

Final Project Proposal
Collection of papers in
Swagat Robotics



Submitted to the
Project Committee appointed by the
International School of Engineering (ISE)
Faculty of Engineering, Chulalongkorn University

Project Advisor
Asst.Prof.Paulo Fernando Rocha Garcia, Ph.D.

Submitted By

6 4 3 8 0 6 7 0 Natthasorn
6 4 3 8 0 7 5 0 Tinnaglini
6 4 3 8 0 7 9 6 Tinnapattisila
6 4 3 8 1 4 2 1 Noppa Wankhirin
6 4 3 8 1 8 7 7 Mehsarma

2/2024: 2147416 Final Project I
Robotics and Artificial Intelligence Engineering (International Program)
International School of Engineering (ISE) Faculty of Engineering, Chulalongkorn University

Contents

1	Research Background	2
2	Objectives	4
3	Literature Survey and Review	5
3.1	Coordination	5
3.2	Object Detection	6
3.3	SLAM	11
3.4	Collective Movement	14
4	Project Concept Development	17
5	Project Planning and Timeline	18
6	Theory Backup	20
6.1	Omnidirectional Movement	20
6.2	Waypoint Navigation	20
6.3	Swarm Robotics Theory	20
6.4	SLAM	21
6.5	Object Detection and Computer Vision (CV)	21
6.6	Multi-Robot Collaboration and Coordination	21
6.7	Grasping and Object Manipulation in Robotics	21
7	Project Outcome	22
8	Summary and Benefits to Real Industry	25
9	Project Contribution per Student	26

Abstract

This report serves as the proposal for our final project, entitled "Collective Transport using Swarm Robotics." It covers various facets of our project, including the project concept, expected outcomes, project timeline, and potential benefits to the industry. Additionally, a comprehensive review of existing literature and a robust theoretical foundation are provided to substantiate our objectives.

Structured into nine chapters, the report begins with an exploration of the research background and our objectives. The literature survey evaluates multiple research papers to identify relevant concepts, enhancing and applying them in innovative ways. Subsequent chapters delve into the development of the project concept, detailing the planning and execution strategy. A theoretical backup section fortifies our project methodology. The anticipated project outcomes are discussed, followed by the potential benefits to the industry. The report concludes by outlining each team member's contributions to the project.

As a proposal for our final project, this report is intended to outline our plans and should not be considered as a finalized product. The comprehensive information provided across the nine chapters aims to furnish sufficient details for an effective project proposal.

Chapter 1

Research Background

In recent years, swarm robotics has gained significant popularity in mobile robotics. Researchers have explored various aspects to create robot swarms, focusing on communication between the robots, the coordination of each robot, navigation, and the realization of animal-inspired behaviors in robotic systems [1]. Additionally, certain properties exhibited by natural swarms have been identified as crucial for swarm robotics. G. Beni proposes the following properties: Flexibility, defined as "the capability to adapt to new, different, or changing requirements of the environment" [2]; Scalability, where "a swarm system is said to be scalable if it can work with different numbers of its members" [3]; and Robustness, "the ability of a swarm robotic system to continue operating, although at a lower performance, despite disturbances in the environment or failures in individuals" [4]. However, additional properties are essential for swarm robotics systems that researchers address when designing their robots. These properties include Autonomy, the ability of individual robots within a swarm to operate independently while coordinating with others to achieve a common goal; Self-organization, "a process whereby pattern at the global level of a system emerges solely from interactions among the lower levels of the system... using only local information, without any central authority determining their course of action" [1]; Self-assembly, where "the robots not only stay close to each other, but they also are able to connect themselves, forming a single organism" [3]; Decentralization, where "each individual makes its own decisions based on local information" [5]; and Stigmergy, described as "a form of indirect communication between natural or artificial agents where the work performed by an agent leaves a trace in the environment that stimulates the performance of subsequent work by the same or other agents" [1].

Given the current state of the art, it is still impossible to create a perfect swarm robot system that perfectly manifests such properties, let alone the challenges of dealing with the sub-system inside the robot system itself, for example, the implementation of Simultaneous Localization and Mapping (SLAM) on robot navigation. Moreover, the metrics for evaluating swarm SLAM methods are impractical in real-life situations; one might make arbitrary decisions on the quantification of aspects [6].

This project aims to address the communication, coordination, and navigation challenges within decentralised swarm robot systems. By enabling the robots to share environmental data and make collective decisions in real time, this framework can improve

flexibility, scalability, and adaptability in diverse industrial applications.

Chapter 2

Objectives

Many physical feats to automate daily tasks are only available using robotic solutions. However, a single complex robot is not viable in every scenario due to its limited reach and high spatial and monetary investment. Thus, we turn to a simpler, more scalable solution, which is swarm robotics. Our objective is to have 3 decentralised robots working together to achieve a goal, more specifically collective transportation of an object for cleaning/tidying up a room. This requires robustness in localisation and awareness of other robots' movements.

Given the time constraints, creating a finished fleet of cleaning robots would be unrealistic. The project is then scaled down and split into 2 major milestones: localisation with object identification, and object grasping through coordinated formations. The first milestone is crucial as it provides a functional component relevant to domestic use cases, enabling object identification, localization, and communication, all of which are key to managing a scalable fleet. Secondly, object grasping through coordinated formations is a crucial follow-up as it expands further upon an abstract concept to pinpoint specific use cases such as tidying items in a room. For this semester we will be prioritising simulation, more specifically simulating a system with three robots on Webots, ensuring collaborative localisation, and establishing communication between them. For computer vision (CV) object detection, a video of the room will be taken to train the model on identifying objects. When concerned with the hardware, we aim to have one robot having elementary movement. The other two robots will soon be purchased.

Chapter 3

Literature Survey and Review

3.1 Coordination

Swarm robotics solution is a design architecture that obtains inspirations from biological interactions; thus, they should operate autonomously to solve problems rather than relying on a central authority[7]. This decentralised approach is also crucial to achieve increased resilience and flexibility given a complex environment of deployment[8].

Communication protocols can be categorised into two principal types depending on the nature of information transmission: Direct communication and Indirect communication[8]. Direct communication refers to robotic agents that are able to coordinate via networked communication. Direct communication use cases are highlighted in many studies.

Ibrahim et al. [9] studied direct communication to determine the optimal distance for robotic agents to achieve consensus. Three strategies were tested within a 50 cm range: Close-neighbour, Far-neighbour, and Rand-neighbour. Close-neighbour excels in stable environments but performs poorly in complex ones. Both Close-neighbour and Far-neighbour introduce bias, reducing accuracy. Rand-neighbour, which randomly selects swarm members for communication, proved superior due to efficient information flow and minimal bias. This strategy can be one of our key designs for swarm communication in this project.

Ayari and Bouamama[10] and Perera et al.[11], S et al.[12] and Z. Wang et al.[13] conducted studies to promote robots' actions based on their sensory observations and objectives. S et al.[12] proposed a Multi-Agent Deep Deterministic Policy Gradient (MA-DDPG), an observation-based decision making flow with an architectural twist, where there is a central swarm manager that evaluates decentralized agents' reinforcement learning for optimal rewards. Z. Wang et al.[13] proposed increasing sensory inputs from both visual data and communication for redundancy, which highly aligns with the proposed swarm system.

