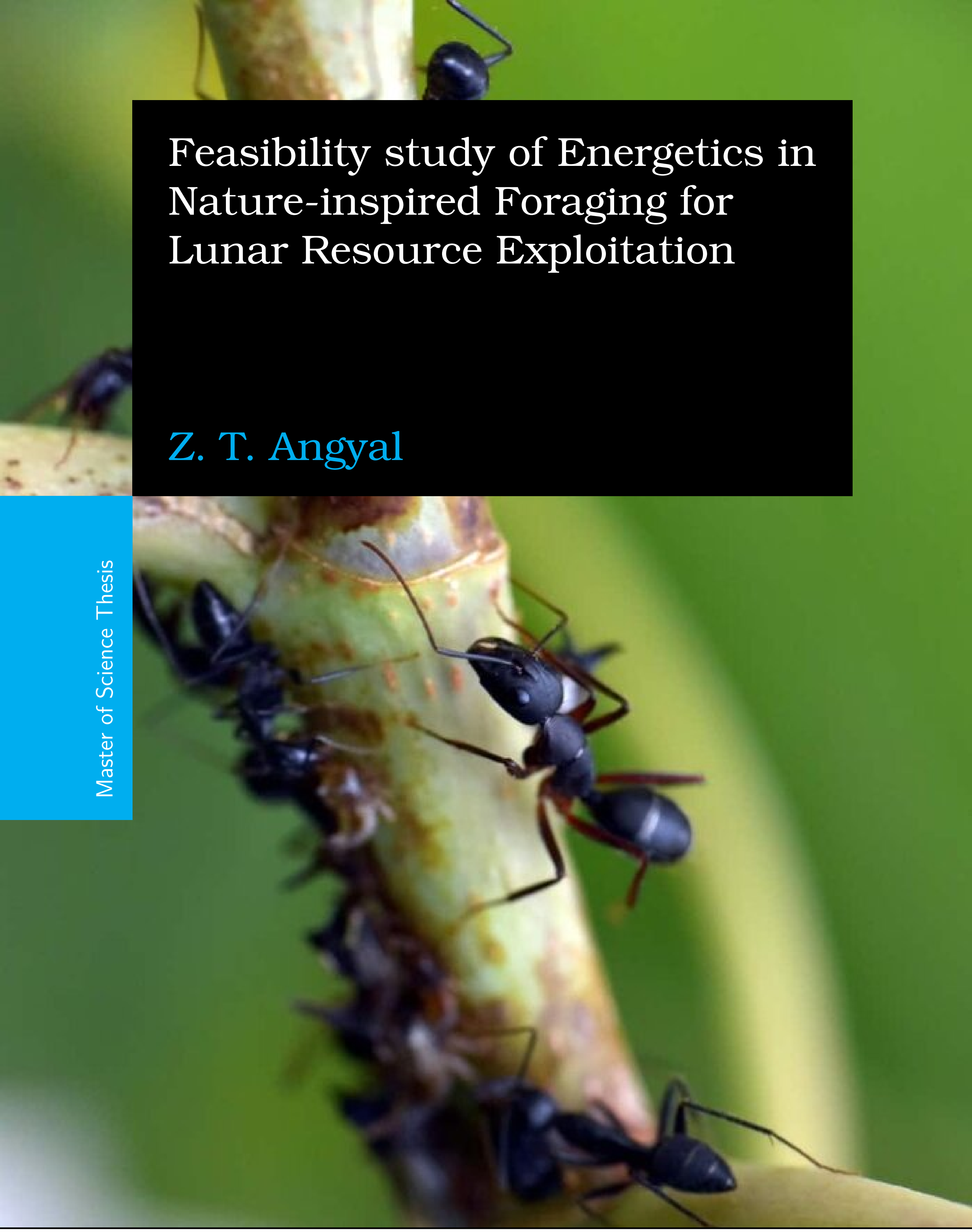


# Feasibility study of Energetics in Nature-inspired Foraging for Lunar Resource Exploitation

Z. T. Angyal

Master of Science Thesis

Delft University of Technology

Aerospace Engineering
Delft Center for Systems and Control

# Feasibility study of Energetics in Nature-inspired Foraging for Lunar Resource Exploitation

Master of Science Thesis

For the degree of Master of Science in Aerospace Engineering, Master of Science in Systems & Control at Delft University of Technology

Z. T. Angyal

March 20, 2024

Supervisors:
Dr. M. Mazo Espinosa
Dr. A. Menicucci

Faculty of Aeropsace Engineering, Faculty of Mechanical Engineering
Delft University of Technology

Delft University of Technology
Department of
Spaceflight Engineering, Delft Center for Systems and Control

The undersigned hereby certify that they have read and recommend to the Faculty of Aeropsace Engineering, Faculty of Mechanical Engineering for acceptance a thesis entitled

Feasibility study of Energetics in Nature-inspired Foraging for Lunar Resource Exploitation

by

Z. T. Angyal

in partial fulfillment of the requirements for the degree of

Master of Science in Aerospace Engineering, Master of Science in Systems & Control

Dated: March 20, 2024

Supervisor(s): ______________________
Dr. M. Mazo Espinosa

______________________
Dr. A. Menicucci

Reader(s): ______________________
Dr. R.T. Rajan

# Abstract


The rediscovered interest in space exploration has led to plans to establish outposts on the Moon and beyond. The lunar bases currently planned are to be manned incrementally, with robots performing most work. With new trends in robotics, the use of collaborating swarms has become more abundant. To support lunar operation, a distinct area within swarming, namely foraging, is proposed. Foraging systems have the primary objective of recovering resources. From literature, no foraging framework was found that included system maintenance in its mechanisms. This report aims to answer the question if this is possible while maintaining the benefits of foraging.

The research considers the fundamental act of recharging as its required maintenance task. To evaluate it dynamically, a rudimentary energetics model is included. For the framework of foraging, the work of [1] is used as a baseline. The newly proposed system implements an additional recharging region and role to perform recharging activities, both having major implications for role selection and agent operation. Furthermore, to enable navigation based on energy considerations, the experience communicated by mobile agents is amended to include the energy cost of a travelled path. In doing so, additional quality indicators of paths are available making path optimization a more dynamic process resulting in finer population behaviour. Finally, the decaying of beacons is updated and a fallback feature is introduced to maximize agent utilization.

The newly developed foraging system is evaluated using data collected through simulation in *Webots*. Simulation scenarios included obstacles with impenetrable boundaries and surfaces with increased rolling friction to emulate cost-expensive regions. Qualitative analysis identified all features of the foraging system as expected, both in the exploration and exploitation phase. Quantitative results proved that the system is able to function with the added requirements of recharging, perform path optimization with the additional path quality indicator, and can do so in various types of scenarios. With this, the research statement that foraging functionality is achievable with the practical considerations of robotics is confirmed to hold.

# Table of Contents

# List of Figures

# List of Tables

# Acknowledgements

With the completion of this thesis research, my long, eventful, and exceptional period at the Delft University of Technology comes to an end. Not only have I gained insight into and appreciation for the beauty of engineering, but I also learnt to take on the role of an engineer. I begin this next phase of my life without any regrets.

I would like to thank my supervisors Alessandra and Manuel for their unwavering help throughout this final stage of my academic career. I would furthermore like to thank my family and friends for all the encouragement they supported me with. Finally, I would like to thank Susan in particular for having stood by me through thick and thin. Our endless rubber-ducking sessions, unwinding adventures, and you telling me to focus time and time again made this all possible. Without the help of each and every one of you, this adventure of becoming an engineer would have been quite a bit less thrilling.

*"Utánuk! Utánam!"*

Delft, University of Technology — Z. T. Angyal
March 20, 2024

# Chapter 1

# Motivation of Research-field

The previous decade saw eleven missions attempting to touch down on the Moon[1]. The past four years alone saw nearly as many missions, with this increased rate expected to continue in the coming years. This growth in mission count is a result of a rediscovered motivation to venture into space and establish outposts in the process. Given the Moon's proximity and potential for science, its surface is a certain destination for endeavours. But what will these undertakings look like? Can old mission architectures be employed to support future missions?

## 1-1 Emerging Trends in Lunar Missions

This past year alone, five individual missions were launched to the Moon. Each attempt had mixed levels of success touching down on the lunar surface. While missions of the likes of Indian Space Research Organisation (ISRO)'s Chandrayaan-3 [20], China National Space Administration (CNSA) Chang'e-5 [49], and Intuitive Machines' IM-1[2] managed to reach the surface of the Moon and perform some if not all of their objectives, other missions like Roscosmos' LUNA25 [34] or Astrobotic's Peregrine mission [43] failed to either land safely or reach the Moon's vicinity at all. This illustrates how even 55 years after the first man set foot on the Moon, these endeavours are difficult to execute.

An interesting aspect of several recent missions is the shift in mission architecture. Where the likes of Chang'e-5 and IM-1 employed static vehicles, the missions of Chandrayaan-3, Chang'e-4 [23], and Japan Aerospace Exploration Agency (JAXA)'s Smart Lander for Investigating Moon (SLIM) [3] all employed one or multiple mobile rovers to explore the lunar surface. Chandrayaan-3's Pragyan rover managed to demonstrate roaming capabilities, and Chang'e-4's YUTU-2 (Figure 1-1) went beyond, taking measurements to this day [30]. But perhaps SLIM demonstrated the added robustness of mobile agents the most. Having landed at a

[1] https://science.nasa.gov/moon/missions/
[2] https://www.intuitivemachines.com/im-1

slanted angle, the lander's functionality is limited by its decreased solar power. The two lunar excursion vehicles LEV-1 and LEV-2 on the other hand have been able to complete their primary tasks without being impacted by the lander's fate. This robustness achieved by having multiple vehicles is starting to gain traction in application.

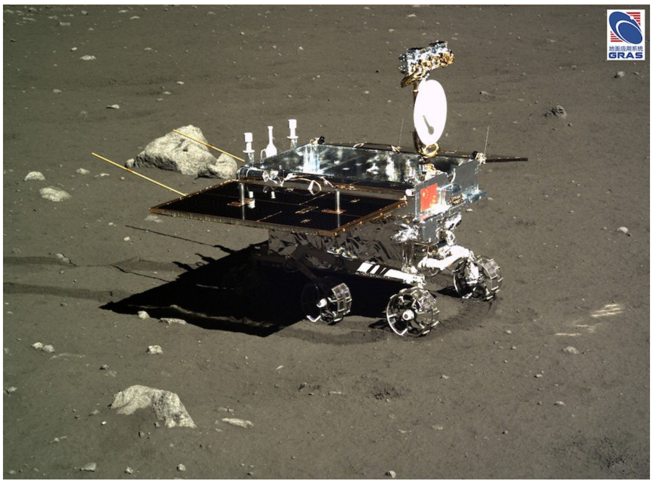



**Figure 1-1:** CNSA's YUTU-2 rover [3].

The trend of multiple vehicles being employed for excursions is in line with the proposed plans of agencies wanting to establish outposts on the Moon. To support plans such as its Gateway mission [4], National Aeronautics and Space Administration (NASA) conducted conceptual research into what it takes to uphold a lunar outpost. Both Robotic Lunar Surface Operations (RLSO) studies[44][7] concluded that the use of multiple collaborating robots is vital for sustainability. Future missions such as NASA's Cooperative Autonomous Distributed Robotic Exploration (CADRE)[4] already aim to demonstrate multi-agent functionality. With other agencies including CNSA, ISRO, JAXA, and European Space Agency (ESA)[27] showing interest in such outposts, the emergence of contributing technology is likely in the coming years.

Another development of the 21$^{st}$ century is the involvement of privatized actors in space operations. With the widened availability of space technology, companies of the likes of SpaceX, Blue Origin, Dawn Aerospace, AstroForge, and iSpace have expressed desires to contribute to launches and In-Situ Resource Utilization (ISRU). Such widespread interest is likely to boost the rate of progression given the concurrent nature of the commercial world.

## 1-2 Robotic Exploration

The role of robotics in planetary exploration and outpost upkeep cannot be underestimated. With the RLSO study identifying the need for such mechanisms as early as 1989, the impact of technology maturity on future missions is significant.

[3] https://spaceflightnow.com/2020/01/06/china-publishes-change-4-data-one-year-after-first-landing-on-far-side-of-the-moon/

[4] https://www.jpl.nasa.gov/missions/cadre

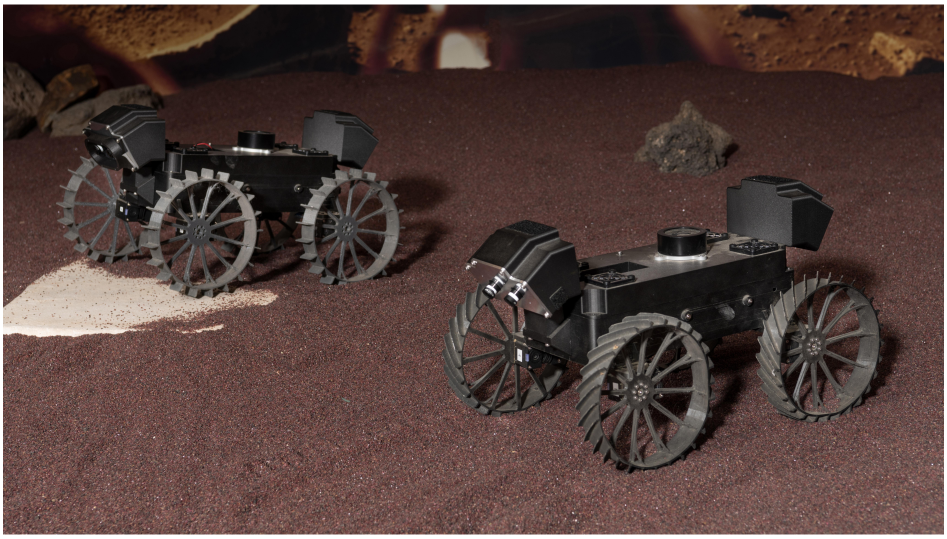


**Figure 1-2:** NASA's CADRE mission with rovers demonstrating multi-robot exploration [5].

The use of robotics depends on several factors. Mobility and tooling of similar vehicles have been and are being demonstrated with lunar excursions. A field not extensively touched upon is the intelligence of these vehicles. Equipping robots with extensive autonomy and collaboration removes the necessity of humans acting as an intermediary and often leads to optimized results. Achieving this will truly reshape lunar mission architecture.

The necessity of robotic application is underlined by both RLSO studies, and those in [18] and [46]. As Section 2-1 will touch upon, the resources readily available on the Moon are limited. Supporting human presence is a costly process, and highly inefficient during the early stages of an outpost. Having robots use the limited resources to construct a base of operation accelerates the process.

Developing highly intelligent robotic systems is key to enabling sustained lunar operations. With current missions in the early phases of implementing such mechanisms, the maturity of this field is considered limited. This hurdle needs to be addressed before future endeavours of the likes proposed by agencies can be attempted.

## 1-3 Summary

All in all, trends observed in most recent lunar missions point towards a shift in mission architecture towards the use of multiple mobile rovers. With current and past missions demonstrating capabilities and upcoming missions addressing the notion of multi-agent systems, the development of robotics on the Moon is solidified.

This trend is deemed a necessity given the plans of both agencies and private entities. To ensure operations while not requiring extensive amounts of resources or oversight, robotic intelligence is required. With current missions failing to implement such mechanisms, this topic is identified to be a critical field of research.

[5] https://www.jpl.nasa.gov/videos/nasas-cadre-mini-rovers-to-explore-the-moon-as-a-team

# Chapter 2

# Background

To achieve a well-rounded and robust intelligent robotic system, the challenges faced on the lunar surface and possible approaches to implementing autonomy must be understood. With that in mind, multiple literature resources are compiled in this section to provide the reader with ample knowledge on the topics at hand. An extended summary of sources can be found in the complete literature survey [6].

This section starts with Section 2-1 characterizing the lunar environment, surface, and resources it holds, of interest for In-Situ Resource Utilization (ISRU). To manoeuvre the challenges this climate presents, the concept of swarming and its use of multiple collaborating agents is introduced in Section 2-2. Section 2-3 dives into foraging, a distinct area within swarming, along with its benefits, sources of inspiration from nature, and the mechanism of stigmergy.

## 2-1 Lunar Environment

The Moon has a climate unlike that of Earth. To know what challenges any lunar rover is designed for, understanding the distinct features and mechanisms within this environment is crucial. This section outlines the most important characteristics.

The surface of the Moon consists of highly local and static terrain, with little mixing between regions due to the lack of natural activity [13]. The first of two distinct regions found on the Moon is the lunar highlands, marked with reddish hues in Figure 2-1. These regions are characterized by their light colour due to high calcium levels and can be further split into high-($>6$[W%]) and low-Ti($<6$[W%]) basalts for their titanium content. The second region is the lunar mares, darker in colour and predominantly consisting of ancient basaltic lava flows. With no atmosphere or significant core activity, the regions remain mostly separated. The little mixing that does occur is attributed to particle migration caused by electrostatic and thermal interactions, and to meteoroid-impacts dispersing surface and deep material.

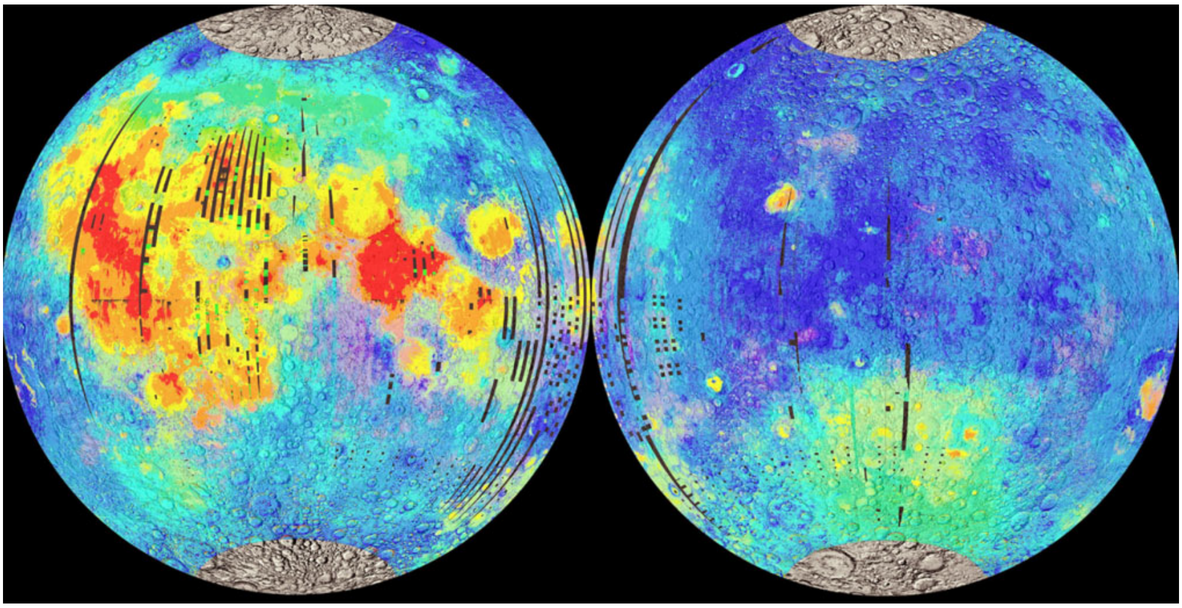


**Figure 2-1:** Distribution of regolith compositions on the lunar nearside (left) and the farside (right) based on Clementine multi-spectral imaging data. Blue: anorthositic highlands; yellow: low-Ti basalts; red: high-Ti basalts. The large yellow/greenish area in the southern hemisphere of the farside is the South Pole-Aitken Basin, where the colours mostly reflect the more Fe-rich nature of the lower crust exposed by the basin rather than basaltic material [13].

Unlike its static regions, the lunar climate poses dynamic challenges. With lunar days and nights lasting 15 $[earth - days]$, extended periods of varying solar circumstances are experienced. Temperatures can vary between 29-390 [K] [16] depending on latitude and solar cycle. Furthermore, the lack of a protective atmosphere results in increased levels of radiation being present, originating from the Sun and interstellar space [40][48]. This also causes secondary reactions, with radiation interacting with surface material forming secondary particles. Although it is widely understood that radiation is a predominant factor on the Moon, few measurements have been taken in situ to characterize it.

The surface of the Moon is made up of several layers. The surface is covered by rocks and fine-grained particles referred to as regolith [5]. The depth of this layer can vary between a couple of centimetres up to meters. The composition of regolith as mentioned previously is highly local, with meteoroid strikes, migrating particles, and radiation all affecting the present materials. For lunar rovers, regolith poses a challenge given its highly abrasive features as well as their potential to adhere electrostatically to surfaces, reducing solar array efficiencies [11][18].

A prevalent surface feature of the Moon is craters caused by meteoroid strikes. These craters are important as they not only expose the inner crust of the Moon, but also provide shielded environments that retain elements of interest. Craters can form Permanently Shadowed Regions PSR that never receive sunlight, causing their temperatures to be as low as 29[K] [5][11]. These shielded environments allow particles to collect and be trapped, with high scientific value [45].

### 2-1-1 Lunar Resources

The Moons composition holds a variety of resources. These have scientific importance, but also potentially use for permanent outposts through ISRU.

Water and ice are considered the most important elements for both science and ISRU. While water may be indicative of extraterrestrial lifeforms, it can also be used to make rocket fuel for launches of the Moon [7][18]. Along with oxygen, it can also be used to sustain human presence. Measurements of the Lunar Reconnaissance Orbiter (LRO) mission [45] indicate ice deposits in craters such as the Shackleton and within regolith, rising in abundance with increasing latitudes.

Other resources of interest in order of significance are solar wind implanted volatiles (the most common being $H,^3He,^4He,C,N,F,Cl$), metals, and silicon. Although scarce (see Table 2-1), solar wind implanted volatiles provide pure elements that can be utilised for manufacturing to create materials as needed. Additionally, Helium-3 holds the potential to fuel nuclear reactions when combined with deuterium, potentially useful for energy production. Metals and silicon found on the Moon can be used for manufacturing structures and electronics. By applying ISRU, the material having to be flown in to establish a lunar outpost is considerably decreased, motivating resource collection.

**Table 2-1:** Average concentrations of solar wind implanted volatiles and average mass in lunar regolith (assuming a bulk density of 1660 [$\mathrm{kgm}^{-3}$]) [13].

| Volatile | Concentration | Average mass per $m^3$ of regolith |
|---|---|---|
| | $[\mu g/g]$ | $[g]$ |
| H | 46±16 | 76 |
| $^3$He | 0.0042 ± 0.0034 | 0.007 |
| $^4$He | 14 ± 11.3 | 23 |
| C | 124 ± 45 | 206 |
| N | 81 ± 37 | 135 |
| F | 70 ± 47 | 116 |
| Cl | 30 ± 20 | 50 |

## 2-2 Swarm Robotics

Chapter 1 highlighted the occurring trend of using mobile rovers for lunar exploration. Although the use of a mobile rover significantly expands the potential of lunar endeavours, it introduces limitations and single points of failure. The use of several rovers further boosts capabilities of missions, while also implementing distributed responsibilities and robustness.

Swarm robotics introduces the concept of multiple autonomous individuals collaborating to solve a global objective. By working together, actors can distribute tasks, problem-solve concurrently, and perhaps most importantly do so without being affected by malfunctions of others [47]. By breaking down complex problems into particular tasks a simple robot can solve, the performance requirements of an individual are lowered. By designing the operation of an actor with collaboration in mind, Swarm Intelligence (SI) [25] is achieved. SI allows the compound behaviour of a population to solve difficult tasks even with simple individuals. An important distinction to be made is the difference between swarming and multi-agent systems. Swarming operates in a decentralized manner, where agents only have access to information shared rather than to all collected.

Using a swarm of individuals has several benefits. The three leading advantages include flexibility, robustness, and self-organization [10]. *Flexibility* comes from the fact that multiple agents can approach a problem concurrently, leading to solutions and more importantly optimal ones being found. This multiplicity also leads to *robustness*, where the failure of an individual does result in the downfall of the population. Finally, as members collaborate in a population based on the challenges of the scenario, they construct a custom framework for functioning. This *self-organization* allows a system to adapt to its surroundings and achieve its objective in an optimal manner and with full autonomy. Further advantages include *scalability* with the system being able to operate irrespective of population size. This allows the reach and scope of the system to be tailored without the need for redesign. *Low-cost* is achieved as simpler agents can be used enabling mass-production, with *energy-efficiency* a coupled result given that agents can be smaller in size and are only required to maintain simple operations.

