Robert Karlsen Tamang

Deployment of a Mobile Sensor Network using different control methods

Masteroppgave i Cybernetics and Robotics
Veileder: Damiano Varagnolo and Claudio Paliotta
Juni 2020

Masteroppgave

NTNU
Norges teknisk-naturvitenskapelige universitet
Fakultet for informasjonsteknologi og elektroteknikk
Institutt for teknisk kybernetikk

NTNU
Kunnskap for en bedre verden

Robert Karlsen Tamang

Deployment of a Mobile Sensor Network using different control methods

Masteroppgave i Cybernetics and Robotics
Veileder: Damiano Varagnolo and Claudio Paliotta
Juni 2020

Norges teknisk-naturvitenskapelige universitet
Fakultet for informasjonsteknologi og elektroteknikk
Institutt for teknisk kybernetikk

NTNU
Kunnskap for en bedre verden

Abstract

First responders are the first ones to enter a scene of emergency, and in many situations such as fire and search and rescue they expose themself to great risks to help others. New technology may provide safer and more effective work done by the first responders. Having a mobile sensor network to deploy inside the scene of emergency to get an quick overview before first responders enter the dangerous areas is one of the ways technology can help. The deployment of nodes in an unkown environment is a much explored field with good results in terms of coverage, but little research is done in constraints of the total number of nodes used. This thesis investigates the first stage of the deployment problem, how the mobile sensor network should enter a room for fast area coverage while limiting the use of excesive resources.

A simple entry strategy is described where the nodes enter in different group size. Two different controllers are used to spread the nodes once they are inside the environment. Simulations have been conducted using simplified dynamics for the nodes and different rooms to check for environmental dependant behaviour. The results shows that time to cover the area was dependent on multiple variables such as size of sensor and shape of the room. One of the controllers proved very consistent in the amount of drones used to cover the area independent of the room, but the other used more nodes in more complex environments.

Sammendrag

Førsterespondere er de første til å ankomme et åsted og i mange situasjoner utsetter de seg selv i stor fare for å hjelpe andre. Ny teknologi kan øke sikkerheten og gjøre arbeidet lettere for de som ankommer åstedet. Eksempel er a ha en dronesverm som flyr autonomt inn i en bygning og sprer seg for å få oversikt over situasjonen, hvor informasjonen kan bli brukt av førsteresponderne for å få oversikt over hva de har i møte. Utfordringen ved å spre en sverm roboter autonomt i ukjente omgivelser er et bredt utforsket fagfelt med gode resultater på arealdekning, men lite er gjort for å utforske begrensningen av antall noder. Denne avhandlingen utforsker første del av spredningsproblemet, hvordan en sverm med sensornoder kan entre et område for å raskest dekket arealet uten å bruke unødvendig mange noder.

En enkel strategi for entring er beskrevet hvor dronene entrer i ulike gruppestørrelser. To ulike kontrollmetoder er brukt for å spre nodene etter de har entret området. Det er utført simuleringer hvor nodene følger en forenklet systemmodell, og det er brukt ulike rom for å sjekke hvordan området påvirker prestasjonen. Resultatene viser at tida det tar å dekke området er avhenging av flere variabler som sensor størrelse og formen på rommet. En av kontrollmetodene var veldig konsistent på antall noder brukt for å dekke et gitt området uavhengig av rom, mens den andre metoden brukte mange flere noder i komplekse omgivelser.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

A first responder is someone with specialized training who is among the first to arrive and provide assistance at a scene of an emergency such as accidents and natural disasters. They typically include law enforcements, firefighters and paramedics who put themselfs at risk to save other people, property or the environment. In both small emergencies and large emergencies they often deal with life-threatening situations, hazardous environments and to make things worse, limited awareness. New technology can reduce the risk first responders take, providing more safety while performing their helping tasks. They may enable protection of the first responders, enhance their operational capacities by offering them assistance in form of precise positioning, upgraded awareness and allow for information sharing by augmenting their field of view.

Imagine a building exposed to some demolishing power such as an explosion or a fire. The first responders have arrived but there is lots of smoke inside the building preventing the vision of the first responders. Now imagine a swarm of autonomous drones, equipped with sensors and some form of communication, entering the building and spreading out through out the building. In the end, the swarm has mapped and is covering the whole building. With the communication, the swarm can send information to the first responders, and by using augmented sight technology they can use the swarms sensors to get better awareness within the building.

The task of having a swarm of nodes, or a mobile sensor network, spread out autonomously in an unknown environment without central coordination is referred to in the literature as the self-deployment problem Bayindir (2016). The task has two objectives: dispersion or pattern formation. Bayindir (2016) writes: *"In the dispersion task, swarm members must position themselves away from one another, with the objective of maximizing the area covered globally by the swarm..."*, which is what is described in the emergency scenario in the paragraph above.

1.2 Previous work on the deployment problem

The problem of area coverage can be divided into three types of coverage: blanket coverage, barrier coverage and sweep coverage Douglas (1992). In blanket coverage, the objective is a static configuration of nodes that maximizes a cost function over the total area. If the cost function is constant, this is equal to maximizing the covered area. The barrier coverage has the objective to achieve a static configuration of the nodes to minimize the probability of undetected penetration through a barrier and sweep coverage is roughly equivalent to a moving barrier. By these definitions, the problem in this thesis falls into the blanket coverage.

Batalin and Sukhatme (2002) used a bio-inspired method to solve the blanket coverage problem. They made the drones repel off each other the same way as molecules do to spread out in different directions. They also proposed another method where the robots were equipped with visual identification and based on visual communication they coordinated their next move. In both methods, the robots would repel off the obstacles in the environment. Area coverage was an effect of the repelling. They were both compared to a simple random walk and both outperformed the random walk. They concluded that the ability of a robot to differentiate other robots and obstacles is critical and improves performance. David et al. (2001) was also bio-inspired and described a way of using virtual pheromones for search and rescue of humans. Unlike the ants' pheromones which leave trails location-specific on the ground, the information for guiding the swarm is carried with the robots. Hsiangl et al. (2004) tried a leader-follow strategy, where the front node was a leader and entered a passiv state to create guiding for the next node, and then a new node become the leader and kept exploring.

Potential fields as a technique for robotic applications were first described by O (1986) and have since been widely used in mobile robotics for navigation and obstacle avoidance. Howard et al. (2002) applied the same virtual potential field technique on a swarm of robots to make them self-deploy in an unknown environment. The potential field is induced by other robots and obstacles, creating a force that spreads the robots apart. The robots in this work have a range bearing intensity sensor, where the intensity was added to make the robots able to distinct other robots from the obstacles. The area coverage was an emergent property of the system, and they demonstrated that potential fields can be used to deploy a swarm in an unknown environment.

Sotiris and Anthony (2018) took a different approach to the problem of area coverage. Where the other systems had coverage as an effect of the spreading forces, Sotiris and Anthony (2018) made a control law where the robots actively move towards increased area covered. They defined a cost function for the total area covered which could be distributed and each robot could calculate their part of the total area covered. This way each robot can calculate its input to search towards increasing the area. They also provided proof that the total area was monotonically increasing. The simulations showed promising results, but experiments were only conducted in a small convex area. The paper also described modifications due to uncertainty in measurements.