Yasser et al.[14] expanded more on Clustered Dynamic Task Allocation (CDTA), an approach to dynamically assign tasks based on swarm state and the environment [15], for the purpose of increasing the velocity of swarm communication. They proposed

CDTA-CL (Centralized Loop) and CDTA-DL (Dual Loop). CDTA-CL sends information to the leader for computation, while CDTA-DL compares information before sending it to the leader. CDTA-DL outperformed CDTA-CL, increasing speed by 75.976% compared to 54.4%. CDTA-DL will be considered as a design pillar for this project.

In addition to direct communication improvements, indirect communication, known as Stigmergy, also plays a significant role in swarm intelligence. This concept involves individual robot actions modifying the environment, impacting the decision-making of other robots. For example, construction robots can leave blocks and materials to signal ongoing work[8]. This is an intriguing notion to consider in the interaction of the swarm system.

For effective communication, especially in interdependent tasks, robots need to be aware of each other and task requirements. Semantic communication, where contextually relevant data is prioritized, is necessary[16]. The balance between communication and context must align with task demands. High sensory data tasks require reliable, real-time communication, while tasks with lower communication demands can prioritize contextually relevant information[17].

3.2 Object Detection

Object Detection plays a crucial role in computer vision by identifying and locating objects within images or video feeds. Its primary goal is to identify objects as well as determining its precise location through bounding boxes. In order for the robots to interact with the environment and eventually complete their task, it is crucial that they have to understand the dynamic and complex surroundings with their sensors, followed by recognising all targets around them as well as targeting them. Another key requirement is that each robot should distinguish other team members from objects in the environment, known as “Kin detection”, and know their relative position to others with data from their sensors

In various real-world applications, such as autonomous vehicles [18] and multi-robot coordination in warehousing [19], 2D object detection is effective for basic identification and localization tasks. However, detecting objects in complex environments presents challenges, particularly when object occlusion occurs [20]. The limitations of 2D object detection, which relies on single-plane images, make it difficult to accurately identify objects that are distant or situated in cluttered settings [21].

Three-dimensional data can be acquired using various devices such as RADAR, RGB-D cameras, and stereo cameras [22]. Each of these devices has distinct advantages and disadvantages:

RADAR employs radio signals to measure distance and estimate the shape and size of objects but does not detect colour and has low spatial resolution [22]. This limitation makes it challenging to distinguish between closely spaced or thin objects [23]. However, RADAR is particularly effective in poor visibility conditions, such as for collision avoidance and adaptive cruise control.

RGB-D cameras, like the Intel RealSense D455, capture both colour and depth information, which allows for detailed spatial analysis [24]. These cameras use techniques such as time-of-flight and structured light to measure depth [24]. The data from RGB images and depth maps can be processed to create 3D point clouds or models, facilitating object detection and scene analysis. These cameras are adept at perceiving attributes like colour and shape in real-time, making them suitable for dynamic environments. Nonetheless, their performance can be impacted by lighting conditions, which affects depth accuracy, and they have a more limited sensing range compared to RADAR.

Stereo cameras operate with two lenses, capturing left and right images to mimic human vision [25]. The data collected typically consists of a 3D point cloud that shows the spatial distribution of objects in the environment. Creating this 3D point cloud involves processing stereo images to identify matching points and calculate their depth. While stereo cameras can detect distant and small objects due to their dense pseudo point cloud[26], they may struggle with object detection in dynamic environments because of noise in the point cloud data during rapid changes[27].

Considering stereo camera and RGB-D camera performance, as it can perceive colour which is the main characteristic that is required in object detection and classification, in 3D object detection. The RGB-D camera produces 0.82% average percentage error, while the stereo camera produces 2.61% average percentage error measuring under the same condition and doing the same tasks in multi-object detection[28].

Considering the data format obtained from the selected device which is the RGB-D camera will help to identify the suitable candidate of the 3D object detection approach. The input RGB-D data contain two formats, one with normal colour image which is red, green, and blue, another one uses black and white to represent the distance information in a single channel, The darker colour represents an object that is further from the camera than the ones with lighter colour.

Given the variety of tools and algorithms used in 3D object detection research with RGB-D cameras, a proper quantitative comparison has not been consistently conducted. Only a few approaches can be categorised into one of the four main groups of 3D object detection. Therefore, this section aims to provide an overview of the key techniques used. Each technique will include a discussion of its core concepts, along with its strengths and weaknesses. A quantitative comparison will be presented at the end of this section.

There are four major categories for 3D object detection using RGB-D cameras. Firstly, **2.5D processing methods**. This method utilises the 3D convolution kernel. The concept of this method is to use depth images obtained from the RGB-D camera as 2D images directly. Additionally, HHA encoding is used for enhancing the geometric representation including object's height, surface angle compared to the ground etc.[29]. The combination of RGB and depth data in this approach enhances detection accuracy, especially in complex environments[29][30]. However, it can potentially perform not

so well in highly dynamic or unstructured scenes especially when depth data is less accurate[29].

Secondly, **2D Driven 3D Methods** combine 2D object detection techniques with 3D processing to increase efficiency of 3D object detection. It basically narrows down the space with 2D detection followed by refining and defining 3D object boundaries using depth data. While RGB data remain unchanged, the depth data are transformed into point clouds[29]. A more advanced model called “The series of Frustum PointNet” which utilises RGB-D data for enhanced 3D object detection by integrating 2D image information (RGB data) with 3D point cloud data. 3D frustums are transformed from 2D object proposals using camera projection metrics allowing effective feature extraction and object localisation using several advanced neural networks e.g. PointNet and LDG CNN. The integration of RGB and depth data boost the detection performance especially in small object detection like pedestrian as well as optimise computational resources while maintaining high detection rates[31]. However, the complexity of the system and processing time can be increased from relying on both point cloud and RGB data[32].

Geometric Descriptor-based Methods utilises depth information from RGB-D camera for enhancing object recognition capabilities by using geometric features extracted from RGB image and depth data. This category comprises three main methods; Center of Gravity (COG) utilises the geometric centre of the detected objects’ features to improve the accuracy of the objects’ localisation. The boundaries of the object are refined and the detection reliability is enhanced by analysing spatial distribution of points in 3D space. Latent Surface Shape (LSS) identifies the pattern of the indoor settings by focusing on the surface that an object rests on. This approach helps explain the differences in object shapes. Although this approach allows small object detection to be done easier, it can struggle with objects with unclear supporting surfaces[33]. Lastly, H3DNet[34] is a neural network that uses geometric shapes to detect objects. A detailed description of points is created for predicting the object’s shapes. This is then used to make and improve proposals of object detections. The network refines object detection by adjusting the bounded box or other geometrical shape followed by classifying the object and eventually adding labels[29].

Lastly, **3D convolution based models** which highlight the importance of texture data that is obtained from the 3D object detection. There are several methods utilising both image and point clouds for enhancing overall performance with fusion schemes implementation.

The series of sliding windows including The Sliding Shapes method and Deep Sliding Shapes (DSS). The Sliding Shape approach performs the object classification using 3D sliding windows which are able to handle occlusions and changes in viewpoint effectively. However, the processing can be slowed down due to the hand-crafted features which this method relies on extensively. For the Deep Sliding Shapes, a 3d convolutional neural network (CNN) is used to improve the efficiency upon the Sliding Shapes method. This significantly solved two main challenges; speeding up the target

3D object detection performance comparisons on 10 classes on SUN RGB-D datasets.