Swarming systems also have drawbacks. Without global knowledge, agents may tend to local optima rather than global, hereby achieving sub-optimal results. Furthermore, without absolute information-sharing, agents may rediscover areas others have already visited. The simplicity of agents can only be applied with tasks that can be broken down into smaller simpler tasks. Without the possibility of this and tasks requiring highly specific operations, the use of similar individuals within a population becomes less possible hereby diminishing some advantages of a swarmed approach.

### 2-2-1 Swarm Taxonomy

The use of swarms of individual robots is motivated by its significant benefits. These allow systems to solve specific tasks while adapting to their environment. However, swarm intelligence is only achieved when the system design supports the combination of individuals' behaviour.

Swarming systems, unlike their members, require extensive definitions. Defining the requirements of an individual's behaviour is key to ensuring their combination adds up to desired global mechanisms. In defining such systems, several aspects need to be defined[9][29]. Understanding a system's environment is key to defining how the population and individuals approach problems and the nature of its objective. The collaboration and strategy as applied within the group is decisive for the flow of information and approach to a solution. Appendix A presents the taxonomy used to define swarms throughout this report.

A key factor for swarming is the size of the population. Although swarms are scalable, their size is indicative of convergence to solutions but also operational requirements. The structure of collaboration may dictate the utilization of individuals for supporting tasks, to which swarm size is imperative. The definition of environment and objectives affect the information shared within the population, which in turn dictates the possible approaches to collaboration. Furthermore, factors such as the connections of an individual, clustering, and population policy all affect how the swarm moves through the search space [26].

## 2-3 Notion of Foraging

The goal of applying a swarm of robots on the Moon is to implement their robustness and flexibility. These characteristics are especially useful when venturing into the harsh lunar environment to retrieve resources. In nature, many species rely on similar collaboration methods to achieve tasks beyond the capabilities of an individual. These mechanisms can be mimicked to achieve similar performance.

Foraging is a distinct area within swarming [42]. Heavily inspired by examples of animals collecting food as Section 2-3-1 will present, the primary objective is the manipulation of a resource, be it retrieval, utilization, or destruction. By collaborating in a swarm, a group of individuals can reliably and efficiently achieve its resource-oriented goal. This concept is especially useful in unknown environments containing a specific target. With this in mind, the applicability of foraging expands beyond physical domains.

The act of foraging is generally classified into two phases, namely

1. Exploration: exploring the environment to locate a certain target region
2. Exploitation: found targets are manipulated while constructing optimality through the shared experience of the population

In the exploration phase, the objective of the swarm is to locate the sought-after target. Members of the population engage in search algorithms where they collect information on their environment and share it with others. By collaborating, individuals can perform more directed searches. This phase may also include the establishment of a communication layer to aid the availability of collected information. The exploration phase ends once the sought-after region is located, or a good enough result is attained.

The exploitation phase is initiated once the system is done exploring. Agents start to perform their primary objective of resource manipulation. In the process, they continue sharing their findings with others: positive-outcome solutions are reinforced while unsuccessful decisions are discouraged. In doing so, the population "learns" to solve the problem in a more optimal manner.

### 2-3-1 Inspiration from Nature

The desired tendencies of foraging swarms can be achieved by a variety of architectures. Depending on the taxonomy of swarming, some solutions may be more appropriate than others. To inspire possible mechanisms, three foraging species as observed in nature are presented.

### Birds and Fish

Bird-flocks and schools of fish were one of the earliest inspirations used to mimic foraging. In nature, members of these groupings move in close proximity of one another in a directed motion without colliding [15][24][39]. Each individual animal aims to find food for itself, while maintaining the safety of a larger group. Taking both personal desires and input from others into account, a more informed decision-making is achieved.

Take a flock of birds as an example. While flying between feeding, every bird looks for likely sources of food in the fields below them. In doing so, they build personal experience on which area has the highest potential. Searching the surroundings of this region makes sense, as perhaps its neighbouring field has an even better return. Other birds do the same, compiling personal information. Once the bird finds a good enough field, it feeds. Were the birds not to take communicate, they would only have information on fields they themselves visited. By sharing information, they can consider fields others are very interested in to search.

This knowledge sharing is exemplified in Equation (2-1). The bird chooses heading vector $v_{ij}$ between feeding based on three aspects. The first term indicates the bird's willingness to not make tiring turns, with $\omega$ representing an inertial resistance to change. The next element implements the bird considering its personal experience of best location $p_{ij}$, with the position-offset $p_{ij}(k)-x_{ij}(k)$ steering towards it. Finally, $g_{ij}$ describes the best location found by others in the flock. Including this position in decision-making allows the bird fly to fields outside its own discovered area containing even better potential. By combining both personal and shared experiences, a more informed decision can be made, increasing the likelihood of finding great feeding grounds.

$$v_{ij}(k+1) = \omega v_{ij}(k) + C_1(p_{ij}(k) - x_{ij}(k)) + C_2(g_{ij}(k) - x_{ij}(k)) \tag{2-1}$$

### Ants

Ants are a classic example of foraging, as they depict a clear form of collaboration. Although an ant is small in size and capability, by working together ant colonies can cross kilometres to retrieve food or materials. This makes ant species a rich source of inspiration.

Ants start in a static and established nest [35]. From here, they explore their unknown environment in search of food. When an individual is successful in doing so, it returns with what it can carry. In the process, it simultaneously deposits pheromones on the floor, marking its path taken [14][17][22]. Other agents are attracted to this pheromone and now can follow it to the resource. With these new ants also successfully reaching the target, they reinforce the path in strength by depositing pheromones on their return. This method of leaving indicators in one's environment is an intricate mechanism further explained in Section 2-3-2. In their approach, ants develop and optimize distinct paths between regions.

Ant-like foraging is considered highly robust to changing environments, with guidance information locally available at every step. Changes in paths are therefore immediately communicated. In the event of a branch falling on the path, ants can find a path around it, with the new pheromone markers leading others around it. However, the functionality of these pheromones relies on complex mechanisms as they need to implement recruitment of others and time-based dynamics of fading. For further explanation, see Section 2-3-2.

### Honeybees

Honeybees in nature can cross several kilometres to reach their target while operating entirely on their own. Being highly efficient in their journey is key given the size of a single bee. To achieve optimality, honeybees rely heavily on mechanisms such as Path Integration (PI) [8][36].

Honeybees, similar to ants, also start at a static hive location. From here, they explore their surroundings for resource-rich areas. While traveling outbound, bees keep track of their heading and distance. Once at a target, they combine these segments to compute a direct path back to the hive. Once at the hive, they communicate this heading to others using body language, providing instructions on direction and distance. As Figure 2-2 depicts, the heading is conveyed using the Sun's location and distance using the length of the movement. Others now can follow this vector to efficiently reach the targeted patch of flowers without unnecessary deviations.

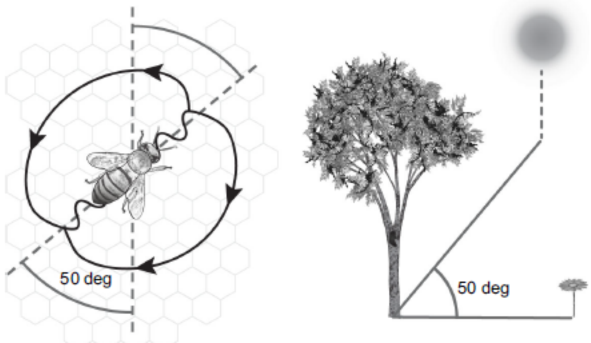



**Figure 2-2:** Representation of direction-sharing through waggle runs [8].

The power of bee-like foraging stems from PI. Individuals can follow headings with high precision as well as measure distance. Furthermore, for longer journeys or ones with obstacles, bees split the vector into shorter more precise segments. Segmented vectors utilise intermediate landmarks to re-calibrate their heading [28][12]. The downside of bee-like foraging however is the requirement of precise positioning and that path-information is only available at the hive. Any updates in path therefore take time to reach others, making the system less change-tolerant.

## 2-3-2 Stigmergy

Having access to information in decentralized systems is an intricate process. Having up-to-date guidance on paths is crucial for both the execution and optimization of processes. A reliable and strong manner of doing so is similar to what ants do by laying pheromones: they are encoding experience in their environment.

Stigmergy is a mechanism achieving both indirect and mediated communication using one's environment [21]. By leaving indicators of success of an action, subsequent ones are triggered. In simple terms, the success of an action is encoded in the environment to motivate others to perform follow-up tasks. In the case of ants, this is implemented by ants leaving pheromones for others to follow as illustrated inFigure 2-3. With the pheromone's recruiting feature, others are triggered to follow a specific path to the resource. Similarly, negative success is also implemented, with depleted targets not leading to reinforcement of trails which in turn fade. The intensity of pheromones is therefore an indication of success potential, with stronger pheromone trails considered better.

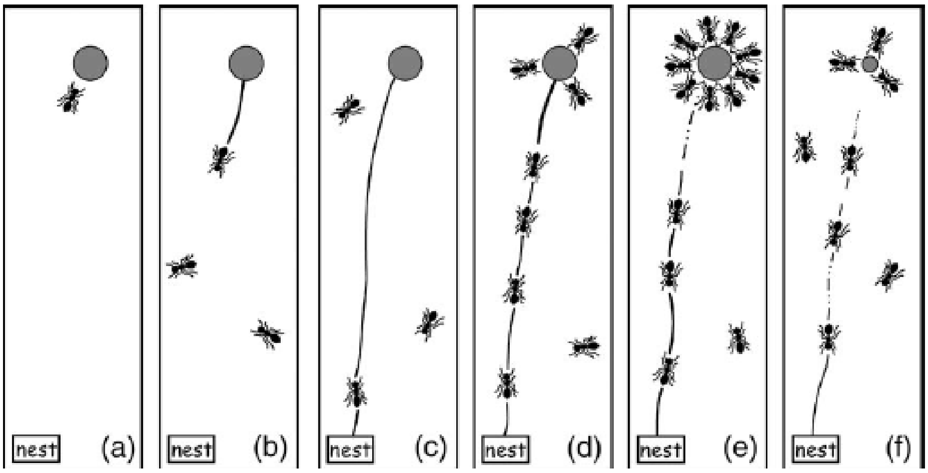



**Figure 2-3:** Outline of the life of a foraging trail [35]. a) A worker randomly finds a resource in its exploration phase. b) The worker retraces its path, laying a pheromone trail in the process c) In the nest, other workers are recruited to follow the trail to the target d) As more workers use the trail, more other workers are encouraged to follow the trail e) As the resource is depleted, unsuccessful workers will not reinforce the trail f) The trail fades and new targets are looked for.

The first attribute of stigmergy is the possibility of conveying local knowledge indirectly, without the need for both agents to be present within the communication range. This allows for the second aspect of stigmergy, namely that local information is a compound result of multiple inputs. Serving as a local memory of recent visits, the information stored can stem from multiple individuals allowing for a more well-informed and optimized marker. As both positive and negative knowledge is encoded in the indicator, the evolution of paths is also enabled, implementing a factor of flexibility.

### Mimicking Stigmergy

Each application of stigmergy as encountered in nature developed over the course of thousands of years and fit for a single objective. Implementing these exact mechanisms in engineering solutions is therefore not possible given the change in environment and tasks. However, by repurposing their methodology, their utility can be reproduced.

Achieving effective stigmergy involves emulating multiple features. Markers should be implantable in the environment to achieve local availability of past experience. Markers should also have a quality with dimension, conveying levels of success. This measure of potential should be dynamic to represent both positive and negative findings and allow paths to fade. Furthermore, markers should be detectable to allow others to extract their information. Achieving these mechanisms can be done in many ways, depending on the application and coming at varying costs to the system.

The most rudimentary way of achieving local markers is by using the physical presence of individuals as markers. [31] and [37] describe networks where individuals follow the outline of interconnected branches of other members of the population. Depositing material indicators such as alcohol [33] or other substances can also be used, resembling the laying of pheromones. [41] proposes the use of more fundamental resources such as the heat trail left behind by an individual traveling a path. Others propose the use of memory dumps laid in the environment in the form of Radio Frequency Identification (RFID)-nodes [32], where the stored information is updated by individuals passing by. While previous methods propose the use of finite resources, [2][17][22] propose the use of members of the population as beacons in key locations. Similar to RFID-nodes, a beacon's information is updated by passers-by. However, beacon agents can position themselves dynamically in key locations and can also rejoin the foraging population when their path is deemed useless.

### 2-3-3 Selection of System Framework

Having presented the possibility of using swarming systems more specifically its foraging subset, for lunar exploration and exploitation, it is deemed a viable option to explore. With its advantages of robustness, flexibility, and self-organizing, it holds valuable potential. To narrow down the aspects of a possible system a certain framework is selected.

Taking inspiration from foraging in nature, the main notions of the three animal-like foraging can be weighed. Given the goal of supporting a static lunar outpost as described in Chapter 1, a static base of operation is assumed. Both honeybee- and ant-like systems implement such a region, whereas bird- and fish-like concepts do not. Furthermore, both former systems implement levels of information legacy. While bee-like systems do so only at the hive, ant-like setups do so in the field as well. Furthermore, honeybees rely heavily on precise monitoring of their location for PI-purposes, adding a straining requirement for operation. Ant-like foraging, although with its benefits, requires precise stigmergy replication with possibly resource-costly mechanisms. With the latter deemed possible through intricate system design, ant-like foraging will be considered from here on out.

The approach to replicating stigmergy can vary from simplistic approaches to more intricate ones, each with its difficulties and costs. To select a method considered, Table 2-2 presents a trade-off between previously mentioned types of solutions. Each method is evaluated for five aspects. First, the cost of the implementation in terms of population utility is considered. Systems reducing the number of foraging members are considered undesired. Next, the sustainability of the method is expressed, where methods using finite resources are deemed short-term. Tunable dynamics of markers are considered beneficial, as these affect the collaboration achieved within the swarm. The detectability of markers is an important aspect to ease the availability of information. Finally, the robustness of markers is considered, given that loss of path-marking can have major implications for population integrity.

The five concepts considered to reproduce foraging are as follows. Member-link systems consider the use of members of the population as markers similar to that in [31]. Material approach refers to a marker material being deposited on the path. The physical method proposes the use of natural markers such as vehicle heat or walking trails left behind. Passive beacons refer to memory storages deposited by individuals where the device is immobile. Finally, active beacons refer to beacons acting as a communication platform with the possibility of disengaging this role when deemed futile.

**Table 2-2:** Trade-off of approaches to stigmergy reproduction.

| | Population cost | Sustainability | Marker dynamics | Detectability | Robustness |
|---|---|---|---|---|---|
| Member-link | - - | ++ | - | ++ | + |
| Material | ++ | - | - | - | . |
| Physical | ++ | ++ | - - | + | - |
| Passive beacon | ++ | - | ++ | + | - |
| Active beacon | - | ++ | ++ | + | + |

From Table 2-2, the approach of using active beacons is selected primarily for its sustainability and tunable marker-dynamics. Although being individual-costly, individuals can rejoin the population in an active role, restoring swarm size. Although member-links perform well in sustainability for similar reasons and outperform all others in terms of detectability, the sheer cost to population size disqualifies it. Materialistic markers fare poorly in sustainability as the material used is finite, meaning it will run out unless sourced locally. Physical markers are disqualified for the reason their dynamics cannot be tuned to fit the application. Although passive beacons perform similarly to active beacons, their inability to self-maintain worsens their robustness.

With this, a foraging system implementing active beaconing is selected for consideration. In this foraging system, robots venture out into the environment in search of their target. At key points, individuals assume a beacon role supporting the guidance of other foragers. The internal logic of beacons can dictate marker-dynamics. Finally, individuals can return to the foraging population once deemed unnecessary to uphold the path.

## 2-4 Summary

This chapter introduced both the environment faced when considering lunar exploration as well as the concept of swarmed foraging to approach the problem of lunar exploitation. Having been presented with possible implementations of foraging in application, an ant-like setup was selected.

Section 2-1 discussed the environment as encountered on the Moon. The lunar surface consists of highly localized and changing regions, where conditions can differ drastically. The surface is characterized by layers, with the top regolith consisting of loose rocks and sand. Robots need to be able to handle challenges posed by the abrasive surface, changing and extreme temperatures, and radiation climate. Resources of interest primarily include ice and oxygen, likely to be found in Permanently Shadowed Region (PSR)s.

A viable option for supporting lunar outposts is the use of swarms of robots. By distributing tasks within the population, these systems implement advantages in the form of robustness, flexibility, and self-organization, among others. Given the resource-oriented necessities of maintaining a lunar base, foraging is proposed for its resource-oriented approach.

To select the type of foraging used, three types inspired by nature were weighed. Of these three, an ant-like approach was used with the benefits of stigmergy. To do so, weighing multiple options, an approach using members of the population as landmarks was selected for its sustainability and marker dynamics. Throughout this report, a system of this type will be considered.

# Chapter 3

# Problem Statement

Throughout the process of familiarizing with lunar missions and foraging systems as a possible solution to challenges, several critical shortcomings were found. While foraging does present features highly beneficial for lunar exploitation, it lacks maturity in application.

Section 3-1 identifies that theoretical proposals until this point failed to implement mechanisms to sustain robotic operations. Section 3-2 then raises the question if foraging is achievable when combined with system maintenance tasks. To provide the research involved in answering this question, Section 3-3 provides basic definitions used throughout this report, Section 3-4 formally defines the foraging problem, and Section 3-5 limits the scope with predefined assumptions. Finally, Section 3-6 introduces the structure used.

## 3-1 Problem Formulation

A fundamental element of robotic systems in application is that operation costs energy. Activities such as the actuation of wheels, performing measurements, processing data for decision-making, and as simple as communicating with others all use power. When considering mobile agents, the amount of available onboard energy is limited to battery size. To enable sustained operation beyond the capacity of a single battery, strategies are employed to maintain ample energy levels at all times.

In volatile and harsh environments such as that of the Moon, maintaining charge levels becomes a challenge. With long-lasting dark periods and extremely low temperatures, energy is a scarce resource. Replenishing energy levels may prove to be difficult given environmental and time limitations. As a result, the lunar terrain underlines the need for power strategies even further.

Throughout the preliminary literature survey, a handful of sources were found implementing foraging in simulated environments. Although some voiced the importance of fundamental system upkeep, none had specific mechanisms in place. Robots were either assumed to have access to infinite energy, travel without power expenditure, or be implemented on homogeneous and idealized environments. In doing so, while most proved the theoretical potential of foraging strategies, all failed to provide a system fit for implementation. This is the unknown this report aims to address.

## 3-2 Research Question

This research focuses on the applicability of foraging swarms with respect to fundamental system requirements to maintain sustained operation. Specifically, this report focuses on the energy needs of physical world systems. With energy being a core requirement for active systems, its availability and harvesting are critical to any form of foraging.

With the previously discussed in mind, the following main research question is to be answered:

***Can an applied energetic housekeeping strategy be developed for a beaconed ant-inspired foraging algorithm with the intent of resource gathering on the lunar surface?***

As this is a compound question, it is broken down into several secondary questions:

- *What global tendencies are required to enable system maintenance from an energetics perspective?*
- *How should an individual's operation logic be structured to include personal maintenance operations while contributing to population objectives?*
- *How do optimality mechanisms within foraging affect population and individual operation with respect to energetics?*

Answering these sub-questions will help determine if the problem of sustained foraging and energetics as addressed by the research question is achievable and implementable. In doing so, the applicability of foraging swarms in real-life usage can be proven and developed further.

## 3-3 Nomenclature

To provide the reader with a set of terms used throughout this report, consult the following list of definitions:

- Energetics: a branch of mechanics that deals primarily with energy and its transformations[1]
- Agents: autonomous individuals partaking in a swarming system
- Beacon: an agent with an assumed beacon role fulfilling guidance network tasks
- Outbound: a motion with an objective to reaching the target region $\mathcal{T}$
- Inbound: a motion with an objective to reaching the nest region $\mathcal{N}$

[1] https://www.merriam-webster.com/dictionary/energetics

- Origin: the region where an agent started its motion from
- Destination: the region the agent is attempting to reach
- Obstacle: a region within the environment with undesired characteristics or an impenetrable boundary
- Region of interest: a region within the search-space an agent may wish to visit

Furthermore, specific notations are used throughout this report.

- Sets are indicated with the calligraphic notation: $\mathcal{A}$
- Scalars are indicated with regular letters: $a \in \mathbb{R}$
- Foraging roles are indicated with a capital $F$ and subscript defining the type: $F_{out}$
- Vectors are indicated with an overhead arrow: $\overrightarrow{a}$
- The Euclidean norm of vectors is indicated as follows: $||\overrightarrow{a}||$ is used
- The normalized form of a vector $\overrightarrow{a}$ is denoted as $\overrightarrow{\hat{a}}$ and is described as

$$\overrightarrow{\hat{a}} := \langle\overrightarrow{a}\rangle := \frac{\overrightarrow{a}}{||\overrightarrow{a}||} \tag{3-1}$$

## 3-4 Formal Definition of the Foraging Problem

The basic foraging problem to be tackled is defined as the following:
Consider discrete time dynamics $k \in \mathbb{N}$ with sampling time $\tau \in \mathbb{R}_+$ and system update time $\mathrm{T} \in \mathbb{R}_+$. An agent may self-trigger intermediate updates given its state. Equation (3-2) describes the assumed discrete time progression.

$$t = k \cdot \tau \tag{3-2}$$

Furthermore, a swarm consisting of $N$ agents $\mathcal{A} = \{1, 2, ..., N\}$ is placed within a bounded domain $\mathcal{D} \in \mathbb{R}^2$.

Each agent $a \in \mathcal{A}$ has

- a role $s_a(k) \in [outbound, inbound, recharge, beacon]$
- a position $x_a(k) \in \mathcal{D}$
- a heading $\theta_a(k) \in [-\pi, \pi\}$
- a velocity vector $\overrightarrow{v_a}(k) = v_0 \left(\cos\theta_a(k), \sin\theta_a(k)\right)^T \in \mathbb{R}^2$ with standard velocity $v_0$
- a battery State-of-Charge (SOC) of $SOC_a(k) \in [0, .., 1]$
- and resources onboard $r_a(k) \in [0, .., r_{max.resource}]$

The dynamics of an agent with respect to movement are defined to be

$$x_a(k+1) = x_a(k) + v_a(k)\tau \tag{3-3}$$

Within the domain $\mathcal{D}$, four static sub-regions are defined, namely nest $\mathcal{N}$, target $\mathcal{T}$, recharging region $\mathcal{R}$, each with radius $\delta_{\mathcal{N}}, \delta_{\mathcal{T}}, \delta_{\mathcal{R}}$ respectively. Region $\mathcal{N}$ has a resource-amount $r_{\mathcal{N}}(k) \in$

$[0, .., r_{resource,available}]$ and $\mathcal{T}$ a resource-amount of $r_{\mathcal{T}}(k) \in [0, .., r_{resource,available}]$. Region $\mathcal{R}$ is defined as a region with constant solar irradiation $Irr_{\mathcal{R},solar} > \eta_{solar}$.

Furthermore, an additional region may be defined, namely that of an obstacle $\mathcal{O}$. An obstacle region is defined as a rectangle with width $w_{\mathcal{O}}$ and length $l_{\mathcal{O}}$. An obstacle's height is assumed to be infinite for hard obstacles and infinitely small for soft obstacles. Soft obstacles have an additional friction coefficient parameter $\xi_{\mathcal{O}}$. For an example of the described domain $\mathcal{D}$ with as described, consult Figure 3-1.