1.3 Thesis objective

Even though the deployment problem has been widely researched, the work often assumes a fixed number of robots and that the robots are already inside the environment. This is often not the case in real-life situations, where for example the swarm of rescue robots are stored at one location and need to be transported from their storage location to the scene of the emergency. The robots arrive at the scene and should enter the environment and spread out as fast as possible. At the same time they enter as fast as possible, they should not enter too many. The number of robots used should be so many that they cover the area in a reasonable time, but if one can prevent using too many, the rest can be relocated to another scene of an emergency. Hence the problem arrise: how to enter the robots into the environment to get a fast coverage without using unnecessary many robots? This thesis will make use of a simple entry method to see if there are any advantages to how robots enter the scene. The method will be tested using two different methods for dispersion. The focus will be on a high level controller for a simplified system to explore main concepts. Chapter 2 contains background theory needed for the implementations of the methods. Chapter 3 presents the assumptions done to specify the problem before describing the entering strategy and the two controllers for dispersion. Chapter 4 presents the tests performed and clarify the evaluation criteria before a discussion around the results is done in chapter 5. Chapter 6 contains the conclusion that wraps up the report.

Chapter 2

Backgroun Theory

2.1 System modelling

To simulate a system, a model of the system is needed. Every rigid body follows Newton's second law Balchen et al. (2016):

$$\sum F = ma$$

where F represents the forces acting on the system, m is the mass of the system and a is the acceleration. By identifying the possible forces acting on the system, it can be described by the equations

$$\begin{aligned} \dot{x_1} &= x_2 \\ \dot{x_2} &= \frac{1}{m}\sum F = \frac{1}{m}(\sum F_{external} + F_u) \end{aligned}$$

where x_1 is the position of the system and x_2 is the velocity. $F_{external}$ is all the external forces acting on the system and F_u is the force generated by the system's actuators, initiated to change its movement.

Assuming powerful actuators and very little to no inertia the system can be simplified to:

$$\dot{x} = \sum F \tag{2.1}$$

where $\dot{x}$ represents the velocity. This is a simple model useful when the focus is on testing high-level controllers. The model was used by Sotiris and Anthony (2018) when testing their methods for area coverage.

2.2 Virtual potential fields

Virtual potential field is common method in robotics, escpecially in obstacle avoidance but are also used for deploying a swarm of nodes, as in Howard et al. (2002). The method is based on the environment setting up a virtual potential field U that exerts a force onto the nodes.

$$\mathbf{F} = -\nabla U \tag{2.2}$$

The potential field are created both from the obstacles in the environment and other nodes, $U = U_o + U_n$. Which leads to $\mathbf{F} = \mathbf{F_o} + \mathbf{F_n}$, where subscript 'o' signifies the effects from the obstacles and subscript 'n' signifies effects from other nodes.

Imagine a node and a nearby obstacle each having an electrical charge. Then the electrostatic potential between them are givin by $U = k\frac{1}{r}$ for some constant k and r is the euclidean distance between the node and the obstacle. Taking the sum of all obstacles around the node gives:

$$U_o = k_o \sum_i \frac{1}{r_i} \tag{2.3}$$

k_o is a constant describing the strength of the field and r_i is the euclidean distance between the node and obstacle i.

If $\mathbf{x}$ is the position of the node, and $\mathbf{x_i}$ is the position of the i'th obstacle, r_i can be given by

$$r_i = |\mathbf{x_i} - \mathbf{x}| = |\mathbf{\Delta x_i}| \tag{2.4}$$

Inserting equations 2.3 and 2.4 into 2.2 and calculate using chain rule we get

$$\begin{aligned}
\mathbf{F_o} &= -\frac{dU_o}{d\mathbf{x}} \\
&= -\sum_i \frac{dU_o}{dr_i} \cdot \frac{dr_i}{d\mathbf{x}} \\
\mathbf{F_o} &= -k_o \sum_i \frac{1}{r_i^2} \cdot \frac{\mathbf{\Delta x_i}}{r_i}
\end{aligned} \tag{2.5}$$

Since all the terms are in relative positions $\mathbf{\Delta x_i}$ and its norm, the force can be computed locally from sensor data without any globalized map.

The same reasoning can hold for the force generated from other nodes. Be taking the same formulas and consider nodes instead of obstacles we get the same equations:

$$U_n = -k_n \sum_i \frac{1}{r_i} \qquad \text{and} \qquad \mathbf{F_n} = -k_n \sum_i \frac{1}{r_i^2} \cdot \frac{\mathbf{\Delta x_i}}{r_i} \tag{2.6}$$

Note that k_n and k_o can be different to make the nodes repell more or less from each other compared to obstacles.

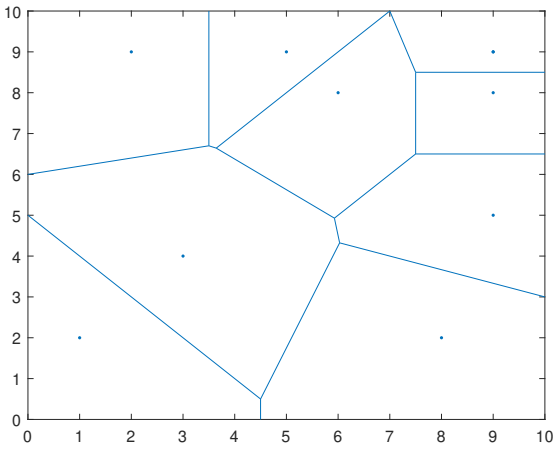


Figure 2.1: Example of voronoi diagram with nine seeds

2.3 Space partitioning

Consider a set of points $Q = \{q_1, ..., q_n\}$, called seeds, within a space Ω. A voronoi diagram, Mumm (2004), is a partitioning of Ω such that all points in the same cell as seed q_i, has q_i as the closest seed. Cell P_i can be written as:

$$P_i(\Omega, Q) = \{p \in \Omega : ||p - q_i||^2 \leq ||p - q_j||^2, \text{for } j = 1, ..., n, j \neq i\}, i = 1, ..., n \quad (2.7)$$

The voronoi diagram is a complete tesselation of Ω which gives the properties $\cup_{i=1}^{n} P_i = \Omega$, and $Interior(P_i) \cap Interior(P_j) = \emptyset$, for $i \neq j$. An example of a voronoi diagram is given in fig 2.1. Voronoi diagram has a dual relationship with Delaunay Triangulations which is commenly the way to calculate voronoi diagrams.

A generalization of the voronoi diagram is the power diagram. Instead of the seeds being points, the seeds are disks with a radius R_i. The partition is then written:

$$P_i(\Omega, Q) = \{p \in \Omega : ||p-q_i||^2 - R_i^2 \leq ||p-q_j||^2 - R_j^2, \text{for } j = 1, ..., n, j \neq i\}, i = 1, ..., n \quad (2.8)$$

2.4 Deriving the open border control law

The information in this section is found in Sotiris and Anthony (2018) which includes the proof that the control law leads to a monotonic increase in the total area covered. The

reader is referred to the paper for the proof, but this section will present the theory and the control law.

Imagine a planar region Ω with n nodes acting as seeds for a voronoi diagram as described in sec. 2.3, where node i has a sensing disk with radius R, covering area S_i. Node i is located at position q_i and follows the dynamics $\dot{q_i} = u_i$

For any two nodes i and j, if some part of node i's sensing area is inside the voronoi cell of node j, then by definition in eq. 2.7 this area must be within the sensing area of node j. This gives rise to define the R-limited voronoi cell of node i as $P_i^R = P_i \cap S_i$. By utilizing this the total area covered can be written as an objective function:

$$H = \sum_{i=1}^{n} \int_{P_i^R} \phi(p)\, dp \tag{2.9}$$

, where ϕ is a density function to describe if there are some areas more valuable to cover than others. Since H can be written as a sum of integrals over R-limited voronoi cell and each node can compute its own voronoi cell using Delaunay neighbours, the computation of H can be distributed. This can be used to derive a control law to maximize the objective function.

$$\frac{\partial H}{\partial t} = \sum_{i=1}^{n} \frac{\partial H}{\partial q_i} \cdot \frac{\partial q_i}{\partial t} = \sum_{i=1}^{n} \frac{\partial H}{\partial q_i} \dot{q_i} = \sum_{i=1}^{n} \frac{\partial H}{\partial q_i} u_i$$

Choosing the control law $u_i = \alpha_i \frac{\partial H}{\partial q_i}$, $\alpha_i > 0$ gives a guarantee monotonic increase of the coverage objective.

$$u_i = \alpha_i \int_{\partial P_i^R \cap \partial S_i} n_i \phi(q)\, dq \tag{2.10}$$

where α_i is a positive constant, ∂ denote the border of the area and n_i is the outward unit normal vector.