Methods	Bathtub	Bed	Bookshelf	Chair	Desk	Dresser	Nightstand	Sofa	Table	Toilet	mAP-10	Running Time
DSS [45]	44.2	78.8	11.9	61.2	20.5	6.4	15.4	53.5	50.3	78.9	42.1	19.55 s
2D Driven [56]	43.5	64.5	31.4	48.3	27.9	25.92	41.9	40.39	37.0	80.4	45.1	4.2 s
PointFusion [66]	37.26	68.57	37.69	55.09	17.16	23.95	32.33	53.83	31.03	83.80	45.4	1.3 s
COG [61]	58.3	63.7	31.8	62.2	45.2	15.5	27.4	51.0	51.3	70.1	47.6	10-30 m
3D-SSD [57]	57.1	76.2	29.4	9.2	56.8	12.3	21.9	1.7	32.5	38	50.7	0.22 s
LSS [62]	76.2	73.2	32.9	60.5	34.5	13.5	30.4	60.4	55.4	73.7	51	
F-VoteNet [60]	43.3	81.1	33.3	64.2	24.7	32.0	58.1	61.1	51.1	90.0	53.89	0.12 s
F-PointNet [58]	43.3	81.1	33.3	64.2	24.7	32.0	58.1	61.1	51.1	90.9	54.0	0.12 s
VoteNet [50]	74.4	83.0	28.8	75.3	22.0	29.8	62.2	64.0	47.3	90.1	57.7	
MLCVNet [68]	79.2	85.8	31.9	75.8	26.5	31.3	61.5	66.3	50.4	89.1	59.8	
H3DNet [63]	73.8	85.6	31.0	76.7	29.6	33.4	65.5	66.5	50.8	88.2	60.1	
HGNet [69]	78.0	84.5	35.7	75.2	34.3	37.6	61.7	65.7	51.6	91.1	61.6	
Group-Free [49]	77.5	87.0	35.7	78.1	32.0	37.2	66.7	70.2	52.2	91.3	62.8	
IMVoteNet [106]	75.9	87.6	41.3	76.7	28.7	41.4	69.9	70.7	51.1	90.5	63.4	

Figure 3.1 – 3D object detection performance comparisons on 10 classes on SUN RGB-D datasets. Table reused from *Recent advances in 3D object detection based on RGB-D: A survey*[29]

detection and eliminating the need of CAD design that is done manually. Although the performance is enhanced, the demand for higher computation which is a result of 3D convolutions complexity.

VoteNet employs the voting mechanism when locating objects as well as generating high quality proposals which both help solve the sparse point cloud issue, but this can be limited by noisy or missing data points. Additionally, imVoteNet is built on the VoteNet by incorporating 2D image data together with texture and semantic information added. This enhances the object detection performance especially when there is a sparse point cloud data. Although the 2D and 3D data fusion increases the accuracy, the complexity is raised as well.

The VoteNet’s object detection is enhanced by BRNet which traces representative points back to their original vote centres and re-examining point clusters, allowing a more detailed understanding of object structures which also comes with higher cost of computational requirement due to higher complexity. MLCVNet improves VoteNet’s performance by incorporating multi-level contextual information which enables the model to better grasp the relationships between objects and their surroundings, leading to more accurate detections. However, the increased complexity of simultaneously modelling both global and local context causes higher computational costs.

The Hierarchical Graph Network (HGNet) is the VoteNet method that adopts a graph-based approach for improving object detection performance. This features a shape-attentive feature extractor as well as integrated global scene context, making the prediction even more accurate. However, the improved detection precision of this approach leads to higher computational cost as well.

The author performs comparison of the 3D object detection performance of each approach by using the SUN RGB-D dataset which is a single-view RGB-D dataset containing 47 distinct indoor scenes shown in Figure 3.1.

Although 3D convolution-based models tend to outperform other approaches, it’s important to consider additional factors when selecting a method for our project, such

as computational requirements, cost, deployment, integration with SLAM, and hardware compatibility.

As some 3D object detection approaches build upon 2D object detection methods, such as convolutional networks, it's equally important to evaluate the various available 2D object detection approaches in terms of their concepts, functionality, and performance.

There are two major stages in 2D object detection development; the traditional stage and the new stage with deep learning. The concept of traditional object or target detection uses sliding window methods to generate boxed on target images or videos followed by manual feature extraction. Lastly, classifiers like Support Vector Machine (SVM) and Logistic regression will classify the extracted features and there will be a box bounded around the target position. This can be seen in some models like SIFT and Cascades and some typical algorithms are HOG[35], Viola Jones[36], etc. which have limitations causing low detection speed and precision.

In the second stage, Convolutional Neural Network (CNN) helps increase average detection accuracy by approximately 30%[37]. There are two major algorithms in target detection with deep learning, one stage target detection and two-step target detection.

One-stage object detection algorithms predict both the coordinates and class probabilities of objects in a single step using a single neural network[38]. Two prominent examples of these algorithms are Single Shot MultiBox Detector (SSD) and You Only Look Once (YOLO), both of which perform differently in various applications. SSD employs a VGG16 backbone for feature extraction and generates predictions at multiple layers, enabling it to handle objects of varying scales without the need for a region proposal network[39]. On the other hand, YOLOv8, the latest iteration of the YOLO family, uses a more advanced Dark-53 backbone, alongside improved data augmentation techniques and anchor box clustering, which collectively boost its accuracy and efficiency[37]. To compare the performance of SSD and YOLOv8 in multi-object detection tasks, factors such as accuracy, adaptability, and speed are analysed. This comparison is based on the findings from the study "Performance Analysis of YOLOv8, RCNN, and SSD Object Detection Models for Precision Poultry Farming Management," where SSD, YOLOv8, and Faster R-CNN were tested for multi-object detection, as shown in Table 3.1.

Additionally, Two-stage Target Detection involves two steps: the first stage detects potential object locations, while the second stage refines these locations using a deep neural network to extract features from the proposed regions[38]. The addition of a Region Proposal Network (RPN) in the second stage introduces extra computational demands, resulting in higher processing time and hardware requirements compared to one-stage detectors[40]. The main detectors in this category include; R-CNN detector uses selective search to generate region proposals before applying CNN to each region to classify the target and bound each target with boxes. However, this ap-

proach is computationally expensive as thousands of regions are processed by CNN independently[20].

Fast R-CNN improves R-CNN by integrating region proposal generation and classification into one network for faster processing. It uses a CNN to process the image and classify ROIs from a shared feature map but still relies on external region proposals, limiting its real-time performance[41].

Faster-CNN eliminates the need of selective search as it features an internal Regional Proposal Network (RPN) offering a strong balance between speed and accuracy which makes it suitable for real-time task[42].

In addition to the aforementioned characteristics of various one-stage and two-stage target detection algorithms, three selected models; YOLOv8, SSD, and Faster-CNN are compared focusing on their speed, precision, and adaptability. The result is measured from the multi-object detection in precision poultry farming management.

Algorithms	Recall Value	Mean Average Precision (mAP@0.5)	Precision
YOLOv8	1.00	98.7%	96.77%
SSD	0.65	77%	89%
Faster R-CNN	0.67	59%	77%

Table 3.1 – A comparative analysis of YOLOv8, SSD, and Faster R-CNN based on key performance metrics, including Recall Value, Mean Average Precision (mAP@0.5), and Precision. Table reused from *Performance analysis of YOLOv8, RCNN, and SSD object detection models for precision poultry farming management*[43]

Following this comparison, it is evident that YOLOv8 excels in multi-object detection, combining high accuracy with balanced speed and efficiency. This makes YOLOv8 particularly well-suited for real-time applications. Furthermore, YOLOv8's lower hardware demands make it a more practical choice compared to the more resource-intensive Faster R-CNN and other two-stage detectors[44].