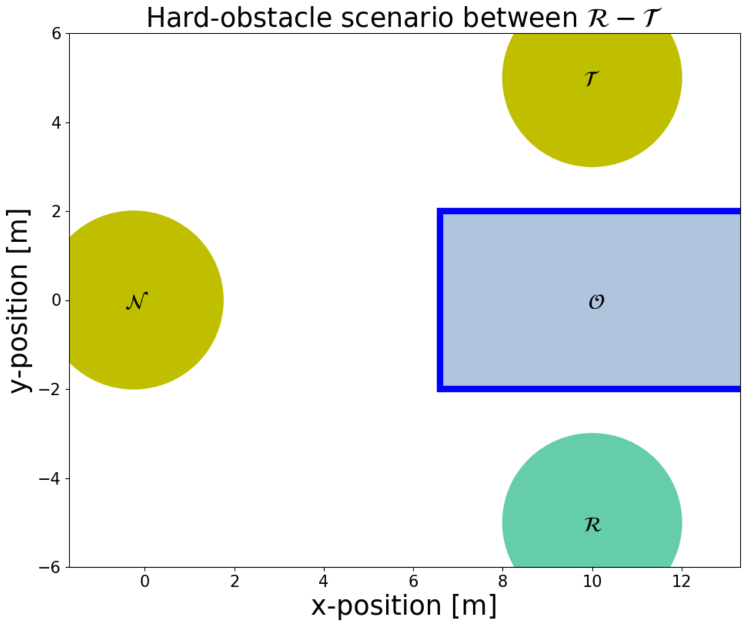



**Figure 3-1:** Example scenario with regions $\mathcal{N}, \mathcal{T}, \mathcal{R}$ and hard obstacle $\mathcal{O}$ on the $\mathcal{R}-\mathcal{T}$ trajectory.

Finally, two additional sub-regions within domain $\mathcal{D}$ are defined. Each agent $a$ is assumed to be able to exchange information within its communication region $\mathcal{C}_a$ with radius $\delta_{comm}$. The union of all regions $\mathcal{C}_a$ forms the guidance network region $\mathcal{G} := \bigcup_{a \in A} C_a$ within which mobile agents can communicate with at least one beacon $b$.

For energetics considerations, additional dynamics are introduced. These correlations are described in Section 5-1.

The objective of the swarm is to recover as many resource units from target $\mathcal{T}$ and return it to nest $\mathcal{N}$, whilst maintaining an agent SOC of $SOC_a(k) > 0 \; \forall a \in \mathcal{A}$. To do so, the agent will have to explore the domain $\mathcal{D}$ in search for both $\mathcal{T}$ and $\mathcal{R}$. As agents fulfilling the role of a beacon are stationary and do not directly contribute to the retrieval of resources, the goal statement of the foraging system as a whole is broadened to be:

*Establish a (semi) optimal route along which retrieval of resources from $\mathcal{T}$ to $\mathcal{N}$ is possible, whilst maintaining a (semi) optimal route to $\mathcal{R}$ from all positions encompassed by the guidance network $\mathcal{G}$.*

## 3-5 Problem Assumptions

This research is a study into foraging swarms in application. For this, as stated previously, the system proposed in [1] is assumed as a baseline as presented in Chapter 4. The basic notion and concepts are transferred and assumed to be known, even if not implemented.

Furthermore, fundamental assumptions are made on the system and environment considered. This limits the scope of the study. The following is assumed to be known or to hold:

- ASU.1 All agents have equal capabilities and dynamics, meaning a homogeneous system is used
- ASU.2 All agents have memory capable of storing both operational and navigational values and parameters
- ASU.3 All agents have computational power to perform scalar and vector (in $\mathcal{R}^2$) operations, including summation and multiplication
- ASU.4 All agents lack the computational power to precisely integrate path or determine global location
- ASU.5 All agents have communication capabilities for transferring a message consisting of up to seven scalars, three vectors, and a string
- ASU.6 All agents have communication capabilities for transferring messages in full without error within a range of $\delta_{comm}$
- ASU.7 All agents have omnidirectional communication capabilities
- ASU.8 All agents lack the ability to determine signal strength or direction of communication pathways
- ASU.9 All agents have moving capabilities of holonomic type
- ASU.10 All agents have a measure of global angular orientation
- ASU.11 All agents have a measure of angular rotation of their wheels
- ASU.12 All agents are equipped with a finite-capacity battery
- ASU.13 All agent operations requiring energy are supplied from the onboard battery
- ASU.14 All agents have solar panels and Battery Management System (BMS) capable of recharging the onboard battery
- ASU.15 All agents have a BMS capable of measuring battery SOC and power input from solar panels
- ASU.16 All agents have sensing abilities for nest $\mathcal{N}$ and target $\mathcal{T}$ regions
- ASU.17 All agents have onboard resource storage compartments capable of retaining resources while in operation
- ASU.18 All agents have resource-handling abilities to load and unload the onboard resource storage compartment

## 3-6 Report Structure

This report presents the mechanisms and findings achieved while developing a foraging system fit for applied resource retrieval and energy-level maintenance. This report consists of several sections, each aimed at presenting specific elements of the system or results achieved by it.

This report started with Chapter 1, providing the incentive that triggered this research. Next, Chapter 2 provided background information into the environment considered and foraging concepts used throughout this report. Chapter 3 has now discussed the identified problem of energetics in applied foraging systems, along with a formal definition and assumptions bounding the scope of research. Subsequently, Chapter 4 introduces concepts and a baseline system upon which the newly developed system relies. Chapter 5 then presents the proposed system designed to tackle the problem of energy maintenance of a foraging system. Chapter 6 describes how the system is tested for performance and results are quantitatively evaluated. From quantitative indicators, qualitative conclusions are made. Chapter 7 provides a conclusive summary of the findings in this research, answers to the research questions, and proposes recommendations for future work.

Chapter 4

# Foraging Baseline

Foraging systems utilise the benefits of swarm systems to achieve their resource-oriented objectives. Global tendencies of the population can be achieved by careful design of individual behaviour. As this research aims to study the applicability of foraging, a previously developed theoretical system as found in [1] is used as a baseline. This chapter introduces this system along with fundamental concepts.

Section 4-1 introduces concepts important for operation and notions encoded in logic. Next, Section 4-2 presents the system of [1] in its operation and structure. This work will be modified in Chapter 5 to expand the system's applicability.

## 4-1 Concepts used in Foraging Logic

To understand how foraging systems function, several concepts are introduced. These describe the utility of foraging as applied to the specific problem of resource-retrieval. The concepts of roles, region of interaction, and collected experience will be used specifically or as motivation for agent behaviour.

### 4-1-1 Roles

Every agent within the foraging population executes a set of tasks. These tasks are motivated by either an agent's personal maintenance or a global objective. Each role an agent can assume dictates a behavioural policy. The role of an agent is selected based on its state and past experience. The following three roles are considered fundamental in theoretical foraging.

- *Outbound role* $F_{out}$: outbound foraging agents $F_{out}$ attempt to travel to the target region $\mathcal{T}$ from their current location
- *Inbound role* $F_{in}$: inbound foraging agents $F_{in}$ attempt to travel to the nest region $\mathcal{N}$ from their current location

- *Beacon role $B$*: beacon agents $B$ perform guidance network tasks while remaining static. Beacons receive input from foragers, compute guidance indicators, and relay these to passing agents

Each role has a specific goal to achieve. Based on this, each role can have a different behavioural policy, affecting agent decision-making and actions even under the same circumstance.

From here on out, the role of an agent may be used to refer to the agent assuming said role. This implies that $x_{F_{out}}$ refers to the location of an agent with its current role $F_{out}$, or formally defined as $x_{F_{out}} := \{x_a \in \mathcal{D} | a \in \mathcal{A}, s_a(k) = F_{out}\}$. Similarly, internal values are also denoted by an agent's role, where $\omega_b$ denotes the internal weight-value of an agent with role $b$.

### 4-1-2 Region of Interaction

Agents within the foraging system have a communication region $\mathcal{C}_a$ within which they can interact with other agents. Roles $F_{out}$ and $F_{in}$ are referred to as mobile roles as agents are moving around in $\mathcal{D}$. Agents assuming mobile roles can be collected as a set $\mathcal{M} := \{a \in \mathcal{A} | s_a(k) \in \{F_{out}, F_{in}\}\}$. The role $B$ on the other hand keeps agents static, with beacon agents $\mathcal{B} := \{a \in \mathcal{A} | s_a(k) = B\}$.

Mobile roles interact with a set of beacons $\mathcal{B}_m(k)$ within their $\mathcal{C}_m(k)$. These beacons provide the mobile agent with guidance vectors to guide the agent towards its destination. Beacons with whom the mobile agent $m$ is in contact with is defined as the set Equation (4-1).

$$\mathcal{B}_m(k) := \{b \in \mathcal{B} | x_b(k) \in \mathcal{C}_m(k)\} \tag{4-1}$$

Beacons on the other hand interact with passing mobile agents $\mathcal{M}_b(k)$ within their own $\mathcal{C}_b(k)$. These mobile agents are reliant on the beacon for guidance, and can be collected as in Equation (4-2).

$$\mathcal{M}_b(k) := \{m \in \mathcal{M} | x_m(k) \in \mathcal{C}_b(k)\} \tag{4-2}$$

### 4-1-3 Collected Experience

As described in Chapter 2, one of the advantages of swarming systems is that they have a level of self-organization. They can construct entire guidance networks for their application by compounding agent experience. The concept of inherent experience is defined below.

Take the example of a system with one beacon, one mobile agent, and two nodes $N_1, N_2$ as depicted in Figure 4-1. The mobile agent $m$ initiates at node $N_1$ and assumes a role $s_{N_1 \to N_2}$ with intended destination $N_2$. The agent's role indicates that it is coming from $N_1$, or in other words that it "experienced" the node's location. As the agent is moving away from node $N_1$, the direction of this node is encoded as the negative velocity of the agent. This "experience" can be collected by beacons to develop guiding vectors. Upon $m$ completing its role $s_{N_1 \to N_2}$ and initiating role $s_{N_2 \to N_1}$ at $N_2$, the beacon provides $-\overrightarrow{v}_{s_{N_1 \to N_2}}$ as guidance towards $N_1$.

The beacon simultaneously uses the available agent vector to update the guidance vector for role $s_{N_1 \to N_2}$, being $-\overrightarrow{v}_{s_{N_2 \to N_1}}$.

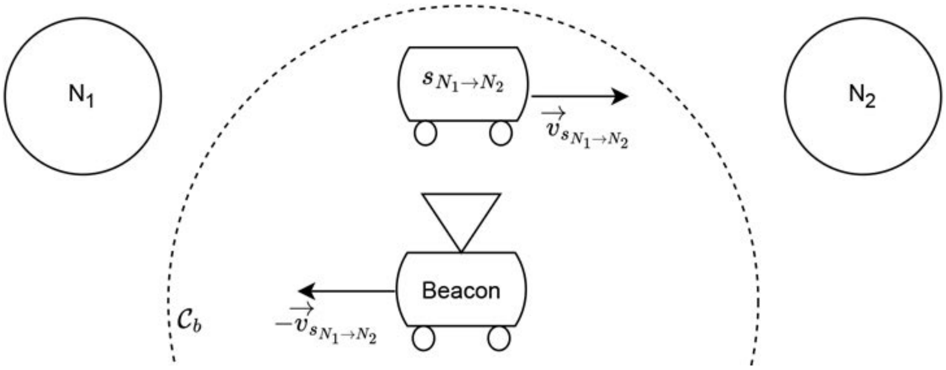



**Figure 4-1:** Example scenario of collected experience.

As touched upon in Section 2-3-2, stigmergy also requires a dimension to indicate success potential. As a requirement, this measure should allow agents to distinguish between successful and unsuccessful paths, as well as allow for paths to fade over time if not reinforced.

A possible indicator of path potential can be a success boolean. If a path leads to a target, a reward is attributed to all guidance vectors along the path. However, an absolute measure of success is undesirable as this does not allow distinguishing between paths in terms of optimality. Another possible value that can be used as path potential is the frequency a path is travelled. Successful paths are travelled more often as they have a definite end-point, namely the destination, whereas unsuccessful paths continue endlessly. The use of frequency also has the benefit of inherent optimization: shorter paths are travelled more frequently, meaning they are reinforced more compared to longer paths. Agents choosing paths with higher potential automatically choose the more optimal path with respect to travel time. The choice of success potential may vary for each application

## 4-2 Description of Baseline System

As stated previously, the foraging behaviour of a swarm can be achieved in various ways. To study the possibility of developing a system fit for application, the system of [1] is taken as an initial framework to develop. This section will present the core functionality of the system along with the dynamics of agents on which Chapter 5 builds.

### 4-2-1 Roles

The objective of the baseline system is defined to construct a guidance network facilitating agent travel between a nest and a target. To do so, the system employs agents in three roles: beacon $B$, outbound foragers $F_{out}$, and inbound foragers $F_{in}$. Agents switch between roles depending on their current role and environment. The switching is performed according to Equation (4-3). This switching states that an agent becomes a beacon when it no longer receives guidance from a beacon. Agents assume the outbound role when they are inbound and have successfully reached the nest. Similarly, agents become outbound when they successfully reach the target in their outbound role.

$$s_a(k+1) = \begin{cases} B & \text{if} \quad \mathcal{B}_a = \emptyset \\ F_{out} & \text{if} \quad s_a(k) = F_{in} \wedge x_a(k) \in \mathcal{N} \\ F_{in} & \text{if} \quad s_a(k) = F_{out} \wedge x_a(k) \in \mathcal{T} \\ s_a(k) & \text{else} \end{cases} \qquad \forall a \in \mathcal{A} \tag{4-3}$$

Additionally, [1] proposes the possibility of beacon decay. A beacon's potential weight decreases with its paths not being used. Weights dropping below a decay-threshold $\eta_w \in \mathbb{R}_+$ indicates that the path is no longer of use and the agent best supports other tasks such as foraging or exploration. Decayed beacons assume their role $\overline{s}_a$ before becoming a beacon (formally defined in Equation (4-4)) and perform a random search to rejoin the guidance network. In doing so, unused beacons are released back into the population as mobile agents improving agent utilization.

$$s_a^-(k) = s_a(k^-) \qquad \text{with} \qquad k^- := \max(k_{switch} | k_{switch} \in \mathbb{N}, k_{switch} < k, s_a(k) \neq s_a(k_{switch})) \tag{4-4}$$

This addition in functionality resulted in the switching as seen in Equation (4-5) and visualized in Figure 4-2. The added guards state that beacons whose weights fall below a "decayed"-threshold and have no active agents within their communication region revert to their last active role. Decayed beacons can only become new beacons if two time steps have passed since.

$$s_a(k+1) = \begin{cases} B & \text{if} \quad \mathcal{B}_a = \emptyset \wedge s_a(k-2) \in \{F_{out}, F_{in}\} \\ F_{out} & \text{if} \quad s_a(k) = F_{in} \wedge x_a(k) \in \mathcal{N} \quad \vee \\ & \quad\quad s_a(k) = B \wedge s_a^-(k) = F_{out} \wedge \sum_s w_b^s(k) < \eta_w \wedge \mathcal{M}_b(k) = \emptyset \\ F_{in} & \text{if} \quad s_a(k) = F_{out} \wedge x_a(k) \in \mathcal{T} \quad \vee \\ & \quad\quad s_a(k) = B \wedge s_a^-(k) = F_{in} \wedge \sum_s w_b^s(k) < \eta_w \wedge \mathcal{M}_b(k) = \emptyset \\ s_a(k) & \text{else} \end{cases} \qquad \forall a \in \mathcal{A} \tag{4-5}$$

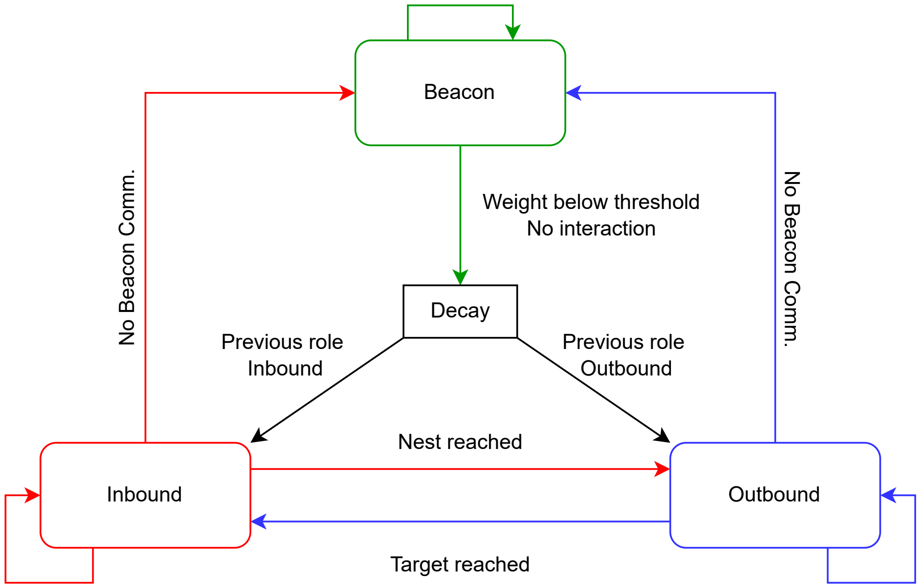



**Figure 4-2:** Role switching with decay as per [1].

### 4-2-2 Knowledge Dynamics

The foraging system can construct a vector-field of guidance vectors using the collected experience as described in Section 4-1-3. For this, the baseline system uses two guidance weight-vector pairings, one for each mobile role. In doing so, a complete guidance network is attained for agent coordination.

Beacons store two guidance values for each role, similar to what is presented in Section 4-1-3. Here, potential weight $\omega_b^s(k) \in \mathbb{R}_+$ combines both success boolean and frequency of path-travel. Vector $\vec{v_b}^s(k) \in \mathbb{R}^2$ is derived from a combination of current and past agent velocity vectors. Both values initialize to zero and change depending on mobile agent input. Beacons transmit these guidance values in their vicinity to any passerby.

Mobile agents receive both path weights and vectors from their surrounding beacons $\mathcal{B}_m(k)$. At every update time step, each agent computes a potential reward for its current state with respect to its role's destination. When an agent is at its role's origin, it assumes a reward value $\gamma^s(k) \in \mathbb{R}_+$ as described in Equation (4-6) with static reward value $r \in \mathbb{R}_+$. This can be interpreted as the agent being rewarded for completing its previous role when starting a new one.

$$\gamma^s(k) = \begin{cases} r & \text{if} \quad s_a(k) = F_{out} \wedge x_a(k) \in \mathcal{N} \\ r & \text{if} \quad s_a(k) = F_{in} \wedge x_a(k) \in \mathcal{T} \\ 0 & \text{else} \end{cases} \tag{4-6}$$

Furthermore, agents also describe their current potential given their guidance environment. Being close to a beacon on the verge of a destination region indicates positive potential. As such, the potential of a mobile agent also includes the maximum of its received guidance weights. The sum of the situational potential and possible reward as per Equation (4-7) becomes the current state-potential the agent transmits to its surrounding beacons. Here, $\lambda \in [0,1]$ represents a diffusion rate.

$$\Delta_m^s(k) = \gamma^s(k) + \lambda \max_{b \in \mathcal{B}_m(k)} (\omega_b^s(k)) \tag{4-7}$$

Beacons receive two values from each agent, namely their state-potential and current velocity vector. Beacons use the encoded experience of agents to update their guidance values. The potential weight of a beacon is updated using Equation (4-8) where $\rho_w \in \mathbb{R}_+$ is the weight evaporation rate. Changing the evaporation rate tunes the dynamics of the system, with an increase resulting in faster learning of success but also quicker fading of paths.

$$\omega_b^s(k+1) = (1-\rho_w)\omega_b^s(k) + \rho_w \frac{\sum_{m \in \mathcal{M}_b^s(k)} \Delta_m^s(k)}{|\mathcal{M}_b^s(k)|} \tag{4-8}$$

Received potential rewards are averaged over the number of updates provided. This also implies that values are only updated according to Equation (4-8) when at least one update is received, $|\mathcal{M}_b^s(k)| \geq 1$. In other cases, the weight of a beacon only experiences decay as described by the first half of Equation (4-8).

Similarly, the update of guiding vectors is performed using vector update weight $\rho_v \in \mathbb{R}_+$. This value describes the willingness of a beacon to learn from the latest experience transmitted by walking agents.

$$\overrightarrow{v}_b^s(k+1) = (1-\rho_v)\overrightarrow{v}_b^s(k) + \rho_v \frac{\sum_{m \in \mathcal{M}_b^s(k)} -\overrightarrow{v}_m(k)}{|\mathcal{M}_b^s(k)|} \tag{4-9}$$

Note the negative vector used to update the guidance vector. Consider again Section 4-1-3 explaining how experience is gathered through an agent's past: moving towards a destination node implies leaving a starting node. With this in mind, potential rewards and velocities of outbound foragers describe their motion away from the nest, and vice versa.

### 4-2-3 Motion Dynamics

The motion of an agent is dictated by its state. Depending on its connections with beacons, values received, its current and past role, or its vicinity to an obstacle, the movement of a mobile agent may differ. The following scenarios describe each case.

#### Following an Established Path

When following an established path, mobile agents receive non-zero guidance weights and vectors. A mobile receives a set of potential weights and guidance vectors from each of its surrounding beacons $b \in \mathcal{B}_m(k)$. By taking the normalized weighted sum of all mobile opposite-role $\overline{s}_m(k) \in [F_{out}, F_{in}]$ values (Equation (4-10), Equation (4-11)), a guidance vector is attained.

$$\overline{s}_m(k) := \begin{cases} F_{in} & \text{if} \quad s_m(k) = F_{out} \\ F_{out} & \text{if} \quad s_m(k) = F_{in} \end{cases} \tag{4-10}$$

$$\overrightarrow{\hat{v}_f}^s(k) := \left\langle \sum_{b \in \mathcal{B}_m(k)} \omega_b^{\bar{s}_m(k)}(k) \overrightarrow{v}_b^{\bar{s}_m(k)}(k) \right\rangle \tag{4-11}$$

At each update instance, a mobile agent decides between following the guidance vector $\overrightarrow{\hat{v}}_f^s(k)$ and engaging in Brownian random search from its current position. This decision is made stochastically, with an exploration probability $\varepsilon \in (0, 1)$. When opting to perform a random-search, the agent selects a heading offset $\theta_{random} \in [-\pi, \pi\}$ at random and computes its new heading accordingly. This decision is described in Equation (4-12).

$$\begin{aligned} &Pr\{\overrightarrow{v}_f(k+1) = \overrightarrow{\hat{v}}_f^s(k)\} = 1 - \varepsilon \\ &Pr\{\overrightarrow{v}_f(k+1) = v_0 \left[\cos(\theta_a(k) + \theta_{random}), \sin(\theta_a(k) + \theta_{random})^T\right]\} = \varepsilon \end{aligned} \tag{4-12}$$

### Exploring

In the event a mobile agent enters a region where no established paths are available, it receives zero values from all its neighbouring beacons. This indicates to the agent that further exploration is required. This is achieved by always performing random searches, equivalent to the case $\varepsilon = 1$ for Equation (4-12).

### Reaching a Destination

Additionally, when an agent reaches its role's destination, it performs a 180° rotation as dictated by Equation (4-13). This is to ensure that the reverse-velocity of the agent points towards the reached region.

$$\overrightarrow{v}_f(k+1) = -\overrightarrow{v}_f(k) \qquad \text{if } s_m(k+1) \neq s_m(k) \tag{4-13}$$

### Beacon Decay

Beacons decay when the sum of their potential-weights drops below a threshold $\eta_w$ and there are no mobile agents to service $\mathcal{M}_b(k) = \emptyset$. Upon decaying, the agent disengages from its beacon role and assumes its previous non-beacon role, as per Equation (4-5). The agent initializes in random-search until re-establishing a connection with the guidance network.

$$\overrightarrow{v}_f(k+1) = v_0 \left[\cos(\theta_a(k) + \theta_{random}), \sin(\theta_a(k) + \theta_{random})^T\right] \tag{4-14}$$

### Object Avoidance

To ensure agent safety, a rudimentary obstacle avoidance algorithm is implemented. This system uses infrared sensors to determine separation from surrounding objects. When an offset violation is determined, the agent changes its heading with angle $\theta_{avoidance} \in [-\pi, \pi\}$ based on where the obstacle is perceived. This feature is based on the functionality of the Elisa-3 robot.

$$\overrightarrow{v}_f(k+1) = v_0 \left[\cos(\theta_a(k+1) + \theta_{avoidance}), \sin(\theta_a(k+1) + \theta_{avoidance})^T\right] \tag{4-15}$$

### 4-2-4 Summary of Baseline System

To summarize the previously described baseline system, this section presents pseudocode illustrating steps within the framework. This representation of system logic is split between mobile agents and beacons, as provided in Algorithm 1 and Algorithm 2.

