



Bachelor Thesis

Stigmergic interaction in robotic multi-agent systems using virtual pheromones

ERIC JANSSON & JONAS SCHARIN
Computer Science & Engineering

Studies from the School of Science and Technology at Örebro University



Eric Jansson & Jonas Scharin

Stigmergic interaction in robotic multi-agent systems using virtual pheromones

Supervisor: Franziska Klügl, AASS

Examiner: Lars Karlsson

Abstract

Swarm robotics is a field of study focusing on achieving swarm intelligence within robotic multi-agent systems. It has potential applications within disaster recovery, defense, inspection, mapping, farming and space systems. This thesis investigates the possibilities of achieving swarm intelligence in a robotic-multi agent system of Kilobots through the integration of virtual pheromones. The virtual pheromones are established with the use of a projector and a camera. Combining the hardware with a developed software allows the position of the Kilobots to be calculated using data retrieved from the camera and virtual pheromones to be displayed accordingly through the projector. The approach taken to establishing virtual pheromones shows promise. However, challenges such as ambient light sensor accuracy and arena configuration limit the capabilities of the proposed system.

Keywords

Swarm Intelligence, Swarm Robotics, Swarm Behavior, Stigmergy, Virtual Pheromones, Trail Pheromone, Kilobot, Follow the leader, Multi-agent system, Computer Science, Computer Engineering

Contents

1	Introduction	4
1.1	Problem Formulation	4
1.2	Division of Work	5
1.3	Outline	5
2	Swarm Robotics	6
2.1	Swarming Behaviors	6
2.2	Stigmergy in Swarm Robotics	7
2.2.1	Storing Digital Pheromones in RFID Tags	8
2.2.2	Virtual Pheromones Using Projector	9
2.3	Extending the Capabilities of Swarm Robotic Systems	10
2.3.1	Augmented Reality for Kilobots	10
2.3.2	Kilogrid	11
3	Architecture	12
3.1	Environment Architecture	12
3.2	Software Architecture	12
4	Hardware	14
4.1	Kilobot	14
4.1.1	Charging Station	15
4.1.2	Overhead Controller	15
4.1.3	Kilobot Calibration	15
4.1.4	KiloGUI	17
4.2	Arena	17
4.3	Camera	17
4.4	Projector	18
4.5	Other hardware	18
5	Implementation	19
5.1	Kilobot Position Detection	19
5.2	Creating Artificial Trail Pheromones	20
5.3	Parallel Processing and Threading	22
5.4	Kilobot Program	23
5.4.1	States	23
5.4.2	Filtering Sensor Input	26
6	Experimentation and Evaluation	27
6.1	Static Pheromone Tracing	27
6.2	Follow The Leader	28

7	Results and Discussion	30
7.1	Static Pheromone Tracing	30
7.1.1	Variant 1	30
7.1.2	Variant 2	31
7.2	Follow The Leader	32
8	Conclusions	34
8.1	Review of Project Goals	34
8.2	Challenges	34
8.2.1	Arena and Movement Blocking	34
8.2.2	Ambient Light Sensor	35
8.2.3	Trail Following	35
8.2.4	Color Detection	35
8.3	Context	36
8.3.1	Ethical Considerations	36
8.3.2	Environmental Aspects	36
8.4	Future Works	36
	Acronyms	38
	References	39

Chapter 1

Introduction

This is a thesis about virtual pheromones for swarm robots. Pheromones are a chemical substance that is spread from an agent to affect other agents' behavior [1]. This can be observed in groups of social insects or other swarms in nature, which communicate indirectly by spreading pheromones throughout the environment. Insects can then use pheromone-based communication to coordinate their actions. This brings fourth a so called *swarm intelligence*, allowing swarms to achieve complex tasks despite their individual limitations [2]. Venomous ants use their sting to lay down a product of the poison glands as a trail pheromone. This allows the ants to create paths between resources and nests, allowing other members in the colony to follow the pheromone trail and efficiently obtain and drop off resources [3]. The field of study concerned with achieving swarm intelligence within robotic multi-agent systems goes by the name of *swarm robotics* [4]. An evaluation of swarm robotics shows that swarm robotic applications are rare and that the majority of current communication architectures used within these systems are not equipped to establish the collective behaviors observed in nature [5]. Therefore, establishing indirect communication through the environment seen in natural swarms in robotic multi-agent systems is a challenge. Mimicking these collective behaviors within swarm robotic systems allows us to better understand the mechanisms that bring fourth swarm intelligence which allow for potential applications such as; disaster recovery, defense, inspection, mapping, farming, and space systems [6].

1.1 Problem Formulation

The objective of this project is to investigate the possibilities of creating a system that gives the swarm robots the ability to collectively achieve tasks via indirect communication through the environment using artificial trail pheromones. The objective can be divided into 4 sub-goals:

1. Establishing an environment that allows for pheromone-based communication between agents.
2. Develop a program that can create and integrate artificial pheromones into the environment.
3. Develop a program for the robots that allow them to act upon changes in the environment.
4. Evaluate the systems possibility of stigmergic-interaction with pheromone-based communication.

1.2 Division of Work

Almost all of the work done within this project is the result of both the authors cooperating. Thus, the development of software was primarily accomplished through pair programming. As for the setup and configuration of hardware, this was always done cooperatively at a workspace provided to the authors by the Centre for Applied Autonomous Sensor Systems (AASS) at Örebro University. The work divided below is more of an indicator of who took the initiative and responsibility of the given tasks.

Jonas Scharin

- Report: Wrote the abstract. Chapters 1, 2, and 7. Sections 4.1.3, 4.1.4, 5.4, 8.1, 8.2 and 8.3.2.
- Reading and Research: Researched into scientific papers within the field of swarm robotics. Found the papers which are presented in chapter 2 and inspiration for how to go about implementing pheromone-based communication.
- Coding: Developed the Kilobot program mentioned in section 5.4 and integrated the color detection into the system.

Eric Jansson

- Report: Wrote chapters 3 and 6. Sections 4.1, 4.1.1, 4.1.2, 4.2, 4.3, 4.4, 4.5, 5.1, 5.2, 5.3, 8.3.1 and 8.4. Created all the digital images.
- Reading and Research: Investigated swarm robot applications, robot-ethics and some various background topics for the project. Examined the specifications for the implemented hardware.
- Coding: Created the digital environment and integrated the projection processing with the color detection.

1.3 Outline

The structure and content for the remaining part of the thesis is as follows:

Chapter 2: Introduces the scientific field of swarm robotics and presents papers relevant to the mentioned sub-goals.

Chapter 3: Shows the environment and software architecture used in this paper.

Chapter 4: Demonstrates the necessary hardware used to establish the system

Chapter 5: Explains the different steps of implementing the system. Giving an overview of the different methods and tools used.

Chapter 6: Illustrates the different experiments and tests carried out to evaluate the system.

Chapter 7: Presents the results and explores the contributing factors.

Chapter 8: Concludes the paper by reviewing the project goals, revealing the challenges of the work, ethical considerations and how others can build upon this paper.

Chapter 2

Swarm Robotics

Swarm robotics is a field of study focusing on multi-agent systems consisting of homogeneous robots that can work together to solve complex tasks without the need for centralized control. The field of swarm robotics takes inspiration from the collective behavior of social insects and swarms in nature. The behaviors are achieved through local interaction in the form of either direct communication or indirect communication through the environment [4]. The properties of natural swarms are *robustness*, *scalability* and *flexibility* [5]. Therefore, some key features have been proposed for swarm robots:

- Robots are *autonomous*.
- Robots are *situated* in a specific environment.
- Robots have *sensing and communication abilities*.
- Robots operate in a *decentralized* manner.
- Robots *cooperate* to achieve tasks.

2.1 Swarming Behaviors

Swarming behaviors define the collective behaviors present in nature. A part of the research within swarm robotics consists of finding ways to achieve these collective behaviors within robotics. Taxonomies categorizing and defining the different behaviors observed in swarms have been proposed. One of the existing taxonomies divides the behaviors into four sub-categories as shown in fig 2.1; *spatial organization*, *navigation*, *decision making* and *miscellaneous* [5]. Behaviors that allow agents within swarms to spatially organize themselves or objects in the environment belong to spatial organization. Examples of these are *aggregation*; used to assemble the agents in a specific area of the environment and *pattern formation*; arranging the different agents to form a specific pattern. Navigation consists of behaviors that allow for coordinated movement. *Collective exploration* is a form of such behavior and allows the agents to collectively explore the environment and can be used to search for objects. *Decision making* are behaviours that allow the agents to collectively take an approach for a given issue. *Task allocation* belongs to this category and is based on dynamically assigning tasks to the agents to improve the performance of the swarm system. *Miscellaneous* includes behaviors that cannot not be assigned to either of the three categories mentioned, an example of such behavior is *self healing* that allows swarms to recover from defects in individual agents.

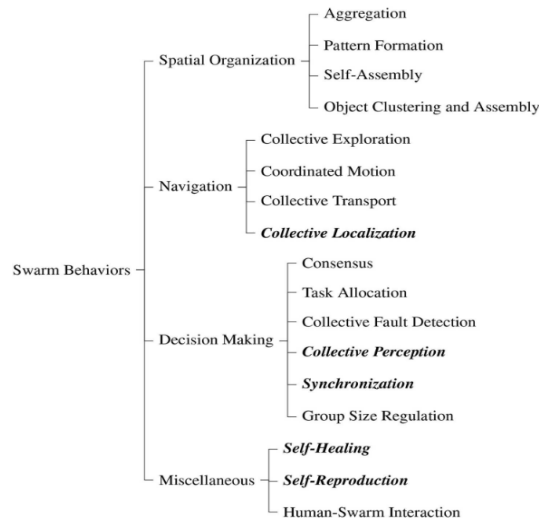


Figure 2.1: Taxonomy of Swarm Behaviors [5]

2.2 Stigmergy in Swarm Robotics

Swarm behaviors can be achieved by indirect communication through the environment. *Stigmergy* is a method based on such communication, it is defined by the principal that "*work performed by an agent leaves a trace in the environment that stimulates the performance of subsequent work by the same or other agents*" [7]. Stigmergy is a popular concept within the field of swarm robotics. It has been used to implement tasks and behaviors such as dynamic target searching and tracking, aggregation and foraging [8–10].

A way of establishing stigmergic-interaction is through *trail pheromones*. Trail pheromones, often found in social insects such as ants are a way to communicate through the environment allowing for stigmergic-interaction between agents [11]. Trail pheromones can be measured by features such as concentration for example amount of pheromones deposited on a trail and diffusion; the rate of which a pheromone's concentration decreases. The concentration of a pheromone can affect the way an agent responds to it. In the context of foraging within ant colonies a higher concentrated pheromone trail often implies that the trail is highly populated and therefore the reward (food source) must be of high quality and/or quantity. The distance of the trail pheromone can affect the concentration as well and therefore the efficiency of foraging [12].

There are studies within the field of swarm robotics that apply the principal of stigmergy by using trail pheromones as indirect communication. This is achieved in a variety of different ways, some existing approaches are; using extensive hardware that is made to interact with and be modified by the robots i.e using Random Frequency Identification (RFID) tags that can store data of *digital pheromones* which can be read and modified by agents [10, 13]. Displaying *virtual pheromones* onto the environment using projectors or Liquid Crystal Display (LCD) screens which then can be sensed using the robots existing sensors [14, 15]. Developing an augmented reality in which the robots can display and sense the pheromones and retrieve data gathered in the augmented reality threw wireless messaging protocols in real time [16].

This paper aims to establish pheromone-based communication that is done trough either digital or virtual pheromones and that can be developed without having to obtain highly expensive hardware. Therefore, in upcoming sections two studies consisting of the

approaches that are most inline with the goals of this paper are presented and explained in further detail.

