **Inputs** : detection requirement ($pd_{overall}^{req}$), number of sensors (N) and A
2. **Output** : deployment vector (v_n)
3. **Global Variables** : deployment variable (v_n) of dimension ($N_x N_y \times 1$), overall detection probability at n^{th} stage ($pd_{overall}^n$) of dimension ($N_x N_y \times 1$) and overall miss probability at n^{th} stage ($pm_{overall}^n$) of dimension ($N_x N_y \times 1$)
4. **Initialize** $n = 0$, $v_n = 0$, $pd_{overall}^n = 0$ and $pm_{overall}^n = 0$
5. **Calculate** $X_n = pm_{overall}^n - pm_{overall}^{req}$
6. **while** ($n < N$) **or** $ESE \neq 0$ **do**
7. **Find** the set of the satisfactory points i in the grid
(i.e., $i : pd_{overall(i)}^n \geq pd_{overall(i)}^{req}$)
8. **Set** $X_n(i) = 0$ (i.e., The satisfactory points are taken out of calculation)
9. **Calculate** $v_n = -G_n X_n$ from equation (3.29)
10. **Set** $j_{max} = MAX_{Index}(v_n)$ (i.e., j_{max} is the value of index, respective to which v_n is the largest)
11. **Set** $v(j_{max}) = 1$
12. **Calculate** $pm_{overall}^n = Av$
13. **Calculate** $pd_{overall}^n = 1 - e^{pm_{overall}^n}$
14. **Calculate** $X_n = pm_{overall}^n - pm_{overall}^{req}$
15. **Set** $n = n + 1$
16. **end while**
17. **if** $v_n(i) == 1$
18. **for** $i = 1$ to $length(v)$

19. *Calculate*

$$\mathbf{X}(n) = \text{ceil}\left(\frac{i}{N_y}\right) \text{ and } \mathbf{Y}(n) = i - (\mathbf{X}(n) - 1)N_y$$

20. *Discard* the corresponding indexes of $\mathbf{X}(n)$ and $\mathbf{Y}(n)$ where one or both of their values are zero.

21. *end if*

22. *end for*

The 19th line gives us the coordinates of the deployed sensors. In the beginning of the chapter, we have described that the deployment vector \mathbf{v} is of dimension $(N_x N_y \times 1)$ and $((i - 1) \times N_y + j)$ th point of \mathbf{v} is the relative to (i, j) point in the operational grid. The \mathbf{X} and \mathbf{Y} coordinates are calculated by taking into consideration of the dimension of \mathbf{v} and the its relation to the operation grid. In case of \mathbf{X} coordinate calculation, if we divide the index of \mathbf{v} by N_y and take the nearest positive round number by applying ceiling function over it, we get the dimension N_x for \mathbf{X} . This is actually correct. For \mathbf{Y} coordinate calculation, we use the above relation between \mathbf{v} and operation grid as, $\text{Index}(\mathbf{v}) = (\mathbf{X} - 1) \times N_y + \mathbf{Y}$ and then reshape it to get $\mathbf{Y} = \text{Index}(\mathbf{v}) - (\mathbf{X} - 1) \times N_y$. That are the initial coordinates. Next, we discard all the zero values from those and get the final coordinates of sensor deployment.

Now then, we get all the coordinates of sensors to deploy in the operation grid. However, there arises a new challenge about which point would contain which type of sensor. For that we calculate Euclidean distance of every pair of grid points where sensors have been placed. For example, there are two points $\mathbf{m}(\mathbf{X}(\mathbf{m}), \mathbf{Y}(\mathbf{m}))$ and $\mathbf{i}(\mathbf{X}(\mathbf{i}), \mathbf{Y}(\mathbf{i}))$ where \mathbf{m} is coordinate of the 1st sensor and \mathbf{i} is the coordinate for

the 2^{nd} sensor. Henceforth, from 1^{st} sensor's point of view, it sees the Euclidean distance from 2^{nd} sensor as

$$d(\mathbf{X}, \mathbf{Y}) = \sqrt{(\mathbf{X}(\mathbf{m}) - \mathbf{X}(\mathbf{i}))^2 + (\mathbf{Y}(\mathbf{m}) - \mathbf{Y}(\mathbf{i}))^2} \quad (3.30)$$

Then, the minimum Euclidean distance is taken keeping in mind that the result is discarded when $\mathbf{m} = \mathbf{i}$ since the result is always zero at this case as we are taking the same point twice to calculate the distance. Now, since heterogeneous sensors are used according to their sensing ranges, we can actually sort out the sensors like $r_{d1} > r_{d2} > r_{d3} > \dots > r_{dL}$. Meanwhile, from each minimum distance calculated for each sensor, there can be at best three cases. Firstly, the distance falls in between a sensor range bracket of an upper and a lower value. Secondly, the distance is lower than or equal to the lowest sensor range type. Finally, the distance is higher than the highest sensor range type. From this concept, the sensor type at each point can easily be determined. The proposed process is summarized in *Algorithm 4*.

Algorithm 4 : Proposed algorithm for choice of sensor type for each occupied point

1. **Inputs** : \mathbf{X} and \mathbf{Y}
 2. **Outputs** : M , \mathbf{X}_f and \mathbf{Y}_f
 3. **Initialize** $M = \{[1], [1], [1]\}$
 4. **for** $m = 1 : \text{length}(\mathbf{X})$
 5. **Set** $c = \text{MIN}(d(m, :))$
 6. **Set** $i = L$
- where, L is types of sensors
7. **while** ($i \neq 0$)

```

8.   if ( $c > \text{rad}(i)$ )
9.     Set  $i = i - 1$ 
10.  else
11.    Set  $X_f\{i\}(1, M\{i\}) = X(1, m)$ 
12.    Set  $Y_f\{i\}(1, M\{i\}) = Y(1, m)$ 
13.    Set  $M\{i\} = M\{i\} + 1$ 
14.    Break
15.  end if
16.  end while
17.  if  $i == 0$  do
18.    Set  $X_f\{1\}(1, M\{1\}) = X(1, m)$ 
19.    Set  $Y_f\{1\}(1, M\{1\}) = Y(1, m)$ 
20.    Set  $j = 1$ 
21.    Set  $M\{j\} = M\{j\} + 1$ 
22.  end if
23. end for
24. for  $i = 1 : L$ 
25.   Set  $M\{i\} = M\{i\} - 1$ 
26. end for

```

3.2 System Model Alteration Due To Change of Parameters

In this section, we discuss about two features added on the initial system model to make it more realistic. The first one is effect of obstacles on the operation grid and second one is the uncertainty of sensor deployment on the intended place.

3.2.1 Effects of Obstacles

If there remain one or more obstacles in the terrain, this will affect the outcome of the initial sensor deployment model.

First of all, if we put a sensor in front of an obstacle, it will not detect the points straight in front of the obstacle. In the algorithm, we consider every obstacle as a polygon which has edge points. Henceforth, we calculate the angular distances of the edge points of the obstacle from each sensor point. Then the difference between highest and lowest angle is considered as deleted angular range for each sensor and we call a LOS (line of sight) between a detection point and a sensor point if there is no such deleted angular range or the detection point is not inside the obstacle's edge points.

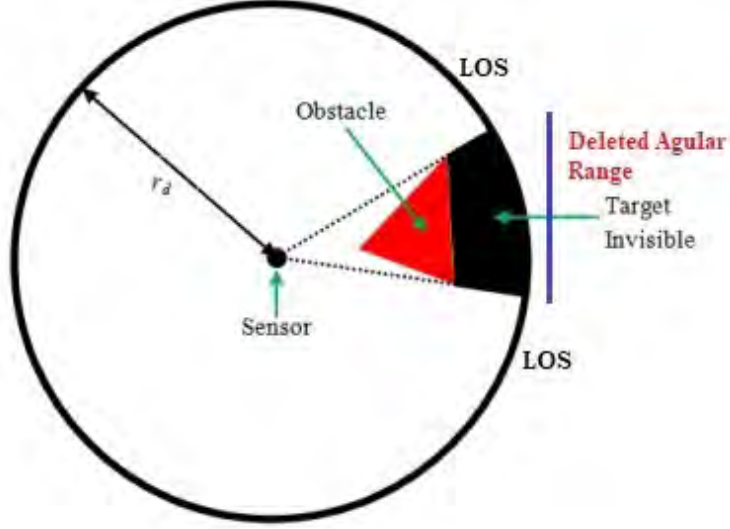


Figure 3.1: Presence of obstacle in the operation grid

Figure 3.1 shows the placement of sensor, obstacle, LOS, deleted angular range and the portion where target can be invisible. If there is no LOS, we call the case inaccessible and in that case the $A(\text{row}, \text{column}) = 0$, $pd_{overall}^{req}(\text{inaccessible}) = 0$, $Q(\text{inaccessible}) = Q_f(\text{inaccessible}) = 0$, $v_n(\text{inaccessible}) = 0$ and $R(\text{inaccessible})$ is set to a very large value. The algorithm is described below.

Modification of Algorithm due to the presence of obstacles:

1. *if* sensor point (i, j) is inaccessible or detection point (x, y) is inaccessible
Set $A(\text{row}, \text{column}) = 0$
2. *else if* $d(x, y) \leq r_d$ and LOS is present between (x, y) and (i, j) then A will follow *Algorithm 1*
3. *else Set* $A(\text{row}, \text{column}) = 0$
4. *end if*
5. **Set** $pd_{overall}^{req}(\text{inaccessible}) = 0$
6. **Set** $Q(\text{inaccessible}) = Q_f(\text{inaccessible}) = 0$
7. **Set** $R(\text{inaccessible})$ to a very large positive value

8. *Set* $v_n(\textit{inaccessible}) = 0$
 9. *Keep* rest of the algorithm same.
-

3.2.2 Effects of Uncertainty of Deployment

Uncertainty plays a vital role in deployment method. For example, if the sensors are to be deployed from airplane, then it is possible that the sensors will not be deployed in the exactly intended positions but a slight deviation will occur and hence there will be difference in the outcome from theoretical result. In this thesis, we consider this deviation due to uncertainty of sensor deployment as a Gaussian probability distribution model [40]. The joint probability density function for uncertainty can be written as

$$P_G(\mathbf{m}, \mathbf{n}) = \frac{e^{-\frac{(m-i)^2}{2\sigma_i^2} - \frac{(n-j)^2}{2\sigma_j^2}}}{2\pi\sigma_i\sigma_j}. \quad (3.31)$$

where, (i, j) is the mean value which is the sensor placing point, (\mathbf{m}, \mathbf{n}) is the coordinate of the operation grid and σ_i and σ_j are the standard deviations in \mathbf{X} and \mathbf{Y} directions which are independent from each other. Hence, the modified \mathbf{A} matrix will be

$$\mathbf{A}_{\textit{modified}}(\textit{row}, \textit{column}) = \frac{\sum_{(m,n) \in \eta} \mathbf{A}(\textit{row}, \textit{column}) P_G(\mathbf{m}, \mathbf{n})}{\sum_{(m,n) \in \eta} P_G(\mathbf{m}, \mathbf{n})} \quad (3.32)$$

We keep the rest of the algorithms exactly same as before.

