Master's thesis

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

Magnus Berdal

Incremental Self-Deployment of Multi-Agent Networks in Unknown Environments

A Potential Field-Based Approach

Master's thesis in Cybernetics and Robotics
Supervisor: Damiano Varagnolo
Co-supervisor: Claudio Paliotta
October 2020

NTNU
Kunnskap for en bedre verden

Magnus Berdal

Incremental Self-Deployment of Multi-Agent Networks in Unknown Environments

A Potential Field-Based Approach

Master's thesis in Cybernetics and Robotics
Supervisor: Damiano Varagnolo
Co-supervisor: Claudio Paliotta
October 2020

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

First responders (FRs) in search-and-rescue missions frequently expose themselves to unknown and dangerous environments. Thanks to the increasing capabilities of micro areal vehicles (MAVs) and other mobile agent technologies, some of the risks FRs must take in order to secure the well-being of others can now be mitigated. For example, it is possible to deploy a swarm of MAVs into the environment and use them as beacons that can facilitate the indoor localization of the FRs and others, and in this way, improve the security of all parties involved in search-and-rescue missions.

This master's thesis focuses on the example above and studies how multi-agent networks can be deployed into environments of unknown topology. More precisely, we propose and study a novel potential field-based distributed high-level control law for deploying mobile agents with an unknown range of communication and little environmental awareness. Importantly, this control law considers incremental deployment in opposition to existing literature that focuses instead on simultaneous deployment.

Initially, a distributed control law for incremental deployment within a 1-dimensional obstacle-less environment is investigated and fully characterized. By modelling the control law as an attractive force towards the centre of mass of a virtual set of particles generated by agents already residing within the environment, conditions on the particle masses and locations are found so that an increase in network size guarantees that the network spans a larger portion of the environment.

Inspired by the approach taken for deployments in 1-dimensional environments, the control law is expanded for incremental deployments within unknown 2-dimensional environments. The proposed distributed high-level control law consists of two parts: *a)* a force resulting from an attractive potential field generated by agents already residing within the environment, and *b)* a force resulting from a repulsive potential field generated by sensor readings from range sensors mounted on the agent.

Simulations are performed in two different 2-dimensional environments. In a ten-by-ten meter convex environment, simulations show that deploying sufficiently many agents yields a network that spans a significant portion of the environment. In a non-convex environment including two narrow passages, simulations show that, in most situations, deploying sufficiently many agents yields a network spanning a significant portion of the environment. However, in some situations, no agent manages to pass the narrow passages yielding a network spanning only a fraction of the environment.

The results presented in this report show though that the proposed high-level control law cannot yet be utilized in real-world situations as it lacks robustness. Furthermore, a coverage metric and a deployment termination criterion depending only on information available locally for an agent must be synthesized in order for the proposed approach to be applicable and hopefully yield results that can directly reduce the dangers to which FRs expose themselves.

This project has been carried out in collaboration with the INGENIOUS project funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No 833435 and the Norwegian Research Council project "Autonomous Underwater Fleets" under the grant agreement No. 302435.

Sammendrag

Utrykkningspersonell i søk- og redningsoperasjoner utsetter seg ofte for ukjente og farlige miljøer. Med den økende kapasiteten til mikrodroner og andre mobile robot-teknologier, kan noen av risikoene utrykkningspersonell tar for å sikre andres velvære til dels elimineres. For eksempel er det mulig å distribuere en sverm av mikrodroner i et miljø og benytte dem som ankerpunkter som tilrettelegger for innendørs lokalisering av utrykkningspersonell og andre, og på denne måten forbedre sikkerheten til alle involverte parter i et søk-og-redningsoppdrag.

Denne masteroppgaven fokuserer på det nevnte eksemplet og studerer hvordan multi-agent-nettverk kan distribueres i miljøer med ukjent topologi. Mer presist, foreslås og studeres en ny potensialfeltbasert distribuert høynivå kontrollov for distribusjon av mobile agenter med et ukjent kommunikasjonsområde og liten miljøbevissthet. Essensielt for kontrolloven er at den tar for seg trinnvis distribusjon i motsetning til eksisterende litteratur som fokuserer på parallel distribusjon.

Først undersøkes og karakteriseres en distribuert kontrollov for trinnvis distribusjon av agenter i et 1-dimensjonalt hinderfritt miljø. Kontrolloven modelleres som en attraktiv kraft mot massesenteret til et sett virtuelle partikler, generert av agenter som allerede befinner seg i miljøet. Betingelser på de virituelle partiklenes på masse og posisjon utledes slik at en økning i nettverksstørrelsen garanterer at nettverket dekker en større del av miljøet.

Inspirert av tilnærmingen benyttet for distribusjon i 1-dimensjonale miljøer, utvides kontrolloven for trinnvis distribusjon i ukjente 2-dimensjonale miljøer. Denne høynivå distribuerte kontrolloven består av to deler: *a)* en kraft som resulterer fra et attraktivt potensialfelt generert av agenter som allerede befinner seg i miljøet, og *b)* en kraft som skyldes et frastøtende potensialfelt generert av sensormålinger fra avstandssensorer montert på agenten.

Simuleringer utføres i to ulike todimensjonale miljøer. I et ti-ganger-ti-meter konvekst miljø viser simuleringer at distribusjon av tilstrekkelig mange agenter gir et nettverk som spenner over en betydelig del av miljøet. I et ikke-konvekst miljø, bestående av blant annet to smale passasjer, viser simuleringer at distribusjon av tilstrekkelig mange agenter i de fleste situasjoner gir et nettverk som spenner over en betydelig del av miljøet. Imidlertid oppstår det situasjoner hvor ingen agenter klarer å passere de smale passasjene, som resulterer i et nettverk som bare spenner over en brøkdel av miljøet.

Resultatene som presenteres i denne avhandlingen viser at den foreslåtte høynivå kontrolloven, på dette stadiet, ikke kan anvendes i virkelige situasjoner da den mangler robusthet. Videre må et mål på nettverks-dekning og et kriterium for avslutning av distribusjon, basert utelukkende på informasjon tilgjengelig lokalt for en agent, utvikles for at den foreslåtte kontrolloven skal kunne benyttes og forhåpentligvis gi resultater som direkte kan bidra til å redusere farene søk- og redningspersonell utsetter seg for.

Dette prosjektet er utført i samarbeid med INGENIOUS-prosjektet finansiert av EUs Horizon 2020 forsknings- og innovasjonsprogram under tilskuddsavtale nr. 833435 og Norges forskningsrådsprosjekt "Autonome undervannsflåter"under tilskuddsavtalen nr. 302435.

Acknowledgements

I wish to extend my most sincere gratitude to my supervisors Damiano Varagnolo (NTNU) and Claudio Paliotta (SINTEF), for their guidance, insights and encouragement during my work on the master's project in the fall of 2020 and during my work on this thesis during the spring of 2021. Furthermore, I want to thank Erlend Vevle Lone, Behdad Aminian and Simon Hoff, who throughout this semester have participated in weekly meetings and contributed to my progress, but most of all for making the experience of writing a master's thesis an enjoyable one.

I would also like to thank my brothers for being two cool dudes and my parents for their support and for encouraging me to stay diligent in my studies. A huge thank you to my study mates for providing me with a great deal of motivation and for lighting up my days during our daily lunch- and coffee breaks.

Trondheim, June 6, 2021

Magnus Berdal

Contents

Chapter 1

Introduction

Natural- and man-made disasters are unexpected events that hit society, causing damage to infrastructure and loss of lives. Globally, an average of 60000 lives is lost due to natural disasters per year [1]. A first responder (FR) is one among those responsible for going immediately to the scene of an accident or emergency to provide assistance [2] and plays an integral role in the event of a disaster. They will typically be employed by the emergency services such as the police, fire department, or health services and take it upon themselves to secure the health of people, property, and the environment. Often this includes sacrificing their own safety in order to secure that of others. The need for quick action gives them no other choice.

Search-and-rescue (SAR) personnel, in particular, expose themselves to significant risks. Burning buildings, collapsed or flooded caves and landslides, are just some examples of the rapid-changing hazardous environments in which SAR personnel can find themselves. Often they initiate their mission without any knowledge of what is waiting for them on the other side. Imagine a burning building. The local fire department has just arrived. There is no time to spare, so the first firefighters enter to save the lives of those inside before inspecting the floor plan. It might not even be valid as the roaring fire continues to eat up the walls and support beams for the roof. Still, the firefighters enter in order to rescue the people inside. If something is to happen to the entering firefighters, there is no way to locate them, and the ones entering to save them still do not know what is waiting for them inside.

Advancements in technology can be used to limit the dangers to which SAR personnel expose themselves. With the increasing performance and decreasing cost of mobile robots [3], and the ability to outfit them with whatever equipment one might think of, it is relevant to think of how this might be used to allow for faster, more efficient, safer and possibly cheaper SAR operations. Autonomous air- and ground vehicles, in particular, could profit the perception and action capacities of FRs [4]. In addition, using disposable autonomous

robots for tasks such as mapping unknown environments or identifying the locations of victims could drastically improve the safety of FRs.

In the example stated above, mobile robots could be dispatched into the burning house and continuously feed the firefighters with a real-time map of the environment. Robots could also enter the building to set up an ad-hoc network, a dynamic and self-configuring network formed by a collection of mobile nodes [5], used for locating FRs inside GNSS denied environments. Alternatively, the collection of robots could supply information about events happening, such as changes in the environment or the presence of poisonous gasses.

The INGENIOUS project is an EU-funded project that has as its mission to develop and test a next-generation integrated toolkit (NGIT) for collaborative response. The toolkit is based on several components such as smart boots and uniforms, localization tags, as well as different areal drones that will all enhance the perception capabilities of the participants involved in a SAR mission [4]. Included in the toolkit are micro indoor aerial drones (MINs).

The MINs are quadcopters characterized by their small size and limited payload. Due to the limited payload, MINs cannot carry advanced sensors that could aid in precise detection of the environment around them. Furthermore, they are equipped with ultra-wideband modules/other radio modules allowing for MIN-to-MIN or MIN-to-Base communication. The main objective of the MINs is to enter a dangerous GNSS denied environment of unknown layout and extent and create a connected ad-hoc mesh network consisting of a base station and a collection of MINs. The MINs are to be used as beacons for distributed localization of FRs within the environment.

Developing a distributed deployment scheme for micro-drones in order to provide such a localization network is the task of this thesis. Although this work is motivated by the INGENIOUS project, and the proposed deployment scheme is tailored to fit the constraints imposed by the limited payload of the MINs, we focus here on the deployment of a general multi-agent system.

Chapter 2

Background

This chapter introduces the basic concepts upon which the contribution of this thesis is built. In particular, Section 2.1 provides a definition of Mobile Wireless Sensor Networks, Section 2.2 introduces the concept of coverage, and in Section 2.3 the task of self-deployment of mobile wireless sensor networks is briefly presented. In Section 2.4, a literature review is presented. Here, previous work on the self-deployment and coverage problem using mobile sensor networks, inspiring the approach taken in this thesis, is presented. In Section 2.5, the problem of target localization and methods for solving it in a distributed manner are introduced for completeness. Section 2.6 presents the concept of potential fields, upon which the deployment scheme presented in this thesis is built.

2.1 Mobile Wireless Sensor Networks

Wireless Sensor Networks (WSNs) consist of large numbers of sensor nodes deployed to observe or detect certain phenomena that occur in a region of interest (ROI) [6]. In a mobile WSN (M-WSN), in addition to sensing- and communication capabilities, nodes have locomotive platforms allowing them to deploy/redeploy in order to increase the quality of service (QoS) within the ROI.

2.2 Coverage

Maintenance of the QoS within an ROI using multi-robot systems is a paradigm in itself. In [7], Gage defines coverage as "[...] the maintenance of a spatial relationship which adapts to specific local conditions to optimize the performance of some function [...]".

Coverage is further partitioned into three disciplines: Blanket coverage, in which the goal for the M-WSN is to reach a static configuration such that the total detection area is maximized; Barrier coverage, where the goal is to reach a static configuration where the probability of undetected penetration of the barrier set up by the wireless sensor nodes is minimized; and Sweep coverage which is, in essence, equivalent to a moving barrier.

2.3 Self-deployment

Self-deployment is the task of enabling mobile sensor nodes to "[...] deploy themselves in an environment without central coordination" [8]. Self-deployment is done in one of two fashions: incrementally or concurrently. In incremental self-deployment, only a single node deploys at any given time. When the currently deploying node has settled at some location, the next one deploys. In concurrent self-deployment, all nodes are allowed to re-position themselves at the same time.

2.4 Previous work on the deployment- and coverage problem

Howard et.al. present in [9] an *incremental* self-deployment algorithm for M-WSNs with the goal of achieving blanket coverage. The proposed algorithm assumes that the environment is static (unchanged during deployment) and not known a priori. The positions of all nodes are assumed to be known in some global coordinate frame. Nodes are equipped with an omnidirectional range sensor of known maximum range and a broadcast communication device enabling them to communicate with a base station, either directly or via multi-hop communication through other nodes.

The proposed algorithm defines three different states for a node: *waiting*, meaning the node has not yet deployed; *active*, meaning the node is currently deploying; and *deployed*, meaning the drone has settled. The locations to which the *active* nodes should travel are chosen by analyzing a reachability grid map of the environment generated by fusing sensor data from the *deployed* nodes into a common map. The algorithm applies heuristics in order to determine the cell in the reachability grid (goal location) that ensures that (1) nodes deploy to the boundary between reachable and unreachable space, (2) nodes cover the largest area of presently unknown space and (3) nodes are deployed to locations ensuring line-of-sight visibility from at least one other node.

Simulations of the sequential deployment of 50 nodes are performed. The metrics used to evaluate the performance of the algorithm are the elapsed time and coverage.

Coverage is measured as the number of free cells (cells not containing obstacles while being in the field of view of at least one node) in the occupancy grid multiplied by cell size. The proposed algorithm yields results "close to those obtained using a greedy model-based strategy" [9], in which the environment is known a priori.

In [10], Damer et.al. present the Backbone Dispersion Algorithm (BDA). The proposed algorithm makes use of signal intensity in order to *concurrently* self-deploy a connected M-WSN in an unknown environment. It is assumed that two moving nodes will never lose contact with a stationary node simultaneously. The assumption is motivated by the fact that the speed of communication between nodes is much greater than the maximum speed of the nodes [10]. Furthermore, it is assumed that nodes possess a sonar array that provides ranges to obstacles and a communication device allowing for information passing between nodes.

By four simple rules (see Table 2.1), BDA maintains and expands a set of static sensor nodes ("backbone") over time. Nodes that are also backbone nodes are guaranteed never to move.

Node in direct contact with	Action	Goal
0 nodes	Random walk	Regain contact with the backbone
≥ 2 nodes ≥ 1 backbone node	Random walk	Discover uncovered areas
1 backbone node	Remain stationary until one other node wanders into range	Stay in contact with the backbone
≥ 1 node 0 backbone nodes	Join backbone and tell one neighbour to join the backbone	Extend backbone to an unexplored area

Table 2.1: Rules used in BDA. Table recreated from [10].

The random walk behaviour presented in Table 2.1 is defined as follows: For each time step turn by (1) a random angle in the range of $\pm 10^\circ$ if the node does not detect any obstacles; (2) a random angle in the range $[120^\circ, 240^\circ]$ if the node detects an obstacle. Thus, obstacle avoidance is implemented by simply turning away from the detected obstacle.

The authors conclude that the dispersion yielded by BDA is effective due to the fact that for any pair of nodes that join the backbone, one of them is completely disconnected from all nodes previously in the backbone.

Pac et al. present in [11] an adaptive, fluid dynamic-inspired approach as a scaleable, robust solution to the problem of *concurrent* self-deployment of M-WSNs in unknown en-

vironments, for the purpose of blanket coverage. Inspired by the diffusive, self-spreading nature of compressible fluids, the M-WSN as a whole is modelled as a single fluid body. Sensor nodes are modelled as infinitesimal fluid elements, and node velocity, pressure and local density are defined as flow variables.

Nodes are assumed to be homogeneous, equipped with range sensors of known maximum range and a communication device allowing them to pass information to other nodes within a pre-determined distance. Furthermore, it is assumed that nodes know their velocity relative to a local or global frame and that they can determine both the relative positions and velocities of other nodes within communication range.

Dispersion of the deployed sensor nodes is caused by the dispersive nature of compressible fluids, in which the fluid will conform with the outlines of its container [6]. In order to preserve connectivity of the M-WSN, artificial viscosity termed *damping viscosity* is applied to the reference velocity of sensor nodes.

Obstacle avoidance comes as an effect of the only boundary condition on inviscid fluid flow [11]: fluid elements immediately close to a surface must have a velocity vector parallel to the surface. The boundary condition causes sensor nodes to follow the boundaries of the environment whenever a boundary is detected. In order to avoid clustering of sensor nodes at the boundaries of the environment, fluid elements (sensor nodes) in the vicinity of obstacles raise their temperature. An increase in temperature leads to an increase in local pressure. Nearby fluid elements (sensor nodes) are repulsed due to the higher pressure region generated along the obstacle's surface, causing them to move away.

Deployment is performed by dynamically injecting new nodes into the environment at some pre-determined initial locations. The dynamic injection of nodes is based on the local density at the initial deployment location(s). When the previously deployed sensor nodes spread and reach an equilibrium, new successors are placed at the initial deployment location(s) in order to attain or preserve some pre-determined density level. As pouring water into a half-full cup "rearranges" the fluid particles already present in the cup, injecting new nodes affects the locations of the already deployed nodes.

Assuming that coverage in both sensor- and communication range is deterministic, i.e. the disk with radius equal to the sensor-range is covered, it is concluded that coverage and robustness come as an effect of the self-spreading nature of fluids. Through simulators, the adaptive injection of new nodes is shown to provide effective deployment and coverage in unknown dynamic environments.

Howard et.al. present in [12] a *concurrent* virtual potential field-based approach to the blanket coverage problem in static but unknown environments. Their approach is inspired by electrostatic potential in which equally charged particles exert repulsive forces on one another. Sensor nodes are assumed to be equipped with omnidirectional range sensors of

known maximum range. The range sensors are assumed to be able to distinguish between obstacles and other nodes.

Modelling both obstacles and sensor nodes as positively charged particles, both obstacle- and collision avoidance is attained due to the repulsive effect that equally charged particles have on each other. Both node-on-node and obstacle-on-node forces are dependent only on relative distances, making the presented approach suitable for on-line local computation as no global localization is needed.

A simulated experiment is conducted in which the deployment is initiated with 100 sensor nodes placed in a dense configuration within a complex static environment. Results show an initially rapid increase in both coverage and average node separation, followed by a slower convergence to some constant value. At termination, the final configuration attained by the simulated deployment of 100 nodes yields a tenfold increase in coverage compared to the initial configuration. The authors conclude that "area coverage [...] can emerge from a combination of purely local rules"[12].

As opposed to in [12] where only repulsive forces are used, Yu et al. present in [13] a potential field-based *concurrent* deployment scheme, with the goal of blanket coverage, that utilizes a mix of attractive and repulsive forces. It is assumed that nodes are homogeneous, equipped with communication devices and sensors of known sensing- and communication range. The sensing area of a node is the disk centred at the position of the node with radius determined by the sensor range. Furthermore, it is assumed that the location of all nodes can be acquired (through GNSS or by other means).

In the virtual force approach presented in [13], each node is affected by three forces: (1) a repulsive force exerted by obstacles; (2) an attractive force exerted by areas of interest; and (3) a composite force exerted by adjacent nodes. The composite force is a sum over node-to-adjacent-node forces, which may be repulsive or attractive based on the inter-nodal distance. The “adjacent relationship” of nodes are defined using Delaunay triangulation [14] of the nodes’ positions.

Simulations are performed in both a 100×100 square meter environment and in an unbounded environment. The results are compared to those obtained in [15], in which Delaunay is not used. Starting from an initial random configuration, the mixed virtual potential field-based approach using Delaunay triangulation yields "[...] better coverage rate and faster convergence time [...] which proved that the proposed approach is an effective node deployment algorithm".

Heo and Varshney present in [16] the Distributed Self-Spreading Algorithm: a distributed self-deployment algorithm for mobile sensors in known environments. Nodes are assumed to possess ideal communication and sensing devices giving communication and omnidirectional sensing within disks restricted by some maximum radius. Nodes are said

to be neighbours if they are within communication range of each other.

The area of the region of interest (ROI) and the number of nodes to be deployed are assumed known a priori, and are used as parameters to the proposed virtual force scheme. Initiated in a random topology within the ROI, repulsive forces between neighbouring nodes are used to disperse the WSN throughout the ROI.

Variation in magnitude of inter-nodal forces depends on two variables: it increases with density factor and decreases with inter-nodal distance. The density factor is computed as the ratio between local density (number of neighbours) and expected density (average number of nodes needed to cover the ROI). In sparse areas, the density factor is small, giving small inter-nodal forces. In dense regions, the opposite holds.

Four metrics are used to quantify the quality of the topology generated by the proposed deployment scheme: coverage, the ratio of sensor-covered area to ROI area; uniformity, the average standard deviation of the distances between nodes; time, time elapsed until the WSN reaches a static topology; and distance, the average distance travelled by the WSN. Through simulations, it is shown that the deployment scheme proposed by Heo and Varshney "[...] successfully obtains a uniform distribution from an initial uneven distribution [...]"[16] and outperforms a simulated-annealing driven deployment algorithm in all performance metrics.

2.5 Target localization in (M-)WSNs

Here, a brief definition of target localization is presented along with the different approaches for performing distributed target localization in WSNs. Although the objective of this thesis does not include performing distributed localization, the work presented here lays the groundwork for future development, and a brief presentation is made for completeness.

The problem of target localization is that of inferring an estimate on the position of a target using multiple noise polluted measurements [17]. Given a set of beacons, there are several ways of inferring estimates of a target's position depending on the type of measurements made: Received Signal Strength (RSS), Bearing Only, Range Only, Time of Arrival (TOA), Time Difference of Arrival (TDOA) or a combination of measurements [18].

Performing distributed localization using TOA or TDOA measurements requires accurate synchronization of the internal clocks of the nodes (beacons) used for localization. In (M-)WSNs comprised of nodes with limited computational power, complex synchronization schemes [19, 20] might be too expensive. For Bearing Only localization, relative

angle measurements from an antenna array are required [18] which may also be too expensive. RSS-based localization demands no extra data transmission and is relatively inexpensive to implement [18]. Due to this, RSS-based localization is the most appropriate scheme in (M-)WSNs comprised of low-cost nodes with limited computation power and limited sensing capabilities, where energy usage is an issue.

2.6 Virtual potential fields

Virtual potential fields (VPFs) are commonly used in path-planning and coverage/exploration tasks by many researchers due to their elegance, simplicity, and safety [21]. In a VPF scheme, mobile nodes are regarded as point particles under the influence of a potential field.

A scalar potential field, $U : \mathbb{N}^N \longrightarrow \mathbb{R}_{\geq 0}$, relates a point in N-dimensional space to a scalar energy level. The potential field produces a force on the particles affected by it. By Equation (1.16) in [22], the force on a particle affected by a potential field relates to the scalar potential field, U, by:

$$\mathbf{F} = -\nabla U(\mathbf{x}) \tag{2.1}$$

where $\mathbf{x} = \begin{bmatrix} x_1 \dots x_N \end{bmatrix}^T \in \mathbb{R}^N$ is the position of a particle relative to some given origin and ∇ is the gradient operator:

$$\nabla U(\mathbf{x}) = \begin{bmatrix} \frac{\partial}{\partial x_1} U(\mathbf{x}) & \dots & \frac{\partial}{\partial x_N} U(\mathbf{x}) \end{bmatrix}^T := \frac{\partial}{\partial \mathbf{x}} U(\mathbf{x})$$

There are two distinct types of scalar potential fields: attractive and repulsive. An attractive potential field defines a point of zero potential. The potential of a particle influenced by an attractive potential field increases as its distance to the point of zero potential increases. In literature, attractive potential fields are conventionally modelled as quadratic functions of the distance from the point of zero potential [23]:

$$U_{att}(\mathbf{x}) = \alpha \|\mathbf{x}\|^2$$

where $\alpha > 0$ is some scaling factor and $\mathbf{x}$ is the position of a particle relative to the point of zero potential. Figure 2.1 shows a depiction of a typical quadratic attractive potential field and the force exerted on a particle affected by the potential field.

A repulsive potential field defines a point of maximum (possibly infinite) potential. When a repulsive potential field influences a particle, the particle's potential energy decreases as its distance to the point of maximum potential increases. Inspired by electrostatic potential fields, repulsive potential fields used for mobile robot control are commonly modelled as [12]:

$$U_{rep} = \beta \|\mathbf{x}\|^{-1}$$

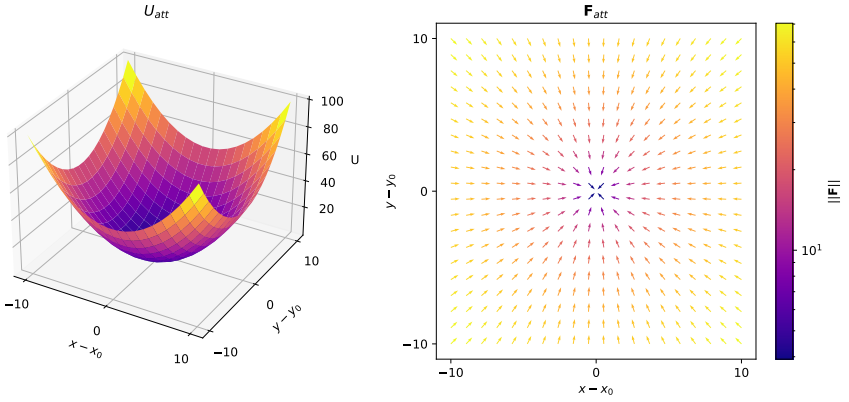


Figure 2.1: An attractive potential field $U = \frac{1}{2}((x-x_0)^2 + (y-y_0)^2)$. Left: Potential as function of position relative to the point of zero potential (at (x_0, y_0)). Right: Force on particle affected by potential field as function of position relative to the point of zero potential.

where $\beta > 0$ is some scaling factor and $\mathbf{x}$ is the position of a particle relative to the point of infinite potential.

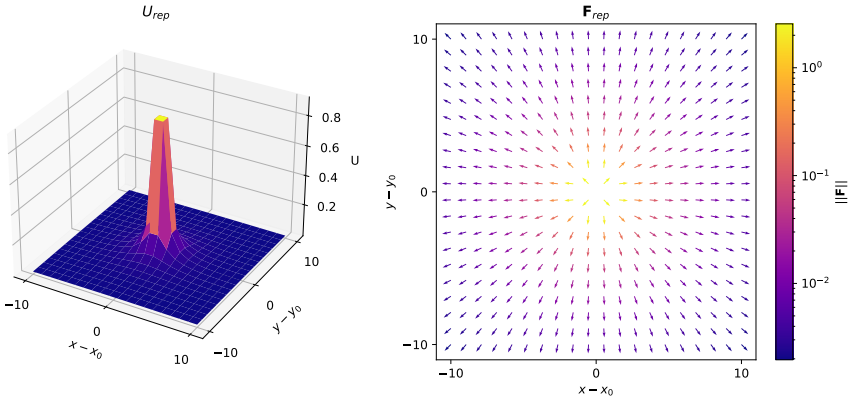


Figure 2.2: A repulsive potential field $U = \frac{1}{2}\left((x-x_0)^2 + (y-y_0)^2\right)^{-1/2}$. Left: Potential as function of position relative to the point of infinite potential (at (x_0, y_0)). The U-axis is clipped at 0.8. Right: Force on particle affected by potential field as function of position relative to the point of infinite potential.

Chapter 3

System description

In this section, we introduce the platform upon which the contribution of this thesis is built. In Section 3.1, we introduce the RSS-mapped offset distance: a quantity assumed available to all nodes through communication. The reference frames used to relate sensor data to nodal positions are presented in Section 3.2. In Section 3.3 and Section 3.4 the properties of two types of sensor nodes: beacons and agents, are presented. A definition of a generic environment is presented in Section 3.5. Lastly, the properties and functionality of a range sensor are presented in Section 3.6.