Beacons perform their guidance network tasks using the logic as described in Algorithm 1. In doing so, they gather and update their knowledge using mobile agent experience. At each update, new values are computed, while between updates the beacon receives input.

**Algorithm 1** Baseline Beacon Behaviour

**while** $s_a(k) = "Beacon"$ **do**
  Broadcast $\omega_b^s(k)$, $\overrightarrow{v}_b^s(k)$ for all roles
  Listen for mobile agent input for T seconds
  **for** each role $s \in ["F_{out}", "F_{in}"]$ **do**
    Compute $\omega_b^s(k+1)$ as per Equation (4-8)
    Compute $\overrightarrow{v}_b^s(k+1)$ as per Equation (4-9)
  **end for**
  Check possible beacon decay as per Equation (4-5)
**end while**

Mobile agents on the other hand provide past experience encoded in their velocity vector and compute new guidance vectors to follow using Algorithm 2. While in motion, mobile agents continuously monitor if they reach destination regions or for obstacles to evade.

**Algorithm 2** Baseline Mobile Agent Behaviour

Initialize $s_a(0) = "F_{out}"$
**while** $s_a(k) \in ["F_{out}", "F_{in}"]$ **do**
  Listen for $\omega_b^s(k)$, $\overrightarrow{v}_b^s(k)$ for opposite role $\overline{s}_m(k)$
  Broadcast personal $\Delta_m^s(k)$, $\overrightarrow{v}_m(k)$
  Compute $\overrightarrow{v}_m(k+1)$ as per Equation (4-11) - (4-14)
  Perform motion along $\overrightarrow{v}_m(k+1)$ for an interval T seconds
  **while** InMotion **do**
    **if** Obstacle **then**
      Perform obstacle avoidance as per Equation (4-15)
    **end if**
    **if** $x_a(k) \in \mathcal{T}$ **and** $s_a(k) = "F_{out}"$ **then**
      Extract resource
    **else if** $x_a(k) \in \mathcal{N}$ **and** $s_a(k) = "F_{in}"$ **then**
      Deposit resource
    **end if**
    Check possible role switch as per Equation (4-5)
  **end while**
**end while**

Agents working together as described establish a guidance network enabling the exploration of the domain $\mathcal{D}$ as well as the construction of paths between regions of interest.

# Chapter 5

# Sustained Foraging

The preliminary system presented in Section 4-2 proved to be successful in establishing paths between a nest region and a target region with agents travelling back and forth. Although this indicates the potential of the foraging system, it is limited to theoretical application. The set of amendments in this chapter defines both the energetic model and proposes mechanisms to solve the problem of maintaining energy levels in agents.

Section 5-1 establishes the energy-balance used to estimate the evolution of the system energetics. Section 5-2 then discusses what implications the inclusion of a recharging region $\mathcal{R}$ has for the system mechanisms and introduces the new role of recharging $R$. Subsequently, Section 5-3 up until Section 5-8 propose amendments to the baseline system to include energetic maintenance as a fundamental task of the agents and the populations.

## 5-1 Energy Model

The first change with respect to the baseline system is the inclusion of a numerical estimation of energy consumption. When in motion, computing, communicating, or performing any active role, robots consume energy. They do so from their energy storage. Furthermore, agents can replenish their energy reserves using solar power. As such, by balancing both power consumption and power harvesting, energy levels can be estimated.

Consider the power consumption of an agent. Agents perform a variety of tasks depending on their role and situation. Beacon agents continuously execute general computations, partake in communication, and perform updates to guidance values. Mobile agents additionally move around, monitor obstacles, scan for resources, and if at a resource node, perform resource operations. The power consumed by an agent consists of the ongoing tasks, as described in Equation (5-1).

$$P_{out} := P_{motion} + P_{avoidance} + P_{computation} + P_{communication} + P_{scan} + P_{exploit} \tag{5-1}$$

For computation, a constant static power amount is assumed. Avoidance, scanning, and exploitation are assumed to be a lump sum for sub-activities performed incrementally. Communication is a sum of power amounts needed to maintain every active communication link. The primary power expenditure is assumed to be the motion performed by the agent. For actuation, the model in Equation (5-2) uses motor-parameters $k_V, k_I$ dictating voltage $U_{motor}$ and current $I_{motor}$ with respect to revolution speed $\omega_m$ and torque $T$.

$$P_{motion} := U_{motor} \cdot I_{motor} := \left(\frac{\omega_{motor}}{k_V}\right) \cdot \left(\frac{T_{motor}}{k_I}\right) \tag{5-2}$$

Agents can recharge using their onboard solar array. With an assumed array-size and array efficiency, and taking the solar irradiance on the surface of the Moon, the power absorbed is expressed in Equation (5-3). Note that the term $Irr_{solar}$ takes the incidence angle into account.

$$P_{in} := \eta_{array} \cdot Irr_{solar} \cdot A_{array} \tag{5-3}$$

The sum of these two power terms is what is used to continuously estimate the state of the battery. By computing the balance at each time step and integrating with respect to time, the battery levels $E_{battery} \in [0, E_{battery,capacity}]$ and State-of-Charge (SOC) $SOC_{battery} \in [0, 1]$ can be computed using Equation (5-4) and Equation (5-5).

$$E_{battery}(k+1) := E_{battery}(k) + (P_{in} - P_{out}) \cdot dt \tag{5-4}$$

$$SOC_{battery}(k) = E_{battery}(k)/E_{battery,capacity} \tag{5-5}$$

The described approach to energy-level estimation is implemented for each agent. In doing so, estimates of energetic dynamics within the foraging population can be monitored.

## 5-2 Recharging Region $\mathcal{R}$

Until now, the baseline system considered two regions of interest, nest $\mathcal{N}$ and target $\mathcal{T}$. Within this baseline, mobile agents are instructed to move endlessly between these two regions. However, with the element of draining energy levels, an additional region and role is introduced to allow the system to replenish energy levels.

A major amendment to the baseline with grand consequences is the introduction of recharging region $\mathcal{R}$. This region is defined by a minimum level of solar irradiance as denoted by Equation (5-6) and also stated in Section 3-4. When within this region, agents can harvest solar power using their solar arrays and charge their batteries.

$$\mathcal{R} := \{x_{\mathcal{R}} \in \mathcal{D} | Irr_{solar}(x_{\mathcal{R}}) > \eta_{solar}\} \tag{5-6}$$

Region $\mathcal{R}$ is considered a fundamental region in that agents visit this region out of necessity rather than to contribute to the foraging objective. Failing to reach the recharging region in time degrades the population whereas failing to reach either of the other two regions merely delays results.

Adding a third region of interest also introduces a recharging role $R$. This role is defined in similar fashion to $F_{out}$ and $F_{in}$:

- *Recharging role* $R$: recharging agents $R$ attempt to travel to the recharging region $\mathcal{R}$ from their current location and perform recharging activities

The recharging role is triggered when an agent's SOC falls below either the SOC required to reach its destination or the recharging region. This logic ensures that an agent can always recharge. A more detailed switching scheme is presented in Section 5-4. With this, the set of mobile agents is expanded to $\mathcal{M} := \{a \in \mathcal{A} | s_a(k) \in \{F_{out}, F_{in}, R\}\}$.

The addition of a third region of interest and a role has major implications for the structure of coordination. Until now, the baseline had two active roles and two origins of path exclusive to one another: if the agent was "$F_{out}$", it previously had to be "$F_{in}$" and vice versa. See Figure 5-1: target region $\mathcal{T}$ is only achievable using the red sequence coming from nest $\mathcal{N}$ and $\mathcal{N}$ only from $\mathcal{T}$. This exclusivity of role-origin is what is utilised in Equation (4-9) where one role is directly used to update the counterpart's guidance.

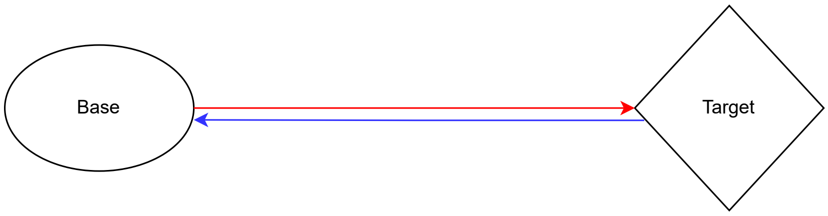



**Figure 5-1:** Guidance between two regions of interest with role-origin exclusivity.

The addition of a third active role and region removes the role-origin exclusivity. Agents moving towards their destination can come from either one of the remaining two regions, as illustrated in Figure 5-2. Considering agent experience only encodes the direction to its last visited region, knowing the path's origin is necessary for interpretation. As a result, guidance updates are here on out performed using an agent's last successfully completed role $s^c(k) \in [$"$F_{out}$", "$F_{in}$", "$R$"$]$ (Equation (5-7)) instead of its current role. An agent who successfully completed its previous role could only do so at the role's destination region. For example, an "$F_{out}$" role is only successful when the individual reaches the target region, meaning its experience will correspond to direction $\mathcal{T}$.

$$\begin{aligned} s^c(k) = s_a(k_{complete}) \quad \text{with} \quad k_{complete} := max(k | k \in \mathbb{N}, & s_a(k) = \text{"}F_{out}\text{"} \wedge x_a(k) \in \mathcal{T} \\ & \vee s_a(k) = \text{"}F_{in}\text{"} \wedge x_a(k) \in \mathcal{N} \\ & \vee s_a(k) = \text{"}R\text{"} \wedge x_a(k) \in \mathcal{R}) \end{aligned} \tag{5-7}$$

Finally, consider an agent starting at $\mathcal{N}$, recharging at $\mathcal{R}$, then walking to region $\mathcal{T}$. The experience of the agent beyond the recharging region no longer encodes the direction of $\mathcal{N}$, only that of $\mathcal{R}$. Using this experience to construct guidance to the nest would result in a single sequential path stringing all three regions in order of visitation. By making the distinction based on the last region visited, disjointed paths are enabled allowing individual optimization. The downside is that beacons beyond the recharging region no longer receive experience on the nest's direction. This knowledge can only be attained from agents travelling directly from the nest to the target. This local lack of guidance is addressed in Section 5-8.

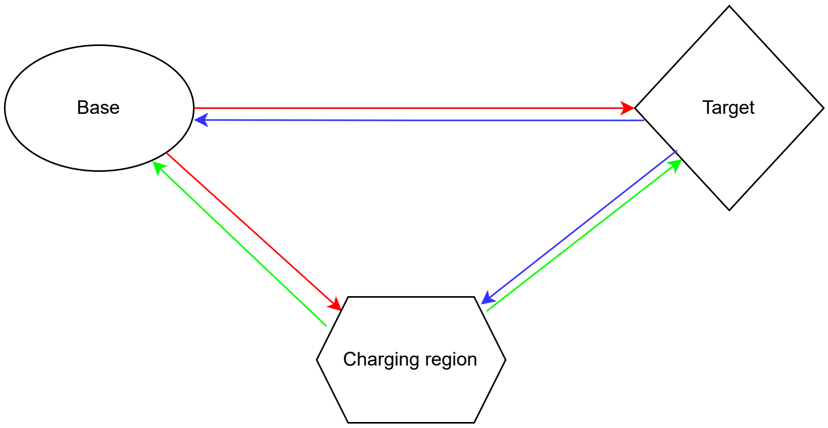



**Figure 5-2:** Expanded number of regions of interest.

## 5-3 Guidance Values

The baseline system expresses path quality using rewards attributed to finding destination regions, with indicators distributed between beacons through travel frequency. Agents select paths based on whether they are successful, with quality as a second measure. With the added functionality of battery SOC monitoring, both definitions must be expanded.

Agents have the fundamental task of ensuring they at all times have enough energy reserves to reach the recharging region. This is to ensure the sustainability of the population. This requirement implies that paths requiring more SOC than the agent has available are no longer viable choices as they cannot be completed without the agent disengaging to recharge. This indication of viability is achieved by including the path-required SOC in the agent's experience and beacon knowledge. Furthermore, similar to potential weights, required SOC-levels can be used to weigh guidance vectors. With lower energy consumption deemed desirable, agents prompted with multiple guidance vectors select paths requiring less energy. In doing so, the system includes power costs in path optimization.

First, Equation (5-8) implements the change proposed by Section 5-2. Beacon values are henceforth updated using $s^c$ rather than the agent's current role $s$. Furthermore, guidance to region $\mathcal{R}$ using role $R$ is added.

$$\begin{Bmatrix} To\mathcal{N}: & \{ & \omega_b^{F_{out}}(k), & \vec{v}_b^{F_{out}}(k) & \} \\ To\mathcal{T}: & \{ & \omega_b^{F_{in}}(k), & \vec{v}_b^{F_{in}}(k) & \} \end{Bmatrix} \rightarrow \begin{Bmatrix} To\mathcal{N}: & \{ & \omega_b^{s^c=F_{in}}(k), & \vec{v}_b^{s^c=F_{in}}(k) & \} \\ To\mathcal{T}: & \{ & \omega_b^{s^c=F_{out}}(k), & \vec{v}_b^{s^c=F_{out}}(k) & \} \\ To\mathcal{R}: & \{ & \omega_b^{s^c=R}(k), & \vec{v}_b^{s^c=R}(k) & \} \end{Bmatrix} \tag{5-8}$$

To enable SOC-based decision-making, beacon values are appended with SOC-values (see Equation (5-9)). Mobile agents now include their used SOC since leaving the role-origin as part of their experience. Beacons construct the amount of SOC required to reach a destination. Here, $SOC_b^{s^c=F_{out}}(k)$ represents the SOC required to reach region $\mathcal{T}$ from the communication region $\mathcal{C}_b$. Be aware that these values implement an additional safety factor $\phi_{SOC}$ as seen later in Equation (5-13). Agents with ample SOC continue on their path, whereas agents with not enough energy can divert to the recharging region.

$$
\begin{Bmatrix}
To\mathcal{N}: & \{ & \omega_b^{s^c=F_{in}}(k), & \vec{v}_b^{s^c=F_{in}}(k) & \} \\
To\mathcal{T}: & \{ & \omega_b^{s^c=F_{out}}(k), & \vec{v}_b^{s^c=F_{out}}(k) & \} \\
To\mathcal{R}: & \{ & \omega_b^{s^c=R}(k), & \vec{v}_b^{s^c=R}(k) & \}
\end{Bmatrix} \rightarrow
\begin{Bmatrix}
To\mathcal{N}: & \{ & \omega_b^{s^c=F_{in}}(k), & \vec{v}_b^{s^c=F_{in}}(k), & SOC_b^{s^c=F_{in}}(k) & \} \\
To\mathcal{T}: & \{ & \omega_b^{s^c=F_{out}}(k), & \vec{v}_b^{s^c=F_{out}}(k), & SOC_b^{s^c=F_{out}}(k) & \} \\
To\mathcal{R}: & \{ & \omega_b^{s^c=R}(k), & \vec{v}_b^{s^c=R}(k), & SOC_b^{s^c=R}(k) & \}
\end{Bmatrix}
\tag{5-9}
$$

## 5-4 Roles and Role Selection

Agents within the baseline framework are employed in two mobile roles, each fulfilling the objective of moving endlessly between the nest and the target. Unless the agent is required to take on a facilitating role in the form of a beacon, role-selection is sequential. With the addition of the fundamental task of recharging, the decision of which role to assume gains an additional dimension, and with it added complexity. This section will describe the developed role selection.

Consider the matter of recharging. For sustained operation, an agent needs to have enough SOC to reach the recharging region at all times. The recharging role is to be assumed pre-emptively, to ensure agent SOC never falls below these levels. Mechanisms are implemented in role selection to allow agents to disengage from their current tasks of foraging and deviate to the recharging region $\mathcal{R}$. Furthermore, after having performed recharging operations, agents resume their previous foraging tasks. The proposed tendencies are used to update Equation (4-5) with added functionality, resulting in the updated role selection of Equation (5-10) (line-numbers before the bracket are added for clarity).

$$
s_a(k+1) =
\begin{array}{r}
\scriptstyle 1: \\ \scriptstyle 2: \\ \scriptstyle 3: \\ \scriptstyle 4: \\ \scriptstyle 5: \\ \scriptstyle 6: \\ \scriptstyle 7: \\ \scriptstyle 8: \\ \scriptstyle 9: \\ \scriptstyle 10: \\ \scriptstyle 11: \\ \scriptstyle 12: \\ \scriptstyle 13: \\ \scriptstyle 14:
\end{array}
\begin{cases}
B & \text{if} \quad \mathcal{B}_a = \emptyset \wedge s_a(k-2) \in \{F_{out}, F_{in}, R\} \\
R & \text{if} \quad SOC_a(k) < SOC_b^s(k) \quad \vee \\
 & \quad\;\; SOC_a(k) < SOC_b^R(k) \quad \vee \\
 & \quad\;\; x_a(k) \in \mathcal{N} \wedge x_a(\forall k_{past} \leq k) \notin \mathcal{R} \\
F_{out} & \text{if} \quad s_a(k) = F_{in} \wedge x_a(k) \in \mathcal{N} \quad \vee \\
 & \quad\;\; s_a(k) = B \wedge s_a^-(k) = F_{out} \wedge \sum_s w_a^s(k) < \eta_w \wedge \mathcal{M}_b(k) = \emptyset \vee \\
 & \quad\;\; s_a(k) = R \wedge x_a(k) \in \mathcal{R} \wedge SOC_a(k) = 1 \wedge s_a^c(k) = F_{in} \quad \vee \\
 & \quad\;\; s_a(k) = R \wedge x_a(k) \in \mathcal{N} \wedge SOC_a(k) > SOC_b^{F_{out}} \wedge SOC_a(k) > SOC_b^R \\
F_{in} & \text{if} \quad s_a(k) = F_{out} \wedge x_a(k) \in \mathcal{T} \quad \vee \\
 & \quad\;\; s_a(k) = B \wedge s_a^-(k) = F_{in} \wedge \sum_s w_a^s(k) < \eta_w \wedge \mathcal{M}_b(k) = \emptyset \vee \\
 & \quad\;\; s_a(k) = R \wedge x_a(k) \in \mathcal{R} \wedge SOC_a(k) = 1 \wedge s_a^c(k) = F_{out} \quad \vee \\
 & \quad\;\; s_a(k) = R \wedge x_a(k) \in \mathcal{T} \wedge SOC_a(k) > SOC_b^{F_{in}} \wedge SOC_a(k) > SOC_b^R \\
 & \quad\;\; s_a(k) = R \wedge x_a(k) \in \mathcal{R} \wedge SOC_a(k) = 1 \wedge ||v_b^R(k)|| = 0 \\
s_a(k) & \quad\;\;\; \text{else}
\end{cases}
$$

$$
\forall a \in \mathcal{A} \tag{5-10}
$$

The following updates are applied:

- Line 2-3: An agent assumes the recharging role when it either does not have enough SOC to safely reach its destination or the recharging region $\mathcal{R}$
- Line 4: An agent assumes a recharging role if it is at the nest and has not visited a recharging region before. Agents who found the target region before the recharging region and have returned with resources are instructed to make sure the recharging region is attainable
- Line 7: An agent becomes outbound if it reaches the recharging region $\mathcal{R}$ replenishes its battery, and previously completed its inbound tasks
- Line 8: An agent becomes outbound if it accidentally reaches the nest region $\mathcal{N}$ and has ample SOC to reach both the target and recharging region
- Line 11: An agent becomes inbound if it reaches the recharging region $\mathcal{R}$, replenishes its battery, and previously completed its outbound tasks
- Line 12: An agent becomes inbound if it accidentally reaches the target region $\mathcal{T}$ and has ample SOC to reach both the nest and recharging region
- Line 13: Additionally, an agent becomes inbound when it successfully reaches the recharging region, replenishes its batteries, and determines that no recharging path has been constructed locally between the nest and the recharging region. This feature ensures that a viable path between the two fundamental regions is available

Figure 5-3 provides a visualization of the proposed role-selection process. Considering Figure 4-2, the role-selection procedure has become significantly more complex.

## 5-5 Updated Knowledge Dynamics

The way in which received experience is handled within the system remains unchanged. Using past experience of mobile agents, potential weights and guidance vectors are monitored. However, the addition of energy consumption as an additional "experience" value as well as the updated concept of experience gathering without exclusive roles changes the practical methods of the beacons.

The first change applied is a result of the removal of role-origin exclusivity, as discussed in Section 5-3. As the agent's current role no longer describes its origin, it is replaced by the last completed role, $s_m^c(k)$. This is used to update potential weights and vectors as per Equation (5-11) and Equation (5-12).

$$\omega_b^s(k+1) = (1-\rho_w)\omega_b^s(k) + \rho_w \frac{\sum_{m\in\mathcal{M}_b^{s^c}(k)} \Delta_m^s(k)}{|\mathcal{M}_b^{s^c}(k)|} \tag{5-11}$$

$$\overrightarrow{v}_b^s(k+1) = (1-\rho_v)\overrightarrow{v}_b^s(k) + \rho_v \frac{\sum_{m\in\mathcal{M}_b^{s^c}(k)} -\overrightarrow{v}_m(k)}{|\mathcal{M}_b^{s^c}(k)|} \tag{5-12}$$

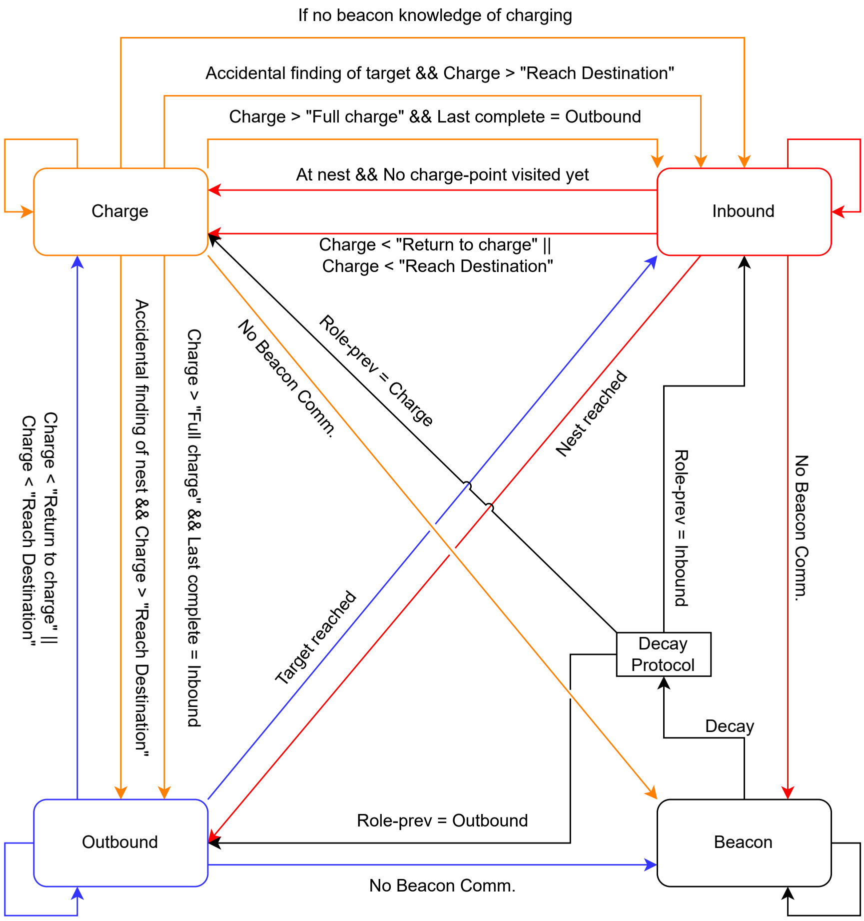



**Figure 5-3:** Updated role switching of the proposed system.

Furthermore, as per Equation (5-9), an additional SOC parameter is included for each destination. The value of $SOC_b^{s^c} \in [0, 1]$ remains zero until a valid input is received, upon which the value is initialized to that SOC. This is done to prevent over-confidence or conservatism affecting role selection. Updates are performed similar to that of the potential weight and guidance vector, with the introduction of a learning weight $\rho_p \in \mathbb{R}_+$. Additionally, a safety-factor $\phi_{SOC} \in \mathbb{R}_+$ is applied over the SOC value to ensure safety of operation.

$$SOC_b^{s^c}(k+1) = (1-\rho_p)SOC_b^{s^c}(k) + \rho_p \frac{\sum_{m \in \mathcal{M}_b^{s^c}(k)} SOC_m^{s^c}(k)}{|\mathcal{M}_b^{s^c}(k)|}\phi_{SOC} \tag{5-13}$$

## 5-6 Updated Motion Dynamics

The motion dynamics of mobile agents remain unchanged with respect to the baseline. What does change is how the guidance vector is computed based on beacon input.