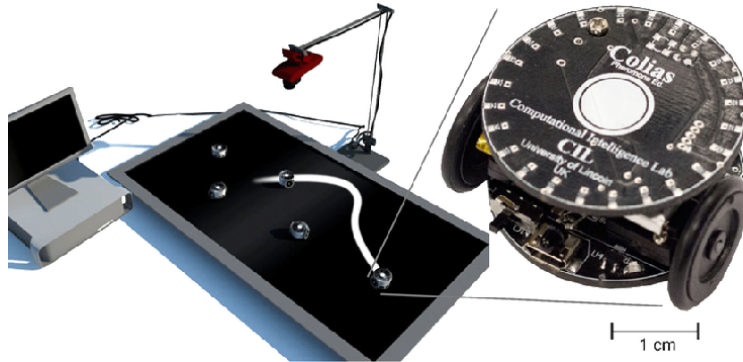


Figure 2.2: Virtual pheromones achieved with an LCD screen

2.2.1 Storing Digital Pheromones in RFID Tags

In this section a paper based on establishing pheromone-based communication with the use of digital pheromones stored in RFID tags is presented [10]. Each robot carries an RFID tag to read and change the data stored in the tags and a sensor to detect the signal intensity measured from the target which is calculated by a fitness function shown in equation 2.1. The stronger the signal intensity detected the closer the robot is to the target and therefore finding the target can be solved by maximizing the signal intensity function.

$$f(t, x_i, y_i) = S_{\max} - \alpha[x_i - \mathbf{tar}.x(t)]^2 - \beta[y_i - \mathbf{tar}.y(t)]^2 \quad (2.1)$$

Where $f(t, x_i, y_i)$ represents the signal intensity detected by robot i at position (x_i, y_i) at time t . $S_{\max}$ represents the signal strength at the target position, parameters α and β are authentication factors which reflect the reduction of the signal strength with respect to distance. Parameters (x_i, y_i) are the horizontal and vertical coordinates of the robot's position. The term $\mathbf{tar}$ is a function of time that describe the trajectory of the target.

The robots update their motion state depending on the pheromones it detects through the RFID tags. The pheromone model used is a vectorial model containing both size and direction. This allows the robot to determine the subsequent motion state through the vectorial pheromone given in a single RFID tag, using the concept of *intermediate vectors*. The role of the vectorial pheromone is to guide the robots towards areas with the strongest signal intensity.

To test the practical applicability of the proposed strategy self developed swarm robots with a diameter of approximately 24 centimeter (cm) and a unique marker on top to allow the tracking system to obtain the position of the robot were used. The tracking system consists of a computer running an open-source software called *ARToolKit 5* and a camera. The size of the experiment area is 9 meter (m)² and consists of 601 RFID tags. The experiments contain a target whose trajectory is either a triangle or a circle that is to be searched and tracked by the robots. The results from both experiments show that the stigmergic strategy proposed enables the robots to find and track the target and is feasible with a different number of robots, demonstrating the methods scalability.

This approach is robust and scalable and a solution to achieving stigmergic-interaction through pheromone-based communication. However as mentioned a total of 601 RFID tags is used to achieve an environment area of 9 m², making the setup process tedious compared to other approaches. Also implementing pheromones digitally removes the visual aspect and can make it harder to test and debug.

2.2.2 Virtual Pheromones Using Projector

An existing method to achieve foraging behavior within a robotic multi agent system is using a projector to display virtual pheromones onto the environment [14]. This method allows the robots to use their existing sensors to sense the pheromones in order to communicate through the environment. A camera was used to retrieve information about the environment and the data captured by the camera was then used to calculate where to display the pheromones.

The pheromones consists of two key features *evaporation*; allowing pheromones to disappear after time, avoiding useless information in the environment and *diffusion*; allowing the pheromone trails to obtain a gradient making the pheromones richer in information. These features can be tuned and changed to achieve optimal results within different experiments. A mathematical model is proposed to describe the dynamics of the pheromone:

$$\dot{\rho} = \delta + D\nabla^2\rho - \kappa\rho \quad (2.2)$$

The variable ρ represents the concentration of the pheromone, δ the injection concentration, D is a diffusion coefficient and κ is the rate of evaporation.

The robot used has a size of 75 millimeter (mm) in diameter and 140 mm in height. The sensors equipped on the robot consist of a touch sensor that can detect collisions, a pair of infrared sensors and three color sensors. Its movement is established with a wheel positioned on the bottom of the robot. The three color sensors are used to detect the color and brightness of the pheromones displayed through the projector.

The systems performance is tested through a foraging task performed by the robots using the established pheromone-based communication. The goal of the task is for the robots to search for a marked food source in the environment and when food sources are found return to a marked position acting as the nest. When transporting food from a source the robots display pheromones and robots who have not yet found food follow these pheromone trails if detected. The field used in the experiment was a black surface with the dimensions of 90 x 90 cm. A nest was represented using a yellow circle in the environment and an Infrared (IR) Light Emitting Diode (LED) is positioned at the nest to be detected by the robots. The food sources are either homogeneously or locally distributed and are displayed as points on the black surface. In this specific experiment a maximum of 3 robots where used.

By evaluating the results it is shown that less pheromone left in the field increases the amount of food collected in a homogeneous field, and that the overall performance improves when a pheromone trail is formed continuously in a field with localized distribution. The pheromone diffusion seems to have the strongest effect in cases with a single robot in a homogeneously distributed field.

This approach is the inspiration for the environment illustrated in 3.1 used to achieve pheromone-based communication. The idea of isolating the robots and their behavior from any type of second-part software, utilizing the hardware that is provided with the robots is inline with the motivations of this paper. However, the weaknesses of the projector-based system presented; robots detecting and getting stuck in their own pheromone trails is to be taken into account.

2.3 Extending the Capabilities of Swarm Robotic Systems

As seen in the papers presented in 2.2 achieving pheromone-based communication within these systems means having to develop ways of extending the systems functionality. In this section papers are focusing on improving the robots and the system as a whole.

2.3.1 Augmented Reality for Kilobots

A system made to combat the limited functionality of the *Kilobot* swarm robot is Augmented Reality for Kilobots (ARK) [16]. ARK works as a virtual environment that can receive and send information about the robots state in real time. This allows the user to create experiments based on virtual sensors, which are software based models of physical sensors [17]. The virtual sensors can sense changes in the augmented reality and send the data to the Kilobots.

The architecture of ARK is dependant on the modification and the incorporation of components. The Base Control Software (BCS) handles the processing of all data received from the environment and the messaging protocol used to send unique messages to the Kilobot in real time. The BCS requires a machine with a NVIDIA Graphical Processing Unit (GPU) because of the systems use of the Cuda toolkit; a NVIDIA toolkit for general computing on GPUs. The 4 cameras capture images simultaneously and send the images to the BCS which stitches them together using the GPU to create a image covering an area of 4.84 m² at 30 Frames Per Second (FPS). Communication from the BCS to the Kilobots is established through IR signals. The IR signals are sent from the Overhead Controller (OHC). The OHC is a hardware part belonging to the Kilobot system and is responsible for sending data to the Kilobots. ARK has decided to add extra IR light expanding the range of the OHC to allow for larger scale experiments. To achieve large scale real time messaging to Kilobots, ARK has developed a messaging protocol that constructs packets which are sent to the OHC from the ARK software.

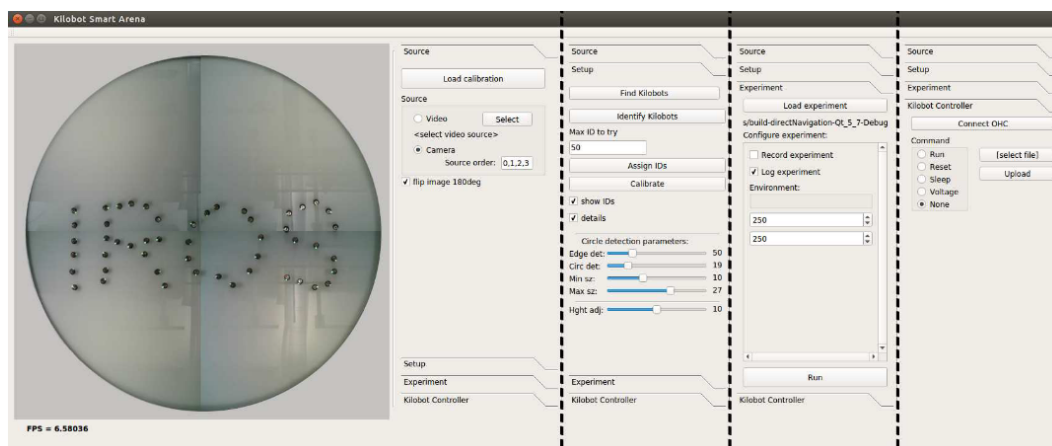


Figure 2.3: Graphical User Interface (GUI) of ARK.

In the paper, three demos (A, B and C) were conducted to evaluate the ARK systems capabilities. *Demo A* uses ARK to automate the process of assigning a unique Identification (ID) number to each Kilobot. The demo displays ARK's capability of retrieving data from the Kilobots. *Demo B* utilizes collective coordination of the Kilobots to form words. The word collectively displayed in the demo is *IROS* with a swarm size of 50 Kilobots, which is shown in figure 2.3. The demo displays how the ARK system with relatively high accuracy and short time can guide Kilobots to certain positions. *Demo C* consists of 50 Kilobots performing a *foraging* task. In this demo the source is a green circle which decreases in

size depending on the amount of resources available and the destination is a yellow circle which increases with the amount of resources returned to it. The goal is to move all resources from the source to the destination. This displays the robots capability of sensing and modifying the virtual environment depending on the state of the Kilobot.

ARK is an existing alternative to establishing pheromone-based communication through a virtual environment. However, the system has constraints being (1) The system only works with the Kilobot swarm robot, (2) requires a computer with a NVIDIA GPU, (3) modification of hardware and (4) uses a distinct messaging protocol for communication between Kilobots and the virtual environment. This paper aims to establish pheromone-based communication without the need of modifying existing hardware or establishing messaging protocols beyond those already available for the swarm robots.

2.3.2 Kilogrid

The Kilogrid is modular and scalable environment that aims to combat the limitations of the Kilobots and make it easier to conduct and evaluate experiments with large groups of robots [18]. Its built up of a grid of modules each covering an area of 100 cm^2 and a *dispatcher* that works as a interface between the modules and the computer. Each module contains 4 cells and each cell contains an IR transceiver and two RGB LEDs. The cell can both send and receive data via infrared light to or from the Kilobots. The user can program these modules via the dispatcher and therefore achieve functionalities not available with the Kilobots alone.

Kilogrid has been used to create *virtual sensors* extending the Kilobots sensing capabilities. In the paper an experiment is conducted using the Kilogrid to allow Kilobots to detect passive obstacles. Specific cells are programmed to act as a beacon alerting the Kilobots of passive obstacles. The experiment showcases how the sensory capabilities can be extended and that the Kilogrid improves the quality of experiments. Another application that has been tested is the implementation of *virtual actuators* for the Kilobots. This possibility allows the Kilobots to change and receive information about the environment, which in it self allows for the possibility of pheromone-based communication.

Kilogrid is a possible approach to establishing stigmergic-interaction through pheromone-based communication. However, similar to ARK it is only applicable for Kilobots and the Kilogrid modules are expensive, costing 75 euros each [19]. This paper is motivated by creating a low-cost approach to the challenge of establishing pheromone-based communication.

