Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

Erlend Lone

Incremental self-deployment using virtual potential fields

Enabling localization of first responders by constructing a mesh network

Master's thesis in Cybernetics and Robotics
Supervisor: Damiano Varagnolo
Co-supervisor: Claudio Paliotta
May 2021

NTNU
Kunnskap for en bedre verden

Erlend Lone

Incremental self-deployment using virtual potential fields

Enabling localization of first responders by constructing a mesh network

Master's thesis in Cybernetics and Robotics
Supervisor: Damiano Varagnolo
Co-supervisor: Claudio Paliotta
May 2021

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

Abstract

This thesis investigates how Micro Indoor droNes (MINs) can be incrementally deployed into an unknown environment for enabling the localization of First Responders (FRs). FRs are the first to enter a disaster area and put their own lives at risk for saving others. As many disaster areas are Global Navigation Satellite System (GNSS) denied environments, the positions of the FRs are not available. The goal of the MINs is therefore to deploy into the unknown environment to form a mesh network that enables localization of the FRs. If the FRs are equipped with a device that can communicate with the MINs, and that the network is connected to a base station placed outside the GNSS denied environment, the base station will be able to generate a good estimate of the positions of the FRs.

Two deployment schemes for the MINs are proposed. As the previous research on incremental deployment is very limited, the first scheme considers deployment in 1D. A repelling potential field is defined, aiming to ensure that the deploying MIN explores parts of the real line that has not been explored prior to its deployment.

The second deployment scheme introduces a novel procedure for calculating an exploration direction and combines this with the classical virtual potential field method. The deployment scheme assumes that the environment is unknown, and depends solely on the limited local knowledge of each MIN. This limited knowledge is utilized to the fullest so that the deploying MIN follows an obstacle free path until it reaches the latest deployed MIN and starts exploring. The way the deployment scheme is constructed will ensure that the network of MINs will be connected when the deploying MIN finishes its deployment.

Sammendrag

Denne avhandlingen undersøker hvordan mikrodroner kan trinnvis bli sendt inn i et ukjent område for å muliggjøre lokalisering av førsterespondere. Førsterespondere er de første som entrer et katastrofeområde og setter sine egne liv i fare for å redde andre. Siden mange katastrofeområder er utilgjengelige for globale satellittbaserte systemer for navigering og posisjonering, vil ikke posisjonene til førsteresponderene være tilgjengelige. Målet for mikrodronene er derfor å deployere inn i det ukjente området for å danne et mesh-nettverk som muliggjør lokaliseringen av førsteresponderene. Dersom førsteresponderene er utstyrt med en enhet som kan kommunisere med mikrodronene, og nettverket er koblet til en basestasjon som er tilgjengelig for global posisjonering, så vil basestasjonen kunne estimere posisjonene til førsteresponderene.

To metoder for å sende mikrodronene inn i det ukjente området er foreslått. Siden tidligere forskning innenfor trinnvis deployering er veldig begrenset, tar den første metoden for seg deployering i 1D. Et frastøtende potensialfelt er definert, med det mål om å sørge for at mikrodronen som for øyeblikket blir deployert utforsker deler av den reelle linjen som ikke har blitt utforsket før.

Den andre metoden introduserer en ny prosedyre for å finne en retning som skal undersøkes og kombinerer den med klassisk bruk av potensialfelt. Deployeringsmetoden antar at området er ukjent og avhenger bare av den begrensede lokale kunnskapen til hver mikrodrone. Denne begrensede kunnskapen er fullt utnyttet slik at den mikrodronen som for øyeblikket blir deployert følger en kollisjonsfri bane til den når frem til den forrige deployerte mikrodronen og begynner å utforske videre. Måten deployeringsmetoden er bygget opp vil sørge for at nettverket av mikrodroner vil være sammenkoblet når hver drone fullfører sin deployering.

Preface

Problem description

FRs are the first to enter the emergency scene in disaster situations, exposing themselves to extreme risks. Developing tools that assist the FRs in these dangerous situations will not only enable them to work more efficiently, but make sure that their work can be done in a safer manner. By introducing different types of autonomous robots equipped with a variety of sensors, the overall perception the FRs have of the emergency scene can be greatly enhanced [1].

The disaster scenes considered in this thesis will mainly take the form of semi-collapsed buildings. This leads to that the localization of the FRs becomes a challenge, as these types of scenes typically are GNSS denied environments. With this in mind, the project will investigate how MINs can be deployed with the aim of enabling the localization of the FRs. Assuming that the FRs are equipped with a device that can communicate with the MINs, the position of the FRs relative to the mesh network can be estimated. If at least one MIN is within range of a base station that is located outside the GNSS denied environment, a fairly good estimate of the global position of the FRs can be made.

The deployment of the MINs has to take the limited sensing and computation capabilities of the drones into account. A method that has been used extensively for deployment and obstacle avoidance in mobile robotics is the *virtual potential fields method* [2]. The main advantages of this method is that it does not require that the robots have thorough knowledge about their surroundings and constructing the potential field is not computationally expensive. The robots typically form a potential field by using information about the inter-robot distances and measured distance to obstacles in the environment. The movement of the robots is decided by the force that the robots experience as they are placed in the virtual potential field.

Deploying the MINs incrementally will reduce the complexity of the deployment in the sense that each MIN does not have to consider the trajectories of the other MINs while flying. Another benefit of deploying the drones one-by-one is that the already deployed drones may aid the navigation of the currently deploying one. An example of this is letting the currently deploying drone follow a path that already has been taken by previously deployed drones. This will lead to that the currently deploying drone follows a collision-free path up until it reaches the last deployed drone.

The main tasks for this thesis can be summarized as:

1. Propose an exploration strategy for MINs that are incrementally deployed into an unknown environment using virtual potential fields
2. Ensure that the MINs construct a mesh network to aid the localization of the FRs

Work description

The work in this thesis started out as a continuation of my semester project [3]. The goal of that project was to investigate how a network of mobile agents with perfect circular sensing areas could maximize their combined coverage while remaining connected. As I started working more closely with the INGENIOUS project at SINTEF than what was the case for the semester project, the aim shifted from area-coverage to the construction of a mesh network. These two problems are related in the sense that the maximum coverage is typically achieved when the agents are distributed uniformly, and a network operates most energy efficienly when its nodes are uniformly distributed in the area of interest.

Assuming that the drones that are used in the INGENIOUS project have perfect information of the area around itself is not feasible. The sensors mounted on the drone are range sensors with a limited field of view. INGENIOUS also aims to deploy the drones in unknown environments, which impose further constraints of the knowledge that is available to the drones.

These restrictions introduced many new challenges compared what was investigated in the semester project. The fact that the environment is unknown and that the drones will not form any kind of map of the area, results in that the movement of the drones must be decided based on the local knowledge of each drone. Restricting the sensing information of the drones means that only linear distance measurements within a limited field of view are available. The reduction in sensor field of view leads to that each drone is unable to sense obstacles in some directions relative to itself, leading to an increased risk of collision.

All the removed assumptions mean that the performance of the deployment had to be reformulated. Defining a new performance metric was therefore also an important challenge that needed to be addressed.

Acknowledgements

I have to thank my supervisors Damiano Varagnolo and Claudio Paliotta for all the help they have given while I have been working on this project. Their feedback and advice have been invaluable. I also have to thank Magnus Berdal that has been investigating exploration strategies of MINs parallel to my work. Our discussions on how to solve different problems have been greatly appreciated.

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme and the Korean Government under Grant Agreement No 833435. Content reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

Erlend Lone Trondheim, 31.05.2021

Contents

List of Tables

List of Figures

Acronyms

DSSA	Distributed Self Spreading Algorithm
EN	East-North
EU	European Union
FR	First Responder
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
INGENIOUS	Next-GENeration IoT sOlutions for the Universal Supply chain
LiPo	Lithium Polymer
MIN	Micro Indoor droNe
NGIT	Next Generation Integrated Toolkit
PA	Power Amplifier
RSS	Received Signal Strength
SAR	Search and Rescue
SCS	SINTEF Control Station
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
USB	Universal Serial Bus

Nomenclature

$i.j$,k	Generic indices
ν_i	Agent i
ν_0	Used to denote the SINTEF Control Station
ν_n	The latest deployed agent
ν_{n+1}	The currently deploying agent
x_i	Position of ν_i in 1D
$\mathbf{x}_i$	Position of ν_i in 2D
$\mathcal{N}_i$	Set consisting of the indices of all neighboring agents of ν_i. ν_i and ν_j are neighbors if and only if $\xi_{i,j} \geq \xi_\tau,\ i \neq j$
$\xi_{i,j}$	The the inverse RSS distance between ν_i and ν_j given as $\xi(d_{i,j})$
$d_{i,j}$	Euclidean distance between ν_i and ν_j
τ	Subscript used for denoting a threshold
ξ_τ	The inverse RSS distance threshold for what is considered neighbors
d_τ	Distance between two agents that satisfy $\xi(d_\tau) = \xi_\tau$
$\mathcal{U}$	Used to denote uniformity

Chapter 1

Introduction

INGENIOUS is an European Union (EU) project that aims to assist FRs in crises and disaster scenarios. The project includes the development, integration, testing, and validation of the Next Generation Integrated Toolkit (NGIT) for collaborative response. The use of smart uniforms, boots and helmets, augmented reality, data intelligence and drone swarms are just some of the tools and services the project wants to implement in what is called The First Responder of the Future. The main task that SINTEF has been assigned as part of this project is the construction and development of the MIN platform. The MINs will form a mesh network, and each MIN will be equipped with sensory capabilities enabling the network to localize the FRs within the perimeter.

The MIN that is used in the project is Crazyflie 2.1, depicted in fig. 1.1. This is a small, lightweight drone developed and manufactured by Bitcraze. In the initial prototyping phase, SINTEF controls the CrazyFlies by running Python scripts on a laptop that acts as a base station, communicating with the drones through the Crazyradio PA. Crazyradio PA is a long-range open USB radio dongle based on the nRF24LU1+ from Nordic Semiconductor. The MINs should be able to be deployed in GNSS denied areas, where it is not feasible to assume that the base station would be able to communicate with each MIN throughout the whole deployment. The long-term goal is therefore that the control of the network is implemented through the embedded software running on each individual drone.

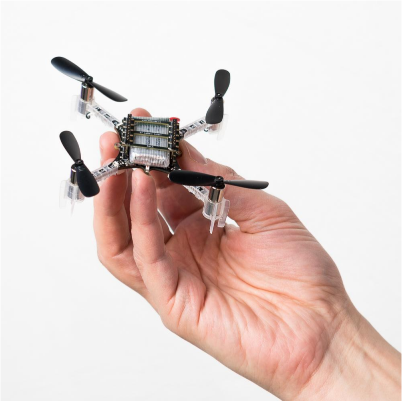


Figure 1.1: Crazyflie 2.1.[1]

1.1 Motivation

Making the use of Unmanned Aerial Vehicles (UAVs) in Search and Rescue (SAR) missions more efficient have been put forward by several researchers [4–6]. In [4], Alotaibi et al. propose a new technique where the SAR tasks are distributed among multiple UAVs in an attempt to save as many victims as possible in the shortest amount of time. Rivera et al. implements a human detection and geolocation system in outdoor environments [5]. The detection system consists of two cameras, an optical camera that is used during daytime, and a thermal camera aids the detection during nighttime. The data from the detection system is combined with GPS data for localizing the detected persons. Kulkarni et al. use reinforcement learning in [6] with the aim that a single UAV should localize a victim that is trapped in an indoor environment as fast as possible.

When used in SAR missions, the goal of the UAVs is traditionally to locate persons in imminent danger. An advantage of using MINs instead of larger UAVs for SAR missions inside partially collapsed buildings and other disaster areas, is that MINs are a lot smaller, and are therefore able to enter tighter areas than traditional UAVs.

Technological advances within data processing and drone technology in general, combined with the increased energy density in LiPo batteries, has enabled the development of swarms of collaborating MINs. The improvements within battery technology has lead to increased flight time, which for a long time was - and still are - the main limitation of multirotor drones.

[1] https://store.bitcraze.io/products/crazyflie-2-1

1.2 Thesis contribution

FRs have to work in unknown GNSS denied environments. In the absence of GNSS signals there is no system that provide the locations of the FRs. The idea that is studied in this thesis is that MINs are deployed incrementally into mission area to form a mesh network. As the MINs reach their final configuration, they function as beacons and could support the localization of the FRs. Assuming that the FRs are equipped with receiving tags, the relative position of the FRs can be estimated. If at least one MIN in the mesh network is within the range of a base station that is placed outside the GNSS denied area, the global position of the FR can be estimated fairly well. This allows the FRs to work more efficiently, and those who are in charge of the SAR mission will have a better picture of where the different FRs are located.

Previous research on for incrementally deploying mobile agents into a mission area is extremely limited. Most authors either assume that the robots are deployed all at once or that the robots initially are distributed randomly in the mission area. The majority of the methods that are proposed for mobile robots tries to solve the area-coverage problem assuming that the robots have perfect circular sensing areas. In this project however, the drones have limited sensing information, and each drone must utilize its local knowledge so that all the drones collectively form a mesh network that enables localization of FRs. This thesis introduces therefore a novel deployment scheme that can be expanded within several branches within mobile robotics.

1.3 Report outline

Chapter 2 starts of by providing the necessary background theory needed to get an introduction to how the virtual potential fields methodology works. After the background theory is presented, chapter 2 proceeds to give a detailed description of previous work that has been conducted within multi-robot systems, incremental deployment and the using potential fields within robotics. Chapter 2 is concluded by a presentation of how the system is modeled in this thesis.

Chapter 3 presents results for an exploration case in 1D. As the previous work on initial incremental deployment in very limited, the 1D case was considered as a starting point, building a theoretical first approach in an oversimplified environment. The idea was to start from a simple case and proceed to build upon this by adding complexity to the problem.

Chapter 4 expands the mission area from 1D to 2D. Increasing the dimensionality posed several new challenges and implied that the approach build in the 1D case had to be adapted. Chapter 4 gives a thorough description of the new proposed deployment scheme, and its performance is then analyzed.

Chapter 5 concludes the thesis by summarizing the work. The results are discussed in detail, and suggestions on how the novel deployment scheme can be improved in the future is presented.

Chapter 2

Background

2.1 Background theory

In the subsections below, the terms “agent”, “sensor node”, “mobile robot”, etc. are used interchangeably. This is done in an attempt to use the same wording as the referenced articles.

2.1.1 Virtual potential fields

Constructing virtual potential fields is a common technique for deploying sensor nodes as well as for obstacle avoidance in mobile robotics. The goal is that the nodes that are placed in the field will be repelled from areas that might cause harm to the them, and possibly attracted to areas that are considered important [2]. In most cases, the total potential field repels each agent from obstacles and other agents in the mission area. If a target position is known to the agents, the total potential field will also contain an attractive component, which pushes each agent toward its target position. This means that the total potential field will usually be $U_{tot} = U_{rep} + U_{att}$. All obstacles within a given threshold distance, d_o, will contribute to the repelling field, which can be written as

$$U_{obs,j} = \begin{cases} \frac{1}{2}k_{obs}\left(\frac{1}{\|\mathbf{r}_{i,j}\|} - \frac{1}{d_o}\right)^2 & \text{if } \|\mathbf{r}_{i,j}\| \leq d_o \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

leading to

$$U_{rep} = \sum_j U_{obs,j} \tag{2.2}$$

k_{obs} is a positive constant and $\|\mathbf{r}_{i,j}\| = \|\mathbf{x}_i - \mathbf{x}_j\|$ is the shortest distance between agent i and the jth obstacle..

The attractive field is traditionally defined as

$$U_{att} = \frac{1}{2} k_{att} \left\| \mathbf{x}_i - \mathbf{x}_g \right\|^2 \tag{2.3}$$

where k_{att} is a positive constant and $\mathbf{x}_g$ is the target position.

