

Swarm Robotics: Exploration and Mapping in Simulated testing Environments

Charlie Anthony [candNo: 246537]

Supervisor: Dr Chris Johnson

Interim report

Computer Science and Artificial Intelligence BSc



**UNIVERSITY
OF SUSSEX**

Department of Informatics and Engineering

University of Sussex

November 2023

Contents

1	Introduction	2
2	Professional and Ethical Considerations	2
3	Related Work	2
3.1	SLAM	2
3.2	Graph-based SLAM	3
3.3	Particle Filters	3
3.4	Swarm	3
3.5	Random Walks	4
4	Requirements Analysis	4
5	Project Plan	5
6	Supervisor Meetings	5
6.1	Meeting 1 - 11/10/2023	5
6.2	Meeting 2 - 27/10/2023	6
7	Appendices	6
8	References	6

1 Introduction

Swarms exist everywhere in life. Nearly all organisms exhibit some form of swarming behaviours within their communities. Starlings display impressive organisational behaviour, positioning themselves with respect to the movement of their neighbours [?]. Humans show swarm behaviours when moving in crowds, for example, moving around sports venues or exiting buildings in emergencies. No matter how hard you look, regardless if the context, swarms are typically present.

These behaviours can also be artificially created in robotics [1]. Within the realm of computing, parallelising processes is breaking barrier after barrier - swarm robotics brings the same benefits. Being able to divide and conquer a problem has the ability massively increase the rate of work by employing multiple robots. Therefore, it would be wasteful not to properly dedicate the time which this discipline deserves.

For my project, I am going to try and reproduce some of these behaviours artificially. I will start by simulating robotic agents in a 2D environment. The agents will be placed within close proximity inside a simulated environment and then allowed to explore and combine their findings; ultimately creating a visualization map of its environment. The agents will need to both navigate the environment and avoid collisions, whilst creating an internal representation of its surroundings. The best-case scenario for the agents within the swarm is to be fully independent; creating a decentralized system.

I will initially explore this problem by creating SLAM simulations, and then attempting to apply similar techniques to a centralised system. These initial simulations will employ techniques such as graph-based SLAM, random walks and particle filters in order to create a base-line representation of the environment. This project will also have the flexibility to potentially implement physical robots, given time permits.

2 Professional and Ethical Considerations

My project maintains compliance towards all ethical considerations, as there is minimal external involvement from humans. The majority of my project will be carried out in simulation, therefore no ethical approval is required. Should my project progress to physically implementing agents, considerations such as safety around the robots, will be considered. All tests will be carried out in an environment where people cannot be hit, therefore mitigating any trip hazards.

Research in this project is within my professional competence, as it significantly relies upon knowledge obtained from modules such as "Acquired Intelligence and Adaptive Behaviour" and "Fundamentals of Machine Learning." I will further ensure all relevant gaps in knowledge are explored through reading extensively in the area and communicating any areas of concern with my supervisor.

3 Related Work

3.1 SLAM

SLAM (Simultaneous Localisation and Mapping) is a technique used in robotics to create a map of an unknown environment. It is an important area of research in robotics as it

is heavily used in autonomous vehicles, drones and vacuum cleaners; allowing agents to understand and navigate their environment effectively.

SLAM can be broken down into two sub-problems: localisation and mapping. Localisation is the process of determining the location of a robot in its environment, whilst mapping is the process of constructing a map of the environment. The maps are constructed using data collected from sensors, such as cameras and laser scanners. A lot of existing work in SLAM is based on single robot applications, however, there is a growing interest in multi-agent SLAM. One of the greatest challenges in SLAM is crossing the simulation to reality gap, as in the real world, sensor readings are noisy and environments are dynamic, which increases the complexity of the problem.

3.2 Graph-based SLAM

Graph-based SLAM is a technique used to create a map of an environment, by using a graph to represent the environment. It works by having a robot move around its environment, whilst taking measurements of its surroundings. These measurements are usually received by a sensor, such as a camera or laser scanner. The robot then uses these measurements to create a plot of where objects may be, by combining the measurements from sensors with its memory of the route it has taken. This technique is used in many applications, such as autonomous vehicles

One limitation of graph-based SLAM is that it is computationally expensive, as it requires a lot of memory to store the sensor readings. Also, it is not very scalable, as the more sensors that are added, the more memory is required. Another limitation is that it cannot always be reliable, as the algorithm depends heavily on detecting loop closures, which when not detected, can lead to a lot of errors in the map. This, combined with even the slightest inaccuracy from sensors/motors makes it difficult to apply to the real world.

3.3 Particle Filters

Another technique used in SLAM is particle filters. Particle filters, also known as Monte Carlo localisation, is a probabilistic technique used to estimate the localisation of a robot in its environment. It starts with a set of particles randomly distributed across the environment, which represent possible locations of the robot. Then as the robot moves around the environment, the particles weights are updated based on the robots sensors. Overall, this technique is very effective, as it is able to localise the robot in its environment, even when the environment is dynamic. Also, this process is highly parallelisable, as each particle can be updated independently; which makes this a viable solution for real-world applications. However, particle filters are susceptible to the "curse of dimensionality", which means that as the number of dimensions increases, the problem's complexity rapidly increases.