3.1 Received signal strength offset distance

A signal travelling from one node to another experiences fast fading, shadowing, and path-loss [24]. Many models, most commonly the path-loss model under log-normal fading [18, 24, 25], have been proposed in order to predict the signal strength at a receiver at a certain distance from a sender, taking into account the aforementioned phenomena. Furthermore, Chitte and Dasgupta present in [26] a series of estimators of distance given RSS measurements. As an accurate model of signal strength nor RSS-based distance estimation falls within the scope of this thesis, all environmental effects on a signal are ignored, and a simple and deterministic model is applied.

The strength of a signal sent from an entity j, received by an entity i is assumed to be a descending function of the inter-entity distance $d_{i,j} = \|\mathbf{x}_i - \mathbf{x}_j\|$, where $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^2$ denote the position of the receiving- and transmitting entity respectively. We assume that there exists a map, $\xi_{i,j}$, that maps the RSS (Received Signal Strength) value at a receiver, i, of a signal sent from an entity j, $RSS_{i,j}$, to a distance. It is assumed that the map fulfills:

$$\frac{\partial \xi_{i,j}}{\partial RSS_{i,j}} > 0, \frac{\partial RSS_{i,j}}{\partial d_{i,j}} \leq 0 \implies \frac{\partial \xi_{i,j}}{\partial d_{i,j}} = \frac{\partial \xi_{i,j}}{\partial RSS_{i,j}} \frac{\partial RSS_{i,j}}{\partial d_{i,j}} \leq 0 \tag{3.1}$$

where $\xi_{i,j} \in \mathbb{R}$ is called the RSS-mapped offset distance between entity i and j. The assumption that increasing internodal distance has an inverse relationship with RSS value is motivated by the standard path-loss model.

It is assumed that the RSS-mapped offset distance is non-negative and bounded above by a constant $\bar{\xi}$. Formally:

$$0 \leq \xi_{i,j} \leq \bar{\xi} \; \forall \, i, j$$

where $\bar{\xi}$ is the maximum offset distance, achieved when the RSS at the receiver attains its maximum value.

In addition, sender-receiver-symmetry is assumed, meaning the RSS-mapped offset distance between an entity j relative to another entity i is assumed to be equal to the RSS-mapped offset distance from entity i to j:

$$\xi_{i,j} = \xi_{j,i} \; \forall \, i, j$$

The final assumption made is that of receiver invariance. For two receiving entities, i and j, and a transmitting entity, k, it is assumed:

$$\mathbf{x}_i = \mathbf{x}_j \implies \xi_{i,k} = \xi_{j,k} \; \forall \, i, j, k \tag{3.2}$$

where $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^2$ denote the position of the two receiving entities.

In simulations, the following map conforming with (3.1) - (3.2) is used:

$$\xi_{i,j} = \begin{cases} \bar{\xi} & , d_{i,j} < d_{perf} \\ \frac{\bar{\xi}}{2} + \frac{\bar{\xi}}{2} \cos\left(\pi \frac{d_{i,j} - d_{perf}}{d_{none} - d_{perf}}\right) & , d_{perf} \leq d_{i,j} \leq d_{none} \\ 0 & , d_{i,j} > d_{none} \end{cases} \tag{3.3}$$

where $d_{i,j}$ is the *de facto* distance between the sending- and receiving entity.

Figure 3.1 shows a plot of the RSS-mapped offset distance defined in (3.3) as a function of the actual distance between two entities.

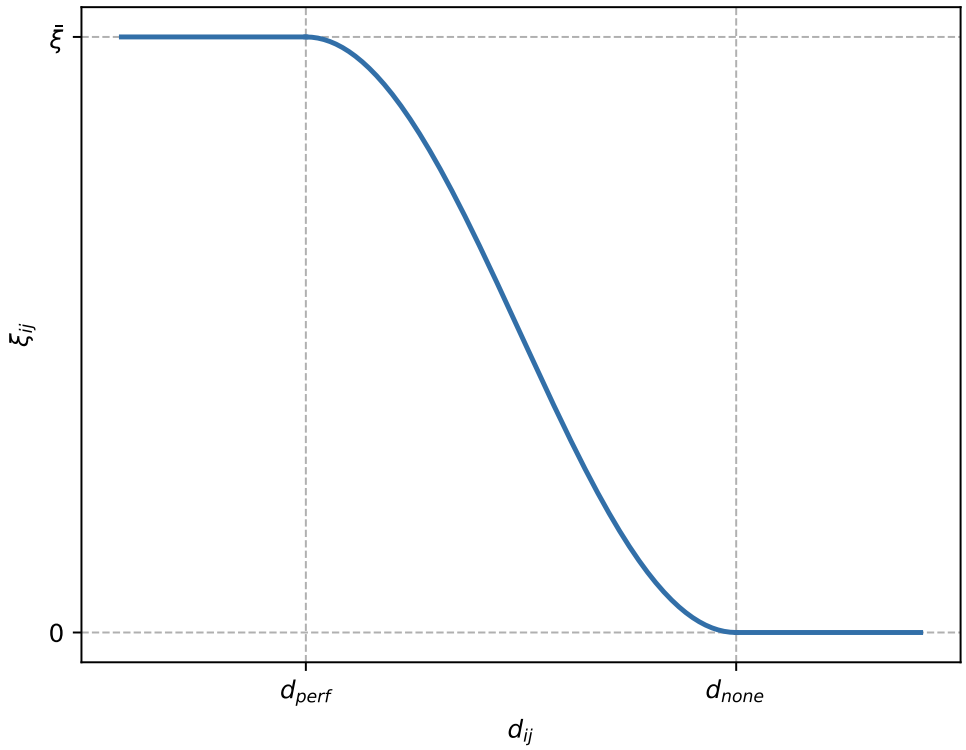


Figure 3.1: RSS-mapped offset ($\xi_{i,j}$) vs. actual distance ($d_{i,j}$) between two entities i and j.

3.2 Reference frames

It is convenient to represent motion/position/measurements relatively to different reference frames. Sensors usually retrieve data relative to some host, whereas the position of an entity is usually represented relative to some fixed frame. Due to this, two reference frames are defined. A depiction of the relation between these two frames is shown in Figure 3.2.

3.2.1 The inertial NE frame

The East-North coordinate system, denoted by $\{n\} = (x_n, y_n)$ with origin $\mathbf{o}_n \in \mathbb{R}^2$ is considered inertial with its origin fixed at some point on the surface of the Earth. The x_n-axis points to Earth's true east, and the y_n-axis point to Earth's true north. A superscript n is used to define a vector expressed relative to the inertial NE frame.

3.2.2 The HOST frame

The HOST frame, denoted as $\{h\} = (x_h, y_h)$, is a moving coordinate system with it's origin, $\mathbf{o}_h \in \mathbb{R}^2$, fixed to the host, h, of interest. Assuming a host, h, has it's nose pointing in a direction ψ_h relative to the x_n-axis (positive counter clockwise), the x_h-axis points in the direction defined by ψ_h and the y_h-axis points in the direction $\psi_h + \frac{\pi}{2}$. A vector defined relative to a HOST frame, $\{h\}$, is denoted with a superscript h.

3.2.3 Transformations between frames

To relate a HOST frame, $\{h\}$, to the NE frame, $\{n\}$, a displacement matrix, $\mathbf{T}_h^n \in SE(2)$ is defined as [27]:

$$\mathbf{T}_h^n = \begin{bmatrix} \mathbf{R}_h^n & \mathbf{o}_h^n \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \tag{3.4}$$

where $\mathbf{0}_{1\times 2}$ is a 1-by-2 matrix of all zeros, $\mathbf{o}_h^n$ is the origin of the HOST frame relative to the NE frame, and $\mathbf{R}_h^n \in SO(2)$ is defined as [28]:

$$\mathbf{R}_h^n = \mathbf{R}(\psi_h) = \begin{bmatrix} \cos(\psi_h) & -\sin(\psi_h) \\ \sin(\psi_h) & \cos(\psi_h) \end{bmatrix}$$

where ψ_h is the orientation of the x_h-axis relative to the x_n-axis.

For a vector, $\mathbf{x}^h$, relative to a HOST frame, its representation relative to the EN frame is computed as [27]:

$$\hat{\mathbf{x}}^n = \mathbf{T}_h^n \hat{\mathbf{x}}^h$$

where $\hat{\mathbf{x}}^n = \begin{bmatrix} (\mathbf{x}^n)^T & 1 \end{bmatrix}^T$ and $\hat{\mathbf{x}}^h = \begin{bmatrix} (\mathbf{x}^h)^T & 1 \end{bmatrix}^T$ are the *homogeneous representations* of the vectors $\mathbf{x}^n$ and $\mathbf{x}^h$ respectively [27] and $\mathbf{T}_h^n$ is defined in (3.4). By abuse of notation, the transformation of a vector $\mathbf{x}^h$ represented in a HOST frame to its representation in the EN frame will be denoted by:

$$\mathbf{x}^n = \mathbf{T}_h^n \mathbf{x}^h$$

where the conversions between homogeneous representations of vectors are implicit.

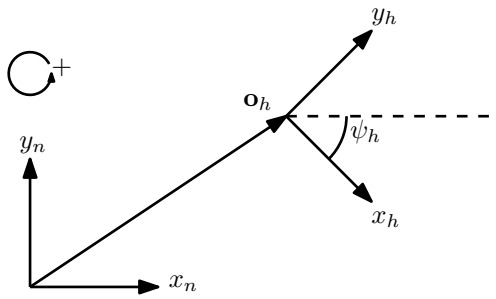


Figure 3.2: Example of a host frame, $\{h\}$, and the inertial frame $\{n\}$. $\mathbf{o}_h$ denotes the position of a host, h, relative to the NE-frame, and ψ_h denotes the orientation (positive counter-clockwise) of the the host relative to the x_n-axis.

3.3 Beacon

A beacon, defined by an index i, is an entity equipped with a communication device allowing it to communicate with other entities. The position of a beacon i, $\mathbf{x}_i^n$, is constant in the inertial EN frame. In general, the position of a beacon will be denoted without a superscript n as it usually relates to quantities also expressed relative to the inertial EN frame. Although not explicitly defined by the 2D position vector of a beacon, it is assumed that beacons are at all times located at ground level.

A beacon i has a neighbour set as defined as:

$$\mathcal{N}_b(i) = \{j \in \mathcal{B} \setminus \{i\} : \xi_{i,j} \geq \xi_{\mathcal{N}} > 0\} \tag{3.5}$$

where $\mathcal{B}$ is the set of all beacons in existence, $\xi_{i,j}$ is the RSS-mapped offset distance between beacon i and beacon j, and $\xi_{\mathcal{N}}$ is the neighbour threshold. Note that only the RSS-mapped offset distance to another beacon, j, determines whether or not j is considered a neighbour of beacon i. Thus there are no constraints on the *de facto* distance between neighbouring beacons.

A beacon i is said to cover the area, c_i, defined by:

$$c_i = \{\mathbf{x}_j : \xi_{i,j} \geq \xi_{\mathscr{N}}\} \tag{3.6}$$

where $\mathbf{x}_j$ is the position of an entity with communication capabilities. It is noteworthy that the coverage supplied by a beacon depends on the unknown RSS-mapped offset distance and is hence itself an unknown region.

3.4 Agent

An agent, defined by an index i, is a robot equipped with a set of four range sensors (see section 3.6) and a communication device allowing it to communicate with other entities. Furthermore, an agent is equipped with a locomotive platform (in the case of this thesis, the locomotive platform is a set of propellers), allowing it to reposition itself.

The position of agent i, represented in the inertial EN (East-North) frame, is denoted by $\mathbf{x}_i^n$, and its orientation relative to the inertial x-axis (positive counter-clockwise) is denoted by ψ_i. In general, the position of an agent will be denoted without a superscript n as it usually relates to quantities also expressed relative to the inertial EN frame. In calculations including vectors expressed in different frames, the superscript n is added for clarity Although not explicitly stated in the 2D position vector of an agent, it is assumed that it at all times flies at some altitude above ground level, preventing it from colliding into beacons.

The neighbour set of an agent i is defined as:

$$\mathscr{N}_a(i) = \{j \in \mathscr{B} : \xi_{i,j} > \xi_{\mathscr{N}} > 0\} \tag{3.7}$$

where $\mathscr{B}$ is the set of all beacons, $\xi_{i,j}$ is the RSS-mapped offset distance between agent i and beacon j, and $\xi_{\mathscr{N}}$ is the neighbour threshold. As for beacons, there are no constraints on the *de facto* distance between an agent and its neighbours. It is assumed that an agent can learn the position of all its neighbours through direct communication.

It is noteworthy that an agent i's neighbour set does not include beacons, j, that satisfy $\xi_{i,j} = \xi_{\mathscr{N}}$ as opposed to the neighbour set of beacons (3.5). Thus, for an agent, i, and a beacon, j:

$$\mathbf{x}_i = \mathbf{x}_j \implies \mathscr{N}_a(i) \subseteq \mathscr{N}_b(j) \tag{3.8}$$

Note that an agent guaranteed not to move at any time in the future possesses all of the properties of a beacon, except for a more strict neighbour set. Hence, agents that are guaranteed never to move are re-instantiated as beacons, and their index is added to the set of beacons, $\mathscr{B}$.

3.4.1 Dynamic model

An agent, i, is assumed to have single-integrator positional dynamics:

$$\dot{\mathbf{x}}_i = \mathbf{u}_i \tag{3.9}$$

where $\mathbf{u}_i \in \mathbb{R}^2$ is the control input. Assuming single-integrator dynamics simplifies the derivation of a high-level control law [29], and is based on the assumption that there are lower-level controllers in place that cancel any inherent dynamics and enforce (3.9) [30].

3.5 Environment

An obstacle O, defined by its boundary ∂O restricts the movement of entities, as it is impossible for an entity to travel through the boundary of an obstacle. Formally:

$$\mathbf{x} \notin O$$

where $\mathbf{x}$ is the position of an entity. The boundary of an obstacle may or may not enclose a subset of $\mathbb{R}^2$.

An environment E is defined as a set of obstacles:

$$E = \{O_1 \dots O_{N_O}\} \tag{3.10}$$

where N_O denotes the number of obstacles in the environment.

3.6 Range sensor

A range sensor with maximum sensing range R_s is a device capable of measuring the distance to obstacles. Assuming a range sensor is placed at $\mathbf{x}_{rs}^n$ in the inertial EN frame with orientation ψ_{rs} relative to the x-axis of the inertial frame, it returns the measured range, r, to the nearest visible obstacle according to:

$$r = \begin{cases} \min_{\lambda \in \mathscr{R}} \lambda & , \mathscr{R} \neq \emptyset \\ R_s & , \text{otherwise} \end{cases}$$

where

$$\mathscr{R} = \{0 \leq \lambda \leq R_s : \mathbf{x}_{rs}^n + \lambda \mathbf{R}(\psi_{rs})\mathbf{e}_x \in \partial O, O \in E\}$$

where $\mathbf{e}_x = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$ denotes the unit x-vector, E is as defined in (3.10) and $\mathbf{R}(\cdot) \in SO(2)$ is the two dimensional rotation matrix [28].

A range sensor is usually mounted on a host. Denoting the position (mounting point) of a range sensor in the HOST frame by $\mathbf{x}_{rs}^h$, the position of the sensor in the inertial frame is defined as:

$$\mathbf{x}_{rs}^n = \mathbf{T}_h^n \mathbf{x}_{rs}^h = \mathbf{R}(\psi_h)\mathbf{x}_{rs}^h + \mathbf{x}_h^n$$

where $\mathbf{T}_h^n$ is as defined in (3.4), ψ_h is the orientation of the x_h-axis relative to the inertial x_n-axis and $\mathbf{x}_h^n$ is the position of the host relative to the inertial EN frame.

Furthermore, denoting a range sensor's orientation relative to the host's x_h-axis by $\psi_{rs,h}$ its orientation relative to the inertial x_n-axis is:

$$\psi_{rs} = \psi_h + \psi_{rs,h}$$

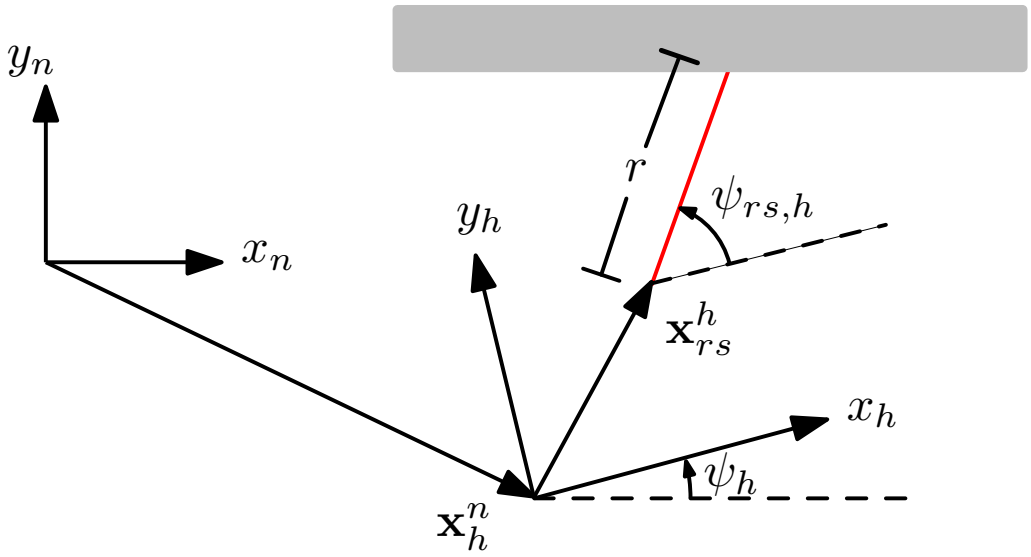


Figure 3.3: Range sensor setup. The range sensor placed at $\mathbf{x}_{rs}^h$ in the HOST frame returns the measured range, r, to the obstacle (grey).

Chapter 4

Objective and novelty

The objective of this thesis is to develop high-level control laws for *incrementally* dispersing mobile nodes into an *unknown* environment, using only local information, with the ultimate goal of providing a (1-)connected ad-hoc localization network. The initial deployment of mobile robots has not been fully addressed in literature, and only a small number of studies assessing its relevance have been conducted [31]. In particular, the problem of incremental self-deployment using only local information has to our knowledge, not been previously addressed, posing an extra challenge as the deployment strategy presented here must be developed with little aid from literature.

In literature, it is often assumed that nodes possess omnidirectional sensors or cameras, allowing them to accurately detect and measure the area around them [9, 10, 11, 12, 13, 15]. As discussed in Chapter 3, agents/beacons only possess sensors allowing them to measure the distance to obstacles in four directions, meaning area measurements are not readily available for an agent/beacon. This reduced assumption on the sensor capabilities of nodes contributes to the novelty of the findings in this thesis.

Furthermore, it is assumed that beacons can localize an entity if it can communicate with said entity. For this reason, a beacon is said to cover the area within which it is able to communicate with other entities. The fact that the coverage is decided by the RSS-mapped offset distance, an *unknown* function of inter-nodal distance, contributes to the novelty of the approach presented in this thesis, as the coverage itself is an unknown quantity.

Accurate position estimation can be a difficult task in GNSS denied environments. Incrementally deploying nodes allows the deploying node to use positional data received from already deployed nodes when estimating its position, yielding more accurate position estimates. Although position estimation in GNSS denied environments is not addressed in this thesis, the incremental nature of the deployment scheme presented here

will allow deploying nodes to benefit from positional data from nodes already residing within the environment.

Summarizing, we formulate briefly the overarching objective of this thesis:

Main objective *Construct and analyse a high-level control law that deploys a set of mobile nodes with an unknown range of communication and four range sensors incrementally into an unknown environment, subject to the constraint that the set of deployed nodes should be dispersed within the environment so as to cover the largest area possible. The deployed network should, moreover, at all times form a (1-)connected network, so that information from any node can be acquired at a base station through (multi-hop) communication.*

Chapter 5

Method

In this section, the iterative manner in which a potential field used for incremental self-deployment within unknown two-dimensional environments is presented. An attractive potential field approach is adapted in order to generate virtual inter-nodal (beacon-to-agent) forces. Applying an attractive potential field for inter-nodal forces is motivated by the fact that it can be synthesized in terms of only relative positions, a quantity available to all agents through single-hop communication with neighbouring beacons (see Section 3.4). A repulsive potential field depending only on range measurements from the four range sensors of an agent, used for obstacle avoidance, is also presented.

Seen as the incremental deployment of mobile agents with the aim of area coverage of unknown environments, where the coverage itself is an unknown quantity (see (3.6)), has not yet been addressed in litterateur, we take a step back and initially address the problem in its simplest form, i.e. the 1-dimensional case: *Construct a high-level control law for incrementally deploying agents along an obstacle-free line, ensuring (1-)connectivity of the resulting network of beacons, and that the deploying agent explores a previously unexplored portion of the line*. In the 1-dimensional case, it is possible to define exploration in terms of quantities available to the deploying agent. Hence, in Section 5.2, a neighbour-induced potential field for sequential deployment along a line, guaranteeing exploration, is presented.

Then, a potential field concerned with fulfilling the objective defined in Chapter 4 is presented. In unknown 2D space, defining the essence of exploration using only information available to the deployed beacons/agents has proved to be a difficult task. As agents are only equipped with four range sensors, area measurements are not readily available, causing a definition of exploration in terms of sensor area coverage to be inconceivable. Furthermore, as the coverage supplied by a beacon depends on the *unknown* RSS-mapped offset distance and is thus itself an unknown quantity, no definition of exploration in terms of communication coverage is available at this stage. Defining exploration in terms of

quantities available to the agents has been attempted but has not proven fruitful. For this reason, we present in Section 5.3 the potential field used for exploring unknown 2D environments. However, we do not state a definition nor proof that the deploying agent explores previously unexplored areas.

The proposed field used for self-deployment in unknown 2D-environments depends on information received from neighbouring beacons and range measurements from four range sensors. The potential field produces a force that guides the deploying agent to possibly un-explored areas and away from obstacles. Furthermore, the relative angles of neighbouring beacons and local knowledge of the environment are used to heuristically decide the most favourable direction in which succeeding agents should be guided.

5.1 General assumptions

Throughout this section, it is assumed that a network of beacons, $\mathscr{B}$, has already been deployed. It is assumed that at least one beacon has been deployed such that:

$$\{0\} \subseteq \mathscr{B}$$

where beacon 0 is called the base station and is located at $\mathbf{x}_0 = \mathbf{0}$.

Furthermore, it is assumed that the indices of beacons correspond to the order in which they were deployed, such that for two indices $i, j \in \mathscr{B}$, the beacon with the largest index was the last to be deployed.

A force, $\mathbf{F}_i$, generated by a potential field U_i is assumed to affect an agent, i, according to the single integrator dynamics described in Section 3.4.1. When a deploying agent, i, arrives at its equilibrium, $\mathbf{x}_i^{eq}$ ($\mathbf{F}_i = \mathbf{0} \iff \mathbf{x}_i = \mathbf{x}_i^{eq}$), it lands and is re-instantiated as a beacon. Its index, i, is added to the set of beacons, $\mathscr{B}$. It is assumed that all beacons except for the base, 0, reached their location by courtesy of the force produced by the potential field U_i.

It is assumed that the RSS-mapped offset distance, $\xi_{i,j}$, for an agent i relative to a beacon j can be acquired through direct communication at regular intervals, and is assumed to be constant between updates such that $\frac{\partial \xi_{i,j}}{\partial \mathbf{x}_i} = 0$ between updates.

5.2 Neighbour induced exploration in 1D environments

For ease of the reader, we restate here the problem addressed in this section:

Construct a high-level control law for incrementally deploying agents along an obstacle-free line, ensuring (1-)connectivity of the resulting network of beacons, and that the deploying agent explores a previously unexplored portion of the line.

Incrementally exploring a line is defined as filling the line with beacons positioned on the line progressively further away from the base station at $\mathbf{x}_0 = \mathbf{0}$. In the following section, the iterative process, in which a potential field that ensures that the deploying agent explores a previously unexplored portion of the line, and ensuring connectivity of the network, is presented.

A line, l, parametrized by the direction vector $\mathbf{v}$ of unit magnitude ($\|\mathbf{v}\| = 1$) is defined as:

$$l(\mathbf{v}) = \{\mathbf{x} \in \mathbb{R}^2 : \mathbf{x} = \gamma\mathbf{v}, \gamma \geq 0\}$$

Thus, a line with parameter $\mathbf{v}$ contains all points in Euclidean space that lies in the direction $\mathbf{v}$ from the origin.

5.2.1 Assumptions

It is assumed that the beacon set, $\mathscr{B}$, contains all indices between 0 and $i-1$ upon the deployment of agent i:

$$\mathscr{B} = \{0 \dots i-1\}$$

and that the deployed beacons all lie on the line defined by $\mathbf{v}$:

$$\mathbf{x}_j = \gamma_j \mathbf{v} \; \forall \; j \in \mathscr{B}$$

where $\mathbf{x}_j$ is the position of beacon j, and γ_j is the distance between beacon j and the origin.

Furthermore, it is assumed that the base station is located at the origin, i.e. $\mathbf{x}_0 = \mathbf{0} \iff \gamma_0 = 0$, and that beacons are positioned along l in order defined by their indices, meaning:

$$j > k \iff \gamma_j - \gamma_k > 0 \; \forall \; j,k \in \mathscr{B} \tag{5.1}$$

It is also assumed that the deployed beacons form a connected network in the sense that $|\mathscr{N}_b(j)| \geq 1 \; \forall \; j \in \mathscr{B}$.

5.2.2 Exploration

As previously stated, the objective of the potential field affecting the deploying agent is to ensure that it explores a line. Formally an agent i is said to explore the line l with parameter $\mathbf{v}$ iff:

$$\mathbf{x}_i^{eq} = \gamma_i \mathbf{v}, \gamma_i - \gamma_j > 0 \; \forall \; j \in \mathscr{B} \tag{5.2}$$

where $\mathbf{x}_i^{eq}$ denotes the equilibrium point imposed on agent i by the potential field affecting it.

Taking into account the assumption that beacons are placed progressively further away from the base station (5.1), (5.2) can be restated as:

$$\mathbf{x}_i^{eq} = \gamma_i \mathbf{v}, \gamma_i - \gamma_{i-1} > 0$$

5.2.3 Local exploration

Local exploration is defined as an agent moving beyond (further from the base station) than all its neighbours. For and agent i with neighbour set $\mathscr{N}_a(i)$, the agent is said to explore locally iff:

$$\mathbf{x}_i^{eq} = \gamma_i \mathbf{v}, \gamma_i - \gamma_j > 0 \; \forall \; j \in \mathscr{N}_a(i)$$

Defining m as the maximum beacon index in the neighbour set of agent i, i.e.:

$$m = \max_{j \in \mathscr{N}_a(i)} j$$

and taking into account the assumption that beacons have been deployed progressively further away from the base station (5.1), the condition for local exploration can be restated as:

$$\mathbf{x}_i^{eq} = \gamma_i \mathbf{v}, \gamma_i - \gamma_m > 0, \; m = \max_{j \in \mathscr{N}_a(i)} j$$

5.2.4 Synthesizing a potential field that ensures local exploration along a line

For an agent i with a single beacon as its neighbour, $\mathscr{N}_a(i) = \{j\}$, the potential field:

$$U_i = \frac{1}{2} \| \mathbf{x}_i - (\mathbf{x}_j + \xi_{i,j} \mathbf{v}) \|^2 \tag{5.3}$$

induces a force in the agent according to:

$$\mathbf{F}_i = -\big(\mathbf{x}_i - (\mathbf{x}_j + \xi_{i,j} \mathbf{v})\big)$$

where $\mathbf{F}_i = -\frac{\partial}{\partial \mathbf{x}_i} U_i$ by (2.1) and the assumption that RSS-mapped offset distance is constant between updates is used.