The control law also works if the nodes have different radius sizes on their sensor. P_i^R representing the R-limited voronoi cell is simply replaced by the R-limited power cell.

2.5 Intersection of circles

The theory presented in this section can be found at W. (2000).

The formula for a circle with radius r and center (x, y) is given by

$$(x - x_1)^2 + (y - y_1)^2 = r^2 \tag{2.11}$$

When considering any two circles, a rotated coordinate system can be used to simplify the coordinates without loss of generality. The new coordinate system has origin in one

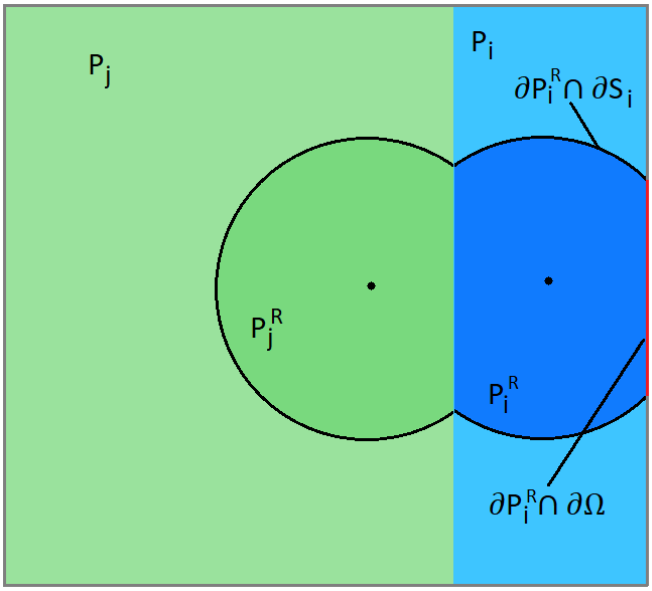


Figure 2.2: Shows border intersection used in eq. 2.10

of the circle centers with the x-axis pointing toward the other circle center and y-axis 90 degrees on the x-axis. This gives the two circle equations

$$x^2 + y^2 = r_1^2 \tag{2.12}$$
$$(x-d)^2 + y^2 = r_2^2 \tag{2.13}$$

where d is the distance between the circle centers. The intersection points is found by solving for x and y. Re-arrange the first equation for y^2 and substitute in the second equation gives:

$$\begin{aligned}(x-d)^2 + r_1^2 - x^2 &= r_2^2 \\ x &= \frac{d^2 + r_1^2 - r_2^2}{2d} \qquad (2.14)\\ y_{1,2} &= \pm\sqrt{r_1^2 - x^2} \qquad (2.15)\end{aligned}$$

2.6 Map storing

A map can be stored in two different ways: a continuous map or a discrete map (also called grid map) Ben-Ari and Mondada (2017). In a grid map, the map is divided into

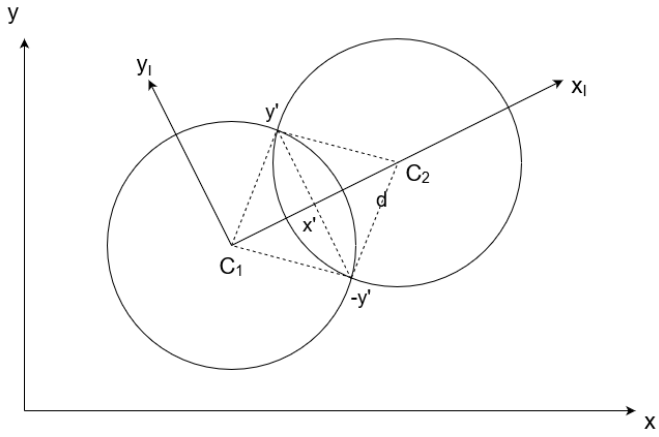


Figure 2.3: Showing intersection points of two circles. d is the euclidean distance between C_1 and C_2.

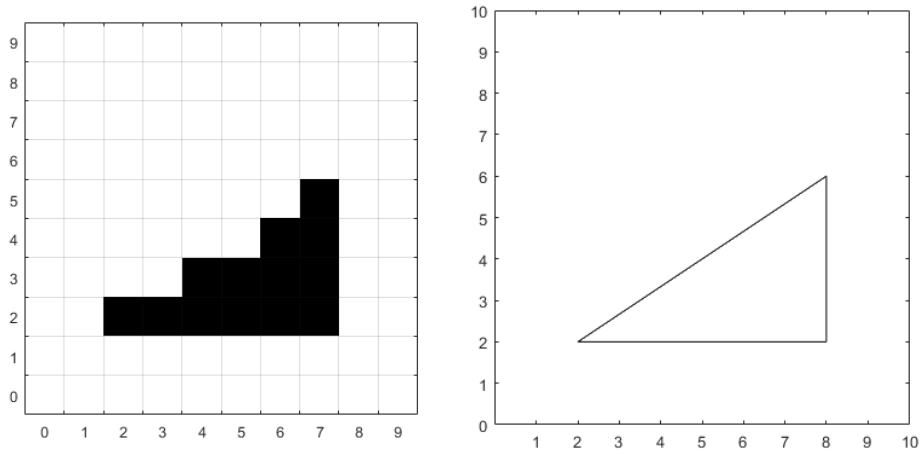
cells where each cell has a value if there is an object in that location or not. Fig. 2.4a shows an example of a 10 x 10 grid map with a triangle object. The object is stored as a list of the coordinates of each cell the object is occupying. The list for this object is:

(2,2), (3,2), (4,2), (5,2), (6,2), (7,2), (4,3), (5,3), (6,3), (7,3), (6,4), (7,4), (7,5)

Fig 2.4b show the same object stored in a continuous map. The object is now stored as the coordinates of the corners:

A = (2,2), B = (8,2), C = (8,6).

Continuous maps has the advantage of accuracy, and if the environment contains few and simple objects the continuous map can be very efficient. However, if the objects are many and have complex figures, it may require a lot to represent the shapes and it may be very cost demanding to store the objects. The discrete map on the other hand has low accuracy and can be made more accurate at the expense of higher resolution on the cells and more processing power.



(a) A discrete map showing occupied cells of an object.

(b) A continuous map of same object

Figure 2.4: Illustrating the difference of storing a triangle in a discrete map and a continuous map

Chapter 3

Design/Implementation

The code implementation can be found in https://github.com/robertkt/AreaCoverage

3.1 Problem specification

How to deploy a mobile sensor network in an environment and make them spread for maximum coverage is a big and general problem. To limit the scope of the project the problem is narrowed to evaluate high-level control strategies for a simplified system model. The assumptions are:

1. The objective is maximum coverage $\Rightarrow \phi(q)$ in eq. 2.10 is constant.
2. The nodes are at a constant height, motion is in 2 dimensions.
3. Each node is represented by a point mass and can move in any direction.
4. The nodes' motion can be described by a simplified system: $\dot{x} = u$, where x is the position of the node and u is the control input.
5. Environment is bounded by walls and only contains 90^{o} corners. There are also walls at the entrance point.