As the agents are placed in the virtual potential field, they will experience a force that is defined as

$$\mathbf{F}_{tot} = -\nabla U_{tot} \tag{2.4}$$

∇ is a linear operator, and it can therefore readily be seen that $\mathbf{F}_{tot}$ can be written as

$$\begin{aligned} \mathbf{F}_{tot} &= -\nabla U_{rep} - \nabla U_{att} \\ &= \mathbf{F}_{rep} + \mathbf{F}_{att} \end{aligned} \tag{2.5}$$

The repulsive force that agent i experiences will be

$$\mathbf{F}_{rep} = \sum_j \mathbf{F}_{obs,j} \tag{2.6}$$

where

$$\mathbf{F}_{obs,j} = \begin{cases} -k_{obs} \frac{1}{\|\mathbf{r}_{i,j}\|^2} \left(\frac{1}{\|\mathbf{r}_{i,j}\|} - \frac{1}{d_0} \right) \frac{\mathbf{r}_{i,j}}{\|\mathbf{r}_{i,j}\|} & \text{if } \|\mathbf{r}_{i,j}\| \leq d_o \\ 0 & \text{otherwise} \end{cases} \tag{2.7}$$

The attractive force acting on agent i is expressed as

$$\mathbf{F}_{att} = -k_{att}(\mathbf{x}_i - \mathbf{x}_g) \tag{2.8}$$

2.2 Previous work

2.2.1 Multi-robot systems

The goal of multi-robot systems is to either execute a task more efficiently than what a single robot is able to do, or perform a mission that simply is not possible to accomplish using a single robot. Cooperation between the individual robots that constructs the multi-robot system is crucial for evaluating whether a multi-robot system performs better than what an equivalent system consisting of a single robot could. Cooperative robotics is the studied in [7, 8].

Swarm robotics is a sub-domain of multi-robot systems, and has some main characteristics that distinguishes it from general multi-robot systems. According to [9], these are:

- robots are *autonomous*;
- robots are *situated* in the environment and can act to modify it;
- robots' *sensing and communication capabilities* are local;
- robots do not have access to *centralized control* and/or *to global knowledge*;
- robots *cooperate* to tackle a given task.

In [10] Barca and Sekercioglu identifies challenges one is faced with when designing a swarm robotics systems for real-life implementation. The highlighted challenges are:

1. Selecting appropriate centralized or decentralized communication and control schemes.

2. Incorporating important behaviours and traits such as self-organization, scalability and robustness.

3. Devising mechanisms that support goal-directed formations, control and connectivity.

4. Implementing mapping, localization, path planning, obstacle avoidance, object transport and object manipulation functions that enable swarms of robots to interact efficiently with the environment.

5. Addressing problems related to energy consumption.

Barca and Sekercioglu describe how the challenges have been addressed i previous research, and come to the conclusion that in most cases, only some of the challenges have been solved. They proceed to state that in order to successfully implement a swarm robotics system in real life applications, one has to address larger number of challenges simultaneously.

A common task for swarm robotics is area coverage. There are mainly three different approaches to area coverage using a swarm of mobile agents: *blanket coverage*, *barrier coverage* and *sweep coverage* [11]. Barrier coverage is distinguished from blanket and sweep coverage in the way that it aims to prevent unwanted penetration through the barrier constructed by the swarm, whereas blanket and sweep coverage aim to maximize the probability that some event in the mission area is detected. Sweep coverage is characterized by the fact that the swarm moves continuously, while blanket cover aims to reach a static equilibrium that maximizes the covered area.

A more recent study from 2017 [12], defines multiple sub-categories of coverage. For this thesis however, it is sufficient to compare the definitions of blanket and sweep cover found in [11]. Examples of the blanket cover approach are found in [13–17]. Repeated sweep coverage is investigated in [18–21], where the authors try to mimic how ants produce pheromones for navigating the swarm. The robots leave chemical markings that evaporate with time. Each robot can sense the intensity of the chemical marks in its neighborhood, and based on these measurements, each robot decide what area that should be covered/explored next. Tasks that require repeated coverage are typically mine sweeping, surveillance and search and rescue missions [18].

For a comprehensive review of swarm robotics, the reader is referred to [22].

2.2.2 Potential fields

After being introduced by Khatib in [2], virtual potential fields have been used extensively for obstacle avoidance in mobile robotics due to its simple structure and the small amount of calculations needed.[1] Even though the method is simple to implement, it also has some inherent limitations. According to [25] two such limitations are the presence of local minima and that the agents are not able to pass between closely spaced obstacles. Khatib developed the potential field approach with the intent that it should be used for on-line obstacle avoidance in situations where a robot does not have prior knowledge about the obstacles in its configuration space. The robot would construct a virtual potential field based on what

[1] Khatib actually introduced the potential field approach as early as 1980 in [23], but [2], that was published five years later, is the first paper that describes the approach comprehensively [24].

obstacles it sensed at any given time.

For reference, the repulsive and attractive fields defined in [2] are stated. The total artificial potential field is the sum of the two.

The repulsive field is

$$U_O(\mathbf{x}) = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{\rho} - \frac{1}{\rho_0}\right)^2 & \text{if } \rho \leq \rho_0 \\ 0 & \text{if } \rho > \rho_0 \end{cases} \tag{2.9}$$

where η is a positive constant gain, ρ is the shortest distance between the robot and an obstacle, and ρ_0 denotes the limit distance the the potential field has influence on the robot.

The attractive field is

$$U_{\mathbf{x}_d}(\mathbf{x}) = \frac{1}{2}k\left(\mathbf{x} - \mathbf{x}_d\right)^2 \tag{2.10}$$

where k is a positive constant gain and $\mathbf{x}_d$ is the goal position of the robot.

The majority of other previous research that has been done for deploying mobile agents into an area using the potential field method either adopts the original approach from Khatib, or propose improvements by changing the repulsive field. The attractive potential field is in most cases the same as what Khatib used. However, in [26] Mac et al. proposes modifications of the repulsive field as well as the attractive one. They add a term including the velocity of the robot to the attractive field and multiplies the original repulsive field by the distance between the agent and the goal position. These modifications result in the new potential fields defined in eqs. (2.11) and (2.12). This ensures that the risk of being trapped in a local minimum is avoided. The latter modification is also performed in [27, 28].

$$U_O^*(\mathbf{x}) = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{\rho} - \frac{1}{\rho_0}\right)^2\left(\mathbf{x} - \mathbf{x}_d\right)^2 & \text{if } \rho \leq \rho_0 \\ 0 & \text{if } \rho > \rho_0 \end{cases} \tag{2.11}$$

$$U_{\mathbf{x}_d}^*(\mathbf{x}, \dot{\mathbf{x}}) = \frac{1}{2}k_p\left(\mathbf{x} - \mathbf{x}_d\right)^2 + k_v\dot{\mathbf{x}}^2 \tag{2.12}$$

Contrary to the approaches described above, Howard et al. construct only repulsive potential fields in [13]. The agents in the network are repelled from obstacles in the environment and each other. This approach can be used in situations where the each agent does not have a clear target position and the main goal of the network is to spread out evenly through the environment. As this sounds like a similar problem as the problem this thesis studies, it should be noted that Howard et al. deploys all agents at once. Unfortunately, using their solution in an incremental deployment scheme where agents are stationary once they finish their deployment would not be applicable. This is due to the fact that deployed agents would easily form a "wall" that the next deploying agent is not able to pass. This is illustrated in fig. 2.1. The forces from ν_1, ν_2 and ν_3 will push ν_4 towards the lower left corner, and the forces from the walls and ν_0 will push it towards the upper right corner. ν_0 will not be able to pass the red "wall" as the forces will cancel each other out.

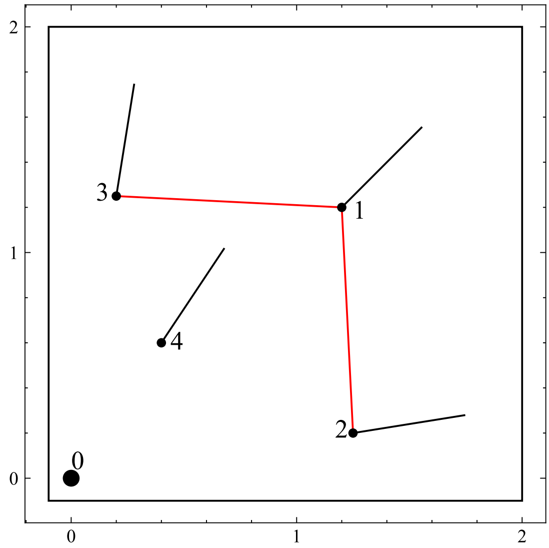


Figure 2.1

2.2.3 Incremental deployment

Initial incremental deployment of sensor nodes is not as thoroughly researched as deployment strategies where all nodes are deployed at the same time. Both [29] and [30] does however study incremental deployment but they assume that the mission area contains sensor nodes prior to the execution of the incremental deployment algorithm. In [29] an *on-demand* strategy is designed, for which a new sensor node is deployed when the fraction of the functioning sensor nodes in the network drops below a threshold. The strategy therefore also takes node failure into account, and aims to maximize the fraction of the sensor network that is operative at all times. Node failure is also considered in [30], where it is assumed that the energy level of each node is known. Based on this information, the minimum number of nodes that have to be deployed for preventing a reduction in covered area is calculated, and where they should be positioned.

As stated in [31, 32], the initial deployment of mobile robots is not a topic that has been researched thoroughly. In most cases it is assumed that the robots already are placed in the mission area and then have to spread or achieve a given formation [14, 17, 33, 34]. This is not the case for [35], where a node is assigned to be an *anchor* that provides a starting point for the other nodes. The nodes are deployed in an unknown environment, and the goal of the proposed algorithm is to sequentially deploy nodes into the area as long as there are nodes available. After each iteration, the target point of the next deploying agent is computed based on four policies. Two of which are based on randomness, and the other two select a target position that maximizes the potential of increasing the covered area. Even if the drones studied in this thesis only have knowledge about

In [36], Lin et al. propose methods for deploying sensor networks using one or multiple agents to deploy sensors incrementally. The first sensor is deployed in the target area using some heuristics. Each agent computes the *Voronoi diagram* of the target area using the positions of the deployed sensors as *site points*. Voronoi diagrams are partition sets consisting of *Voronoi regions*, and Voronoi regions are generated using site points. Each Voronoi region consists of all points that are closer to its associated site point than to any other site point. The Voronoi region associated with the site point $\mathbf{p}_i$ can be expressed as

$$V_i = \{\mathbf{x} \in \Omega \mid \|\mathbf{x} - \mathbf{p}_i\| \leq \|\mathbf{x} - \mathbf{p}_j\| \; \forall \, j \in [1, k], \; j \neq i\} \tag{2.13}$$

Where Ω denotes the target area and $\mathbf{p}_j$ denotes any other site point than $\mathbf{p}_i$ located in Ω. This will result in $V_i \cap V_j = \emptyset$ for $i \neq j$ and that the Voronoi diagram will then be $\{V_i\}_{i=1}^{k}$. A more detailed description of Voronoi diagrams are found in [37]. [38, 39] are some examples where Voronoi diagrams are used for area coverage.

After each agent has computed the Voronoi diagram of the target area, it computes its moving target point based on whether or not the Voronoi region that the agent is located in is fully covered by sensors. If it is not covered, the agent is assigned a target point based on the point in the Voronoi region with lowest detection probability. If the Voronoi region on the other hand is fully covered, the agent moves to an adjacent Voronoi region and is assigned a target point in the same manner as described. The agent moves towards the target point as long as it is within communication range of two sensors. New sensors are deployed sequentially using this scheme until the area is fully covered.

An interesting approach that has been implemented in a real-life environment is found in [40]. Rybski et al. use Unmanned Ground Vehicles (UGVs) (*rangers*) that carries multiple small ground robots (*scouts*) into the mission area. When the rangers have reached their target positions, they deploy the scouts sequentially using a *launcher*. The launcher system deploys the scouts into the mission area by a compressed spring, and a platform that selects the elevation angle at which the scouts are deployed. The idea of using a UGV as a utility platform for the drones, and deploy the drones sequentially would be very interesting to pursue when implementing the problem studied in this thesis in real life. How the drones are transported to the starting point of the deployment is however beyond the scope of this thesis.

2.3 Model of the system

2.3.1 Agent

The work done in this thesis has been conducted with development of the Crazyflie 2.1 drone in mind. In an attempt to keep the discussions generic, the drones will be referred to as *agents* in the remaining chapters. It should be noted, however, that the deploying agent is allowed to move “above” an already deployed agent.

2.3.2 Dynamics

Single integrator dynamics are described as

$$\dot{x}_t = u(t) \tag{2.14}$$

where $\dot{x}_t$ is the time derivative of the state of the system at time t and $u(t)$ is the input at time t. If the state is defined as the position of the system, the input in eq. (2.14) will directly control the system’s velocity. Newton’s second law states that the sum of all forces acting on an object equals the mass of the object times its acceleration.

$$\sum F = ma \tag{2.15}$$

For the purpose of evaluating the overall properties of the schemes proposed in chapters 3 and 4, the agents are modeled as unitary point masses with little to no inertia. Assuming also that the agents are equipped with powerful actuators and that lower controllers handles any inherent dynamics, eqs. (2.14) and (2.15) can be combined to form a simplified model of the system [41]

$$\dot{x}_t = u(t) = \sum F \tag{2.16}$$

The position is updated by using Euler’s method, given as

$$x_{t+1} = x_t + \dot{x}_t dt \tag{2.17}$$

It should be clearly stated that these models are preliminary, and considering more sophisticated models will be a topic of future work.

2.3.3 Sensing model

The Crazyflie 2.1 drones have five VL53L1x time-of-flight laser sensors mounted in five orthogonal directions (front/back/left/right/up). As this thesis only considers deployment in 1D and 2D, the sensor that points upwards is not taken into account. The VL53L1x sensors have a maximum field of view of 27° and a range of $4 - 400\ \mathrm{cm}$. In the simulation environment, each of the range sensors are discretized so that it consists of a total of five rays. One ray points in the direction the sensor is mounted relative to the Crazyflie 2.1, and two rays are spread out evenly on each side of the one pointing in the sensing direction. For simplification purposes, it is also assumed that the range sensors return the angle where the closest obstacle was detected, and not only the distance reading. Figure 2.2 shows the directions the sensors are mounted and how the field-of-view is discretized.

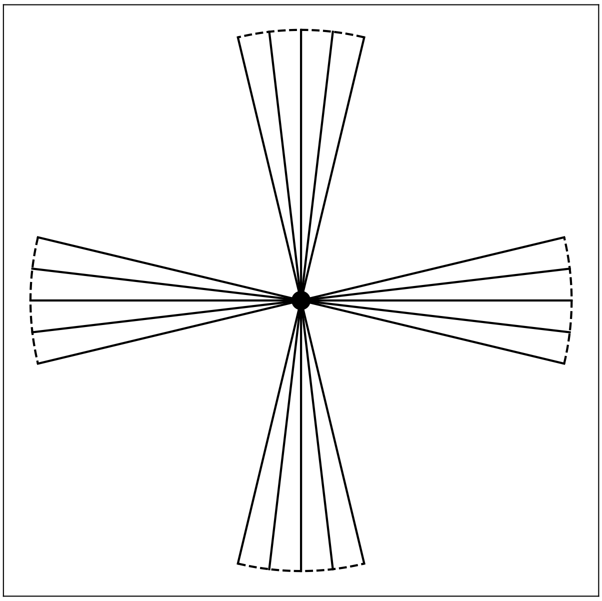


Figure 2.2: Discretization of sensor field of view

2.3.4 Inverse RSS distance

Signals between two agents will experience fast fading, shadowing and path-loss [42]. Modeling the path-loss under log-normal shadowing has been used in [42–44] for predicting the received signal strength when the receiver is placed a given distance from the sender. [43] also propose several estimators for estimating the distance a signal has travelled, given the RSS measurement.

An accurate model of the RSS and distance estimation is outside the scope of this thesis, and it is therefore assumed that the RSS is described purely in terms of the distance $d_{i,j}$ between ν_i and ν_j. Further, the RSS is mapped to an *inverse RSS distance*. This distance is denoted $\xi_{i,j}$ and $\xi_{i,j} \in [0, \overline{\xi}]$, where $\overline{\xi}$ is a constant. The mapping is assumed to be monotonically decreasing with respect to $d_{i,j}$, which can be expressed as

$$\frac{\mathrm{d}\xi_{i.j}}{\mathrm{d}d_{i,j}} = \frac{\mathrm{d}\xi_{i,j}}{\mathrm{d}RSS_{i,j}} \frac{\mathrm{d}RSS_{i,j}}{\mathrm{d}d_{i,j}} \leq 0 \tag{2.18}$$

In the simulations described in the succeeding chapters, $\xi_{i,j}$ is modeled as shown in eq. (2.19).

$$\xi_{i,j} = \begin{cases} \overline{\xi} & \text{if } d_{i,j} \leq d_{perf} \\ \frac{\overline{\xi}}{2}(1 + \cos(\omega d_{i,j} + \phi)) & \text{if } d_{perf} < d_{i,j} < d_{none} \\ 0 & \text{if } d_{none} \leq d_{i,j} \end{cases} \tag{2.19}$$

where

$$\omega = \frac{\pi}{d_{none} - d_{perf}} \tag{2.20}$$

and

$$\phi = -\frac{\pi d_{perf}}{d_{none} - d_{perf}} \tag{2.21}$$

demanding $d_{perf} < d_{none}$.