Computing the equilibrium point of agent i when under the influence of the potential field in (5.3) yields:

$$\mathbf{F}_i|_{\mathbf{x}_i = \mathbf{x}_i^{eq}} = \mathbf{0} \iff \mathbf{x}_i^{eq} = \mathbf{x}_j + \mathbf{v}\xi_{i,j} = (\gamma_j + \xi_{i,j})\mathbf{v} \tag{5.4}$$

where the assumption that $\mathbf{x}_j = \gamma_j \mathbf{v}$ is used. Thus, for an agent, i, with only one neighbouring beacon, j, the potential field in (5.3) ensures that the agent explores locally due to the fact that:

$$\gamma_i - \gamma_j = \gamma_j + \xi_{i,j} - \gamma_j > \xi_{\mathcal{N}} > 0$$

where $j = m$ is the maximum beacon index in the neighbour set of agent i, and $\xi_{\mathcal{N}} > 0$ is the neighbour threshold (3.7).

From (5.4) it can be seen that the equilibrium point imposed on agent i by its neighbour j lies at the location of beacon j, perturbed in the direction of $\mathbf{v}$ by the RSS-mapped offset distance between the two entities. Seen as the RSS-mapped offset varies positively with RSS value, which in turn varies inversely with de-facto distance, the equilibrium point imposed on agent i by its neighbour j moves further from beacon j as the inter-nodal distance decreases and vice-versa. Hence, agent i will settle some unknown distance away from beacon j where the trade-off between inter-nodal distance and RSS value is in equilibrium.

Directly extending the field in (5.3) to the more general case, in which agent i has an arbitrary number of neighbours, yields:

$$U_i = \frac{1}{2} \sum_{j \in \mathcal{N}_a(i)} \|\mathbf{x}_i - (\mathbf{x}_j + \xi_{i,j}\mathbf{v})\|^2 \tag{5.5}$$

Computing the force on agent i by applying (2.1) yields:

$$\mathbf{F}_i = -\frac{\partial}{\partial \mathbf{x}_i} U_i = -\sum_{j \in \mathcal{N}_a(i)} \mathbf{x}_i - (\mathbf{x}_j + \xi_{i,j}\mathbf{v})$$

The equilibrium point for agent i is thus:

$$\begin{aligned} \mathbf{F}_i|_{\mathbf{x}_i = \mathbf{x}_i^{eq}} = \mathbf{0} \iff \mathbf{x}_i^{eq} &= \frac{1}{|\mathcal{N}_a(i)|} \sum_{j \in \mathcal{N}_a(i)} (\mathbf{x}_j + \xi_{i,j}\mathbf{v}) \\ &= \frac{1}{|\mathcal{N}_a(i)|} \sum_{j \in \mathcal{N}_a(i)} (\gamma_j + \xi_{i,j})\mathbf{v} \end{aligned}$$

such that:

$$\gamma_i = \frac{1}{|\mathcal{N}_a(i)|} \sum_{j \in \mathcal{N}_a(i)} (\gamma_j + \xi_{i,j})$$

where $|\mathcal{N}_a(i)|$ is the number of neighbours of agent i.

The local exploration condition demands

$$\begin{aligned}\gamma_i - \gamma_m > 0 \iff & \sum_{j\in\mathcal{N}_a(i)} (\gamma_j + \xi_{i,j}) - |\mathcal{N}_a(i)|\gamma_m \\ = & \sum_{j\in\mathcal{N}_a(i)\setminus\{m\}} (\gamma_j - \gamma_m) + \sum_{j\in\mathcal{N}_a(i)} \xi_{i,j} > 0\end{aligned} \tag{5.6}$$

As it is assumed that the beacon of largest index, m, is also positioned furthest away from the base station, every term in the first sum in (5.6) is known to be negative. Since there is no known relation between the values of $\xi_{i,j}$ and the distance between agent i's neighbours, it is impossible to conclude that the local exploration condition holds for the potential field defined in (5.5).

In order to secure local exploration along the line l with parameter $\mathbf{v}$, a final change is made to the potential field in (5.5), resulting in the following potential field:

$$U_i = \frac{1}{2} \sum_{j\in\mathcal{N}_a(i)} \kappa_i \|\mathbf{x}_i - \alpha_i(\mathbf{x}_j + \xi_{i,j}\mathbf{v})\|^2, \quad \alpha_i, \kappa_i > 0 \tag{5.7}$$

Computing the equilibrium of agent i under (5.7) yields:

$$\begin{aligned}\mathbf{F}_i|_{\mathbf{x}_i=\mathbf{x}_i^{eq}} = \mathbf{0} \iff \mathbf{x}_i^{eq} &= \frac{1}{\sum_{j\in\mathcal{N}_a(i)} \kappa_j} \sum_{i\in\mathcal{N}_a(i)} \kappa_j\alpha_j(\mathbf{x}_j + \xi_{i,j}\mathbf{v}) \\ &= \frac{1}{\sum_{j\in\mathcal{N}_a(i)} \kappa_j} \sum_{i\in\mathcal{N}_a(i)} \kappa_j\alpha_j(\gamma_j + \xi_{i,j})\mathbf{v} \\ \implies \gamma_i &= \frac{1}{\sum_{j\in\mathcal{N}_a(i)} \kappa_j} \sum_{i\in\mathcal{N}_a(i)} \kappa_j\alpha_j(\gamma_j + \xi_{i,j})\end{aligned} \tag{5.8}$$

The equilibrium point for an agent i, defined in (5.8), can be viewed as the center of mass of a set of point masses where the particle of mass κ_j is located at $\mathbf{p}_j = \alpha_j(\gamma_j + \xi_{i,j})\mathbf{v}$. Thus, increasing α_j moves the point $\mathbf{p}_j$ further from the origin in the direction defined by $\mathbf{v}$.

Using the equilibrium point in (5.8), the local exploration condition demands:

$$\begin{aligned}\gamma_i - \gamma_m > 0 \iff & (\gamma_i - \gamma_m) \sum_{j\in\mathcal{N}_a(i)} \kappa_j = \sum_{i\in\mathcal{N}_a(i)} \kappa_j\alpha_j(\gamma_j + \xi_{i,j}) - \gamma_m \sum_{j\in\mathcal{N}_a(i)} \kappa_j \\ = & \sum_{i\in\mathcal{N}_a(i)\setminus\{m\}} \kappa_j\alpha_j\gamma_j + \sum_{i\in\mathcal{N}_a(i)} \kappa_j\alpha_j\xi_{i,j} + \gamma_m\Big(\kappa_m\alpha_m - \sum_{j\in\mathcal{N}_a(i)} \kappa_j\Big) > 0\end{aligned} \tag{5.9}$$

where the first term is non-negative by (5.1) and the fact that the gains α_i and κ_i are both positive. The second term is positive due to $\xi_{i,j} > \xi_{\mathcal{N}}\ \forall\ j \in \mathcal{N}_a(i)$ and $m \in \mathcal{N}_a(i) \implies$

$\mathcal{N}_a(i) \neq \emptyset$. For $m = 0$ we have:

$$(\gamma_i - \gamma_0) \sum_{j \in \mathcal{N}_a(i)} \kappa_j = \kappa_0 \alpha_0 \xi_{i,0} > 0 \; \forall \; \alpha_0, \kappa_0 > 0 \iff \gamma_i - \gamma_0 > 0$$

For $m > 0 \iff \gamma_m > 0$ we have:

$$(\gamma_i - \gamma_m) \sum_{j \in \mathcal{N}_a(i)} \kappa_j > \gamma_m \Big(\kappa_m \alpha_m - \sum_{j \in \mathcal{N}_a(i)} \kappa_j \Big)$$

where the inequality arises due to the sum of the first two terms in (5.9) being positive. Thus:

$$\kappa_m \alpha_m - \sum_{j \in \mathcal{N}_a(i)} \kappa_j \geq 0 \implies (\gamma_i - \gamma_m) \sum_{j \in \mathcal{N}_a(i)} \kappa_j > 0 \iff \gamma_i - \gamma_m > 0 \tag{5.10}$$

Seen as $\kappa_j > 0 \; \forall \; j \in \mathcal{B}$, $\mathcal{N}_a(i) \subseteq \mathcal{B} = \{0 \ldots i-1\}$ and $m = \max_{j \in \mathcal{N}_a(i)} \; j$ we have that:

$$\sum_{j \in \mathcal{N}_a(i)} \kappa_j \leq \sum_{j=0}^{m} \kappa_j = \kappa_m + \sum_{j=0}^{m-1} \kappa_j \tag{5.11}$$

Applying (5.11) to (5.10) yields a condition on gains α_m and κ_m that ensures that the deploying agent, i, explores locally:

$$\alpha_m, \kappa_m > 0, \; \kappa_m(\alpha_m - 1) - \sum_{j=0}^{m-1} \kappa_j \geq 0 \implies \gamma_i - \gamma_m > 0 \tag{5.12}$$

It is noteworthy that the gains α_m and κ_m can be static and only decided by the index of the neighbouring beacon, m. As agent m lands and re-instantiates as a beacon, it can calculate its gains α_m and κ_m satisfying (5.12), and thus always guarantee that any succeeding agent whose largest neighbour index is m will explore locally.

5.2.5 Proof of global exploration

We denote by $\mathbf{x}_{i-1}$ the position of the previously deployed beacon, $i-1$. By the assumption that beacons are positioned along the line in order by their index, beacon $i-1$ was positioned furthest from the base, i.e. $\gamma_{i-1} > \gamma_j \; \forall \; j \in \mathcal{B} \setminus \{i-1\}$, upon deploying agent i.

The path taken by the deploying agent, i, can be partitioned into two phases. In the first phase, the previously deployed beacon, $i-1$, is not a neighbour of the deploying agent, i. In order to simplify the reasoning, assume that beacon $i-1$ has not been deployed, i.e. $\mathcal{B} = \{0 \ldots i-2\}$. Due to the receiver invariance property (3.2), and the fact that the equilibrium point induced by the potential field (5.8) depends only on properties of an

agent's neighbours, deploying agent i with $\mathscr{B} = \{0 \dots i-2\}$ would place it's equilibrium at $\mathbf{x}_i^{eq} = \mathbf{x}_{i-1}$. Furthermore, the path taken by agent i would match exactly the path taken by beacon $i-1$.

In phase two, the deploying agent, i, has followed the exact path taken by beacon $i-1$ until it encounters beacon $i-1$, i.e. $i-1 \in \mathscr{N}_a(i)$. Since $i-1 = \max_{j \in \mathscr{B}} j$, we must have $i-1 = \max_{j \in \mathscr{N}_a(i)} j$. Due to the potential field (5.7) guaranteeing local exploration, it can be concluded that $\gamma_i > \gamma_{i-1}$. Thus the potential field in (5.7) guarantees global exploration.

After the deployment of agent i, the agent is re-instantiated as a beacon and added to the beacon set ($\mathscr{B} = \{0 \dots i\}$). As agent i reached its position by virtue of the potential field defined in (5.7), the location of agent i is now either at the equilibrium point defined in (5.8) or at some unknown location where the agents neighbour set became empty, causing the force affecting the agent to vanish. Such a situation might occur due to gains $\alpha_j, \kappa_j \; j \in \mathscr{N}_a(i)$ placing the center of mass, $\mathbf{x}_i^{eq}$, of the virtual set of particles to which the deploying agent is gravitated towards by the potential field at a location such that $\mathbf{x}_i = \mathbf{x}_i^{eq} \implies \xi_{i,j} \leq \xi_{\mathscr{N}} \; \forall \; j \in \mathscr{B} \iff \mathscr{N}_a(i) = \emptyset$.

In case agent i arrived at the equilibrium point defined in (5.8), (3.8) dictates that the network is still connected after the deployment of agent i due to the network being connected before deploying agent i and beacon i having at least one neighbour.

If agent i halted deployment due to its neighbour set being empty, by (3.9) it stopped instantaneously as its neighbour set emptied out. Thus, by (3.7) we must have:

$$\exists \; j \in \mathscr{B} = \{0 \dots i-1\} : \xi_{i,j} = \xi_{\mathscr{N}} \implies \mathscr{N}_b(i) \neq \emptyset$$

meaning beacon i has at least one neighbour. By the same argument as before, the network of beacons is still connected after deploying agent i.

5.3 Exploration in unknown 2D environments

For ease of the reader, we restate here the problem addressed in this section:

Construct a high-level control law that deploys a set of mobile nodes with an unknown range of communication and four range sensors incrementally into an unknown environment, subject to the constraint that the set of deployed nodes should be dispersed within the environment so as to cover the largest area possible. The deployed network should, moreover, at all times form a (1-)connected network, so that information from any node can be acquired at a base station through (multi-hop) communication.

Although exploration along a line can be guaranteed with the neighbour-induced potential field defined in (5.7), it is not suitable for exploration in unknown 2D environments. In unknown 2D environments, corners might need to be passed or avoided. With the proposed potential field in (5.7), the equilibrium point for any single agent lies on the line connecting all previously deployed beacons, clearly making it impossible for the deploying agent to avoid getting stuck in or pass a corner. Due to this, inspiration is taken from the proposed line-exploring potential field, but modifications are made to make it better suited for unknown 2D environments.

5.3.1 Neighbour-induced potential field

For an agent i, the potential induced in i by its neighbouring beacons is defined as:

$$U_{i,n} = \frac{1}{2} \sum_{j \in \mathcal{N}_a(i)} \kappa_{i,j} \|\mathbf{x}_i - (\mathbf{x}_j + \xi_{i,j}\mathbf{v}_j)\|^2 \tag{5.13}$$

where $\mathcal{N}_a(i)$ is the neighbour set of agent i, $\mathbf{x}_i$ is the position of agent i, $\mathbf{x}_j$ is the position of agent i's neighbouring beacon j, $\xi_{i,j}$ is the RSS-mapped offset distance between i and j, and $\mathbf{v}_j$ is the *exploration vector* of beacon j (see Section 5.3.4). The agent relative gains, $\kappa_{i,j}$, are defined as:

$$\kappa_{i,j} = \frac{\kappa_j}{\sum_{j' \in \mathcal{N}_a(i)} \kappa_{j'}} \tag{5.14}$$

where the beacon gains, $\kappa_j = f(j) > 0$, are static and determined only by the index of the neighbouring beacon, j.

The force on an agent, i, generated by the potential field created by its neighbours is found by applying (2.1) to (5.13):

$$\mathbf{F}_{i,n} = -\frac{\partial}{\partial \mathbf{x}_i} U_{i,n} = -\sum_{j \in \mathcal{N}_a(i)} \kappa_{i,j}(\mathbf{x}_i - (\mathbf{x}_j + \xi_{i,j}\mathbf{v}_j))$$

5.3.2 Selecting beacon gains

Beacon gains κ_j are chosen based on the notion that for two consecutive neighbour sets of an agent i, $\mathcal{N}_{a,1}(i)$ and $\mathcal{N}_{a,2}(i)$, containing consecutively deployed beacons such that $\mathcal{N}_{a,1}(i) = \{j, j+1\}, \mathcal{N}_{a,2}(i) = \{j+1, j+2\}$, the difference in contribution from beacon j versus $j+1$ should equal the difference in contribution from beacon $j+1$ versus $j+2$:

$$\Big[\kappa_{i,j+1} - \kappa_{i,j}\Big]\Big|_{\mathcal{N}_a(i)=\mathcal{N}_{a,1}(i)} = \Big[\kappa_{i,j+2} - \kappa_{i,j+1}\Big]\Big|_{\mathcal{N}_a(i)=\mathcal{N}_{a,2}(i)} \tag{5.15}$$

Expanding (5.15) yields:

$$\frac{\kappa_{j+1} - \kappa_j}{\kappa_{j+1} + \kappa_j} = \frac{\kappa_{j+2} - \kappa_{j+1}}{\kappa_{j+2} + \kappa_{j+1}} \iff \kappa_{j+1}^2 - \kappa_j \kappa_{j+2} = 0 \tag{5.16}$$

Choosing beacons gains according to:

$$\kappa_j = t^j, t > 0 \tag{5.17}$$

Yields a valid solution to (5.16):

$$\kappa_{j+1}^2 - \kappa_j \kappa_{j+2} = \left(t^{j+1}\right)^2 - t^j t^{(j+2)} = t^{2(j+1)} - t^{2j+2} = 0$$

where $t > 0$ is named the gain factor, and is common for all agents.

With the choice of beacon gains as in (5.17), an agent i has its equilibrium at:

$$\begin{aligned}
\mathbf{F}_{i,n}|_{\mathbf{x}_i=\mathbf{x}_i^{eq}} = \mathbf{0} &\iff -\sum_{j\in\mathcal{N}_a(i)} \kappa_{i,j}(\mathbf{x}_i - (\mathbf{x}_j + \xi_{i,j}\mathbf{v}_j)) = \mathbf{0} \\
&\iff \sum_{j\in\mathcal{N}_a(i)} \kappa_{i,j}\mathbf{x}_i = \sum_{j\in\mathcal{N}_a(i)} \kappa_{i,j}(\mathbf{x}_j + \xi_{i,j}\mathbf{v}_j) \\
&\iff \mathbf{x}_i^{eq} = \sum_{j\in\mathcal{N}_a(i)} \kappa_{i,j}(\mathbf{x}_j + \xi_{i,j}\mathbf{v}_j)
\end{aligned}$$

where it is used that agent relative gains are normalized so that $\sum_{j\in\mathcal{N}_a(i)} \kappa_{i,j} = 1$. Thus, the equilibrium point of agent i can be viewed as the center of mass of the system of virtual particles located at $\mathbf{p}_j = \mathbf{x}_j + \xi_{i,j}\mathbf{v}_j$ with masses $m_j = \kappa_{i,j}$.

Seen as the most recently deployed neighbour of agent i is the beacon of highest index, the gain factor, t, is chosen as a value greater than one:

$$t > 1$$

Thus, the centre of mass of the system of particles consisting of agent i's neighbours is moved in the direction of the virtual particle defined by its most recently deployed neighbour.

By (5.13) and (5.14), choosing beacon gains as in (5.17) with gain factor $t > 1$ yields the following neighbour-induced potential for an agent i:

$$U_{i,n} = \frac{1}{2}\Big(\sum_{j\in\mathcal{N}_a(i)} t^j\Big)^{-1} \sum_{j\in\mathcal{N}_a(i)} t^j \|\mathbf{x}_i - (\mathbf{x}_j + \xi_{i,j}\mathbf{v}_j)\|^2,\ t > 1 \tag{5.18}$$

5.3.3 Sensing the environment

In litterateur, agents are often assumed to be equipped with omnidirectional range sensors such that an agent is always able to determine the *shortest* distance and direction to surrounding obstacle [9, 11, 12, 32]. In this thesis, however, agents are assumed to be equipped with only four range-sensors capable of measuring the range to obstacles in orthogonal/anti-parallel directions. Due to this, an alternative scheme for detecting the range and proximity to obstacles is presented.

It is assumed that an agent, i, is equipped with 4 range sensors (Section 3.6), $rs_j,\ j = 1\ldots4$ mounted in the center of origin in the agent's HOST frame at equally spaced angles $\psi_{rs_j,i} = \frac{\pi}{2}j$. All range sensors are assumed to have the same maximum sensing range R_s. In order to attain the proximity and direction to obstacles surrounding an agent, the obstacle avoidance vector $\mathbf{v}_{i,o}$ is defined as:

$$\mathbf{v}_{i,o} = \sum_{j=1}^{4} r_j \mathbf{R}(\psi_{rs_j})\mathbf{e}_x = \sum_{j=1}^{4} \mathbf{r}_j \tag{5.19}$$

where r_j is the range measurement made by rs_j, $\psi_{rs_j} = \psi_i + \psi_{rs_j,i}$ is the orientation of range sensor rs_j relative to the inertial x-axis and ψ_i is the orientation of the agent on which the range sensors are mounted.

Computing the magnitude of the obstacle avoidance vector yields:

$$\begin{aligned}
\|\mathbf{v}_{i,o}\| &= \left\| \sum_{j=1}^{4} r_j \mathbf{R}(\psi_{rs_j})\mathbf{e}_x \right\| = \left\| \sum_{j=1}^{2} (r_j - r_{j+2})\mathbf{R}(\psi_{rs_j})\mathbf{e}_x \right\| \\
&= \left\| (r_1 - r_3)\mathbf{R}(\psi_{rs_1})\mathbf{e}_x + (r_2 - r_4)\mathbf{R}(\psi_{rs_2})\mathbf{e}_x \right\| \\
&= \left\| \begin{bmatrix} r_1 - r_3 \\ r_2 - r_4 \end{bmatrix} \right\| = \sqrt{(r_1 - r_3)^2 + (r_2 - r_4)^2}
\end{aligned}$$

where it is used that $\psi_{rs_{j+2}} = \pi + \psi_{rs_j},\ j = 1,2$ and $\mathbf{R}(\psi_{rs_1})\mathbf{e}_x \perp \mathbf{R}(\psi_{rs_2})\mathbf{e}_x$.

Figure 5.1 shows the resulting obstacle avoidance vector, $\mathbf{v}_o$, for an agent where a single range sensor senses an obstacle. As shown in the figure, the measured range equals the minimum distance between the agent and the obstacle if and only if the range sensor making the measurement has its axis perpendicular to the obstacle's surface.

The direction and magnitude of the obstacle avoidance vector are decided by the asymmetry in range measurements. We will refer to the magnitude of the obstacle avoidance vector as the level of asymmetry in the environment. When there is no asymmetry in range measurements ($r_i = r_{i+2},\ i = 1,2$), as depicted in Figure 5.2, the obstacle avoidance vector has a magnitude of zero. When there *is* asymmetry in range measurements ($\exists\, i \in \{1,2\} : r_i \neq r_{i+1}$), the obstacle avoidance vector has non-zero magnitude, and is

said to be pointing in the direction leading away from the sensed virtual wall. As depicted in Figure 5.3, when placed in a right corner, the obstacle avoidance vector points in the direction leading away from the virtual wall intersecting both sensed walls at an angle of 45 degrees.

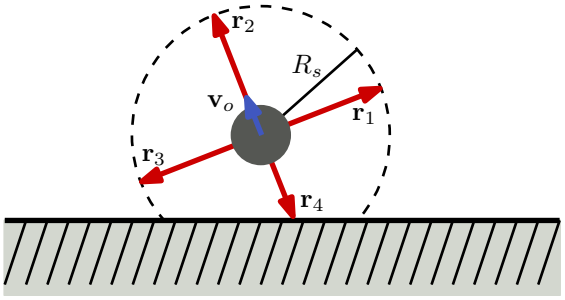


Figure 5.1: Range measurement vectors, $\mathbf{r}_j,\ j = 1 \ldots 4$ and resulting obstacle avoidance vector, $\mathbf{v}_o$, for an agent obstructed by a wall.

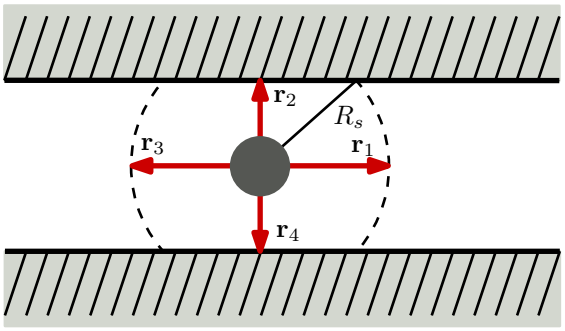


Figure 5.2: Range measurement vectors, $\mathbf{r}_j,\ j = 1 \ldots 4$ and resulting obstacle avoidance vector, $\mathbf{v}_o = \mathbf{0}$, for an agent obstructed by two walls equidistant from the agent.

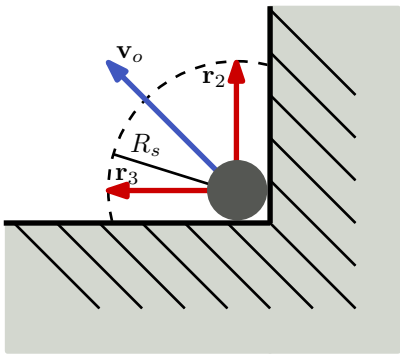


Figure 5.3: Range measurement vectors, $\mathbf{r}_j,\ j = 1 \ldots 4$ ($\mathbf{r}_1 = \mathbf{r}_4 = \mathbf{0}$) and resulting obstacle avoidance vector, $\mathbf{v}_o$, for an agent obstructed by two walls.

5.3.4 Selecting exploration vectors

As in the potential field used for exploring a line (5.7), the exploration vectors, $\mathbf{v}_j$, in (5.13) defines a direction in which the equilibrium point imposed by a single neighbour, j, is perturbed away from the location, $\mathbf{x}_j$, of that neighbour. The exploration vectors are static for all beacons and defined upon the event that an agent has deployed and re-instates as a beacon.

When an agent, i, has landed and has just been re-instantiated as a beacon, it computes its exploration vector, $\mathbf{v}_i$, based on the notion that agents that are yet to be deployed should be guided to the region of most open/unoccupied space. We denote by i the index of the beacon that has just landed and present in the following section how beacon i computes its exploration vector, $\mathbf{v}_i$.

It is assumed that beacon i has neighbours $\mathcal{N}_b(i)$ placed at $\mathbf{x}_j = \begin{bmatrix} x_j & y_j \end{bmatrix}^T,\ j \in \mathcal{N}_b(i)$, and an obstacle avoidance vector, $\mathbf{v}_{i,o}$, as defined in (5.19). Using the position of beacon i's neighbours relative to beacon i and the obstacle avoidance vector of beacon i, the unit circle is partitioned into a set of sectors. A threshold value, τ_o, is used to define a level of asymmetry that can be sensed before agent i should guide other agents in the direction leading away from the sensed (virtual) wall.

We define the angle of the line connecting a beacon i to a beacon j, measured at beacon i relative to the x_n-axis as:

$$\theta_{i,j} = \text{arctan2}(y_j - y_i, x_j - x_i)$$

where arctan2 is the arc-tangent function that takes into account the signs of its arguments and returns the angle as a value in the range $[0, 2\pi)$ in the correct quadrant and $\mathbf{x}_i = \begin{bmatrix} x_i & y_i \end{bmatrix}^T$ is the position of beacon i.

We define $\mathscr{A}_i$ as a set of angles:

$$\mathscr{A}_i = \begin{cases} \{\theta_{i,j},\ j \in \mathcal{N}_b(i)\} & ,\|\mathbf{v}_{i,o}\| < \tau_o \\ \{\theta_{i,j},\ j \in \mathcal{N}_b(i)\} \cup \{\angle\mathbf{v}_{i,o} - \frac{\pi}{2}, \angle\mathbf{v}_{i,o} + \frac{\pi}{2}\} & ,\|\mathbf{v}_{i,o}\| \geq \tau_o \end{cases} \tag{5.20}$$

where $\angle(\cdot)$ returns the angle of its argument relative to the inertial x_n-axis. Thus, $\mathscr{A}_i$ contains the direction from beacon i to all its neighbours. In addition, depending on the magnitude of the obstacle avoidance vector, two additional angles perpendicular to $\mathbf{v}_{i,o}$ might or might not be added to $\mathscr{A}_i$. The reason for this will be explained in time.

A sector, S_k, is defined as a 2-tuple according to:

$$S_k = \{\theta_{k,start}, \theta_{k,end}\}$$

where $\theta_{k,start} < \theta_{k,end}$ denotes the starting and ending angle of the sector respectively. We say that a sector S_k contains an angle, θ, and denote it by $\theta \in S_k$ iff:

$$\theta_{k,start} \leq \theta \leq \theta_{k,end}$$

Using the set of angles, $\mathscr{A}_i$, as defined in (5.20), the unit circle surrounding beacon i is partitioned into a set of sectors, denoted by $\mathscr{S}_i$. It is assumed that $\mathscr{A}_i$ is sorted so that $a_{j+1} > a_j \; \forall j = 1 \ldots |\mathscr{A}_i| - 1$ where a_j denotes the j'th element in $\mathscr{A}_i$. Furthermore it is assumed that $a_j \in [0, 2\pi) \; \forall \; j = 1 \ldots |\mathscr{A}_i|$, and by some abuse of notation we let $a_{|\mathscr{A}_i|+1} = a_1 + 2\pi$, where a_1 is the first, and thus the smallest element in $\mathscr{A}_i$. Using this, the unit circle surrounding beacon i is partitioned:

$$\mathscr{S}_i = \{S_j, j = 1 \ldots |\mathscr{A}_i|\}$$

where

$$S_j = \{a_j, a_{j+1}\}$$

denotes the j'th circle sector containing all angles between a_j and a_{j+1}.