An agent in the baseline system attempts to reach its destination by following paths high in success potential. It promotes paths with higher potential weights hereby performing an optimization step. With the addition of energy considerations, the same methodology can be applied to path-required SOC-amounts. Paths requiring less energy are preferred over energy-expensive ones. Accordingly, Equation (4-11) is updated to Equation (5-14). Note tunable weights $\rho_f \in \mathbb{R}$ and $\rho_c \in \mathbb{R}$, allowing for tuning of optimization considerations.

$$\vec{v}^{\,s}_{f}(k) := \left\langle \sum_{b \in \mathcal{B}_m(k)} \rho_f \;\; \omega_b^{s_m(k)}(k) \;\; \rho_c \left[1 - SOC_b^{s_m(k)}(k)\right] \;\; \vec{v}_b^{\,s_m(k)}(k) \right\rangle \tag{5-14}$$

## 5-7 Decay

A feature included as an amendment in the baseline is the repurposing of unused beacons using decay. Beacons deciding to decay engage in a random search to rejoin the network, a strategy not guaranteeing immediate reconnection to the swarm. This strategy is replaced by a more directed approach.

The proposed change relies on the knowledge that decayed beacons are at the outskirts of the guidance network. Beacons furthest away from successful paths receive the least potential weight and thus decay first. As a result, they always have the guidance network behind them.

In the newly proposed system, upon decaying, an agent performs a 180° turn and starts its motion. In doing so, it attempts to reconnect with its last beacon passed. Having achieved communication, the decayed agent moves along the network using the guidance of its last completed role $s_m^c(k)$. This guides agents back toward the origin of their role, at the stem of the guidance network for the current role branching. Having backtracked a certain distance, the agent re-engages in its current role and with it in the attributed navigation. Should an agent be unsuccessful in establishing a connection with the guidance network, it engages in a random search similar to the baseline.

This mechanism of backtracking moves agents back toward their origin, where all paths for their role start. Encountering a successful path to its destination is therefore more likely. Furthermore, agents are able to get out of obstacle traps such as dead-ends. Finally, as agents do not switch roles, they continue providing experience during backtracking, albeit negative experience to help degrade unsuccessful paths.

The depth of backtracking depends on the number of unsuccessful attempts the agent has made to reach its destination. The act starts with small backtracks, but increases in distance with the number of unsuccessful attempts, $n_{failed\ attempts}$, as seen in Equation (5-15). This allows for backtracking further towards the origin. Although the backtracking distance is estimated using encoder readings, these are not required to be accurate and could be replaced by counting the number of beacons contacted.

$$d_{backtrack} = n_{failed\ attempts} \cdot 0.5 \cdot \delta_{comm} \tag{5-15}$$

## 5-8 Fallback

With the addition of a third region of interest and loss of role-origin exclusivity, information availability for all paths is no longer guaranteed at all beacons. As Section 5-2 pointed out, intermediate destinations remove any experience of previous areas. Agents may not have immediate local guidance to their destination.

To address this issue, a fallback mechanism is proposed. This feature kicks in when no direct guidance is available to an agent's destination, but a secondary intermediate destination can be used for approach. Consider the following scenarios:

- Agents move from nest $\mathcal{N}$, through recharging point $\mathcal{R}$, to target $\mathcal{T}$: in this case, beacons between the recharging point and target only receive experience on where the recharging point is, not where the nest is. Agents wanting to return from the target to the nest now have no guidance
- Agents move from nest $\mathcal{N}$ to target $\mathcal{T}$ in a region where recharged agents never rejoin the system: in this case, a path is established between nest and target without guidance to the recharging point. Any agent willing to recharge now has no knowledge on its direction

These two scenarios depict how the limited agent experience causes gaps in coordination. However, what is known is that agents start off in the nest and intend to find the recharging point first. Furthermore, role-switching ensures that a bi-directional path is established upon finding the recharging region. As a result, reaching one of the regions will result in attaining information on the other.

This feature is exploited for intermediate destinations. When an agent has no guidance available to reach the nest, it assumes the recharging region as temporary destination up until it regains guidance to its primary goal. The same mechanism holds for the recharging region, with the nest proposed as intermediate destination. Fallback ensures guidance for returning or recharging agents, even if indirect. The same cannot be applied for target regions as there is no guarantee that either the recharging region or the nest region previously established a direct path. Equation (5-16) depicts the selection of guidance vector for agents in roles "$F_{in}$" or "$R$".

$$\overrightarrow{v}^{s}_{f}(k) = \begin{cases} \overrightarrow{\hat{v}}^{s}_{f}(k) & \text{if} \quad ||\overrightarrow{v}^{s}_{f}(k)|| > 0 \\ \overrightarrow{\hat{v}}^{["F_{in}","R"]-s}_{f}(k) & \text{if} \quad ||\overrightarrow{\hat{v}}^{s}_{f}(k)|| = 0 \wedge ||\overrightarrow{\hat{v}}^{["F_{in}","R"]-s}_{f}(k)|| > 0 \\ [0,0] & \text{else} \end{cases} \tag{5-16}$$
$$\forall a \in \mathcal{A}, s_a(k) \in ["F_{in}","R"]$$

## 5-9 Summary of Proposed System

Chapter 5 provides an extensive update to the baseline system of Section 4-2. The goal of these changes are to include energetic maintenance tasks in system behaviour, both high- and low-level. This consideration makes the system viable for real-life implementation. This section provides an overview of the changes.

Adjustments can best be presented in chronological order of system operation:

1. Agents are initialized to look for a recharging point. The availability of power is the primary objective for sustained operation
2. Agents who accidentally find target locations do construct and return on a path between nest and target. However, they continue to look for a recharging point until initial recharging is performed
3. Beacons are able to distinguish experience using an agent's last completed role, enabling the establishment of multiple independent paths
4. Mobile agents choose their destination region based on the availability of guidance and SOC required. Agent SOC below the minimum required charge for the path triggers the recharging role
5. Mobile agents rejoin the foraging process after having recharged. Their new role is selected is based on their last successful role
6. Mobile agents include the SOC of a path in their construction of guidance. Mobile agents promote power-optimal paths and reinforce these within the guidance system
7. Recently decayed agents backtrack through the guidance network to attempt to find a successful path stemming more towards its origin. Should backtracking be unsuccessful, agents fall back on random-search
8. Agents with destinations for which no local guidance is available fallback on intermediate destinations to approach the original destination

Similar to Algorithm 1 and Algorithm 2, the behaviour of agents can be represented in pseudo-code. First, consider the beacon behaviour.

Algorithm 3 demonstrates the changes to beacon logic. The guidance dictionary is expanded to three destinations to include paths to $\mathcal{R}$. The addition of SOC values to each destination region is also updated. Finally, beacons now distinguish between received experience using the provided last successful role $s^c$ for each agent.

**Algorithm 3** Beacon Behaviour

**while** $s_a(k) = "B"$ **do**
  Broadcast $\omega_b^{s^c}(k), \overrightarrow{v}_b^{s^c}(k), SOC_b^{s^c}(k)$ for all roles
  Listen for mobile agent input for T seconds
  **for** each role $s^c \in ["F_{out}", "F_{in}", "R"]$ **do**
    Compute $\omega_b^{s^c}(k+1)$ as per Equation (5-11)
    Compute $\overrightarrow{v}_b^{s^c}(k+1)$ as per Equation (5-12)
    Compute $SOC_b^{s^c}(k+1)$ as per Equation (5-13)
  **end for**
  Check possible beacon decay as per Equation (5-10)
**end while**

The update to mobile agents is more significant. Algorithm 4 depicts multiple changes proposed throughout this chapter. First, line 2 shows the expansion of possible mobile roles to include $R$. Line 5-7 implements the decaying method of returning to the network and backtracking using the previous successful role. Finally, Line 18-19 implements recharging activities when at $\mathcal{R}$. Although the notion of unmentioned steps remain the same, their execution uses the previously updated equations.

**Algorithm 4** Mobile Agent Behaviour

1: Initialize $s_a(0) = "R"$
2: **while** $s_a(k) \in ["F_{out}", "F_{in}", "R"]$ **do**
3:   Listen for $\omega_b^s(k), \overrightarrow{v}_b^s(k), SOC_b^s(k)$ for role $s_m(k)$
4:   Broadcast personal $\Delta_m^s(k), \overrightarrow{v}_m(k)$
5:   **if** $s_a^-(k) = "B"$ **then**
6:     Follow $\overrightarrow{v}_m(k+1) = -\overrightarrow{v}_m(k)$ until connection with a beacon
7:     Perform backtrack motion along $\overrightarrow{v}_m^{s^c}(k)$ for distance $d_{backtrack}$ as per Equation (5-15)
8:   **else**
9:     Compute $\overrightarrow{v}_m(k+1)$ as per Equation (5-14) and Equation (4-12) - (4-14)
10:     Perform motion along $\overrightarrow{v}_m(k+1)$ for an interval T seconds
11:   **end if**
12:   **while** InMotion **do**
13:     **if** Obstacle **then**
14:       Perform obstacle avoidance as per Equation (4-15)
15:     **end if**
16:     **if** $x_a(k) \in \mathcal{T}$ **and** $s_a(k) = "F_{out}"$ **then**
17:       Extract resource
18:     **else if** $x_a(k) \in \mathcal{N}$ **and** $s_a(k) = "F_{in}"$ **then**
19:       Deposit resource
20:     **else if** AtDestination **and** $s_a(k) = "R"$ **then**
21:       Recharge
22:     **end if**
23:     Check possible role switch as per Equation (5-10)
24:   **end while**
25: **end while**

Chapter 6

# Experimental Analysis

This research assesses the implementation of a foraging system in scenarios where system maintenance is required. Evaluating how the system performs is therefore done in an environment simulating energy dynamics in robots. Section 6-1 presents the setup of the simulated scenario used to attain simulated run performance. Results are described qualitatively in Section 6-2. To evaluate the system on its robustness and performance achieved, Section 6-3 performs a quantitative evaluation of the system together with a sensitivity analysis for optimization and flexibility.

## 6-1 Simulated Application

This research studies the feasibility of using a nature-inspired foraging system in application. To ensure that the system can maintain its operation with respect to energy requirements, its performance is put to the test with scenarios implementing energy drainage.

### 6-1-1 Simulation Software

Simulations were performed using *Webots* software. *Webots* was selected for its low-level resource requirements optimal for swarm simulations [19][38]. Logic was provided in a *Python*-script run through *Webots* in individual process instances. Along with being optimal in resource allocation, the disconnected nature of instances represents the disconnected nature of swarming.

For reference of all objects discussed hereafter, consult the following *GitHub* repository.

### 6-1-2 Simulated Rover

The rover used for system simulation is heavily inspired by the Yutu-2 rover [23] and CADRE[1] rovers. Although no extensive application study was performed on the vehicle, its capabilities

[1] https://www.jpl.nasa.gov/missions/cadre

are considered simplistic and feasible in use.

The modelled robot (see Figure 6-1) is a compact, four-wheeled rover capable of traversing the lunar environment. Steering is done using counter-steering wheels with in-place turning possible. Systems onboard are encoded using either *Webots* elements or *Python*-code. The energy subsystem consists of a light sensor to mimic a solar array with Battery Management System (BMS) and a battery simulated in code. Communication is achieved using an omnidirectional emitter-receiver pair of infrared type on top of the vehicle with limited range. Obstacle avoidance relies on seven infrared distance sensors placed at the front of the rover. Finally, as resource manipulation remains abstract, an omnidirectional emitter-receiver pair is used to communicate with the nest and the target region. Resource communication occurs on a separate channel and the storage of resources is simulated in code.

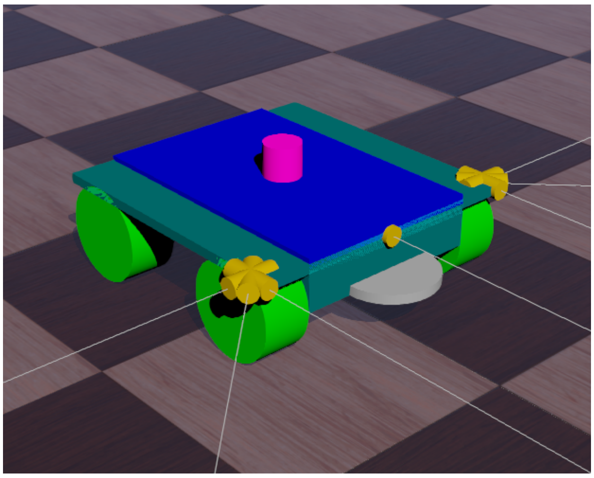


**Figure 6-1:** Agent rover used in simulated scenarios.

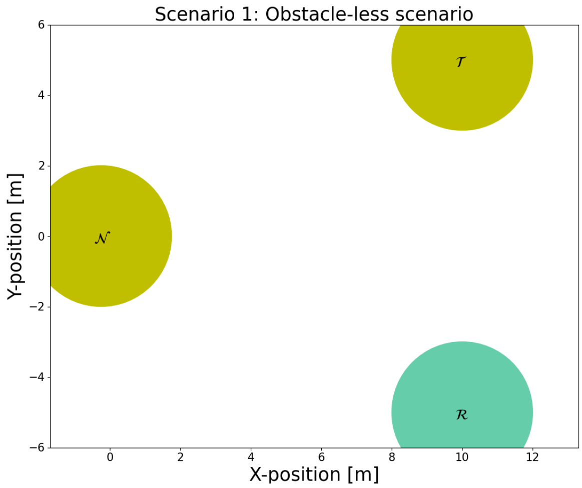
### 6-1-3 Simulated Scenarios

Multiple test scenarios were designed to test system behaviour. Simulating system behaviour gathers numerical data on its performance for both quantitative and qualitative evaluation. Each scenario is intended to bring specific system tendencies to light.

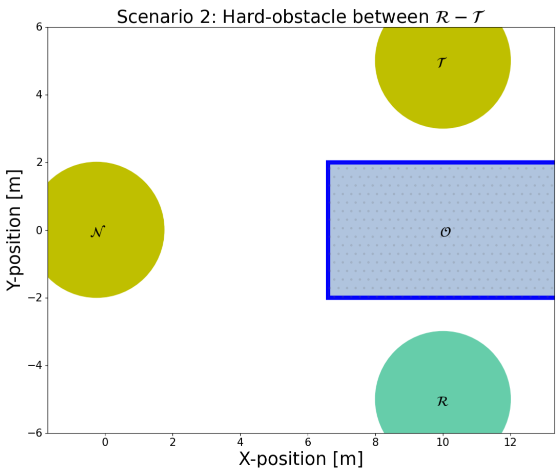
The primary objective of the simulation is to evaluate the system's ability to solve the foraging problem as described in Section 3-4 while needing to fulfill energetic needs. Furthermore, the effects of support mechanisms on the system's optimality are also assessed. The latter should reveal if agents continue improving pathways with the added factor of State-of-Charge (SOC) required.

The agents were tested using three primary scenarios as presented in Figure 6-2. The general setup includes a flat surface with walled boundaries. The three regions of interest, nest $\mathcal{N}$, target $\mathcal{T}$, and recharging region $\mathcal{R}$, are placed on top of this field. In line with the resource-handling of agents, region $\mathcal{N}$ and $\mathcal{T}$ are represented by omnidirectional emitter-receiver pairs. Region $\mathcal{R}$ is achieved by placing a directional light above the playing field to illuminate a region.

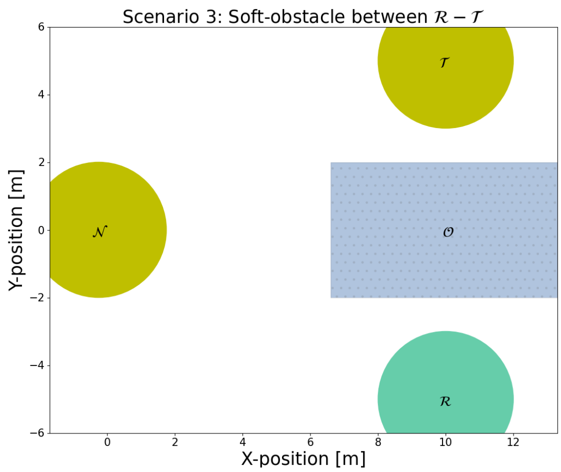
The three scenarios differ in the type of obstacle present. Scenario 1 (Figure 6-2a) does not include an obstacle. This is considered a benchmark where the system is expected to perform optimally. Scenario 2 (Figure 6-2b) places a hard obstacle between two regions. An agent cannot cross a hard obstacle and is therefore forced to walk around. This is a similar setup to that used in [1]. Finally, scenario 3 (Figure 6-2c) includes a soft obstacle. Soft obstacles refer to regions that can be traversed by agents but at higher energy costs. In *Webots*, this is achieved by increasing the ground-friction value $\xi_{ground}$ within the obstacle boundaries using *ContactProperties*. This imitates energy-expensive obstacles such as hills or loose regolith.




**(a)** "No obstacle" scenario with regions $\mathcal{N}, \mathcal{T}, \mathcal{R}$.




**(b)** Hard obstacle: scenario with $\mathcal{N}, \mathcal{T}, \mathcal{R}$ and hard obstacle $\mathcal{O}$ between $\mathcal{R} - \mathcal{T}$.




**(c)** Soft obstacle: scenario with $\mathcal{N}, \mathcal{T}, \mathcal{R}$ and soft obstacle $\mathcal{O}$ between $\mathcal{R} - \mathcal{T}$.

**Figure 6-2:** Simulated control scenarios with varying obstacle types.

The scenarios were designed with reproducibility in mind. Agents and objects are included as *.proto* files to allow scripts to call and manipulate these. Agents are inserted individually by the *Supervisor*-script at a predetermined spawn point when this location does not contain other agents. Obstacles are included in the *World*-files of scenarios.

### 6-1-4 Simulation Parameters

The system as described throughout Chapter 4 and Chapter 5 is sizeable and adjustable using several system parameters. These values do not only influence the system's behaviour, but also allow for tailoring the swarm for its goal and environment. The fundamental scalability advantage of swarming is achieved by adjusting these parameters.

The following parameter values are assumed to be used to achieve the results presented throughout this section. Changes to these parameters are evaluated in Section 6-3.

**Table 6-1:** System parameters.

| $N_a$ | $\tau$ | T | $\rho_w$ | $\rho_v$ | $\rho_p$ | $\rho_f$ | $\rho_c$ | $\delta_{comm}$ | $\eta_w$ | $\eta_{solar}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $[-]$ | $[s]$ | $[s]$ | $[-]$ | $[-]$ | $[-]$ | $[-]$ | $[-]$ | $[m]$ | $[-]$ | $[W/m^2]$ |
| 50 | 0.064 | 10 | 0.1 | 0.1 | 0.1 | 1.0 | 3.0 | 2.0 | 1E-4 | 900 |

| $\phi_{SOC}$ | $v_0$ | $\varepsilon$ | $r$ | $\lambda$ | $\xi_{ground}$ | $E_{battery,capacity}$ |
|---|---|---|---|---|---|---|
| $[-]$ | $[m/s]$ | $[-]$ | $[-]$ | $[-]$ | $[-]$ | $[Wh]$ |
| 1.25 | 1.25 | 0.05 | 1 | 0.8 | [0.0165, 0, 0] | 10 |

## 6-2 System in Application

The next section qualitatively characterizes the acquired system performance. The main goal is to identify certain mechanisms in action during both the exploration and exploitation phase by looking at agents' decisions and movement. Doing so helps verify that the system is operating as intended in Chapter 5

Before specific scenes are discussed, consider Figure 6-3. This figure represents a freeze-frame of a guidance network achieved during a simulation run in a "Soft Obstacle" scenario. Observe the illustrated regions $\mathcal{N}, \mathcal{T}, \mathcal{R}$ and $\mathcal{O}$ in this figure. The first additional elements present are agents in beacon role, marked by a node "B". Each beacon can have up to three guidance vectors, conform to Section 5-3, each pointing along a constructed path. Mobile agents are indicated by a dot, with their colour representing their role. Mobile agents show their guidance vectors computed using an arrow. Mobile agents without guidance vectors are performing random searches, as beacons do not provide guidance vectors for their roles.

### 6-2-1 Exploration Phase

The system starts of in the exploration phase. The objective of the exploration phase is to find the regions of interest $\mathcal{T}$ and $\mathcal{R}$. The target $\mathcal{T}$ is of significance for the primary objective of foraging, namely resource collection. Recharging region $\mathcal{R}$ is required to support operation with energy. The latter is therefore deemed priority, as population activity ends when all agents deplete. Figure 6-4 presents the main milestones of the exploration phase.

Agents initialize at the nest, as seen in Figure 6-4a. Each individual starts in the recharging role $R$ to look for $\mathcal{R}$, following Algorithm 4. As the agents leave the nest, the explored region

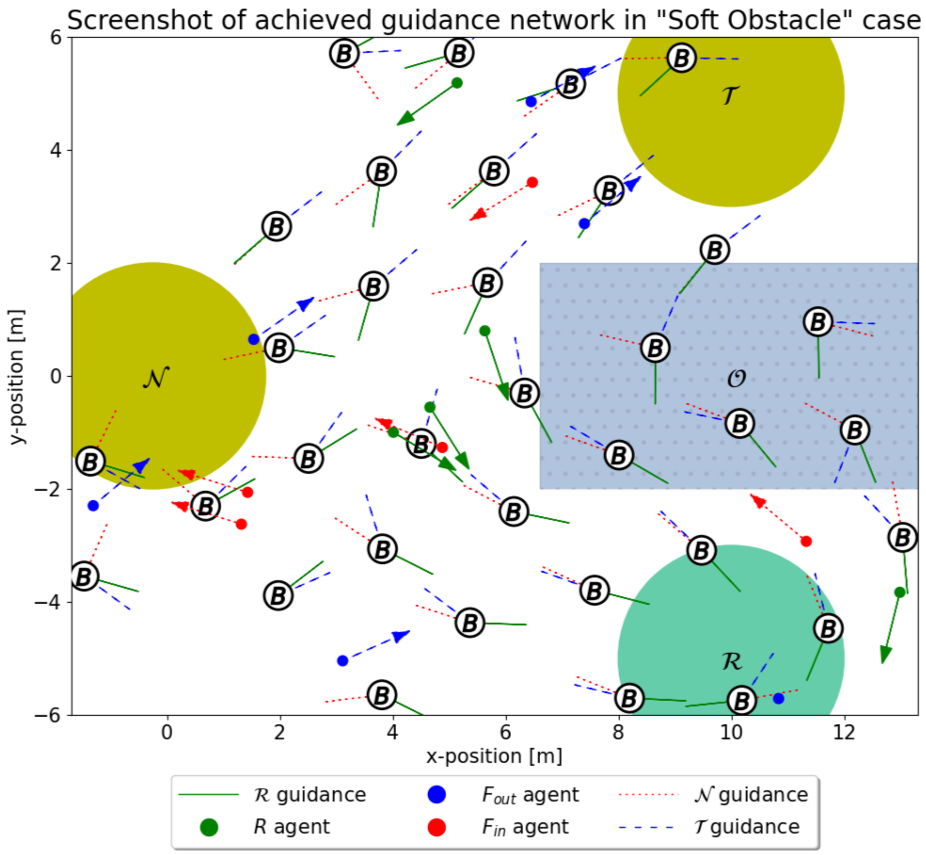


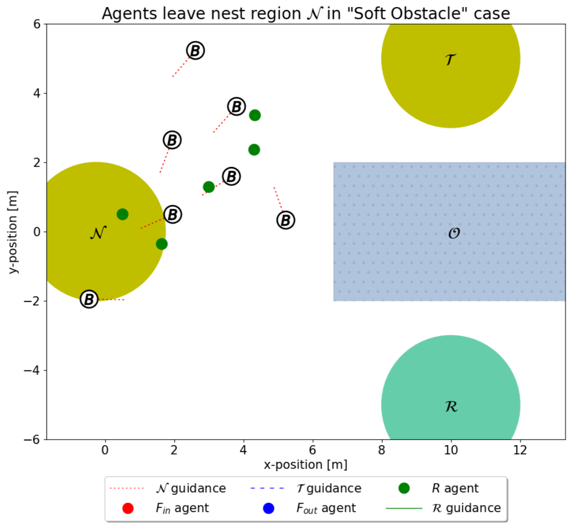

**Figure 6-3:** Example of guidance-network state with agents traversing the area.

and guidance network expands. At this stage, the only guidance vectors present are the ones pointing back to $\mathcal{N}$.