So to summarize this chapter, it can be said that we have stated four algorithms in a sequential order to model our problem as a LQR problem and solve it accordingly. Specifically at first, we propose an algorithm (**Algorithm 1**) to find the best possible \mathbf{A} matrix to get the minimum logarithmic overall achieved miss probability at each point of the grid for the whole combination of sensors rather than choosing one for each type of sensors and then try to solve all of those at a time. In that

way we are able to comply with the same method that we can use on a homogeneous sensor deployment as well as in a heterogeneous sensor deployment without broadening the complexity of calculation. Finally, we propose another algorithm (*Algorithm 4*) to determine the type of sensors at each occupied point. Although one might think that to determine the type of sensors after deployment is a bit odd, this is in fact a smart way to reduce the complexity and volume of calculation especially in case of heterogeneous deployment and since the knowledge of the types of sensors in the occupied points does not necessarily change the deployment points in the grid to achieve optimization, this sequence serves the purpose perfectly. Additionally, we have added two extra features: obstacles and uncertainty of deployment as augmentations of our original generalized system model to make it more pragmatic for deploying in the real scenarios. The simplified block diagram of proposed deployment process is given in Figure 3.2.

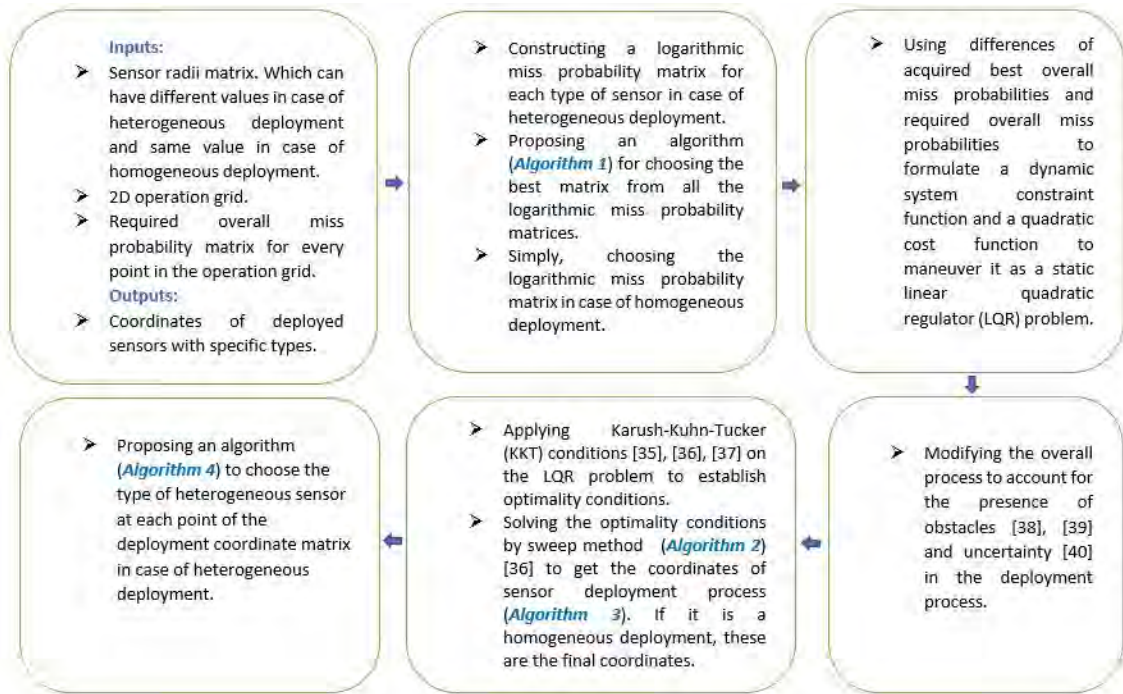


Figure 3.2: Block diagram of the proposed deployment process

Chapter 4

Simulations and Results

In this chapter, we evaluate the performances of the proposed algorithms by simulating it in MATLAB. In simulation set-up we construct 3 cases.

Case 1: We choose a heterogeneous combination of three type sensors and compare their combined performances with the homogeneous deployment of each of the three type of sensors individually by varying the attenuation factor and grid-matrix size. Then, we calculate the difference between achieved and required probability for each point and analyze whether or not it suffices our expectations. Though it is sufficient from this step to declare if the proposed algorithm holds or not, we still want to troubleshoot the process in a simpler manner by target injection. So we inject random targets in the operation grid and try to find out if one or more sensors can actually detect each of the targets regardless of its positions.

Case 2: We do the same thing as in Case 1 with the inclusion of obstacles.

Case 3: We repeat the process as in Case 1 with the addition of uncertainty of deployment process.

Now we elaborate these 3 cases below.

4.1 Case 1: Simulation without Obstacles or Uncertainty

4.1.1 Simulation Parameters

Here, for our problem we select three types of sensors according to their sensing radii. The radius matrix \mathbf{r}_d is [8 4 3] and we set attenuation factor α as 0.10, 0.15 and 0.20, respectively. We consider two grids (η) of size [20 \times 20] and [40 \times 40], consecutively. Figure 4.1 shows the required detection probabilities for the operation grid. In our thesis, the operation grid and sensor radius are unitless and it is immaterial that whatever unit we choose, our system model will always hold true irrespective of that. However, in case of practical sensors we have studied that in real world proximity sensors are of mainly two types: IR sensors and ultrasound sensors. Long range IR sensors can have range from 1-5 m and long ultrasound sensors can have range from 10-50 m for target detection depending on price. So, if we take an operation grid of 100 $m \times$ 100 m and take 10 m radius ultrasound sensors, we can choose 1 unit = 1 m for our case. Then the field will be 100 unit \times 100 unit and each sensor has a 10 unit sensing radius. The sensing radii matrix is constructed based on real life proximity IR sensors [41] and ultrasound sensors [42]. Attenuation factors are chosen based on [34].

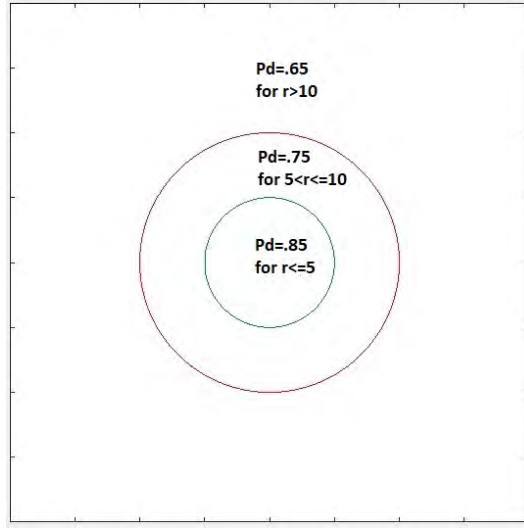


Figure 4.1: Required Detection Probabilities

4.1.2 Simulation Results

Firstly, we observe the performance of the deployment algorithm without obstacles and uncertainty into the grid and compare it with homogeneous sensor deployment [34]. Table 4.1 shows the deployment results. In Table 4.1 the first four columns show the result of our developed generalized sensor deployment framework whereas the last column shows results in Ref. [34].

Table 4.1: Simulation results for Case 1

type	α	[20 × 20]	[40 × 40]	[25 × 25]	[25 × 25] in [34]
Homogeneous	0.10	[11 28 37]	[31 102 147]	26	27
Homogeneous	0.15	[12 29 39]	[35 98 147]	32	28
Homogeneous	0.20	[15 31 39]	[44 104 152]	33	34
Heterogeneous	0.10	11 [9 0 2]	31 [31 0 0]	-	-
Heterogeneous	0.15	13 [11 2 0]	34 [30 2 2]	-	-
Heterogeneous	0.20	16 [14 0 2]	44 [38 4 2]	-	-

It is to be noted that in Table 4.1, for a given attenuation factor and grid size, rows under Homogeneous type contain information about the individual number of sensors having sensing radii of 8, 4 and 3 units respectively for homogeneous deployments whereas rows under Heterogeneous type contain information about the number of the combination of sensors for heterogeneous deployments. The fourth row under Homogeneous type describes the number of sensors of 5 unit radius needed for a grid size of [25 × 25] and fifth row gives us the number of homogeneous sensors that we attain from [34] for same grid size and sensing radius. That is how we are able to compare our results with that of [34]. From Table 4.1, we can see that when we use sensors of 8 unit radius for homogeneous sensor deployment, the result is almost same for both homogeneous and heterogeneous deployment but when we take sensors of 4 or 3 unit radius, we need a huge number of sensors for homogeneous sensor deployment compared to heterogeneous sensor deployment.

So our developed generalized sensor deployment framework is successful for both homogeneous and heterogeneous sensor deployment. Table 4.1 also shows that with increase in the value of α or grid size, number of sensors also increases as expected. Also we can observe that homogeneous sensor deployment obtained from our developed generalized sensor deployment framework very closely matches with that in [34].

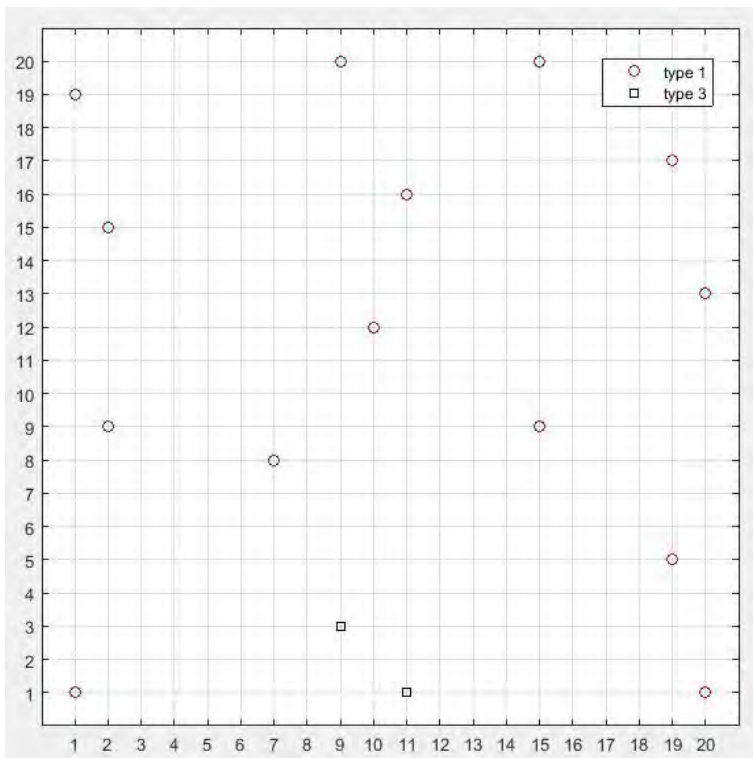


Figure 4.2: Heterogeneous sensor deployment for a $[20 \times 20]$ grid size with 30 input sensors

Figure 4.2 shows heterogeneous sensor deployment for a $[20 \times 20]$ grid size with $\alpha = 0.20$. From the figure, we can see that only sensors of 8 and 3 unit radius are required. We use 30 sensors as input and find that an output of total 16 $[14 \ 0 \ 2]$ sensors is required. Next, we change the number of input sensors to 50 to see if the optimal result of 16 sensors holds or not.