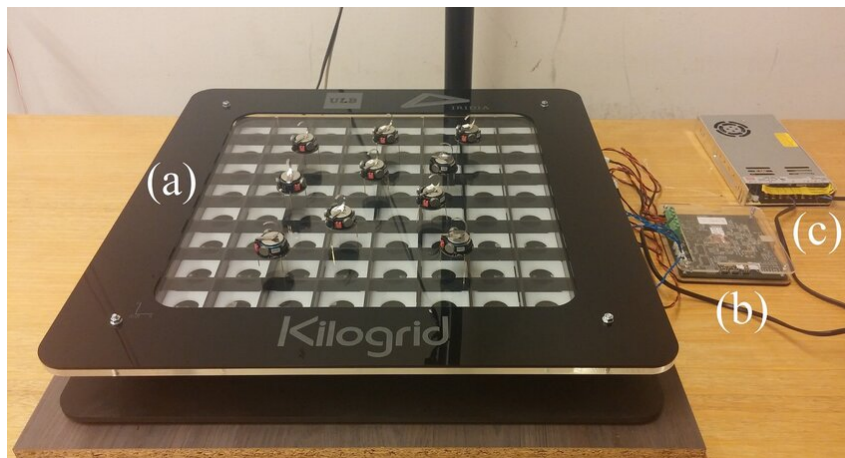


Figure 2.4: Kilogrid components: (a) Kilogrid (b) Dispatcher (c) Power Supply

Chapter 3

Architecture

3.1 Environment Architecture

The environment is depicted in figure 3.1 and consists of: (1) a wooden arena where the robots are situated, (2) an OHC used to communicate with the robots, (3) a webcam to capture the position of the robots and send it to the (4) virtual environment system, (5) a projector which displays the virtual pheromones onto the arena and (6) a three-legged steel frame to attach the camera, projector and overhead controller. A convenient aspect of the steel frame is that components are able to be attached at different heights. This enables easier calibration for synchronizing the sizes for projection and capturing between the camera and projector.

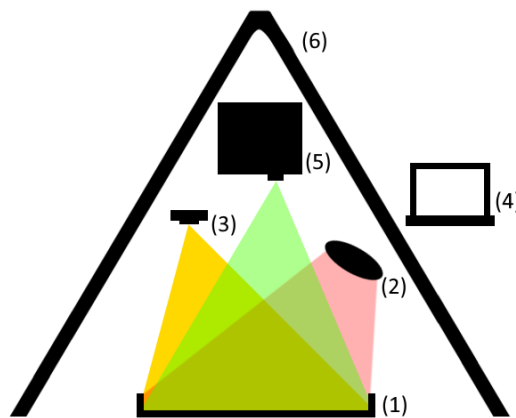


Figure 3.1: Hardware architecture

3.2 Software Architecture

The software architecture is based on the interplay between the arena's real-world state, the camera, the projector and the program. This architecture is represented in figure 3.2. It is utilizing the image capturing established with the camera, which then uses color detection to gather the Kilobots' positional data. This will add the position from the desired Kilobots that are displaying the specified color. This positional data is then stored in a shared resource within the program. From there, the data will be retrieved by the second thread and updated through a data processing layer. This process will find the pheromones that have expired and remove them from the shared resource. Then it will

use the processed shared resource to calculate pheromone data and generate a pheromone image for both the digital representation and the projector to display into the Kilobot arena. The Kilobots in the arena will react to these virtual pheromones and perform certain actions. This whole process is a closed system that will loop and make the program run indefinitely.

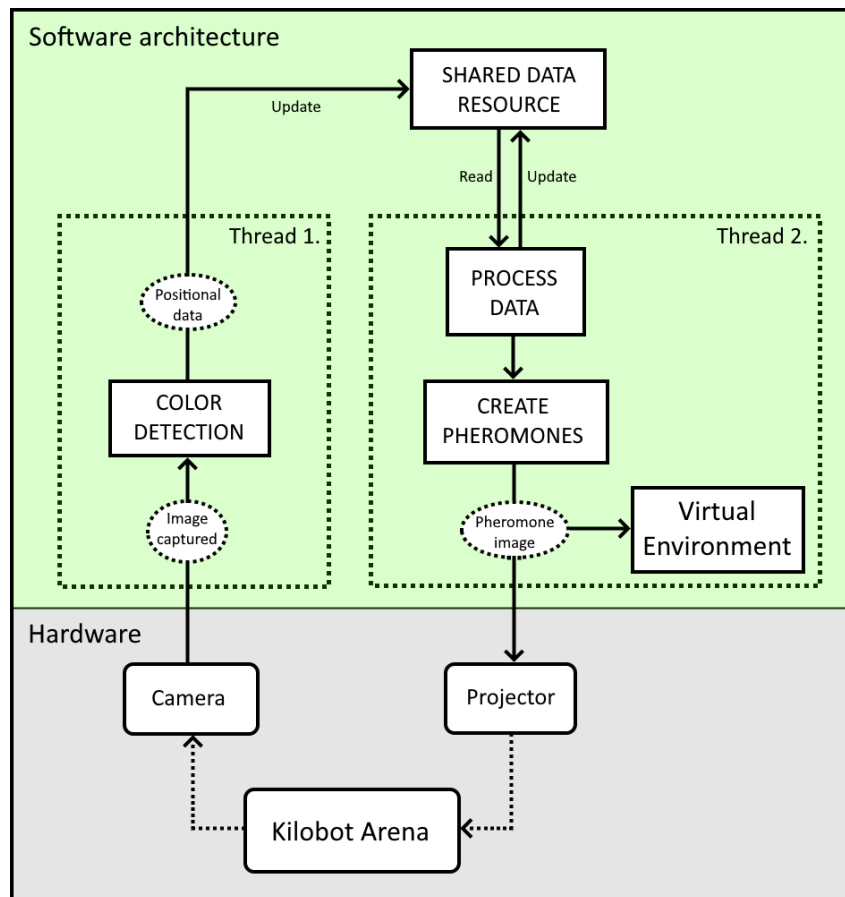


Figure 3.2: Software architecture and its interplay with the hardware.

Chapter 4

Hardware

4.1 Kilobot

The Kilobot is a robot designed with a focus on scalability, allowing for large scale experiments with possibly hundreds of robots. While being low cost, the Kilobot still includes some necessary abilities from other more expensive robots such as movement and inter-robot communication as seen in figure 4.1. The Kilobot's movement is based on *differential drive locomotion* achieved through vibration motors. When vibrating one of its motors, the Kilobot will turn in that direction, whereas running both motors simultaneously will cause it to move forward. This allows the movement to be either left, right or forward. It is possible to change the magnitude of the motors to achieve an as consistent movement as possible. This enables the robot to move approximately 1 cm per second and rotate approximately 40 degrees/second [20]. The slip-stick based locomotion that the robot uses makes it imprecise while traveling for longer distances but is an essential part of making the robot inexpensive. The robot's battery life is highly dependent on the robot's activity level, ranging from 3 to 24 hours based on usage. The battery can last up to 3 months when the robot is in sleep mode. [21]

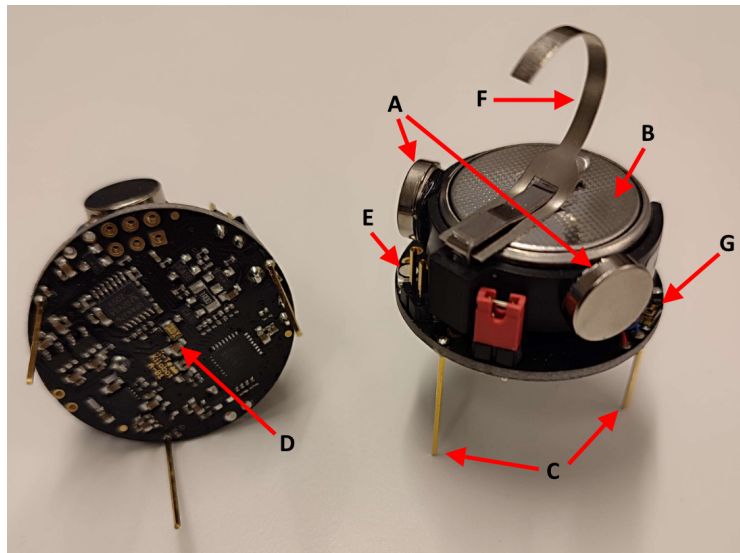


Figure 4.1: The architecture of the Kilobot is shown with its corresponding components marked from A-G. (A) Vibration motors, (B) Battery, (C) Supporting legs, (D) IR transmitter and receiver, (E) RGB LED, (F) Charging tab and (G) Ambient light sensor.

A common trait for swarm robots is to be able to communicate and sense each other which is why the Kilobot is equipped with an IR transmitter and receiver. This is located at the center of the Printed Circuit Board (PCB) on the bottom of the robot. This enables the robots to communicate with each other within a range of approximately 7 cm [20]. It only provides distance sensing and not bearing sensing. This means that the robot can sense how far away a neighbor is but not the direction. The functionality this enables provides sufficient capabilities for interesting collective behaviors. [21]

The LED on the robot's side can be used together with the robot's program. It can be programmed for specific functions and is also used for the pre-built battery check function. When debugging, the LED serves as a convenient tool due to the robot's limited feedback during runtime.

The ambient light sensor is positioned on the side of the Kilobot as seen in figure 4.1 and is used to detect the light levels in the current environment. There are alternative versions of the Kilobot that utilizes a LED-like bulb as its light sensor, but this is not the case for the robots in this project.

4.1.1 Charging Station

The charging station is a device that enables multiple Kilobots to charge at once. This is a very convenient tool since managing a larger number of robots can become time-consuming and therefore unscalable. The robots' batteries are rechargeable by attaching the charging tab and their back legs to the charging station. The charging can be seen in figure 4.2 and there the robots will fully charge after approximately 3 hours [20]. This charging station allows for a maximum of 10 robots to be charged at a time.



Figure 4.2: Kilobots on charger.

4.1.2 Overhead Controller

The communication to the Kilobots is made through an OHC, which is specifically designed to send messages or actions to the Kilobots wirelessly using IR signals [22]. This is done by connecting the OHC to a computer through a Universal Serial Bus (USB) cable, which then can send commands using the kiloGUI. The OHC has the capability of uploading code to multiple Kilobots at once and sending commands such as *run*, *pause*, *reset*, *sleep*, *battery*. It can also be connected with a wire to individual Kilobots for debugging or to upload firmware, which is required as an initial first step before using the robots. When the OHC is about 1 m away from the ground it has a reach of approximately 1 m in diameter. The OHC's infrared signals are easily disrupted by blocking the signal path physically, with for example a hand, which can disrupt uploading code or block messages to certain robots. This hardware is an essential part when working with the Kilobots.

4.1.3 Kilobot Calibration

After running some tests on the basic movement mechanics of the Kilobots, it was discovered that the robots were unpredictable in their movement which lead to difficulties controlling the robots behavior. This was combated by calibrating their vibration motors leading to a more predictable movement.

The calibration was done in two phases with the same criterion's which are as follows:

- Robots should achieve a turn of approximately 180 degrees in 5 seconds.

- Robots should achieve a forward speed of 1 cm per second.

The calibration was carried out on a total of 30 robots and those who could not achieve the mentioned criterion's were not to be used in the experiments to be conducted.

During the calibration in phase one all Kilobots were assigned a unique ID, this allowed the program uploaded to the Kilobots to send instruction based on specific IDs. This functionality proved to be useful when conducting experiments. To test the movement capabilities a test was run that programmed the Kilobots to go straight for 10 seconds starting from a marked point in the arena. After 10 seconds the Kilobots were instructed to stop and their change in horizontal direction (x) and vertical direction (y) was measured in centimeters and their rotation in degrees. Every Kilobot with a max offset of 45 degrees and that achieves an ending position of $5 \leq x \leq 15$ and $-5 \leq y \leq 5$ cm passes the test. Figure 4.3a shows the ending position of all Kilobots that were tested starting from the point ($x = 0, y = 0$), the area bounded by green and blue represents the area of ending positions that pass the test. Fig 4.3b shows the rotation offset measured at the end of the test, all Kilobots started with a rotation of $y = 0$ and the x axis represents the different robots tested. For this test all measurements within the blue area are approved. Note that there are less than 30 points in figure 4.3a, this is because of overlapping results. Figure 4.3b shows a total of 29 points, the reason being that a Kilobot was discarded before the tests due to calibration issues.