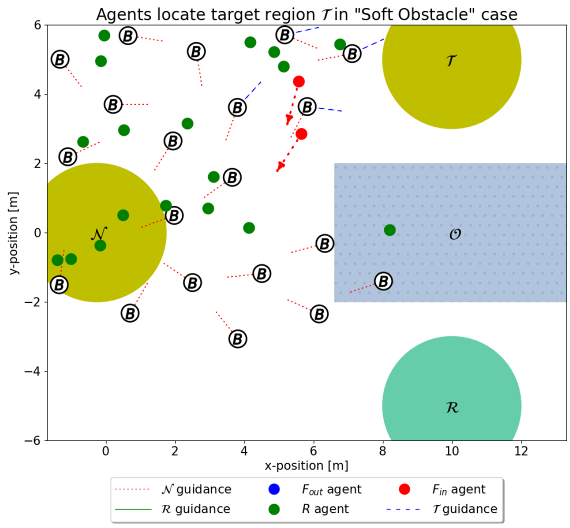
In the depicted run, the target region $\mathcal{T}$ is discovered first. Two agents in Figure 6-4b are seen in inbound roles with guidance vectors pointing towards $\mathcal{N}$. These agents successfully found the target region, recovered some resources, and are returning to $\mathcal{N}$. While doing so, they are providing experience on the direction of $\mathcal{T}$ from which beacons successfully construct new guidance vectors.

The final milestone in the exploration phase is depicted in Figure 6-4c. An agent successfully identifies a recharging region, recharges, and is seen on an inbound path at the bottom of the screenshot while surrounding beacons construct guidance vectors to the region $\mathcal{R}$. Notice how the path to the target has evolved since the previous milestone.

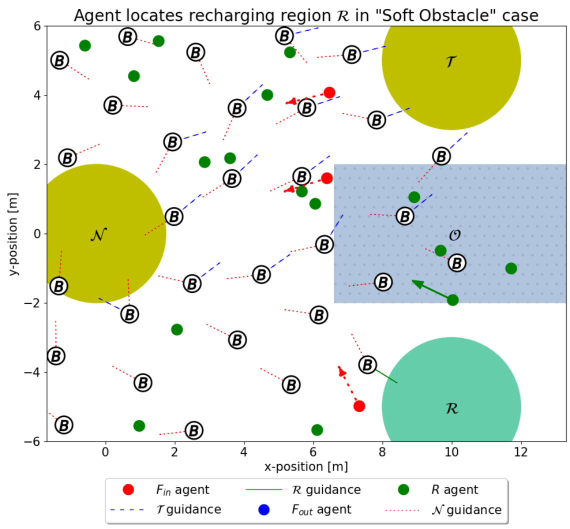
The discovery of both additional regions concludes the exploration phase. From here on out, the guidance network contains all information necessary to construct initial paths. Through reinforcement in the exploitation phase, distinct paths to each region of interest are created and optimized.




**(a)** Agents initialize and leave nest at 50[s].




**(b)** An agent discovers target region $\mathcal{T}$ at 400[s].




**(c)** An agent discovers recharging region $\mathcal{R}$ at 650[s].

**Figure 6-4:** Three major milestones within the exploration phase of the system.

## 6-2-2 Exploitation Phase

With the exploration phase discovering regions of interest and introducing initial guidance information completed, the exploitation phase can commence. The goal of this phase is to extract as many resources from the target and return them to the nest. In the process, the population develops paths given its tendency to choose more optimal paths with respect to potential and energy requirements. Furthermore, it is required to perform fundamental housekeeping tasks in the form of recharging batteries.

Agents continue to operate in the exploitation phase as described in Algorithm 4. Mobile agents select their velocity vectors using guidance vectors provided by beacons with their respective potential weight $\omega_b$ and energy required $1-SOC$. Travelling along selected optimal vectors, an agent's experience on its origin region is deposited along this route. This results in optimal paths being reinforced, increasing the likeliness of other agents choosing the same. The guidance network in Figure 6-5 is the outcome of this iterative learning, with the mobile agents travelling the converged routes.

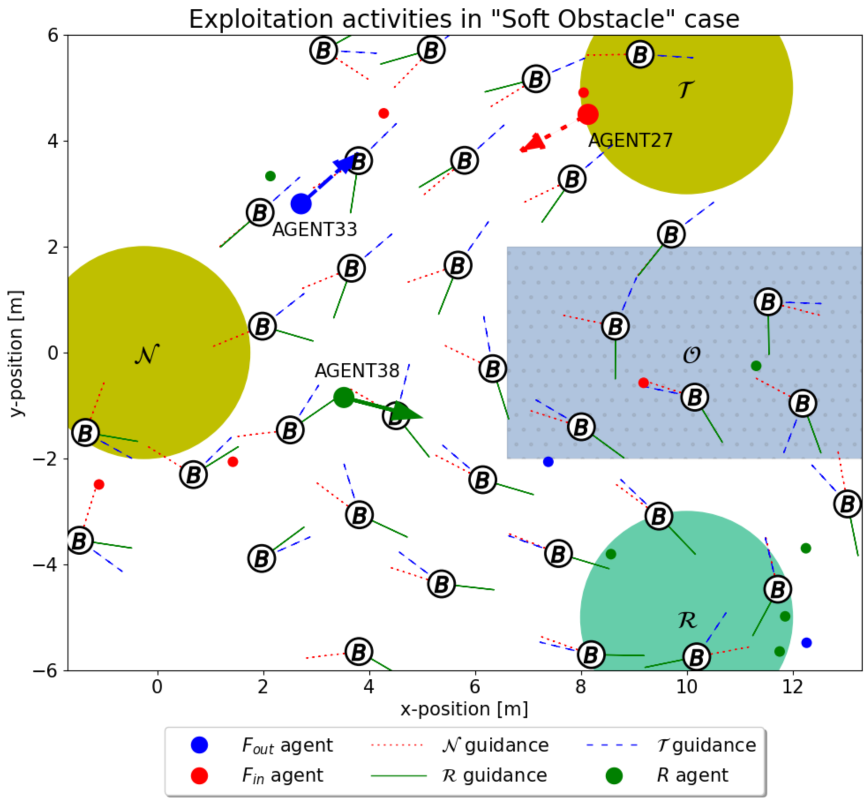



**Figure 6-5:** Guidance network of the exploiting system in "Soft Obstacle" case with highlighted agents.

In Figure 6-5, consider the three highlighted agents, AGENT27, AGENT33, and AGENT38. Agent AGENT33 is an outbound agent in the top-left of the figure, currently moving towards the target region. AGENT27 in the top-right on the other hand has just fulfilled its outbound role by reaching its destination. Reaching the target, it retrieved resources and is now seen leaving the area along inbound guidance vectors. Finally, AGENT38 assumes a charging role and navigates the guidance network to the recharging region.

## Path Optimization

Agents in the exploitation phase walk paths between regions incrementally and endlessly. As stated previously, this selection of the best heading allows agents to deposit their experience in the most optimal reverse direction. This mechanism combined with the reinforcement cycle results in the network learning more suitable paths.

This choice of best heading is represented by Equation (5-14). Mobile agents combine received beacon guidance vectors as a weighted sum with potential weight and SOC-cost as weights.

Agents aim to maximize the potential of reaching a destination while minimizing the energy required to get there. This optimization cost function can be expressed as Equation (6-1).

$$\max_{b \in \mathcal{B}_m(k)} \rho_f \omega_b^{s_m(k)}(k) \;\; \rho_c \left[1 - SOC_b^{s_m(k)}(k)\right] \tag{6-1}$$

Beacons in turn learn from the optimal paths agents travel. With agent experience used to update the respective guidance vectors in beacons, these improve with respect to Equation (6-1). Note that this means that established paths can consist of non-direct SOC-cheap segments.

However, this optimality of paths does not always apply to agent trajectories. Agents update their heading at update intervals using only the beacon information available at that time. As their navigation is based solely on heading, an offset in agent location from intended paths is likely. As a result, agents only loosely follow the paths that the guidance network dictates, leading to sub-optimal performance.

Take the example of AGENT44, highlighted in Figure 6-6, in the same scenario as previously. The agent can be seen cutting the corner of the soft obstacle, despite following the instructions of beacons. Because the network was reinforced using agent heading irrespective of location, agents passing below these beacons taught it a specific direction. Now that AGENT44 is more north than expected, although following the correct heading, it is steered to cut the corner of $\mathcal{O}$ and achieve sub-optimal results. By not being able to execute strict paths with the precision of location, only sub-optimal results can be guaranteed. Increasing domain resolution by decreasing communication range may improve performance as it can map intermediate headings more precisely, but this comes at the cost of mobile agent numbers.

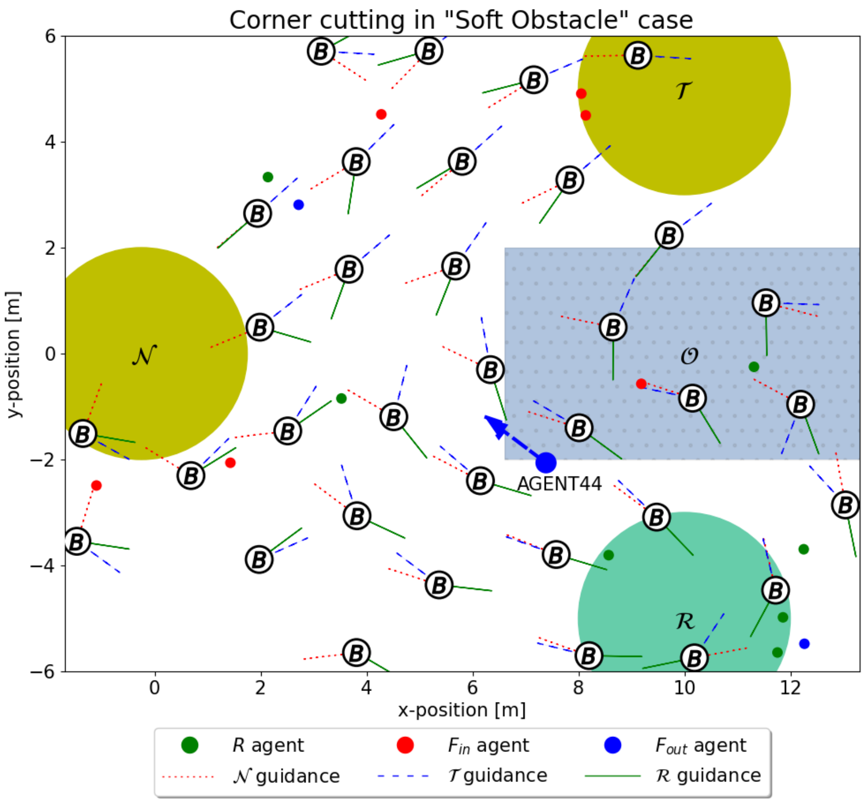



**Figure 6-6:** AGENT44 cutting through soft-obstacle boundaries in "Soft Obstacle" case.

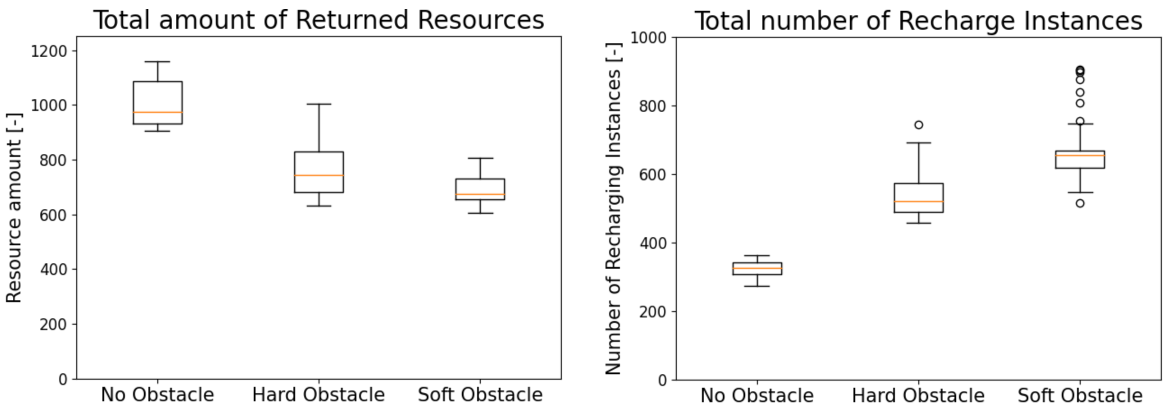
## 6-3 Performance Analysis

The goal of this research is to establish the feasibility of a foraging system implementing maintenance tasks. The previous section illustrated the functionality of the system in stages, describing milestones observed in system evolution. This section builds on this, evaluating multiple simulated runs in varying scenarios, and providing statistical proof of performance certainty.

### 6-3-1 Sustained Foraging with Recharging

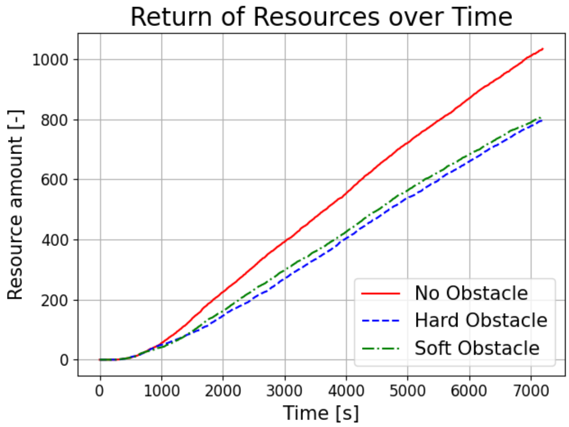
The primary objective of the proposed system is the retrieval of resources from a target region $\mathcal{T}$ and returning it to a nest location $\mathcal{N}$. The newly developed system of Chapter 5 was used to take finite energy reserves into account from which agents drain power for operation. This section ascertains if foraging is still attained with the additional system requirement.

To determine if the system is indeed still foraging while fulfilling the need for recharging, multiple performance indicators are considered. These are compiled from simulated runs performed in all three scenarios of Figure 6-2. To build statistical certainty that performance is not incidental, 50 runs were performed for each scenario. Of those 50 runs, the 30 resulting in the most resources recovered are selected. This is motivated by the assumption that simulation runs with high amounts of returned resources experienced the most agent activity, making system tendencies more apparent. To establish these performance trends, the considered values are averaged for each time step of a run. The indicators are presented in Figure 6-7.

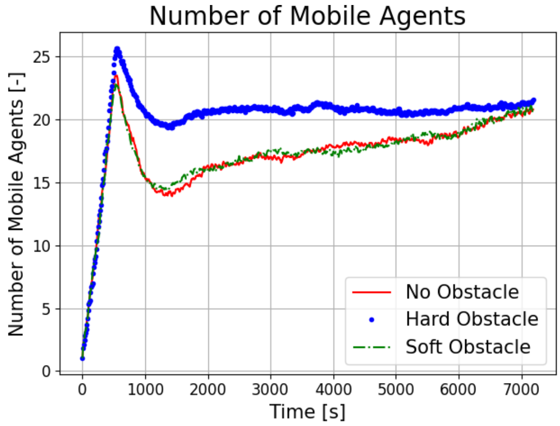



**(a)** Spread of total amount of returned resources within the simulation time of 7200[s].

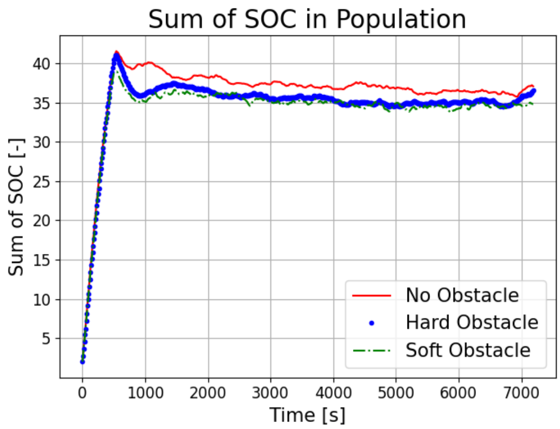
**(b)** Spread of total count of recharging instances within the simulation time of 7200[s].




**(c)** Accumulation of resource at the nest with respect to time.




**(d)** Progression of agent utilization in mobile roles.




**(e)** Progression of population SOC with respect to time.

**Figure 6-7:** System performance under unobstructed and obstructed conditions.

Using the figures in Figure 6-7, the following conclusions can be drawn:

- The main conclusion of Figure 6-7a is that the system is able to forage in all three scenarios. For each scenario, returns of over 600 units of resource imply at least 100 foraging trips between $\mathcal{T}$ and $\mathcal{N}$. With a population size of 50 and beacon agents not foraging, this implies multiple foraging trips per agent
- The system is capable of foraging in both unobstructed and obstructed scenarios. The system is therefore flexible in solving different problems. This adaptability is further explored in Section 6-3-3
- From Figure 6-7b indicates that unobstructed runs experience fewer recharging interruptions. This is explained by the fact that an unobstructed environment allows for direct near-ideal paths to form between regions. These routes require less energy than the deviations enforced by obstacles. The soft-obstacle scenario is found to recharge the most, motivated by the fact that the soft obstacle significantly increases the power consumption of moving agents, beyond that of longer routes over normal surfaces

- Figure 6-7c further depicts the difference in application between obstructed and unobstructed scenarios. The unobstructed returns more resources in the allotted time due to a higher rate of return. Using the same reasoning as before, direct paths between regions result in shorter travel time and thus higher frequency. Obstructed scenarios also have to recharge more often due to the added energy consumption, thereby lowering the travel frequency even further. The fact that the system can maintain a steady rate in each scenario implies continued functionality even if at a lower rate
- The number of mobile agents at a specific time-instance as shown in Figure 6-7d serves as an indication for agent utilization. The number of agents immobilized to become beacons reaches its peak at around 1200[s], where the guidance network comprises of most agents. Later on, agents become mobile again proving the ability of beacons to decay. The presence of mobile agents together with the continued return of resources implies sustained foraging while the guidance network remains operational
- Figure 6-7d depicts the number of mobile agents in the scenarios run. The guidance network requires a number of beacons to spatially cover the domain. After setting up at around 1200[s], the system is seen to reintroduce mobile agents as a result of beacons decaying. This maximum amount of beacons ($N$-mobile agents) is considered the minimum swarm size required to achieve the guidance network, with additional agents helping foraging
- The sum of agent SOC in the population is summarized in Figure 6-7e. An equilibrium is achieved around 1200[s], the approximate time the guidance network is at its peak size. The downward trend from this point is a result of beacons discharging slowly. The oscillations however are a result of mobile agents incrementally draining and recharging their batteries. This proves that the system is actively maintaining system charge

All of these statements support the proposed system's capability of performing resource foraging while simultaneously maintaining a level of SOC. The system is capable of returning resources in both unobstructed and obstructed scenarios, confirming the inherent flexibility of swarming systems. Furthermore, the system is able to recharge agents in the meantime, with the number of charging activities dictated by the application. By being able to maintain a steady rate of return whilst also upholding continuous availability of energy, the sustained application of this system for foraging purposes is motivated.

### 6-3-2 Self-Organizing and Optimization

The second feature considered is the self-organization of the foraging system to achieve more optimal performance. As stated previously, it is both desired and expected that the system can identify and reinforce paths more optimal in terms of the cost function in Equation (6-1).

To assert and study this optimization tendency, the battery capacity of the agents is varied. The battery capacity is selected for two reasons: first, consider Equation (5-14). This equation describes the agent's tendency to weigh potential weight and SOC-usage. Substituting the definition of SOC (Equation (5-5)) into this, Equation (6-2) is attained.

$$\vec{v}^{\,s}_{f}(k) := \left\langle \sum_{b\in\mathcal{B}_m(k)} \rho_s \omega_b^{s_m(k)}(k) \;\; \rho_e \left[ 1 - \left( \frac{E_{battery}(k)}{E_{battery,capacity}} \right)_b^{s_m(k)} (k) \right] \;\; \vec{v}_b^{\,s_m(k)}(k) \right\rangle \tag{6-2}$$

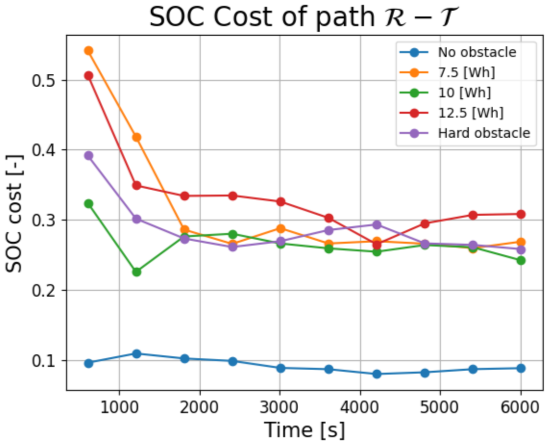
It can be concluded that changing the battery capacity results in an inverse effect on the severity of SOC costs. A similar effect can be achieved by tuning $\rho_f$ and $\rho_c$. This is omitted in this study for the second reason the battery capacity is selected, namely to study the effects of physical agent design. In application, rovers are limited in size and weight, hereby restricting the battery size. Varying this parameter allows for the effects of physical agent limitations to be deduced.

Given Equation (6-2), it is expected that the system makes more conservative choices with decreasing battery capacity. The same energy cost will have a larger effect on a smaller battery's SOC, meaning the agent will choose to avoid SOC-costs more actively. This is expected to lead to paths with lower energy expenses at the cost of longer travel distances and times. Agents with larger batteries on the other hand can incur more energy usage to follow paths with higher potential, likely resulting in shorter paths through soft obstacles.

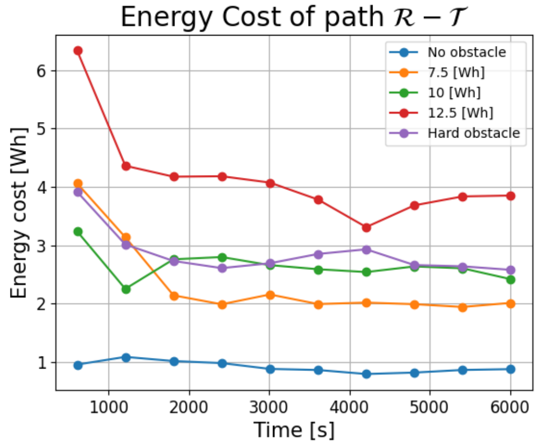
The standard battery capacity of 10 [Wh] as per Table 6-1 is varied with -25% and +25%. The soft-obstacle scenario is used as it allows the system to perform a dynamic trade-off between energy consumption and potential weight. Furthermore, the two previously introduced scenarios are included. The unobstructed scenario serves as a lower bound of the attainable path to assess how aggressively the increased system approaches this at the cost of energy. The hard-obstacle scenario on the other hand is considered the attainable path completely avoiding the soft obstacle. The dataset of Section 6-2 is used for its abundant foraging activity.