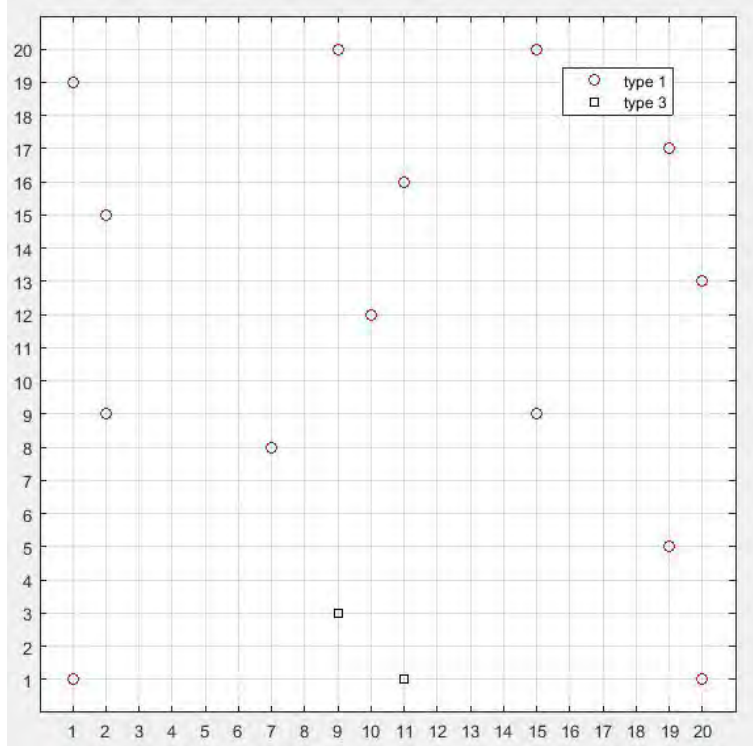


Figure 4.3: Heterogeneous sensor deployment for a $[20 \times 20]$ grid size with 50 input sensors

Figure 4.3 shows that the output for an input of 50 sensors is also the exactly same. As we can see same location and same type of total 16 $[14 \ 0 \ 2]$ sensors are deployed as before. So we can conclude that optimum result holds irrespective of number of input sensors.

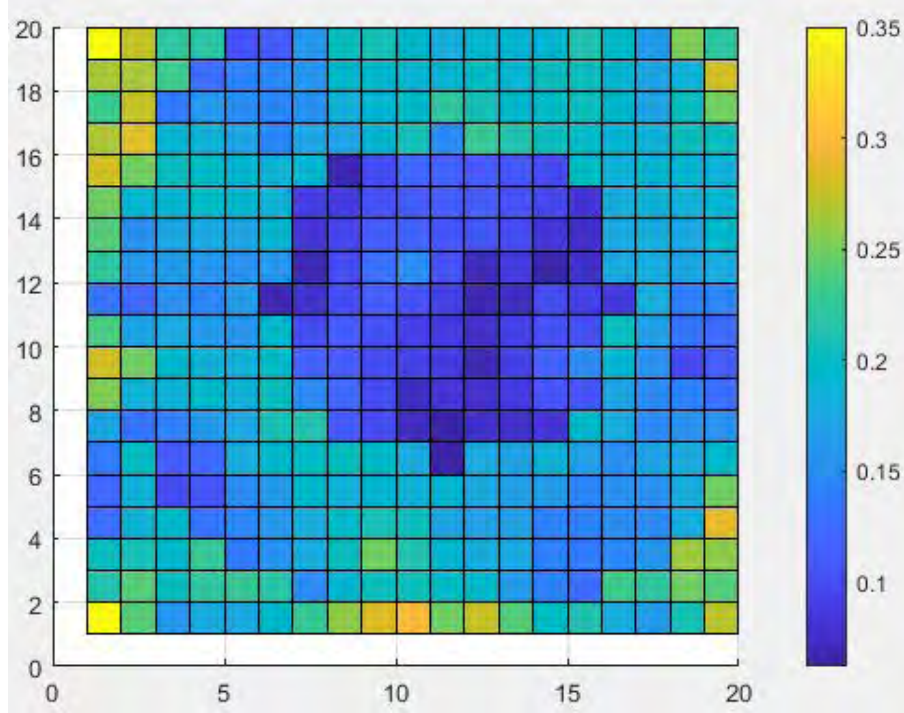


Figure 4.4: Difference between achieved and required detection probability for Heterogeneous sensor deployment for a $[20 \times 20]$ grid size

Figure 4.4 depicts the difference between required and achieved detection probability at $\alpha = 0.20$ after heterogeneous sensor deployment for a $[20 \times 20]$ grid size and we can see that the difference is positive at all points so the algorithm achieves the required probability at all points.

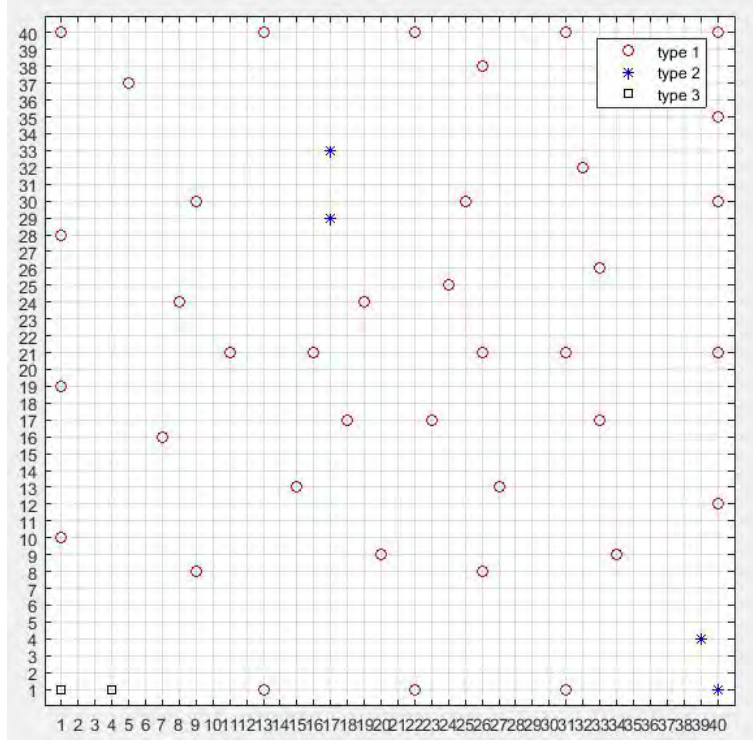


Figure 4.5: Heterogeneous sensor deployment for a $[40 \times 40]$ grid size

Figure 4.5 is for heterogeneous sensor deployment for a $[40 \times 40]$ grid size at $\alpha = 0.20$. From the figure, we can see all types of sensors (8, 4 and 3 unit radius) are required for this.

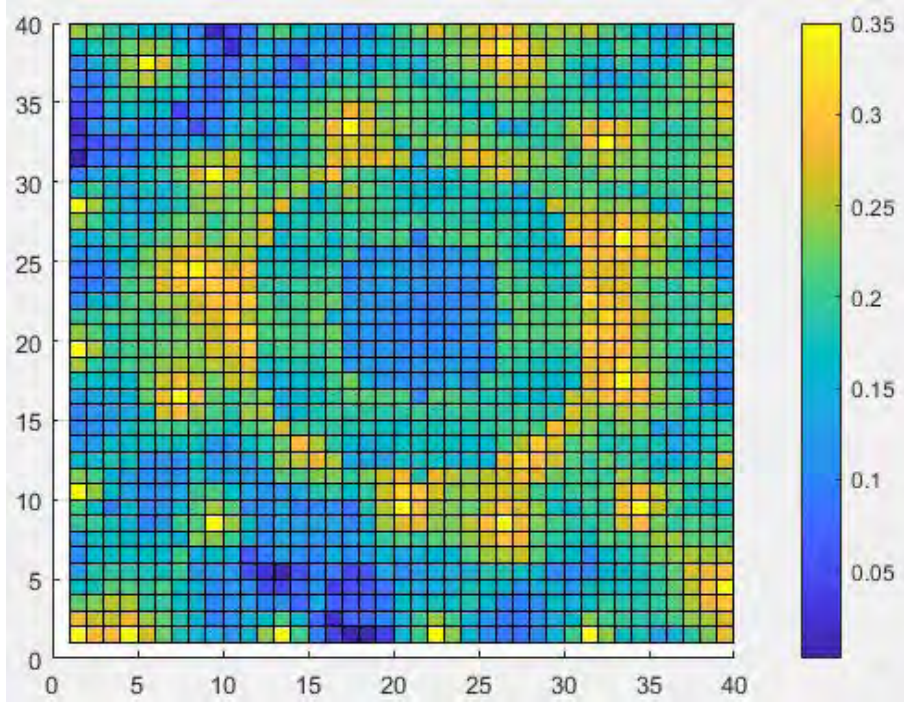


Figure 4.6: Difference between achieved and required detection probability for Heterogeneous sensor deployment for a $[40 \times 40]$ grid size

Figure 4.6 depicts the difference between required and achieved detection probability at $\alpha = \mathbf{0.20}$ after heterogeneous sensor deployment for a $[40 \times 40]$ grid size and we can see that the difference is positive at all points so the algorithm achieves the required probability at all points.

Next, we inject three different targets at random positions to figure out whether or not our proposed model can detect all of them regardless of their positions in the simulation grid.

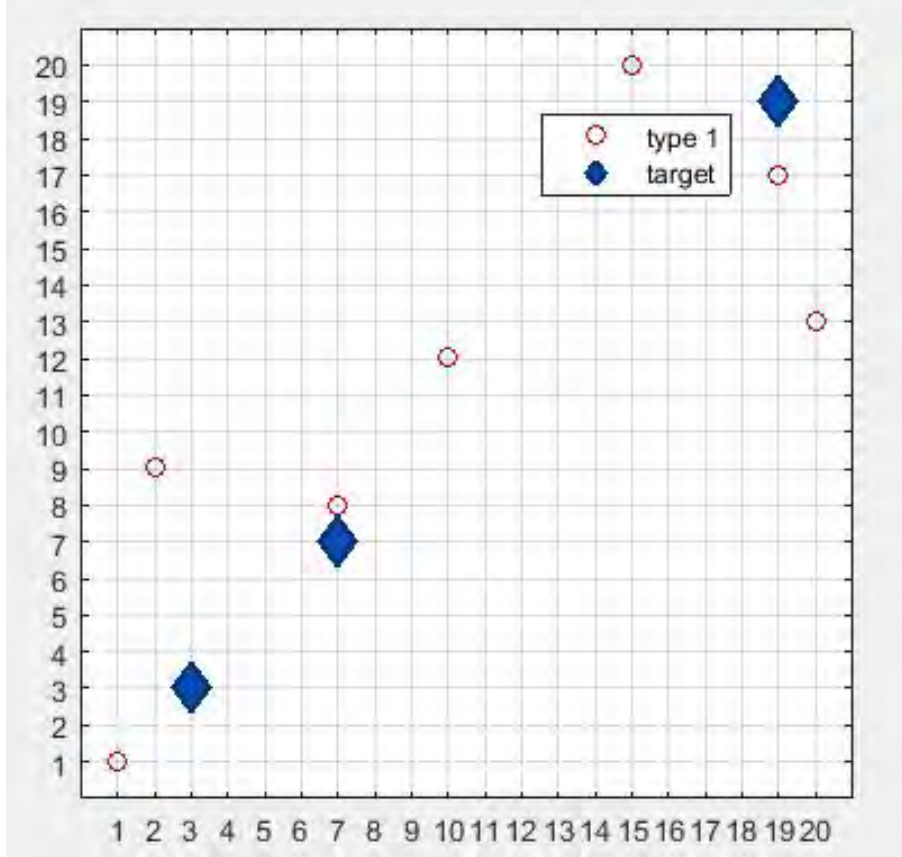


Figure 4.7: Sensors used only for injected target detection for a $[20 \times 20]$ grid size

Figure 4.7 shows that at $\alpha = 0.20$ for a grid size of $[20 \times 20]$, three targets are injected and more than one sensors can detect each of the targets. We use random coordinates for these targets. So we can say that our proposed algorithm successfully handles the detection of targets regardless of their positions. It is also to be noted here that the mobility of targets does not affect the detection probability since all the points under in operation grid are covered by the deployed sensors.