After the calibration and testing in phase one a total of fourteen robots were discarded that either couldn't be calibrated to achieve the given criteria of speed and rotation, or didn't pass the test conducted. During the calibration several tests were run and showed that behavior could vary a lot between different tests. The awareness of this is used to interpret the results with caution, knowing that the robots discarded could possibly be of use if the need for more than those approved arises. However, this proved to be a good starting point, allowing for a greater knowledge of each Kilobot and their movement. Phase two consisted of handpicking the amount of robots required to run the experiments from the subset of robots that passed phase one. Thereafter, the handpicked robots were tested again to ensure the predictability of their movement. This allowed for efficient development of the program controlling their behavior, which will be presented in section 5.4.

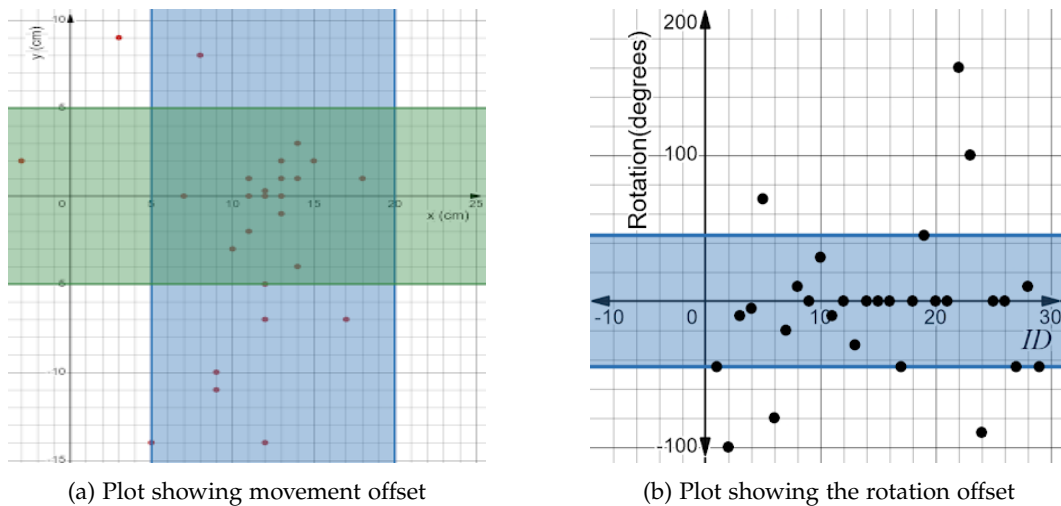
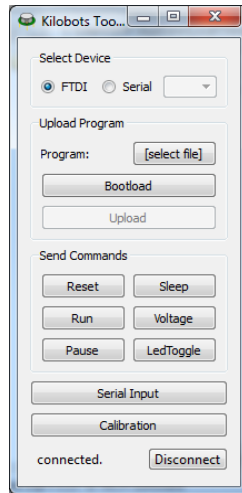


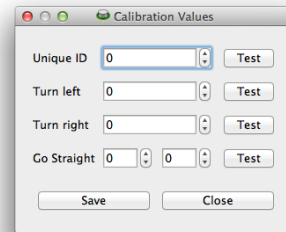
Figure 4.3: Plots showing the results of phase 1

4.1.4 KiloGUI

The calibration was done through *KiloGUI*; a software that is used to connect to the OHC and interact with the Kilobots through commands [22]. As seen in figure 4.4a there is a button titled *Calibration*, pressing this opens the window shown in 4.4b. In this window the respective movements (left, right and straight) can be tuned to establish a desired movement for the Kilobots. The calibration values are saved to variables stored in the Kilobot code library and are named; *kilo_turn_left*, *kilo_turn_right*, *kilo_straight_left* and *kilo_straight_right*. These variables can be sent as parameters to the *set_motors* function that assigns the desired power to the respective motors [23].



(a) KiloGUI start window



(b) KiloGUI calibration window

Figure 4.4: Kilo GUI start screen and calibration

4.2 Arena

The arena used for the project was a pre-built, homemade structure, built without a larger understanding of the robots' capabilities. It is made up of a white, flat melamine board which has wooden walls nailed to it to keep the robots in an enclosed area. The arena's smooth surface is important for the robots' IR messaging to work properly, as well as their fine movement which is very sensitive to minor irregularities in the floor. The dimensions of the arena are 65 x 83 cm and the walls are 8 cm above the floor surface. It is also important to keep the arena itself in a flat, stable environment to prevent slippage of the robots when they're moving.

4.3 Camera

The camera functionality does not need to meet high standards. There were more important factors like weight and size that was beneficial for this project. Since the camera will hang in the steel frame it is favorable to have a lightweight and small camera to attach. This project uses a *Plexgear WC-800 camera*, selected for its cost-effectiveness as one of the more affordable 1080p camera available. It is adjustable in an up- and downward fashion which is beneficial when trying to synchronize the camera with the projector. These attributes fulfill the limited requirements necessary for this project.

4.4 Projector

It is important for the projector for this project to have attributes like; appropriate weight, illumination strength and screen size. The architecture decided upon requires the projector to display a screen from above the arena and therefore needs to be mounted in a specific way. It must be high enough to display the correct projection image size.

The screen from the projector must be displayed from a straight top-down angle because the Kilobot's light sensor is placed upwards on its PCB. If the angle of the projector would be tilted it could give the Kilobot unpredictable light input, which would make the robot behave differently. This further proves the importance that the projector should be small and lightweight since it requires to be mounted in this specific fashion.

The screen size of the projector is what decides how it will be displayed onto the arena and is an important factor. It is required for the projector screen to cover the arena to communicate with the robots properly. The projector used for the project was an *Optoma ML750e WXGA LED* which has beneficial specifications that was fitting for the current architecture. The screen size for this product will be approximately 40 inches (102 cm) when it is attached at a distance of 1.3 m above the arena [24].

4.5 Other hardware

To ensure that the hardware components work well together, an easily modifiable setup is beneficial. This was achieved by using a three-legged steel frame, which functions similarly to a larger tripod. The steel frame enabled precise adjustments to the height and position of the entire contraption. It also simplifies the height optimization for the components that were attached onto the legs of the steel frame.

Lastly a computer is necessary to run the software and connect everything together. The computer must be capable of connecting to the camera and the OHC via two USB inputs, and to the projector via a High-Definition Multimedia Interface (HDMI) input. It is required to have the camera and projector connected to the same computer during runtime. This is not true for the OHC, which can be connected to another computer to make commands and upload code.

Chapter 5

Implementation

5.1 Kilobot Position Detection

For the program to create virtual pheromones, it first requires to know which Kilobot should release the pheromones and where the robot is located. This is done by using color detection and the camera. The robots that are programmed to leave pheromones will emit a red light with their LED and is therefore distinguishable between all other robots in the environment. The color red was chosen because it was especially easy for the camera to pickup and wasn't affected as much by light interference in the environment. An example of this can be seen in 5.1. Displaying pheromones in the complementary color green makes the color delineation between the pheromones and the Kilobot's LED increase, which also improves performance [25].



Figure 5.1: Camera view

The functionality of the color detection is implemented by using the python library Open Source Computer Vision (OpenCV) and can be seen in algorithm 1. The algorithm starts by taking an image frame from the camera, resizing that image and converting it into the Hue Saturation Value (HSV) color system. Then it creates a mask based on a lower and upper bound red color. Masking in this context can be explained by figure 5.2, which is masking the orange color and sets the mask to only black and white depending on the given lower and upper bound of the HSV values. With that mask it is then possible for the program to find where the pixels of interest are residing. The only source of red color in the present environment should be from the red LED emitted by the Kilobot and therefore the HSV boundaries for the mask are tuned to only recognize that robot. Then contours are created based on the red mask and if a contour is big enough, it is defined as a recognized Kilobot and the positional data will be saved. The data is then safely stored

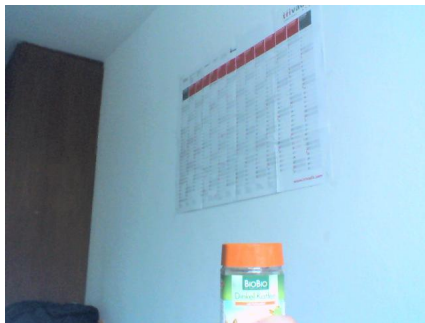
with a locked mutex in a shared resource for the digital environment to use later on. The program uses two threads and the color detection is running on one of them. The shared resource is used only by the second thread but updated by both of them. Updating and reading the shared resource will be safely managed by the locked mutex and will ensure that there will be no data conflicts within the program.

Algorithm 1 Color detection program using the python library; OpenCV.

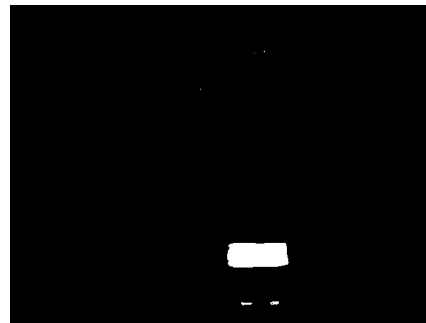
```

1: Initialize video capture
2: while True do
3:   Get new image frame from video capture
4:   Resize image frame
5:   Convert image into the HSV color system
6:   Create a mask of the red colors
7:   Dilate mask
8:   Create contours based on the mask to track red color
9:   for all contours do
10:    if countourArea != small then
11:      Add coordinates and contour size to Kilobot data
12:      Add contour to image frame # Show where the Kilobot is found
13:    end if
14:  end for
15:  Lock mutex and add Kilobot data to the shared resource
16:  Display video image frame in digital environment
17: end while

```



(a) Original image



(b) Masked image

Figure 5.2: Before and after masking of orange color [26].

5.2 Creating Artificial Trail Pheromones

The artificial pheromones are virtually created by the Kilobots, using the digital environment. This is done through the use of the red lit Kilobot's positional data, gathered by the color detection mentioned in section 5.1. It is then possible to use that positional data to display the pheromones in the real world environment and the digital environment. This part of the program is executed through a second thread, which controls the digital environment as well as the pheromone tracks. The thread initializes the digital environment by creating the window which can be explained by algorithm 3. Here the program starts by assigning the bindings for the key presses and initializes the grid size and attributes of the GUI. This process is required only once, which is why it is excluded from the loop in algorithm 2.

Algorithm 2 Creating and displaying pheromones

```

1: Create the digital environment window
2: while True do
3:   Create new black background and get the current time
4:   Lock mutex to handle shared data resource
5:   for all Data frames do
6:     if Time since data frame was created > Pheromone lifespan then
7:       Remove data frame
8:     end if
9:   end for
10:  Make a copy of shared data resource and unlock mutex
11:  for all Copied data frames do
12:    Set pheromone opacity based on time since creating that data frame
13:    Draw pheromone circles onto black background
14:  end for
15:  Resize image to fit projector screen
16:  Display full pheromone image on the screen for projector to show
17:  Display full pheromone image in digital environment
18: end while

```

Algorithm 3 Creating the Digital Environment Window

```

1: Set bindings for key presses
2: Initialize grid size and attributes
3: Create label and entry for Pheromone size
4: Create label and entry for Pheromone life span
5: Create label and entry for Kilobot size
6: Create labels and entries for boundaries for the colors red, green and blue
7: Create a submit and a clear button

```

Before creating the pheromones, this part of the program starts with updating the shared resource. Accessing and updating the shared resource requires a lock with a mutex. When the shared resource is updated it is then copied to avoid blocking the program for too long. The mutex will be unlocked and the rest of the process will use this copied version of the shared resource. To remove the expired pheromones, the program will check all the data points time of creation in the shared resource. A pheromone counts as expired if $\text{Pheromone_life_span} < \text{current_time} - \text{frame_data}[i].\text{time}$. The $\text{frame_data}[i].\text{time}$ is the time when the data frame was created and the $\text{Pheromone_life_span}$ is the time a pheromone is active for and is a variable that can be set dynamically in the program's GUI. The GUI is responsible for dynamically handling the changes of the values Pheromone_size , $\text{Pheromone_life_span}$ and Kilobot_size . This can be seen in figure 5.3 together with the color detection's HSV boundaries.