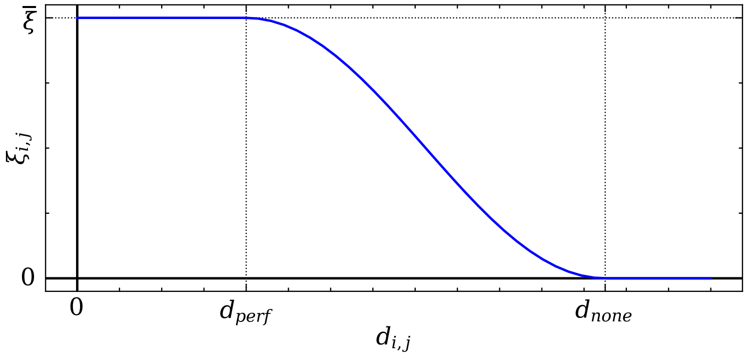


Figure 2.3: Plot of $\xi_{i,j}$ as a function of the actual distance between ν_i and ν_j

2.3.5 Reference frames

It is often useful to express positions, motions and directions relative to different reference frames. In many situations, it is convenient to represent the position of an entity relative to some fixed frame, whereas sensor measurements usually is described relative to the position and orientation of the sensor itself. Four frames are therefore defined below, which will be convenient for the 2D case studied in chapter 4.

The *inertial* frame, $\mathcal{I}$, is an East-North (EN) frame with origin at the position of the SCS, the x-axis is directed towards Earth's true east and the y-axis towards Earth's true north.

The *agent* frame of ν_i is denoted $\mathcal{A}_i$, and is also an EN frame. The agents are modeled as point masses, allowing the origin of $\mathcal{A}_i$ to be at $\mathbf{x}_i^{\mathcal{I}}$.

$\mathcal{B}_i$ denotes the *body* frame of ν_i, and has the same origin as $\mathcal{A}_i$. Its x-axis points in the heading direction of the agent, $\psi_{h,i}$. This results in that $\psi_{h,i}$ is the angle between $x^{\mathcal{B}_i}$ and $x^{\mathcal{A}_i}$, and the angle between $y^{\mathcal{B}_i}$ and $x^{\mathcal{A}_i}$ can be expressed as $\psi_{h,i} + \frac{\pi}{2}$. Figure 2.4 depicts the relationships between $\mathcal{I}$, $\mathcal{A}_i$ and $\mathcal{B}_i$.

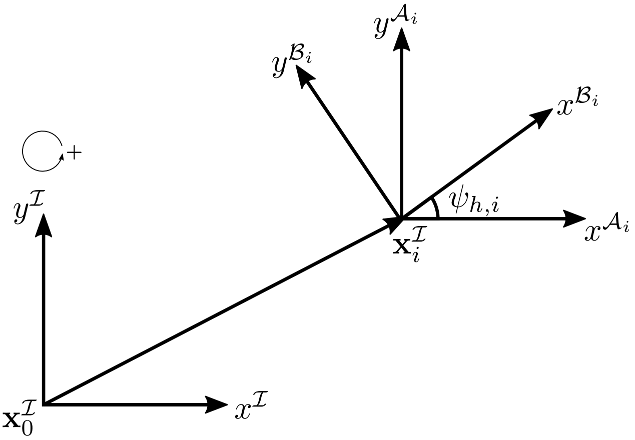


Figure 2.4: Inertial frame, agent frame and body frame of ν_i

The rotation matrix $\mathbf{R}_{\mathcal{B}_i}^{\mathcal{A}_i} \in SO(2)$ [45] defined in eq. (2.22) describes the relationship between $\mathcal{B}_i$ and $\mathcal{A}_i$.

$$\mathbf{R}_{\mathcal{B}_i}^{\mathcal{A}_i}(\psi_{h,i}) = \begin{bmatrix} \cos\psi_{h,i} & -\sin\psi_{h,i} \\ \sin\psi_{h,i} & \cos\psi_{h,i} \end{bmatrix} \tag{2.22}$$

Assuming that sensor j is mounted on ν_i, its sensor frame, denoted $\mathcal{S}_{i,j}$ has its origin at $\mathbf{x}_i^{\mathcal{I}}$. The angle between $x^{\mathcal{B}_i}$ and $x^{\mathcal{S}_j}$ is $\psi_{rel,i,j}$, which is the angle that sensor j is mounted on ν_i relative to the heading of ν_i. The angle between $x^{\mathcal{B}_i}$ and $y^{\mathcal{S}_{i,j}}$ is defined to be $\psi_{rel,i,j} + \frac{\pi}{2}$. For simplicity, assuming that there is only one sensor mounted on ν_i, fig. 2.5 shows how $\mathcal{I}$, $\mathcal{B}_i$ and $\mathcal{S}_{i,1}$ are related. $\mathcal{A}_i$ is omitted for clarity. The body frame of agent i and the frame of sensor j is described by the rotation matrix in eq. (2.23).

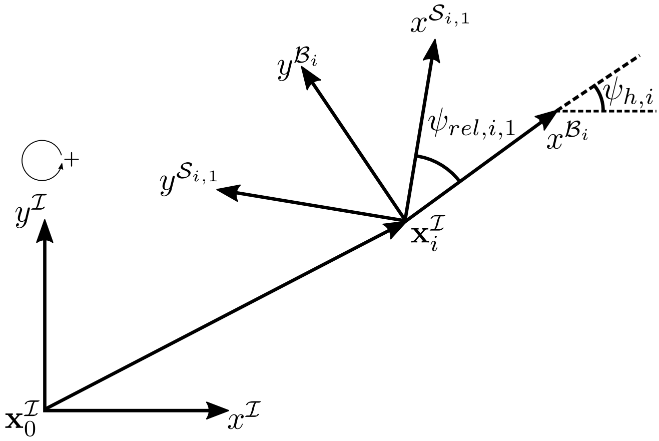


Figure 2.5: Inertial frame, body frame of ν_i and sensor frame of sensor j

$$\mathbf{R}_{\mathcal{S}_{i,j}}^{\mathcal{B}_i}(\psi_{rel,i,j}) = \begin{bmatrix} \cos\psi_{rel,i,j} & -\sin\psi_{rel,i,j} \\ \sin\psi_{rel,i,j} & \cos\psi_{rel,i,j} \end{bmatrix} \tag{2.23}$$

Chapter 3

1D

Since the previous work on initial incremental deployment is extremely limited, it was decided to start from the simplest approach, namely 1D. For simplifying the problem further, the agents were restricted to only be able to move in one direction. As the environment is assumed to be unknown prior to the deployment, no predefined goal configuration exists. A consequence of this is that the agents do not have target points they may be attracted to, and the potential field has to be designed to be purely repelling. The potential field defined in eq. (3.1) is therefore proposed in an attempt to lead ν_{n+1} to a position that ensure exploration. The exploration condition is stated as $\|x_{n+1} - x_0\| > \|x_n - x_0\|$, which in 1D simplifies to $|x_{n+1} - x_0| > |x_n - x_0|$. Assuming $x_0 = 0, x_i \geq 0 \; \forall \; i \in [1, n]$ simplifies the exploration condition further, resulting in $x_{n+1} > x_n$. κ_i and α_i are gains that are to be chosen so that the exploration condition is satisfied, and $\xi_{n+1,i}$ denotes the inverse RSS distance between ν_{n+1} and ν_i.

$$U_{n+1} = \frac{1}{2} \sum_{i \in \mathcal{N}_{n+1}} \kappa_i \left\| x_{n+1} - \alpha_i (x_i + \xi_{n+1,i}) \right\| \tag{3.1}$$

3.1 Equilibrium point

As described in section 2.1.1, agents that are placed in a potential field will experience a force that is equal to the negative gradient of the potential field. ν_{n+1} will therefore experience a force that is described as

$$\begin{aligned} F_{n+1} &= -\nabla_{x_{n+1}} U_{n+1} \\ &= - \sum_{i \in \mathcal{N}_{n+1}} \kappa_i (x_{n+1} - \alpha_i (x_i + \xi_{n+1,i})(1 - \beta_i \operatorname{sgn}(x_{n+1} - x_i))) \end{aligned} \tag{3.2}$$

where $\beta_i = \alpha_i \frac{\mathrm{d}\xi_{n+1,i}}{\mathrm{d}d_{n+1,i}}$

The equilibrium point of the force described in eq. (3.2) will be a minimum of the potential field defined in eq. (3.1), and indicate where ν_{n+1} ideally will stop. For deriving the equilibrium point of F_{n+1}, the neighbor set of ν_{n+1}, denoted $\mathcal{N}_{n+1}$, is divided into three subsets, depending on the location of ν_{n+1} relative to ν_i.

$$A_{n+1} \coloneqq \{i \in \mathcal{N}_{n+1} \mid x_{n+1} - x_i > 0\} \tag{3.3a}$$
$$B_{n+1} \coloneqq \{i \in \mathcal{N}_{n+1} \mid x_{n+1} - x_i < 0\} \tag{3.3b}$$
$$C_{n+1} \coloneqq \{i \in \mathcal{N}_{n+1} \mid x_{n+1} - x_i = 0\} \tag{3.3c}$$

Defining these subsets is useful because:

$$\operatorname{sgn}(x_{n+1} - x_i) = 1 \; \forall \; i \in \mathcal{A}_{n+1} \tag{3.4a}$$
$$\operatorname{sgn}(x_{n+1} - x_i) = -1 \; \forall \; i \in \mathcal{B}_{n+1} \tag{3.4b}$$
$$\operatorname{sgn}(x_{n+1} - x_i) = 0 \; \forall \; i \in \mathcal{C}_{n+1} \tag{3.4c}$$

Utilizing the subsets, eq. (3.2) can be rewritten as:

$$\begin{aligned} F_{n+1} = &- \sum_{i \in \mathcal{A}_{n+1}} \kappa_i (x_{n+1} - \alpha_i (x_i + \xi_{n+1,i})(1 - \beta_i \operatorname{sgn}(x_{n+1} - x_i))) \\ &- \sum_{i \in \mathcal{B}_{n+1}} \kappa_i (x_{n+1} - \alpha_i (x_i + \xi_{n+1,i})(1 - \beta_i \operatorname{sgn}(x_{n+1} - x_i))) \\ &- \sum_{i \in \mathcal{C}_{n+1}} \kappa_i (x_{n+1} - \alpha_i (x_i + \xi_{n+1,i})(1 - \beta_i \operatorname{sgn}(x_{n+1} - x_i))) \end{aligned} \tag{3.5}$$

Inserting eq. (3.4) yields

$$\begin{aligned}F_{n+1} =& -\sum_{i\in\mathcal{A}_{n+1}} \kappa_i(x_{n+1} - \alpha_i(x_i + \xi_{n+1,i})(1 - \beta_i \cdot 1))\\ &- \sum_{i\in\mathcal{B}_{n+1}} \kappa_i(x_{n+1} - \alpha_i(x_i + \xi_{n+1,i})(1 - \beta_i \cdot (-1)))\\ &- \sum_{i\in\mathcal{C}_{n+1}} \kappa_i(x_{n+1} - \alpha_i(x_i + \xi_{n+1,i})(1 - \beta_i \cdot 0))\\ =& -\sum_{i\in\mathcal{A}_{n+1}} \kappa_i(x_{n+1} - \alpha_i(x_i + \xi_{n+1,i})(1 - \beta_i))\\ &- \sum_{i\in\mathcal{B}_{n+1}} \kappa_i(x_{n+1} - \alpha_i(x_i + \xi_{n+1,i})(1 + \beta_i))\\ &- \sum_{i\in\mathcal{C}_{n+1}} \kappa_i(x_{n+1} - \alpha_i(x_i + \xi_{n+1,i}))\end{aligned} \tag{3.6}$$

Factoring out x_{n+1} yields:

$$\begin{aligned}F_{n+1} =& -x_{n+1}\left(\sum_{i\in\mathcal{A}_{n+1}} \kappa_i(1-\beta_i) + \sum_{i\in\mathcal{B}_{n+1}} \kappa_i(1+\beta_i) + \sum_{i\in\mathcal{C}_{n+1}} \kappa_i\right)\\ &+ \sum_{i\in\mathcal{A}_{n+1}} \kappa_i\alpha_i(x_i + \xi_{n+1,i})(1-\beta_i)\\ &+ \sum_{i\in\mathcal{B}_{n+1}} \kappa_i\alpha_i(x_i + \xi_{n+1,i})(1+\beta_i)\\ &+ \sum_{i\in\mathcal{C}_{n+1}} \kappa_i\alpha_i(x_i + \xi_{n+1,i})\end{aligned} \tag{3.7}$$

Setting $F_{n+1} = 0$ and denoting the equilibrium point as x^*_{n+1} gives:

$$\begin{aligned}x^*_{n+1} =& \frac{\sum_{i\in\mathcal{A}_{n+1}} \kappa_i\alpha_i(x_i + \xi_{n+1,i})(1-\beta_i)}{\sum_{i\in\mathcal{A}_{n+1}} \kappa_i(1-\beta_i) + \sum_{i\in\mathcal{B}_{n+1}} \kappa_i(1+\beta_i) + \sum_{i\in\mathcal{C}_{n+1}} \kappa_i}\\ &+ \frac{\sum_{i\in\mathcal{B}_{n+1}} \kappa_i\alpha_i(x_i + \xi_{n+1,i})(1+\beta_i)}{\sum_{i\in\mathcal{A}_{n+1}} \kappa_i(1-\beta_i) + \sum_{i\in\mathcal{B}_{n+1}} \kappa_i(1+\beta_i) + \sum_{i\in\mathcal{C}_{n+1}} \kappa_i}\\ &+ \frac{\sum_{i\in\mathcal{C}_{n+1}} \kappa_i\alpha_i(x_i + \xi_{n+1,i})}{\sum_{i\in\mathcal{A}_{n+1}} \kappa_i(1-\beta_i) + \sum_{i\in\mathcal{B}_{n+1}} \kappa_i(1+\beta_i) + \sum_{i\in\mathcal{C}_{n+1}} \kappa_i}\end{aligned} \tag{3.8}$$

3.1.1 Gains that ensure desired convergence

Let x_m denote the position of the neighbor of ν_{n+1} that is farthest from $x_0 = 0$. This is formally expressed as $x_m = \max_{i \in \mathcal{N}_{n+1}} x_i^*$. The goal of the potential field in eq. (3.1) is to ensure that $x_{n+1}^* > x_m$, which means that $x_{n+1}^* > x_i \ \forall \ i \in \mathcal{N}_{n+1}$. For $x_{n+1} = x_{n+1}^*$, the subsets defined in eq. (3.4) will be $\mathcal{A}_{n+1} = \mathcal{N}_{n+1}$ and $\mathcal{B}_{n+1} = \mathcal{C}_{n+1} = \emptyset$, leading to that the expression for the equilibrium point can be simplified as

$$x_{n+1}^* = \frac{\sum_{i \in \mathcal{N}_{n+1}} \kappa_i \alpha_i (x_i + \xi_{n+1,i})(1 - \beta_i)}{\sum_{i \in \mathcal{N}_{n+1}} \kappa_i (1 - \beta_i)} \tag{3.9}$$

Further, constraints on the different gains can be found by inserting eq. (3.9) into the exploration condition given as $x_{n+1}^* > x_m$.

$$\begin{aligned} x_{n+1}^* &> x_m \\ x_{n+1}^* - x_m &> 0 \\ \frac{\sum_{i \in \mathcal{N}_{n+1}} \kappa_i \alpha_i (x_i + \xi_{n+1,i})(1 - \beta_i)}{\sum_{i \in \mathcal{N}_{n+1}} \kappa_i (1 - \beta_i)} - x_m &> 0 \end{aligned} \tag{3.10}$$

Before deriving conditions for κ_i and α_i so that eq. (3.10) is satisfied, it should be noted that these gains will be dynamic. This is due to the fact that $\mathcal{N}_{n+1}$ in most cases will not be static throughout the entire deployment of ν_{n+1}. β_i was introduced in eq. (3.2) as

$$\beta_i = \alpha_i \frac{\mathrm{d}\xi_{n+1,i}}{\mathrm{d}d_{n+1,i}} \tag{3.11}$$