Now, we add two quantitative analysis. Firstly, we construct sensor numbers vs ESE (dynamic minimum squared error) graph for homogeneous and heterogeneous deployment and results of homogeneous deployment are compared to that of [34].

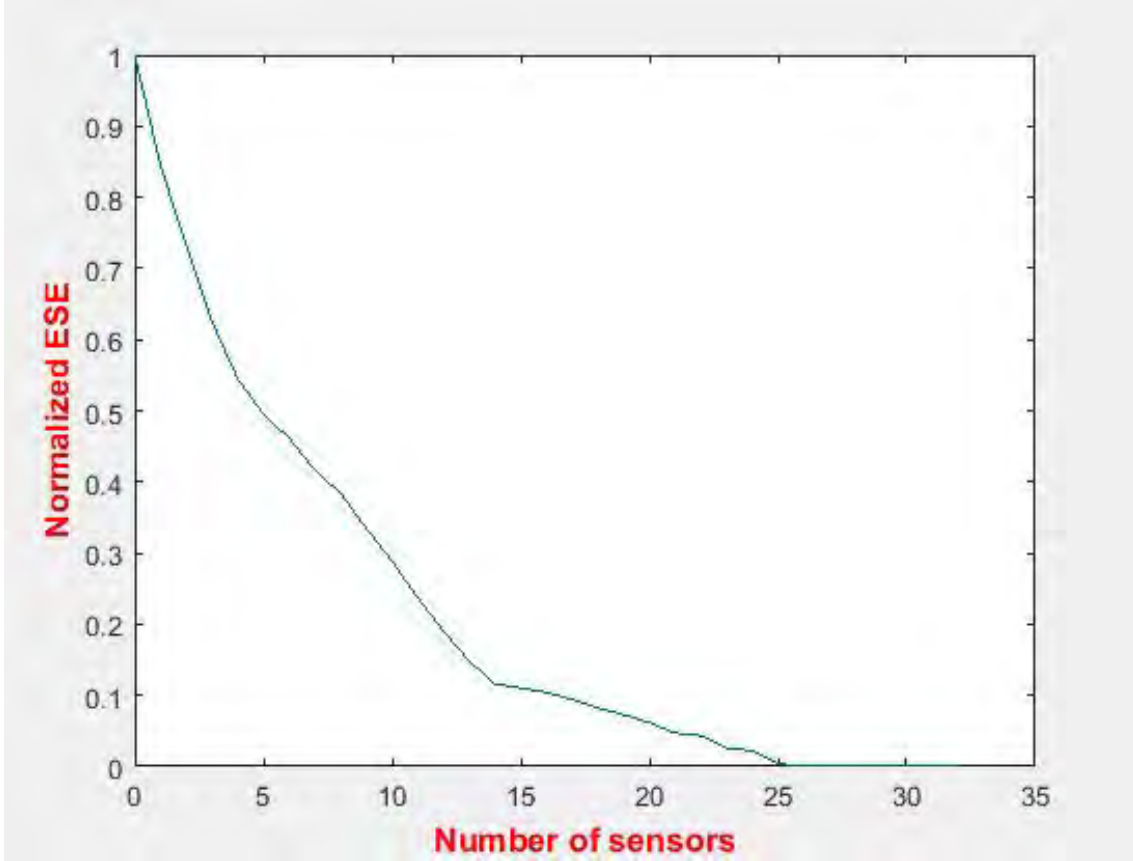


Figure 4.8: Effective SE convergence graph of homogeneous sensor deployment for a $[20 \times 20]$ grid size with $\alpha=.15$

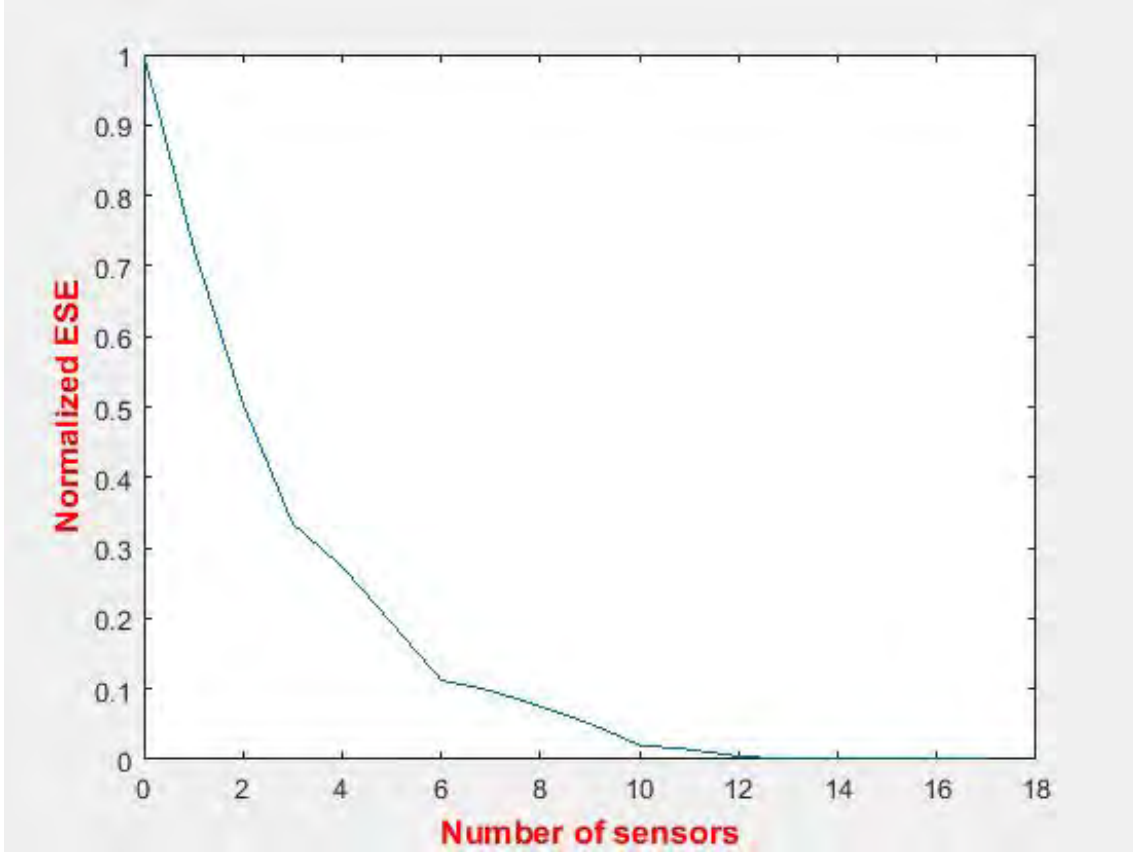


Figure 4.9: Effective SE convergence graph of heterogeneous sensor deployment for a $[20 \times 20]$ grid size with $\alpha=.15$

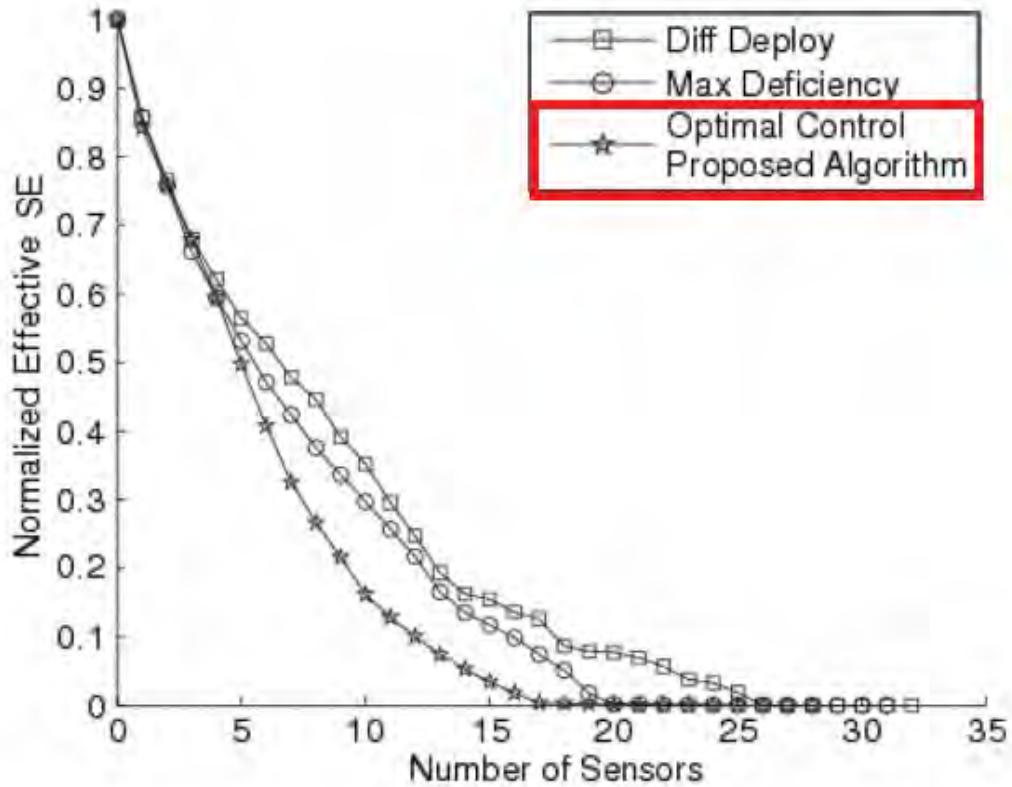


Figure 4.10: Effective SE convergence graph of homogeneous sensor deployment for a $[20 \times 20]$ grid size with $\alpha=.15$ at [34]

From Figure 4.8 and 4.10 comparing our graph with the graph of Optimal Proposed Algorithm of [34] we can palpably reach into conclusion that the results of our thesis are very close to that of [34].

Secondly, we add line graphs of grid dimensions vs sensor number for both homogeneous and heterogeneous deployments.