The figures in Figure 6-8 show four measures of the obstructed path $\mathcal{R} - \mathcal{T}$, namely the average SOC-cost of the path, the average actual energy cost, the average distance travelled along the path, and the average time needed to traverse it. The following can be deduced from the figures in Figure 6-8:

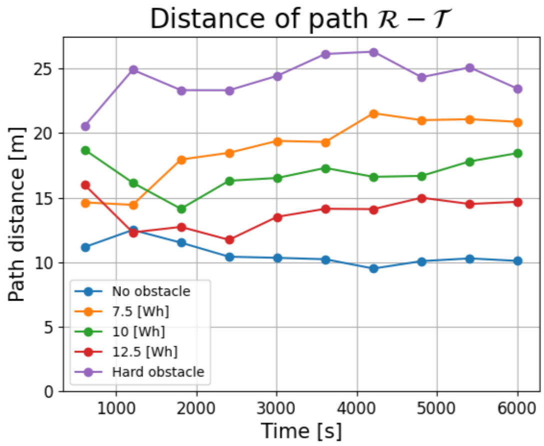
- All SOC-costs in Figure 6-8a converge to similar values just below 0.3. This is in line with Equation (6-2) given that the ratio between potential weight and SOC remains the same. Achieving the same equilibrium is therefore expected. The unobstructed scenario is found to be well below the value of others as no significant power obstacle is present
- When considering the actual energy spent to travel the path $\mathcal{R} - \mathcal{T}$ however, the system with the largest battery capacity is seen to spend the most energy, while the one with the least battery also spends the least energy. This is in line with the previous conclusion and the definition of SOC in Equation (5-5)
- Conversely, agents with the largest battery capacity are seen to travel the least amount of distance and time. Their values approach that of the unobstructed ideal path. Given the setup of the soft-obstacle scenario, this can only be achieved if the agents cross the soft obstacle, incurring the additional energy costs for directness benefit. This is therefore in line with the previous statement. Small capacity agents can be observed following a route approaching the length and time of the hard-obstacle case. This is expected to be a result of the system attempting to avoid the soft obstacle altogether, minimizing energy usage
- Given that the velocity of each rover is predetermined, the time required to travel the path as seen in Figure 6-8d follows Figure 6-8c



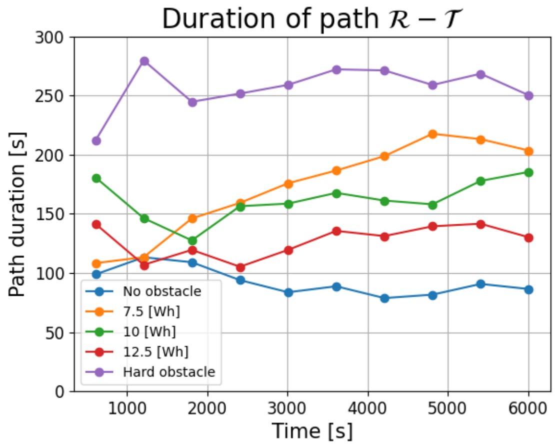

**(a)** SOC cost between $\mathcal{R}-\mathcal{T}$.




**(b)** Energy used between $\mathcal{R}-\mathcal{T}$.




**(c)** Distance travelled between $\mathcal{R}-\mathcal{T}$.



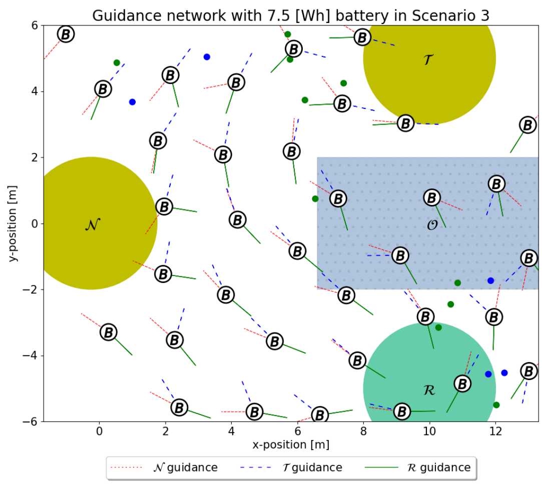

**(d)** Time required to travel between $\mathcal{R}-\mathcal{T}$.

**Figure 6-8:** Run performance with varying battery capacity.

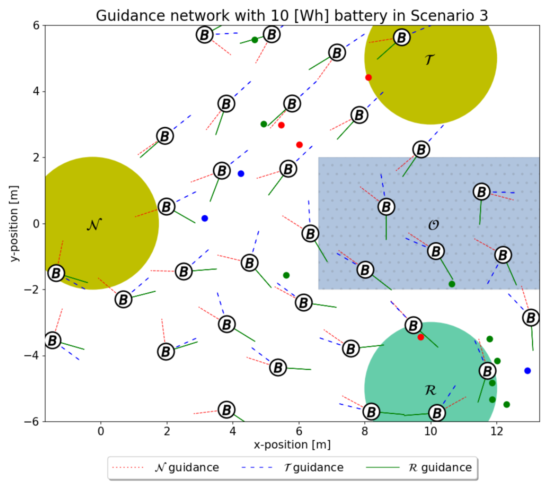
- Observe that the distance of the hard-obstacle scenario increases in distance and travel time as it progresses. This is attributed to agents introducing an additional offset to the obstacle. Paths from the exploration phase are developed from agents whose obstacle avoidance was triggered, a time and energy-inefficient function. With agents converging to a path with a larger offset, this mechanism is less often activated, leading to more efficient paths in both optimization aspects
- A noteworthy observation is that the hard-obstacle and 10[Wh] case use similar energy while travelling other distances. This is most likely a result of approximate following of routes and corner cutting as described in Section 6-2-2. Where systems in the soft-obstacle scenario can cut corners, hard-obstacle agents are led around their obstacle with an enforced clearance. This leads to additional path distance and obstacle-avoidance costs. The same holds for the small-battery case having less energy costs

Having made these observations, it can be concluded that the system actively takes part in optimizing the main route communicated by the guidance network. Using Equation (6-2), the system reinforces paths more fit for the system given its parameters. Furthermore, the path is observed to evolve over time, indicative of iterations being performed. By changing the battery capacity, the equilibrium agents strive for was influenced, resulting in changed system behaviour. By lowering the battery capacity, the weight of SOC-usage was increased, causing the system to find more conservative routes around obstacles. Agents with ample battery reserves and lowered SOC considerations allowed themselves more costly but direct paths. The system is therefore concluded to be self-organizing with tunable tendencies. Similar results can be achieved by changing the optimization weights in Equation (6-2).

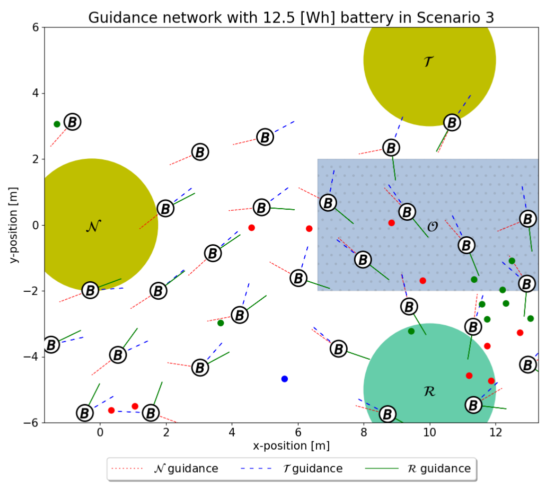
The same behaviour can also be observed when looking at screenshots of established guidance networks for varying battery capacity in Figure 6-9. Figure 6-9a shows the guidance network of the decreased-capacity system steering agents around the obstacle with ample clearance. Figure 6-9b depicts the system with more capacity allowing its agents to approach the obstacle and cross its boundaries incidentally. Finally, agents of Figure 6-9c with ample battery capacity develop a system with near-direct paths between all three regions, with few surrounding beacons required.




**(a)** Established guidance network for a battery capacity of 7.5 [Wh].




**(b)** Established guidance network for a battery capacity of 10 [Wh].



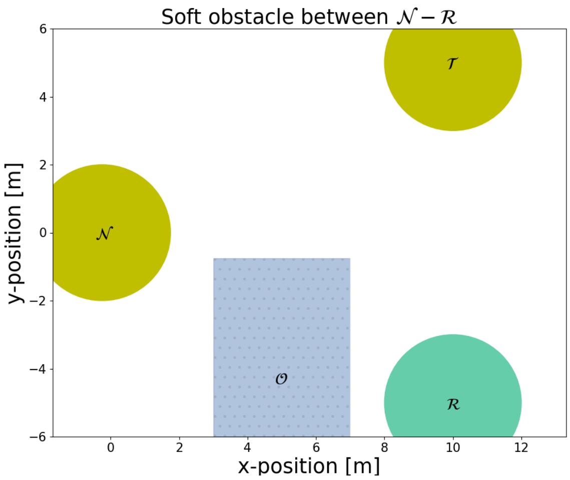

**(c)** Established guidance network for a battery capacity of 12.5 [Wh].

**Figure 6-9:** Guidance networks established with varying battery capacity.

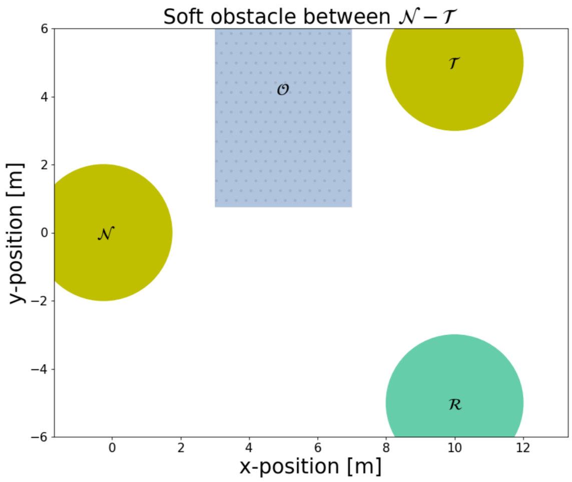
### 6-3-3 Adaptability to Environment

Finally, the effect of the obstacle in the search space is studied. This is done to assess the flexibility of the system for different problems and environments.

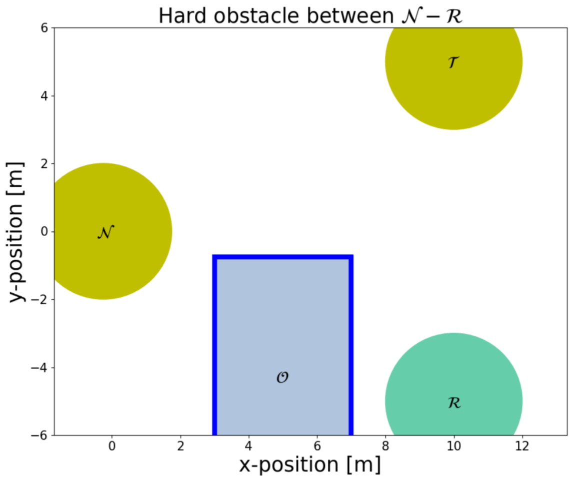
The system was applied in four additional scenarios, each with a different location for the obstacle. Whereas the hard obstacle as used up until now placed the region between target $\mathcal{T}$ and recharging region $\mathcal{R}$, the new scenarios place both soft and hard obstacles between $\mathcal{N}-\mathcal{T}$ and $\mathcal{N}-\mathcal{R}$. In doing so, multiple possible scenarios are tested analogous to either one. These scenarios can be seen in Figure 6-10.



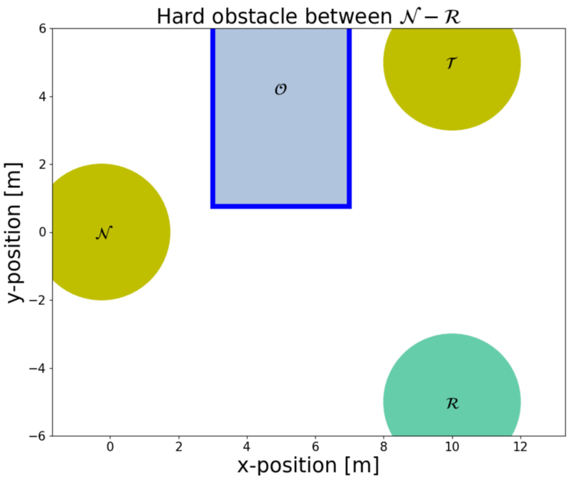

**(a)** Soft obstacle NR: scenario with $\mathcal{N}, \mathcal{T}, \mathcal{R}$ and soft obstacle $\mathcal{O}$ between $\mathcal{N}-\mathcal{R}$.




**(b)** Soft obstacle NT: scenario with $\mathcal{N}, \mathcal{T}, \mathcal{R}$ and soft obstacle $\mathcal{O}$ between $\mathcal{N}-\mathcal{T}$.




**(c)** Hard obstacle NR: scenario with $\mathcal{N}, \mathcal{T}, \mathcal{R}$ and hard obstacle $\mathcal{O}$ between $\mathcal{N}-\mathcal{R}$.

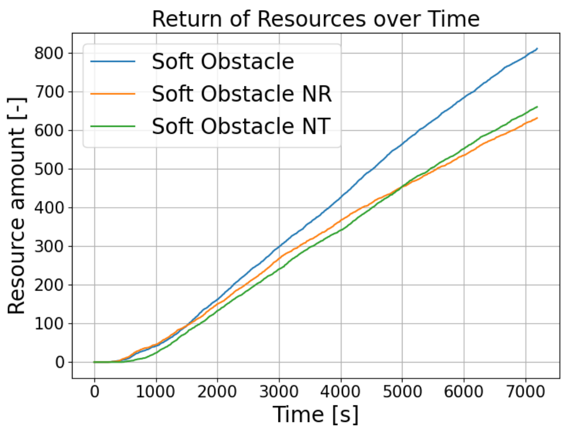



**(d)** Hard obstacle NT: scenario with $\mathcal{N}, \mathcal{T}, \mathcal{R}$ and hard obstacle $\mathcal{O}$ between $\mathcal{N}-\mathcal{T}$.

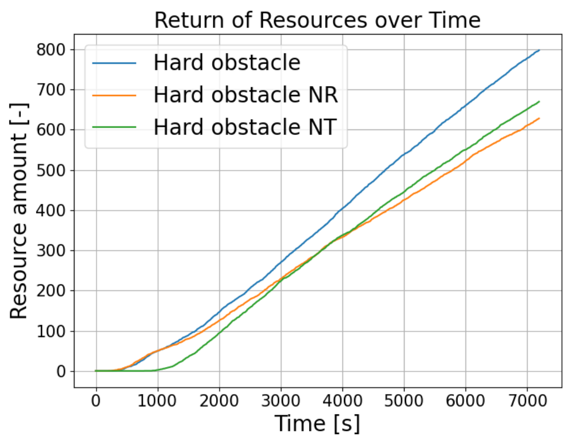
**Figure 6-10:** Additional scenario with varying obstacle location and resulting system performance.

Given the advantages of swarming systems, it is expected that the system will be able to perform in all six scenarios. In each scenario, the system should achieve a functional guidance network and facilitate foraging. However, given that recharging activities can trigger at any location, obstacles secluding $\mathcal{R}$ will have a major impact on the rate of return due to the increased path lengths and travel time. Furthermore, soft obstacles are expected to form a complete guidance network earlier, given that agents can travel through the region. Hard obstacles closer to the initialization point limit the region within which agents can explore, leading to more crowding and delayed system exploration.

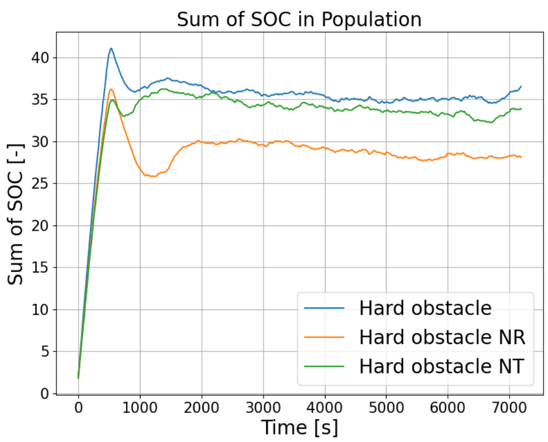
Figure 6-11 depicts the results of all six hard- and soft-obstacle scenarios. As expected, all six figures indicate that paths are established in beacons in the presence of all obstacle types and locations. These paths can be used by agents in a similar fashion to previous cases.



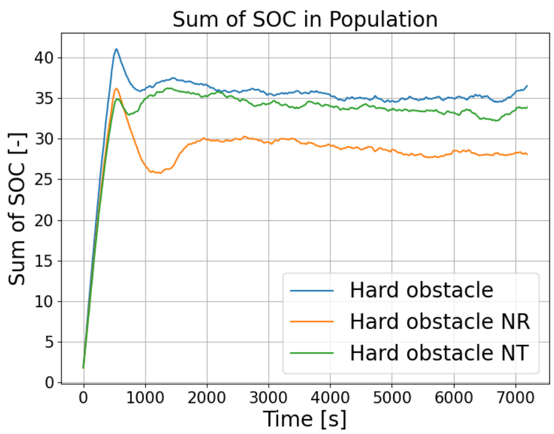

**(a)** Progression of resource return for varying soft obstacle locations.



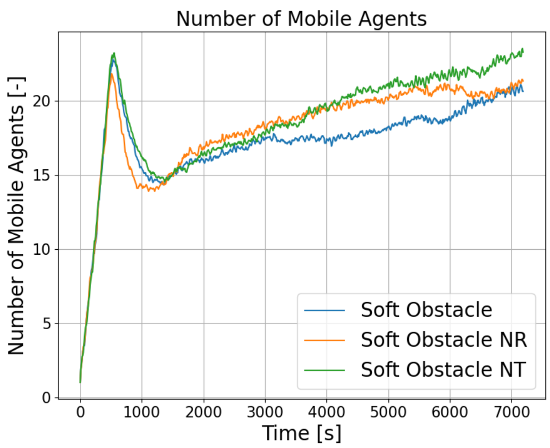

**(b)** Progression of resource return for varying hard obstacle locations.



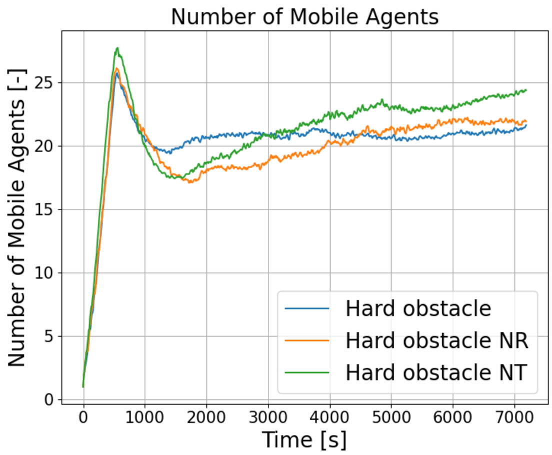

**(c)** Progression of population SOC for varying soft obstacle locations.




**(d)** Progression of population SOC for varying hard obstacle locations.




**(e)** Progression of agent utilization in mobile roles for varying soft obstacle locations.




**(f)** Progression of agent utilization in mobile roles for varying hard obstacle locations.

**Figure 6-11:** Additional scenario with varying obstacle location and resulting system performance.

From the figures in Figure 6-11, the following can be concluded:

- The first deduction that can be made is that the system succeeds in performing both foraging and housekeeping tasks. While Figure 6-11a and Figure 6-11b show ongoing foraging, Figure 6-11e and Figure 6-11f show active discharging and recharging, conform to results in Section 6-2
- Scenarios with obstacles furthest away from the nest perform best in terms of return rate. In both Figure 6-11a and Figure 6-11b, the initial scenarios outperform the rest. This is most likely due to the initialization space agents get to explore. The lack of crowding decreases the noise received at the start, resulting in more efficient pathways
- Obstacles placed between $\mathcal{N} - \mathcal{R}$ perform worst in terms of resource return rate. This is expected to be the result of the recharging region being more shielded, requiring mobile agents along the foraging path to be more conservative in terms of SOC
- The number of agents required to achieve and sustain the guidance networks as seen in Figure 6-11e and Figure 6-11f depicts the same behaviour as described in Section 6-2. However, scenarios with $\mathcal{N} - \mathcal{T}$ obstacles perform best. This is attributed to the fact that as recharging activities are triggered more often, the early development of the path $\mathcal{N} - \mathcal{R}$ and $\mathcal{R} - \mathcal{T}$ helps return agents to a mobile role. Increased swarm sizes would result in increased rates of return and faster settling on an optimal guidance network size as a result of increased rates of reinforcement

Having run the system in multiple scenarios, the main conclusion that can be made is that the system is able to perform both foraging and maintenance tasks in varying scenarios. Although some effect was expected, the difference in resource amounts returned between $\mathcal{R} - \mathcal{T}$ obstacle placement and the other two is significant. A probable explanation is over-conservatism with the recharging region being hard to reach. Furthermore, $\mathcal{N} - \mathcal{R}$ obstacles prove decreased efficiency in maintaining population SOC, also attributed to the system having increased difficulties reaching the recharging region $\mathcal{R}$.

## 6-4 Summary

This chapter discussed how the system was implemented in a simulated environment, how the system evolved in application, and if conclusive results could be attained to characterize the system performance.

The foraging system in question was implemented through the use *Python* code and a simulated environment in *Webots* as presented in Section 6-1. Agents were modelled as four-wheeled rovers, with a solar array and communication transceiver mounted on top. Furthermore, distance-sensors were used for obstacle-avoidance, and another transceiver acted as the resource-retrieving mechanism. All other features were integrated in code. The test-scenarios were set up manually and called from *Python* to ensure reproducibility.

The proposed system of Chapter 5 managed to depict the stages of foraging in Section 6-2 as expected. With agents initializing and leaving the nest, a guidance network was constructed. Using this to explore further, the regions $\mathcal{T}$ and $\mathcal{R}$ were detected. Once identified, the system was able to use this knowledge to support exploitation activities as well as recharging. The system was also observed to dynamically learn and optimize its guidance depending on system tendencies.

Through quantitative analysis over multiple simulated runs, it was concluded in Section 6-3-1 that the system does indeed perform as intended by theoretical architecture, even with the added functionality of visiting the recharging region. Not only did the system retrieve resources, it was seen to dynamically recharge its population. The self-organization of the network was measured in Section 6-3-2, with changing agent parameters affecting the type of path it promotes. Changing battery capacity alone had major impacts on the system choosing for more aggressive or conservative paths with respect to SOC. Finally, Section 6-3-3 confirmed that the system is able to operate in a variety of scenarios. Although the effect of obstacle placement between $\mathcal{N}$ and $\mathcal{R}$ is more significant than expected, the systems adaptiveness is discernable.

To conclude, it is both qualitatively and quantitatively proven that the system, as proposed by this report, is able to fulfil its maintenance tasks while foraging for resources and illustrating both flexibility and self-organization as can be expected from a foraging swarm.

# Chapter 7

# Conclusions

## 7-1 Conclusions of Proposed System Implementation

This research aimed to answer the question if foraging is viable for application with energetics considerations. Fundamental considerations for robot operation have gone unaddressed, limiting the maturity of this technology in its use. With this gap in knowledge in mind, a research question was formalized in Chapter 3. Having performed extensive system design to achieve conclusive results, this chapter will present these findings and propose areas of interest for future work.

This report selected energetics as the aspect to consider given it is a fundamental necessity of active agents. By fulfilling this system requirement through recharging mechanisms, the system's foraging performance would be supported in application. Achieving this would indicate the feasibility of implementation for means such as lunar resource exploitation. The main research question of this research therefore became:

*Can an applied energetic housekeeping strategy be developed for a beaconed ant-inspired foraging algorithm with the intent of resource gathering on the lunar surface?*

To answer this question, the baseline system of Chapter 4 was amended with additional functionality as presented in Chapter 5. These changes primarily addressed the possibility of an additional non-exclusive role for recharging activities. This required a change in the way information was gathered and handled in beacons. Furthermore, the agent's state in the form of State-of-Charge (SOC) now became an indicator for path quality. By providing energy costs as an additional term in an agent's experience, paths gained another quality value. With the utilities of foraging, these values could be used to not only ensure agents engage in recharging but also to reinforce routes beneficial in terms of energy costs. The functionality of these mechanisms is backed up by findings in Chapter 6, where Section 6-3-1 provided proof that the system remains functional over time even with the added necessity of recharging, Section 6-3-2 proved the population's tendency to adopt optimal paths, and Section 6-3-3 proving flexibility to environment in which the system is applied.