3.2 How nodes are entering

Assuming there are borders at the entrance point as well, the nodes will move away from the entrance point after entering the environment. If there are no obstacles or other robots pushing them back to the entrance point, there is free space for the robots to move into. As long as the robots move away from the entry point there are more free space to cover hence more robots are needed. This creates a simple method for entering robots: if there are no robots inside the entrance point, enter more robots.

$$\text{if } ||p_e - p_i||^2 < r \ \forall i, i = 1, ..., n \Rightarrow \text{enter new node} \tag{3.1}$$

where p_e is the position of the sensor located at the entrance point, r is the radius for which the measurement is conducted and p_i is the position of node i. This is also something that is easy to implement and use in a real case scenario as it only needs one additional sensor at the entrance point. The nodes are entering in groups of j, $j = 1, .., 4$ in a horizontal line with $0.5m$ space between them. This might create an unfair advantage to high number groups as they initially have more coverage than low number groups when entering. This is something to bear in mind when analyzing the results.

3.3 Implementation of Open border

One of the control laws implemented for the dispersion is the one described in sec. 2.4, from now on referred to as the Open border control law. Sotiris and Anthony (2018) and other work mentioning the computation as using Delaunay triangulation to find the voronoi diagram but to the writer of this thesis knowledge, there are no open source code of voronoi as control law and how the diagram changes when there are obstacles in the environment. Hence a different approach was used for calculating the input. Below follows how the Open border was implemented.

In eq. 2.10, P_i^R is determined by how close the node is to other nodes or obstacles, which can be seen in fig. 2.2. If the node are closer than 2 times their sensing radius, their sensing disk intersect as described in sec. 2.5. Each node store a set of free angles, given as pairs of start angle and end angle. A free angle, illustrated in fig. 3.1, is defined as an angle on the node's disk, referenced to the x-axis, such that the radius of the sensing disk at that angle also lies on $\partial P_i^R \cap \partial S_i$. So when a node's sensing disk intersect with another one's sensing disk at at angles θ_1 and θ_2, $\theta_1 < \theta_2$, the set $[\theta_1, \theta_2]$, is subtracted from the set of free angles. This lets the input be calculated as integrals over the free angles:

$$\begin{aligned}
\mathbf{u_i} &= \alpha_i \sum_i \int_{StartAngle(i)}^{EndAngle(i)} \begin{bmatrix} cos(\theta) \\ sin(\theta) \end{bmatrix} d\theta \\
&= \alpha_i \sum_i \begin{bmatrix} sin(\theta) \\ -cos(\theta) \end{bmatrix}_{StartAngle(i)}^{EndAngle(i)} \\
&= \alpha_i \sum_i \begin{bmatrix} sin(EndAngle(i)) - sin(StartAngle(i)) \\ -cos(EndAngle(i)) + cos(StartAngle(i)) \end{bmatrix}
\end{aligned} \tag{3.2}$$

where i iterates through all pairs of free angles.

3.4 Potential field

The other control method implemented for dispersion was based on the potential field described in sec. 2.2, from here on referred to as the Potential field controller. The input u is given by the force F in eq. 2.2.

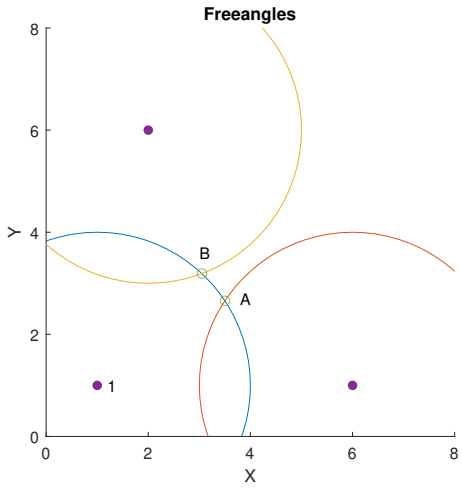


Figure 3.1: Illustrating free angles for node 1. Free angles = [[A,B]]

```
1 -  close all;
2
3 -  fRadius = 4;
4 -  afAngles = 0:pi/40:2*pi;
5 -  afX = fRadius*cos(afAngles(1:end-1));
6 -  afY = fRadius*sin(afAngles(1:end-1));
7
8 -  tPolyshape = polyshape(afX,afY);
9 -  fApproxArea = tPolyshape.area()
0 -  fActual = pi*fRadius^2
```

```
fApproxArea =

   50.2138


fActual =

   50.2655

fx >>
```

Figure 3.2: Code for how to use polyshape and size of error due to circle approximation

3.5 Calculating covered area

To calculate the area covered by all the nodes, the Matlab function ”polyshape” was used. This takes a set of point coordinates and creates a shape where the corners of the shape are represented by the points. Polyshape also has a built-in function that calculates the area of the shape. Each node creates a polyshape using its sensed ranges of the walls. Since polyshape creates a polygon it is only an approximation of a circle, but by using enough points the error between the approximation and a complete circle is minimal. Each node creates a shape of their sensed area and to calculate the total area, each node’s area is being intersected together to one big shape before the area is calculated. Fig. 3.2 shows how polyshape can be used to approximate the area of a circle with radius 4. It also shows how big the error is in the approximated area when the radius is 4.

Chapter 4

Experiments and results

4.1 Experiment setup

This chapter presents a set of experiments conducted to evaluate how different enterings of the mobile sensor network effect the performance. Two metrics are of particular interest for evaluation:

- How fast do the nodes spread to cover the area
- How many nodes are used

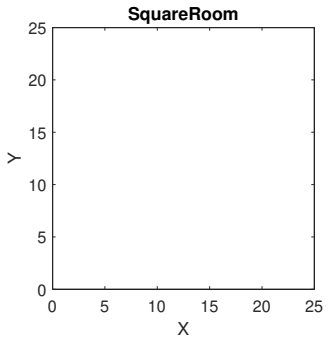
The simulations were conducted in a self built simulator using the simplified dynamics $\dot{x} = u$, where u is the input from the controller. The discretization was done using first order euler method.
For each simulation:

- There are a maximum of 100 nodes for deployment. Each equipped with a 360 degree range bearing sensor to cover the area.
- Range of the detection sensor = r, $r \in \{2, 3, 4\}$
- The nodes are deployed into a room with shape s, $s \in \{Square\_room, C\_shaped\_room, Square\_room\_with\_obstacle\}$. The different rooms are shown in fig. 4.1.
- The nodes enter the rooms at position p_e,

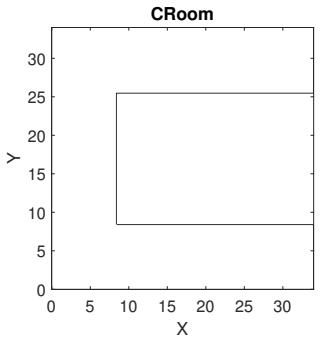
$$p_e \in \begin{cases} [2,0],[12,0] & \text{if } s = Square\_room \\ [2,0],[16,0] & \text{otherwise} \end{cases}$$

- The nodes enter the rooms in groups of size j, $j \in \{1, 2, 3, 4\}$.
- One of two different controllers is used, as described in chapter 3.

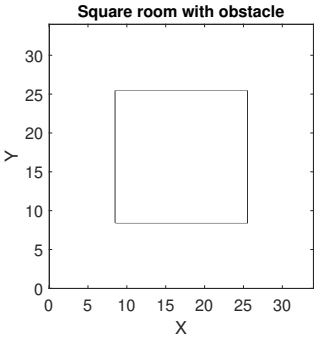
Any nodes a distance l away from each other, where $l <$ than 2 times the sensor radius will have overlapping sensing disks. Since the Open border control law uses the free angles it must know about any node within 2 times the range of the sensor radius. To know about nearby nodes outside the sensor range, each nodes is also equipped with some communication device that lets them communicate with other nodes at a distance $d > 2$ times the sensor radius, to know if there are any nodes' sensing disk that intersect with their's.