The fact that $\xi(d_{n+1,i})$ is a monotonically decreasing function, and demanding $\alpha_i \geq 0$, β_i will be bounded from above by 0, and $-\beta_i \geq 0$. To prevent zero-division in eq. (3.10), κ_i must be constrained so that $\exists \ i \in \mathcal{N}(n+1)$ s.t. $\kappa_i > 0$. Multiplying eq. (3.10) by $\sum_{i \in \mathcal{N}_{n+1}} \kappa_i (1 - \beta_i)$ results in

$$\sum_{i \in \mathcal{N}_{n+1}} \kappa_i \alpha_i (x_i + \xi_{n+1,i})(1 - \beta_i) - x_m \sum_{i \in \mathcal{N}_{n+1}} \kappa_i (1 - \beta_i) > 0 \tag{3.12}$$

Defining $\gamma_i := 1 - \beta_i \geq 1$ to simplify notation, eq. (3.12) becomes

$$\begin{aligned} \sum_{i \in \mathcal{N}_{n+1}} \kappa_i \alpha_i (x_i + \xi_{n+1,i}) \gamma_i - x_m \sum_{i \in \mathcal{N}_{n+1}} \kappa_i \gamma_i &> 0 \\ \sum_{i \in \mathcal{N}_{n+1}} \kappa_i \alpha_i x_i \gamma_i + \sum_{i \in \mathcal{N}_{n+1}} \kappa_i \alpha_i \xi_{n+1,i} \gamma_i - x_m \sum_{i \in \mathcal{N}_{n+1}} \kappa_i \gamma_i &> 0 \end{aligned} \tag{3.13}$$

Extracting the mth component from $\sum_{i\in\mathcal{N}_{n+1}} \kappa_i\alpha_i x_i\gamma_i$ in eq. (3.13) gives

$$\kappa_m\alpha_m x_m\gamma_m + \sum_{i\in\mathcal{N}_{n+1}\setminus\{m\}} \kappa_i\alpha_i x_i\gamma_i + \sum_{i\in\mathcal{N}_{n+1}} \kappa_i\alpha_i\xi_{n+1,i}\gamma_i - x_m \sum_{i\in\mathcal{N}_{n+1}} \kappa_i\gamma_i > 0 \tag{3.14}$$

Which is simplified as

$$x_m\left(\kappa_m\alpha_m\gamma_m - \sum_{i\in\mathcal{N}_{n+1}} \kappa_i\gamma_i\right) + \sum_{i\in\mathcal{N}_{n+1}\setminus\{m\}} \kappa_i\alpha_i x_i\gamma_i + \sum_{i\in\mathcal{N}_{n+1}} \kappa_i\alpha_i\xi_{n+1,i}\gamma_i > 0 \tag{3.15}$$

The bounds on the parameters and variables in the above expressions is summarized as:

$$\begin{aligned} &\kappa_i \geq 0,\ \alpha_i \geq 0,\ \gamma_i \geq 1,\ x_0 = 0,\ x_i \geq 0\,\forall\, i \in [1,n] \\ &\xi_{n+1,i} > \xi_\tau\ \forall\ i \in \mathcal{N}_{n+1} \end{aligned} \tag{3.16}$$

Lower bounds on each of the terms in eq. (3.15) can be found by using eq. (3.16).

Conditions on the second and third term in eq. (3.15)

As $\gamma_i \geq 1$, the lower bound of $\sum_{i\in\mathcal{N}_{n+1}\setminus\{m\}} \kappa_i\alpha_i x_i\gamma_i$ will simply be

$$\begin{aligned} \sum_{i\in\mathcal{N}_{n+1}\setminus\{m\}} \kappa_i\alpha_i x_i\gamma_i &\geq \sum_{i\in\mathcal{N}_{n+1}\setminus\{m\}} \kappa_i\alpha_i x_i\cdot 1 \geq 0 \\ \sum_{i\in\mathcal{N}_{n+1}\setminus\{m\}} \kappa_i\alpha_i x_i\gamma_i &\geq 0 \end{aligned} \tag{3.17}$$

Proceeding to $\sum_{i\in\mathcal{N}_{n+1}} \kappa_i\alpha_i\xi_{n+1,i}\gamma_i$, a lower bound can be found as

$$\sum_{i\in\mathcal{N}_{n+1}} \kappa_i\alpha_i\xi_{n+1,i}\gamma_i \geq \sum_{i\in\mathcal{N}_{n+1}} \kappa_i\alpha_i\cdot\xi_\tau\cdot 1 \geq 0 \tag{3.18}$$

If $\exists\ i \in \mathcal{N}(n+1)$ s.t. $\kappa_i, \alpha_i > 0$

$$\sum_{i\in\mathcal{N}_{n+1}} \kappa_i\alpha_i\xi_{n+1,i}\gamma_i > 0 \tag{3.19}$$

Conditions on the first term eq. (3.15)

When evaluating $x_m\left(\kappa_m\alpha_m\gamma_m - \sum_{i\in\mathcal{N}_{n+1}} \kappa_i\gamma_i\right)$ it should be noted that $m \in [1,n]$, and therefore, $x_m \geq 0$, as stated in eq. (3.16). This leads to the fact that

$\kappa_m \alpha_m \gamma_m - \sum_{i \in \mathcal{N}_{n+1}} \kappa_i \gamma_i$ must be nonnegative for the term $x_m \left(\kappa_m \alpha_m \gamma_m - \sum_{i \in \mathcal{N}_{n+1}} \kappa_i \gamma_i \right)$ to be nonnegative.

$$\begin{aligned} \kappa_m \alpha_m \gamma_m - \sum_{i \in \mathcal{N}_{n+1}} \kappa_i \gamma_i &\geq 0 \\ \kappa_m \alpha_m \gamma_m - \kappa_m \gamma_m - \sum_{i \in \mathcal{N}_{n+1} \setminus \{m\}} \kappa_i \gamma_i &\geq 0 \\ \kappa_m \gamma_m (\alpha_m - 1) &\geq \sum_{i \in \mathcal{N}_{n+1} \setminus \{m\}} \kappa_i \gamma_i \end{aligned} \tag{3.20}$$

The way $\xi_{n+1,i}$ is modeled in eq. (2.19), it can be shown that

$$-\frac{\bar{\xi}}{2} \omega \leq \frac{\mathrm{d}}{\mathrm{d} d_{n+1,i}} \xi(d_{n+1,i}) \leq 0 \tag{3.21}$$

Defining $\delta = \frac{\bar{\xi}}{2} \omega$, the inequality can be written as $-\delta \leq \frac{\mathrm{d}\xi_{n+1,i}}{\mathrm{d}d_{n+1,i}} \leq 0$. Together with the fact that $\gamma_i = 1 - \beta_i = 1 - \alpha_i \frac{\mathrm{d}\xi_{n+1,i}}{\mathrm{d}d_{n+1,i}}$, this inequality can be utilized as

$$\begin{aligned} -\delta \leq \frac{\mathrm{d}\xi_{n+1,i}}{\mathrm{d}d_{n+1,i}} &\leq 0 && \text{Multiplying by } (-\alpha_i) \\ 0 \leq -\alpha_i \frac{\mathrm{d}\xi_{n+1,i}}{\mathrm{d}d_{n+1,i}} &\leq \alpha_i \delta && \text{Adding } 1 \\ 1 \leq 1 - \alpha_i \frac{\mathrm{d}\xi_{n+1,i}}{\mathrm{d}d_{n+1,i}} &\leq 1 + \alpha_i \delta \\ 1 \leq \gamma_i &\leq 1 + \alpha_i \delta && \text{Multiplying by } (\kappa_i) \\ \kappa_i \leq \kappa_i \gamma_i &\leq \kappa_i (1 + \alpha_i \delta) \end{aligned} \tag{3.22}$$

Choosing $\alpha_m \geq 1$ and restating that $\gamma_m \geq 1$, will lead to

$$\kappa_m \gamma_m (\alpha_m - 1) \geq \kappa_m (\alpha_m - 1) \geq 0 \tag{3.23}$$

It can be seen from eq. (3.22) that $\kappa_i \gamma_i \leq \kappa_i (1 + \alpha_i \delta)$, which results in

$$\sum_{i \in \mathcal{N}_{n+1} \setminus \{m\}} \kappa_i \gamma_i \leq \sum_{i \in \mathcal{N}_{n+1} \setminus \{m\}} \kappa_i (1 + \alpha_i \delta) \tag{3.24}$$

Combining eqs. (3.23) and (3.24), if κ_i and α_i are chosen so that

$$\kappa_m (\alpha_m - 1) \geq \sum_{i \in \mathcal{N}_{n+1} \setminus \{m\}} \kappa_i (1 + \alpha_i \delta) \tag{3.25}$$

Then

$$\kappa_m\gamma_m(\alpha_m - 1) \geq \kappa_m(\alpha_m - 1) \geq \sum_{i\in\mathcal{N}_{n+1}\backslash\{m\}} \kappa_i(1+\alpha_i\delta) \geq \sum_{i\in\mathcal{N}_{n+1}\backslash\{m\}} \kappa_i\gamma_i \tag{3.26}$$

Which will lead to

$$\begin{aligned} x_m\left(\kappa_m\gamma_m(\alpha_m - 1) - \sum_{i\in\mathcal{N}_{n+1}\backslash\{m\}} \kappa_i\gamma_i\right) \geq 0 \\ x_m\left(\kappa_m\alpha_m\gamma_m - \sum_{i\in\mathcal{N}_{n+1}} \kappa_i\gamma_i\right) \geq 0 \end{aligned} \tag{3.27}$$

Restating eq. (3.15) for reference

$$x_m\left(\kappa_m\alpha_m\gamma_m - \sum_{i\in\mathcal{N}_{n+1}} \kappa_i\gamma_i\right) + \sum_{i\in\mathcal{N}_{n+1}\backslash\{m\}} \kappa_i\alpha_i x_i\gamma_i + \sum_{i\in\mathcal{N}_{n+1}} \kappa_i\alpha_i\xi_{n+1,i}\gamma_i > 0 \tag{3.28}$$

Looking at eq. (3.16), it can be seen that $\kappa_i, \alpha_i, x_i \geq 0$ and $\gamma_i \geq 1$, giving

$$\sum_{i\in\mathcal{N}_{n+1}\backslash\{m\}} \kappa_i\alpha_i x_i\gamma_i \geq 0 \tag{3.29}$$

As $\xi_{n+1,i} > \xi_\tau \ \forall \ i \in \mathcal{N}_{n+1}$ and demanding that $\exists \ i \in \mathcal{N}(n+1)$ s.t. $\kappa_i, \alpha_i > 0$

$$\sum_{i\in\mathcal{N}_{n+1}} \kappa_i\alpha_i\xi_{n+1,i}\gamma_i > 0 \tag{3.30}$$

If κ_i and α_i is chosen so that

$$\kappa_m(\alpha_m - 1) \geq \sum_{i\in\mathcal{N}_{n+1}\backslash\{m\}} \kappa_i(1+\alpha_i\delta) \tag{3.31}$$

the below inequality will hold

$$\kappa_m\gamma_m(\alpha_m - 1) \geq \kappa_m(\alpha_m - 1) \geq \sum_{i\in\mathcal{N}_{n+1}\backslash\{m\}} \kappa_i(1+\alpha_i\delta) \geq \sum_{i\in\mathcal{N}_{n+1}\backslash\{m\}} \kappa_i\gamma_i \tag{3.32}$$

This will guarantee that

$$\begin{gathered}
x_m \left(\kappa_m \gamma_m (\alpha_m - 1) - \sum_{i \in \mathcal{N}_{n+1} \setminus \{m\}} \kappa_i \gamma_i \right) \geq 0 \\
x_m \left(\kappa_m \alpha_m \gamma_m - \sum_{i \in \mathcal{N}_{n+1}} \kappa_i \gamma_i \right) \geq 0
\end{gathered} \tag{3.33}$$

Summarizing the findings

Equation (3.33) concludes the analysis of conditions that need to hold for ensuring desired covergence of ν_{n+1}. The parameters used for simplifying notation are summarized in eq. (3.34), and the conditions that need to be satisfied is shown in eq. (3.35).

$$\begin{aligned}
\beta_i &= \alpha_i \frac{\mathrm{d}\xi_{n+1,i}}{\mathrm{d}d_{n+1,i}} \leq 0 \\
\gamma_i &= 1 - \beta_i = 1 - \alpha_i \frac{\mathrm{d}\xi_{n+1,i}}{\mathrm{d}d_{n+1,i}} \geq 1 \\
-\delta &= -\frac{\overline{\xi}}{2}\omega \leq \frac{\mathrm{d}\xi_{n+1,i}}{\mathrm{d}d_{n+1,i}} \leq 0
\end{aligned} \tag{3.34}$$

$$\begin{aligned}
& x_0 = 0 \\
& x_i \geq 0 \; \forall \; i \\
& \kappa_i \geq 0 \\
& \alpha_i \geq 0 \\
& \exists \; i \in \mathcal{N}_{n+1} \text{ s.t. } \kappa_i, \alpha_i > 0 \\
& m = \underset{i \in \mathcal{N}_{n+1}}{\arg\max}\, x_i \\
& \alpha_m \geq 1 \\
& \gamma_i = 1 - \beta_i = 1 - \alpha_i \frac{\mathrm{d}}{\mathrm{d}d_{n+1,i}} \xi(d_{n+1,i}) \geq 1 \\
& \xi_{n+1,i} > \xi_\tau \; \forall \; i \in \mathcal{N}_{n+1} \\
& \kappa_m(\alpha_m - 1) \geq \sum_{i \in \mathcal{N}_{n+1} \setminus \{m\}} \kappa_i (1 + \alpha_i \delta)
\end{aligned} \tag{3.35}$$

3.2 Simulation

For implementing the deployment scheme, a simulator has been developed using Python. The simulator is available at [46].

3.2.1 Simulation setup

The SINTEF Control Station (SCS) is denoted ν_0, and placed at $x_0 = 0$. For simplification purposes, this point will be the entrance point of all agents. The agents will be incrementally deployed, meaning that there is at most one moving agent at any given time.

As an agent is deploying, it will be subject to the field and force described in the previous section. The fact that the gains used in the potential field will be dynamic, leads to that the force experienced by the deploying agent will vary quite a lot depending on its neighbors. The deployment of ν_{n+1} will continue until $|F_{n+1}| < F_\tau$ and/or $\xi_{n+1,i} < \xi_{stop} \; \forall \; i \in \mathcal{N}_{n+1}$. The constraint on $\xi_{n+1,i}$ ensures that the networks stays connected throughout the entire deployment.

3.2.2 Parameters and results

Choosing $\kappa_i = 0 \; \forall \; i \in \mathcal{N}_{n+1} \setminus \{m\}$, $\kappa_m = 1$, $\alpha_i = 0 \; \forall \; i \in \mathcal{N}_{n+1} \setminus \{m\}$ and $\alpha_m = 1$ satisfy the conditions in eq. (3.35). By selecting these gains, the original expression of F_{n+1} in eq. (3.2) is simplified to

$$F_{n+1} = -(x_{n+1} - (x_m + \xi_{n+1,m}))(1 - \beta_m \operatorname{sgn}(x_{n+1} - x_m)) \tag{3.36}$$

Parameter	**Fig. 3.1**
r_{max}	3.0
d_{perf}	0.1
d_{none}	2.5
$\bar{\xi}$	1.0
ξ_τ	0.2
ξ_{stop}	0.22
F_{sat}	3.0
F_τ	0.1
$\kappa_i \; \forall \; i \in \mathcal{N}_{n+1} \setminus \{m\}$	0
κ_m	1.0
$\alpha_i \; \forall \; i \in \mathcal{N}_{n+1}$	0
$\delta = \frac{\bar{\xi}}{2}\omega$	$\frac{5}{24}\pi$

Table 3.1

Using the above-mentioned gains and the parameters shown in table 3.1, the final configuration ends up being as shown in fig. 3.1. It can be seen that the agents are nicely spread and that the simple exploration condition given as $x_{n+1} > x_n$ is satisfied. Figure 3.2(a) shows the force that is applied to each of the agents. It should be noted that the force which ν_6 experiences is plotted last, and it is therefore partially plotted on top of the other forces. Let k be an arbitrary index such that $k < n+1$, where $n+1$ is the index of the currently deploying agent. In the given setup it will always be the case that $F_{n+1} = F_k$ up until ν_{k+1} becomes a neighbor of ν_{n+1}, i.e. up until $k+1 \in \mathcal{N}_{n+1}$.