3.3 SLAM

SLAM, or Simultaneous Localization and Mapping, is a widely spread algorithm for navigation in the field of mobile robotics because of the exponential improvement in computer processing speed and the accessibility of sensors such as cameras and LiDAR [45]. Using SLAM, a mobile robot can construct an internal environment map while simultaneously using the map to estimate its location without needing predefined knowledge of area [46].

Environment mapping is one of the vital techniques in SLAM. The algorithm consists of building a mathematical model for the spatial information of an actual environment, which encapsulates the necessary information for navigation and interaction.

However, as for the SLAM technique, additional requirements are needed; the mathematical model must be able to represent the robot's state and the position of landmarks relative to the robot's location [46]. Hence, the challenge with the requirements is that the robot must perform the localization and the mapping simultaneously.

Given these complexities, the backbone of all principal SLAM methods is the utilization of these SLAM frameworks consisting of odometry, landmark prediction, landmark prediction, landmark extraction, data association, and matching, pose estimation, and map update [47].

Building on this, situational awareness becomes an extreme component of SLAM, the precision and accuracy of the robot's perception play a huge role in defining the characteristics of other variations of the SLAM implementation. Therefore, a thorough understanding of advantages and disadvantages of each common perception device is an imperative concept not just for the robot's components but also the structure of the SLAM's backend algorithm.

Firstly, acoustic sensors are widely used across the preliminary stage of SLAM implementations to minimize the pose drift with time, with most of the sensors being SONAR, or Sound Navigation and Ranging [48]. These sensors are well operated in dark environments, as well as dusty and humid, due to their insensitivity towards illumination and opaqueness [49].

Secondly, LiDAR, or Light Detection and Ranging Sensor, is relatively similar to an ultrasonic sensor in terms of functionality [48]. However, LiDAR uses electromagnetic waves as a radiation reference instead of acoustic waves. A LiDAR renders a 3-dimensional representation of its surroundings known as the Point Cloud [50]. The strength of LiDAR is that the sensor can provide 360 degrees of perception with high precision [51].

Thirdly, depth cameras' mechanism works based on the illumination of the site with infrared light and measures the time-of-flight [52]. Comparing the range of measurement and accuracy, a depth camera performs poorer than a 3D LiDAR scanner because the depth camera can only acquire data within a limited range of field of view; moreover, environmental factors may affect the accuracy of the depth camera; for example, the depth camera's output is susceptible to certain materials of surfaces, such as reflective or transparent materials [53]. However, a depth camera is still a popular option for SLAM as it's a relatively economical device compared to its relatives, 3-D LiDAR, for instance.

Ultimately, event-based cameras present the local bitmap-level motion alterations to an event that took place, which is different from conventional framing-based cameras [48]. The new technique has gained popularity more recently in the field of SLAM as an event-based camera yields more efficient computational performance and better overall accuracy [54].

After reviewing the different sensors and addressing technological advancements available in today's world, it is crucial to understand how researchers have implemented those ideas to different variations of SLAM. This understanding helps in overcoming challenges and limitations that their predecessors had faced and set new standards for new research frontiers. Moreover, it becomes essential to effectively classify those SLAM variations under different criteria.

Li et al.[26] perfectly encapsulated how SLAM techniques can be classified:

Simultaneous Localization and Mapping (SLAM) techniques can be categorized by using different factors. Firstly, they can be divided into categories based on the type of sensors employed. They may include vision-based SLAM using cameras, LIDAR-based SLAM using LIDAR sensors, and RGB-D SLAM, which combines RGB cameras with depth sensors. Secondly, feature-based SLAM, which tracks distinguishing characteristics and direct SLAM, which executes mapping intensity or depth directly can be considered as different categories. Thirdly, the estimated approach, such as filter-based SLAM, which uses filters such as Particle Filter and graph-based SLAM, which is formulated as a graph optimization problem, provides another classification criterion. Finally, SLAM can be categorized based on time synchronization, with offline SLAM processing data in batches after collection and online SLAM estimating pose and map incrementally in real-time.

Among the various SLAM methods, one of the most well-known is Visual Slam, or V-SLAM, a variation of SLAM that uses images from cameras, ranging from a conventional camera to an RBD-D camera (depth and ToF camera). V-SLAM itself can be divided into two sub-categories [55]. Firstly, a feature-based SLAM system that matches camera data using sparse methods; one example of this robust and popular algorithm is the ORB-SLAM [56]. Secondly, the direct dense methods that evaluate based on the general luminance of the pixels in images, one of the famous algorithms is Direct Sparse Odometry [57].

In addition to V-SLAM, another key variation is LiDAR-based SLAM, which utilizes LiDAR sensors to localize itself by collecting data from its surroundings while building the map of the data representation. Registration algorithms, for instance, iterative closest point (ICP), are used to estimate the relative transformation of the point clouds during the operation [58]. On the other hand, feature-based algorithms, such as LiDAR Odometry and Mapping, are used to represent 2D or 3D point cloud maps as grid maps [59].

Furthermore, by combining the strengths of different sensors and overcoming each's limitations, the multi-sensor SLAM utilizes multiple sensors, such as cameras, Inertial Measurement Units (IMUs), Global Positioning System, LiDAR, etc. FAST-LIO, or a Fast LiDAR-Inertial Odometry, is a decent example of an algorithm that combines LiDAR feature points with IMU data [60].

As SLAM continues to evolve, one of the emerging trends is collaborative SLAM, especially in a distributed framework. This includes multi-robot SLAM that leverages stability of the system by minimizing global error accumulation, risk concentration [61].

In this context, Slide-SLAM presents a real-time decentralized metric-semantic SLAM method that creates an object-based representation to append autonomous exploration functionality to a robot team. This is achieved by attaching a communication module to each robot, and then a unified map is obtained from each robot's observation [62].

Similarly, C-SLAM, an open-source system of Swarm-SLAM that has main purposes to be a scalable, decentralized, and sparse system for multi-robot, especially swarm-like robot teams, to perform navigation operations in unknown environments. C-SLAM is designed to support IMU, LiDAR, stereo, and RGB-D sensing [63].

3.4 Collective Movement

In the domain of swarm robotics, collective movement coordination and dynamic role assignment are crucial for enabling robots to work together efficiently. Research on coordinated motion in swarms often emphasises the need for algorithms that allow robots to adapt their roles and behaviours in real-time. For example, the study on "Efficient Strategies for Coordinated Motion and Tracking in Swarm Robotics" is a comprehensive overview of various coordination algorithms, contrasting different techniques for multi-robot collaboration.

The first coordination algorithm mentioned is the leader-follower model. This algorithm is rather straightforward in the sense that one or more robots are designated to guide the swarm while the other robots adjust their positions. The leader can be pre-programmed or autonomously chosen depending on the path while the followers maintain a set distance and set angle. This model as mentioned before is simple to apply while also being centralised providing clear direction for the followers. Additionally, the followers do not need the full knowledge of the environment meaning that this model can be scalable. However, this swarm being centralised means that it is prone to a single point of failure and having reduced flexibility [64]. This model would only work well for a simple structured environment with predefined paths which unfortunately does not match with our objectives.