(a) Map of square room. Total area = 625



(b) Map of C-shaped room. Total area = 722.5

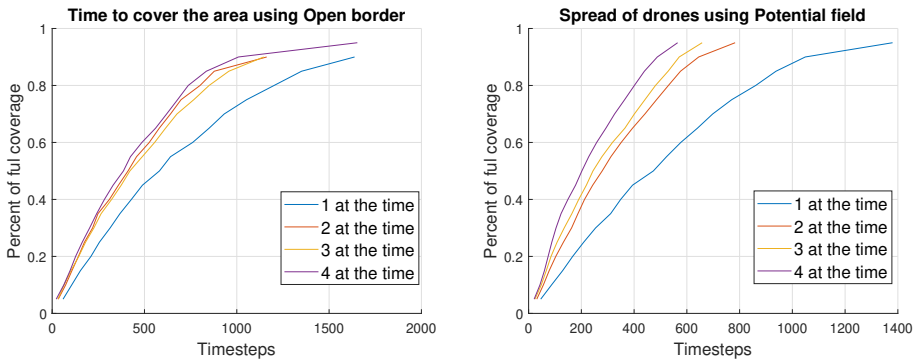


(c) Map of square room with obstacle in the center of room. Total area = 867

Figure 4.1: Maps of the different rooms used in the simulations

4.2 Square room

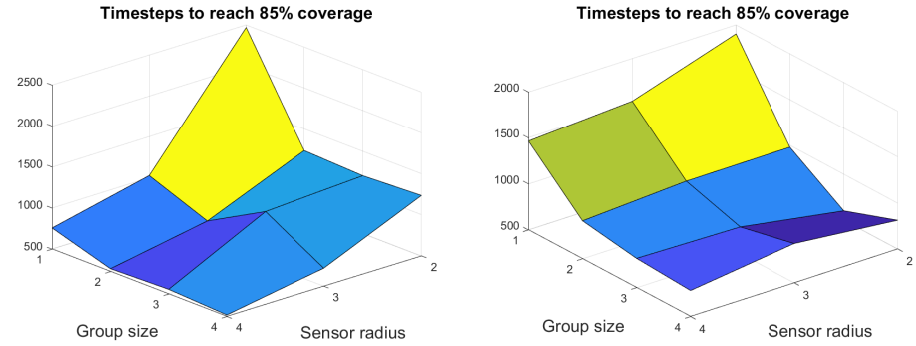
The first simulations were done in a square room. Fig. 4.2 shows how fast the nodes spread when entering in different group sizes. For both control laws, entering 1 node at the time gave slower spread. For the Open border control it doesn't show a significant decrease in time it takes to spread when entering more nodes at the time, but for Potential field it seems to give a slight advantage to enter them more at the time. Another thing to notice is that the increase rate seems to be similar for all tests, with a small gain difference.



(a) Open border as control law

(b) Potential field as control law

Figure 4.2: Showing time it takes to spread out in square room when nodes are entering in different group sizes



(a) Open border as control law

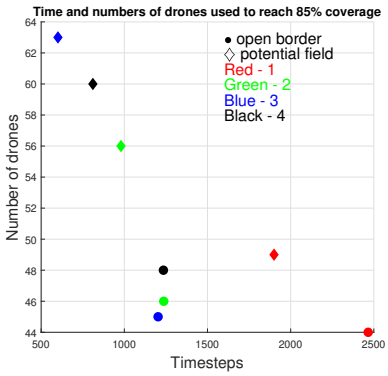
(b) Potential field as control law

Figure 4.3: Showing time it takes to reach 85% of ful coverage, with different sensor range and entering group sizes.

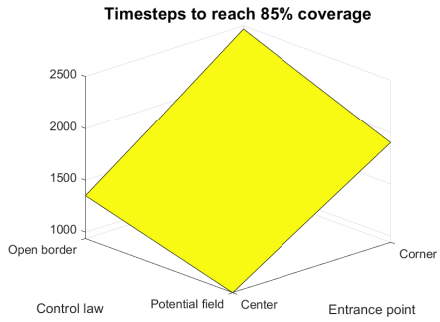
This makes comparing which simulation to be fastest to 60% equal to comparing which simulation was fastest to 90% of ful coverage.

A difference between the control laws is how the sensor range affect the time to spread. Fig. 4.3 shows that for the Open border control, an increase in sensor radius gives a decrease in time it takes to spread. A change in sensor radius seems to affect the spread time for the Potential field control very little to nothing. It might even reduce time a little by raducing sensor radius.

The biggest difference between the control laws is the total number of nodes they use to spread out. Fig. 4.4a show that the Potential field control law is faster to spread then the Open border when the sensor radius is small, but uses more nodes as well to reach the faster



(a) How fast to spread and how many total nodes was used to reach that spread. Entrance at the corner. Shape of point represents which contorl law used and colour of point represents group size of entering nodes.



(b) How spread time changes with control laws and entrance point. Enter 1 node at the time.

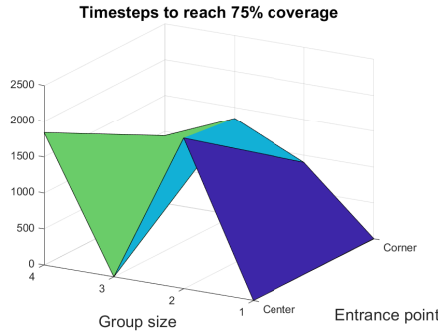
Figure 4.4: Shows how fast to spread to 85% coverage. Sensor range = 2.

coverage. The detection radius of the nodes in this simulation is 2, which gives a possible maximum coverage of $63 \cdot \pi 2^2 = 791.7$ for the Potential field when entering 3 at the time. At this timestep the nodes are only covering 85% of ful coverage $= 0.85 \cdot 625 = 531.3$. While at the same coverage the Open border is only using 45 nodes to cover 85%. Both control laws spread the nodes faster when the nodes entered from the bottom center of the room compared to from the corner as shown i fig. 4.4b.

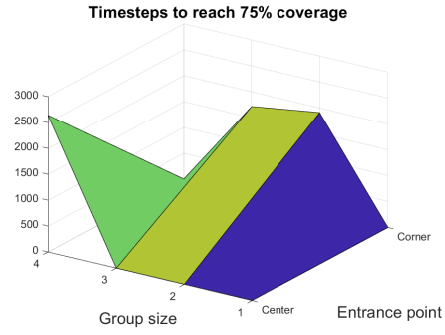
4.3 C-shaped room

The results presented in this section are all extracted from simulations done in the C-shaped room fig. 4.1b. The main differences between this room and the square room are the tighter space and the room is no longer convex. This often lead to the swarm getting stuck at an equilibrium at low coverage. Fig. 4.5b shows the Potential field control fails to reach 75% coverage half of the times when the sensor radius = 4. The Open border also fails but not in as many scenarios as the Potential field. Both control laws struggled more when entering from the bottom center of the room compared to the corner. This is only a problem when the sensor radius is big. Fig. 4.5c and 4.5d shows the same scenario as 4.5a and 4.5b but with sensor radius = 2. The Open border reaches the coverage faster when entering from the corner compared to entering from the bottom center, but the Potential field is close to uneffected of the entrance point. Putting aside the scenarios that didn't reach 75% coverage, another result the figures show is the Open border control spreads faster when the sensor radius increases. But for the Potential field control the time to reach 75% doesn't seem to be affected by how big the sensor radius is.