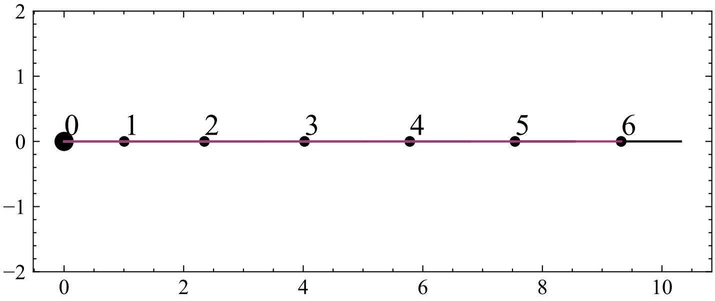
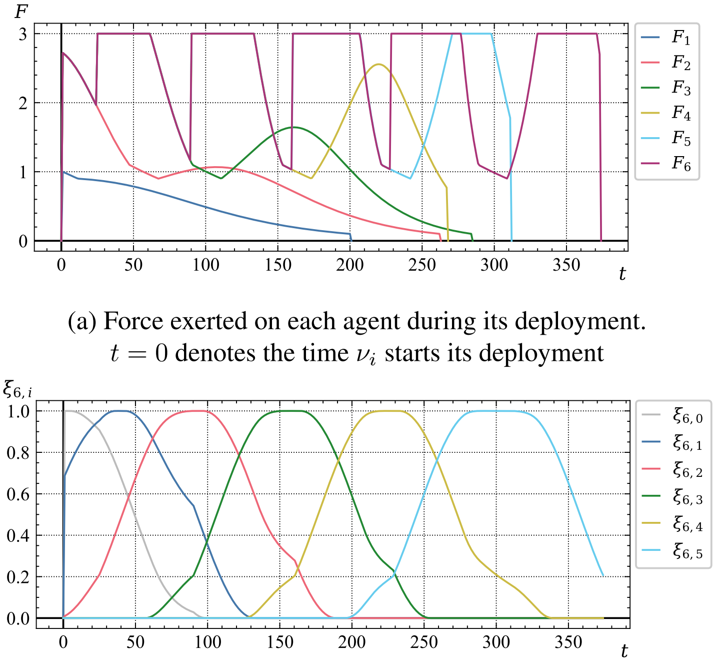


Figure 3.1: Final configuration when deploying six agents using the potential field proposed for 1D

The vertical increases in force occurs when the deploying agent detects a neighbor that is positioned farther from ν_0 than any of its current neighbors. As the signal strength threshold for what is considered to be neighbors, ξ_τ, is set to be 0.2, it is nicely illustrated in figs. 3.2(a) and 3.2(b) that the rapid increases in F_6 occurs the first time $\xi_{6,i} = 0.2 \;\forall\; i \in [1,5]$. Such aggressive increases in applied force is of course only theoretically possible, and if this approach is to be extended, a maximum change in applied force has to be added.



(a) Force exerted on each agent during its deployment.
$t = 0$ denotes the time ν_i starts its deployment

(b) The development of the inverse RSS distance between ν_6 and the other agents during the deployment of ν_6.
$t = 0$ denotes the time instant ν_6 starts its deployment

Figure 3.2

Chapter 4

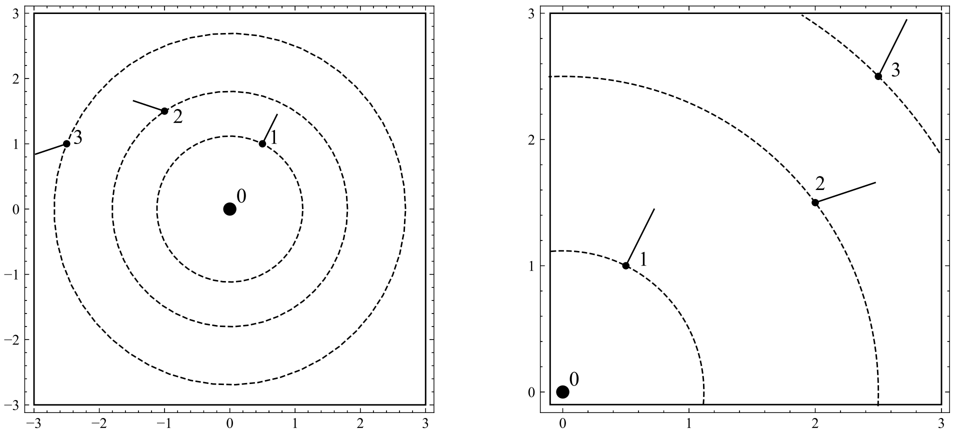
2D

4.1 Directly expanding exploration condition

As described in chapter 3, the main goal for the potential field U_{n+1} is to guide ν_{n+1} to x^*_{n+1} so that the network has explored more area when ν_{n+1} stops than what it did prior to the deployment of ν_{n+1}. The exploration condition defined in chapter 3 was given as $\|x_{n+1} - x_0\| > \|x_n - x_0\|$. Directly expanding this in 2D will be $\|\mathbf{x}_{n+1} - \mathbf{x}_0\| > \|\mathbf{x}_n - \mathbf{x}_0\|$, which will severely constrain the feasible landing positions of the deploying agent. Figure 4.1(a) illustrates that when ν_{n+1} finishes its deployment, it has to be positioned outside the circle that is centered at $\mathbf{x}_0$ and has a radius of $\|\mathbf{x}_n - \mathbf{x}_0\|$. In situations where ν_0 is placed in the lower left corner of the environment its get even clearer why it is not applicable to directly expand the exploration condition. This is shown in fig. 4.1(b), and it is clear that the area where the next deploying agent is allowed to position itself eventually will become infinitely small.

4.2 Investigating other exploration conditions

The aggregation index r derived by Clark and Evans in [47] is a metric that often is used for evaluating the spatial relationship in a population [48, 49]. The index is defined as the ratio between the average distance from each node to its nearest neighbor, $\overline{r}_O$, and the expected mean distance between each node and its nearest neighbor in a Poisson distribution, denoted $\overline{r}_E$.



(a)

(b)

Figure 4.1

$$\begin{aligned}\overline{r}_O &= \frac{\sum_{i=1}^{N} r_i}{N}\\ \overline{r}_E &= \frac{1}{2\sqrt{N/A}}\\ r &= \frac{\overline{r}_O}{\overline{r}_E}\end{aligned} \tag{4.1}$$

r_i is the distance between agent i and its nearest neighbor, and N is the number of agents in the area A.

As $r > 1$ indicates dispersion and $r < 1$ indicates clustering, investigating how the aggregation index could be used for defining an exploration condition seems tempting. However, as the area of the mission space is unknown to the agents, it would be hard to find a valid way to calculate $\overline{r}_E$. Computing the minimum bounding rectangle [50] could be a possible workaround, but in the given setup, and considering the limited knowledge of each agent, this would not be applicable.

Tweaking the repelling potential field was also looked into. The main idea was that ν_{n+1} should be repelled from each of the already deployed agents in such a way that it sequentially visited $\mathbf{x}_1^{\mathcal{I}}, \mathbf{x}_2^{\mathcal{I}}, \ldots, \mathbf{x}_n^{\mathcal{I}}$. After ν_{n+1} had reached $\mathbf{x}_n^{\mathcal{I}}$, it would then proceed to explore the unknown environment. This proved to be a difficult task using solely a repelling field, and mathematically expressing the conditions that had to be fulfilled ended up being hard. Therefore, it seemed reasonable to use a more classical potential field approach for solving the 2D exploration problem.

4.3 Exploration in 2D using classical potential fields

The majority of previous research that has been done on deploying agents into an area using potential fields defines some sort of target position of the agents. As the problem discussed in this thesis assumes that there is no knowledge available about the environment prior to the deployment, it is not feasible to define a unique target position for each agent. The only knowledge each agent has about its surroundings is the position of its neighbors and the measurements from its sensors. Using this information, a scheme that is somewhat inspired by how the each agent calculates a *moving* target point in [36] (see Incremental deployment) is proposed. During the deployment, the deploying agent is placed in the classical potential field described in section 2.1.1. This field, and the resulting forces are restated here for reference.

$$U_{tot} = U_{rep} + U_{att} \tag{4.2}$$

$$U_{rep} = \sum_j U_{obs,j} \tag{4.3}$$

$$U_{obs,j} = \begin{cases} \frac{1}{2} k_{obs} \left(\frac{1}{\|\mathbf{r}_{i,j}\|} - \frac{1}{d_0} \right)^2 & \text{if } \|\mathbf{r}_{i,j}\| \leq d_0 \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

$$U_{att} = \frac{1}{2} k_{att} \|\mathbf{x}_i - \mathbf{x}_g\|^2 \tag{4.5}$$

The total virtual force experienced by agents placed in the potential field will be $\mathbf{F}_{tot} = -\nabla U_{tot}$. The fact that ∇ is a linear operator gives

$$\begin{aligned} \mathbf{F}_{tot} &= -\nabla U_{rep} - \nabla U_{att} \\ &= \mathbf{F}_{rep} + \mathbf{F}_{att} \end{aligned} \tag{4.6}$$

The repelling force will be

$$\mathbf{F}_{rep} = \sum_j \mathbf{F}_{obs,j} \tag{4.7}$$

where

$$\mathbf{F}_{obs,j} = \begin{cases} -k_{obs} \frac{1}{\|\mathbf{r}_{i,j}\|^2} \left(\frac{1}{\|\mathbf{r}_{i,j}\|} - \frac{1}{d_0} \right) \frac{\mathbf{r}_{i,j}}{\|\mathbf{r}_{i,j}\|} & \text{if } \|\mathbf{r}_{i,j}\| \leq d_0 \\ 0 & \text{otherwise} \end{cases} \tag{4.8}$$

k_{obs} is a positive constant gain and $\mathbf{r}_{i,j}$ is the Euclidean distance between ν_i and obstacle j. The attractive force is given as

$$\mathbf{F}_{att} = -k_{att}(\mathbf{x}_i - \mathbf{x}_g) \tag{4.9}$$

where k_{att} is a positive constant and $\mathbf{x}_g$ is the target position of ν_i.

As mentioned in the problem description, the overall goal of this thesis is to propose an exploration strategy for deploying agents incrementally into a disaster area. The environment where the mesh network is to be constructed is unknown, but the initial search direction is assumed to be easily available (i.e. the SCS, denoted ν_0, is placed in the opening of the unknown area or close to a door leading into an unknown room).

4.3.1 Workflow of the incremental deployment scheme

Throughout the deployment, the agents go through four states; *spawned*,*following*, *exploring* and *stopped*. The SCS is placed at $\mathbf{x}_0^{\mathcal{I}}$ which is defined as the origin of the inertial frame. For simplification purposes, it is also assumed that $\mathbf{x}_0^{\mathcal{I}}$ is the entrance point of the agents. Since $\mathcal{I}$ and $\mathcal{A}_i$ are EN frames, the choice of entrance point results in that the two frames will be aligned before ν_i starts its deployment. As the agents are placed at $\mathbf{x}_0^{\mathcal{I}}$, all agents except ν_1 are assigned the *spawned* state. ν_1 is the first agent to deploy, and is therefore directly assigned the *exploration* state.

For the novel target *direction* approach presented here to be compatible with the virtual potential field in eq. (4.2), ν_1 has to be assigned a target *point*. Utilizing the fact that the agent frame of ν_1 is aligned with the inertial frame, the initial search direction can be expressed directly in the $\mathcal{A}_1$ frame. The SCS, denoted ν_0, generates a target point for ν_1, given as

$$\mathbf{x}_{g,1}^{\mathcal{A}_1} = \begin{bmatrix} d_g \cos\left(\theta_{init}^{\mathcal{A}_1}\right) \\ d_g \sin\left(\theta_{init}^{\mathcal{A}_1}\right) \end{bmatrix} \tag{4.10}$$

where d_g is a pre-defined distance from ν_0 the target point is to be generated at, and $\theta_{init}^{\mathcal{A}_1}$ is the initial search direction.

Being subjected to the force in eq. (4.6), ν_1 will move towards $\mathbf{x}_{g,1}^{\mathcal{A}_1}$. For the target point defined in eq. (4.10) to be used so that ν_1 moves in the initial search *direction*, the target point has to be updated at each time step. Let $\Delta\mathbf{x}$ be the displacement of ν_1 between $t = t_1$ and $t = t_2$ and the target point at $t = t_1$ be denoted $\mathbf{x}_{g,1,t_1}^{\mathcal{A}_1}$. The target point at $t = t_2$ then relates to the target point at $t = t_1$ as

$$\mathbf{x}^{\mathcal{A}_1}_{g,1,t_2} = \mathbf{x}^{\mathcal{A}_1}_{g,1,t_1} + \begin{bmatrix} \|\Delta\mathbf{x}\| \cos\left(\theta^{\mathcal{A}_1}_{init}\right) \\ \|\Delta\mathbf{x}\| \sin\left(\theta^{\mathcal{A}_1}_{init}\right) \end{bmatrix} \tag{4.11}$$

Updating the target point of ν_1 as shown in eq. (4.11) enables the target direction approach to be implemented using the classical potential field method.

ν_1 will be in the *exploring* state as long as the magnitude of the force exerted on it is above a threshold F_τ, and the inverse Received Signal Strength (RSS) distance between ν_1 and the SCS is larger than ξ_{stop}. For ν_1 to be connected to the SCS, ξ_{stop} must be designed such that $\xi_{stop} + \epsilon > \xi_\tau$, where $0 < \epsilon \ll \overline{\xi}$.

When the force that ν_1 is subject to or the inverse RSS distance between ν_1 and the SCS drops below their given threshold, ν_1 will stop and be assigned the *stopped* state. The agent will then send its position to the SCS and proceed to calculate the target direction of ν_2, and ν_2 is assigned the *following* state.

Since the SCS is a neighbor of ν_1, the SCS will know the position of ν_1 relative to itself. To keep the below derivations generic, assume that the index of the agent that finished its deployment is n. The final position of ν_n is the transmitted to the SCS through the network, so that the SCS can build a *path tree* consisting of the positions of all the previously deployed agents. When the SCS receives information about the position of ν_n, the path tree will be $\{\mathbf{x}^{\mathcal{I}}_1, \mathbf{x}^{\mathcal{I}}_2, \ldots, \mathbf{x}^{\mathcal{I}}_n\}$. The inertial frame and all the agent frames are EN frames, and it is assumed that all agents know the relative position of its neighbors. This results in that the position of ν_n can be sent to the SCS in a relative simple manner, assuming effective routing. [1]

In the *following* state, the target position of ν_{n+1} will sequentially be assigned the positions in the path tree. The first target position will be $\mathbf{x}^{\mathcal{I}}_1$, and as ν_{n+1} moves towards $\mathbf{x}^{\mathcal{I}}_1$, the inverse RSS distance will decrease. When $\xi_{n+1,1}$ becomes smaller than a threshold ξ_{sw}, the target position of ν_{n+1} switches from $\mathbf{x}^{\mathcal{I}}_1$ to $\mathbf{x}^{\mathcal{I}}_2$. This procedure will continue until ν_{n+1} has reached $\mathbf{x}^{\mathcal{I}}_n$. As the environment is assumed to be static, the path between $\mathbf{x}^{\mathcal{I}}_{i-1}$ and $\mathbf{x}^{\mathcal{I}}_i$ $\forall\ i \in [1, n]$ is obstacle free since it already has been traveled by ν_i. Therefore, the target point assignment procedure ensures that ν_{n+1} follows an obstacle free path from the SCS to $\mathbf{x}^{\mathcal{I}}_n$.

After ν_{n+1} has reached the position of ν_n, it enters the *exploring* state. When ν_n finished its deployment, it computed a *target direction* based on the measurements from its sensors and the positions of its neighbors. This procedure is described below.

[1] How the routing is to be implemented is beyond the scope of this thesis. In the simulations, ν_n does not transmit its position through the network, $\mathbf{x}^{\mathcal{I}}_n$ is appended directly to the path tree.

Computing the target direction

Sensor j returns the distance to the closest sensed obstacle, $r_{s,j}$, and the corresponding angle, $\psi_{s,j}$, in sensor frame. If the sensor does not sense an obstacle within its sensing range, its measurement is labeled *invalid*. Assuming that the measurement returned from sensor j is valid, a vector corresponding to the measurement can be defined as

$$\mathbf{o}_j^{\mathcal{S}_j} = \begin{bmatrix} r_{s,j}\cos(\psi_{s,j}) \\ r_{s,j}\sin(\psi_{s,j}) \end{bmatrix} \tag{4.12}$$

Using eqs. (2.22) and (2.23), $\mathbf{o}_j^{\mathcal{S}_j}$ is then described in the agent frame of ν_n by

$$\mathbf{o}_j^{\mathcal{A}_n} = \mathbf{R}_{\mathcal{B}_n}^{\mathcal{A}_n}(\psi_{h,n})\mathbf{R}_{\mathcal{S}_j}^{\mathcal{B}_n}(\psi_{rel,j})\mathbf{o}_j^{\mathcal{S}_j} \tag{4.13}$$

It is desirable that the exploration direction of ν_{n+1} should be away from the obstacles that ν_n senses, and that obstacles close to ν_n should have greater impact on the calculation of the exploration direction. $\mathbf{o}_j^{\mathcal{A}_n}$ is therefore scaled and rotated such that

$$\begin{aligned} \hat{\mathbf{o}}_j^{\mathcal{A}_n} &= \mathbf{R}(\pi)\left(r_{max} - \|\mathbf{o}_j^{\mathcal{A}_n}\|\right)\frac{\mathbf{o}_j^{\mathcal{A}_n}}{\|\mathbf{o}_j^{\mathcal{A}_n}\|} \\ &= -\left(r_{max} - \|\mathbf{o}_j^{\mathcal{A}_n}\|\right)\frac{\mathbf{o}_j^{\mathcal{A}_n}}{\|\mathbf{o}_j^{\mathcal{A}_n}\|} \end{aligned} \tag{4.14}$$

where $\mathbf{R}(\,\cdot\,) \in SO(2)$ and r_{max} is the upper limit in the sensing range of the range sensors.