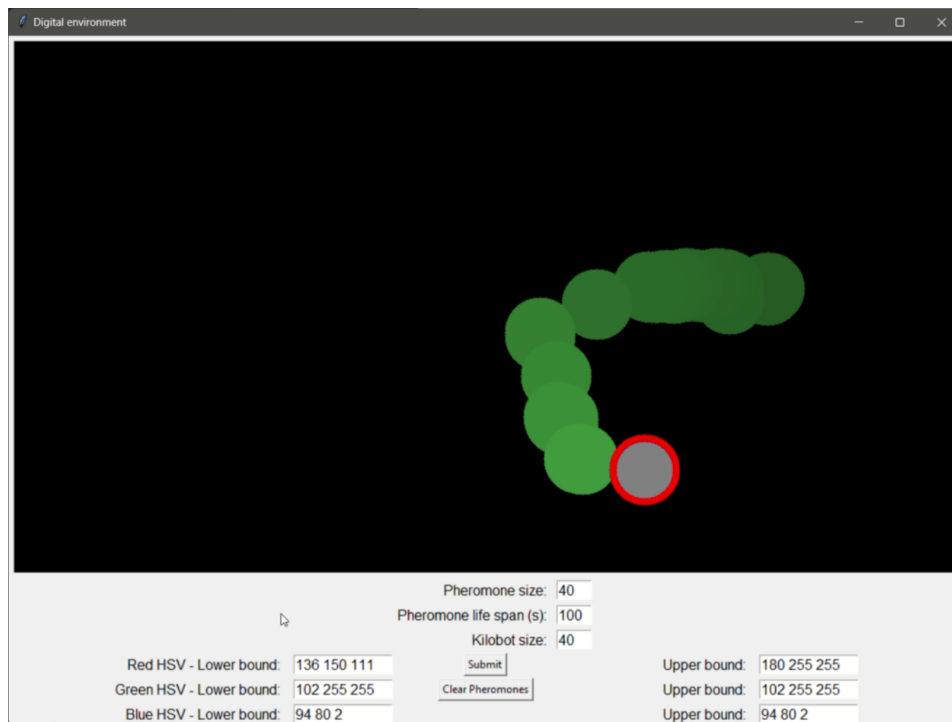


Figure 5.3: Digital environment

The program will create a new black background to draw the new pheromones onto. For each data frame, it will draw a pheromone based on the remaining lifespan of that pheromone. When all the pheromone trails have been drawn, the image will be resized to an appropriate size. This will make it fit properly within the digital environment as well as the physical arena. An example of how this final image would look like can be seen in the GUI in figure 5.3.

5.3 Parallel Processing and Threading

It can be seen that both algorithm 1 and algorithm 2 are running on infinite loops because of their "while True" statement. This requires the color detection program and the trail pheromone program to be run in parallel. This is achieved via the use of different threads. By being on separate threads, it enables more fluid video capturing while also avoiding a bottleneck that would slow down or block the rest of the digital environment's functionality.

The software depends on the camera image, and once the data is collected, the program can use it. Therefore threading was implemented as seen in figure 3.2. This is to avoid the camera from being locked by the data processing and instead work in parallel with each other. Since the second thread calculates all the pheromone data each iteration, it doesn't matter if it is computing faster or slower than the first thread. However, it is crucial while using threads, to have proper synchronization between the processes with locks, flags or barriers [27]. Because of that, a lock is used when the program wants to write, read or edit the shared resource. This all ensures the program not to interfere between the parallel processes and thus avoids synchronization issues.

5.4 Kilobot Program

To achieve pheromone-based communication a program that allowed the Kilobots to follow and sense the pheromone trails was developed. The following program is based on establishing a swarm behavior called *follow the leader*, in which one leader robot is followed by a number of other robots. In this case the leader releases a pheromone trail and the remaining robots goal is to find this trail and follow it, establishing a coordinated movement.

The behavior and state control of the Kilobots was programmed using the Kilobot code library named *Kilolib* consisting of methods to establish movement, message protocols through IR signals and sensor readings [23]. The Kilobots are built on AVR micro controllers and are therefore programmed using AVR C which is a language similar to C [22]. AVR is a suite of open software development tools used for debugging and a compiling within AVR systems [28]. To avoid installing and setting up AVR the Kilobotics website provides a web-based-editor where programs for the Kilobots can be written and compiled through amazon servers. However, this feature is no longer being developed and is unavailable. Therefore, the AVR tool for windows was installed and setup together with the Integrated Development Environment (IDE) Eclipse. The code was written and compiled in Eclipse and thereafter uploaded to the Kilobots through the kiloGUI program and the OHC.

In order to achieve stigmergy through pheromone-based communication within our system, two criteria had to be achieved by the Kilobots, which are:

1. The Kilobots have the ability to sense pheromones.
2. The Kilobots are capable of following a pheromone trail with a certain accuracy.

To make sure criteria 1 was met the ambient-light sensor mentioned in 4.1 was tested with different levels of brightness's making sure that it is sensitive enough to distinguish between the differences in brightness. The readings obtained from the ambient light sensor could range from 0-1023 (maximum brightness reading of 1023) [23]. When running tests within an environment affected by natural daylight and lamps, the sensors values are hovering around the max value of 1023. This meant that increases in brightness through external sources such as the projector had no effect on the sensor reading. Therefore, implementing experiments based on detecting changes in brightness had to be done in a dark environment. Meaning that the level of brightness sensed was low enough to allow the possibility of sensing an increase in brightness from the projector. Criteria 2 was achieved by developing an algorithm based on different behavioural states consisting of *following* and *return back*. The *following* state is assigned when a pheromone is detected and instructs the Kilobots to move forward until it has left the pheromone trail. The *return back* state is enabled when the Kilobot loses a pheromone trail and implements an algorithm of its own, attempting to return back to the trail again. In order to explain the algorithm more in depth the different states and their respective movements need to be explained, which is done in the upcoming section.

5.4.1 States

The program uploaded to each Kilobot is built on different states that control their behavior depending on environmental changes as seen in figure 5.4. There are a total of five states; *Leader*, *Following*, *Return Back*, *Searching* and *Turn 180*. Each state is represented by a unique color which was displayed through the LED of the Kilobot. Red being *Leader*, blue: *following*, green: *return back*, no color (LED off): *searching* and purple: *turn 180*.

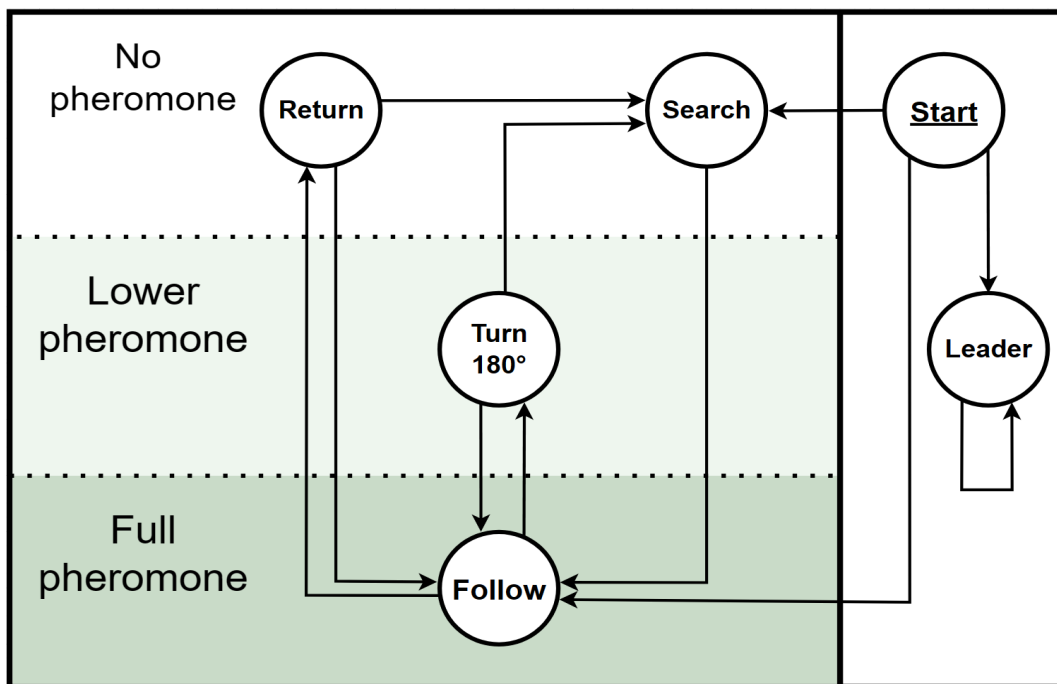


Figure 5.4: State machine

The *Leader* is always a leader meaning that if the Kilobot is in the leader state it can not be assigned another state. This is because the program built is based on one static leader which is decided before runtime. The leader is chosen and identified through their unique ID which is mentioned in section 4.1.3. In order for the digital environment software to detect and distinguish the leader from non-leader agents, the leader's LED is turned red. This allows the software to accurately display pheromone trails for Kilobots in the leader state. The movement of the leader is built on differently distributed probabilities of movements which are *turn left*, *turn right* and *forward*. The method for the leader movement takes a random value and assigns the movement of the Kilobot depending on the value. The random value generated is within the range 1 to 15 where 1 is defined as forward, 2 left and 3 right, if the value is between 4 and 15 it is automatically set to 1 giving the forward movement a probability of approximately 87% (13/15). The leader movement method is called every 2 seconds repeating this procedure. The probability of going forward was set to a high percentage to ensure the presence of pheromone trails. The movements of left and right only rotate the robot in its current positions. Therefore, favoring the forward movement would lead to a higher chance of pheromone trails being created.

The *following state* handles the behaviour of the Kilobots that are on a pheromone trail. The state calls a method which instructs the Kilobot to move forward until the pheromone trail is lost or its intensity (brightness) has decreased. In the case where the Kilobot is in a following state and has lost the pheromone trail, it will be assigned to a state named *return back*.

The *return back state* incorporates an algorithm that is made to return the Kilobot to the pheromone trail that it has lost. The goal is to establish a movement in and out of the pheromone trail. To achieve said movement, algorithm 4 is used. A movement timer containing the time passed in seconds since the last change in movement for the Kilobot is used. This allows the movement of the Kilobots to be sustained for a defined period of time without constantly being overwritten. When entering the return back state and running algorithm 4 for the first time, this timer will be zero and in that case the Kilobot

will swap its direction based on the last direction recorded. The rotation counter keeps track of the amount of rotations the Kilobot has executed. After the Kilobot has moved forward for a certain amount of time, it is instructed to rotate in the same direction as before. The time forward timer is increased after every rotation to increase the time the Kilobot moves forward, avoiding a circular motion. The rotation time defines how long a rotation is carried out for in seconds.

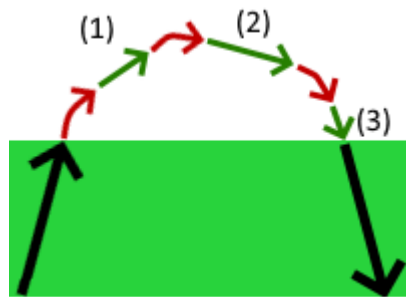
Algorithm 4 Algorithm to return back to pheromone trail

```

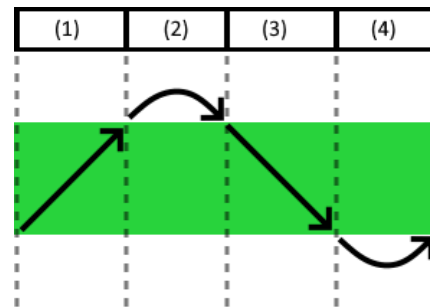
1: if Movement timer has not started then
2:   Swap direction
3:   Increment rotation counter
4:   Reset time moving forward
5:   Set movement timer to current time
6: else if Time passed > Forward Time then
7:   Set motion to previous direction
8:   Increment rotation counter
9:   Set movement timer to current time
10:  Add 1/2 second to time forward
11: else if Time passed > Rotation time then
12:   Set motion to forward
13: end if
  
```

An optimal scenario of the return back algorithm is shown in figure 5.5a. The basic idea is that when leaving the trail depicted by the green area the robot will rotate in a certain direction shown by the red arrows and in between the rotations turn forward which is illustrated by the green arrows (1), (2) and (3). The forward movement is increased after every turn which is represented between the arrows (1) and (2). This whole process allows the robot to not only rotate in place but also move in that direction.