A greedy heuristic is used for choosing the sector, S_i, to which beacon i should guide succeeding agents. In the case that the magnitude of $\mathbf{v}_{i,o}$ does not exceed the threshold τ_o, the sector of largest arc length is chosen. In the case that the magnitude of $\mathbf{v}_{i,o}$ does exceed the threshold τ_o, sectors laying in the same half-plane as the sensed (virtual) obstacle relative to beacon i are not chosen. Thus, the exploration sector of beacon i is defined as:

$$S_{i,exp} = \begin{cases} \arg\max_{S_k \in \mathscr{S}_i} L_k, & \|\mathbf{v}_{i,o}\| < \tau_o \\ \arg\max_{S_k \in \mathscr{S}_i} L_k \; s.t. \; \bar{\boldsymbol{\theta}}_k^T \mathbf{v}_{i,o} > 0, & \|\mathbf{v}_{i,o}\| \geq \tau_o \end{cases} \tag{5.21}$$

where $L_k = \theta_{k,end} - \theta_{k,start}$ and $\bar{\theta}_k = \frac{1}{2}(\theta_{k,end} + \theta_{k,start})$ denote the arc length and the orientation of the internal angle bisector [33] of the k'th circle sector respectively. $\bar{\boldsymbol{\theta}}_k = \begin{bmatrix} \cos(\bar{\theta}_k) & \sin(\bar{\theta}_k) \end{bmatrix}^T$ is the unit vector in the direction of the internal angle bisector of the k'th sector. The constraint applied when $\|\mathbf{v}_{i,o}\| \geq \tau_o$ ensures that the selected sector contains only directions leading away from the sensed obstacle seen as the direction of $\mathbf{v}_{i,o}$ is always perpendicular to the surface of a sensed (virtual) wall. Visual examples of how a beacon partitions the unit circle around it and chooses its exploration sector are provided in Figure 5.4 and Figure 5.5.

Assume now that a beacon i has only one neighbour, j, and denote by $\theta_{i,j}$ the direction of the vector from beacon i to j relative to the inertial x_n-axis. Assume also that $\|\mathbf{v}_{i,o}\| \geq \tau_o$ such that the constrained optimization is performed in (5.21) when finding beacon i's exploration sector. Now, if $\mathscr{A}_i = \{\theta_{i,j}\}$ is used such that $\mathscr{S}_i = \{S_1\} = \{\{\theta_{i,j}, \theta_{i,j} + 2\pi\}\}$, the only possible sector beacon i could possibly choose would have an internal bisector angle of $\bar{\theta}_1 = \frac{1}{2}(\theta_{i,j} + 2\pi + \theta_{i,j}) = \theta_{i,j} + \pi$. If beacon j is positioned so that $\bar{\boldsymbol{\theta}}_1^T \mathbf{v}_{i,o} \leq 0$, the constrained optimization problem in (5.21) has no feasible solution.

The problem of encountering a situation where (5.21) has no feasible solution is solved by including the two angles perpendicular to the obstacle avoidance vector in beacon i's set of sector constraining angles, $\mathscr{A}_i$. We define the two angles perpendicular to $\mathbf{v}_{i,o}$ as $\theta_{\pm} = \angle\mathbf{v}_{i,o} \pm \frac{\pi}{2}$. Without loss of generality assume that, post wrapping the angles to the range $[0, 2\pi)$, we have $\theta_- < \angle\mathbf{v}_{i,o} < \theta_+$.

In the trivial case, θ_- and θ_+ are consecutive elements in $\mathscr{A}_i$ such that $\exists\, S_k \in \mathscr{S}_i : S_k = \{\theta_-, \theta_+\}$ with internal angle bisector:

$$\bar{\theta}_k = \frac{1}{2}(\theta_+ + \theta_-) = \frac{1}{2}(\angle\mathbf{v}_{i,o} + \frac{\pi}{2} + \angle\mathbf{v}_{i,o} - \frac{\pi}{2}) = \angle\mathbf{v}_{i,o}$$

Thus, $\bar{\boldsymbol{\theta}}_k \parallel \mathbf{v}_{i,o} \implies \bar{\boldsymbol{\theta}}_k^T \mathbf{v}_{i,o} > 0$ and (5.21) has at least one feasible solution.

In the general case, there might exist angles, $\theta_a, \theta_b \in \mathscr{A}_i$ satisfying $\theta_- < \theta_a \leq \theta_b < \theta_+$ such that:

$$\{S_-, S_+\} = \{\{\theta_-, \theta_a\}, \{\theta_b, \theta_+\}\} \subseteq \mathscr{S}_i$$

Computing the orientations of the internal angle bisectors of the sectors S_- and S_+ yields:

$$\bar{\theta}_- = \frac{1}{2}(\theta_a + \theta_-) > \theta_-$$
$$\bar{\theta}_+ = \frac{1}{2}(\theta_+ + \theta_b) < \theta_+$$

Thus:

$$\angle\mathbf{v}_{i,o} - \bar{\theta}_- < \angle\mathbf{v}_{i,o} - (\angle\mathbf{v}_{i,o} - \frac{\pi}{2}) = \frac{\pi}{2} \implies \bar{\boldsymbol{\theta}}_-^T \mathbf{v}_{i,o} > 0$$
$$\bar{\theta}_+ - \angle\mathbf{v}_{i,o} < (\angle\mathbf{v}_{i,o} + \frac{\pi}{2}) - \angle\mathbf{v}_{i,o} = \frac{\pi}{2} \implies \bar{\boldsymbol{\theta}}_+^T \mathbf{v}_{i,o} > 0$$

meaning (5.21) has at least two feasible solutions.

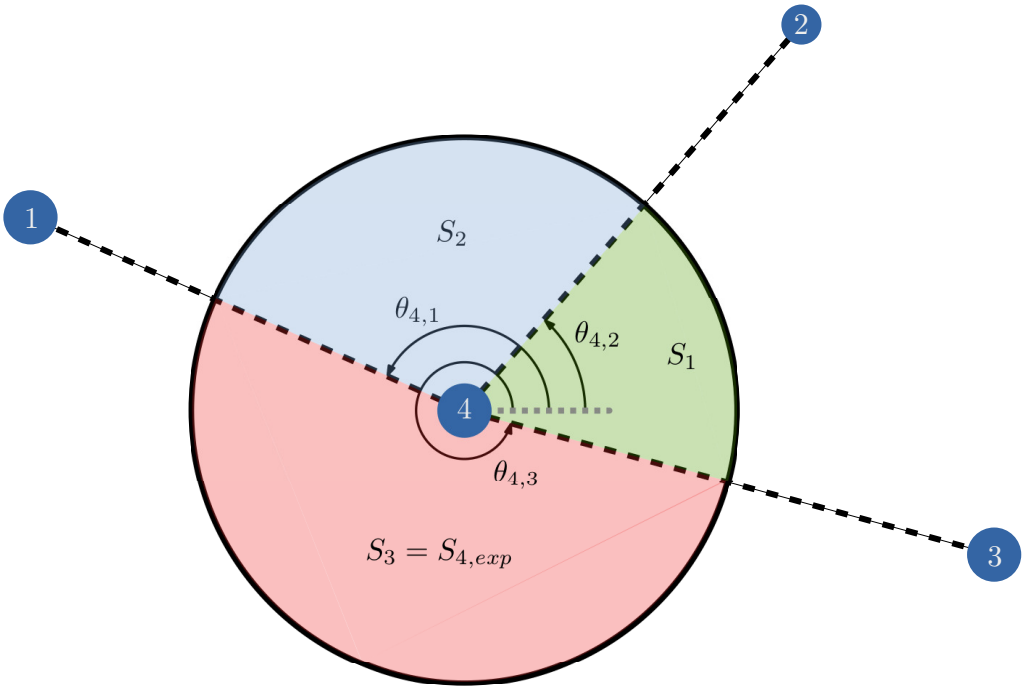


Figure 5.4: Beacon 4 with $\mathcal{N}_b(4) = \{1,2,3\}$ senses no asymmetry in the environment ($\|\mathbf{v}_{4,o}\| = 0$) and chooses its exploration sector, $S_{4,exp}$, as the sector with largest arc length, S_3.

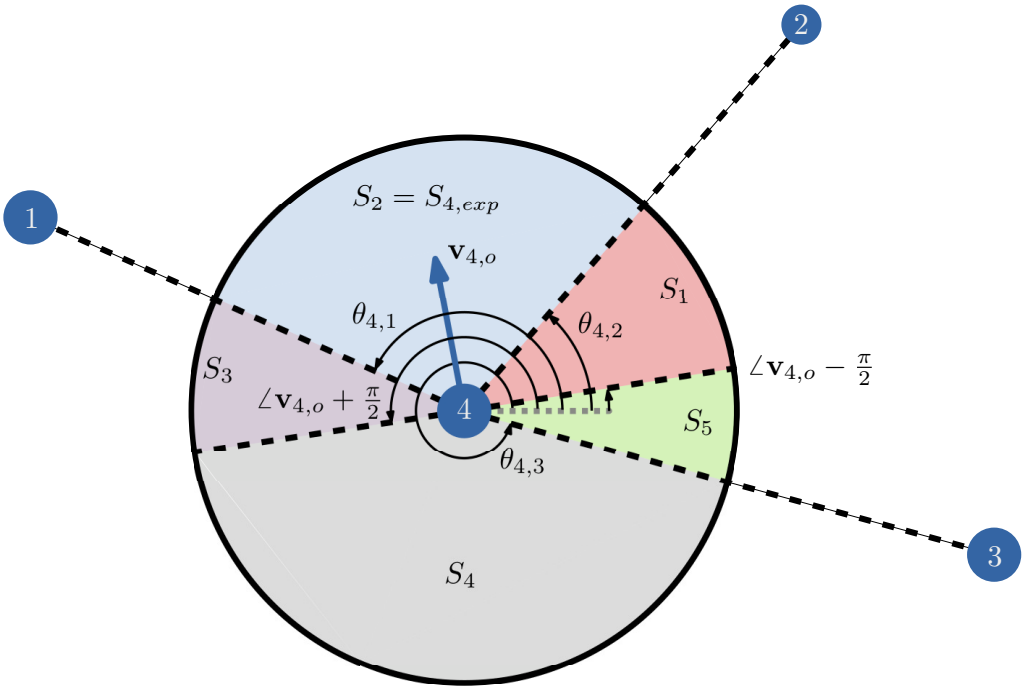


Figure 5.5: Beacon 4 with $\mathcal{N}_b(4) = \{1,2,3\}$ senses sufficient asymmetry in the environment ($\|\mathbf{v}_{4,o}\| > \tau_o$). Sector S_4 and S_5 do not fulfill the constraint in (5.21) and are disregarded. Beacon 4 chooses its exploration sector, $S_{4,exp}$, as the sector within the half-plane define by the obstacle avoidance vector with largest arc length, S_2.

Algorithm 1 implements the scheme for selecting beacon i's exploration sector where the 'wrapAngle'-routine takes as argument an angle and returns a corresponding angle in the interval $[0, 2\pi)$. It is noteworthy that all sectors pass the 'if'-check on line 15 of Algorithm 1, and are accepted as valid sectors, if the magnitude of the obstacle avoidance vector is sufficiently small. If the magnitude of the obstacle avoidance vector exceeds the threshold τ_o, the set of angles is updated on line 8 and 9, and only sectors fulfilling the contraint in (5.21) are accepted as valid sectors.

Algorithm 1 Computing the exploration sector for a beacon, i

1: **procedure** GETEXPLORATIONSECTOR(
   $\mathbf{v}_{i,o}$: obstacle avoidance vector for beacon i,
   $\mathcal{N}_b(i)$: neighbour set of beacon i)
2: $angles \leftarrow []$, initiate angles array as empty list
3: $valid\_sectors \leftarrow []$, initiate array of valid sectors as empty list
4: **for** $j \in \mathcal{N}_b(i)$ **do**
5: $angles \leftarrow \text{wrapAngle}(\text{atan2}(y_j - y_i, x_j - x_i))$
6: **end for**
7: **if** $\|\mathbf{v}_{i,o}\| \geq \tau_o$ **then**
8: $angles \leftarrow \text{wrapAngle}(\angle(\mathbf{v}_{i,o}) + \frac{\pi}{2})$
9: $angles \leftarrow \text{wrapAngle}(\angle(\mathbf{v}_{i,o}) - \frac{\pi}{2})$
10: **end if**
11: $sort(angles)$, sort angles in ascending order (in place)
12: $angles \leftarrow angles[1] + 2\pi$
13: **for** $k \leftarrow 1$ to $length(angles) - 1$ **do**
14: $\bar{\theta}_k \leftarrow 1/2(angles[k+1] + angles[k])$
15: **if** $\|\mathbf{v}_{i,o}\| < \tau_o$ or $\begin{bmatrix}\cos(\bar{\theta}_k) & \sin(\bar{\theta}_k)\end{bmatrix} \mathbf{v}_{i,o} > 0$ **then**
16: $L_k \leftarrow angles[k+1] - angles[k]$
17: $valid\_sectors \leftarrow$ sector with internal bisector angle $\bar{\theta}_k$ and arc length L_k
18: **end if**
19: **end for**
20: **return** $\arg\max_{S \in valid\_sectors} S.L$
21: **end procedure**

Given the exploration sector, $S_{i,exp}$ of beacon i, we denote by $\bar{\boldsymbol{\theta}}_{i,exp}$ and $L_{i,exp}$ the unit vector in the direction of the internal angle bisector and arc length of $S_{i,exp}$ respectively. The exploration vector of beacon i is chosen as:

$$\mathbf{v}_i = \mathbf{R}(r)\bar{\boldsymbol{\theta}}_{i,exp}, \quad r \sim \mathscr{U}(-L_{i,exp}/4, L_{i,exp}/4) \tag{5.22}$$

where $\mathbf{R}(\cdot) \in SO(2)$ is the two dimensional rotation matrix [28] and r is a uniformly distributed random variable. A continuous random variable has a uniform distribution on

the interval $[r_{min}, r_{max}]$ if its probability density function is given by [34]:

$$f(x) = \begin{cases} \frac{1}{r_{max}-r_{min}}, & r_{min} \leq x \leq r_{max} \\ 0, & \text{otherwise} \end{cases}$$

Applying the random rotation, r, within the exploration sector of beacon i when computing beacon i's exploration vector is motivated by the fact that applying a random rotation might prevent agents from deploying in a straight line and instead explore the plane, as illustrated by a simple example:

Assume that beacon 1 has deployed and landed with the base station (beacon 0) as its only neighbour such that $\mathbf{x}_1 = \mathbf{x}_0 + \xi_{1,0}\mathbf{v}_0$. By Algorithm 1, beacon 1 would compute its exploration sector as $S_{1,exp} = \{\angle(\mathbf{x}_0 - \mathbf{x}_1), \angle(\mathbf{x}_0 - \mathbf{x}_1) + 2\pi\}$. Ignoring the random component in (5.22), computing the exploration vector for beacon 1 yields:

$$\begin{aligned} \bar{\theta}_{1,exp} &= \frac{1}{2}(\angle(\mathbf{x}_0 - \mathbf{x}_1) + \angle(\mathbf{x}_0 - \mathbf{x}_1) + 2\pi) = \angle(\mathbf{x}_0 - \mathbf{x}_1) + \pi \\ \implies \mathbf{v}_1 = \bar{\boldsymbol{\theta}}_{1,exp} &= \begin{bmatrix} \cos(\angle(\mathbf{x}_0 - \mathbf{x}_1) + \pi) \\ \sin(\angle(\mathbf{x}_0 - \mathbf{x}_1) + \pi) \end{bmatrix} = -\frac{\mathbf{x}_0 - \mathbf{x}_1}{\|\mathbf{x}_0 - \mathbf{x}_1\|} \\ &= -\frac{\mathbf{x}_0 - (\mathbf{x}_0 + \xi_{1,0}\mathbf{v}_0)}{\|\mathbf{x}_0 - (\mathbf{x}_0 + \xi_{1,0}\mathbf{v}_0)\|} = \frac{\xi_{1,0}\mathbf{v}_0}{\|\xi_{1,0}\mathbf{v}_0\|} = \mathbf{v}_0 \end{aligned}$$

where it is used that $\xi_{1,0} > 0$.

Thus, beacon 1 would induce other agents to explore along the line between itself and beacon 0. If more agents were to be deployed, following the same scheme for choosing their exploration vectors, all agents would land on the line defined by $\mathbf{v}_0$. Performing multilateration/position estimation using beacons positioned on a line is not possible [35]. If the deployed beacons are to be used to support the navigation of the deploying agent by multilateration, they must not be positioned along a line. They should form a sparse configuration, and we enforce this by applying the random rotation, $\mathbf{R}(r)$, when computing the exploration vector of a beacon.

Choosing the random component in (5.22) as a uniformly distributed random variable in the range $\pm L_{i,exp}/4$ is motivated by the fact that it guarantees that agents affected by only one beacon, i, will have their equilibrium point, when under the influence of the potential field in (5.13), somewhere within the exploration sector of beacon i. For an agent j with beacon i as its only neighbour, the potential field defined in (5.13) yields an equilibrium point for agent j at:

$$\mathbf{x}_j^{eq} = \mathbf{x}_i + \xi_{j,i}\mathbf{v}_i = \mathbf{x}_i + \xi_{j,i}\mathbf{R}(r)\bar{\boldsymbol{\theta}}_{i,exp}$$

where $\mathbf{v}_i = \mathbf{R}(r)\bar{\boldsymbol{\theta}}_{i,exp}$ as per (5.22). This gives the angle of agent j, when at its equilib-

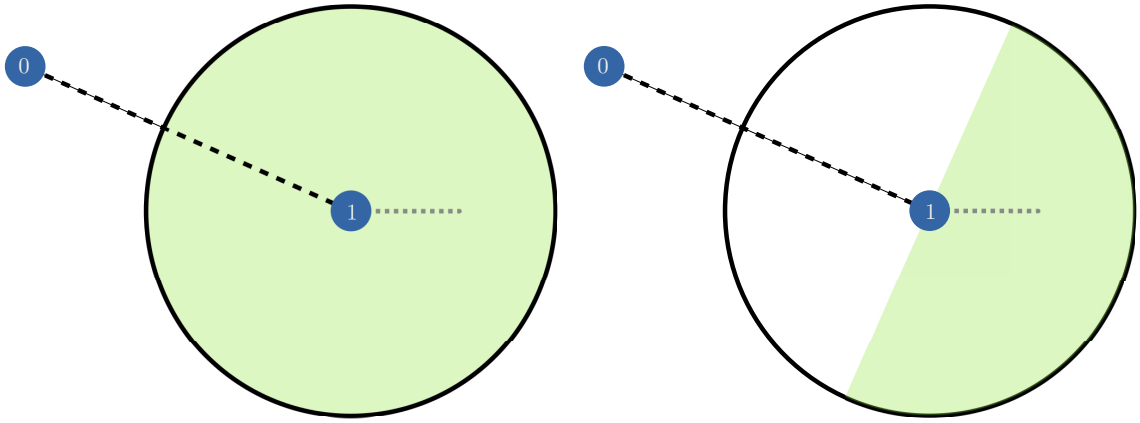


Figure 5.6: A beacon, 1, with one neighbour, 0. Beacon 1's exploration sector spans the entire unit circle. The green sub-sectors contain all directions in which beacon 1 might choose it's exploration vector when $r \sim \mathscr{U}(-L_{i,exp}/2.L_{i,exp}/2)$ (left) and $r \sim \mathscr{U}(-L_{i,exp}/4.L_{i,exp}/4)$ (right).

rium, relative to beacon i:

$$\begin{aligned} \angle(\mathbf{x}_j^{eq} - \mathbf{x}_i) &= \angle\left(\mathbf{x}_i + \xi_{j,i}\mathbf{R}(r)\bar{\boldsymbol{\theta}}_{i,exp} - \mathbf{x}_i\right) \\ &= \bar{\theta}_{i,exp} + r \in S_{i,exp} \; \forall \; |r| \leq L_{i,exp}/2 \end{aligned} \tag{5.23}$$

where $S_{i,exp}$ is the exploration sector of beacon i. Since the random component, r, is bounded by $|r| \leq L_{i,exp}/4 < L_{i,exp}/2$, agent j has its equilibrium within the exploration sector of beacon j.

By (5.23), it is clear that the range within which r is distributed could be chosen less conservatively and would still yield equilibrium points for agents within the exploration sector of the beacon. Increasing the bounds to $\pm L_{i,exp}/2$ would, however, allow for the possibility of guiding agents directly towards neighbouring beacons as illustrated in Figure 5.6. This is not satisfactory as it might cause agents to cluster. Selecting the bounds on r as $\pm L_{i,exp}/4$ is thus a trade-off between inducing agents to explore the plane while at the same time avoiding clustering.

5.3.5 Obstacle-induced potential field

In potential field approaches, the direction and the *shortest* distance to an obstacle are usually retrieved by omnidirectional range sensors and are used to compute a repelling force causing the agent to move away from the obstacles [12, 32]. As agents are assumed to be equipped with only four range sensors, there is no guarantee that the measured range to an obstacle equals the minimum distance between the agent and the obstacle. Hence, we present here a potential field-based obstacle repelling approach inspired by [32], but tailored to fit agents possessing only four range sensors.

Given the four range sensors of an agent, $rs_j,\ j = 1\ldots4$, returning corresponding measurements $r_j,\ j = 1\ldots4$, the range induced potential from the j'th range sensor is defined as:

$$U_{i,rs_j} = \kappa_o \left(\ln\left(\frac{R_s}{r_j}\right) - \frac{R_s - r_j}{R_s} \right) \tag{5.24}$$

where $\kappa_o \geq 0$ is a parameter deciding the strength of the potential field, R_s is the maximum sensing range for all range sensors and $r_j \in [0, R_s]$ is the measurement returned by the j'th range sensor of agent i.

The force induced in an agent, i, by its j'th range sensor is found by applying (2.1) to the obstacle induced potential field in (5.24):

$$\begin{aligned}
\mathbf{F}_{i,rs_j} &= -\frac{\partial U_{i,rs_j}}{\partial \mathbf{x}_i^n} = -\frac{\partial U_{i,rs_j}}{\partial r_j} \cdot \frac{\partial \|\mathbf{r}_j\|}{\partial \mathbf{x}_i^n} \\
&= -\frac{\partial U_{i,rs_j}}{\partial r_j} \cdot \frac{\partial \|\mathbf{r}_j\|}{\partial \mathbf{r}_j} \cdot \frac{\partial \mathbf{r}_j}{\partial \mathbf{x}_i^n} \\
&= -\kappa_o \left(\frac{1}{r_j} - \frac{1}{R_s} \right) \frac{\mathbf{r}_j}{r_j}
\end{aligned}$$

where $\|\mathbf{r}_j\| = r_j$ by (5.19) and it is used that $\mathbf{r}_j$ can be written as:

$$\mathbf{r}_j = \mathbf{o}_j^n - (\mathbf{x}_i^n + \mathbf{R}(\psi_i)\mathbf{x}_{rs_j}^i) \tag{5.25}$$

where $\mathbf{o}_j^n$ is the point on an obstacle sensed by range sensor rs_j, $\mathbf{x}_i^n$ and ψ_i are the position and orientation of agent i relative to the inertial EN frame respectively, and $\mathbf{x}_{rs_j}^i$ is the position of range sensor rs_j with respect to agent i's HOST frame. Differentiation of (5.25) yields:

$$\frac{\partial \mathbf{r}_j}{\partial \mathbf{x}_i^n} = -\mathbf{I}_{2\times2}$$

where $\mathbf{I}_{2\times2}$ is the identity matrix.

Combining the potential from all range sensors of an agent yields the total potential

induced in an agent due to the environment:

$$U_{i,o} = \kappa_o \sum_{j=1}^{4} \ln\left(\frac{R_s}{r_j}\right) - \frac{R_s - r_j}{R_s} \tag{5.26}$$

This yields the force exerted on an agent, i, by the environment as:

$$\mathbf{F}_{i,o} = -\kappa_o \sum_{i=1}^{4} \left(\frac{1}{r_j} - \frac{1}{R_s}\right) \frac{\mathbf{r}_j}{r_j} \tag{5.27}$$

Plots of the potential and force induced in an agent, i, by its range sensor rs_j are shown in Figure 5.7 and Figure 5.8 respectively. It is noteworthy that the force induced by a range sensor is directed in the opposite direction of the range measurement and that the force grows in magnitude as the measured range decreases.

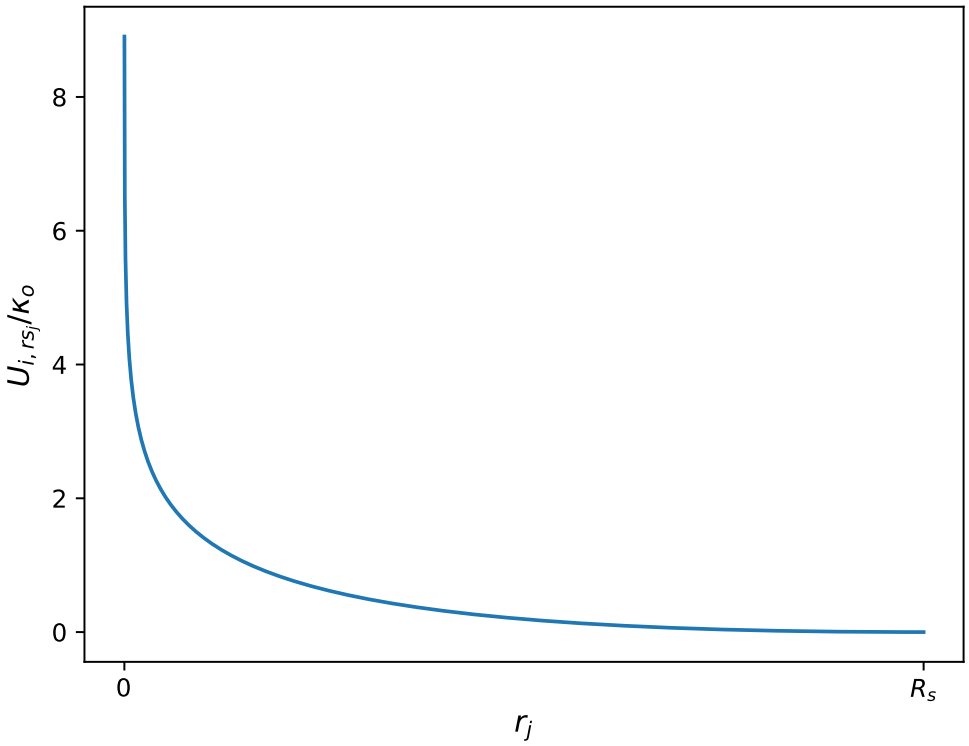


Figure 5.7: Potential U_{i,rs_k} induced by a range sensor, rs_j, in an agent i. The potential tends to infinity as the measured range tends to zero.

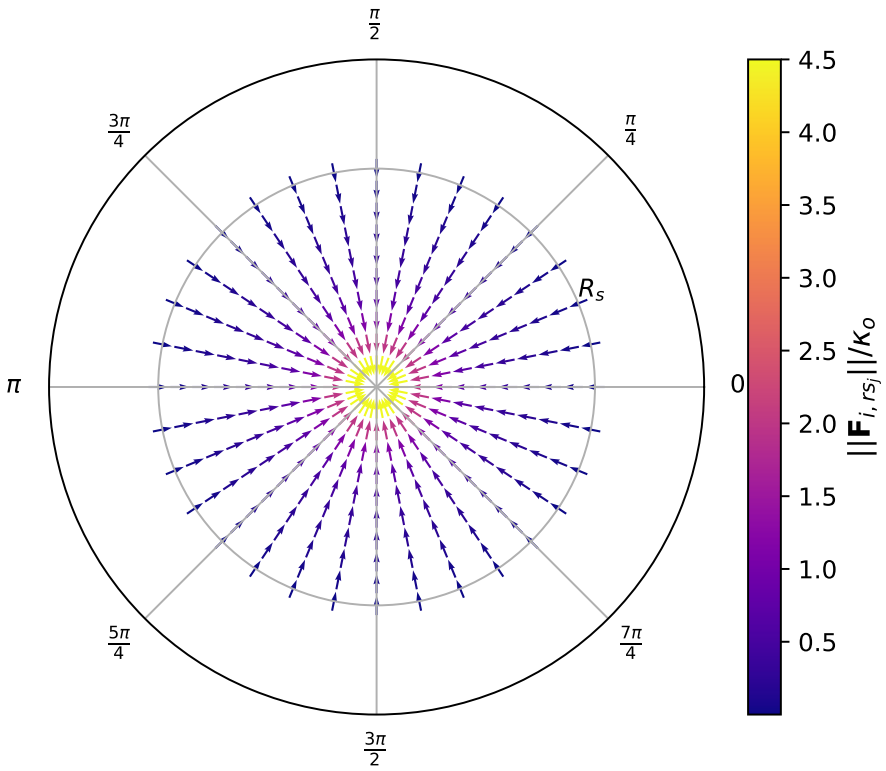


Figure 5.8: Magnitude and direction of the force $\mathbf{F}_{i,rs_k}$ induced by a range sensor, rs_j, on an agent i. As seen in the polar plot, the direction of the force is in the opposite direction of the sensor measurement ($\mathbf{r}_j$), and the magnitude of the force tends to infinity as the measured range tends to zero, and is zero when the measured range equals the maximum sensing range, R_s.