Another coordination algorithm is the potential fields algorithm. This algorithm is based on virtual forces with each robot in the swarm being treated as a particle that is influenced by virtual forces exerted by other robots, obstacles and targets. These forces can attract or repel each other. The object is for the robot to be "pulled" towards the goal while avoiding collisions. This model has a couple advantages; namely: Decentralised control as each robot moves autonomously based on the forces acting on it, and smooth movements. However a couple of challenges come with it as well. One challenge is the possibility of the robot being stuck in a local minimum where virtual forces cancel each other out. Secondly, proper fine tuning of the force parameters is required to prevent the robot from oscillating [65]. Overall this approach could be useful

for environments with many obstacles where smooth and continuous navigation can be important. The third algorithm offered is the virtual force algorithm which is similar to the potential fields algorithm but with more constraints thereby being ineffectual to our project [48].

When comparing these three algorithms above, potential field algorithm and virtual force algorithm are decentralised while the leader-follower model is centralised. While the leader-follower model offers a simple, scalable nature, it introduces a single point of failure. In contrast, the potential fields algorithm offers a more decentralised approach, which is better suited for dynamic environments but requires careful turning to avoid local minima.

In swarm robotics, dynamic role assignment plays a crucial role in enabling robots to adapt their behaviours and tasks in real-time. One common method is insect-inspired behaviour, which mimics the role distribution seen in social insect colonies [66]. In this approach, robots assume roles based on simple, local rules, such as task demand or proximity to a target, without centralised control. This method offers high scalability and robustness, as robots can seamlessly take on different tasks as needed, making it suitable for large swarms. However, its reliance on local information can sometimes lead to suboptimal task assignments, particularly in complex environments where global awareness might be needed.

On the other hand, market-based approaches [67] use a more structured mechanism where robots bid for tasks based on their capabilities and availability. This ensures that tasks are allocated to the most suitable robots, leading to more efficient task execution. However, the bidding process requires communication between robots, which may introduce delays and increase system complexity. While market-based approaches tend to be more optimal for task allocation, they may not scale as easily as insect-inspired methods, especially in large or dynamic environments where constant communication is challenging.

Decentralised control in swarm robotics offers several key advantages, particularly in terms of scalability, robustness, and adaptability [68]. In decentralised systems, each robot operates autonomously, relying on local information and interactions with neighbouring robots, which eliminates the need for a central controller. This allows the swarm to scale more easily, as adding more robots does not increase the computational or communication burden on a single entity. Additionally, decentralised systems are more robust, as the failure of one or more robots does not compromise the entire system; each robot can continue functioning independently. This is particularly advantageous in dynamic or unpredictable environments, where flexibility and fault tolerance are critical.

In contrast, centralised control systems rely on a single controller to manage all robots, which creates a bottleneck as the number of robots increases. Centralised systems can suffer from single points of failure—if the controller fails, the entire system

may halt. Moreover, communication delays and computational limits can hinder real-time performance in larger systems. While centralised control offers more efficient coordination in smaller, simpler environments, decentralised control is better suited for real-world applications where scalability and resilience are essential for handling complex and dynamic tasks.

Despite significant advancements in swarm robotics and coordination algorithms, there remain notable gaps in applying these methods to practical, real-world environments such as cleaning tasks. Most research on algorithms like potential fields, leader-follower models, and market-based role assignment has focused on simulations or controlled environments, which often lack the complexity and unpredictability found in real-world scenarios. For instance, limited work has been done on integrating these algorithms with sensor-rich, dynamic settings where robots must navigate cluttered spaces, identify and manipulate diverse objects, and coordinate in real-time without centralised control. Additionally, the scalability of these systems is often not tested in practical, large-scale environments, such as a commercial building cleaning system, where communication constraints, battery life, and real-time decision-making are crucial factors.

Our project aims to address these gaps by developing a swarm of cleaning robots that leverages C-SLAM for real-time mapping and localization, along with a hybrid role assignment approach that adapts based on task proximity and robot capabilities. By testing in realistic environments with obstacles, diverse object types, and multiple robots working simultaneously, our project will not only explore the robustness of these algorithms but also refine them for practical deployment in domestic and commercial cleaning. This will bridge the gap between theoretical research and real-world application, offering a scalable and adaptive solution for multi-robot systems.

Chapter 4

Project Concept Development

In the near future, most repetitive tasks will inevitably be automated by robots. For instance, in the cleaning industry, fleets of robots will be required to communicate and collaborate effectively to optimise task completion. However, the current generation of robots typically operates in isolation, lacking the ability to be aware of or interact with other robots in their environment, which significantly hinders their overall scalability, efficiency, and coordination capabilities. However, a single complex robot is not viable in every scenario due to its limited reach, and high spatial and monetary investment.

Thus, we turn to a simpler, more scalable solution, which is swarm robotics. Our main goal is to provide a stable foundation for swarm solutions to be applicable for domestic and commercial cleaning. We can achieve this by two major milestones: localisation with object identification, and object grasping through coordinated formations. The first milestone is crucial as it provides a functional component relevant to domestic use cases, enabling object identification, localization, and communication, all of which are key to managing a scalable fleet. Secondly, object grasping through coordinated formations is a crucial follow-up as it expands further upon an abstract concept to pinpoint specific use cases such as tidying items in a room.

The metrics to evaluate our swarm system are scalability, flexibility, architectural tolerance, cost-effectiveness, efficiency, autonomous operations, and performance.

The project involves deploying a swarm of three robots in a room with obstacles and targets, each equipped with RGB-D cameras, LiDAR for C-SLAM, and ultrasonic sensors, all within a budget of 250,000 THB. Designed with a fixed 90-degree gripper and a two-hour battery life, the robots independently use LiDAR and depth cameras for cost-effective 3D mapping, creating a shared environmental map. Speed is not a critical factor, as the focus is on task completion. Due to the constraints of operating on a level floor, the robots are constructed from heavy materials, which aids in object manipulation. For object detection, the robots utilise RGB-D cameras, and once consensus on the target is reached, roles are dynamically assigned: Alpha and Beta secure the object, while Gamma provides the pushing force to move it to the destination using a sliding method, ensuring efficient 2D movement.

Chapter 5

Project Planning and Timeline

For project planning, we have divided the workload into four phases: two for the first semester and two for the second semester. The time allocated for each task is detailed in the attached Gantt Chart, which also specifies the number of group members assigned to each task.

During the first phase, we will focus on simulating our fundamental components: communication, object detection, and SLAM. The communication aspect will explore how robots in a swarm coordinate with one another. Object detection will be implemented using computer vision, and SLAM will have a basic implementation as a proof-of-concept. These simulations will be conducted under the assumption that robots can communicate effectively. All simulations will be conducted using Webots, a robotic simulation platform. This phase is scheduled to take one month.

In the second phase, we will begin integrating hardware by incorporating simple movement. Additionally, we will improve our SLAM implementation. Our primary milestones include testing SLAM in two different environments and implementing an enhanced version of SLAM, such as C-SLAM. These tasks will be undertaken in parallel and are expected to take one month. Once the three group members working on SLAM complete their tasks, they will attempt to combine the individual modules from the first phase into a single simulation. Achieving this integration will result in our Minimum Viable Prototype. This phase aims to be completed by the end of the first semester.

The second semester will focus on gripping and advanced movement, including robot formation. The major milestones for this phase include coordinated gripping, which will test the feasibility of a static gripper for the upcoming sliding movement, and robot formation, which will involve the robots moving together to form a cohesive unit. This phase aims to ensure seamless movement after an object has been gripped.

The final phase will integrate all the functional components developed in simulation with actual robotic hardware. This phase is scheduled to start in February and continue until the project's completion. Successfully completing this phase will achieve our advanced project goal and ensure that all components function seamlessly in the real world.