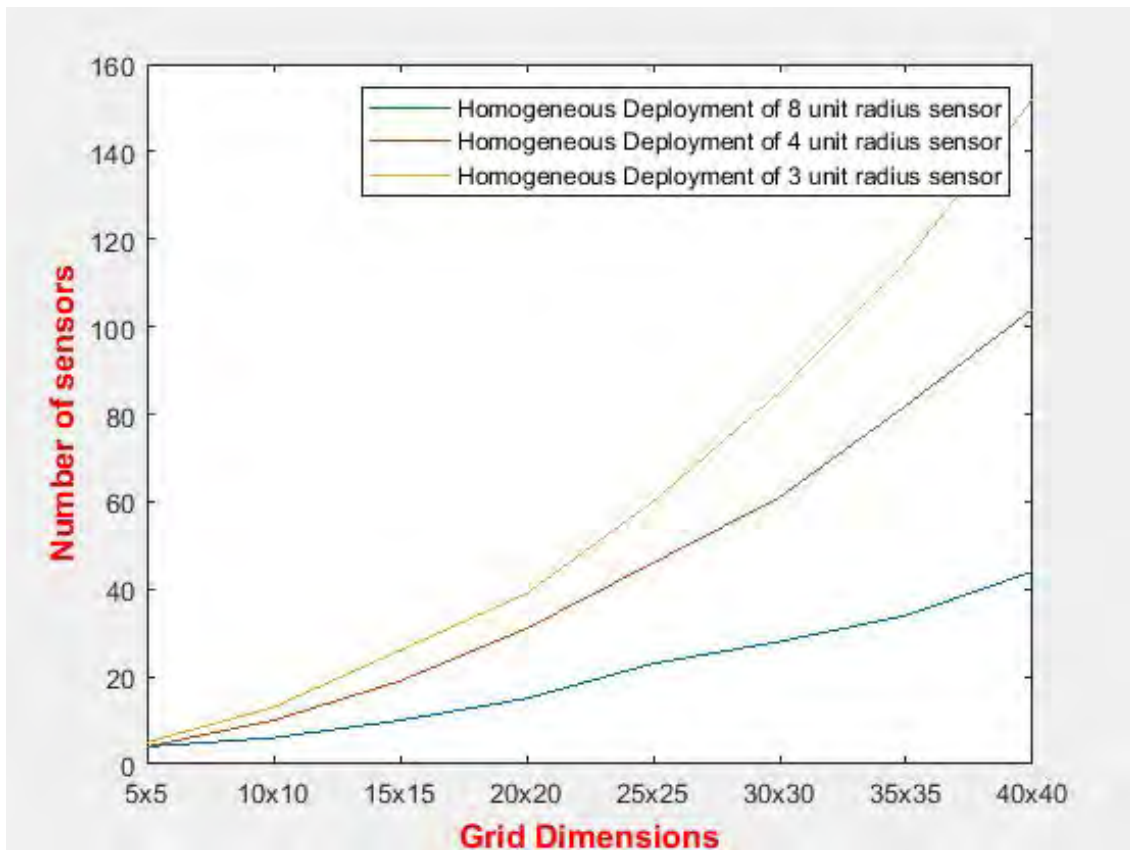


Figure 4.11: Line graph of grid dimensions vs sensor number for homogeneous deployment

Figure 4.11 shows that we need a significantly large number of smaller radius sensors and a comparatively small number of large radius sensors in case of larger operational grids. So, it signifies that it is advantageous to use small radius sensors if the operation grid is small and large radius sensors if the operation grid is large for homogeneous deployment.

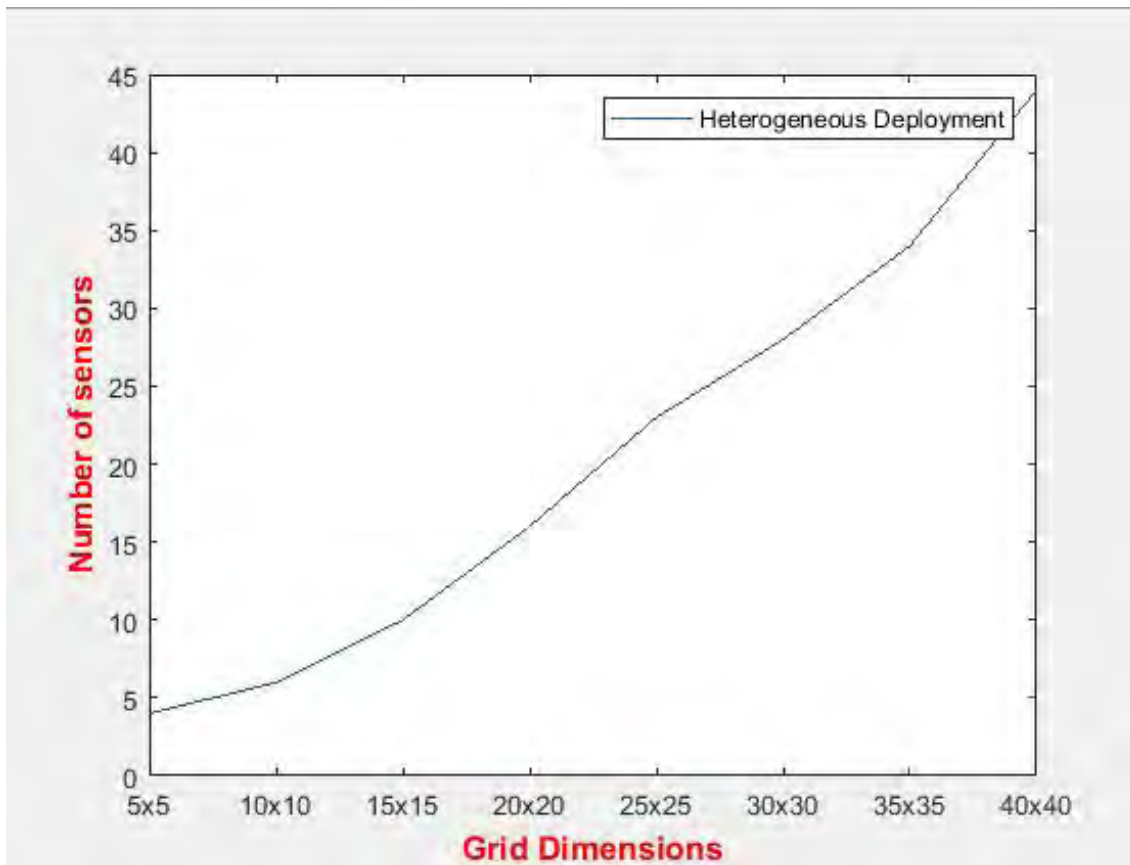


Figure 4.12: Line graph of grid dimensions vs sensor number for heterogeneous deployment

Figure 4.11 and 4.12 shows that in cases of both homogeneous and heterogeneous deployments with larger grid size we need more sensors than smaller grid size for optimal deployment as expected.

4.2 Case 2: Simulation with Obstacles

Now, we add 2 obstacles into our operation grid. One is a polygon whose coordinates are (2, 2), (6, 2), (8, 4), (4, 5), (2, 4). The other is a rectangle whose coordinates are (16, 16), (18, 16), (18, 18), (16, 18). Then the previous procedure is repeated. Table 4.2 shows the distribution of three types of sensors for $[20 \times 20]$ and $[40 \times 40]$ grid sizes.

Table 4.2: Simulation results for Case 2

type	α	$[20 \times 20]$	$[40 \times 40]$
Heterogeneous	0.10	14 [7 1 6]	35 [26 1 8]
Heterogeneous	0.15	15 [8 3 4]	38 [29 1 8]
Heterogeneous	0.20	18 [11 3 4]	47 [32 7 8]

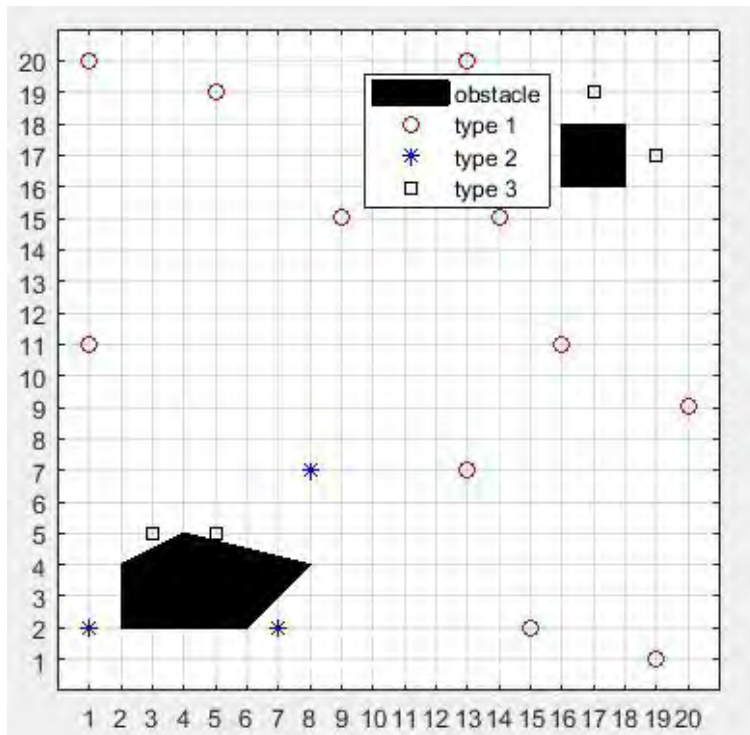


Figure 4.13: Heterogeneous sensor deployment for a $[20 \times 20]$ grid size with obstacles

Figure 4.13 for heterogeneous sensor deployment with obstacles for a grid size of $[20 \times 20]$ at $\alpha = 0.20$. From the figure, we can see all types of sensors (8, 4 and 3 unit radius) are required for this.

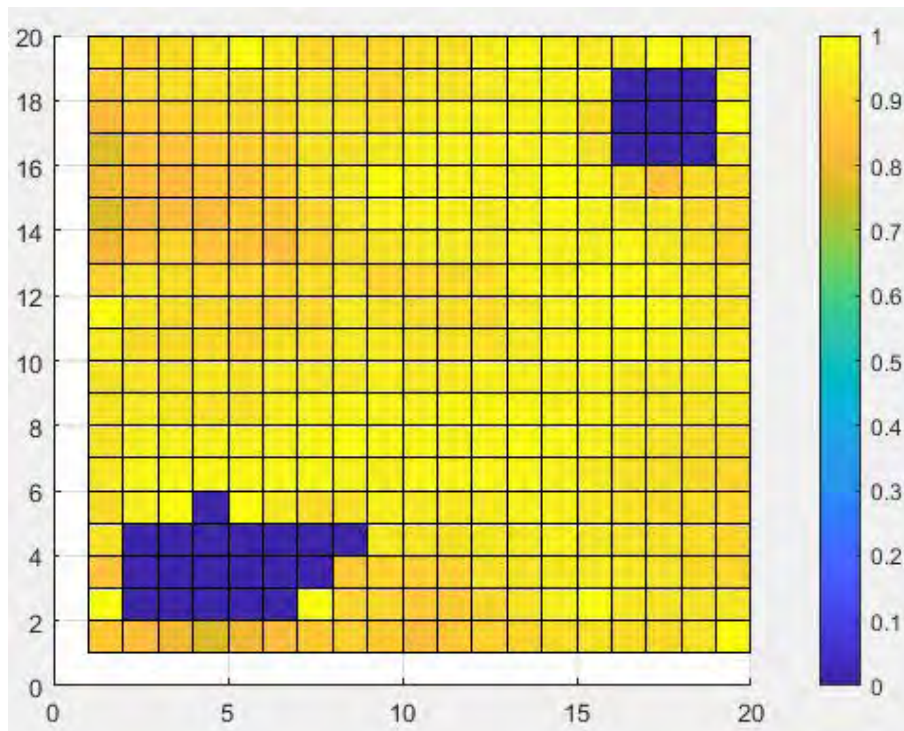


Figure 4.14: Difference between achieved and required detection probability for Heterogeneous sensor deployment for a $[20 \times 20]$ grid size with obstacles

Figure 4.14 depicts the difference between required and achieved detection probability at $\alpha = 0.20$ after heterogeneous sensor deployment of a grid size of $[20 \times 20]$ and we can see that the difference is positive at all points so the algorithm achieves the required probability at all points.