5.3.6 Potential field for exploration of unknown 2D environments

The neighbour- and obstacle induced potential fields presented in Section 5.3.1 and Section 5.3.5 are combined to give a net potential for an agent i:

$$U_i = U_{i,n} + U_{i,o}$$

where $U_{i,n}$ is the neighbour induced potential as defined in (5.13) and $U_{i,o}$ is the obstacle induced potential as defined in (5.26).

The combined potential field generates a reactive force in an agent i according to:

$$\mathbf{F}_i = \begin{cases} \mathbf{F}_{i,n} + \mathbf{F}_{i,o}, & \mathcal{N}_a(i) \neq \emptyset \\ \mathbf{0}, & \text{otherwise} \end{cases} \tag{5.28}$$

where $\mathbf{F}_{i,n}$ is the force on agent i due to its neighbours as defined in (5.18) and $\mathbf{F}_{i,o}$ is the force in agent i due to obstacles surrounding it as defined in (5.27).

The condition on the neighbour set of the agent being non-empty in (5.28) is added in order to ensure that obstacle-induced forces do not cause the agent to settle at a position rendering the network of beacons disconnected: assume that the network of beacons is connected before deploying agent i. If there is no net force on the agent and the neighbour set of the agent is non-empty, the agent will stop moving, re-instantiate as a beacon, and the connectivity of the network will be preserved by (3.8). If there is no net force on the agent due to its neighbour set being empty, the net force vanished the very instant the agent lost its last neighbour, and by (3.9) the agent stopped instantly. We denote by j the last neighbour agent i had before its neighbour set emptied out. The instant agent i's neighbour set became empty, by the definition of the neighbour set of an agent (3.7) we must have:

$$\xi_{i,j} = \xi_{\mathcal{N}}$$

where $\xi_{\mathcal{N}}$ is the neighbour threshold. Upon re-instantiating agent i as a beacon, by definition (3.5), the neighbour set of beacon i contains beacon j. Thus beacon i has at least one neighbour, and the network is still connected.

In practical applications, the locomotive platform of an agent sets a bound on the speed at which the agent can move. The virtual force, $\mathbf{F}_i$, induced in an agent might demand a velocity beyond the capabilities of the locomotive platform of the agent. For this reason, the net force on an agent i is saturated so that:

$$\mathbf{u}_i = \begin{cases} \mathbf{F}_i & , \|\mathbf{F}_i\| \leq v_{max} \\ \frac{v_{max}}{\|\mathbf{F}_i\|}\mathbf{F}_i & , \|\mathbf{F}_i\| > v_{max} \end{cases}$$

where $\mathbf{u}_i$ is the velocity of agent i under (3.9) and v_{max} is the maximum velocity at which the locomotive platform of the agent can operate.

Chapter 6

Simulating deployment in unknown 2D-environments

As previously stated, agents are incrementally deployed from the location of a base station into an unknown environment. The process of detecting that a new agent should be deployed is controlled by the base station upon receiving a message from the previously deployed agent. The deployment of a new agent is handled by the deploying agent entirely as only local information is used during self-deployment.

Algorithm 2 shows the logic used at the base station when deploying a pre-determined number of agents into an unknown environment. Algorithm 3 shows the logic used internally by an agent as it self-deploys into an unknown environment. It is noteworthy that an agent might land due to a lack of neighbours, as captured by the 'if'-check on line 9 in Algorithm 3. Forces produced by the obstacles surrounding an agent might cause it to lose contact with all beacons.

In Section 6.1, two metrics used to evaluate the performance of the deployments generated by the proposed approach are presented. In section Section 6.2 results obtained by deploying agents into two different environments are shown. The results are generated by a custom simulator implementing Algorithms 1 to 3 in the C++ programming language. The simulator is available at [36].

Algorithm 2 Deploying agents into an unknown environment

procedure DEPLOYAGENTS(
$\mathbf{v}_o$: exploration vector pointing to the interior of the unknown environment,
$\mathbf{x}_o$: position of the base station,
N: number of agents to deploy)
    for $i \leftarrow 1$ to N **do**
        Tell agent i to self-deploy using Algorithm 3
        while message that agent i has landed has not been received **do**
            wait
        end while
    end for
end procedure

Algorithm 3 Procedure used by an agent, i, to self deploy in an unknown environment

procedure AGENTSELFDEPLOY($\mathbf{x}_0$: position of base station)
    $\mathbf{x}_i \leftarrow \mathbf{x}_0$, self-deploy agent i starting from base station
    while True do
        broadcast signal and measure $\xi_{i,j}$ from all received signals
        $\mathcal{N}_a(i) \leftarrow$ compute neighbour set using (3.7)
        $\mathbf{F}_i \leftarrow$ compute total force using (5.28)
        if $\mathbf{F}_i = \mathbf{0}$ **then**
            break while, land due to zero net force
        else if $|\mathcal{N}_a(i)| = 0$ **then**
            break while, land due to lack of neighbours
        end if
        $\dot{\mathbf{x}}_i \leftarrow$saturate$(\mathbf{F}_i, v_{max})$, apply saturated force as control input
    end while
    $\dot{\mathbf{x}}_i = \mathbf{0}$, land
    re-instantiate agent i as beacon such that $|\mathcal{N}_b(i)| > 0$
    $\mathbf{v}_i \leftarrow$ compute exploration vector using Algorithm 1
    broadcast signal that agent i has landed
end procedure

6.1 Metrics

As the information available to the deployed beacons is minimal, a suitable metric must be applied to evaluate the performance of the deployed WSN. The only information stored locally at a beacon is its position and range measurements from its four range sensors. Furthermore, it is able to communicate with its neighbours and is hence able to retrieve their positions and range measurements. In this section, two performance metrics are presented. One of which is dependant only on data available locally to agents/beacons and hence may be applied in real-world situations. The other metric depends on information assumed unavailable to agents and is only fitting for evaluating performance in a simulation environment.

6.1.1 Uniformity

Heo and Varshney propose in [16] uniformity as a metric for evaluating the quality of the topology of a distributed sensor network. Uniformity is defined as the average local standard deviation of the distances between nodes:

$$U = \frac{1}{|\mathscr{B}|} \sum_{i \in \mathscr{B}} U_i$$

$$U_i = \sqrt{\frac{1}{|\mathscr{N}_b(i)|} \sum_{j \in \mathscr{N}_b(i)} (\|\mathbf{x}_i - \mathbf{x}_j\| - \bar{d}_i)^2}$$

$$\bar{d}_i = \frac{1}{|\mathscr{N}_b(i)|} \sum_{j \in \mathscr{N}_b(i)} \|\mathbf{x}_i - \mathbf{x}_j\|$$

where $|\cdot|$ takes as argument a set and returns the size of its argument, $\mathscr{B}$ is the set containing indices of all beacons in the network and $\mathscr{N}_b(i)$ is the neighbour set of beacon i. In the calculation of beacon i's local uniformity, U_i, only information available locally to beacon i is used. This makes uniformity a suitable metric in a distributed system.

In a network with a smaller value of uniformity, beacons are more uniformly distributed. In a uniformly distributed WSN, energy is spent evenly throughout the WSN. This causes the energy consumption per communication as well as the expected lifetime of each beacon to be "almost the same" [16], resulting in increased system lifetime.

6.1.2 Covered area within the ROI

Although the area of an unknown ROI is indeed unknown, and the signal strength at a certain distance is unpredictable in real-world applications, it is possible to determine the area covered by the deployed WSN in a simulation environment.

Given a neighbour threshold, $\xi_{\mathscr{N}}$, and the model of the RSS-mapped offset distance as defined in (3.3), the communication range of beacons can be decided:

$$\begin{aligned} \xi_{\mathscr{N}} &= \frac{\bar{\xi}}{2} + \frac{\bar{\xi}}{2}\cos\left(\pi\frac{R_c - d_{perf}}{d_{none} - d_{perf}}\right) \\ \iff R_c &= \frac{d_{none} - d_{perf}}{\pi}\cos^{-1}\left(\frac{2\xi_{\mathscr{N}} - \bar{\xi}}{\bar{\xi}}\right) + d_{perf} \end{aligned} \tag{6.2}$$

where R_c is the beacon communication range and it is assumed that $0 < \xi_{\mathscr{N}} < \bar{\xi}$.

Now, for a WSN comprised of beacons $i \in \mathscr{B}$, the set of points covered by the WSN can be computed as:

$$C_{\mathscr{B}} = \bigcup_{i \in \mathscr{B}} c_i$$

where c_i defined in (3.6) is the circle centered at beacon i's position $\mathbf{x}_i$ with radius R_c.

Given an environment, E, with $N_o > 1$ obstacles, an outer outer boundary ∂O_1 within which we wish to cover area and obstacles $O_2 \ldots O_{N_o}$, the ROI can be defined as:

$$ROI = O_1 \setminus \bigcup_{i=2}^{N_o} O_i$$

As in [16], coverage is defined as the ratio of the area covered by the WSN within the ROI and the entire area of the ROI:

$$C = \frac{\text{Area}(C_{\mathscr{B}} \cap ROI)}{\text{Area}(ROI)}$$

where the $\text{Area}(\cdot)$ function returns the area of a finite set of points.

6.2 Results

The proposed deployment scheme is tested in two different environments of different shape and size. The initial beacon (base station), 0, is always located at $\mathbf{x}_0 = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$ with an exploration vector, $\mathbf{v}_0$, oriented at 45° relative to the inertial x_n-axis. The properties of the environments in which the deployment scheme is tested are summarized in table Section 6.2. For simplicity, we assume an agent i is always aligned with the inertial x_n-axis so that $\psi_i = 0$. This is motivated by the ability of a quadcopter to fly in any direction irrespective of its horizontal rotation.

Name	Area	Description
Ten-by-ten	100 m^2	10-by-10 meter square
Stripa	616 m^2	10-by-64 meter hallway with two narrow passages

Table 6.1: Environment properties

Parameter values that remain unchanged through all simulations are summarized in Table 6.2.

Parameter	Value	Description
κ_o	0.1	Obstacle potential gain
τ_o	1 [m]	Exploration vector obstacle avoidance threshold
R_s	2 [m]	Maximum range sensor range
v_{max}	4 [m/s]	Maximum velocity
t	1.5	Gain factor
$\xi_{\mathcal{N}}$	0.5 [m]	RSS-mapped neighbour threshold distance

Table 6.2: Parameters used in all simulations

6.2.1 Ten-by-ten

Simulations are performed in the Ten-by-ten environment in order to attain results comparable to those in [16]. In [16], a communication range of $R_c = 4$ meters and a sensor range of $R_s = 2$ meters is used during simulation. Hence, parameters for the RSS-mapped offset distance function are chosen so that $\xi_{i,j} = \xi_{\mathcal{N}} \iff \|\mathbf{x}_i - \mathbf{x}_j\| = 4$, and the maximum range for all range sensors is set to $R_s = 2$.

Table 6.3 contains a summary of the parameters used in the RSS-mapped offset distance function in order to achieve a communication range of 4 meters. The parameters are varied in order to investigate the behaviour of the deployment under different conditions. One hundred deployments are performed for each combination of parameters in

order to investigate the effect of the random component used when choosing a beacon's exploration vector (5.22).

Combination name	$\bar{\xi}$	d_{perf}	d_{none}
Higher-max	20	2	~ 4.225
High-max	10	2	~ 4.335
Medium-max	5	2	~ 4.515
Low-max	1	2	6

Table 6.3: Parameters for the RSS-mapped offset distance function in (3.3).

Uniformity, coverage and sample distributions are shown in Figures 6.1 to 6.3, Figures 6.4 to 6.6, Figures 6.7 to 6.9 and Figures 6.10 to 6.12 when Higher-max, High-max, Medium-max and Low-max parameters are used in the RSS-mapped offset distance function.

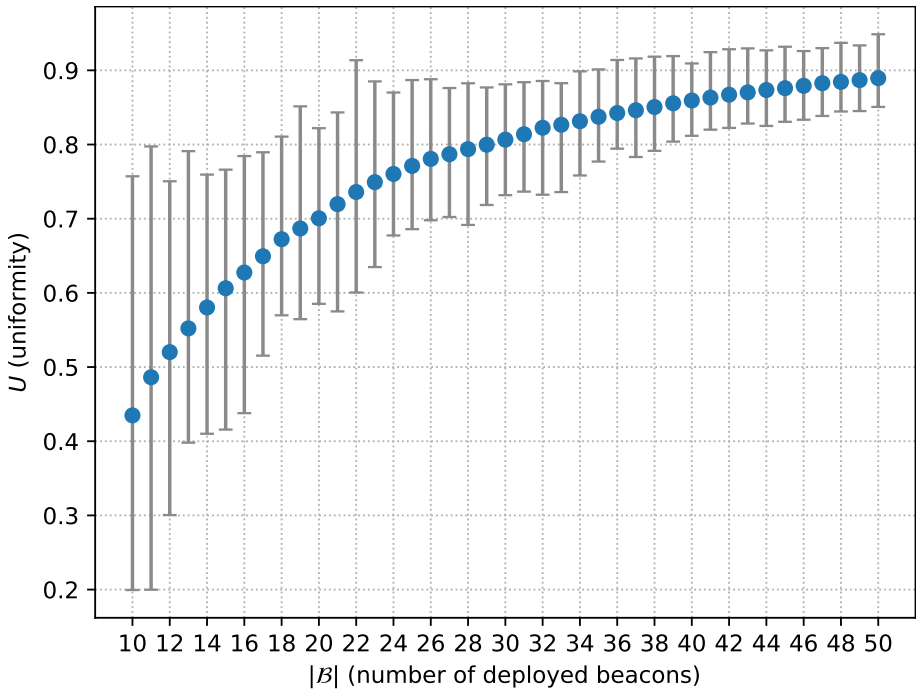


Figure 6.1: Average (blue dot), maximum (upper bar), and minimum (lower bar) uniformity versus the number of deployed beacons over 100 runs in the Ten-by-ten environment with Higher-max RSS-mapped offset distance function parameters.

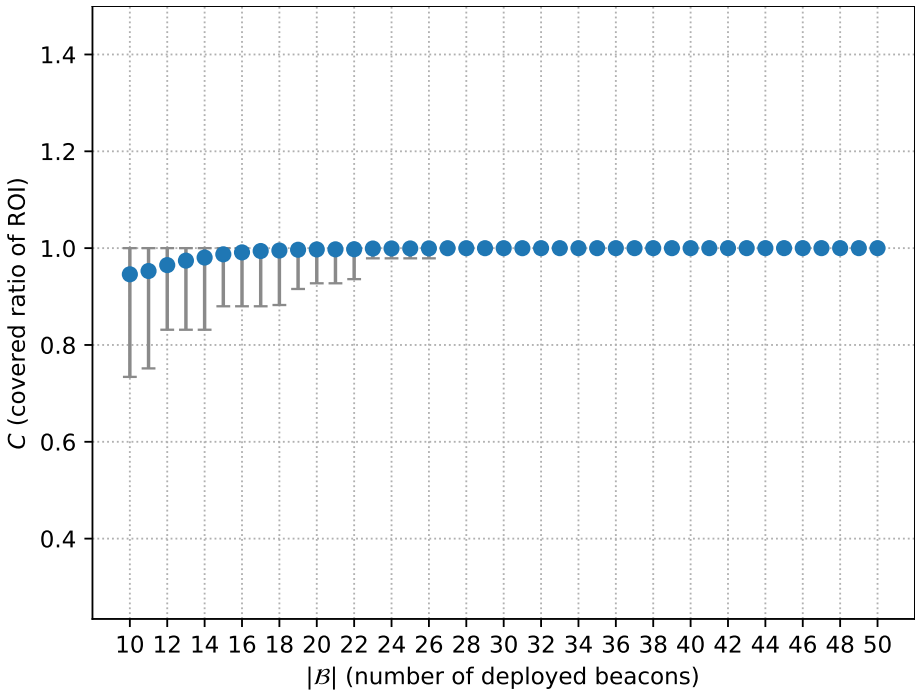


Figure 6.2: Average (blue dot), maximum (upper bar), and minimum (lower bar) coverage versus the number of deployed beacons over 100 runs in the Ten-by-ten environment with Higher-max RSS-mapped offset distance function parameters.

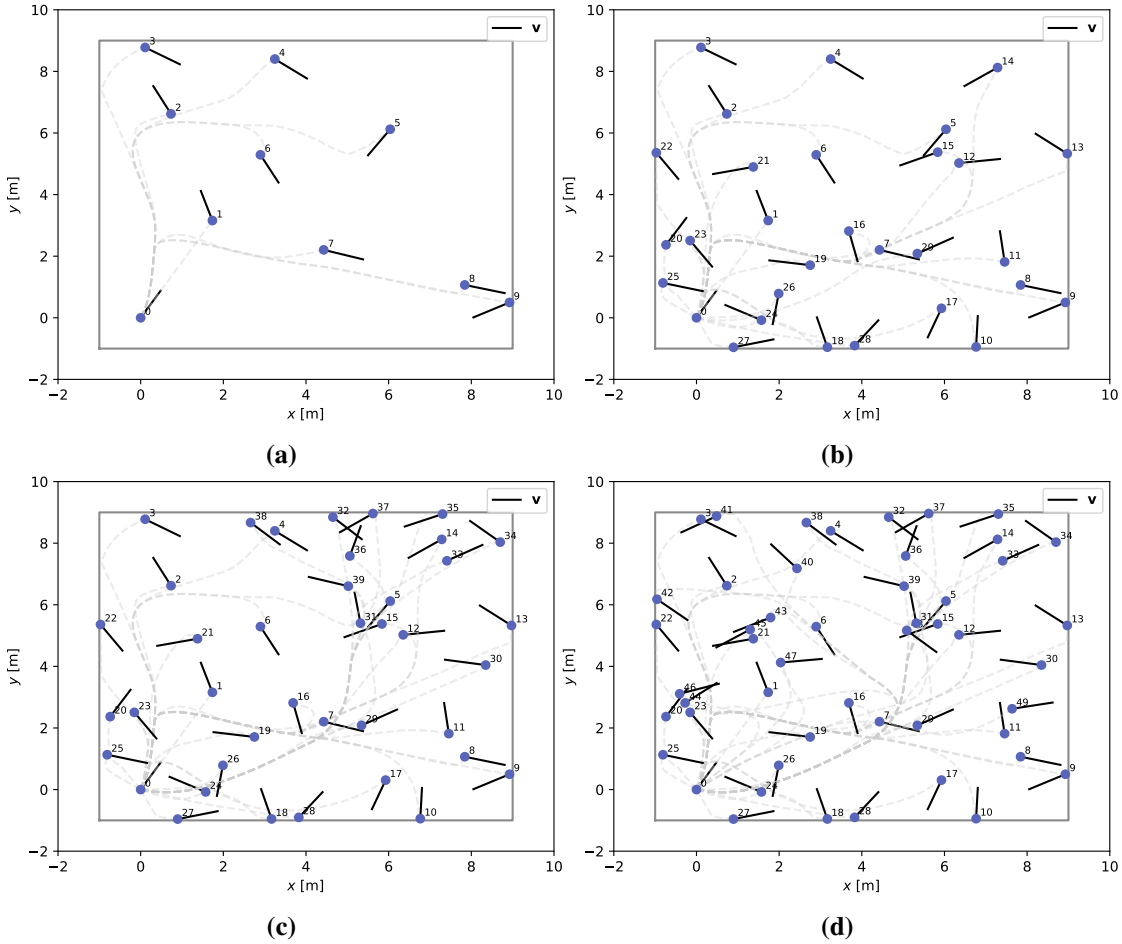


Figure 6.3: Sample configurations after (a) 10, (b) 30, (c) 40 and (d) 50 deployed beacons using Higher-max parameters in the RSS-mapped offset distance function. Exploration vectors are represented as black lines.

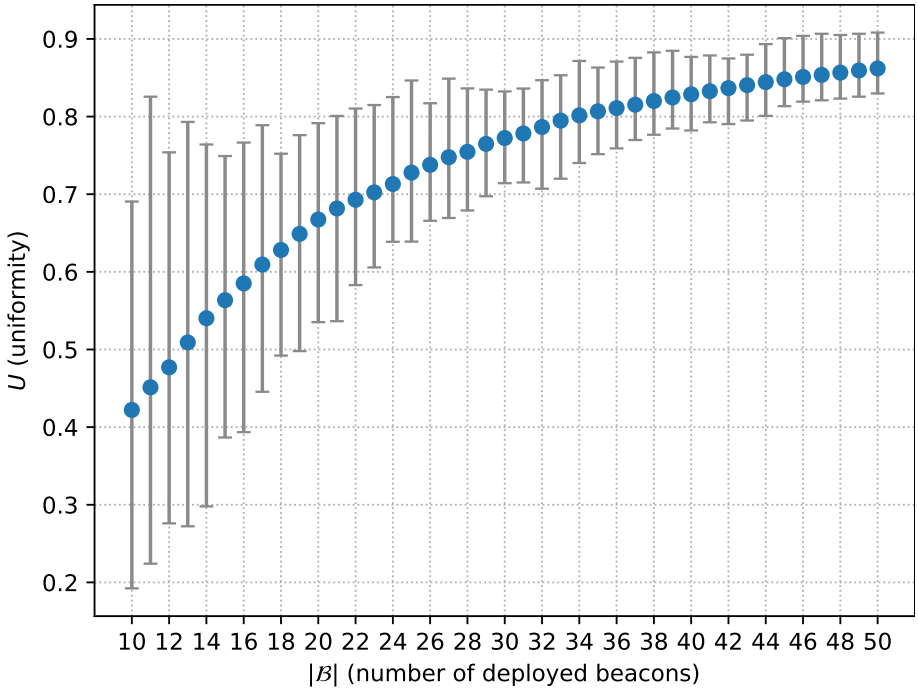


Figure 6.4: Average (blue dot), maximum (upper bar), and minimum (lower bar) uniformity versus the number of deployed beacons over 100 runs in the Ten-by-ten environment with High-max RSS-mapped offset distance function parameters..

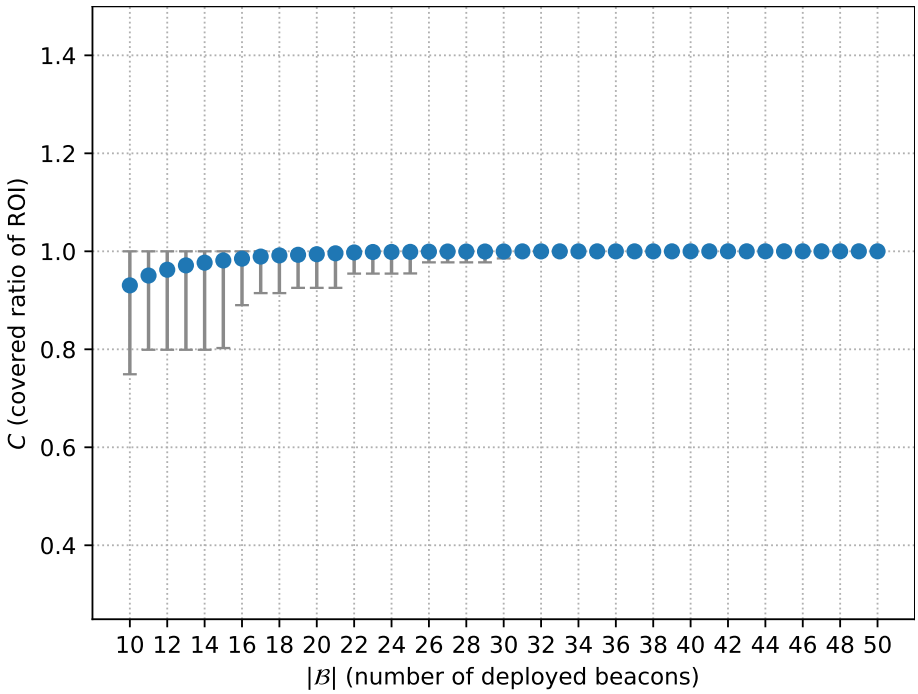


Figure 6.5: Average (blue dot), maximum (upper bar), and minimum (lower bar) coverage versus the number of deployed beacons over 100 runs in the Ten-by-ten environment with High-max RSS-mapped offset distance function parameters.

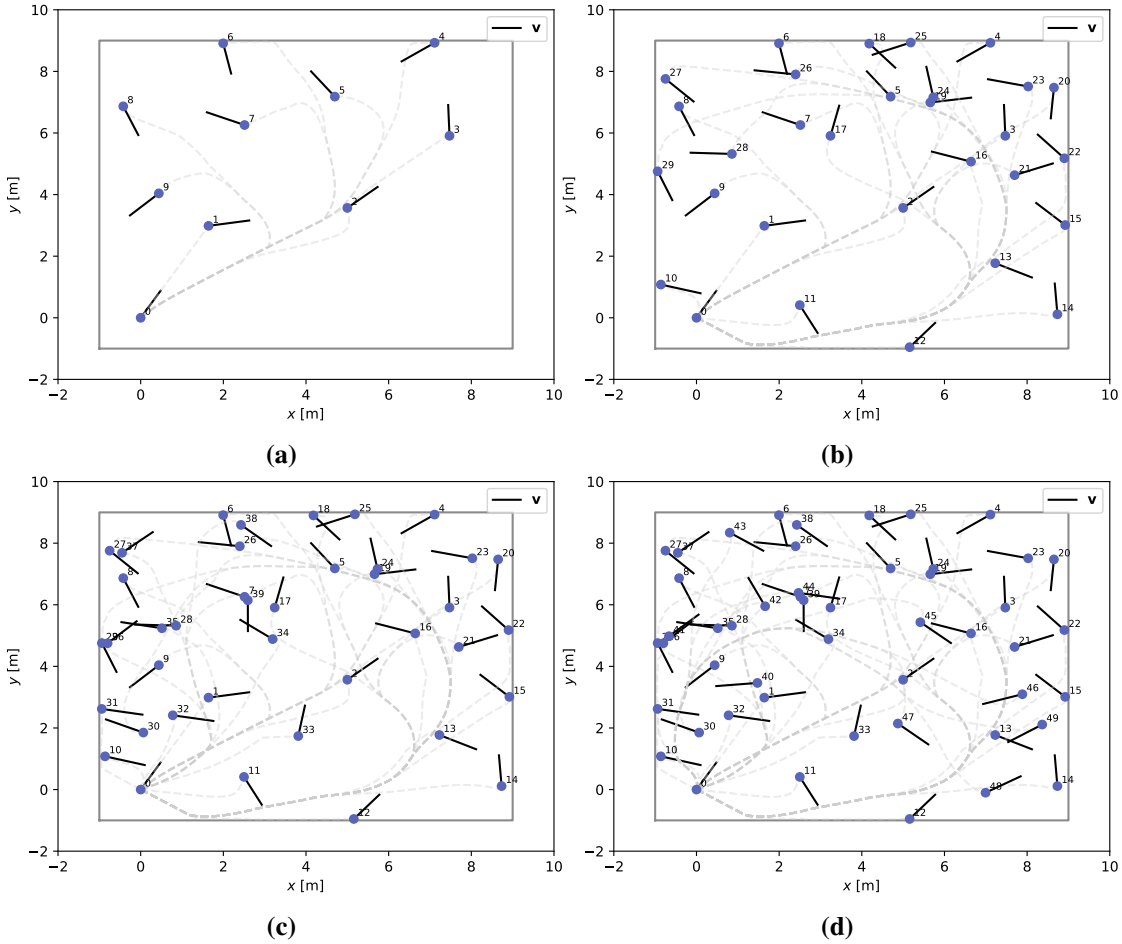


Figure 6.6: Sample configurations after (a) 10, (b) 30, (c) 40 and (d) 50 deployed beacons with High-max RSS-mapped offset distance function parameters. Exploration vectors are represented as black lines.