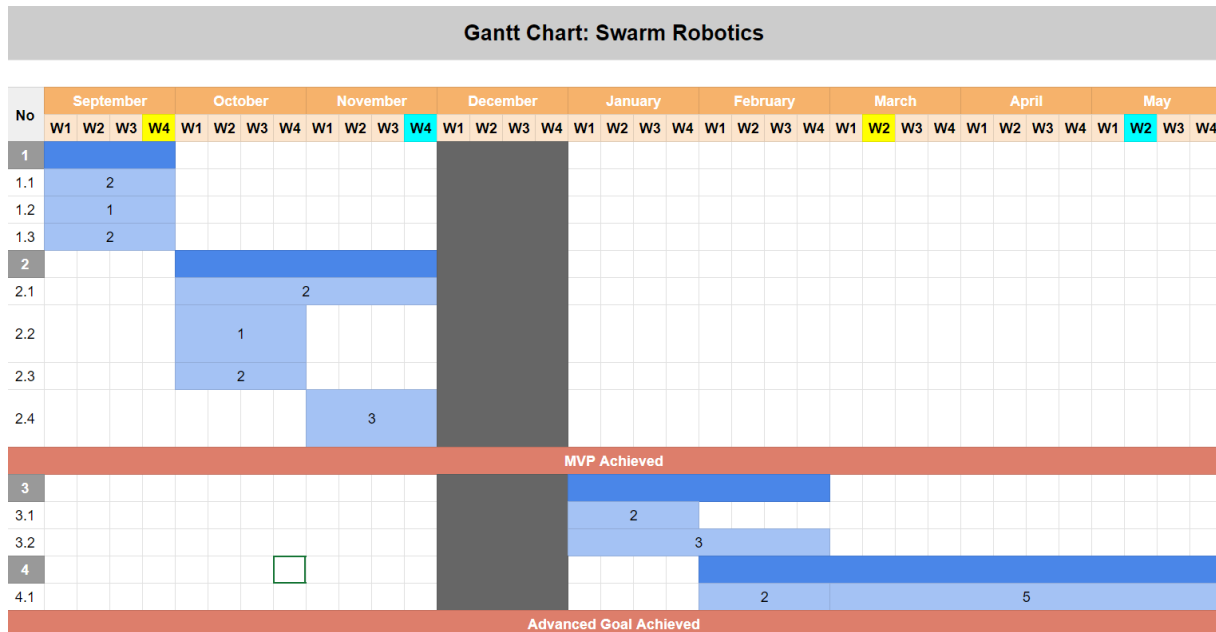


Figure 5.1 – Gantt Chart

1. Simulation
 - 1.1 Communication in the swarm
 - 1.2 Object detection using Computer Vision (CV)
 - 1.3 Simple Simultaneous Localization and Mapping (SLAM)
2. Building the Minimum Viable Prototype (MVP)
 - 2.1 Simple movement with hardware
 - 2.2 SLAM in two environments
 - 2.3 Improved SLAM
 - 2.4 Combine individual modules in simulation
3. Coordinated movement with object interactions
 - 3.1 Coordinated gripping with static grippers
 - 3.2 Assume swarm formation
4. Moving towards the advanced goal
 - 4.1 Combine everything in hardware

Chapter 6

Theory Backup

6.1 Omnidirectional Movement

In the field of mobile robots, omnidirectional wheels have an advantage of moving in 2 degrees, 3 degrees of freedom. In the x-y axis and yaw. It's a holonomic locomotion unlike non-holonomic such as ackerman or diff drive. 90deg dual row would be the option of choice. The reason why we use omnidrive is because it can move in any direction without disengaging the object. Simple mechanism due to no linkage. Need a suspension.

6.2 Waypoint Navigation

Repulsive Potential limitation is the local minima for the velocity at which the robot can travel, which can cause the robot to be stuck. This is where Rotational Fields and Random Fields come into play. By adding a rotational field around obstacles, the symmetry of the potential field is broken. As a result, the potential field will act as a guide for the robot to manoeuvre around groups of obstacles while avoiding local minima. However, this function can cause unstable oscillation during high speeds, narrow corridors, or sudden changes.

The Wavefront Planner applies the brushfire algorithm, starting from the goal and labelling the goal pixel as 2, then adding all zero neighbours. It continues iterating by updating distances for neighbouring cells and adding them to the list until all cells are processed. The final result provides a distance value for each cell, allowing the robot to follow a gradient descent by moving to the neighbour with the lowest distance value.

6.3 Swarm Robotics Theory

Swarm Robotics is a robotics field that mimics the collective behaviour of natural swarms like ants, birds, and fish. The fundamental concept behind swarm robotics is to make use of multiple robots working together and carrying out tasks that would be difficult and inefficient for one single robot to handle. The four main principles of swarm robotics are self-organisation, distributed control, local interaction, and scalability. The principles are crucial for the robots to perform complex tasks by following simple individual instructions. Robots in a swarm are usually decentralised meaning that they operate on simple rules based on the local environment. This approach

allows the swarm to be scaled easily as the number of robots increases[69].

6.4 SLAM

Simultaneous Localisation and Mapping, also known as SLAM, is a crucial aspect of swarm robotics. SLAM is what allows each robot to navigate autonomously in an environment by constructing a map of said environment. While the map is being constructed, the robot is simultaneously tracking down their own position as well[70]. In our project, we will be going a step further by using Collaborative SLAM (C-SLAM) in which multiple robots work together to build a shared map of the environment. C-SLAM builds on classic SLAM by having multiple robots dynamically sharing and updating their individual maps and position data. This novel approach significantly improves the efficiency and accuracy of map generation which could be a big advantage in dynamic environments.

6.5 Object Detection and Computer Vision (CV)

Computer Vision (CV) enables robots to recognise and classify objects within their environment. Cameras such as the RGB-D can be used for real time object detection as these types of cameras can capture both colour (RGB) and depth (D) information. This greatly enhances the ability of the robot to perceive the objects in the environment therefore becoming useful in recognizing objects in cluttered environments. In the last few years, object detection has been massively improved by machine learning with techniques such as Convolutional Neural Network becoming a successful way to recognise objects with high accuracy[71].

6.6 Multi-Robot Collaboration and Coordination

Dynamic Role assignment is going to be able to allow robots to adapt their behaviour based on the tasks at hand. This is important in tasks that require multiple and complex actions. By using the process of distributed decision-making, each robot evaluates its own capabilities and checks the status of other robots when making the decision to assume which role to take[72].

6.7 Grasping and Object Manipulation in Robotics

In robotics, grasping and object manipulation are critical challenges that often require sophisticated algorithms for planning and controlling the end-effector, such as robotic grippers. In traditional robotic systems, grasp planning involves calculating the optimal points on an object to secure it, ensuring stable lifting and transport. This becomes particularly complex in cluttered or dynamic environments, where obstacles and unexpected changes may affect the robot's ability to maintain a stable grasp. However, in swarm robotics, the collective behaviour of multiple robots can simplify this task. Instead of relying on a single robot to execute a complex grasp, multiple robots can collaborate in a more straightforward and effective manner. In our project, this is achieved using a sliding method for object manipulation, where one robot is responsible for pushing the object while the others stabilise it. This coordinated formation approach reduces the need for precise, complex grasp.

Chapter 7

Project Outcome

Assuming we are able to reach our goal by overcoming expected and unexpected challenges.