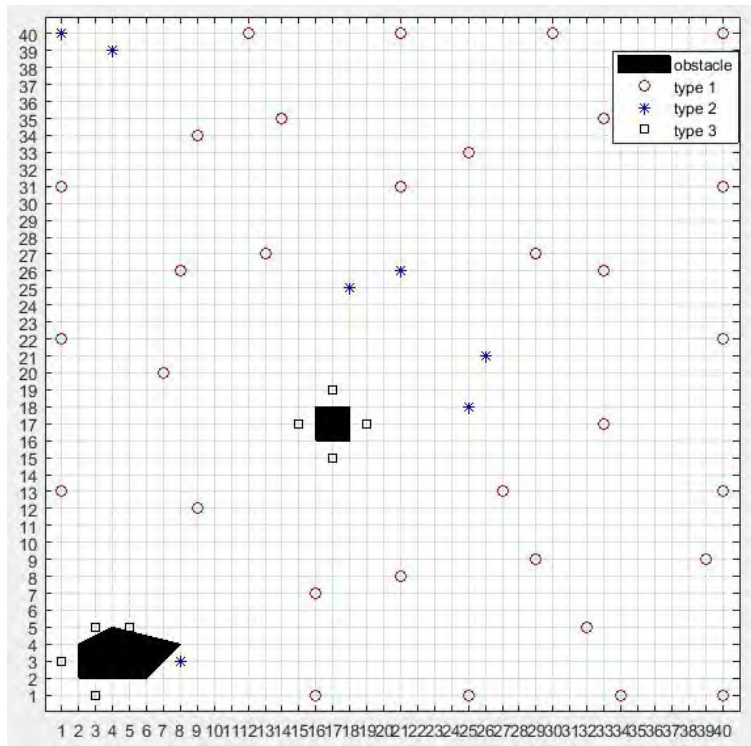


Figure 4.15: Heterogeneous sensor deployment for a $[40 \times 40]$ grid size with obstacles

Figure 4.15 is for heterogeneous sensor deployment with obstacles for a grid size of $[40 \times 40]$. From the figure, we can see all types of sensors (8, 4 and 3 unit radius) are required for this.

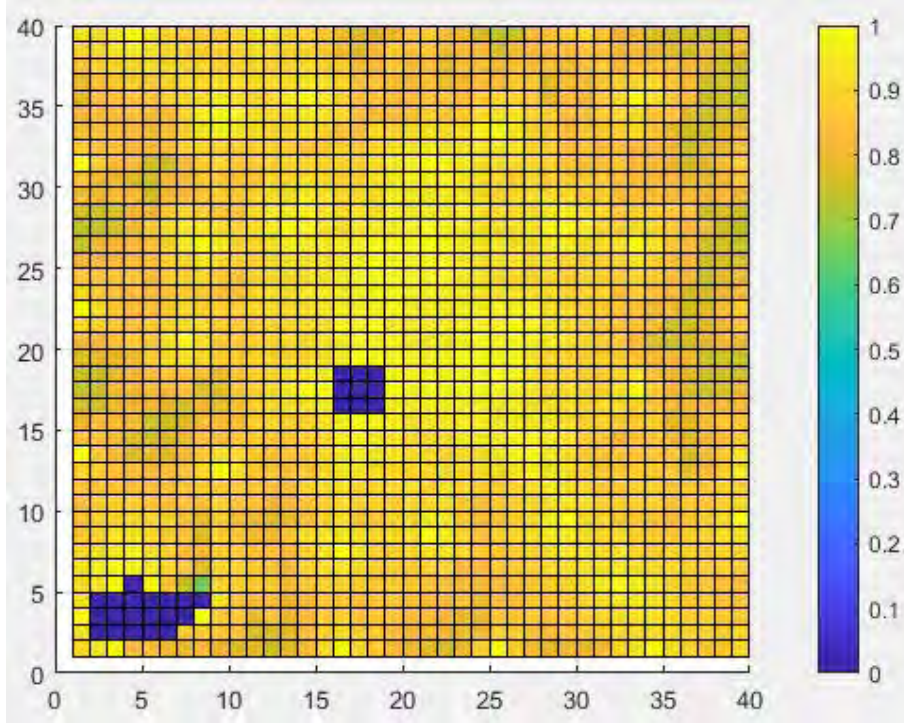


Figure 4.16: Difference between achieved and required detection probability for Heterogeneous sensor deployment of for a $[40 \times 40]$ grid size with obstacles

Figure 4.16 depicts the difference between required and achieved detection probability at $\alpha = 0.20$ after heterogeneous sensor deployment for a grid size of $[40 \times 40]$ and we can see that the difference is positive at all points so the algorithm achieves the required probability at all points.

Next, we inject three different targets at random positions to figure out whether or not our proposed model can detect all of them regardless of their positions in the simulation grid.

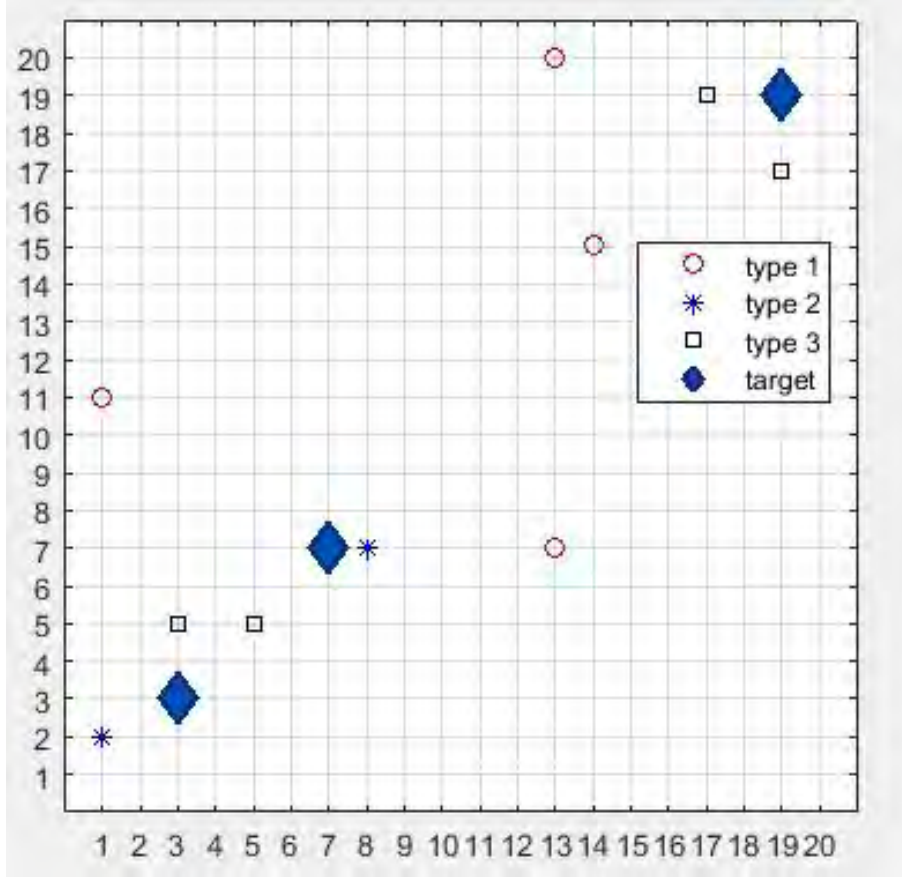


Figure 4.17: Sensors used only for injected target detection for a $[20 \times 20]$ grid size with obstacles

Figure 4.17 shows that for a grid size of $[20 \times 20]$ with the presence of two obstacles and at $\alpha = 0.20$, three targets are injected and more than one sensors can detect each of the targets. We use random coordinates for these targets. So we can say that in case of obstacles presence, our proposed algorithm successfully handles the detection of targets regardless of their positions.

4.3 Case 3: Simulation with Uncertainty

We then add uncertainty with the original heterogeneous sensor deployment model. Then the previous procedure is repeated. There are two points to be noted. Firstly, we consider that every deployed sensor is in a mistaken coordinate rather than its intended coordinate. In every point, we calculate average Gaussian probability distribution rather than its actual probability. The sensor’s capability to achieve the required detection probability decreases. We use an empirical value for

$$\textit{required detection probability} = \frac{6}{\textit{grid dimension along any axis}} \tag{4.1}$$

to get a nominal positive threshold value to model the process. Secondly, in the previous two cases, the threshold for **ESE** was set to **0** but here the **ESE** does not reach absolute **0**. We extract a nominal positive threshold value for **ESE**. We find that **ESE = 1** is the nominal threshold value that gives us good results and holds the algorithm in while-loop breaking situation [line 12 and line 13 in Algorithm for Deployment method] for optimization process. Table 4.3 shows the simulation results for [20 × 20] and [40 × 40] grid sizes.

Table 4.3: Simulation results for Case 3

type	α	[20 × 20]	[40 × 40]
Heterogeneous	0.10	8 [2 2 4]	32 [10 4 18]
Heterogeneous	0.15	11 [4 3 4]	40 [11 5 24]
Heterogeneous	0.20	12 [3 6 3]	48 [13 8 27]

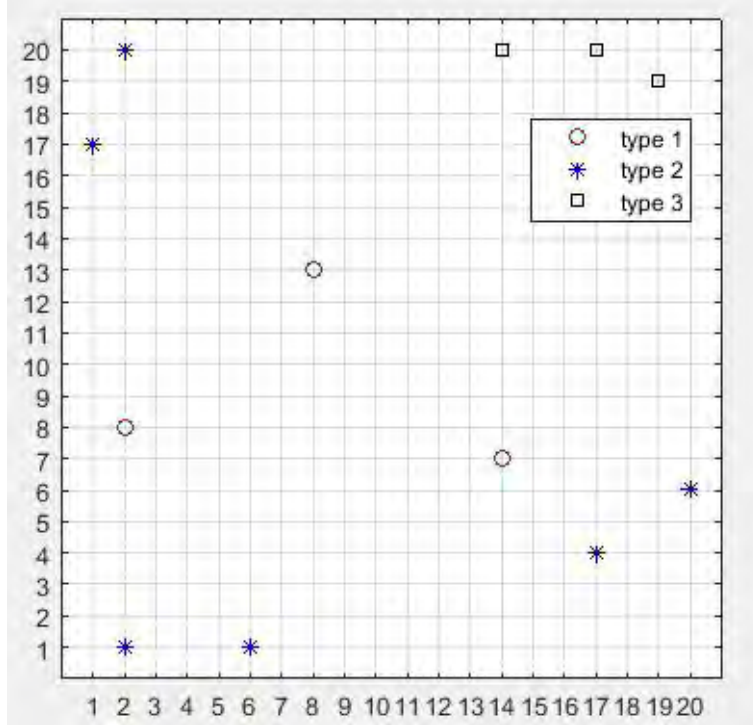


Figure 4.18: Heterogeneous sensor deployment for a $[20 \times 20]$ grid size with uncertainty

Figure 4.18 is for heterogeneous sensor deployment with uncertainty for a $[20 \times 20]$ grid size. From the figure, we can see all types of sensors (8, 4 and 3 unit radius) are required for this.