The *total* obstacle vector defined by the valid measurements from the sensors mounted on ν_n is then expressed as

$$\overline{\mathbf{o}}_n^{\mathcal{A}_n} = \sum_j \hat{\mathbf{o}}_j^{\mathcal{A}_n} \tag{4.15}$$

The agent frame of agent ν_n has its origin at $\mathbf{x}_n^{\mathcal{I}}$. Each agent knows the position of its neighbors relative to itself, meaning that ν_n knows $\mathbf{x}_i^{\mathcal{A}_n}\ \forall\ i \in \mathcal{N}_n$. A vector $\mathbf{n}_i^{\mathcal{A}_n}$ is then defined as

$$\mathbf{n}_i^{\mathcal{A}_n} = \mathbf{x}_i^{\mathcal{A}_n} \tag{4.16}$$

Let $m = \arg\min_{i \in \mathcal{N}_n} \|\mathbf{x}_i^{\mathcal{A}_n}\|$ be the index of the neighbor that is positioned closest to ν_n. The vector corresponding to this neighbor is denoted $\mathbf{n}_m^{\mathcal{A}_n}$. To prevent clustering, $\mathbf{n}_m^{\mathcal{A}_n}$ is scaled and rotated so that

$$\begin{aligned} \hat{\mathbf{n}}_m^{\mathcal{A}_n} &= \mathbf{R}(\pi)\, r_{max} \left(\frac{d_\mathcal{T} - \|\mathbf{n}_m^{\mathcal{A}_n}\|}{d_\mathcal{T}} \right) \frac{\mathbf{n}_m^{\mathcal{A}_n}}{\|\mathbf{n}_m^{\mathcal{A}_n}\|} \\ &= -r_{max} \left(\frac{d_\mathcal{T} - \|\mathbf{n}_m^{\mathcal{A}_n}\|}{d_\mathcal{T}} \right) \frac{\mathbf{n}_m^{\mathcal{A}_n}}{\|\mathbf{n}_m^{\mathcal{A}_n}\|} \end{aligned} \tag{4.17}$$

where $d_\mathcal{T}$ is the distance satisfying $\xi(d_\mathcal{T}) = \xi_\mathcal{T}$, i.e. the maximum Euclidean distance between neighbors. The scaling differs from what is done in eq. (4.14) so that $\|\hat{\mathbf{n}}_m^{\mathcal{A}_n}\|$ and $\|\hat{\mathbf{o}}_j^{\mathcal{A}_n}\|$ are within the same range.

The total obstacle vector and the vector from the closest neighbor are summed to form an *exploration* vector as shown in eq. (4.18). This vector is formed as an attempt to use the local knowledge of ν_n to calculate a direction that points away from sensed obstacles and the neighbors of ν_n.

$$\mathbf{e}_n^{\mathcal{A}_n} = \overline{\mathbf{o}}_n^{\mathcal{A}_n} + \hat{\mathbf{n}}_m^{\mathcal{A}_n} \tag{4.18}$$

Writing $\mathbf{e}_n^{\mathcal{A}_n}$ in polar coordinates gives

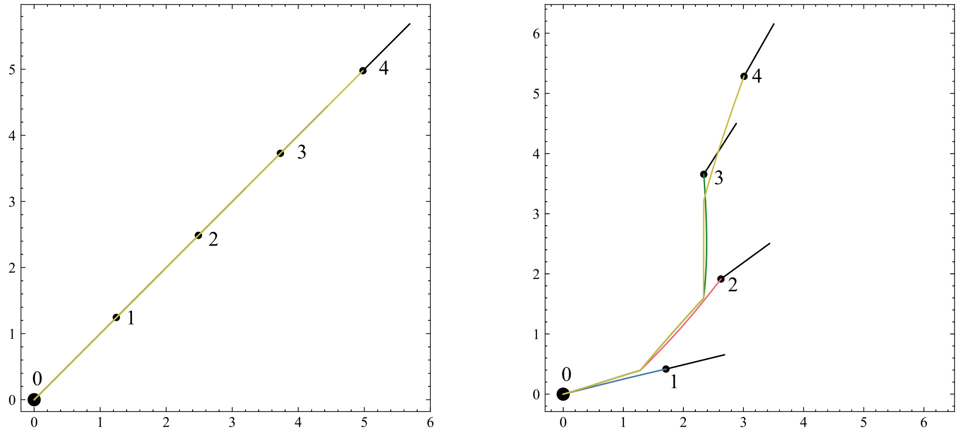
$$\mathbf{e}_n^{\mathcal{A}_n} = \left(\|\overline{\mathbf{e}}_n^{\mathcal{A}_n}\|, \theta_e^{\mathcal{A}_n} \right) \tag{4.19}$$

Using $\theta_e^{\mathcal{A}_n}$ as exploration *direction* works well when at least one sensor mounted on ν_n returns a *valid* measurement. In situations where all sensors return *invalid* measurements and $|\mathcal{N}_n| = 1$ however, using $\theta_e^{\mathcal{A}_n}$ will lead to deployments similar to what is shown in fig. 4.2(a).

As ν_{n+1} does not know its position in the inertial frame, trilateration is assumed used by ν_{n+1} for localizing itself while deploying. In 2D, ν_{n+1} needs minimum three neighbors to determine its own position. Trilateration requires that the points used are non-collinear. Therefore, $\mathbf{e}_n^{\mathcal{A}_n}$ is rotated by a random angle, $\theta_{rand}^{\mathcal{A}_n}$, that is defined to be in the interval $\left[-\theta_\Delta^{\mathcal{A}_n}, \theta_\Delta^{\mathcal{A}_n}\right)$. [2] This leads to

$$\hat{\mathbf{e}}_n^{\mathcal{A}_n} = \mathbf{R}\left(\theta_{rand}^{\mathcal{A}_n}\right) \overline{\mathbf{e}}_n^{\mathcal{A}_n} \tag{4.20}$$

[2] The interval excludes $+\theta_\Delta^{\mathcal{A}_n}$ because $\theta_{rand}^{\mathcal{A}_n}$ is generated by *numpy.random.uniform()*, see the documentation (*link*).



(a) Collinear deployment

(b) Including $\theta_{rand}^{\mathcal{A}_n}$ aids the trilateration of ν_{n+1}

Figure 4.2

Defining $\theta_{tot}^{\mathcal{A}_n} = \theta_{e}^{\mathcal{A}_n} + \theta_{rand}^{\mathcal{A}_n}$ and writing $\hat{\mathbf{e}}_n^{\mathcal{A}_n}$ in polar coordinates yields

$$\hat{\mathbf{e}}_n^{\mathcal{A}_n} = \left(\left\|\hat{\mathbf{e}}_n^{\mathcal{A}_n}\right\|, \theta_{tot}^{\mathcal{A}_n}\right) \tag{4.21}$$

By introducing the random angle, it is less likely that three consecutive drones land at positions that are collinear, which ease the localization of ν_{n+1} during deployment. It should be noted until there has been deployed sufficiently many agents, agent ν_{n+1} will not have a minimum of three neighbors during its deployment. An example where $\theta_{rand}^{\mathcal{A}_n} \in \left[-\frac{\pi}{4}, \frac{\pi}{4}\right)$ is shown in fig. 4.2(b)

Exploring

As briefly discussed above, for the target direction approach to be used with the virtual potential field in eq. (4.2), ν_{n+1} has to be assigned a target *point*. ν_n therefore generates a target point for ν_{n+1}, expressed in the agent frame of ν_n

$$\mathbf{x}_{g,n+1}^{\mathcal{A}_n} = \begin{bmatrix} d_g \cos\left(\theta_{tot}^{\mathcal{A}_n}\right) \\ d_g \sin\left(\theta_{tot}^{\mathcal{A}_n}\right) \end{bmatrix} \tag{4.22}$$

where d_g is the same pre-defined distance as used in eq. (4.10).

As ν_n will be a neighbor of ν_{n+1} when ν_{n+1} starts the exploration phase, $\mathbf{x}_n^{\mathcal{A}_{n+1}}$ will be known to ν_{n+1}. All agent frames are EN frames, leading to that $\mathcal{A}_{n+1}$ and $\mathcal{A}_n$ are related by pure translation. In $\mathcal{A}_{n+1}$, the target point defined in eq. (4.22) will therefore be described as

$$\mathbf{x}_{g,n+1}^{\mathcal{A}_{n+1}} = \mathbf{x}_{g,n+1}^{\mathcal{A}_n} + \mathbf{x}_n^{\mathcal{A}_{n+1}} \tag{4.23}$$

Assuming that ν_n transmits $\theta_{tot}^{\mathcal{A}_n}$ to ν_{n+1}, the deploying agent will have all required information to follow the target *direction*. In the same way as in eq. (4.10), let $\Delta\mathbf{x}$ denote the displacement of ν_{n+1} between $t = t_1$ and $t = t_2$, and the target point of ν_{n+1} at $t = t_1$ be denoted $\mathbf{x}_{g,n+1,t_1}^{\mathcal{A}_{n+1}}$. The target point at $t = t_2$ is then expressed as

$$\mathbf{x}_{g,n+1,t_2}^{\mathcal{A}_{n+1}} = \mathbf{x}_{g,n+1,t_1}^{\mathcal{A}_{n+1}} + \begin{bmatrix} \|\Delta\mathbf{x}\| \cos\left(\theta_{tot}^{\mathcal{A}_n}\right) \\ \|\Delta\mathbf{x}\| \sin\left(\theta_{tot}^{\mathcal{A}_n}\right) \end{bmatrix} \tag{4.24}$$

Moving towards its target point, ν_{n+1} will experience the force defined in eq. (4.6), and the target point will be pushed in the target direction as the deployment of ν_{n+1} proceeds. If there are obstacles present in the environment, the repelling forces they exert on ν_{n+1} will prevent the agent to move in the target direction. In the absence of obstacles, however, ν_{n+1} will converge nicely to the target direction calculated by ν_n.

The deploying agent will go from the *exploring* state to the *stopped* state when the magnitude of the force it experiences becomes smaller than F_τ or the inverse RSS distance between itself and any of its neighbors drops below ξ_{stop}. The way ξ_{stop} is designed, and assuming that the RSS only depend on distance between two agents, the network is ensured to be connected when ν_{n+1} stops. If there are *spawned* agents available, the position of ν_{n+1} will be sent to the SCS through the network, and ν_{n+1} proceeds to calculate the target direction for ν_{n+2} in the same manner as ν_n did for ν_{n+1}.

4.3.2 Evaluating the deployment scheme

It is desirable that the agents should be uniformly spread in the mission area so that the total energy of the network is spent more evenly [51, 52]. The uniformity metric will therefore be used for evaluating the performance of the deployment scheme, and the achieved results will be compared to what Heo and Varshney achieved in [51]. Uniformity of the network can be defined as the average local standard deviation of the distances between agents, which is expressed as

$$
\begin{aligned}
\mathcal{U} &= \frac{1}{N}\sum_{i=1}^{N}\mathcal{U}_i \\
\mathcal{U}_i &= \left(\frac{1}{|\mathcal{N}_i|}\sum_{j=1}^{|\mathcal{N}_i|}(\|\mathbf{x}_i - \mathbf{x}_j\| - m_i)^2\right)^{\frac{1}{2}} \\
m_i &= \frac{1}{|\mathcal{N}_i|}\sum_{j\in\mathcal{N}_i}\|\mathbf{x}_i - \mathbf{x}_j\|
\end{aligned}
\tag{4.25}
$$

where N is the total number of deployed agents and m_i is the mean of the distances between ν_i and its neighbors [51]. A lower value of $\mathcal{U}$ means that the agents are positioned more uniformly in the mission space.

In [51], Heo and Varshney use uniformity to evaluate how their Distributed Self Spreading Algorithm (DSSA) performs compared to a simulated annealing based algorithm. The nodes are assumed to have sensing, communication, computational and locomotion capabilities. Prior to the execution of the DSSA, the nodes are randomly distributed in the area of interest. The DSSA aims to improve the topology for increasing the expected lifetime of the network. The nodes have perfect circular communication ranges, and two nodes are considered neighbors if they are within each other's communication range. Using the nomenclature of this thesis to prevent confusion, an arbitrary sensor node is denoted ν_i in the below description. During the execution of the DSSA, each neighbor of node ν_i exerts a repelling *partial force* on ν_i. The partial force that a neighbor produces depends on the density of all the neighbors and the inverse of the distance between ν_i and the given neighbor. The sum of all partial forces exerted on ν_i will then decide the movement on ν_i. For a more in-depth explanation of the algorithm, the reader is referred to [51].

4.4 Simulation

In the same manner as for the 1D case, a simulation environment has been developed for implementing the novel incremental deployment scheme. This simulation environment is found at [46].

4.4.1 Simulation setup

In all simulations that has been conducted, the SCS is assumed to be placed inside the mission area, and its position is defined as the origin of the inertial frame. The position of the SCS is for simplification purposes, also considered the entrance point of all agents. Starting from the entrance point, the agents will proceed to be incrementally deployed, resulting in that there is at most one moving agent at any

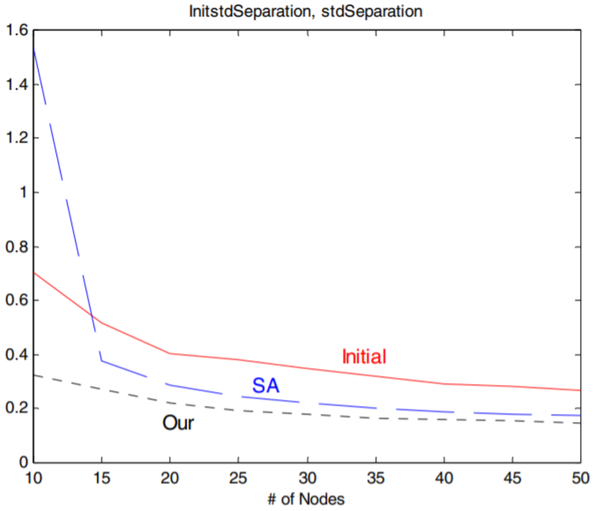


Figure 4.3: The initial uniformity, the resulting uniformity of running a simulated annealing based algorithm (SA) and the uniformity achieved by DSSA (Our) [51]

time during the deployment.

The fact that the work in this thesis mainly is based on that the agents are MINs, it is assumed that a deploying agent may move “above” an agent that already is deployed. This means that $\mathbf{x}_{n+1}^{\mathcal{I}}$ and $\mathbf{x}_{i}^{\mathcal{I}}$ are allowed to be equal as long as ν_{n+1} is the deploying agent and ν_i is already deployed (or is still at the entrance point).

4.4.2 Evaluating achieved uniformity

Heo and Varshney evaluates the performance of the DSSA in a 10×10 obstacle free environment. The communication radius is 4 units and the sensing radius is 2 units. They run the DSSA with a network size varying between 10 and 50, and averages the uniformity achieved by each network size over 100 runs.

Directly comparing the uniformity achieved by the DSSA and the incremental deployment scheme is not appropriate. The DSSA assumes that the mission area is known and uses information about the size of the mission area when computing the partial forces each neighbor exerts on node ν_n. The incremental deployment scheme that is proposed in this thesis however assumes that the area is unknown and will therefore also aim to explore the mission area.

Using $r_{max} = 2$ and designing $\xi_{i,j}$ so that $d_{\mathcal{T}} = 4$, it is inevitable that the agents will cluster near the boundary of the mission area as shown in fig. 4.4. Let ν_{n+1} denote the currently deploying agent. With the above-mentioned r_{max} and $d_{\mathcal{T}}$, the inverse RSS distance between ν_{n+1} and its neighbors will be of such a magnitude when ν_{n+1} reaches the center region that the agent will continue to move in its target direction until the force from the boundary becomes sufficiently large. The parameters used for the deployment shown in fig. 4.4 are found in table 4.1.