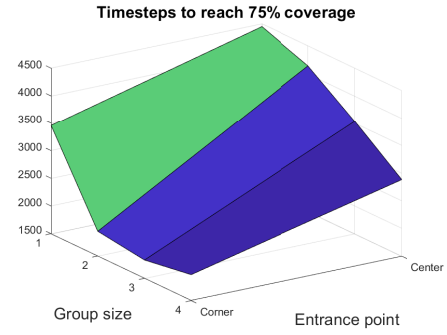
Fig. 4.6 shows time vs total number of nodes to reach 75% of ful coverage. The



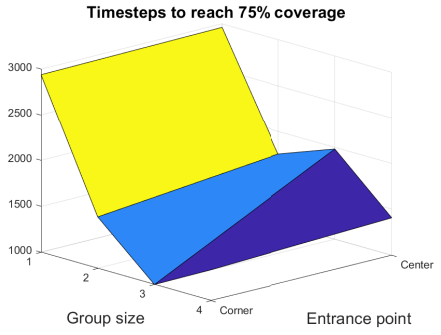
(a) Open border as control law, $r = 4$.



(b) Potential field as control law, $r = 4$.



(c) Open border as control law, $r = 2$.



(d) Potential field as control law, $r = 2$.

Figure 4.5: Illustrates the effect of entrance point on time to reach 75% coverage.

Potential field control is faster to spread but uses a lot more nodes to reach the same coverage as the Open border control uses.

4.4 Square room with obstacle

The results in this section was all conducted from the square room with an obstacle in the center of the room, shown in fig. 4.1c. The shape is very similar to the C-shaped room, but has an extra path between the end points, creating a circluar shape. In some of the simulations, the tight spaces prevents the swarm from ful spread. Fig. 4.7 shows two simulations where the nodes fail to make the turn around the corners. The figure also shows the bigger distance the Open border control creates between the nodes compared to the Potential field control.

A difference between the control laws is how they respond to different sensor radius size, illustrated in fig. 4.8. The Open border control had faster coverage the bigger the sensor radius was. But for the Potential field, bigger sensor radius lead to slower cover-

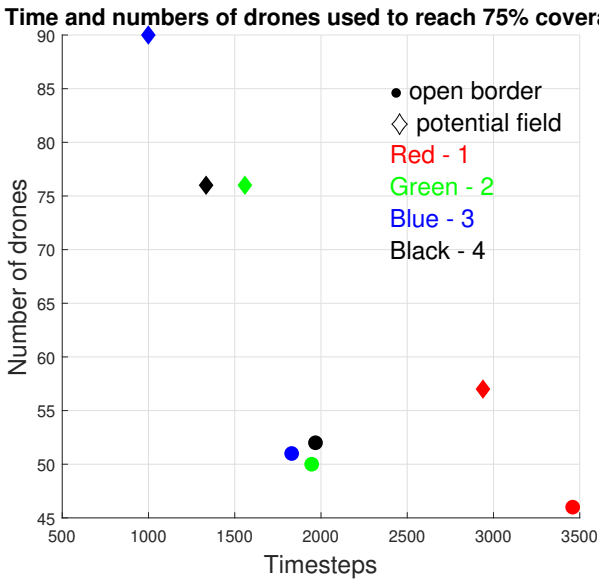
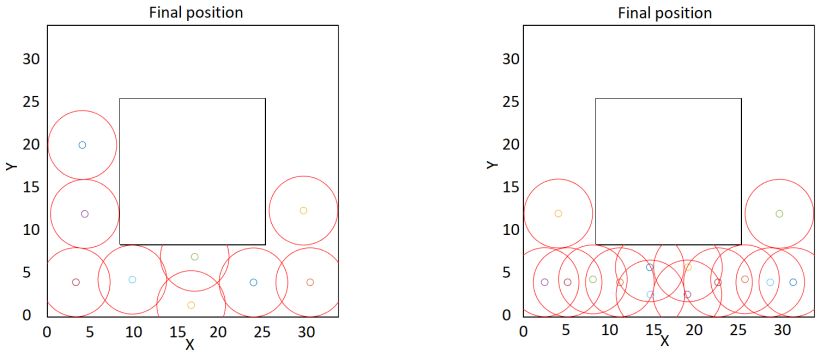


Figure 4.6: How fast to reach 75% of ful coverage and how many total nodes was used to reach that coverage. Entrance at the corner. Shape of point represents which control law used and colour of point represents group size of entering nodes



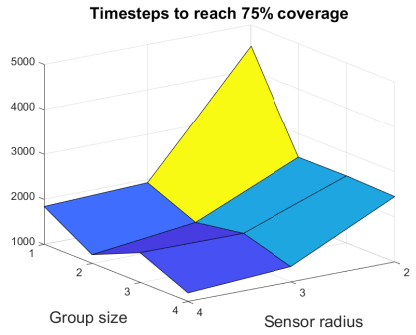
(a) Open border as control law. Group size = 3.

(b) Potential field as control law. Group size = 2.

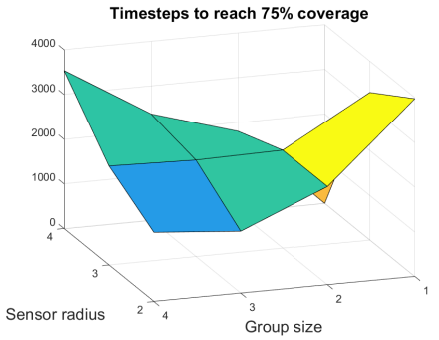
Figure 4.7: Final position spaces when nodes get stuck at low coverage when sensor radius is big.

age when multiple nodes was entering at the same time. When the group size of entering nodes was low, bigger sensor radius gave faster coverage for the Potential field. Fig. 4.9 compares the effect of radius size on the time it takes to reach a coverage of 75%. It shows that the Potential field control uses less time when the radius decreases as opposite to the Open border control which increases time used when the radius decreases.

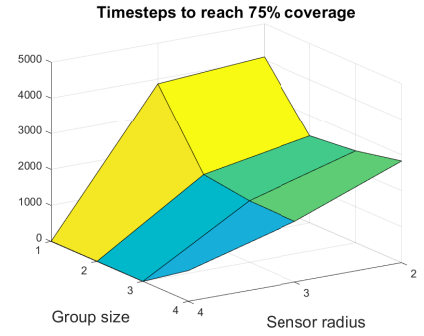
Fig. 4.8 also shows how the controllers reacted to different entry positions. Comparing (a) to (c) the figures show that the Open border controller was slightly faster entering from the corner than from the center. In contrast, comparing (b) to (d), the Potential field controller was faster spreading from the center compared to the corner. Both controllers, on average, ended with a lower coverage entering from the center compared to entering from the corner.



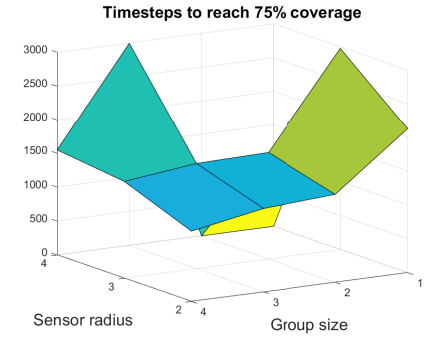
(a) Open border as control law.



(b) Potential field as control law.



(c) Open border as control law.



(d) Potential field as control law

Figure 4.8: Shows how many timesteps to reach 75% coverage. In fig. (a) and (b) nodes entered at the corner, while in fig. (c) and (d) nodes entered at bottom center of the room.