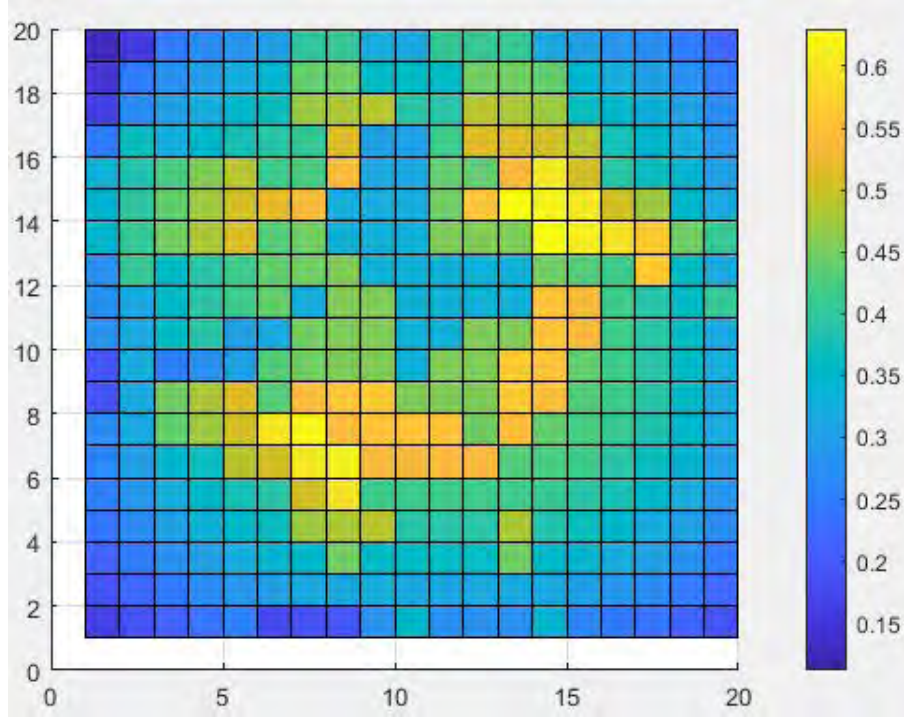


Figure 4.19: Difference between achieved and required detection probability for Heterogeneous sensor deployment for a $[20 \times 20]$ grid size with uncertainty

Figure 4.19 depicts the difference between required and achieved detection probability at $\alpha = 0.20$ after heterogeneous sensor deployment for a grid size of $[20 \times 20]$ and we can see that the difference is positive at all points so the algorithm achieves the required probability at all points.

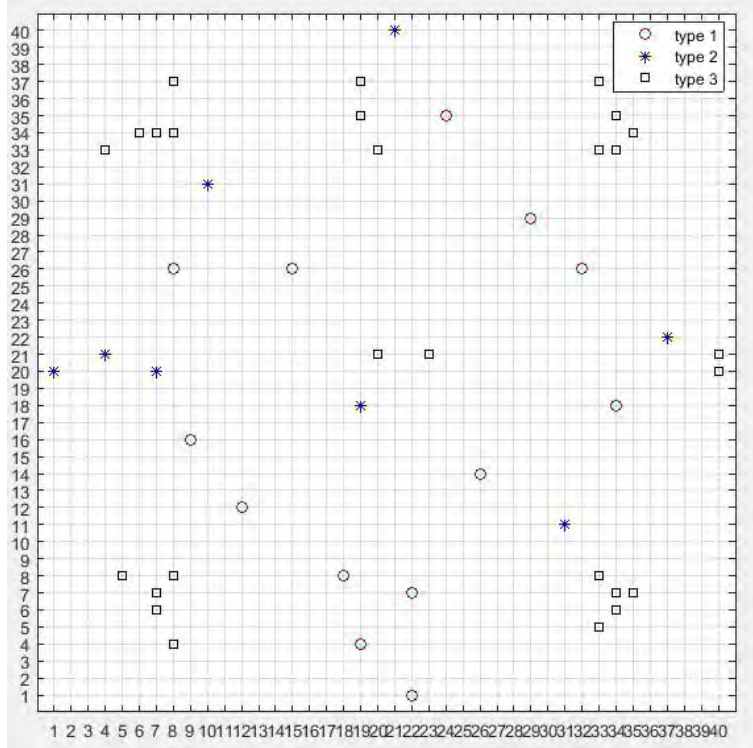


Figure 4.20: Heterogeneous sensor deployment for a $[40 \times 40]$ grid size with uncertainty

Figure 4.20 for heterogeneous sensor deployment with uncertainty for a grid size of $[40 \times 40]$. From the figure, we can see all types of sensors (8, 4 and 3 unit radius) are required for this.

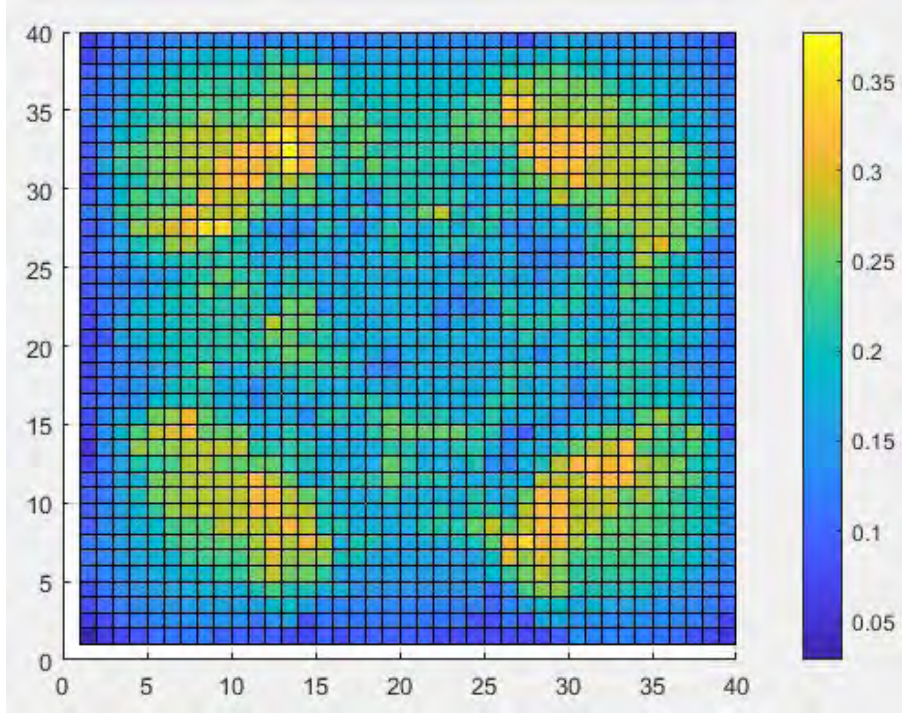


Figure 4.21: Difference between achieved and required detection probability for Heterogeneous sensor deployment for a $[40 \times 40]$ grid size with uncertainty

Figure 4.21 depicts the difference between required and achieved detection probability at $\alpha = 0.20$ after heterogeneous sensor deployment for a grid size of $[40 \times 40]$ and we can see that the difference is positive at all points so the algorithm achieves the required probability at all points.

We also observe that in case of uncertainty we can not set just any random given required detection for a grid point but a specific empirical value from equation 4.1 which is related to grid dimensions. So it is significantly less than previous two cases. However, the achieved probability is greater than required detection probability at each point of the grid.

Next, we inject three different targets at random positions to figure out whether or not our proposed model can detect all of them regardless of their positions in the simulation grid.

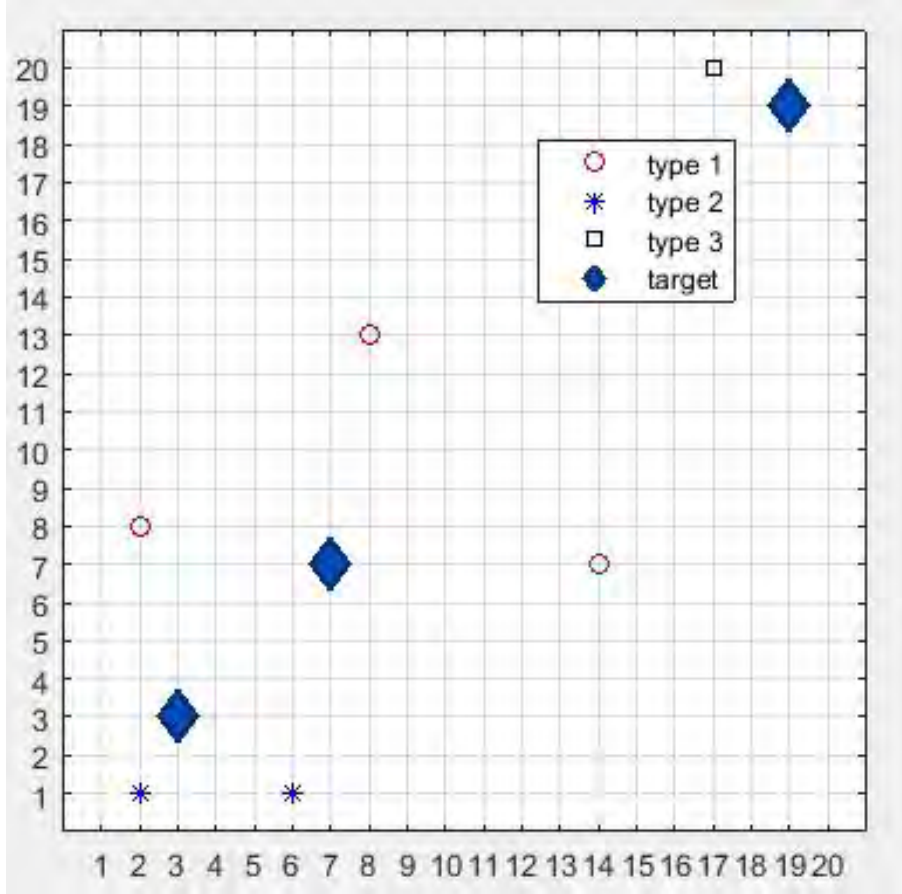


Figure 4.22: Sensors used only for injected target detection for a $[20 \times 20]$ grid size in the presence of uncertainty of deployment

Figure 4.22 shows that at $\alpha = 0.20$ for a grid size of $[20 \times 20]$ and with the presence of deployment uncertainty, three targets are injected and more than one sensors can detect each of the targets. We use random coordinates for these targets. So we can say that our proposed algorithm successfully handles the detection of targets regardless of their positions in case of uncertainty as well.

So in a nutshell it can be said that at first in this chapter, we have evaluated the performances of our developed generalized sensor deployment framework. We have observed that our developed framework is successful in a generalized case. Then, we have compared our results with that of Ref. [34] and these are very close. Next,

we have repeated the process with the augmentation of obstacles and uncertainty of deployment to give a more realistic simulation. It is to be noted that for inclusion of obstacles the number of sensors increases. Moreover, in case of uncertainty the required detection probability for any point in grid is far less than previous two cases since we can not simply choose any random value but an empirical value related to the dimension of axes. However, the achieved detection probability is greater than required detection probability for each points in the grid. Finally, we have injected random target points for all cases and observed that each target has been detected by one or more sensors. So in practical manner our developed generalized sensor deployment framework is successful.