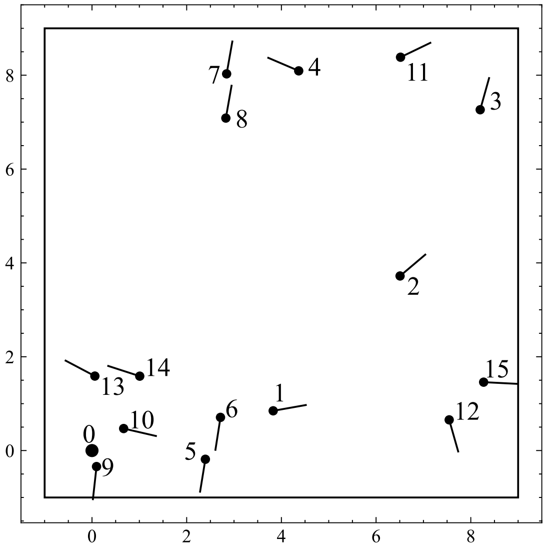


Figure 4.4

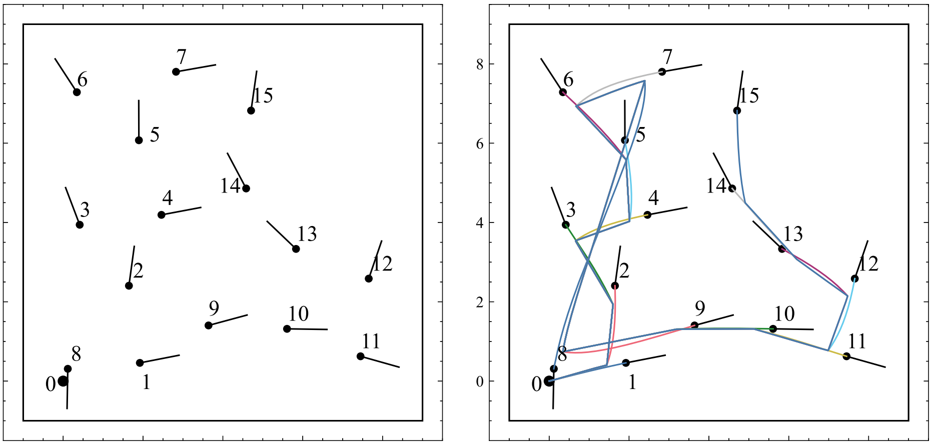
Parameter	Fig. 4.4	Figs. 4.5, 4.8 and 4.10
r_{max}	2.0	2.0
d_{perf}	0.18	0.1
d_{none}	5.6	2.8
$d_{\mathcal{T}}$	4.0	2.0
$\bar{\xi}$	1.0	1.0
$\xi_{\mathcal{T}}$	0.2	0.2
ξ_{sw}	0.95	0.95
ξ_{stop}	0.22	0.22
F_{sat}	2.0	2.0
$F_{\mathcal{T}}$	0.5	0.5
k_{obs}	1.2	1.2
k_{att}	1.0	1.0
d_0	2.0	2.0
$\theta^{\mathcal{A}_n}_{rand}$	$\pi/4$	$\pi/4$
d_g	1.0	1.0

Table 4.1: Parameters used for simulating deployment in 2D

As the deploying agent will move in the target direction as long as $\xi_{n+1,i} \geq \xi_{\mathcal{T}}\ \forall\ i \in \mathcal{N}_{n+1}$ and $\|\mathbf{F}_{tot}\| \geq F_{\mathcal{T}}$, the only form of "coverage" each agent provide is the area where it can transmit sufficiently strong signals. If $\xi_{i,j}$ is modified so that $d_{\mathcal{T}}$ equals the sensing radius used in [51] and keeping $r_{max} = 2$, it is expected that the agents will form a more uniform configuration in the mission area. A deployment where these parameters are used is shown in fig. 4.5. From visual inspection one can conclude that the agents are more spread and do not end up clustering. Figure 4.5(b) also displays the trajectories of the agents, and it can be seen that the agents follow trajectories that already has been taken by the previously deployed ones.

In the same way as what Heo and Varshney did for evaluating the uniformity in [51], 100 simulations were run. It should be emphasized that Heo and Varshney averages the total uniformity achieved by each network size, whereas fig. 4.6 shows the average uniformity after each agent has stopped. Even though it is not applicable to compare the DSSA and the incremental deployment scheme directly, an average total uniformity of 0.136 when deploying fifteen agents confirms that the incremental deployment scheme places the agents uniformly in the mission space.

As the agents are deployed incrementally, the first few agents will not have many neighbors when they finish their deployment, and might even end up having only one neighbor until sufficiently many agents have been deployed. A consequence



(a) Final configuration

(b) Final configuration and the trajectories followed by each agent

Figure 4.5

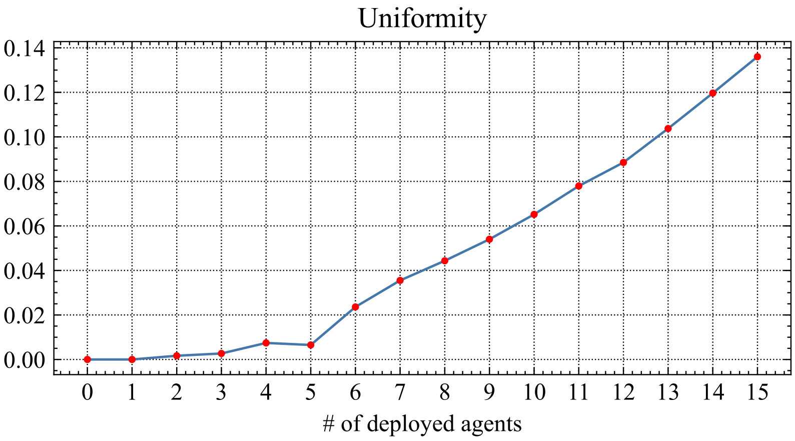
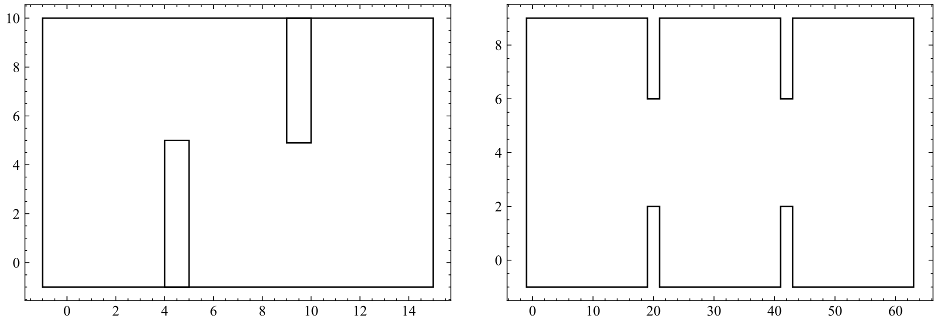


Figure 4.6: Average uniformity over 100 runs

of this is that the local uniformity for each of these agents are 0. In fig. 4.6 it can be seen that the total uniformity of the network is close to 0 after the deployment of each of the first five agents, but increases steadily for the remaining agents.

4.4.3 Deployment in more challenging areas

It can be concluded that the incremental deployment scheme performs well in a mission area without obstacles. To assess what the deployment scheme would accomplish in more complex environments, the mission areas shown in fig. 4.7 was constructed. Both environments are challenging in the sense that they consist of relatively small openings that the agents have to navigate through. This is a demanding task as the target direction of the next deploying agent is calculated using only the local knowledge of the latest deployed agent.



(a) Environment with vertical obstacles

(b) Environment mimicking *Stripa*

Figure 4.7: Challenging environments for testing the incremental deployment scheme

The aim of the environment in fig. 4.7(a) is simply to investigate how the deployment scheme performs when the agents have to be deployed in areas where the boundary is not the only obstacle. Figure 4.7(b) on the other hand, tries to mimic the ground floor hallway connecting Central Building 1 and 2 at NTNU - Gløshaugen. This hallway is known as *Stripa*, which translates to *the Strip*.[3] Stripa is a challenging environment because the three larger regions are separated by narrow passages. As it is hard for the agents to detect these passages, the agents might end up being trapped in each of the regions.

Using the same parameters as what was used in fig. 4.5, thirty agents are deployed into the environment with two vertical obstacles. The final configuration of the agents is shown in fig. 4.8, and it can be seen that the incremental deployment scheme manages to navigate the agents around the two obstacles, and the agents end up begin spread. Figure 4.9 displays the uniformity for the deployment in fig. 4.8,and it can be seen that the uniformity reaches its maximum as ν_6 stops. This happens due to that the agents end up being a bit clustered near the entrance point. It can be seen in fig. 4.8 that the agents deployed after ν_{10} manages to navigate to the open region in the upper left part of the mission area. When agents first are deployed to the open region, the incremental deployment scheme manages to deploy agents in such a way that the obstacles are smoothly avoided. Looking

[3] Blueprint of Stripa is found at [53]

at the uniformity plot, one can see that the uniformity decreases steadily when the first obstacle is passed.

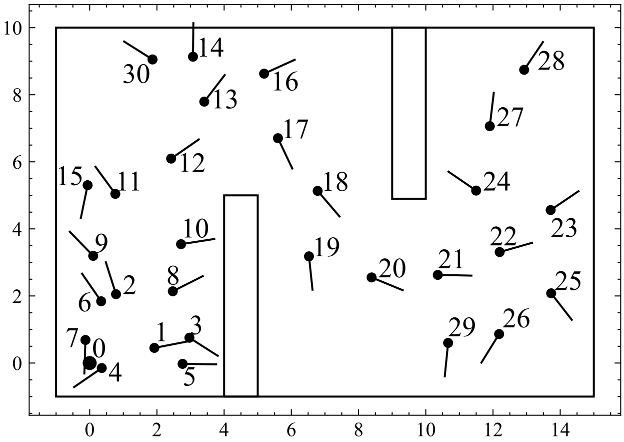


Figure 4.8: Final configuration when deploying thirty drones in the environment with two vertical obstacles

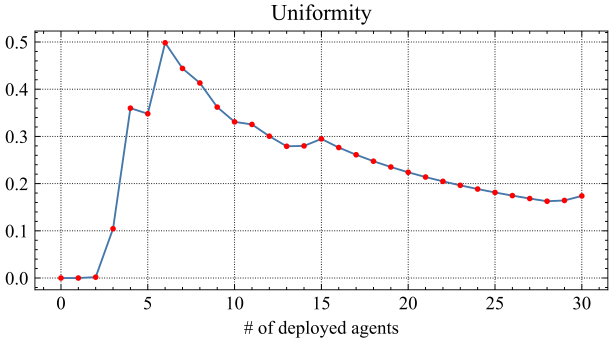


Figure 4.9: Uniformity vs. number of incrementally deployed agents in the mission area shown in fig. 4.7(a)

For simulating a deployment in the *Stripa* environment, the discretization of the sensors was changed so that each sensor only measured the distance that is directly in the direction of which it is mounted on the agent. This had to be done to limit the simulation time. The number of sensing rays was the only parameter that was changed from the simulation in the environment with two vertical obstacles. In fig. 4.10 it can be seen that the incremental deployment scheme is able to overcome the challenge of being stuck in one of the two leftmost regions, and is able to deploy the agents throughout the entire environment.

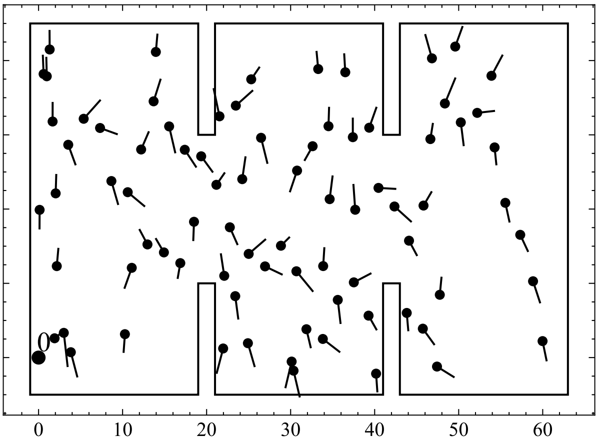


Figure 4.10: Final configuration when deploying eighty agents into the *Stripa* environment. The IDs are omitted for clarity.

The uniformity plot in fig. 4.11 shows that the deployment also manages to produce a satisfactory total uniformity value.

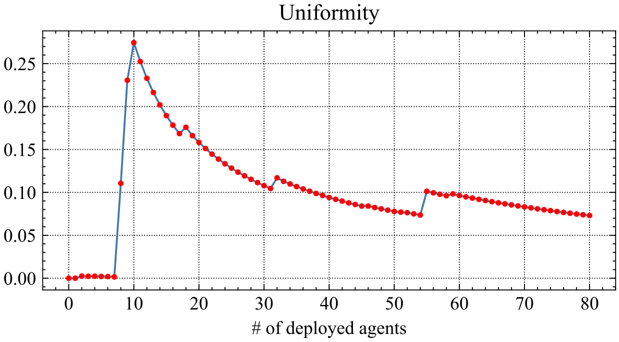


Figure 4.11: Uniformity vs. number of incrementally deployed agents in the *Stripa* environment

4.4.4 The effect the random component has on the deployment

The fact that a random component is used for generating the target direction results in that two deployments in the same mission environment will differ, even if all other parameters are kept constant. The incremental deployment scheme does manage to avoid the obstacles very well in both figs. 4.12 and 4.13, but the agents end up being somewhat clustered. Comparing fig. 4.12 with fig. 4.8 and fig. 4.13 with fig. 4.10, it is evident that the final configuration clearly depends on the random component.

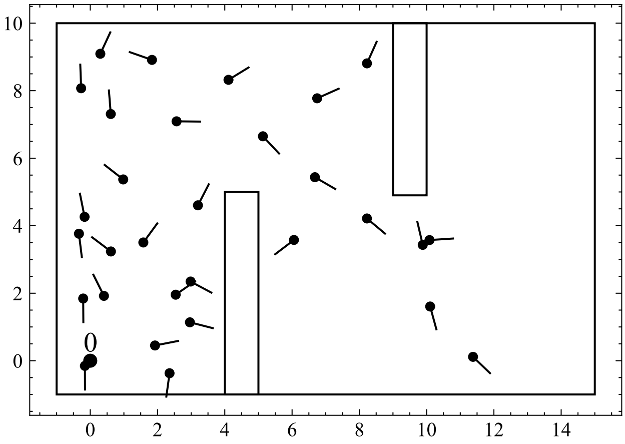


Figure 4.12

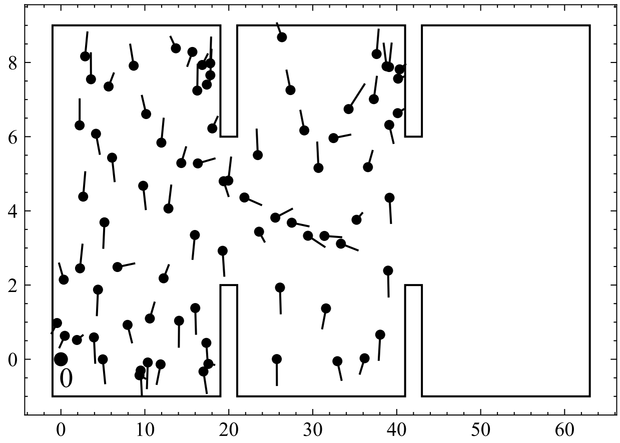


Figure 4.13

Considering the novelty of the incremental deployment scheme, the final configurations shown in figs. 4.12 and 4.13 have to be considered satisfactory. However, when improving the deployment scheme in the future, the calculation of the target direction and how the random component is to be utilized, will be considered a good starting point.

Chapter 5

Summary and Conclusion

5.1 Summary

This report started out giving an introduction of how UAVs previously have been used in SAR missions, and explained potential benefits of using MINs instead of traditional UAVs. After the introduction of the problem, the theory behind the virtual potential field method was presented. This specific method was chosen to be investigated as it does not depend on extensive knowledge of the mission area, and the calculations needed are limited. This was considered ideal, as the Crazyflie 2.1 drones, which the work initially was aimed to be used for physical implementation, have limited sensory and computational capabilities.

Previous research on the use of multi-robot systems, potential fields and incremental deployment was then described in detail. It became apparent that multi-robot systems often are used for different types of area-coverage. Even if this is a different problem than what this thesis tries to solve, it was explained how research on area-coverage provides useful insights in how the robots could spread uniformly in the environment. This has to to with the fact that if the robots have to remain connected during the execution, they robots typically cover the largest area possible if they are uniformly spread.