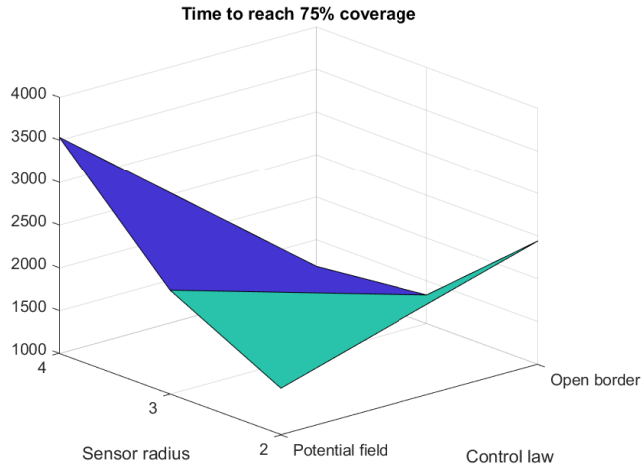


Figure 4.9: Comparing the effect of radius size on different controllers, when entering in group size of 3.

Chapter 5

Discussion

5.1 Main results

The simulation results show that both controllers benefited from entering more nodes at the time. For the Open border, the time used to spread decreased with an increasing number of entering nodes. This might be due to how the nodes were entering. The way they entered rooms, bigger groups had bigger initial spread. Maybe a more optimal way to enter the nodes would be to have a constant size the nodes initial position would cover, e.g. 1 meter, and distribute the nodes evenly within that range. This would not affect the result when entering 1 node at the time but would give the rest of the group sizes an equal initial coverage. For the Potential field control, it was almost the same, except that entering nodes in groups of 3 usually gave faster coverage than entering in groups of 4. The result that entering nodes in groups of 3 gave faster coverage than entering in groups of 4 is an interesting result. 4 nodes give bigger initial spread since the edge nodes are further apart in groups of 4 than in 3. Yet entering 3 at the time was faster. This could mean that the formation of the entering nodes have an effect on how fast the nodes cover the room. Further experiments could be to test and see if different entering formations give an impact on the time it takes to cover the area. Entering 3 nodes at the time also resulted in faster spread away from the entrance point, which resulted in more total nodes used. In all simulations, the Potential field control used more nodes when entering 3 at the time compared to the other group sizes, whereas the Open border control always used most nodes when entering in group size of 4.

Both controllers manage to spread the nodes to cover the area in open spaces. In the C-shaped room and the room with an obstacle in the middle, they struggled to get around the corners. If the radius got too big with respect to the passage they failed to make it through the corners. A surprising result is that the Potential field control failed more often than the Open border control to make the turns around the corners. Since the Open border is based on free angles on the sensing disk, tight areas lead to less movement of the nodes. Fig. 5.1 illustrates the restriction in movements for the Open border controller. An ini-

tial guess was that since the Potential field control doesn't restrict the nodes from moving when in a group, it would create more movement and bigger possibility to make it around the corners. This seemed not the case. However, the ability to move within each other's sensing range had a big impact on coverage time when entering from different locations in the C-shaped room. The Open border control was faster when entering from the corner compared to the bottom center of the room. This makes sense as the corner is closer to the middle of the endpoints of the C-shaped room. In addition, entering from the corner means one less corner to turn around to cover the whole room. But the Potential field control was close to zero affected by the entrance point when entering multiple nodes at the time. The time it took to reach 75% coverage was almost the same between entering from the corner of the room and entering from the bottom center of the room. The exception was when entering in a group size of 1, then the Potential field benefited a lot from entering at the corner. The result can be explained by the computation of the controllers. When entering multiple nodes from the corner, both controllers spread them to both directions, right and upwards. But when entering from the edge, all nodes must travel the same directions. Since the Open borders is based on free angles, entering more nodes won't create more free angles for the front node, hence they are not moving faster with bigger group sizes. But for the Potential field, the force acting on the front node is based on how many and how close the nodes are behind. Entering more nodes will create a bigger force making the front node to move faster. Hence the Potential field is faster than the Open border to spread from the edge point of the room.

An interesting result was the effect the sensor radius had on the coverage time for the different controllers. The Open border controller had faster coverage time when the sensor radius increased. This is expected as the controller input is calculated by an integral over the free angles. The bigger radius the bigger the integral can be. This could maybe be compansated for by using a bigger gain, but might lead to oscillations when using too big gain. But with a constant gain, bigger radius gives bigger control input. In contrast, an increase in radius gave an increase in time used to cover the area with the Potential field controller. Fig. 4.9 illustrates how an increase in sensor radius gave an increase in coverage time for the Potential field control, but a descrease in coverage time for the Open border control. This could be an effect of the control law. Since the input is based on distance, a shorter sensor radius means that nodes are entering faster than for larger radius, but the distance to the wall is the same independent on the sensor radius which gives same input for the newly entered nodes. This only seemed to be the case in the rooms with narrow passages. In the square room the Potential field also had a decrease in coverage time when the sensor radius increased.

A biggest difference in results between the controllers was the total numbers of nodes used to cover the area. The Potential field controller always used more nodes than the Open border controller. This can be explained by how the controllers calculate their input. The Open border is based on free angles to create input. This results in very limited to none movement when surrounded by other nodes, see fig. 5.1. In narrow passages this can resemble a leader-follower strategy. Since the controller searches open space, the first node must move away to create open space for the next node to follow. What seems to be

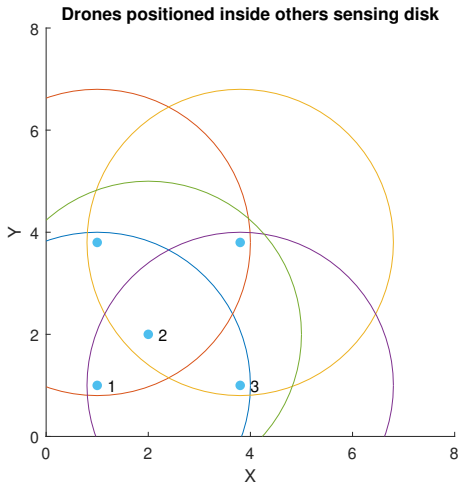


Figure 5.1: Node 1 and 2 have no free angles. This would create no movement for them using the Open border controller, but for the Potential field controller, they woul still get and input to move.

a downside with this controller is that more nodes behind the leader doesn't neccessarely mean more push on the leader. This is the opposite than for the Potential field controller which is based on a push action. The more nodes behind someone results in more push, which lets the nodes get pushed and spread even if there are no open space. The bigger push also resulted in faster coverage time for the Potential field controller, but to the cost of more nodes used. There was also a big difference in number of nodes used when the shape of the room changed. In the square room the Open border and the Potential field controller used an average of 46 and 57 respectively to cover 531.1 m^2, while in the C-shaped room the used and average of 50 and 75 to cover 542 m^2. The Open border controller is very consistent in how many nodes needed to cover a certain amount of area, while the Potential field is very dependent on shape of the room. A room with lots of corners could mean that the Potential field controller uses an unnecessary amount of nodes to cover the area.

Comparing fig. 4.3 with fig. 4.8 the plots are very similar with a constant time difference. The surface plot for the Open border controller in fig. 4.3 seem to just have been shiftet up 1500 timesteps to the surface plot in fig. 4.8a. The same shows for the Potential field controller when the sensor radius is small, but for bigger sensor radius it gets slower in the room with the obstacle.

5.2 Future work

In no experiments did all the nodes get used. There was always more to push into the rooms. An interesting experiment would be to limit the number of nodes to only half of what was used to see what would happen when there are no more nodes to enter the room. Another experiment could be to have all the nodes already inside the room in a cluster to

see which controller spreads them fastest. If the Open border controller is faster to spread them on limited resources a potential new controller could be a combination of the two controllers splitted in stages. The first stage of the deployment uses the Potential field controller to enter the neccessary amount of nodes quickly and get an inital spread before the Open border controller takes over and does the final spread.