In figure 5.5b the idea of the desired "zig-zag" movement is shown where the Kilobot in (1) is in a following state heading out of the pheromone trail depicted by the green line. When leaving the pheromone trail as shown in (2) the Kilobot is instructed to turn towards the lost trail and remember its previous direction so that the Kilobot knows where to turn in stage (4).



(a) Example of the return back algorithm



(b) Sketch of "zig-zag" movement

Figure 5.5: Return back algorithm.

The *turn 180* state is assigned when the kilbot detects an ambient light value which exceeds the threshold value of it being in a pheromone, but has decreased compared to earlier light readings. This essentially means that a decrease in pheromone strength has been detected, meaning it is heading away from the source which is dropping the pheromone. Therefore, this state calls a method that instructs the Kilobot to turn for five seconds which according

to the calibration criteria in section 4.1.3 is equivalent to a turn of 180 degrees.

In order to establish a movement unique for those searching for a pheromone trail the *searching* state was implemented. The state is assigned to the Kilobots who have yet to detect a pheromone. The movement is similar to the leader movement with a slight adjustment in probability. The probability for the searching movement is distributed as follows: forward 80%, left 10% and right 10%. The searching state is also enabled after a certain amount of time has been spent in the return back state without making it back to the trail.

5.4.2 Filtering Sensor Input

The ambient light sensor is very sensitive and therefore the sensor readings needed to be filtered to avoid sudden anomaly values and inconsistent readings. This was done by storing a sample of sensor readings in an array of ascending order and then taking the median value. The size of the array (or amount of readings in a sample) varied and was fine tuned to find a balance between getting a well filtered value and not delaying the program. The sample size used for the experiments was 50 readings. Also paper wrapped in aluminium was used to extend the reach of the sensor as seen in figure 5.6. Allowing light reflected from the aluminum to be detected by the sensor.



Figure 5.6: Extension of ambient light sensor built with paper wrapped in aluminum foil

Chapter 6

Experimentation and Evaluation

6.1 Static Pheromone Tracing

The first experiment carried out was used to measure the Kilobot's ability of following a pheromone trail. The Kilobot would start at one end of the arena and the goal is to reach the other end by following the pheromone trail. A static image is displayed by the projector onto the arena to simulate the trail. This means that both the color detection and pheromone projection wasn't active and the results would only be a product of the Kilobot's capabilities. The trail was created to simulate a possible, real world scenario where the track could've been made by a Kilobot producing pheromones using the developed system. The only difference was that the trail used for this test didn't have a fading pheromone concentration and instead was set at a static value, from start to finish. Meaning that the pheromone has no gradient and therefore information such as direction of the trail is absent. One of the trails used is seen in figure 6.1a, where the path was drawn using green circles, almost identical to the pheromones that are dynamically displayed through the projector program.

The parameters used in the experiment was a variation of angles from where the Kilobot would get its starting position, as well as the size of the pheromones. In figure 6.1 it is apparent that the Kilobot is placed outside the pheromone before the test starts. This is the case for all tests except 180 degrees, where the Kilobot starts inside the pheromone but in the wrong direction. The measurements for the pheromone sizes are based on the scale of the Kilobot's own size, which is approximately 3.5 cm in diameter. The pheromone size parameters for this test procedure is set to a diameter of 7 cm and 14 cm, which is 200% and 400% relative to the size of the Kilobot.

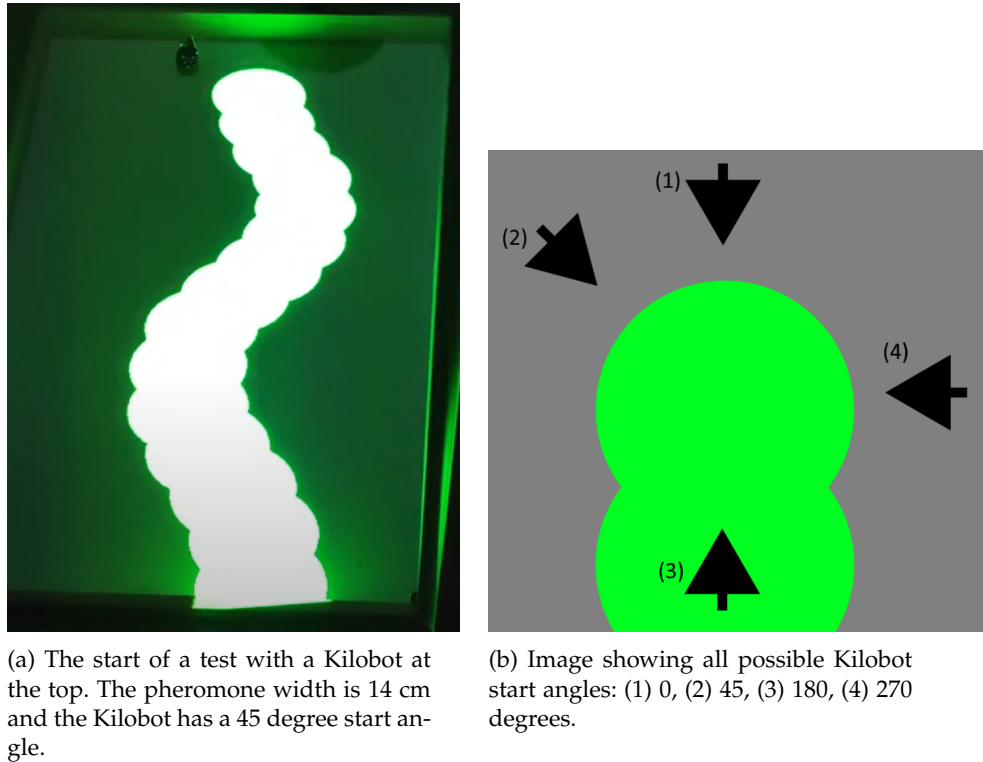


Figure 6.1: Images showing cases from the static pheromone image tests.

The metric measured was the time it took for the Kilobot to reach the end of the trail. To avoid pointless results, a time constraint of 5 minutes was included so the Kilobots wouldn't move around aimlessly until randomly reaching the goal. In the case that the time limit was reached or the Kilobot lost the pheromone completely, the test would get the results: Did Not Finish (DNF). A pheromone counts as lost when the Kilobot enters the *searching* state and therefore also turns off its LED.

After the initial testing of static pheromone image test there was a second variant of the experiment conducted. The tests in the two variants were identical in the sense of environment, parameters and metrics. The only difference were changes made in the Kilobot program.

6.2 Follow The Leader

The second experiment was conducted with a *leader* robot that was instructed to leave pheromone trails instead of displaying a static pheromone image onto the arena. That means that the pheromones are dynamically created by the leader. By selecting a individual Kilobot as the leader before the test, it is possible for that Kilobot to display pheromones in the arena with the help of the color detection and projection program. Using the selected Kilobot's unique ID set during calibration, it is possible to give that specific Kilobot special commands. During this test it is used to instruct the Kilobot to enter the leader state and stay in that state, which means that it will never react to pheromones itself or care about light changes in the environment. This experiment explores the capabilities of the robots ability to achieve stigmergic-interaction and uses every component of the system. All tests consisted of 7 followers and 1 leader. The leader was placed in the middle and all the Kilobot followers were placed on each side and corner of the arena, all pointing inwards. This configuration is shown in figure 6.2.

The parameter for this experiment was pheromone lifespan which indicates how long a pheromone is detectable for. This feature was predicted to have the strongest affect on performance and therefore two different lifespans were tested separately. This allowed for evaluation of any potential correlation between pheromone lifespan and performance.

The goal of the experiment and the definition of a good performance is a coordination of movement where all followers are following the leader. Therefore, metrics measured during the two tests were divided into the number of followers, meaning that the time in seconds and percentage of the program where followers were following the leader was measured for 0-7 followers. Allowing for evaluation of follower stability overtime and any correlation between the amount of followers and performance.

Since the walls of the arena have been a problem from start, a drastic environment change was made to allow tests to run without any interference. When a Kilobot reached the wall, it would struggle to free itself and repeatedly collide with the wall. This was partly due to the Kilobot's round body, which caused it to continue facing the wall even while attempting to turn. Therefore an additional arena wall layer was created. This layer was low enough to avoid the Kilobot's body and instead collide with its legs, which made it easier for the robots to break free. The layer itself was created with basic rulers taped together. To prevent Kilobots from getting stuck in the corners of the arena, the corners were also rounded off, as seen in figure 6.2. The corners were made of cut out paper which was taped together with the rulers to keep it steady when robots would come in contact with it.

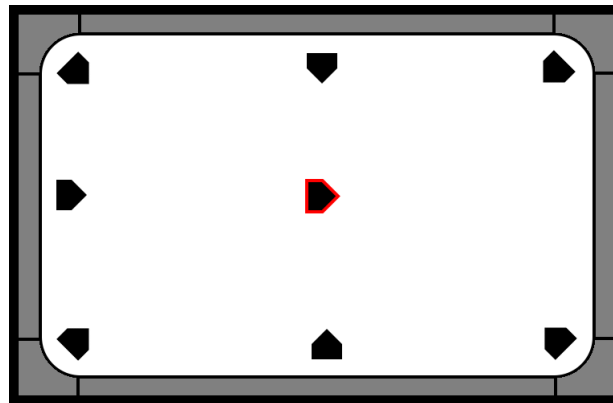


Figure 6.2: Test configuration for one leader and multiple followers. The leader (depicted in red) is placed in the middle and the 7 followers on the sides.

Chapter 7

Results and Discussion

7.1 Static Pheromone Tracing

The results from the static pheromone tracing test were divided into two variants (1 and 2), allowing for any needed improvement based on results from variant 1.

7.1.1 Variant 1

The parameters for variant 1 were four starting angles and two pheromone sizes as mentioned in section 6.1. The combination of all parameters lead to a total of eight tests. The results for the tests conducted with a pheromone diameter of 7 cm can be seen in table 7.1. If a Kilobot did not make it to the end of the pheromone trail the finishing time is marked as DNF and the reason could either be that it lost the trail or the cut off time of 5 minutes was reached.

Table 7.2 shows the results for all tests conducted in variant 1 with the pheromone diameter of 14 cm. The results are similar to the ones shown in table 7.1 with a success rate of 25% and a finishing time difference of 13 seconds. However, the test in which the Kilobot successfully achieved its task of following the trail to the target point was done with the starting angle of 270 degrees. This is most likely not an indication of certain advantages in staring points depending on pheromone diameter. It is most likely correlated to the Kilobots chances of choosing an effective rotation in the return back state, which is essentially a random guess in variant 1. In order to investigate more into the correlation between staring angle, pheromone size and reaching the end of the trail, more tests need to be conducted.