For potential fields, it was presented how most of the previous work either assumed that some final configuration was known to the robots prior to the deployment or proposed improvements on the original repelling potential field defined in [2]. The fact that it would be infeasible to directly implementing previously used methods for potential fields was then put forward, as it is desirable that the problem in this thesis is solved by deploying the agents incrementally.

When describing the previous work on incremental deployment, it became apparent that only a limited amount of previous research had been done on this topic. Especially, it was concluded that the general *initial* deployment problem was something that barely had been studied. Most previous research assumes that the agents are spread randomly in the environment prior to the deployment.

As there was a limited amount of previous work that could be used for solving the problem in this thesis, an approach for for incremental deployment in 1D was proposed. This approach utilized a repelling potential field that aimed to ensure that each agent landed farther from the SCS than any of its neighbors. Conditions that had to be fulfilled for this to be guaranteed was derived, and a deployment was simulated in a self-made simulator [46]. The simulation showed that the potential field did what it was intended to to do, and managed to push the deploying agent to a position that satisfied the exploration condition.

The approach taken for the 1D case proved to be hard to extend to 2D due to the fact that mathematically expressing an *exploration* condition is difficult in 2D. Therefore, a novel deployment scheme were presented, where the agents were subjected to the more classical potential forces. The novelty about the deployment scheme was the way the agents were assigned a target *direction* instead of a target *point*. The calculation of the target direction was calculated based purely on local knowledge and the assumption that neighboring agents could communicate with each other. The deployment strategy was then tested in an obstacle free mission environment, as well as two mission environments containing obstacles. The simulator used for the deployments was similar to the the one used for the 1D case. As the task of the agents is not to provide area-coverage, but to spread out to enable localization of FRs, a fitting performance metric had to be chosen. The metric that ended up being used is called *uniformity*, which is defined as the average local standard deviation of the inter-agent distances. The motivation behind the use of this metric was that uniformly distributed networks tend to be more energy efficient than networks that are spread non-uniformly [51, 52].

5.2 Conclusion

This thesis proposed two novel approaches for incrementally deploying agents in an unknown environment. Due to the lack of previous work on initial incremental deployment, the first proposed approach assumed that the agents only could move in 1D. A repelling potential field was designed so that it would ensure exploration along the real line. The conditions that had to be satisfied for the potential field was derived, and simulations showed that the proposed potential field ended up distributing the agents nicely along the real line.

The second proposed deployment scheme assumed a planar mission environment. As a justifiable exploration condition was hard to define, the deployment scheme had to be adapted. This led to that for the 2D case, a deployment strategy which took inspiration from the traditional use of potential fields was presented. The novelty of this strategy is not only that it considers initial incremental deployment, but also that it implements a target *direction* instead of a target *point*. The vast majority of previous research on the virtual potential field method that includes an attractive force assume that the agents are assigned a target point. The target direction is calculated solely based on local knowledge of the latest deployed agent. More specifically, the target direction is calculated by using the information the latest deployed agent has about the position of its neighbors and the obstacles that it senses.

The deployment scheme was simulated in an obstacle free 10×10 environment as well as two environments that provided a greater challenge for agents. Visually inspecting the final configurations, it became evident that the deployment scheme distributed the agents well in the mission area. However, visual inspection is not satisfactory when the performance of the deployment scheme is to be evaluated. Therefore, the average standard deviation of the inter-agent distances ended up being used. This metric is named *uniformity*, and measures how uniform the geographical distribution of the agents is. The uniformity achieved in the different environments confirmed that the incremental deployment scheme ended up spreading the agents uniformly in the mission area.

5.3 Future work

Given the novelty of the two presented deployment schemes and the limited previous work regarding initial incremental deployment, leads to that there are several ways the work presented in this thesis can be extended. For the scheme that deploys agents along the real line, more work can be done on defining an exploration condition that enables the approach to be extended into 2D. With that being said, a suggested exploration condition is not immediately clear.

Regarding the approach that considers planar mission areas, there are multiple possible topics that could be studied in the future. One such topic is how the deployment scheme should deal with energy depletion of the agents, which potentially can lead to that the network structure is broken. [30] and [54] can be considered good starting points for attacking this problem.

The calculation of the target direction is one of the major contribution of this thesis. It would therefore be of great interest to investigate if there are other ways the local knowledge of the agents can be used to improve this calculation. One suggestion is that the agent calculating the target direction receives information about the obstacles sensed by its neighbors.

If the mission area is small compared to the number of available agents, it will be inevitable that the agents end up being clustered. Examining if there exist some other applicable termination criteria could therefore be something to study in the future. As the uniformity of the network can be calculated when the deploying agent stops, identifying an applicable uniformity value could be a promising starting point for this task.

The uniformity gives information about the geographical distribution of the agents, and not necessarily how well the environment is *explored*. Identifying other ways to evaluate the performance of the deployment might therefore be of great value.

Even if it would require a lot of work, the ultimate goal must be to implement the proposed deployment scheme for 2D using a swarm of Crazyflie 2.1 drones.

Bibliography

[1] Claudio Paliotta, Klaus Ening, and Sigurd Mørkved Albrektsen. “Micro indoor-drones (MINs) for localization of first responders”. In: *Proceedings of the 18th ISCRAM, Blacksburg, VA, USA*. 2021.

[2] O. Khatib. “Real-time obstacle avoidance for manipulators and mobile robots”. In: *Proceedings. 1985 IEEE International Conference on Robotics and Automation*. Vol. 2. Mar. 1985, pp. 500–505.

[3] Erlend Lone. *Efficient area coverage using a network of drones*. Dec. 2020. URL: `https://github.com/erllon/Project-thesis`.

[4] Ebtehal Turki Alotaibi, Shahad Saleh Alqefari, and Anis Koubaa. “LSAR: Multi-UAV Collaboration for Search and Rescue Missions”. In: *IEEE Access* 7 (2019), pp. 55817–55832.

[5] A.J.A. Rivera et al. “Post-disaster rescue facility: Human detection and geolocation using aerial drones”. In: *2016 IEEE Region 10 Conference (TENCON)*. 2016, pp. 384–386.

[6] Shriyanti Kulkarni et al. “UAV Aided Search and Rescue Operation Using Reinforcement Learning”. In: Feb. 2020.

[7] Y.U. Cao et al. “Cooperative mobile robotics: antecedents and directions”. In: *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*. Vol. 1. 1995, 226–234 vol.1.

[8] Luca Iocchi, Daniele Nardi, and Massimiliano Salerno. “Reactivity and Deliberation: A Survey on Multi-Robot Systems”. In: *Balancing Reactivity and Social Deliberation in Multi-Agent Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 9–32.

[9] Manuele Brambilla et al. “Swarm Robotics: A Review from the Swarm Engineering Perspective”. In: *Swarm Intelligence* 7 (Mar. 2013), pp. 1–41.

[10] Jan Barca and Ahmet Sekercioglu. “Swarm robotics reviewed”. In: *Robotica* 31 (May 2013).

[11] Douglas W Gage. *Command control for many-robot systems*. Tech. rep. Naval Command Control, Ocean Surveillance Center Rdt And E Div San Diego CA, 1992.

[12] Shaimaa M. Mohamed, Haitham S. Hamza, and Iman Aly Saroit. “Coverage in mobile wireless sensor networks (M-WSN): A survey”. In: *Computer Communications* 110 (2017), pp. 133–150. URL: `https://www.sciencedirect.com/science/article/pii/S0140366417307235`.

[13] A. Howard, M. Mataric, and G. Sukhatme. “Mobile Sensor Network Deployment using Potential Fields: A Distributed, Scalable Solution to the Area Coverage Problem”. In: *DARS*. 2002.

[14] S. Poduri and G.S. Sukhatme. “Constrained coverage for mobile sensor networks”. In: *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA ’04. 2004*. Vol. 1. 2004, 165–171 Vol.1.

[15] Emre Ugur, Ali E. Turgut, and Erol Sahin. “Dispersion of a swarm of robots based on realistic wireless intensity signals”. In: *2007 22nd international symposium on computer and information sciences*. 2007, pp. 1–6.

[16] Bijan Ranjbar-Sahraei, Gerhard Weiss, and Ali Nakisaee. “A Multi-robot Coverage Approach Based on Stigmergic Communication”. In: *Multiagent System Technologies*. Ed. by Ingo J. Timm and Christian Guttmann. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 126–138.

[17] E. Mathews. “Self-organizing Ad-hoc Mobile Robotic Networks”. In: 2012.

[18] Jonas Svennebring and Sven Koenig. “Building Terrain-Covering Ant Robots: A Feasibility Study”. In: *Auton. Robots* 16 (May 2004), pp. 313–332.

[19] I.A. Wagner, M. Lindenbaum, and A.M. Bruckstein. “Distributed covering by ant-robots using evaporating traces”. In: *IEEE Transactions on Robotics and Automation* 15.5 (1999), pp. 918–933.

[20] Eliyahu Osherovich et al. “Robust and Efficient Covering of Unknown Continuous Domains with Simple, Ant-Like A(ge)nts”. In: *The International Journal of Robotics Research* 27.7 (2008), pp. 815–831. eprint: `https://doi.org/10.1177/0278364908092465`. URL: `https://doi.org/10.1177/0278364908092465`.

[21] Sven Koenig, Boleslaw Szymanski, and Yaxin Liu. "Efficient and Inefficient Ant Coverage Methods". In: *Annals of Mathematics and Artificial Intelligence* 31 (2001), pp. 41–76.

[22] Levent Bayındır. "A review of swarm robotics tasks". In: *Neurocomputing* 172 (2016), pp. 292–321. URL: https://www.sciencedirect.com/science/article/pii/S0925231215010486.

[23] O. Khatib. *Commande dynamique dans l'espace opérationnel des robots manipulateurs en présence d'obstacles*. Ecole nationale superieure de l'aeronautique et de l'espace, 1980. URL: https://books.google.no/books?id=5U0a_gAACAAJ.

[24] Jean-Claude Latombe. *Robot Motion Planning*. USA: Kluwer Academic Publishers, 1991.

[25] Y. Koren and J. Borenstein. "Potential field methods and their inherent limitations for mobile robot navigation". In: *Proceedings. 1991 IEEE International Conference on Robotics and Automation*. 1991, 1398–1404 vol.2.

[26] T. T. Mac et al. "Improved potential field method for unknown obstacle avoidance using UAV in indoor environment". In: *2016 IEEE 14th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*. 2016, pp. 345–350.

[27] S. S. Ge and Y. J. Cui. "New potential functions for mobile robot path planning". In: *IEEE Transactions on Robotics and Automation* 16.5 (2000), pp. 615–620.

[28] Saroj Pradhan et al. "Potential field method to navigate several mobile robots". In: *Appl. Intell.* 25 (Dec. 2006), pp. 321–333.

[29] Safwan Al-Omari and Weisong Shi. "Incremental Sensor Node Deployment for Low Cost and Highly Available WSNs". In: *2010 Sixth International Conference on Mobile Ad-hoc and Sensor Networks*. 2010, pp. 91–96.

[30] Filipe Luiz et al. "Efficient incremental sensor network deployment algorithm". In: (May 2004).

[31] Micael Couceiro et al. "Darwinian Swarm Exploration under Communication Constraints: Initial Deployment and Fault-Tolerance Assessment". In: *Robotics and Autonomous Systems* 62 (Apr. 2014), pp. 528–544.

[32] Micael S. Couceiro et al. "Initial deployment of a robotic team - a hierarchical approach under communication constraints verified on low-cost platforms". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2012, pp. 4614–4619.

[33] Geunho Lee et al. “Three dimensional deployment of robot swarms”. In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2010, pp. 5073–5078.

[34] Simon Bjerg Mikkelsen, René Jespersen, and Trung Dung Ngo. “Probabilistic Communication Based Potential Force for Robot Formations: A Practical Approach”. In: *Distributed Autonomous Robotic Systems: The 10th International Symposium*. Ed. by Alcherio Martinoli et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 243–253. URL: `https://doi.org/10.1007/978-3-642-32723-0_18`.

[35] Andrew Howard, Maja J. Matarić, and Gaurav S. Sukhatme. “An Incremental Self-Deployment Algorithm for Mobile Sensor Networks”. In: *Auton. Robots* 13.2 (Sept. 2002), 113–126. URL: `https://doi.org/10.1023/A:1019625207705`.

[36] Zhiyun Lin, Sijian Zhang, and Gangfeng Yan. “An incremental deployment algorithm for wireless sensor networks using one or multiple autonomous agents”. In: *Ad Hoc Networks* 11.1 (2013), pp. 355–367. URL: `https://www.sciencedirect.com/science/article/pii/S1570870512001175`.

[37] josé Portela and Marcelo Alencar. “Cellular Coverage Map as a Voronoi Diagram”. In: *Journal of Communication and Information Systems* 23 (Apr. 2008).

[38] Yiannis Stergiopoulos and Anthony Tzes. “Coverage-oriented coordination of mobile heterogeneous networks”. In: June 2011, pp. 175–180.

[39] John Stergiopoulos and Anthony Tzes. “Voronoi-based coverage optimization for mobile networks with limited sensing range - A directional search approach”. In: *2009 American Control Conference*. 2009, pp. 2642–2647.

[40] P.E. Rybski et al. “Enlisting rangers and scouts for reconnaissance and surveillance”. In: *IEEE Robotics Automation Magazine* 7.4 (2000), pp. 14–24.

[41] Mac Schwager, Jean-Jacques Slotine, and Daniela Rus. “Decentralized, Adaptive Control for Coverage with Networked Robots”. In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*. 2007, pp. 3289–3294.

[42] Sinan Gezici. “A Survey on Wireless Position Estimation”. In: *Wirel. Pers. Commun.* 44 (Feb. 2008), pp. 263–282.

[43] Sree Divya Chitte, Soura Dasgupta, and Zhi Ding. “Distance Estimation From Received Signal Strength Under Log-Normal Shadowing: Bias and Variance”. In: *IEEE Signal Processing Letters* 16.3 (2009), pp. 216–218.

[44] Zeyuan Li, Pei-Jung Chung, and Bernard Mulgrew. "Distributed target localization using quantized received signal strength". In: *Signal Processing* 134 (2017), pp. 214–223. URL: https://www.sciencedirect.com/science/article/pii/S0165168416303450.

[45] Christopher Triola. "Special Orthogonal Groups and Rotations". In: (Jan. 2009), p. 4.

[46] Erlend Lone. *Simulator_MScThesis*. 2021. URL: https://github.com/erllon/Simulator_MScThesis.

[47] Philip J. Clark and Francis C. Evans. "Distance to Nearest Neighbor as a Measure of Spatial Relationships in Populations". In: *Ecology* 35.4 (1954), pp. 445–453. eprint: https://esajournals.onlinelibrary.wiley.com/doi/pdf/10.2307/1931034. URL: https://esajournals.onlinelibrary.wiley.com/doi/abs/10.2307/1931034.

[48] Hans Pretzsch. "Analysis and modeling of spatial stand structures. Methodological considerations based on mixed beech-larch stands in Lower Saxony". In: *Forest Ecology and Management* 97.3 (1997), pp. 237–253. URL: https://www.sciencedirect.com/science/article/pii/S0378112797000698.

[49] Yousef Erfanifard et al. "Spatial pattern analysis in Persian oak (Quercus brantii var. persica) forests on B&W aerial photographs". In: *Environmental monitoring and assessment* 150 (Mar. 2008), pp. 251–9.

[50] Jordan Wood. "Minimum Bounding Rectangle". In: *Encyclopedia of GIS*. Ed. by Shashi Shekhar and Hui Xiong. Boston, MA: Springer US, 2008, pp. 660–661. URL: https://doi.org/10.1007/978-0-387-35973-1_783.

[51] N. Heo and P. K. Varshney. "A distributed self spreading algorithm for mobile wireless sensor networks". In: *2003 IEEE Wireless Communications and Networking, 2003. WCNC 2003.* Vol. 3. 2003, 1597–1602 vol.3.

[52] Cem Şafak Şahin et al. "Self Organization for Area Coverage Maximization and Energy Conservation in Mobile Ad Hoc Networks". In: Berlin, Heidelberg: Springer-Verlag, 2012, 49–73.

[53] Bergersen Arkitekter. *NTNU Flystripa*. 2011.

[54] M. R. Pac, A. M. Erkmen, and I. Erkmen. "Towards Fluent Sensor Networks: A Scalable and Robust Self-Deployment Approach". In: *First NASA/ESA Conference on Adaptive Hardware and Systems (AHS'06).* 2006, pp. 365–372.

NTNU

Kunnskap for en bedre verden