Having implemented the proposed amendments, the sub-questions of the problem statement can be answered.

- *What global tendencies are required to enable system maintenance from an energetics perspective?*
  A system should encapsulate personal maintenance tasks and provide agents the freedom to execute these. Losing agents leads to population degradation. To allow agents to disengage from their foraging tasks, additional roles are required. However, through the expansion of the role set, the individual's role selection becomes more complex. Furthermore, to make well-informed decisions, the interaction between agents also needs to be amended to allow ample knowledge throughput.
- *How should an individual's operation logic be structured to include personal maintenance in relation to energetics while contributing to population objectives?* The most significant change to agent logic is the added role of maintenance and with it the expanded role selection. The latter needs to be well-defined to allow for improved, pre-emptive decision-making. The selection of a role is based on both the availability of information in the guidance network and personal state. By promoting personal integrity through the decision-making of an agent, the same tendency is observed in the global swarm, making sure operation is guaranteed before performance is sought after.
- *How do optimality mechanisms within foraging affect population and individual operation with respect to energetics?*
  Having access to additional path quality indicators allows agents to make more well-informed decisions. In doing so, the requirements of an individual can be loosened. As proven by Section 6-3-2, this also amounts to the entire system following suit, optimizing its overall operations and decreasing its required resources. With additional optimization factors to tune, the behaviour of the population becomes more dynamic and tunable for desired tendencies.

Finally, this research is concluded by answering the main research question. Through this study, it was proven that foraging systems indeed have the potential to be used in application, provided their system state and role-switching are configured with it in mind. By introducing new roles for recharging, sustained functionality can be maintained. While achieving system performance both for foraging and maintenance, the desired characteristics of self-organizing and flexibility continue to emerge in application.

## 7-2 Recommendations for Future Work

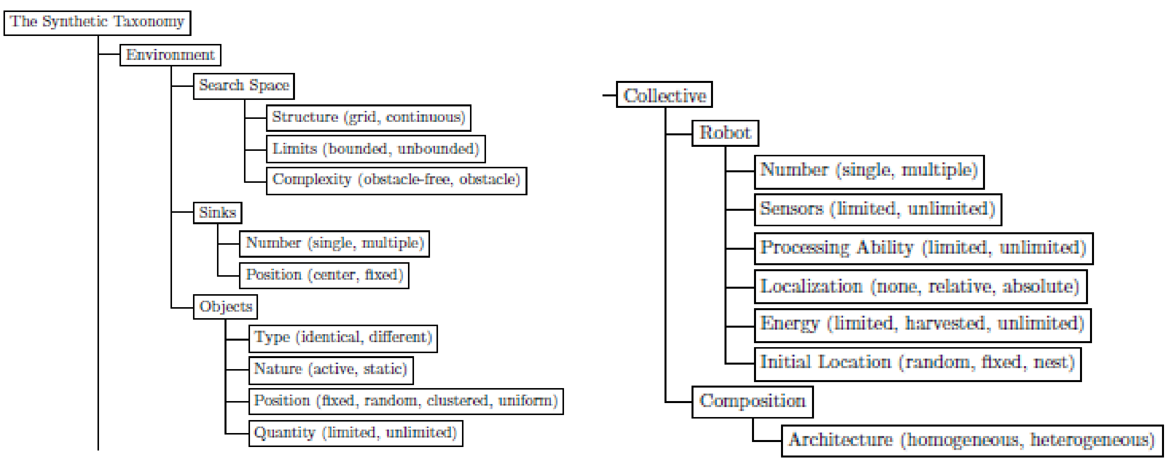
The development of the system aimed to tackle the most significant problems encountered when expanding the baseline foraging system to one fit for application. Although most issues were addressed, some were deemed too detailed for a feasibility study. Furthermore, as the results in Chapter 6 revealed, some inherent issues should be re-evaluated for system improvement. These are stated below:

- **Precise relative guidance**: as seen in the exploitation phase of Section 6-2-2 and Section 6-3, the system cannot follow paths closely. This is a result of the system's limited functionality of using global heading as guidance without location. To improve accuracy, relative positioning is proposed, where a beacon can use signal measurements during communication to determine a mobile agent's relative position and communicate a precise waypoint. In doing so both accuracy and system performance are increased

- **Optimizing swarm size**: this study attempted to introduce energy maintenance functionality. In doing so, it expanded the optimization function to that in Equation (6-1), which now provides additional optimization possibilities. As Figure 6-7d, Figure 6-11e, and Figure 6-11f indicate, the number of agents involved in the guidance network does not always converge within considerable time. Fine-tuning the optimization process will help trim the amount of beacons required and boost system performance
- **Discharging beacons**: the system in Chapter 5 assumes beacons discharge slowly, reaching critical SOC well beyond the scope of simulated runs. However, from a practical and robustness perspective, implementing mechanisms to exchange roles between beacons and mobile agents will not only allow beacons to maintain a required SOC but will also showcase that beacons can be replaced in the event of a malfunction
- **Improved decay**: one of the most problematic features in the system is the method of decaying. Although addressed, the possibility of intermediate beacons decaying exists. This removes continuity in network branching and may leave agents stranded. Implementing a more reliable decay strategy will result in increased robustness and agent utilization
- **Establishing search-space boundaries**: all scenarios used to gather simulation data had a boundary around its search-space. This was done for simulation-time purposes, as agent dispersion would only delay the discovery of regions of interest. In practicality, such boundaries can be formed around a known search space using border-agents fencing others into a region to explore

# Appendix A

# Foraging Taxonomy



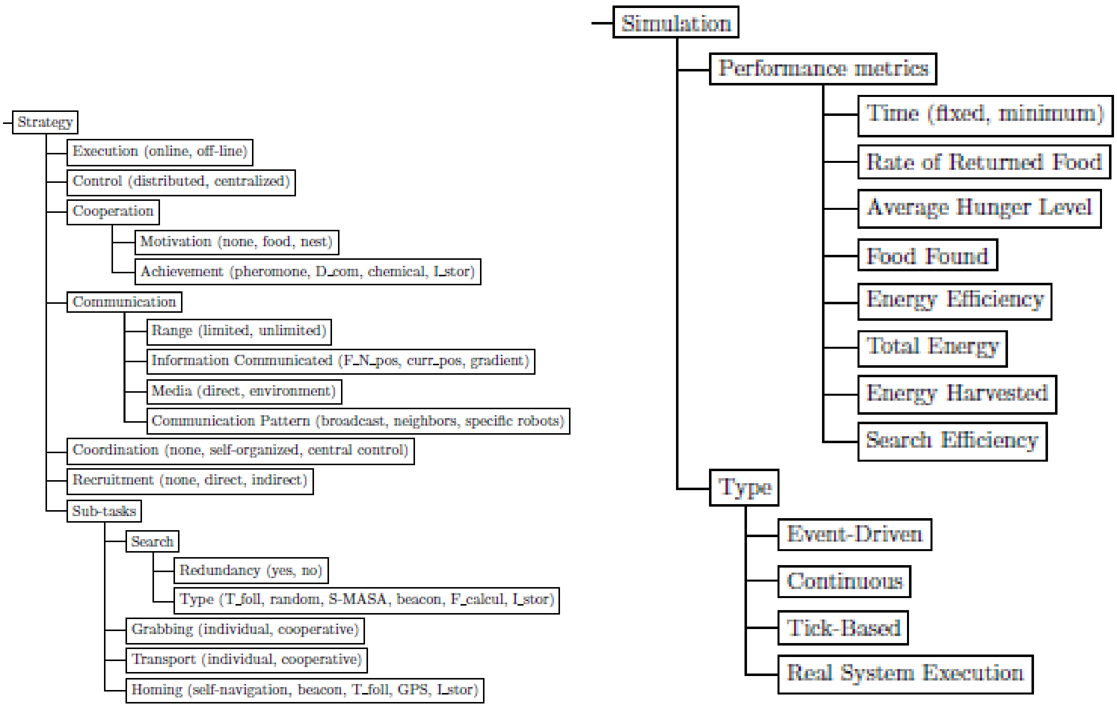



**Figure A-1:** Graphical representation of proposed taxonomy as per [50].

# Bibliography

[1] Steven Adams. A stigmergy-based design for a minimalistic foraging robotic swarm, 2021.

[2] Steven Adams, Daniel Jarne Ornia, and Manuel Mazo Jr. A self-guided approach for navigation in a minimalistic foraging robotic swarm. *CoRR*, abs/2105.10331, 2021.

[3] Bibek Adhikari, Rakesh Yadav, Jinaykumar Patel, and Kamesh Subbarao. Lunar exploration through co-operative rovers using smart navigation. In *AIAA SCITECH 2024 Forum*, page 1874, 2024.

[4] National Aeronautics and Space Administration (NASA). National space exploration campaign report, Sep 2018. Available at https://www.nasa.gov/sites/default/files/atoms/files/nationalspaceexplorationcampaign.pdf.

[5] M. Anand, I.A. Crawford, M. Balat-Pichelin, S. Abanades, W. van Westrenen, G. Péraudeau, R. Jaumann, and W. Seboldt. A brief review of chemical and mineralogical resources on the moon and likely initial in situ resource utilization (isru) applications. *Planetary and Space Science*, 74(1):42–48, 2012. Scientific Preparations For Lunar Exploration.

[6] Z. T. Angyal. Literature study: Nature-inspired foraging of ground measurements and resource deposits on the lunar surface, 2023.

[7] Alex Austin, Brent Sherwood, John Elliott, Anthony Colaprete, Kris Zacny, Philip Metzger, Michael Sims, Harrison Schmitt, Sandra Magnus, Terry Fong, Miles Smith, Raul Polit Casillas, A. Scott Howe, Gerald Voecks, Mar Vaquero, and Vincent Vendiola. Robotic lunar surface operations 2. *Acta Astronautica*, 176:424–437, 2020.

[8] Andrew B. Barron and Jenny Aino Plath. The evolution of honey bee dance communication: a mechanistic perspective. *Journal of Experimental Biology*, 220(23):4339–4346, 12 2017.

[9] Tim M. Blackwell and James Kennedy. Impact of communication topology in particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 23:689–702, 2019.

[10] E Bonabeau and C Meyer. Swarm intelligence. a whole new way to think about business. *Harvard Business Review*, 79(5):106–14, 2001.

[11] J.D. Carpenter, R. Fisackerly, D. De Rosa, and B. Houdou. Scientific preparations for lunar exploration with the european lunar lander. *Planetary and Space Science*, 74(1):208–223, 2012. Scientific Preparations For Lunar Exploration.

[12] Thomas Collett and Matthew Collett. How do insects represent familiar terrain? *Journal of physiology, Paris*, 98:259–64, 06 2004.

[13] Ian A. Crawford. Lunar resources. *Progress in Physical Geography: Earth and Environment*, 39(2):137–167, 2015.

[14] Jean Louis Deneubourg, Serge Aron, S. Goss, and Jacques Pasteels. The self-organizing exploratory pattern of the argentine ant. *J. Insect Behav.*, 3:159, 03 1990.

[15] Kurt Derr and Milos Manic. Multi-robot, multi-target particle swarm optimization search in noisy wireless environments. pages 81 – 86, 06 2009.

[16] Matthew V. D'Ortenzio, John L. Bresina, Alan R. Crocker, Richard C. Elphic, Ken F. Galal, David R. Hunt, Brandon D. Owens, Laura Plice, Alisa M. Hawkins, and Lisa A. Policastri. Operating ladee: Mission architecture, challenges, anomalies, and successes. In *2015 IEEE Aerospace Conference*, pages 1–23, 2015.

[17] Frederick Ducatelle, Gianni Di Caro, Carlo Pinciroli, Francesco Mondada, and Luca Maria Gambardella. Communication assisted navigation in robotic swarms: Self-organization and cooperation. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 484, 01 1015.

[18] Alex Ellery. Sustainable in-situ resource utilization on the moon. *Planetary and Space Science*, 184:104870, 2020.

[19] Andrew Farley, Jie Wang, and Joshua Marshall. How to pick a mobile robot simulator: A quantitative comparison of coppeliasim, gazebo, morse and webots with a focus on the accuracy of motion simulations. *Simulation Modelling Practice and Theory*, 120:102629, 07 2022.

[20] Abel John George. A short review on india's interplanetary missions. *Acceleron Aerospace Journal*, 2(1):138–147, 2024.

[21] Francis Heylighen. Stigmergy as a universal coordination mechanism i: Definition and components. *Cognitive Systems Research*, 38:4–13, 2016. Special Issue of Cognitive Systems Research – Human-Human Stigmergy.

[22] Nicholas R. Hoff, Amelia Sagoff, Robert J. Wood, and Radhika Nagpal. Two foraging algorithms for robot swarms using only local communication. *2010 IEEE International Conference on Robotics and Biomimetics*, 2010.

[23] Yingzhuo Jia, Yongliao Zou, Jinsong Ping, Changbin Xue, Jun Yan, and Yuanming Ning. The scientific objectives and payloads of chang'e4 mission. *Planetary and Space Science*, 162:207–215, 2018. Lunar Reconnaissance Orbiter - Seven Years of Exploration and Discovery.

[24] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948 vol.4, 1995.

[25] James Kennedy. *Swarm Intelligence*, pages 187–219. Springer US, Boston, MA, 2006.

[26] James Kennedy and Rui Mendes. Population structure and particle swarm performance. volume 2, pages 1671 – 1676, 02 2002.

[27] Harald Köpping Athanasopoulos. The moon village and space 4.0: The 'open concept' as a new way of doing space? *Space Policy*, 49:101323, 2019.

[28] Nyree Lemmens. *Bee-inspired distributed optimization.* PhD thesis, Maastricht University, January 2011.

[29] Qunfeng Liu, Wenhong Wei, Huaqiang Yuan, Zhi-Hui Zhan, and Yun Li. Topology selection for particle swarm optimization. *Information Sciences*, 363:154–173, 2016.

[30] Zhaoqin Liu, Kaichang Di, Jian Li, Jianfeng Xie, Xiaofeng Cui, Luhua Xi, Wenhui Wan, Man Peng, Bin Liu, Yexin Wang, and et al. Landing site topographic mapping and rover localization for chang'e-4 mission. *Science China Information Sciences*, 63(4), 2020.

[31] Pattie Maes, Maja J. Mataric, Jean-Arcady Meyer, Jordan Pollack, and Stewart W. Wilson. *Robotic "Food" Chains: Externalization of State and Program for Minimal-Agent Foraging*, pages 625–634. 1996.

[32] Marco Mamei and Franco Zambonelli. Spreading pheromones in everyday environments through rfid technology. *Swarm Intelligence*, 01 2005.

[33] Lino Marques and Aníbal de Almeida. Toward chemical-trail following robots. pages 489–494, 12 2008.

[34] IG Mitrofanov, LM Zelenyi, VI Tret'yakov, and DV Kalashnikov. Luna-25: The first polar mission to the moon. *Solar System Research*, 55:485–495, 2021.

[35] David E. Morgan. Trail pheromones of ants. *Physiological Entomology*, 34(1):1–17, 2009.

[36] Martin Müller and Rüdiger Wehner. Path integration in desert ants, cataglyphis fortis. *Proceedings of the National Academy of Sciences*, 85(14):5287–5290, 1988.

[37] Shervin Nouyan and Marco Dorigo. Chain based path formation in swarms of robots. volume 4150, pages 120–131, 09 2006.

[38] Lenka Pitonakova, Manuel Giuliani, Anthony Pipe, and Alan Winfield. Feature and performance comparison of the v-rep, gazebo and argos robot simulators. 02 2018.

[39] Jim Pugh and Alcherio Martinoli. Inspiring and modeling multi-robot search with particle swarm optimization. pages 332–339, 2007.

[40] Guenther Reitz, Thomas Berger, and Daniel Matthiae. Radiation exposure in the moon environment. *Planetary and Space Science*, 74(1):78–83, 2012. Scientific Preparations For Lunar Exploration.

[41] R.A. Russell. Heat trails as short-lived navigational markers for mobile robots. In *Proceedings of International Conference on Robotics and Automation*, volume 4, pages 3534–3539 vol.4, 1997.

[42] Erol Sahin, Sertan Girgin, Levent Bayindir, and Ali Turgut. *Swarm Robotics*, pages 87–100. 01 2008.

[43] Julie Schonfeld. Summary of the contracted deliveries of nasa payloads to the moon via commercial lunar payload services (clps). In *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 863–866. IEEE, 2023.

[44] Brent Sherwood. Principles for a practical moon base. *Acta Astronautica*, 160:116–124, 2019.

[45] David E. Smith, Maria T. Zuber, Gregory A. Neumann, Erwan Mazarico, Frank G. Lemoine, James W. Head III, Paul G. Lucey, Oded Aharonson, Mark S. Robinson, Xiaoli Sun, Mark H. Torrence, Michael K. Barker, Juergen Oberst, Thomas C. Duxbury, Dandan Mao, Olivier S. Barnouin, Kopal Jha, David D. Rowlands, Sander Goossens, David Baker, Sven Bauer, Philipp Gläser, Myriam Lemelin, Margaret Rosenburg, Michael M. Sori, Jennifer Whitten, and Timothy Mcclanahan. Summary of the results from the lunar orbiter laser altimeter after seven years in lunar orbit. *Icarus*, 283:70–91, 2017. Lunar Reconnaissance Orbiter - Part II.

[46] Christian Stenzel, Lukas Weiss, and Thomas Rohr. Sustainable challenges on the moon. *Current Opinion in Green and Sustainable Chemistry*, 9:8–12, 2018. Selected Papers from the 2nd Green and Sustainable Chemistry conference.

[47] Ying Tan and Zhong-yang Zheng. Research advance in swarm robotics. *Defence Technology*, 239, 03 2013.

[48] R.K. Tripathi, J.W. Wilson, F.F. Badavi, and G. De Angelis. A characterization of the moon radiation environment for radiation analysis. *Advances in Space Research*, 37(9):1749–1758, 2006. Space Life Sciences: Flight Measurements, Calibration of Detectors and Environmental Models for Radiation Analysis.

[49] Long Xiao, Yuqi Qian, Qian Wang, and Qiong Wang. The chang'e-5 mission. *Sample Return Missions*, page 195–206, 2021.

[50] Ouarda Zedadra, Nicolas Jouandeau, Hamid Seridi, and Giancarlo Fortino. Multi-agent foraging: State-of-the-art and research challenges. *Complex Adaptive Systems Modeling*, 5(1), 2017.

# Glossary

## List of Acronyms

| | |
|---|---|
| **ESA** | European Space Agency |
| **NASA** | National Aeronautics and Space Administration |
| **CNSA** | China National Space Administration |
| **ISRO** | Indian Space Research Organisation |
| **JAXA** | Japan Aerospace Exploration Agency |
| **SLIM** | Smart Lander for Investigating Moon |
| **CADRE** | Cooperative Autonomous Distributed Robotic Exploration |
| **RLSO** | Robotic Lunar Surface Operations |
| **PSR** | Permanently Shadowed Region |
| **ISRU** | In-Situ Resource Utilization |
| **SI** | Swarm Intelligence |
| **LRO** | Lunar Reconnaissance Orbiter |
| **PI** | Path Integration |
| **BMS** | Battery Management System |
| **SOC** | State-of-Charge |
| **RFID** | Radio Frequency Identification |

## List of Symbols

| | | |
|---|---|---|
| $a$ | | Agent in the swarm |
| $\mathcal{A}$ | | Set of agents in swarm |
| $A_{array}$ | $[m^2]$ | Solar array surface area |
| $B$ | | Beacon role |
| $\mathcal{B}$ | | Set of agents in beacon role |
| $\mathcal{B}_m$ | | Set of mobile agents in a beacon's communication region |
| $\mathcal{C}_i$ | | Communication region of object $i$ |
| $d_{backtrack}$ | [m] | Backtrack distance |
| $\mathcal{D}$ | | Search space domain |
| $\delta_i$ | [m] | Radius of object $i$ |
| $E_{battery}$ | [Wh] | Energy stored in battery |
| $E_{battery,capacity}$ | [Wh] | Battery capacity |
| $\varepsilon$ | [-] | Probability of exploration |
| $\eta_{array}$ | [-] | Solar array efficiency |
| $\eta_{solar}$ | $[W/m^2]$ | Threshold of solar irradiance for recharging region $\mathcal{R}$ |
| $\eta_w$ | [-] | Lower threshold indicating decay |
| $\emptyset$ | | Empty set |
| $F_{in}$ | | Inbound foraging role |
| $F_{out}$ | | Outbound foraging role |
| $\mathcal{G}$ | | Region covered by the guidance network |
| $\gamma$ | [-] | State-potential |
| $I_{motor}$ | [A] | Motor current |
| $Irr_{i,solar}$ | $[W/m^2]$ | Solar irradiance level striking object $i$ |
| $k$ | | Discrete time |
| $k^-$ | | Discrete time of last role switch |
| $k_{switch}$ | | Discrete time of role switch |
| $k_v$ | [RPM/V] | Motor rpm-voltage constant |
| $k_I$ | [Nm/A] | Motor torque-current constant |
| $l_i$ | [m] | Length of object $i$ |
| $\lambda$ | [-] | Diffusion rate of state-potential |
| $\mathcal{M}$ | | Set of agents in mobile role |
| $\mathcal{M}_b$ | | Set of beacons in a mobile agent's communication region |
| $N$ | | Number of agents in swarm |
| $\mathcal{N}$ | | Nest region |
| $n_{failed\ attempts}$ | [-] | Number of failed attempts to reach the destination |
| $\mathcal{O}$ | | Obstacle region |
| $\omega_b^s$ | [-] | Potential weight in a beacon $b$ for guidance of role $s$ |
| $\omega_{motor}$ | [RPM] | Motor rotation speed |
| $P_{avoidance}$ | [W] | Power used for obstacle avoidance |
| $P_{computation}$ | [W] | Power used for communication |
| $P_{exploit}$ | [W] | Power used for exploitation operations |
| $P_{in}$ | [W] | Power gained |
| $P_{motion}$ | [W] | Power used for motion |
| $P_{out}$ | [W] | Power used |

| | | |
|---|---|---|
| $P_{scan}$ | [W] | Power used for environment scanning |
| $\phi_{SOC}$ | [-] | Safety factor applied to path SOC |
| $R$ | | Recharging role |
| $\mathcal{R}$ | | Recharging region |
| $r$ | [-] | Reward for completing a role |
| $r_i$ | [-] | Resource content of object $i$ |
| $\rho_c$ | [-] | Optimization weight of path SOC $SOC_b$ |
| $\rho_f$ | [-] | Optimization weight of potential weight $\omega_b$ |
| $\rho_v$ | [-] | Guidance vector evaporation rate |
| $\rho_w$ | [-] | Potential weight evaporation rate |
| $SOC_a$ | [-] | State-of-Charge of agent $a$ |
| $SOC_b^{s_c}$ | [-] | SOC cost stored in beacon $b$ for a path for completed role $s_c$ |
| $s_a$ | | Role of agent $a$ |
| $s_a^-$ | | Agent $a$'s role prior to the current role |
| $s^c$ | | Last successfully completed role |
| $\bar{s}_f$ | | Foraging agent $f$'s opposite foraging role |
| T | [s] | Update interval |
| $\mathcal{T}$ | | Target region |
| $T_{motor}$ | [Nm] | Motor torque |
| $t$ | [s] | Continuous time |
| $\tau$ | [s] | Sampling time |
| $\theta_a$ | [rad] | Heading of agent $a$ |
| $\theta_{random}$ | [rad] | Random heading |
| $\theta_{avoidance}$ | [rad] | Heading adjustment from obstacle avoidance mechanism |
| $U_{motor}$ | [V] | Motor voltage |
| $v_0$ | [m/s] | Agent velocity |
| $\overrightarrow{v_a}$ | [m/s] | Velocity vector of agent $a$ |
| $\overrightarrow{v_b}^s$ | [m/s] | Guidance vector in a beacon $b$ for guidance of role $s$ |
| $\overrightarrow{\hat{v}_f}^s(k)$ | [m/s] | Normalized guidance vector of foraging agent $f$ |
| $w_i$ | [m] | Width of object $i$ |
| $x_i$ | [m] | Location of object $i$ in domain $\mathcal{D}$ |
| $\xi_i$ | [-] | Friction coefficient of the surface object $i$ |