Start direction (degrees)	Finishing time (minutes)
0	2:38
45	DNF (lost pheromone)
180	DNF (cut off time reached)
270	DNF (lost pheromone)

Table 7.1: Results with pheromone diameter of 7 cm

Start direction (degrees)	Finishing time (minutes)
0	DNF (lost pheromone)
45	DNF (cut off time reached)
180	DNF (lost pheromone)
270	2:25

Table 7.2: Results with pheromone diameter of 14 cm

7.1.2 Variant 2

In variant 1 the unreliability of the Kilobots trail following algorithm were made clear and therefore improved before conducting variant 2. Meaning that the only difference between variant 1 and 2 was the changes made in the Kilobot program. To improve the trail following ability changes in the return back algorithm (4) were made. The algorithm is based on swapping its direction from the last direction recorded in a return back state and if such direction does not exist it will automatically choose to turn left. This meant that the ability to establish the "zig-zag" movement mentioned in 5.4.1 to accurately follow the trail was dependant on it randomly choosing the most effective direction. The Kilobot had a 50% chance of choosing the correct direction to turn when losing the trail for the first time. If the wrong direction was chosen at start it would cause the whole algorithm to be out of sync, which was the catalyst for the results in variant 1. To combat the problem of wrong turns a rotation counter was introduced, counting the amount of turns the Kilobot had taken to make it back to the trail. This allowed the program to detect and adjust the movement when a Kilobot had taken an abnormal amount of time to return back to the trail, which usually implied that it had been turning the wrong direction.

The results in table 7.3 and 7.4 show a 75% success rate for both pheromone sizes which is an improvement from variant 1. The average finish time for the tests conducted with a pheromone diameter of 7 cm was 3 minutes. A pheromone diameter of 14 cm gave an average time of 3 minutes and 4 seconds. Note that the average finish time does not take into account DNF measurements, meaning that failed tests will not affect the average finish time.

Start direction (degrees)	Finishing time (minutes)
0	1:29
45	4:42
180	DNF (cut off time reached)
270	2:49

Table 7.3: Results with pheromone diameter of 7 cm

Start direction (degrees)	Finishing time (min)
0	DNF (lost pheromone)
45	3:33
180	2:10
270	3:29

Table 7.4: Results with pheromone diameter of 14 cm

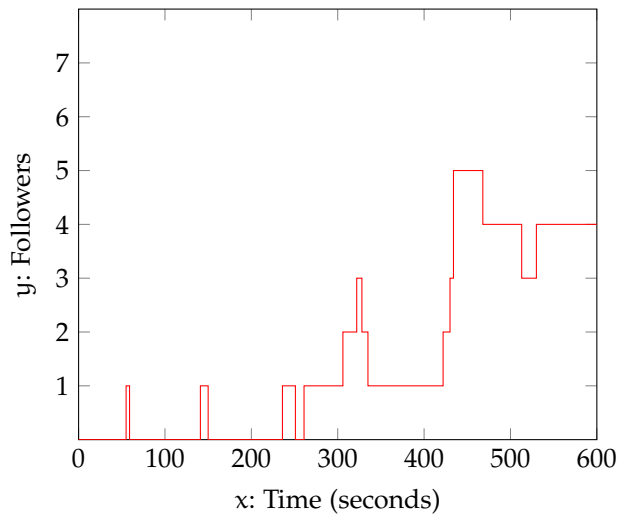
7.2 Follow The Leader

The static pheromone image tests made it clear what needed to be improved and was used to prepare the Kilobots to conduct a follow the leader experiment. The follow the leader experiment is based on dynamic trail pheromones placed in the environment by a leader robot, meaning that trails have a certain lifespan. As mentioned in section 6 the static pheromone image tests were done on a static image, meaning that the trail which was to be followed had an infinite lifespan. This essentially gave the robots an unlimited amount of time and attempts to establish a following behavior and make it to the end goal. However, this would not be the case in the follow the leader experiment in which the trail pheromone left in the environment only are detectable for a certain amount of time.

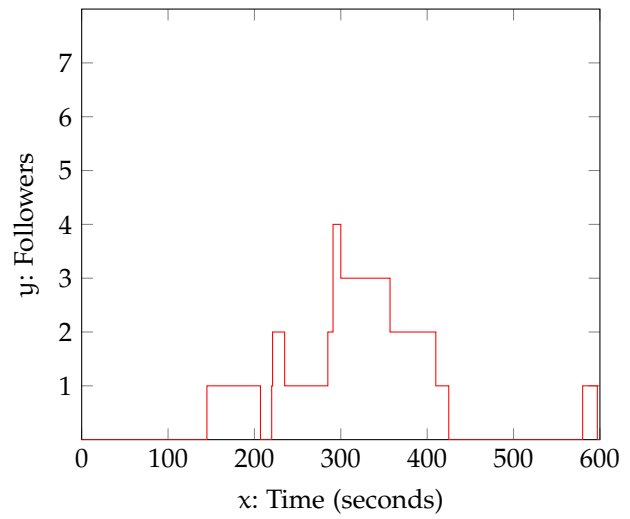
The results measured from the experiment done with a pheromone diameter of 7 cm and a lifespan of 30 seconds is shown in graph 7.1a and table 7.2a. The table shows the time in seconds in which the Kilobot had a certain number of followers and how much that time made up of the total experiment time in percent. The graph shows the amount of Kilobots who were following the leader and for how long. According to the graph the results are showing that after approximately 250 seconds the leader always has at least one follower for remainder of the experiment. However, a problem with these experiments were that Kilobots would get stuck to the wall of the arena making it unable to move until it managed to break free. This caused the leader and the other robots to form a cluster where they all prevented each other from being able to move. This problem is the reason for the stability in followers which is achieved approximately 400 seconds in to the experiment. The first 300 seconds of the experiment is a more accurate representation of the performance, indicating that robots find the trail but quickly lose track of it because of its inability to quickly adjust to the trail.

The second experiment with multiple followers was conducted with a lifespan of 60 seconds, giving the Kilobots more time to detect and follow the trail. The results are given in graph 7.1b and table 7.2b. The problem with Kilobots clustering is still a factor for this test. All followers gained after the time of approximately 300 seconds are because of this clustering, and does not reflect the performance of the Kilobots' trail following algorithm. With an increase of 30 seconds pheromone lifespan, the results do not indicate an improvement in performance.

The follow the leader experiments carried out indicate that a lot of improvement needs to be made before the proposed method for pheromone-based communication can be used for stigmergic-interaction involving trail following. The arena, movement and sensing abilities are all aspects that need to be changed and/or improved.



(a) Plot for 30 second pheromone lifespan test



(b) Plot for 60 second pheromone lifespan test

Followers	Time (s)	Duration Percentage (%)
1	145	24,2
2	31	5,2
3	27	4,5
4	45	7,5
5	62	10,3
6	0	0
7	0	0

(a) Table for 30 second pheromone lifespan test

Followers	Time (s)	Duration Percentage (%)
1	145	24,2
2	73	12,2
3	57	9,5
4	9	1,5
5	0	0
6	0	0
7	0	0

(b) Table for 60 second pheromone lifespan test

Another common method to measure results with pheromones is the time it takes to reach *aggregation* [8, 13]. This means to measure the time taken until all robots are gathered together. This metric could be measured in the follow the leader tests, which would be done by measuring the time taken for all followers to coordinate and follow the pheromone trail simultaneously. However, since the tests never reaches full aggregation it is not a useful measurement.

Chapter 8

Conclusions

8.1 Review of Project Goals

To achieve sub-goal 1: *establishing an environment that allows for pheromone-based communication between agents* the environment architecture presented in section 3.1 was established. The architecture allowed pheromones to be displayed through a projector onto the arena, based on the position of the Kilobots. The camera is connected to a computer running a program that gathers the data of each frame and processes it to calculate the position of where a pheromone should be displayed. This means that *a program that can create and integrate artificial pheromones into the environment* has been established and sub-goal 2 is completed. The process of creating and displaying pheromones is explained in further detail in chapter 5. The Kilobots had to have ability to sense a pheromone and act accordingly depending on the collective behavior being carried out with the system (sub-goal 3). Therefore, a program made specifically for the Kilobots was developed, this program is demonstrated in section 5.4. The ambient light sensor on the Kilobot allowed them to detect whether it was on a pheromone or not, by measuring the brightness perceived around the arena. To evaluate the systems possibility of achieving stigmergic-interaction using pheromone-based communication (sub-goal 4) the *follow the leader* experiment mentioned in chapter 6 was carried out. Allowing an individual Kilobot (leader) to leave traces in the environment that can be sensed by other Kilobots (followers), which then triggers a certain action (following). This is inline with the principle of stigmergy and therefore sub-goal 4 can be marked as complete [7].

By completing these sub goals a flexible, scalable and simple system was achieved. The system allows for establishing pheromone-based communication within robotic multi-agent systems without the need for expensive hardware. It is also compatible with any robot equipped with suitable sensors that can be used to detect the pheromones displayed through the projector. By investing more into the hardware parts of the system, larger scale experiments and improved performance can be achieved. Because the robots are a separate component isolated from the system, it can be modified without having to worry about incompatibilities with the robots. Giving the user the flexibility to adjust the system to the robots being used and the experiments carried out.

8.2 Challenges

8.2.1 Arena and Movement Blocking

The arena made for the Kilobots to move and interact in, which is mentioned in section 4.2 was constructed in a way that gave rise to issues in movement. The walls of the arena blocked the movement of the Kilobots and there was no established way to eliminate this

problem, which affected the results significantly. However, when re configuring the arena by adding an additional wall layer explained in section 6.2, the Kilobot's possibility of making it out of a state of blocked movement was increased. This indicates that a wall lower than the Kilobot's body height and a circular shape diminishes the problem of their movement being blocked.

8.2.2 Ambient Light Sensor

The ambient light sensor mentioned in 4.1 could vary a lot in its readings. According to testing, this most likely was because of the sensors positioning on the Kilobot, the environment and its accuracy. The sensor was placed in a way that rotation of the Kilobot relative to the light from the projector could affect its value substantially. This problem occurred because the body of the Kilobot blocked light from reaching the sensor. The sensor was also unreliable when distinguishing between smaller changes in brightness. Therefore, the light gradient of the pheromone trail was undetectable by the robots. This could have been used to better guide the followers towards the leader creating a more robust trail following algorithm and improving results. The program of the Kilobots mentioned in section 5.4 takes the median of a sample of sensor readings to avoid anomaly values that do not reflect the state of the environment. However, this method was not able to combat the inability of accurately detecting smaller changes in values. Because of this the *turn 180* state responsible for detecting changes in pheromone intensity, allowing the Kilobot to detect a gradient was not used.

8.2.3 Trail Following

Optimizing the trail following algorithm was a hard task when the ambient light sensor input from the Kilobots was only predictable and accurate enough to detect whether it was inside a pheromone or not. The return back state mentioned in section 5.4.1 is based on effectively finding a way back to the pheromone trail when losing track of it. The return back movement was achieved through an educated guess of where to turn, making it slow and prone to errors due to random chance. This came to be a major problem in the follow the leader experiment presented in section 6.2 where the robots had to be fast and accurate in order to not lose track of the leader whose pheromones evaporate at a certain rate and eventually disappears.

8.2.4 Color Detection

A challenge that arose when using color detection within the system was the trade off between having pheromone bright enough to be easily detected by the Kilobots' sensors but also not too bright, making the LED of the Kilobot indistinguishable for the camera when inside a pheromone. The contrast between the LED and the pheromone is crucial in order to accurately calculate the positions of the Kilobots. The overall brightness in the environment is kept as low as possible and therefore pheromones displayed in the arena are bright enough for the camera to overreact to that light. This is amplified by the white, reflective surface of the arena itself and leads to the red colored Kilobot to blend in with the bright pheromones. This however has proved not to be an issue in the context of our experiments because the moment the Kilobot exits the pheromone it is found again. However, it could be a problem in experiments if they would be dependant on being able to track agents that are inside a pheromone.

An object detection solution was explored and showed somewhat promising results. By training a model with images of Kilobots and annotating those images, it was possible to create a detection model that was able to detect Kilobots. But this method was unreliable with its detection and hard to debug since the model doesn't show how it operates. It proved to require extensive work to create a well performing detection model, which made

it into an impractical solution. Therefore color detection was utilized instead because it was easier to implement and adapt to the system.