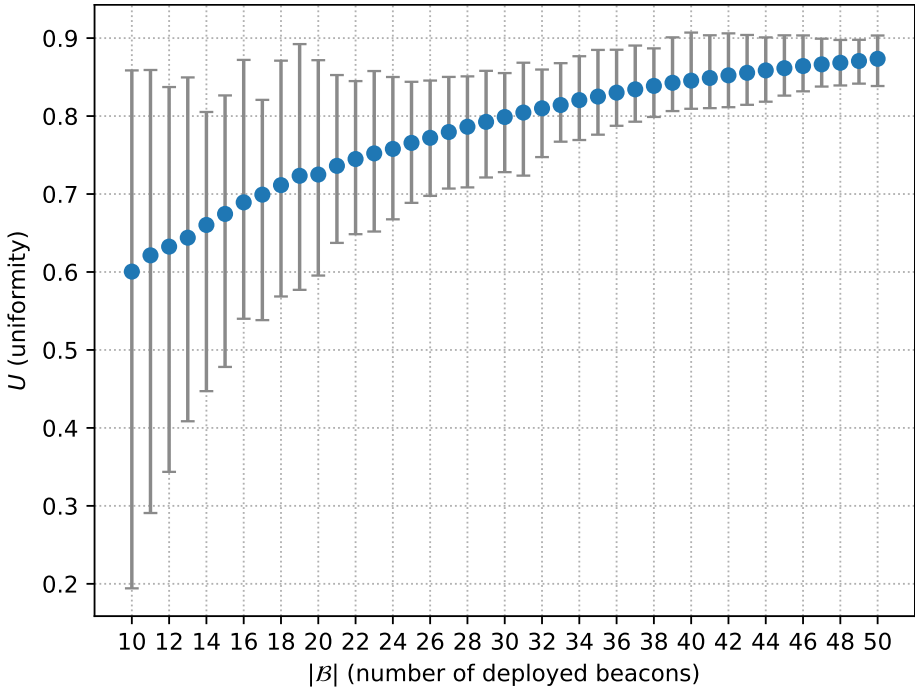


Figure 6.7: Average (blue dot), maximum (upper bar), and minimum (lower bar) uniformity versus the number of deployed beacons over 100 runs in the Ten-by-ten environment with Medium-max RSS-mapped offset distance function parameters.

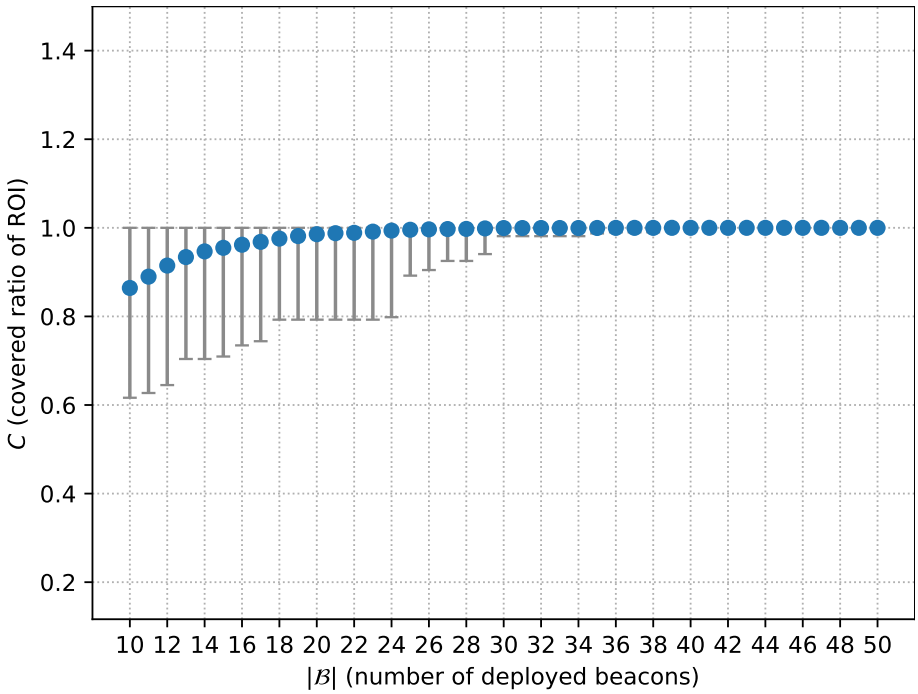


Figure 6.8: Average (blue dot), maximum (upper bar), and minimum (lower bar) coverage versus the number of deployed beacons over 100 runs in the Ten-by-ten environment with Medium-max RSS-mapped offset distance function parameters.

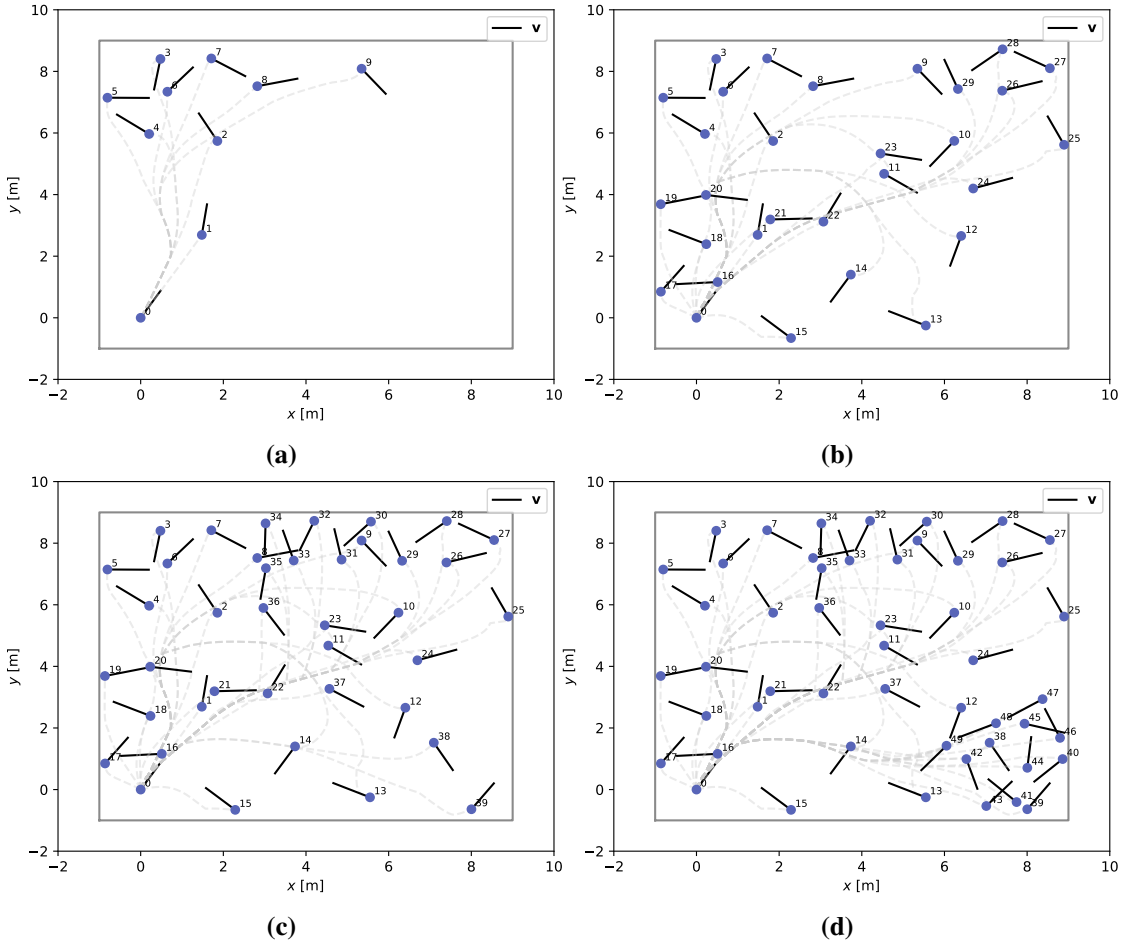


Figure 6.9: Sample configurations after (a) 10, (b) 30, (c) 40 and (d) 50 deployed beacons with Medium-max RSS-mapped offset distance function parameters. Exploration vectors are represented as black lines.

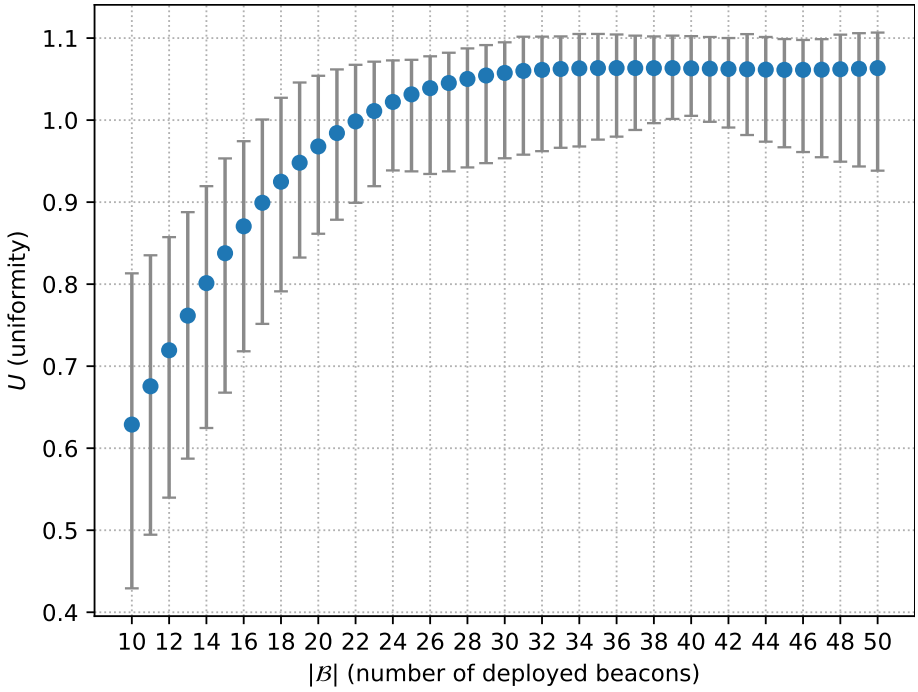


Figure 6.10: Average (blue dot), maximum (upper bar), and minimum (lower bar) uniformity versus the number of deployed beacons over 100 runs in the Ten-by-ten environment with Low-max RSS-mapped offset distance function parameters.

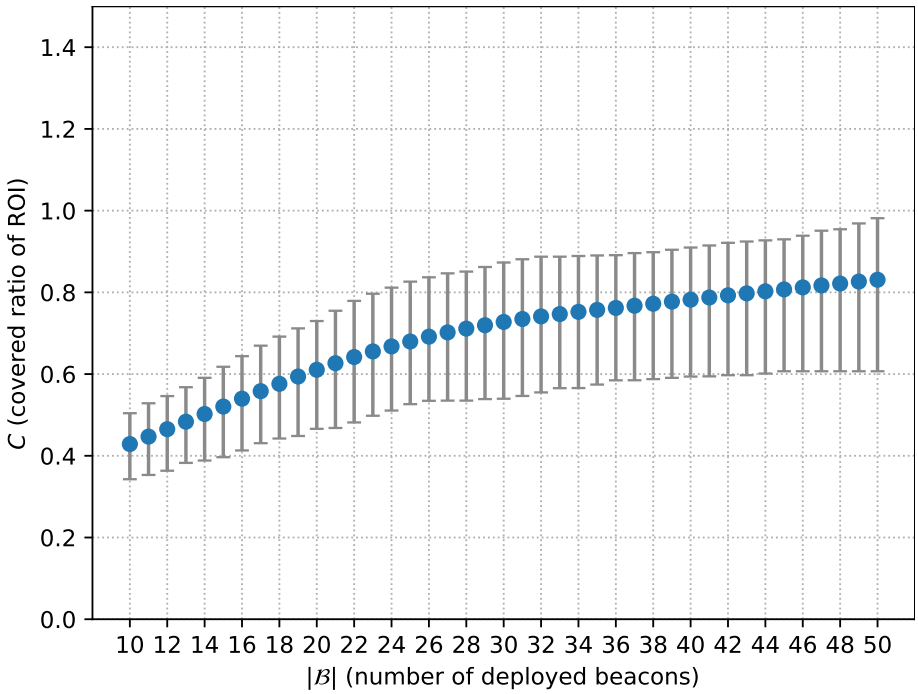


Figure 6.11: Average (blue dot), maximum (upper bar), and minimum (lower bar) coverage versus the number of deployed beacons over 100 runs in the Ten-by-ten environment with Low-max RSS-mapped offset distance function parameters.

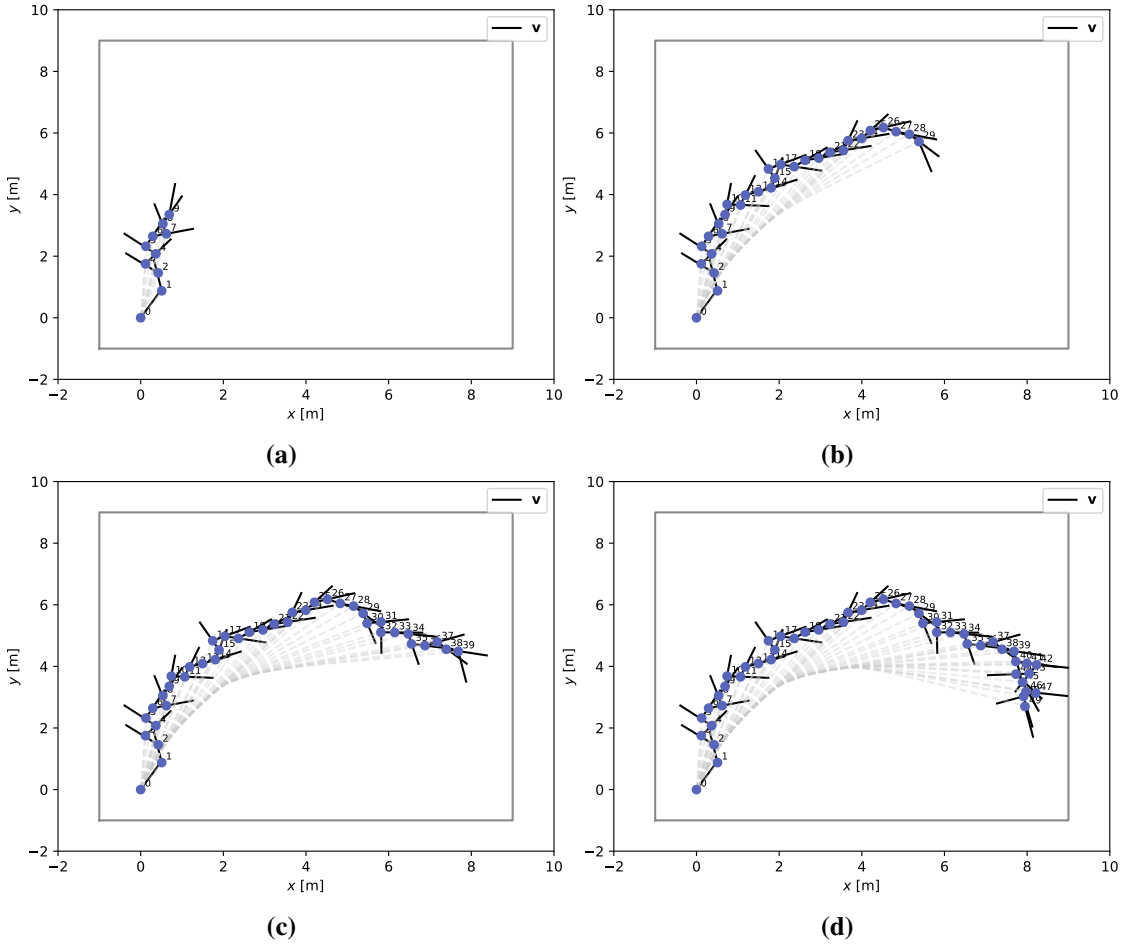


Figure 6.12: Sample configurations after (a) 10, (b) 30, (c) 40 and (d) 50 deployed beacons with Low-max RSS-mapped offset distance function parameters. Exploration vectors are represented as black lines.

6.2.2 Stripa

Simulations in the Stripa environment are performed in order to evaluate the performance of the proposed deployment scheme in a more complex environment. Stripa is chosen as it is a well-known part of the NTNU Gløshaugen campus and allows evaluation of the deployment scheme in a more realistic and non-convex environment. As seen in the layout of the environment in Figure 6.13, there are two narrow passages that must be passed in order to enter new sections of the environment.

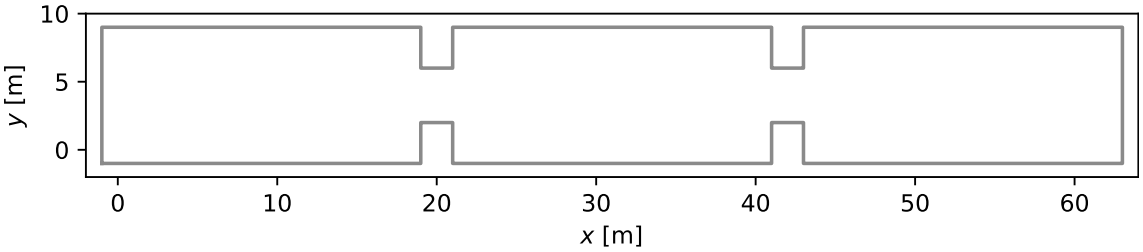


Figure 6.13: The Stripa environment

In the Stripa environment, only Higher-max parameters are used for the RSS-mapped offset distance function (3.3) as these were the values that yielded the fastest coverage (in terms of network size) in the Ten-by-ten environment. For ease of the reader the parameter values are summarized in Table 6.4.

Combination name	$\bar{\xi}$	d_{perf}	d_{none}
Higher-max	20	2	4.225

Table 6.4: Parameters for the RSS-mapped offset distance function in (3.3).

100 deployments of 100 agents are performed. Figure 6.14 and Figure 6.15 show the average, maximum and minimum uniformity and coverage versus number of deployed beacons over the 100 deployments, respectively. Figures 6.16 to 6.19 show the configuration, coverage- and uniformity trajectories of the worst- and best performing (in terms of coverage) deployments after 20 deployed beacons. Figures 6.20 to 6.23 show the configuration, coverage- and uniformity trajectories of the worst- and best performing (in terms of coverage) deployments after 50 deployed beacons. Figures 6.24 to 6.27 show the configuration, coverage- and uniformity trajectories of the worst- and best performing (in terms of coverage) deployments after 100 deployed beacons.

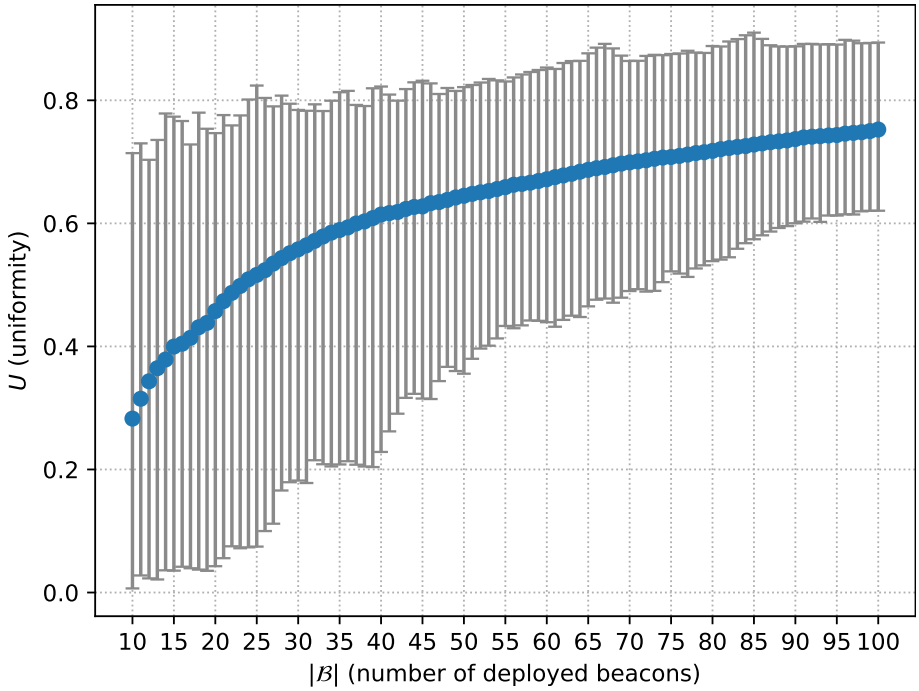


Figure 6.14: Average (blue dot), maximum (upper bar), and minimum (lower bar) uniformity versus the number of deployed beacons over 100 runs in the Stripa environment with Higher-max RSS-mapped offset distance function parameters.

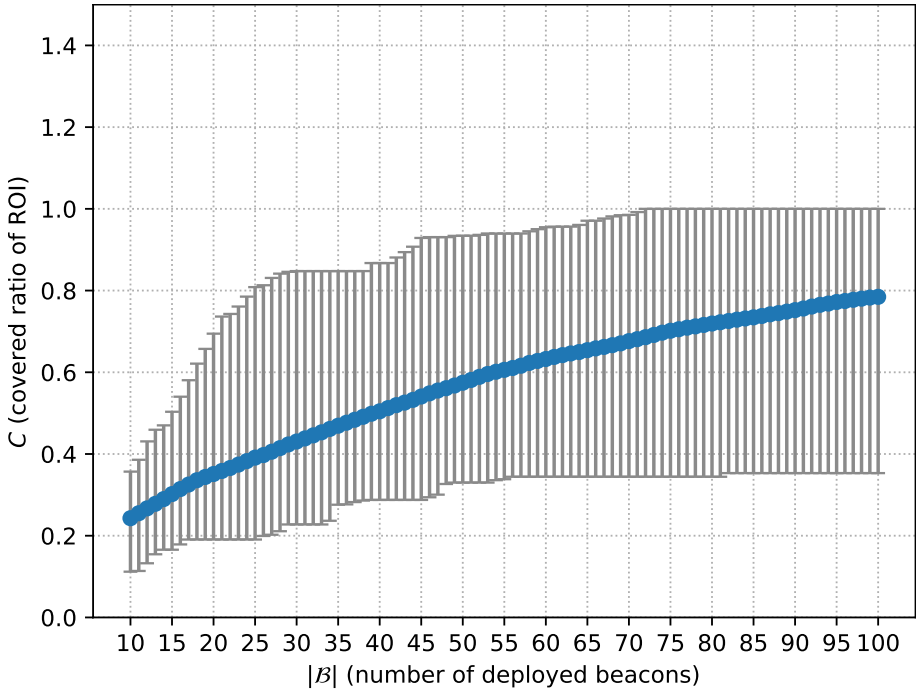


Figure 6.15: Average (blue dot), maximum (upper bar), and minimum (lower bar) coverage versus the number of deployed beacons over 100 runs in the Stripa environment with Higher-max RSS-mapped offset distance function parameters.

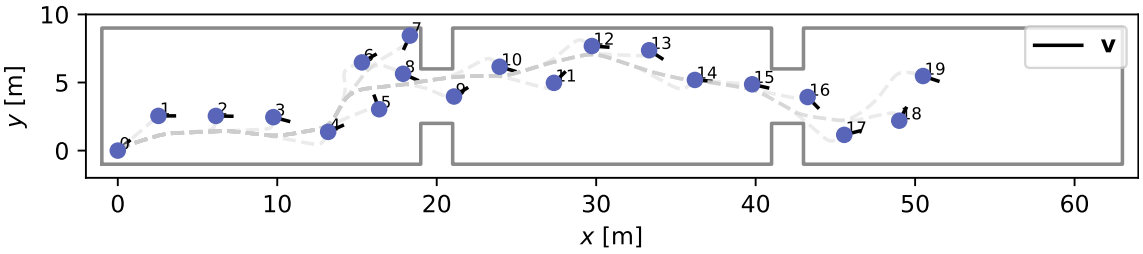
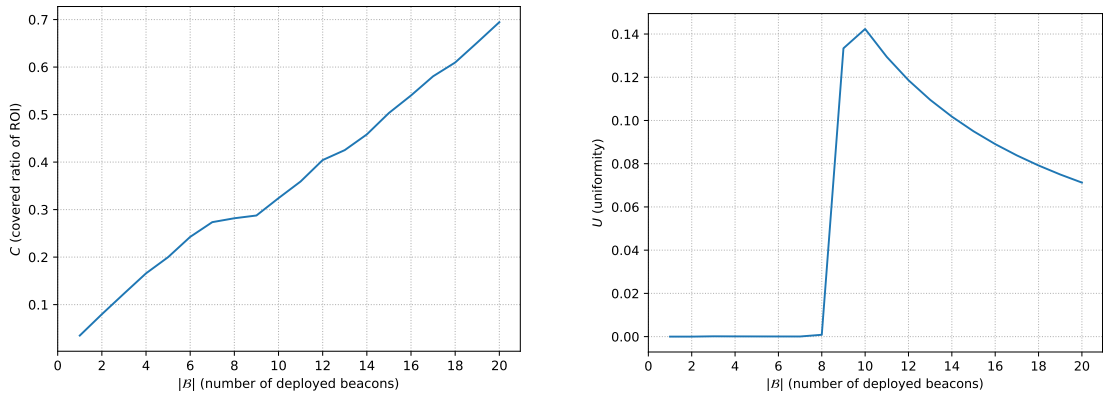


Figure 6.16: Configuration giving highest coverage after 20 deployed beacons in the Stripa environment. Exploration vectors are represented as black lines.



(a) Coverage **(b)** Uniformity

Figure 6.17: Coverage- and uniformity trajectory of deployment yielding highest coverage after 20 deployed beacons in the Stripa environment.

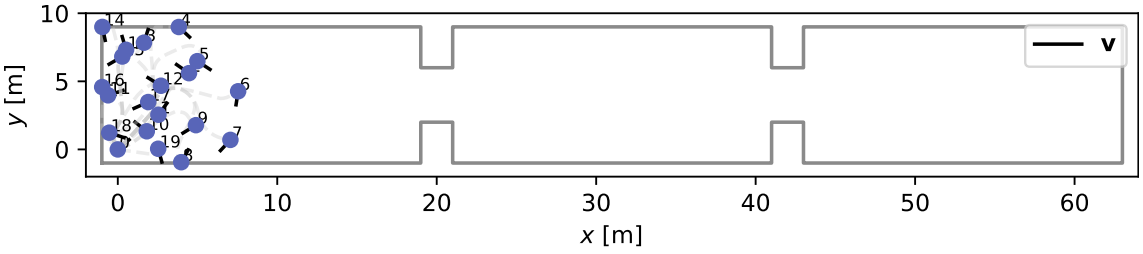
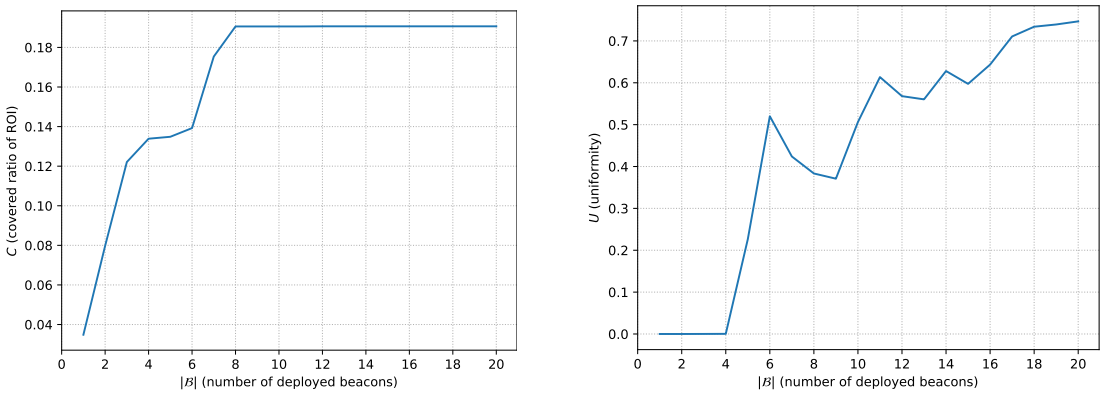


Figure 6.18: Configuration giving lowest coverage 20 deployed beacons in the Stripa environment. Exploration vectors are represented as black lines.