Chapter 5

Conclusion and Future Works

5.1 Conclusion

In this thesis, we have formulated a generalized deployment framework for both homogeneous and heterogeneous sensors using optimal control theory (linear quadratic regulator) for target detection and developed algorithms accordingly. Our formulated generalized deployment framework consists of four algorithms in a sequential order. At first, we have proposed (*Algorithm 1*) to find the best possible \mathbf{A} matrix to get the minimum logarithmic overall achieved miss probability at each grid point for the whole combination of sensors rather than choosing one for each type of sensors and then try to solve all of those at a time. In that way, we have successfully complied with the same method that we can use on a homogeneous sensor deployment as well as in a heterogeneous sensor deployment without broadening the complexity of calculation. *Algorithm 2* describes the sweep method and *Algorithm 3* is the optimal control theory based deployment method. *Algorithm 3* basically gives the co-ordinates of the sensors. Finally, we have proposed another algorithm *Algorithm 4* to determine the type of sensors at each occupied point. Additionally, we have revamped our method to satiate for obstacles and uncertainty of deployment process and observed the performances of the proposed algorithms.

We have simulated our generalized developed framework in MATLAB. We have

observed that our developed framework is successful for both homogeneous and heterogeneous sensor deployment as desired. We have compared our results on homogeneous sensor deployment with that of [34] and these are closely matched. It is evident that in case of obstacles more sensors are required. In the case of uncertainty, the required detection probability for any point in grid is much smaller than the previous two cases since we can not simply choose any random value but an empirical value related to the dimension of axes. However, the achieved detection probability is greater than required detection probability for each points in the grid. Finally, we have injected random target points for all cases and observed that each target has been detected by one or more sensors. So in practical manner, our developed generalized sensor deployment framework is successful.

5.2 Future Works

For future works, firstly, we are interested to add quite a number of additional features to our system model to make it even more realistic which are given below.

- (1) We will try to consider false alarm rate in our model.
- (2) We will try to incorporate the effect of noise in communication channel for our system model.
- (3) Energy constraints of sensors will be taken into consideration. In reality, solar energy driven proximity sensors [43] are available now and we will try to integrate those.
- (4) The performance degradation of sensors with time will be considered and self healing mechanisms will be applied to overcome that.
- (5) In case of obstacles, we will try to incorporate the fringing effect of the transmitted signal of the sensor where it bends over the edges of the obstacles.

Secondly, here, we have only used the term heterogeneous in case of sensing radius meanwhile all the sensors serve the same purpose. However, a combination of sound detectors, smoke detectors and temperature detectors is required to be used

to detect a situation where an industrial explosion occurs. Hence, choosing a heterogeneous combination for each category of sensors and optimizing the deployment process as a whole will be interesting to be studied.

Bibliography

- [1] S. S. Baidya, C. K. Bhattacharyya, and S. Bhattacharyya, “Finding optimal topology for coverage and connectivity using layered deployment model: A comparative study,” *International Conference on Communication, Information and Computing Technology (ICCICT)*, pp. 1–6, 2012.
- [2] N. Hema and K. Kant, “Optimization of sensor deployment in wsn for precision irrigation using spatial arrangement of permanent crop,” *Sixth International Conference on Contemporary Computing (IC3)*, pp. 455–460, 2013.
- [3] A. Raha, S. Maity, M. K. Naskar, O. Alfandi, and D. Hogrefe, “An optimal sensor deployment scheme to ensure multi level coverage and connectivity in wireless sensor networks,” *2012 8th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 299–304, 2012.
- [4] V. Nazarzehi, A. V. Savkin, and B. A., “Distributed 3d dynamic search coverage for mobile wireless sensor networks,” *IEEE Communications Letters*, vol. 19, pp. 33–36, 2015.
- [5] K. Sakai, M. Sun, W. Ku, T. H. Lai, and A. V. Vasilakos, “A framework for the optimal k-coverage deployment patterns of wireless sensors,” *IEEE Sensors Journal*, vol. 15, no. 12, pp. 7273–7283, 2015.

- [6] M. Wei and Y. Shi, “Brain storm optimization algorithms for optimal coverage of wireless sensor networks,” *Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, pp. 120–127, 2015.
- [7] S. Hutchinson and T. Bretl, “Robust optimal deployment of mobile sensor networks,” *2012 IEEE International Conference on Robotics and Automation*, pp. 671–676, 2012.
- [8] M. Jin, G. Rong, H. Wu, L. Shuai, and X. Guo, “Optimal surface deployment problem in wireless sensor networks,” *2012 Proceedings IEEE INFOCOM*, pp. 2345–2353, 2012.
- [9] A. S. Majid and E. Joelianto, “Optimal sensor deployment in non-convex region using discrete particle swarm optimization algorithm,” *IEEE Conference on Control, Systems and Industrial Informatics*, pp. 109–113, 2012.
- [10] R. B. Manjula and S. S. Manvi, “Coverage optimization based sensor deployment by using pso for anti-submarine detection in uwasns,” *Ocean Electronics (SYMPOL)*, pp. 15–22, 2013.
- [11] L. Wang, X. Zou, Meng, and X. Q. Song, “An optimal strategy for the deployment of sensor nodes in green buildings,” *Sixth International Conference on Intelligent Control and Information Processing (ICICIP)*, pp. 209–213, 2015.
- [12] Z. Xiao, J. Yan, X. Yan, G. Cheng, and X. Qiang, “A new optimal deployment algorithm for actor node in wsan,” *International Conference on Network and Information Systems for Computers*, pp. 532–535, 2015.
- [13] u. Z. Y, G. Shan, G. Xu, and X. Duan, “Method of multi-sensor optimal deployment for area coverage,” *International Conference on Electronics Technology (ICET)*, pp. 116–119, 2018.

- [14] S. Singh and A. Kumar, “Novel optimal deployment of sensor nodes using bio inspired algorithm,” *IEEE International Conference on Advanced Communications, Control and Computing Technologies*, pp. 847–851, 2014.
- [15] P. Mondal, K. P. Naveen, and A. Kumar, “Optimal deployment of impromptu wireless sensor networks,” *2012 National Conference on Communications (NCC)*, pp. 1–5, 2012.
- [16] M. Rebai, L. Khoukhi, H. Snoussi, and F. Hnaien, “Optimal placement in hybrid vanets-sensors networks,” *2012 Wireless Advanced (WiAd)*, pp. 54–57, 2012.
- [17] S. Mini, S. K. Udgata, and S. L. Sabat, “Sensor deployment and scheduling for target coverage problem in wireless sensor networks,” *IEEE Sensors Journal*, vol. 14, no. 3, pp. 636–644, 2013.
- [18] S. Singh, S. Chand, R. Kumar, and B. Kumar, “Optimal sensor deployment for wsns in grid environment,” *Electronics Letters*, vol. 49, no. 16, pp. 1040–1041, 2013.
- [19] X. Liu, X. Zhang, and Q. Zhu, “Enhanced fireworks algorithm for dynamic deployment of wireless sensor networks,” *2nd International Conference on Frontiers of Sensors Technologies (ICFST)*, pp. 161–165, 2017.
- [20] X. Xing, D. Xie, and Z. Sun, “An approximate optimal coverage set algorithm in wireless sensor networks,” *2015 Seventh International Conference on Advanced Computational Intelligence (ICACI)*, pp. 300–305, 2015.
- [21] A. Boubrima, F. Matigot, W. Bechkit, H. Rivano, and A. Ruas, “Optimal deployment of wireless sensor networks for air pollution monitoring,” *2015 24th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–7, 2015.

- [22] X. Dong, “Deployment cost optimal for composite event detection in heterogeneous wireless sensor networks,” *2016 3rd International Conference on Information Science and Control Engineering (ICISCE)*, pp. 1288–1292, 2016.
- [23] J. Guo and H. Jafarkhani, “Sensor deployment with limited communication range in homogeneous and heterogeneous wireless sensor networks,” *IEEE Transactions on Wireless Communications*, vol. 15, no. 10, pp. 6771–6784, 2016.
- [24] D. Jiaying, S. Tianyun, L. Xiaojun, and L. Zhi, “Optimal node deployment scheme for wsn-based railway environment monitoring system,” *2016 Chinese Control and Decision Conference (CCDC)*, pp. 6529–6534, 2016.
- [25] H. Yang, L. Ci, S. Zhao, and M. Yang, “An optimal deployment with connectivity information airborne wireless sensor network,” *2016 International Conference on Network and Information Systems for Computers (ICNISC)*, pp. 95–98, 2016.
- [26] R. Bi, X. Fang, X. Zheng, and G. Tan, “Detection quality aware deployment scheme with heterogeneous sensors,” *2017 IEEE International Symposium on Haptic, Audio and Visual Environments and Games (HAVE)*, pp. 1–6, 2017.
- [27] C. Duran-Faundez, D. G. Costa, D. Rocha-Rocha, F. Vasquez-Salgado, G. Habib, and P. Galdames, “On optimal deployment of industrial wireless sensor networks,” *2017 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*, pp. 1–6, 2017.
- [28] F. J. Parrado-García, J. Vales-Alonso, and J. J. Alcaraz, “Optimal planning of wsn deployments for in situ lunar surveys,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 4, pp. 1866–1879, 2017.

- [29] P. Beuchat, H. Hesse, A. Domahidi, and J. Lygeros, “Optimization based self-localization for iot wireless sensor networks,” *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, pp. 712–717, 2018.
- [30] T. O. Olasupo and C. E. Otero, “A framework for optimizing the deployment of wireless sensor networks,” *IEEE Transactions on Network and Service Management*, vol. 15, no. 3, pp. 1105–1118, 2018.
- [31] F. H. Bijarbooneh, P. Flener, E. C. Ngai, and J. Pearson, “An optimisation-based approach for wireless sensor deployment in mobile sensing environments,” *2012 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 2108–2112, 2012.
- [32] A. V. Savkin, F. Javed, and A. S. Matveev, “Optimal distributed blanket coverage self-deployment of mobile wireless sensor networks,” *IEEE Communications Letters*, vol. 16, no. 6, pp. 949–951, 2012.
- [33] H. Park and S. Hutchinson, “Robust optimal deployment in mobile sensor networks with peer-to-peer communication,” *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2144–2149, 2014.
- [34] A. Ababnah and B. Natarajan, “Optimal control-based strategy for sensor deployment,” *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 41, no. 1, pp. 97–104, 2011.
- [35] P. Dorato, C. T. Abdallah, and V. Cerone, *Linear Quadratic Control: An Introduction*. Melbourne, FL: Krieger, 2000.
- [36] A. Sage and C. White, *Optimum Systems Control*. Englewood Cliffs, NJ: Prentice-Hall, 1977.
- [37] M. Mahmoud and M. Singh, *Discrete Systems: Analysis, Control and optimization*. Springer-Verlag, 1984.

- [38] T. S. Helal, P. Mozumdar, and L. Akter, “Evaluating the performance of optimal control based sensor deployment algorithms for realistic terrain model,” *8th International Conference on Electrical and Computer Engineering*, pp. 741–744, 2014.
- [39] A. Syarif, A. Abouaissa, L. Idoumghar, R. F. Sari, and P. Lorenz, “Performance analysis of evolutionary multi-objective based approach for deployment of wireless sensor network with the presence of fixed obstacles,” *2014 IEEE Global Communications Conference*, pp. 1–6, 2014.
- [40] Y. Zou and K. Chakrabarty, “Uncertainty-aware and coverage-oriented deployment for sensor networks,” *Journal of Parallel and Distributed Computing*, pp. 788–798, 2004.
- [41] <https://cdn-shop.adafruit.com/datasheets/gp2y0a710k.pdf>.
- [42] <https://holykell.en.made-in-china.com/product/CyqJpruAnbcd/China-Holykell-5m-10m-50m-60m-Wide-Range-Ultrasonic-Sensor.html>.
- [43] <https://www.youtube.com/watch?v=PU08D3ZYAM8>.