Stage 1: Collaborative Localization and Mapping Using C-SLAM

In the first stage, three robots are randomly placed within a room containing furniture and three distinct target objects. Each robot will independently scan the environment using LiDAR and Collaborative Simultaneous Localization and Mapping (C-SLAM) to determine its own position, identify the relative positions of other robots, and collaboratively create a shared map of the room. This process continues until the entire space is mapped. The use of C-SLAM is advantageous due to its decentralised nature, reducing the risk of a single point of failure and benefiting from mesh redundancy, allowing the task to continue if one robot fails. C-SLAM also enhances efficiency through lightweight communication, and though it requires more expensive sensors like LiDAR, this is offset by improved localization accuracy. Additionally, C-SLAM offers flexibility in environments with strong network connectivity and supports 3D mapping through the integration of depth cameras. This stage will be conducted at the ISE Lab.

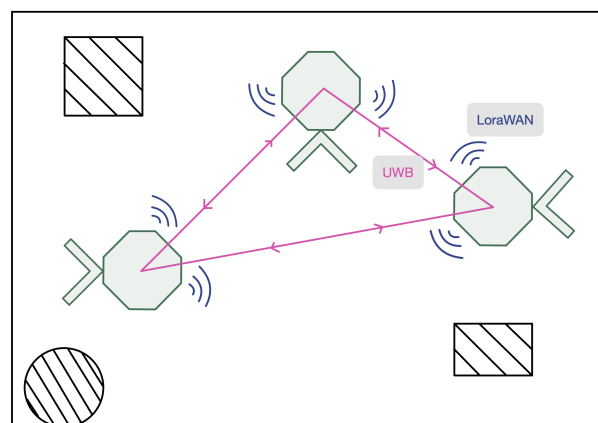


Figure 7.1 – Stage 1: 3 swarm robots(Green) localising and communicating with each other using LoraWAN and Ultra-Wide-Band UWB. the black objects are obstruction

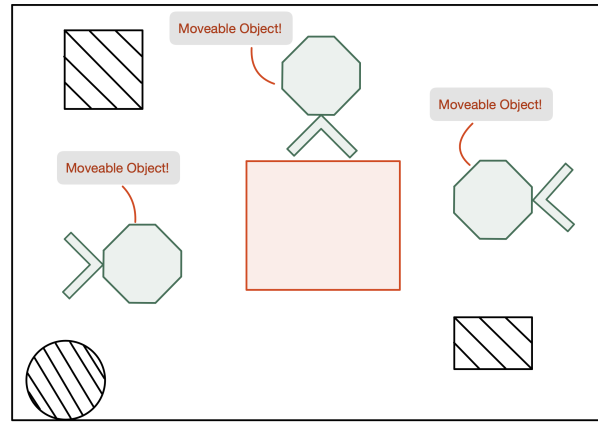


Figure 7.2 – Stage 2: A Red Object placed in the localised map, the robots detects the object using 2D/3D computer vision

Stage 2: Object Detection and Consensus for Identification

After localization is complete, the swarm enters a standby state. When a named object is selected from the model database, the robots will collaborate to search for and identify the object. Upon locating it, the robot communicates the object's position to the rest of the swarm. A consensus mechanism ensures that over 50% of the robots agree that the correct object has been identified before any action is taken. The object, for testing purposes, is a static, slid-able geometric shape, such as a cube. The robots utilise RGB-D cameras for dynamic object detection, ensuring that movable items like pets or furniture are not unintentionally displaced. The depth data from the cameras enhances the robots' 3D mapping capabilities and provides a cost-effective solution for object recognition.

Stage 3: Coordinated Object Movement

Once consensus on the target object is reached, the robots will move towards it. Two robots, named Alpha and Beta, are assigned to secure the object along the X and Y axes at diagonal corners. A third robot, Gamma, provides the necessary force to push the object along a straight line toward its destination. The assignment of these roles is dynamic, with Alpha, Beta, and Gamma selected based on proximity to the object and destination. Alpha and Beta are assigned based on proximity to the object's corners, while Gamma is the robot closest to the straight line between the object and destination. This method is both efficient and flexible, as role assignment occurs in real time, ensuring minimal delay. The destination is selected either by choosing the nearest edge of the object or by providing specific coordinates using the SLAM-generated map, allowing future integration with advanced interfaces like drag-and-drop UI.

Stage 4: Object Movement and Environmental Constraints

In the final stage, the robots push the object to the selected destination. To facilitate this, the object must have sufficiently low friction to allow sliding across the floor, as the team has chosen to restrict the object's motion to two dimensions. The sliding method minimises potential challenges related to lifting or gripping the object

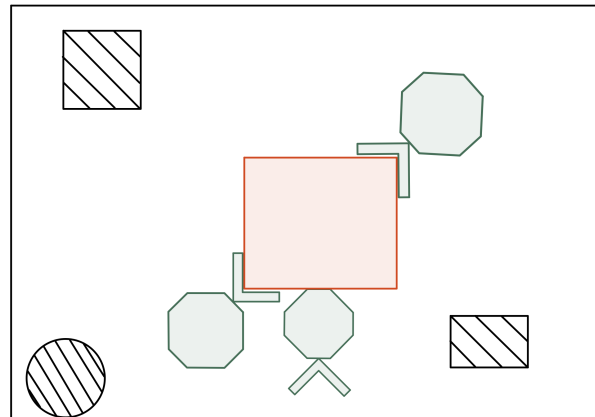


Figure 7.3 – Stage 3: The swarm moves and grip the object and ensure the object is securely in place

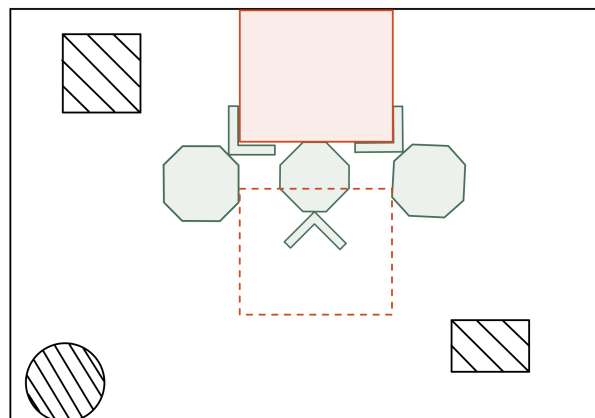


Figure 7.4 – Stage 4: The swarms then move the robot to the designated area and adjust the position of the robot dynamically

Chapter 8

Summary and Benefits to Real Industry

It's becoming increasingly inevitable that robots will take over a significant portion of the workforce in the near future, especially for repetitive and monotonous tasks. As industries like warehousing, manufacturing, and even domestic environments integrate more robots, the ability for multiple robots to work together in the same space becomes critical. This is where swarm robotics comes into play. In such systems, robots must be able to communicate effectively to avoid collisions, coordinate task allocation, and share progress, ensuring smooth and efficient workflows. This project is a step toward achieving that goal by advancing decentralised, collective robotic behaviour.

A key feature of this project is the decentralised communication system, where robots can interact with one another without needing a host computer or central control. This makes the system incredibly flexible and easy to deploy in various settings—whether in the field or at fixed locations like warehouses or homes. The decentralised nature allows for a plug-and-play style of operation, eliminating the need for complex initial setups. The mesh communication system also provides resilience, as the robots can continue working even if one or more units experience failure, ensuring task completion. This makes the system well-suited not just for static environments but also for dynamic, outdoor applications where flexibility is key.