(a) Coverage **(b)** Uniformity

Figure 6.19: Coverage- and uniformity trajectory of deployment yielding lowest coverage 20 deployed beacons.

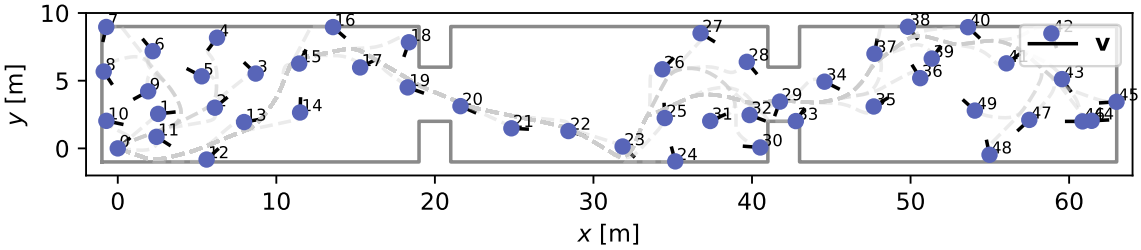
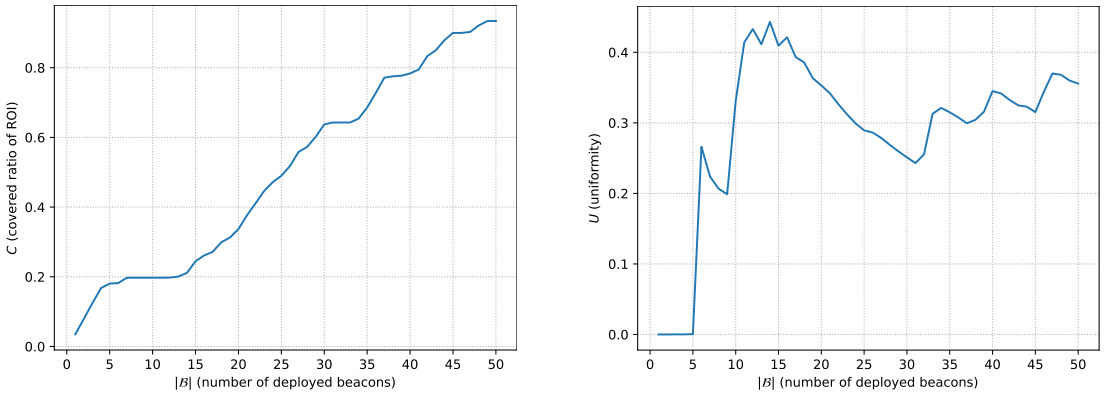


Figure 6.20: Configuration giving highest coverage after 50 deployed beacons in the Stripa environment. Exploration vectors are represented as black lines.



(a) Coverage **(b)** Uniformity

Figure 6.21: Coverage- and uniformity trajectory of deployment yielding highest coverage after 50 deployed beacons in the Stripa environment.

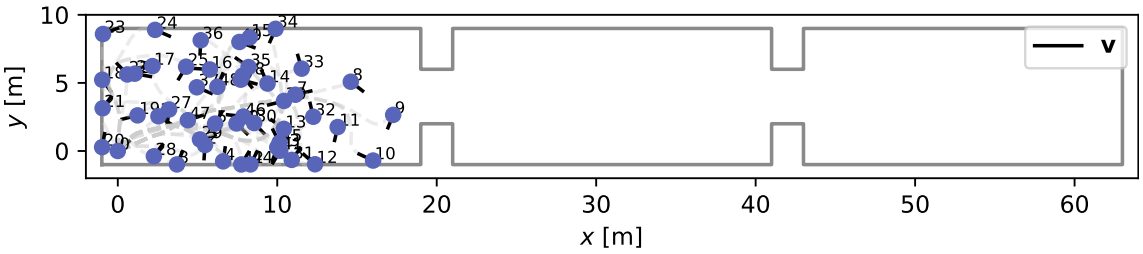
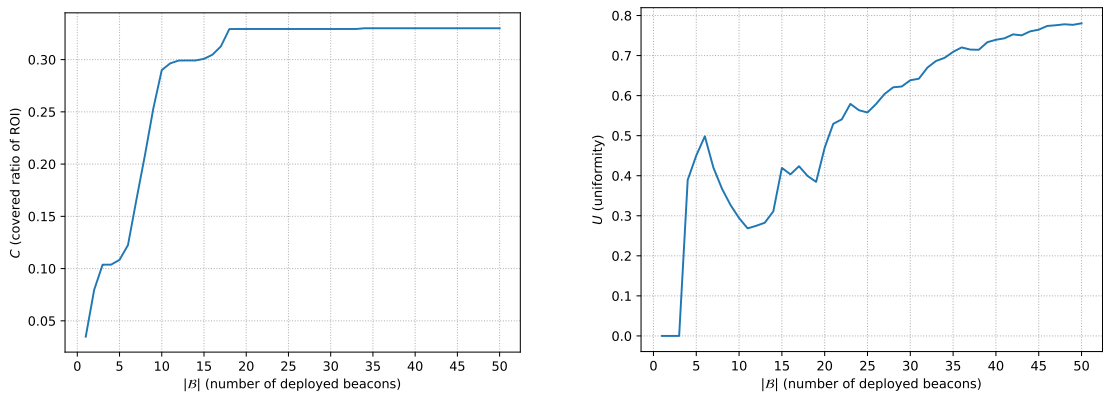


Figure 6.22: Configuration giving lowest coverage 50 deployed beacons in the Stripa environment. Exploration vectors are represented as black lines.



(a) Coverage **(b)** Uniformity

Figure 6.23: Coverage- and uniformity trajectory of deployment yielding lowest coverage 50 deployed beacons in the Stripa environment.

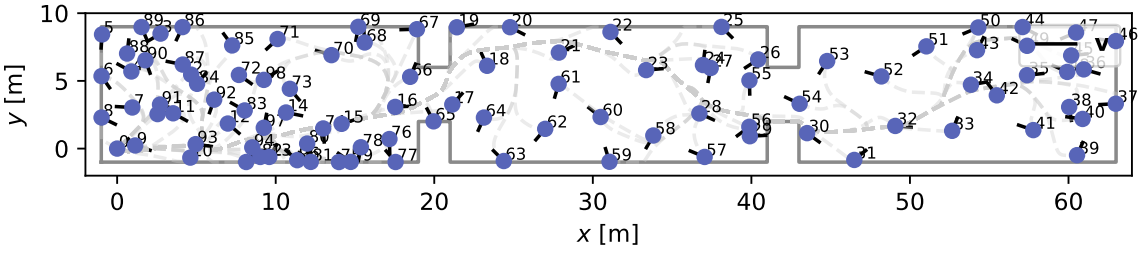
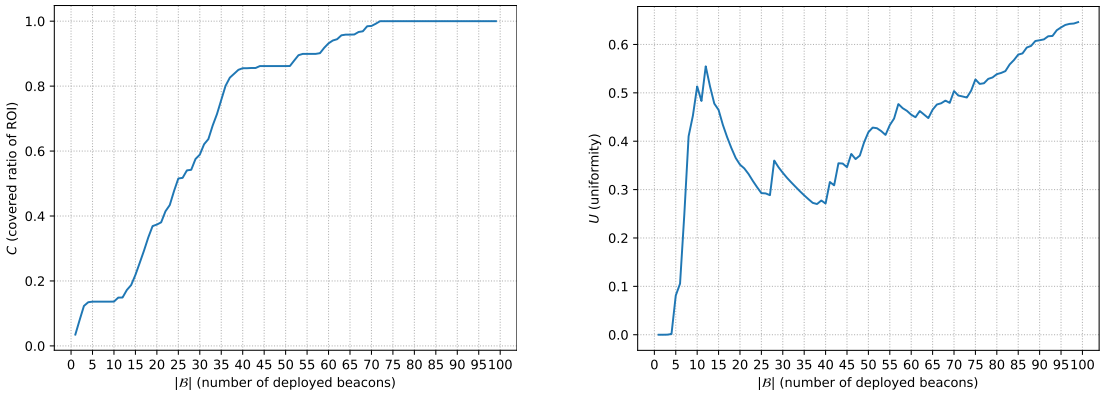


Figure 6.24: Configuration giving highest coverage after 100 deployed beacons in the Stripa environment. Exploration vectors are represented as black lines.



(a) Coverage **(b)** Uniformity

Figure 6.25: Coverage- and uniformity trajectory of deployment yielding highest coverage after 100 deployed beacons in the Stripa environment.

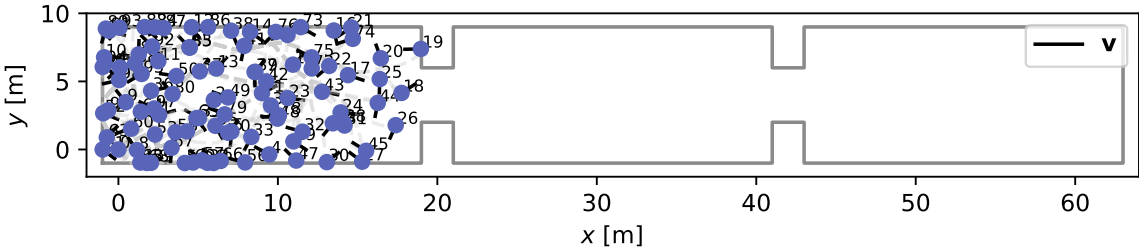
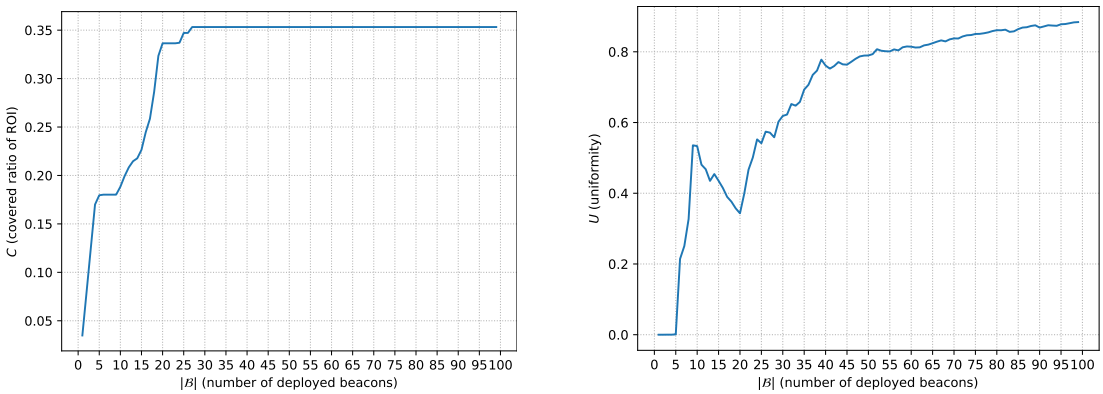


Figure 6.26: Configuration giving lowest coverage 100 deployed beacons in the Stripa environment. Exploration vectors are represented as black lines.



(a) Coverage **(b)** Uniformity

Figure 6.27: Coverage- and uniformity trajectory of deployment yielding lowest coverage 100 deployed beacons in the Stripa environment.

Chapter 7

Discussion

We discuss here the results obtained in Section 6.2. The discussion is partitioned into two sections where each section is dedicated to deployments within one specific environment.

7.1 Ten-by-ten

Simulations were performed in the Ten-by-ten environment in order to attain results comparable to those in [16] where the Distributed Self-Spreading Algorithm is used for distributed multi-agent deployment in a square ten-by-ten meter environment. The parameters of the RSS-mapped offset distance function (see Table 6.3) defined in (3.3) where chosen so that the communication range of beacons equals that used in [16].

We initially compare the results obtained with the proposed deployment scheme to those obtained by DSSA in Section 7.1.1. As coverage in [16] is defined as *sensor* coverage, whereas the focus of this thesis is *communication* coverage, a comparison of the coverage metrics for the two deployment schemes is omitted.

In Section 7.1.2, the effects of the parameters used in the RSS-mapped distance function (3.3) are evaluated.

7.1.1 Comparing with the Distributed Self-Spreading Algorithm (DSSA)

The results obtained in [16] are shown in Figure 7.1.

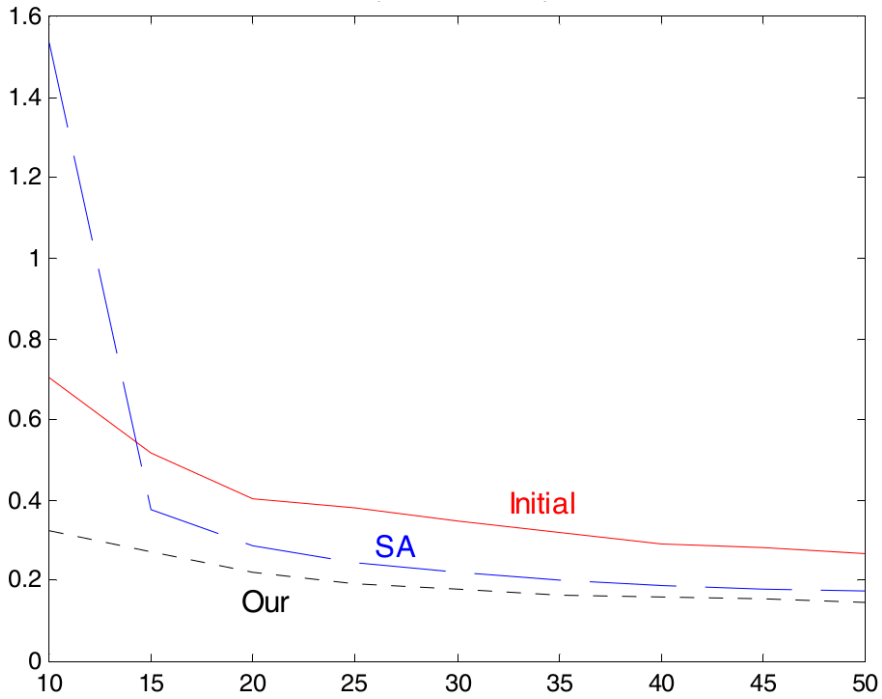


Figure 7.1: Average uniformity (y-axis) vs. network size (x-axis) over 100 runs of the Distributed Self-Spreading Algorithm [16] in the Ten-by-ten environment. For all network sizes, an initial random deployment is assumed yielding the uniformity marked in red. Uniformity at completion of the Distributed Self-Spreading Algorithm is marked in black. Source [16].

Comparing the uniformity trajectories in Figure 6.1, Figure 6.4, Figure 6.7 and Figure 6.10 to that in Figure 7.1, it is clear that the Distributed Self-Spreading algorithm (DSSA) vastly outperforms the proposed deployment scheme for large network sizes and for all RSS-mapped distance function parameters. The uniformity of configurations generated by the Distributed Self-Spreading Algorithm has a marginally inverse relationship with network size, whereas the proposed deployment scheme yields an increase in uniformity with increasing network size.

The opposing relationship between network size and uniformity for DSSA versus the presented scheme arises due to two inherent differences. In DSSA, network nodes exert *repulsive* forces on each other and are allowed to re-position concurrently. The magnitudes of the repulsive forces increase with decreasing inter-nodal distance and increase with increasing density. Thus, low uniformity is an emergent property of DSSA as nodes residing in areas of higher density repel neighbouring nodes, thus lowering the density within its area, causing the densities over all sections of the ROI to converge to the same

average value. Furthermore, as the magnitude of the force depends inversely with inter-nodal distance, a node with some neighbours closer to itself than others will be repelled from its closest neighbours leading to it being increasingly equidistant from all its neighbours, resulting in decreasing local uniformity.

Conversely, the proposed deployment scheme is incremental and attractive in nature. Deploying agents are attracted towards the centre of mass of the virtual set of particles created by its neighbours. The masses and locations of the virtual particles are not chosen with the goal of achieving low uniformity. Rather they are chosen so that the deploying agent should be guided to possibly unexplored space. Thus, the centre of mass towards which the deploying agent is attracted might lie at a location causing the agent's and any surrounding beacons' local uniformity to take on large values. Furthermore, due to the incremental nature of the proposed approach, nodes only re-position themselves once, and they do so one at a time. Thus there is no way for the network to reconfigure itself in a situation where beacons detect large values of local uniformity.

The choice of exploration direction and obstacle-induced forces play important roles in contributing to the increasing relationship between network size and uniformity of the proposed deployment scheme. These dependencies are made especially clear under Higher-max, High-max and Medium-max RSS-mapped distance parameters. As seen in Figure 6.3, Figure 6.6 and Figure 6.9, clustering occurs near walls as the network size increases. As the exploration direction of a beacon is chosen so that succeeding agents are directed away from previously deployed beacons (see Section 5.3.4), agents are to a larger extent guided towards walls as the interior of the environment is populated with beacons. As environment boundaries indirectly induce repelling forces in agents through range sensor measurements, an agent might settle closer to a deployed beacon than the rest of the neighbours of said beacon if the agent detects walls (see beacon 8 and 9 in Figure 6.3 (a)). As the deployed beacons cannot react, in the sense of re-positioning, to agents settling nearby, the local uniformity of beacons increases leading to an increase in uniformity.

The choice of agent-relative gains also plays a role in contributing to increased uniformity with increasing network size. In Figure 6.3, it is noticeable that in some cases, beacons of higher index are positioned close to beacons of lower index. Beacons 15, 31 and 48 in Figure 6.3 (d) exhibit such behaviour. This is caused by an agent choosing agent-relative gains as an exponential function of beacon index. Thus, the neighbour induced force from a beacon of smaller index is largely ignored when an agent has a neighbour of (much) higher index, causing the equilibrium point of that agent to be decided largely by its higher-index neighbours. In case this equilibrium is nearby a beacon of small index, clustering occurs, leading to increased uniformity.

7.1.2 Effect of RSS-mapped offset distance parameters

Parameter combinations used during deployment in the Ten-by-ten environment are summed up in Table 7.1, and the coverage- and uniformity metrics for different parameter values used in the RSS-mapped offset distance function are summarized in Figure 7.2.

Parameter combination name	$\bar{\xi}$	d_{perf}	d_{none}
Higher-max	20	2	~ 4.225
High-max	10	2	~ 4.335
Medium-max	5	2	~ 4.515
Low-max	1	2	6

Table 7.1: Parameters for the RSS-mapped offset distance function in (3.3).

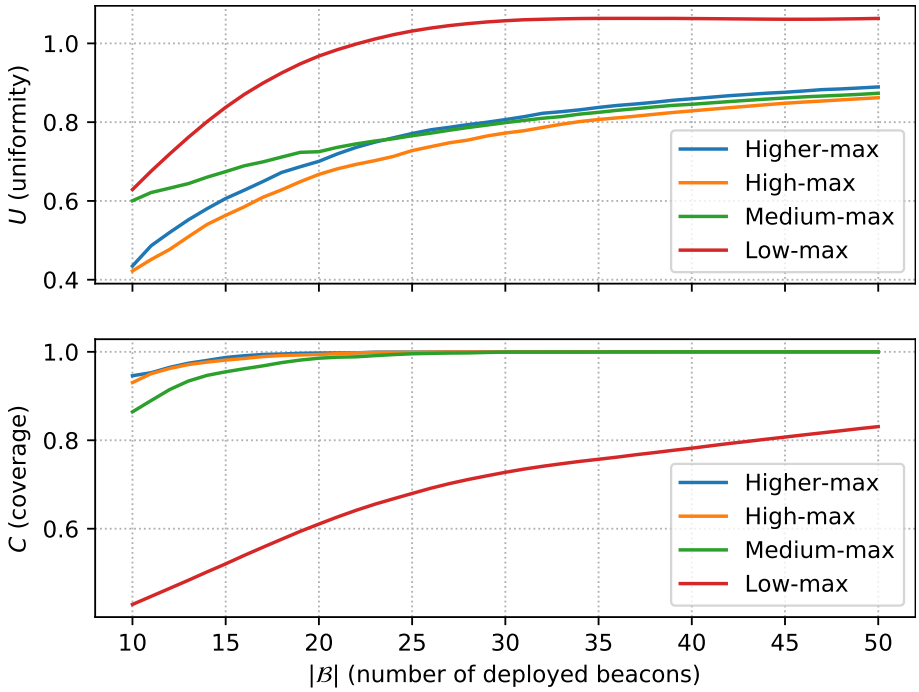


Figure 7.2: Average coverage and uniformity over 100 runs vs. network size for each parameter combination in Table 7.1 used in the RSS-mapped offset distance function (3.3) in the Ten-by-ten environment.

As seen in Figure 7.2, the average coverage is significantly lower for all network sizes when Low-max parameters are used in the RSS-mapped offset distance function than for any other parameter combination. Furthermore, the average uniformity is significantly higher. This is due to the fact that the maximum offset distance, $\bar{\xi}$, determines the maximum distance from a beacon to the equilibrium point the beacon imposes on an agent having that beacon as its only neighbour. Thus, when Low-max parameters are applied,

the maximum distance between a beacon and the equilibrium point it imposes on an agent with that beacon as its only neighbour is one meter as opposed to 5, 10 and 20 meters for the other parameter combinations. Indeed, the actual distance between the landing location of an agent with one neighbouring beacon and that beacon is defined purely by $\bar{\xi}$ iff. $\bar{\xi} \leq d_{perf}$. This, in effect, causes beacons to reach more dense configurations (see Figure 6.12), and thus lower coverage under Low-max RSS-mapped offset distance function parameters than for any other parameter combination.

Furthermore, as seen in Figure 6.12, beacons are in general positioned along a clear path under Low-max parameters, leading to high local uniformity. Low local uniformity is achieved if all neighbours of a beacon are approximately equidistant from the beacon computing its local uniformity (and zero if all neighbours are equidistant from that beacon). Under Low-max parameters, the proposed deployment scheme generates configurations in which the neighbours of any given beacon lie progressively further away from that beacon, leading to high local uniformity. As this is the case for all beacons, a high value of overall uniformity is reached under Low-max RSS-mapped offset distance parameters.

For the Higher-, High- and Medium-max parameters, Figure 7.2 shows that the configurations generated by the proposed deployment scheme yield almost equal performance in terms of uniformity and coverage. This is largely due to the model of the RSS-mapped distance in (3.3). Although the maximum RSS-mapped offset distance, $\bar{\xi}$, sets a maximum bound to how far the equilibrium point imposed on an agent can be from the beacon imposing the equilibrium point at any given time, the distance from the beacon to the landing location of the agent fulfils $d_{i,j} = \xi_{i,j}$, where $d_{i,j}$ is the distance between the agent and the beacon and $\xi_{i,j}$ is the RSS-mapped offset distance. Iteratively solving $d_{i,j} = \xi_{i,j}$ where $\xi_{i,j}$ is defined as in (3.3) for Higher-, High- and Medium-max parameters yields the distances from a beacon to the landing location of an agent with only one neighbour presented in Table 7.2.

Parameter comb. name	Maximum RSS-mapped offset distance ($\bar{\xi}$)	Distance from beacon, j, to equilibrium of an agent under (5.18) with $\mathcal{N}_a(i) = \{j\}$
Higher-max	20 [m]	3.60 [m]
High-max	10 [m]	3.41 [m]
Medium-max	5 [m]	3.07 [m]

Table 7.2: Distance from beacon to landing location of agent with only one neighbour for different parameter combinations applied to the RSS-mapped distance function (3.3).

From Table 7.2 it is evident that a higher maximum RSS-mapped value yields equilibrium points for agents placed further away from its neighbouring beacons than for a lower maximum value. However, this difference is small for Higher- and High-max parameter

combinations and is reflected in the coverage plot in Figure 7.2 where complete coverage ($C = 1$) is reached with a slightly smaller network size under Higher-max- versus High-max parameters. The distance between an agent's landing location and a neighbouring beacon under Medium-max parameters (3.07 [m]) is significantly smaller than for the other two parameter combinations, yielding slower convergence to complete coverage as seen in Figure 7.2.

For smaller network sizes, configurations generated under Medium-max RSS-mapped distance function parameters yield significantly larger average uniformity than configurations generated under Higher- and High-max RSS-mapped distance function parameters as seen in Figure 7.2. This might be caused by beacons residing closer to other beacons under Medium-max RSS-mapped distance function parameters, giving more beacons with more than one neighbour. The greater inter-beacon distances generated under Higher- and High-max parameter combinations (as per Table 7.2 and Figure 6.3, Figure 6.6 vs Figure 6.9) might cause more beacons to have only a single neighbour. As the local uniformity is zero for beacons with only a single neighbour, they make no contribution to the overall uniformity resulting in a lower overall uniformity value.

Although configurations generated by the proposed approach when applying the Low-max parametrization to the RSS-mapped offset distance function yield much slower coverage than with other parametrizations, the fact that increasing network size (i.e. deploying more agents) on average increases coverage (see Figure 7.2) is a promising result in itself. For mobile agents equipped with weak communication units, i.e. low signal strength over short distances, securing short distances between beacons might be needed. Thus a low value can be assigned to the maximum RSS-mapped offset distance in order to ensure that agents settle nearby deployed beacons while at the same time increasing coverage.

7.2 Stripa

Simulations were performed in the Stripa environment in order to evaluate the performance of the proposed deployment scheme in a more complex environment in which narrow passages must be traversed in order to explore and cover new regions. In this environment, only Higher-max parameters were used in the RSS-mapped distance function as it lead to fastest coverage in the Ten-by-ten environment.

In Section 7.2.1, the overall performance of the configurations generated by the proposed deployment scheme in the Stripa environment is assessed. Then, situations in which the proposed deployment scheme yield unsatisfactory coverage are discussed in Section 7.2.2 and Section 7.2.3.

7.2.1 General performance

As seen in Figure 6.15, the average coverage over 100 deployments is increasing steadily as the size of the deployed network increases, even in the more trying Stripa environment. The maximum- and minimum coverage trajectories do not exhibit the same steadily increasing trend. Both trajectories show a step-like behaviour in which the coverage remains unchanged over certain intervals of network size. For certain deployments, increasing the network size by one (i.e. deploying one more agent) does not increase coverage due to the deploying agent landing nearby beacons that already cover the surrounding area. As the network has increased sufficiently in size, however, the deploying agent might be guided to a region of space where no beacons are located. Thus, as the deploying agent settles and re-instantiates as a beacon, it covers the previously un-covered area surrounding it, leading to a sudden increase in coverage.

In the most favourable cases, complete coverage ($C \approx 1$) is reached at a network size of 70 beacons. In some cases, however, coverage of only 0.39 is reached after deploying 100 agents. This discrepancy points to the proposed deployment scheme lacking robustness, as deploying networks of the same size into the same unknown environment might lead to highly different coverage.

The average uniformity depicted in Figure 6.14 shows that the network configurations generated by the proposed deployment scheme in the Stripa environment increase in uniformity and thus become less uniformly distributed as network size increases. As was the case in the Ten-by-ten environment, this average increase in uniformity can be attributed to obstacle-induced forces and the choice of agent-relative gains leading to beacons of high index demanding that agents settle nearby beacons of lower index, both causing unbalanced configurations.

7.2.2 Static exploration vectors and agent-relative gains causing ineffective coverage

As the exploration vectors are the driving factor in inducing agents to move towards and settle within previously unexplored areas, a "badly" chosen exploration vector can have a major impact on the coverage reached by a network.