The result that entering 3 nodes at the time gave faster coverage than entering 4 at the time was an interesting result. Entering 4 at the time gives a bigger inital spread of the group but 3 was still faster. This might show that the formation of the entering nodes is a prefered formation. A way going forward would be to test with different formations of the entering groups to see if they spread in amore optimal way. The formation of entering groups may also vary in initial velocity.

In this thesis the simulations were done on a very simplified system, same as the proof of the Open border controller. Simulations using simplified systems are a good way to test concepts, but to ensure the methods will work on a physical systems, a more realistic system model must be ued. A natural next step from this work would be to implement a more realistic system dynamics and include the kinematics of nodes to see if the results still hold.

A more realistic model of the system could also be excluding the communication between the nodes and make them solely dependant on the detection sensor to spread. The expectation is that system properties such as area coverage will remain, but the static configuration will be with shorter distance between the nodes.

Chapter 6

Conclusion

In this thesis a simple strategy for entering a swarm of nodes into a room have been tested combined with multiple control laws. The objective was to see how different variables, including environmental, affected the performance. The performance metrics used were the time used to reach a certain amount of coverage and how many nodes used to reach that coverage.

The experiments showed that the controllers reacted differently to the variables. In terms of covering the area in least amount of time the Open border controller was fastest when the radius was big, but with smaller sensor radius the Potential field controller was the fastest. The Potential field controller always used more total number of nodes, and in complex environments it used more nodes to reach the same amount of area in open environments. The Open border controller was very consistent in the number of nodes used to reach a certain amount of area no matter the environment.

Entering nodes in bigger group sizes lead to faster coverage, but for the Potential field controller it came at the cost of using more nodes. The Open border controller was very consistent in number of nodes used and was always better at restricting used resources.

The results in this thesis demonstrate that there are multiple variables that affect which controller is fastest to cover the area, both environmental variables and properties of the nodes. In terms of resource constraint the Open border was far better to not over use the drones.

Bibliography

Balchen, J.G., Andresen, T., Foss, B.A., 2016. Reguleringsteknikk. NTNU grafisk senter.

Batalin, M.A., Sukhatme, G.S., 2002. Spreading out: A local approach to multi-robot coverage. Distributed Autonomous Robotic Systems 5, 373–382.

Bayindir, L., 2016. A review of swarm robotics tasks. Neurocomputing 172, 292–321.

Ben-Ari, M., Mondada, F., 2017. Elements of Robotics. Springer.

David, P., Mike, D., Regina, E., Michael, H., Craig, L., 2001. Pheromone robotics. Autonomous Robots 11, 319–324. doi:10.1023/A:1012411712038.

Douglas, G., 1992. Command control for many-robot systems .

Howard, A., Mataric, M., Sukhatme, G., 2002. Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. Distributed Autonomous Robotic Systems 5, 299–308.

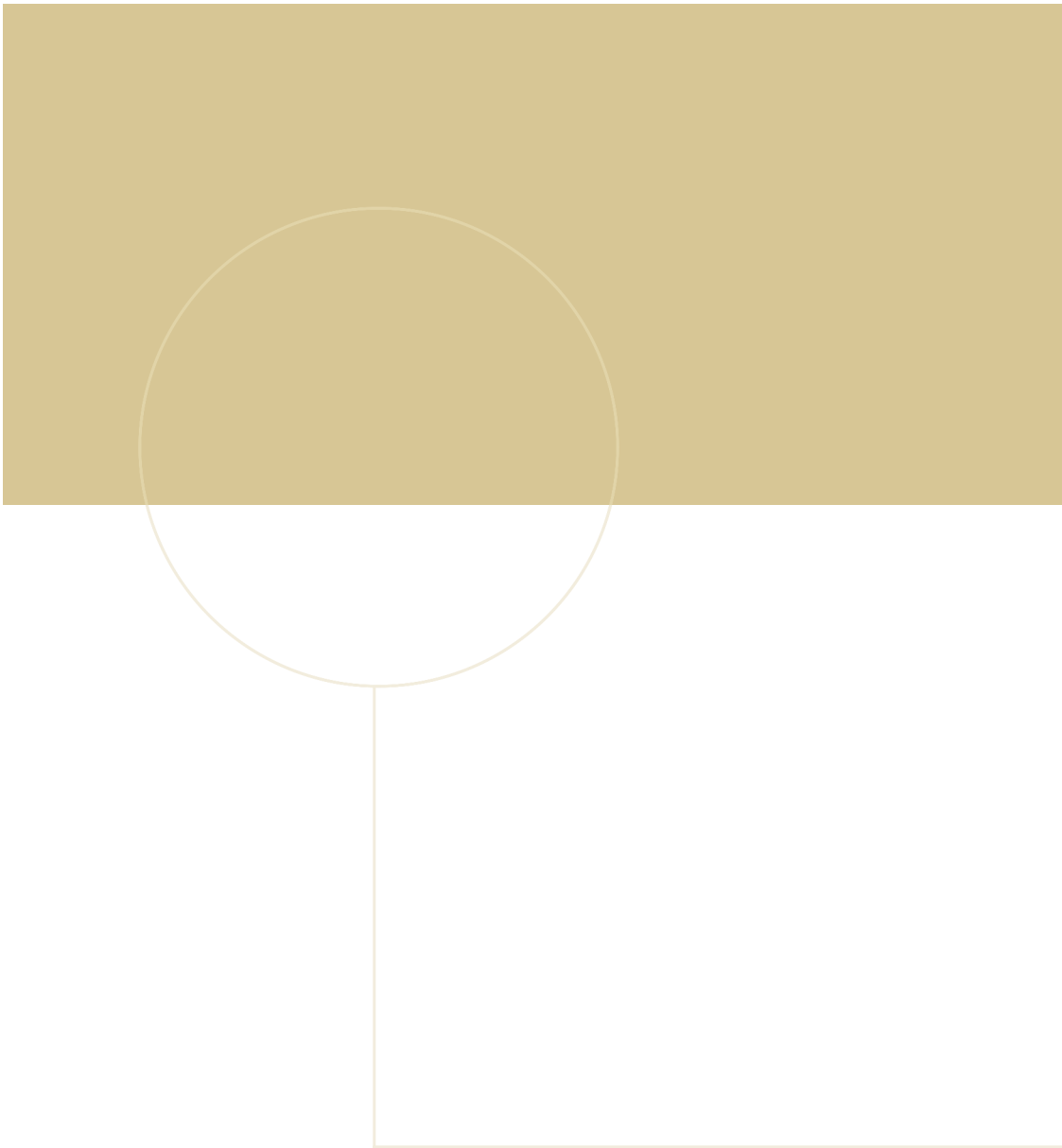
Hsiangl, T.R., Arkinl, E.M., Benderl, M.A., Fekete2, S.P., Mitchelll, J.S.B., 2004. Algorithms for rapidly dispersing robot swarms in unknown environments. Algorithmic Foundations of Robotics 7, 77–93.

Mumm, M., 2004. Voronoi diagrams. The Mathematics Enthusiast 1, 44–55.

O, K., 1986. Real-time obstacle avoidance for manipulators and mobile robots. International Journal of Robotics Research 5, 90–98.

Sotiris, P., Anthony, T., 2018. Theoretical and Experimental Collaborative Area Coverage Schemes Using Mobile Agents. doi:10.5772/intechopen.78940.

W., W.E., 2000. Circle-circle intersection. URL: https://mathworld.wolfram.com/Circle-CircleIntersection.html.



NTNU

Kunnskap for en bedre verden