8.3 Context

The field of swarm robotics has a lot of real world applications and therefore has an impact on different aspects of society. This section demonstrates the ways in which swarm robotics affect society and important questions to consider when developing such systems.

8.3.1 Ethical Considerations

The rapid advancement of robot technology have raised important ethical and moral questions. There are a things to keep into consideration. Who would be accountable for the robots' actions and consequences? What if the robots are taking jobs from people? How do we avoid information leakage events caused by increased robot interaction? These potential social problems prove the necessity of promoting safe applications for robotics. It also requires cooperation between governing adaptable laws and technological advancement. [29]

There is also the context of both defensive and offensive military use with swarm robots, more common with drones. A defensive application has been created that managed to detect and discriminate targets carrying weapons. The misuse of this method could be to target specific people based on their physical appearance. There have also been offensive uses, like drone attacks to government military bases. [6]

8.3.2 Environmental Aspects

Swarm robotics can be used to develop systems that are sustainable for the environment and/or of help in the journey of fighting climate change. *SwarmFarm Robotics*; a robotics company from Australia has developed an autonomous farming platform based on swarm robotics. Their platform increases the efficiency of the agricultural sector and therefore combats the climate challenges that are currently faced. It also improves sustainability by using less herbicide and reduces the environmental impact of pesticide spray drift. [30]

Research from Xi Yu, an assistant professor at the University of West Virginia presents a method that could allow for investigating otherwise inaccessible waters under ocean ice with a swarm of marine robots. Xi Yu's project is based on developing technologies that can control the swarm of robots so that they can reach the depths of the waters and communicate what they have learned. By investigating these waters important information to understanding the impact of climate change can be obtained. [31]

8.4 Future Works

Future work with this system could involve addressing several key challenges that were identified during the development. With the approach of using a projector with light sensors, there is high potential for improvement by refining the light values. This could result in more consistent pheromone detection and could enhance the movement to stay on the correct pheromone path more consistently. With increasingly robust light readings, the system could also enable more fine tuned actions by the Kilobots. This could enable implementations like finding the correct direction of the pheromone path, and adjusting the direction before even losing the trail. The light values are also very dependent on the light in the work environment that the tests are carried out. This means that natural light from

lamps in the room or sunlight have a big impact of the sensors input. With a future adaptation to the overall brightness in the work environment it could be easier to implement such methods.

On the Kilobot, the ambient light sensor is flat and positioned on top of the PCB, which means that the Kilobot itself may cast a shadow on the sensor if the light approaches at a sharp angle. A solution to this could be to have the ambient light sensor on top of the robot. But an alternative approach that doesn't require changes to the hardware on the robot, would be to project the pheromones from different angles. By using two or more projectors at an angle into the arena instead of one from the top, there would be less blind spots on the robots. This could also improve light detection. It could however pose many more challenges. With the creation of different pheromone images, synchronization with the projectors and the computer and the possibly increased interference with the color detection program.

The arena was a prebuilt construct that wasn't adapted much to the Kilobots other than the flat surface for improved motion and messaging with the IR sensors. Therefore some issues with Kilobot collisions in the arena walls were found out as the project progressed. In a future adaptation the arena could be focused more to fit especially for the Kilobots' and therefore improve their performance in several factors. If the Kilobots would break free from the walls easily, the tests could give more promising insights in the performance of the pheromone behavior.

The color detection has potential to improve by detecting a robot within a pheromone. With another camera better suited to detect contrast in colors, it could be easier to detect a Kilobot even while it is inside a pheromone. In a future modification it could also be possible to reconfigure the camera settings to adjust to the brightness in the pheromone filled environment. With that solution the light emitted from the Kilobot could be easier to distinguish and make the whole system potentially perform better. With more consistent pheromone projections in the arena, the robots would have an easier time following the pheromones.

Exploring pheromone behaviors beyond *follow the leader* could also provide valuable insights into the dynamics of pheromone trails. However, since the system at its current stage can not perform a follow the leader scenario of high quality it could be hard to implement something more complex without developing the system further.

There is also the possibility to utilize a completely different swarm robot. This could potentially bring interesting results to research with pheromone behaviors that are beyond the capabilities of the Kilobot. Since the Kilobot is very restricted by its limited precision in movement and its sensing, this could possibly be an interesting approach as well.

Acronyms

AASS Centre for Applied Autonomous Sensor Systems

ARK Augmented Reality for Kilobots

BCS Base Control Software

cm centimeter

DNF Did Not Finish

FPS Frames Per Second

GPU Graphical Processing Unit

GUI Graphical User Interface

HDMI High-Definition Multimedia Interface

HSV Hue Saturation Value

ID Identification

IDE Integrated Development Environment

IR Infrared

LCD Liquid Crystal Display

LED Light Emitting Diode

m meter

mm millimeter

OHC Overhead Controller

OpenCV Open Source Computer Vision

PCB Printed Circuit Board

RFID Random Frequency Identification

USB Universal Serial Bus

References

- [1] Tirindelli R, Dibattista M, Pifferi S, Menini A. From Pheromones to Behavior. *Physiological Reviews*. 2009;89(3):921-56. PMID: 19584317. Available from: <https://doi.org/10.1152/physrev.00037.2008>.
- [2] Group SOSR. Biological Collectives;. Available from: <https://ssr.seas.harvard.edu/insect-collectives>.
- [3] Klowden MJ. CHAPTER 12 - Communication Systems. In: Klowden MJ, editor. *Physiological Systems in Insects (Second Edition)*. second edition ed. San Diego: Academic Press; 2008. p. 597-642. Available from: <https://www.sciencedirect.com/science/article/pii/B9780123694935500134>.
- [4] Iglesias A, Gálvez A, Suárez P. Chapter 15 - Swarm robotics – a case study: bat robotics. In: Yang XS, editor. *Nature-Inspired Computation and Swarm Intelligence*. Academic Press; 2020. p. 273-302. Available from: <https://www.sciencedirect.com/science/article/pii/B9780128197141000269>.
- [5] Schranz M, Umlauf M, Sende M, Elmenreich W. Swarm Robotic Behaviors and Current Applications. *Frontiers in Robotics and AI*. 2020;7. Available from: <https://www.frontiersin.org/articles/10.3389/frobt.2020.00036>.
- [6] Arnold R, Carey K, Abruzzo B, Korpela C. What is A Robot Swarm: A Definition for Swarming Robotics. In: 2019 IEEE 10th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON); 2019. p. 0074-81.
- [7] Heylighen F. Stigmergy as a universal coordination mechanism I: Definition and components. *Cognitive Systems Research*. 2016;38:4-13. Special Issue of Cognitive Systems Research – Human-Human Stigmergy. Available from: <https://www.sciencedirect.com/science/article/pii/S1389041715000327>.
- [8] Salman M, Garzón Ramos D, Birattari M. Automatic design of stigmergy-based behaviours for robot swarms. *Communications Engineering*. 2024.
- [9] Zedadra O, Seridi H, Jouandeau N, Fortino G. Design and analysis of cooperative and non cooperative stigmergy-based models for foraging. In: 2015 IEEE 19th International Conference on Computer Supported Cooperative Work in Design (CSCWD); 2015. p. 85-90.
- [10] Tang Q, Xu Z, Yu F, Zhang Z, Zhang J. Dynamic target searching and tracking with swarm robots based on stigmergy mechanism. *Robotics and Autonomous Systems*. 2019;120:103251. Available from: <https://www.sciencedirect.com/science/article/pii/S0921889019301538>.
- [11] Zablotny JE. Chapter 237 - Sociality. In: Resh VH, Cardé RT, editors. *Encyclopedia of Insects (Second Edition)*. second edition ed. San Diego: Academic Press; 2009. p.

- 928-35. Available from: <https://www.sciencedirect.com/science/article/abs/pii/B9780123741448002460>.
- [12] Traniello JFA. Chapter 219 - Recruitment Communication. In: Resh VH, Cardé RT, editors. *Encyclopedia of Insects* (Second Edition). second edition ed. San Diego: Academic Press; 2009. p. 869-77. Available from: <https://www.sciencedirect.com/science/article/pii/B9780123741448002289>.
 - [13] Tang Q, Ding L, Li J, Zhang Y, Yu F. A stigmergy-based aggregation method for swarm robotic system. In: 2017 IEEE Symposium Series on Computational Intelligence (SSCI); 2017. p. 1-6.
 - [14] Sugawara K, Kazama T, Watanabe T. Foraging behavior of interacting robots with virtual pheromone. In: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566). vol. 3; 2004. p. 3074-9 vol.3.
 - [15] Arvin F, Krajník T, Turgut AE, Yue S. COS: Artificial pheromone system for robotic swarms research. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2015. p. 407-12.
 - [16] Reina A, Cope AJ, Nikolaidis E, Marshall JAR, Sabo C. ARK: Augmented Reality for Kilobots. *IEEE Robotics and Automation Letters*. 2017;2(3):1755-61.
 - [17] Editorial team. The untapped potential of virtual sensors;. Available from: <https://www.iec.ch/blog/untapped-potential-virtual-sensors>.
 - [18] Antoun A, Valentini G, Hocquard E, Wiandt B, Trianni V, Dorigo M. Kilogrid: A modular virtualization environment for the Kilobot robot. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2016. p. 3809-14.
 - [19] Valentini G, Antoun A, Trabattini M, Wiandt B, Tamura Y, Hocquard E, et al. Kilogrid: a novel experimental environment for the Kilobot robot. *Swarm Intelligence*. 2018;12:245-66.
 - [20] K-Team. SPECIFICATIONS;. Available from: <https://www.k-team.com/mobile-robotics-products/kilobot#specifications>.
 - [21] Rubenstein M, Ahler C, Hoff N, Cabrera A, Nagpal R. Kilobot: A low cost robot with scalable operations designed for collective behaviors. *Robotics and Autonomous Systems*. 2014;62(7):966-75. Reconfigurable Modular Robotics. Available from: <https://www.sciencedirect.com/science/article/pii/S0921889013001474>.
 - [22] K-Team. Documentation;. Available from: <https://kilobotics.com/documentation>.
 - [23] K-Team. Kilolib: Kilobot Library API;. Available from: <https://kilobotics.com/docs/index.html>.
 - [24] Optoma. User manual;. Available from: https://www.audiogeneral.com/Optoma/ml750_manual.pdf.
 - [25] Maitlo N, Noonari N, Ghanghro SA, Duraisamy S, Ahmed F. Color Recognition in Challenging Lighting Environments: CNN Approach. *arXiv preprint arXiv:240204762*. 2024.
 - [26] Student FourK. Choosing the correct upper and lower HSV boundaries for color detection with cv::inRange (OpenCV);. Available from: <https://stackoverflow.com/questions/10948589/choosing-the-correct-upper-and-lower-hsv-boundaries-for-color-detection-withcv>.
 - [27] Sedlacek P, Kmec M, Rusnak P. Software Visualization Application for Threads Synchronization Handling in Operating Systems. In: 2020 18th International Conference on Emerging eLearning Technologies and Applications (ICETA); 2020. p. 580-5.

- [28] winAVR. Introduction;. Available from: <https://winavr.sourceforge.net/>.
- [29] Li K, Meng Z. Ethics of Robotics Applications. In: J Jansen B, Liang H, Ye J, editors. International Conference on Cognitive based Information Processing and Applications (CIPA 2021). Singapore: Springer Singapore; 2022. p. 325-30.
- [30] Australian Government. SwarmFarm Robotics is transforming the future of farming;. Available from: <https://www.industry.gov.au/news/swarmfarm-robotics-transforming-future-farming>.
- [31] Statler B. WVU researcher to help send swarm of marine robots on climate change quest beneath ocean ice;. Available from: <https://wvutoday.wvu.edu/stories/2023/09/26/wvu-researcher-to-help-send-swarm-of-marine-robots-on-climate-change-quest-beneath-ocean-ice>.