Figure 6.18 shows a situation where the exploration vector chosen by a single beacon has a lasting negative impact on the configuration, and thus the coverage reached by the network as a whole. In Figure 6.18, the exploration vector chosen by beacon 7 is based solely on it's knowledge of the position of beacon 6. Beacon 6 is beacon 7's only neighbour, and beacon 7 does not sense sufficient asymmetry in the environment in order to adjust its exploration vector to compensate for the nearby wall. Thus, beacon 7 selects its exploration vector as a vector pointing away from beacon 6 and rotated by an angle between $\pm\frac{\pi}{2}$ (due to beacon 7's exploration sector being the entire unit circle). As depicted Figure 6.18, beacon 7's exploration vector points in a direction such that when agent 8 deploys, it is guided towards the nearby wall and thus, upon encountering the wall, towards the direction of beacon 0. This renders beacon 8 at a position such that its largest exploration sector resides within the hull of the previously deployed beacons. In turn, beacon 8 guides beacon 9 towards already covered area, beacon 9 guides beacon 10 towards already covered area and so on.

In Figure 6.22 and Figure 6.23 (a), it is clear that the heuristic applied when selecting exploration directions, alongside the ever increasing beacon gains, can lead to situations in which drastically increasing network size might not lead to increased coverage. For clarity, Figure 7.3 and Figure 7.4 show configurations for the deployment depicted in Figure 6.22 at network sizes of 18 and 30 beacons respectively.

As depicted in Figure 7.3, the network consisting of the initial 18 beacons covers almost the entire first section of the Stripa environment. Using the greedy heuristic, beacon 17's exploration vector points in a direction within the surrounding sector of most free space. Intuitively and necessarily, an increase in coverage would only be achieved if new beacons reached the next section of the environment. However, due to the exploration vectors being static for all beacons, and the beacon gains being increasing functions of beacon index causing the equilibrium point of an agent to be determined largely by its higher-index neighbours, the succeeding beacons do not reach the next section of the environment.

As seen in Figure 7.4, deploying 10 more agents yields no increase in coverage as the heuristic applied when selecting exploration vectors fails to generate exploration vectors leading agents towards previously unexplored space.

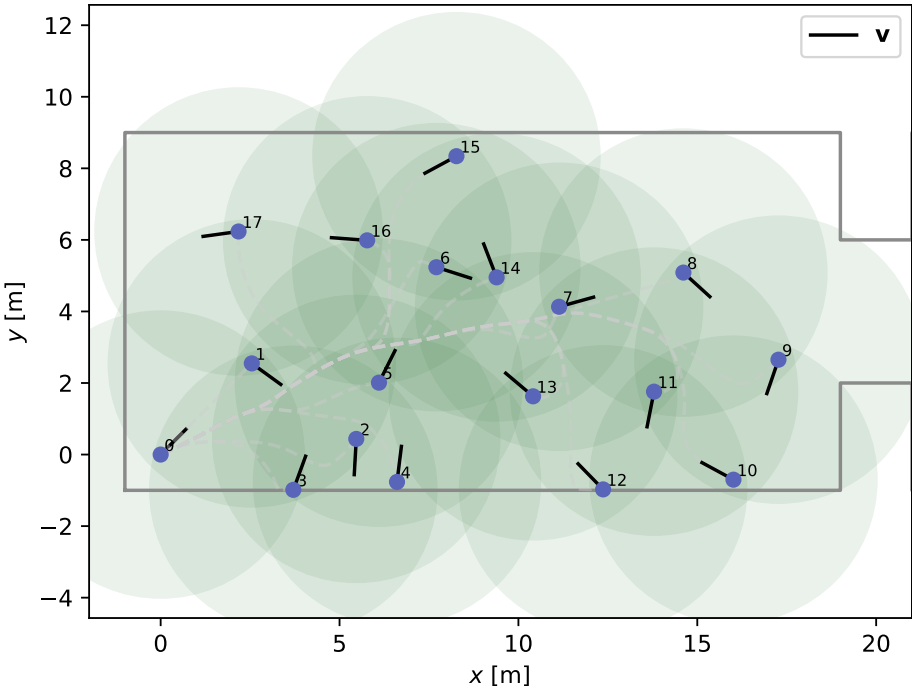


Figure 7.3: Configuration reached during deployment depicted in Figure 6.22 at a network size of 18 beacons. Green disks show area covered by each beacon. Communication radius is computed using (6.2).

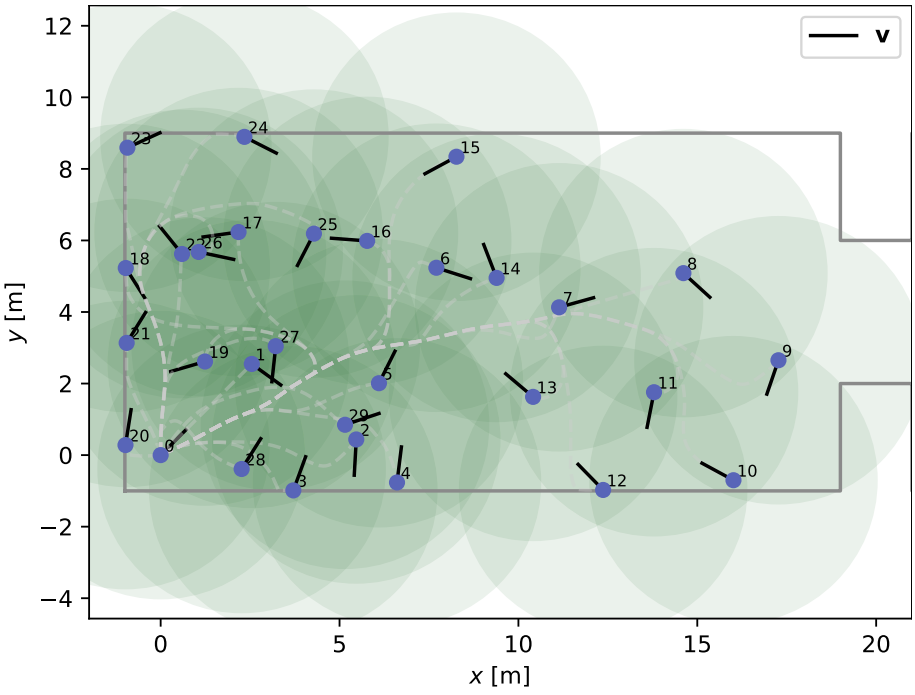


Figure 7.4: Configuration reached during deployment depicted in Figure 6.22 at a network size of 30 beacons. Green disks show area covered by each beacon. Communication radius is computed using (6.2).

7.2.3 Walls hindering exploration

Figure 6.26 shows the configuration of the 100−beacon network yielding lowest coverage of all the one hundred deployed networks of size 100. Although the network size is large enough to cover the entire environment, one single event sparked a chain of events leading to all beacons settling within the first sector of the environment.

For clarity, the configuration in Figure 6.26 is shown in Figure 7.5 with only the initial 20 beacons included. As the figure shows, the exploration vector of beacon 18 points in the direction of the narrow hallway leading to the second sector of the Stripa environment. However, agent 19 does not explore the hallway and reaches an equilibrium point within the first section of the environment.

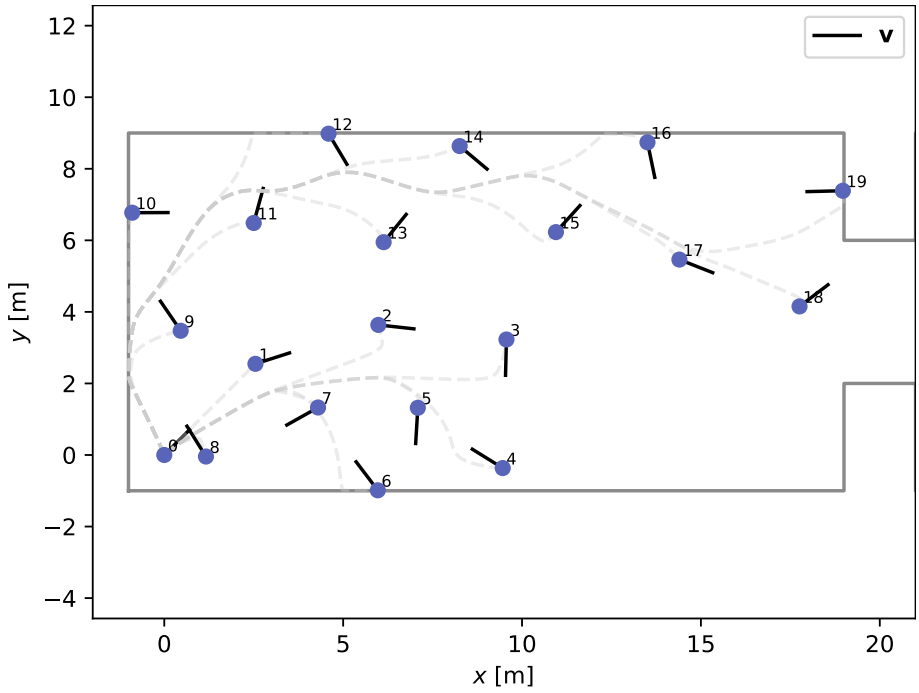


Figure 7.5: Configuration reached during deployment depicted in Figure 6.26 at a network size of 20 beacons.

The RSS-mapped offset distance between an agent and a beacon depends inversely on the distance between the agent and the beacon. This causes the location $\mathbf{p}_j = \mathbf{x}_j + \xi_{i,j}\mathbf{v}_j$ of the virtual particle created by a beacon j to move progressively further away from the beacon as the distance between the agent and the beacon decreases and vice versa.

In the situation depicted in Figure 7.5, the RSS-mapped offset distance between agent 19 and beacon 18 causes $\mathbf{p}_{18}$ to be located behind the wall at $x = 19$ meters for all sufficiently large values of $\xi_{18,19}$. Thus, as agent 18 moves towards beacon 19, the centre of mass of the virtual set of particles to which the agent gravitates moves further in the direction defined by $\mathbf{v}_{18}$, and is also located behind the wall at $x = 19$ meters. As the

neighbour induced force leads the agent directly towards the virtual centre of mass, the agent eventually settles near the wall, far away from the neighbour induced equilibrium, due to the obstacle-induced force counteracting movement into the wall.

Post-deployment, beacon 19 computes its exploration vector by greedy heuristic, taking into account the wall at $x = 19$ meters. As shown in Figure 7.5 the exploration vector of beacon 19 points towards the interior of the first section of the Stripa environment, causing subsequent agents to be guided in that direction, yielding little increase in overall coverage.

Chapter 8

Future work

Coverage metric based on local information As of now, no local metric for *coverage* is defined. Uniformity captures only the topology of the deployed network but does not consider the environment in which the network resides. Hence, uniformity is not a suitable metric for measuring how the network performs in terms of covering an area.

In simulations, coverage was computed as the fraction of the area of the union of the communication disks of all beacons and the area of the region of interest. In real-world applications, however, there exists no clear communication disk as environmental effects alter the range at which a certain RSS can be achieved. Furthermore, the area of the *unknown* environment cannot be known a priori.

Finding a coverage metric using only information available to beacons, i.e. range measurements and the positions of neighbours, is a task that has yet to be completed and should be investigated.

Termination criterion In all simulations, a pre-determined number of agents were deployed. In real-world situations, when the area to be covered is truly unknown, the number of beacons needed to cover the area can not be known a priori. Due to this, agents should be progressively deployed into the environment until some termination criterion has been fulfilled.

A termination criterion must be defined before the proposed deployment approach can be applied in real-world situations and will most likely be some condition on the local coverage metric, which is also yet to be defined.

Adjusting beacon exploration vectors and gains upon clustering Through simulations, it became evident that the static exploration vectors and beacon gains in some situations lead to large networks clustering, and thus covering only a fraction of the region of interest. Re-adjusting beacon gains and exploration vectors upon detecting clustering might prevent this from happening and should be investigated further.

Re-adjusting beacon gains and exploration vectors would demand some form of coordination, as exploration vectors would have to be chosen so that succeeding agents would be guided towards the perimeter of the covered area.

Wall following when settling far from neighbour-induced equilibrium As discussed in Section 7.2.3, the RSS-mapped offset distance can cause agents to diverge from the path taken by previously deployed beacons prematurely and thus encounter obstacles when gravitated towards the neighbour-induced equilibrium. This can cause agents to settle far away from their neighbour-induced equilibrium point leading to poor coverage.

Detecting that an agent settles far from its neighbour induced equilibrium and initiate some sort of wall following might prevent situations where obstacles hinder exploration and lead to lacklustre coverage and should be investigated.

Realistic model of signal strength In this thesis, the RSS-mapped distance function did not take into account environmental effects on RSS. In real-world situations, RSS can drop significantly due to diffraction, reflection and refraction if the sender and the receiver are not within line-of-sight [37]. In the future, RSS could be modelled by the log-normal path-loss model [18], and this should be used as input to the RSS-mapped distance function.

Collision avoidance If the proposed approach is to be applied to ground-travelling robots, some sort of collision avoidance must be implemented. For the purposes of this thesis, it was assumed that beacons resided at ground level and that the deploying agents flew at some altitude above ground level, thus avoiding collisions altogether, and no explicit collision avoidance was needed. For ground-travelling robots, however, the proposed approach must be extended to take into account that the deploying agent might collide into beacons already residing within the environment.

Include heading in simulations Throughout simulations, agents were assumed to have a heading aligned with the inertial x_n-axis. This is valid for quadcopters but is not suitable for wing-based or ground-travelling agents. In the future, a heading term should be added

to the dynamics of the agents and its effect on general behaviour, and especially on the obstacle induced field, should be investigated.

Chapter 9

Conclusion

This thesis proposed a novel potential field-based distributed high-level control law whose objective is to make a set of mobile agents incrementally self-deploy into an environment of unknown topology and extent, with the final purpose of forming a connected network that spans the environment, and this without using centralized forms of control or information.

The novelty of the approach taken here arises from the fact that, to our knowledge, the problem of sequentially self-deploying a wireless sensor network within an unknown environment without centralized information has not been addressed before. The mobile agents are assumed to possess only four range sensor returning the distance to nearby obstacles in their respective directions. Thus, mobile agents are incapable of measuring the area around them, contributing to the novelty of the presented approach. Furthermore, a known inter-agent communication range is commonly assumed when performing deployment/area coverage. This assumption is removed, further contributing to the novelty of the proposed approach.

Due to the novelty of the objective itself, incremental deployment within an obstacle-less 1-dimensional environment was initially considered. Basing the approach on potential fields, a distributed control law, modelled as an attractive force towards the centre of mass of a virtual set of point masses generated by agents already within the environment, was synthesized. We proved that, for certain point-mass locations and masses, increasing the network size by deploying more agents guarantees an increase in network coverage.

Inspired by the field used for deployments in a 1D environment, a potential field-based control law for incremental deployment within unknown 2D environments was synthesized. The potential field used for the self-deployment of agents is a combination of a neighbour-induced attractive potential field used for exploration and a range sensor-induced repulsive potential field used for obstacle avoidance. As in 1D, the neighbour-

induced potential field affecting the deploying agent is generated by static agents already residing within the environment and is attractive towards the centre of mass of a collection of virtual point masses. The point mass locations are decided by the locations of the static agents already within the environment perturbed by an amount decided by an artificial function depending on the signal strength between the static agents and the deploying agent. Point masses are perturbed in the direction defined by the "exploration vector" of static agents. A greedy heuristic was used for selecting the exploration vector of a deployed agent so that succeeding agents were guided towards possibly un-explored areas.

Two performance metrics were used in order to assess the performance of the deployed networks. Uniformity was used in order to assess to what extent the deployed network exhibited clustering. Furthermore, it was deemed especially fitting due to it being dependant only on information available locally to each agent. Coverage was used in order to assess the performance of the deployed networks in terms of providing communication service within the environment. However, this metric was deemed applicable only in a simulation environment as it depends on information not available to agents in real-world scenarios.

Simulations were performed in two different environments. In a convex ten-by-ten meter environment, the parameters of a synthesized signal-strength to distance offset function were varied in order to emulate deployments of agents with different communication capabilities. For all parametrizations of the synthesized signal-strength to distance offset function, self-deploying agents within the ten-by-ten meter environment yielded satisfactory coverage.

Simulations in the Ten-by-ten environment showed that configurations generated using agents of weaker communication capabilities exhibited more clustering behaviour and gave smaller coverage than configurations generated using agents of stronger communication capabilities for all network sizes. This was caused by the weaker communication platforms yielding smaller signal strength to distance offsets, causing agents to settle in more compact configurations. Hence, a larger network comprised of agents with weak communication capabilities had to be deployed in order to achieve satisfactory coverage than in situations where agents possessed stronger communication platforms. Furthermore, the incremental nature of the proposed deployment scheme caused the deployed networks to exhibit a higher degree of uniformity compared to network configurations generated by the Distributed Self-Spreading Algorithm [16]. This was caused by the inability of previously deployed (static) agents to re-position themselves.

In a non-convex environment modelled after a well-known section of the NTNU Gløshaugen campus, the proposed deployment scheme again yielded, on average, satisfactory coverage for large networks comprised of agents with high-performing communication platforms. Some deployments, however, yielded large networks covering only a

small portion of the environment.

In certain situations, due to the fact that the exploration vector of a static agent remains static after initial computation and that the location of the centre of mass towards which the deploying agent gravitate is decided largely by its neighbour of largest index, deploying agents were guided towards already well-covered areas. In situations where one section of the environment was well covered, and the location of the most recently deployed agent was not in the vicinity of the border between covered and uncovered space, deploying agents gravitated towards already well-covered sections causing a snowball effect in which a vast increase in network size yielded little to no increase in coverage.

Furthermore, a large maximum distance offset in some situations caused the deploying agent to prematurely diverge too far from the path taken by its predecessor. If the deploying agent had followed the exact path of its predecessor before following a trajectory along its exploration vector, new areas would have been covered. However, the distance offset and exploration vector of the deploying agent's predecessor in some situations placed the centre of mass towards which the deploying agent gravitated far from the deploying agent's predecessor, and indeed behind a wall. As the neighbour induced force guided the deploying agent directly towards the centre of mass, obstacle encounters sometimes lead the agent to fail in exploring uncovered areas.

The results presented here show that, in both of the tested environments, deploying a sufficient amount of agents using the proposed high-level control law yielded satisfactory coverage in most situations. However, as the deployment scheme lacks robustness, it can not yet be applied in real-world scenarios. In the future, the problem of narrow hallways/passages causing agents not to explore un-covered areas should be addressed. Furthermore, a coverage metric depending only on information available locally to an agent must be synthesized along with a termination criterion deciding when to halt the deployment of new agents.

Bibliography

[1] Hannah Ritchie and Max Roser. Natural disasters. *Our World in Data*, 2014. https://ourworldindata.org/natural-disasters.

[2] Merriam-Webster Merriam-Webster.com Dictionary. First responder.

[3] M. Elbanhawi, A. Mohamed, R. Clothier, J.L. Palmer, M. Simic, and S. Watkins. Enabling technologies for autonomous mav operations. *Progress in Aerospace Sciences*, 91:27–52, 2017.

[4] 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.

[5] Mrunal Gavhale and Pranay D. Saraf. Survey on algorithms for efficient cluster formation and cluster head selection in manet. *Procedia Computer Science*, 78:477–482, 2016. 1st International Conference on Information Security & Privacy 2015.

[6] Shaimaa M. Mohamed, Haitham S. Hamza, and Iman Aly Saroit. Coverage in mobile wireless sensor networks (m-wsn): A survey. *Computer Communications*, 110:133–150, 2017.

[7] Douglas Gage. Command control for many-robot systems. 10, 11 1991.

[8] Levent Bayındır. A review of swarm robotics tasks. *Neurocomputing*, 172:292–321, 2016.

[9] Matarić Howard and Sukhatme. An incremental self-deployment algorithm for mobile sensor networks. *Autonomous Robots*, 13:113–126, 2002.

[10] Steven Damer, Luke Ludwig, Monica LaPoint, Maria Gini, Nikolaos Papanikolopoulos, and John Budenske. Dispersion and exploration algorithms for robots in unknown environments - art. no. 62300q. *Proceedings of SPIE - The International Society for Optical Engineering*, 05 2006.

[11] M.R. Pac, A.M. Erkmen, and I. Erkmen. Towards fluent sensor networks: A scalable and robust self-deployment approach. pages 365–372, 2006.

[12] Andrew Howard, Maja J. Matarić, and Gaurav S. Sukhatme. Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. pages 299–308, 2002.

[13] Xiangyu Yu, Weipeng Huang, Junjian Lan, and Xin Qian. A novel virtual force approach for node deployment in wireless sensor network. pages 359–363, 2012.

[14] Eric W. Weisstein. "delaunay triangulation." from mathworld–a wolfram web resource.

[15] Y. Zou and Krishnendu Chakrabarty. Sensor deployment and target localization based on virtual forces. 2:1293–1303 vol.2, 2003.

[16] 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.*, volume 3, pages 1597–1602 vol.3, 2003.

[17] Adrian N. Bishop, Barış Fidan, Brian D.O. Anderson, Kutluyıl Doğançay, and Pubudu N. Pathirana. Optimality analysis of sensor-target localization geometries. *Automatica*, 46(3):479–492, 2010.

[18] Zeyuan Li, Pei-Jung Chung, and Bernard Mulgrew. Distributed target localization using quantized received signal strength. *Signal Processing*, 134:214–223, 2017.

[19] Bin Xu, Guodong Sun, Ran Yu, and Zheng Yang. High-accuracy tdoa-based localization without time synchronization. *IEEE Transactions on Parallel and Distributed Systems*, 24(8):1567–1576, 2013.

[20] Synchronization, toa and position estimation for low-complexity ldr uwb devices. In *2005 IEEE International Conference on Ultra-Wideband*, pages 480–484, 2005.

[21] Sabudin Elia nadira, Rosli Omar, and Che Ku Nor Hailma. Potential field methods and their inherent approaches for path planning. 11:10801–10805, 01 2016.

[22] J. Safko H. Goldstein, C. Poole. *Classical Mechanics (3rd ed.)*. Addison-Wesley, 2002.

[23] Y.J. Cui S.S. Ge. Dynamic motion planning for mobile robots using potential field method. *Autonomous Robots*, 13:202–207, 2002.

[24] Sinan Gezici. A survey on wireless position estimation, 2008.

[25] N. Patwari, J.N. Ash, S. Kyperountas, A.O. Hero, R.L. Moses, and N.S. Correal. Locating the nodes: cooperative localization in wireless sensor networks. *IEEE Signal Processing Magazine*, 22(4):54–69, 2005.

[26] Sri Divya Chitte and Soura Dasgupta. Distance estimation from received signal strength under log-normal shadowing: Bias and variance. In *2008 9th International Conference on Signal Processing*, pages 256–259, 2008.

[27] F. Bullo and S. L. Smith. *Lectures on Robotic Planning and Kinematics*. 2020.

[28] Christopher Triola. Special orthogonal groups and rotations. page 4, 01 2009.

[29] Sotiris Papatheodorou and Anthony Tzes. *Theoretical and Experimental Collaborative Area Coverage Schemes Using Mobile Agents*. 11 2018.

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

[31] Micael S. Couceiro, Carlos M. Figueiredo, Rui P. Rocha, and Nuno M.F. Ferreira. Darwinian swarm exploration under communication constraints: Initial deployment and fault-tolerance assessment. *Robotics and Autonomous Systems*, 62(4):528–544, 2014.

[32] Jiangmin Chunyu, Zhihua Qu, Eytan Pollak, and Mark Falash. A new reactive target-tracking control with obstacle avoidance in a dynamic environment. In *2009 American Control Conference*, pages 3872–3877, 2009.

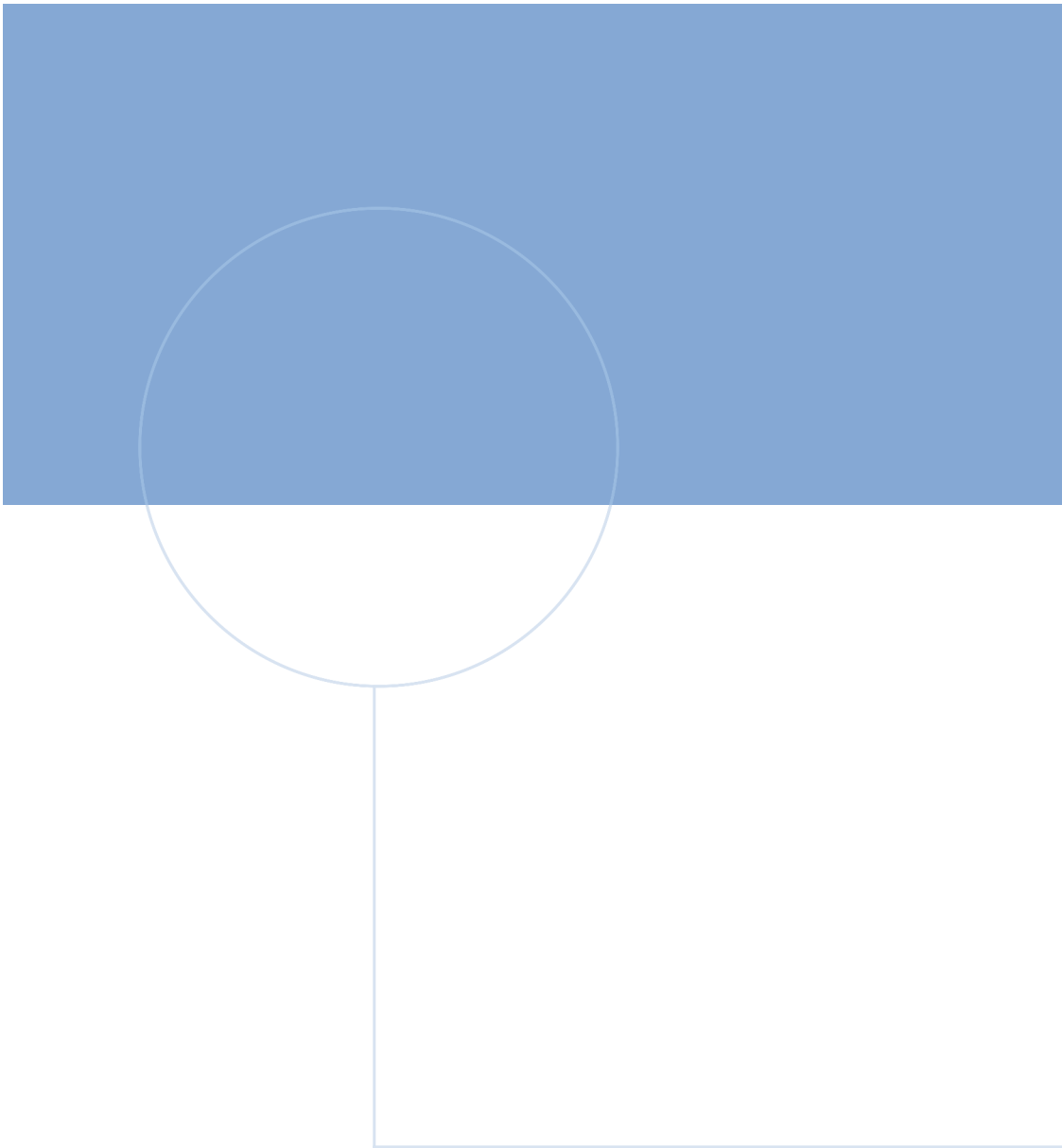
[33] Eric W. Weisstein. Angle bisector, from mathworld–a wolfram web resource.

[34] F.M. Dekking, C. Kraaikamp, H.P. Lopuhaä, and L.E. Meester. *A Modern Introduction to Probability and Statistics: Understanding Why and How*. Springer Texts in Statistics. Springer, 2005.

[35] Abdelmoumen Norrdine. An algebraic solution to the multilateration problem. 04 2015.

[36] Magnus Berdal. Mastercppsim @ https://github.com/mberdal/mastercppsim.

[37] Thomas Mangel, Matthias Michl, Oliver Klemp, and Hannes Hartenstein. Real-world measurements of non-line-of-sight reception quality for 5.9ghz ieee 802.11p at intersections. In Thomas Strang, Andreas Festag, Alexey Vinel, Rashid Mehmood, Cristina Rico Garcia, and Matthias Röckl, editors, *Communication Technologies for Vehicles*, pages 189–202, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.



NTNU

Norwegian University of
Science and Technology