3.4 Swarm

Swarm robotics [1] is a discipline which studies the coordination of large numbers of robots. It is largely inspired by biology, where social organisms achieve complex behaviours through simple interactions with each other and the environment. Swarm robotics is an important area of research in robotics, as it has many applications, such as search and rescue, exploration and mapping.

One of the benefits of swarm robotics is the scalability and flexibility of the system. This is because the system is decentralized, meaning that each agent is able to make decisions independently. This allows for repeatability, as the system can be scaled up or down simply by adding or removing agents. Also, it allows for flexibility, as the system can be adapted to different environments, as each agent is able to make decisions based on its surroundings.

One of the biggest challenges in swarm robotics is the simulation to reality gap. This is because in simulation, the agents are able to easily share information with each other, whilst in the real world, this is more challenging. Furthermore, swarm robotics struggles with more complex environments, like outdoors. As a result, swarm robotics still has a lot of room for research and development.

3.5 Random Walks

Random walk exploration in the context of swarm mapping is a technique where agents individually map an environment, using methods like Graph-based SLAM, Particle Filters, etc. and then combine their findings to a single global map; this is an example of a centralised system.

A example of an implementation of random walk exploration is Brownian motion [2], which is a physics-inspired approach. It works by applying a random force to each agent, which determines its direction. The agents then move in this direction until they detect an obstacle, at which point they will change direction. This process is repeated until the environment is fully mapped. There is randomness provided from the environment, through detecting other agents or obstacles, and randomness in motion, as each path is determined by a random force. Overall, collective behaviour emerges from these simple rules, which lead to a global behaviour which efficiently maps an environment.

One of the biggest drawbacks of this approach is that it is not scalable, as the agents are not able to communicate their maps to each other, which can lead to a lot of redundant exploration [3]. This is because equally, sharing their maps with each other would be very computationally expensive. Also, another limitation is that this approach doesn't guarantee efficiency. This is because of the inherent randomness, which means areas of the environment may be unexplored.

4 Requirements Analysis

Table 1 shows the requirements for my project, along with their justification. Initially, I will create a simulation interface, where the user can see the agents, the environment and a representation of the agents internal map. After I will work towards implementing SLAM and swarm algorithms.

As an extension, I will attempt to implement physical agents, which will act in the real world. This will require a lot of additional work, so will only be attempted if time permits.

When creating the simulation interface, I will be using the Python programming language, along with the PyGame library. This will abstract away a lot of the complexity of creating a graphical user interface, allowing me to focus on the core functionality. I will also use various other scientific python libraries throughout my project, such as NumPy,

SciPy and Matplotlib.

ID	Requirement	Justification
1	Simulation Interface	A graphical user interface, where the user can see the agent, the environment, and a representation of the agents internal map
2	SLAM Implementation	Implement a SLAM algorithm, which allows a single agent to localise and map the environment
3	Customise Environment	Allow the environment to be easily modified in the interface - perhaps by uploading a black and white image
4	Centralized Swarm	Implement a centralized swarm algorithm, where the agents individually collect data and then feed to a global map
5	Decentralized Swarm	Implement a decentralized swarm algorithm, where the agents communicate their findings and create individual representations of the environment

Table 1: Requirements and their justification

5 Project Plan

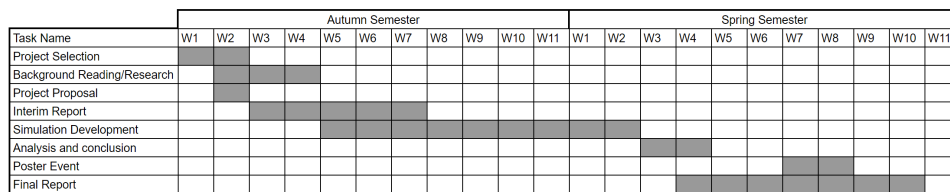


Figure 1: Gantt chart showing the project plan

The execution of my project will be split into various phases, where each phase will focus on an area of development. The majority of the project will be software development, therefore a large portion time will be spent here. Figure 1 shows the project plan, where the grey bars represent the time spent at each phase.

6 Supervisor Meetings

6.1 Meeting 1 - 11/10/2023

Discussed on the project idea and potential directions to take. Discussed the possibility of implementing physical agents, challenges that may occur and potential ways of implementing swarm algorithms. Need to focus on researching SLAM and swarm and looking into existing resources.

6.2 Meeting 2 - 27/10/2023

Discussed potential algorithms, such as particle filters and graph-based SLAM. We also discussed the logistics of the project, ensuring that it remains both realistic and achievable. We also discussed the possibility of implementing physical agents, and where relevant resources could be found.

7 Appendices

8 References

References

- [1] H. Hildenbrandt, C. Carere, C.K. Hemelrijk *Self-organized aerial displays of thousands of starlings: a model* <https://doi.org/10.1093/beheco/arq149>
- [2] Navarro, I., Matía, F. *An Introduction to Swarm Robotics, 2013* <https://doi.org/10.5402/2013/608164>
- [3] Khmel'nitsky, E. *Brownian Motion and Swarm Dynamics. In Autonomous Mobile Robots and Multi-Robot Systems, 2019* <https://doi.org/10.1002/9781119213154.ch12>
- [4] Kegeleirs, M., Garzón Ramos, D., Birattari, M. *Random Walk Exploration for Swarm Mapping, 2019* https://doi.org/10.1007/978-3-030-25332-5_19