The benefits extend across various industries, from cleaning and tidying in commercial or domestic spaces (homes, offices, retail) to more complex environments like schools or factories where multiple robots operate together. The technology's adaptability and resilience make it applicable in any scenario where more than one or two robots are required to collaborate, making it a valuable solution for the future of automation.

Chapter 9

Project Contribution per Student

In the first phase, the project tasks are divided among the group members to ensure a focused effort on simulating the fundamental components. Two group members will concentrate on developing the communication aspect, ensuring that the robots can coordinate and communicate effectively within the swarm. Simultaneously, one member will focus on object detection using computer vision techniques, while another two members tackle the basic implementation of SLAM (Simultaneous Localization and Mapping). This phase is scheduled to span from October to early November.

Moving into the second phase, which runs from early November to early December, the project transitions to integrating hardware and enhancing SLAM functionalities. Initially, all group members will collaborate on incorporating simple movement into the hardware, ensuring a robust foundation for the next tasks. One group member will then test the SLAM implementation in two different environments to verify its effectiveness. Concurrently, two members will work on improving SLAM by developing a more advanced version, such as C-SLAM. After completing these tasks, the dedicated SLAM team, composed of three members, will integrate the individual modules from the first phase into a cohesive simulation, aiming to achieve a Minimum Viable Prototype by the semester's end.

In the third phase, from January to early March of the second semester, the focus shifts to more advanced capabilities, including gripping mechanisms and coordinated movement. Two group members will work on the coordinated gripping task, preparing the static gripper for future sliding movements. Meanwhile, another three members will develop strategies for robot formation, ensuring that the robots can move together seamlessly after gripping objects.

Finally, the fourth phase, starting in March and extending to early May, involves putting all the functional simulation components together with the actual robotic hardware. During this period, all group members will collaborate intensively to integrate these components, ensuring they work together smoothly in the real world.

Bibliography

- [1] A. R. Cheraghi, S. Shahzad, and K. Graffi, *Past, Present, and Future of Swarm Robotics*, 2021, pp. 190–233.
- [2] L. L. Bayindir and E. Şahin, “A review of studies in swarm robotics,” *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 15, no. 2, pp. 115–126, 2007. [Online]. Available: <https://journals.tubitak.gov.tr/elektrik/vol15/iss2/2>
- [3] N. Nedjah and L. J. Silva, “Review of methodologies and tasks in swarm robotics towards standardization,” *Swarm and Evolutionary Computation*, vol. 50, p. 100565, 2019.
- [4] E. Şahin, “Swarm robotics: From sources of inspiration to domains of application,” in *Lecture Notes in Computer Science*, 2005, pp. 10–20. [Online]. Available: https://doi.org/10.1007/978-3-540-30552-1_2
- [5] Y. Koifman, A. Barel, and A. M. Bruckstein, “Distributed and decentralized control and task allocation for flexible swarms,” *arXiv*, 2024.
- [6] M. Kegeleirs, G. Grisetti, and M. Birattari, “Swarm slam: Challenges and perspectives,” *Frontiers in Robotics and AI*, vol. 8, 2021.
- [7] L. Türkler, T. Akkan, and L. Akkan, “Usage of evolutionary algorithms in swarm robotics and design problems,” *Sensors*, vol. 22, no. 12, p. 4437, 2022.
- [8] S. Das, *Bio-Inspired Communication Strategies in Swarm Robotics*, 2024, pp. 77–100.
- [9] R. Ibrahim, M. Alkilabi, A. R. H. Khayeat, and E. Tuci, “Enhancing robustness of swarm robotics systems in a perceptual discrimination task,” *International Journal of Computing and Digital Systems*, vol. 15, no. 1, pp. 1213–1222, 2024.
- [10] A. Ayari and S. Bouamama, “Evolutionary swarm robotics: A methodological approach for task and path planning,” in *2023 IEEE Afro-Mediterranean Conference on Artificial Intelligence (AMCAI)*, 2023.
- [11] S. M. Perera, R. J. Myers, K. Sullivan, K. Byassee, H. Song, and A. Madanayake, “Integrating communication and sensor arrays to model and navigate autonomous unmanned aerial systems,” *Electronics*, vol. 11, no. 19, p. 3023, 2022.
- [12] S. S. R., S. Mohanty, and D. S. Elias, “Control and coordination of a swarm of unmanned surface vehicles using deep reinforcement learning in ros,” *arXiv (Cornell University)*, 2023.
- [13] Z. Wang, J. Li, J. Li, and C. Liu, “A decentralized decision-making algorithm of uav swarm with information fusion strategy,” *Expert Systems With Applications*, vol. 237, p. 121444, 2024.

- [14] M. Yasser, O. Shalash, and O. Ismail, "Optimized decentralized swarm communication algorithms for efficient task allocation and power consumption in swarm robotics," *Robotics*, vol. 13, no. 5, p. 66, 2024.
- [15] N. Nedjah, L. M. Ribeiro, and L. De Macedo Mourelle, "Communication optimization for efficient dynamic task allocation in swarm robotics," *Applied Soft Computing*, vol. 105, p. 107297, 2021.
- [16] E. Beck, B. Shin, S. Wang, T. Wiedemann, D. Shutin, and A. Dekorsy, "Swarm exploration and communications: A first step towards mutually-aware integration by probabilistic learning," *Electronics*, vol. 12, no. 8, p. 1908, 2023.
- [17] S. Zhang, E. Staudinger, R. Pohlmann, and A. Dammann, "Cooperative communication, localization, sensing and control for autonomous robotic networks," in *2021 IEEE International Conference on Autonomous Systems (ICAS)*, 2021.
- [18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.
- [19] V. Kumar, N. Michael, and J. P. How, "Multi-robot coordination in warehousing," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 232–239.
- [20] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2014, pp. 580–587.
- [21] W. Chen, Y. Li, Z. Tian, and F. Zhang, "2d and 3d object detection algorithms from images: A survey," *Array*, vol. 19, p. 100305, 2023.
- [22] B. Karin, A. Gonzalez, and X. Zhou, "A comparative analysis of stereo camera systems for depth sensing in robotics," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 1234–1241.
- [23] S. Wang, L. Zhang, Z. Zhang, and S. Zhang, "Radar and camera fusion for robust object detection and tracking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 5, pp. 2035–2047, 2020.
- [24] K. A. Tychola, I. Tsimperidis, and G. Papakostas, "On 3d reconstruction using rgb-d cameras," *Digital*, vol. 2, no. 3, pp. 401–423, 2022.
- [25] N. V. K. Medathati, H. Neumann, G. S. Masson, and P. Kornprobst, "Bio-inspired computer vision: Towards a synergistic approach of artificial and biological vision," Inria Sophia Antipolis, Tech. Rep. 8698, 2016, fihal-01131645v3f.
- [26] Y. Li, J. Su, L. Liu, and P. Liu, "Object detection based on the fusion of sparse lidar point cloud and dense stereo pseudo point cloud," in *IEEE*, January 19-21 2024.
- [27] T. Eppenberger, G. Cesari, M. Dymczyk, and R. Dube, "Leveraging stereo-camera data for real-time dynamic obstacle detection and tracking," in *Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.
- [28] J. Rodriguez, "A comparison of an rgb-d camera's performance and a stereo camera in relation to object recognition and spatial position determination," *ELCVIA Electronic Letters on Computer Vision and Image Analysis*, vol. 20, no. 1, p. 16, 2021.

- [29] Y. Wang, C. Wang, P. Long, Y. Gu, and W. Li, "Recent advances in 3d object detection based on rgb-d: A survey," *Displays*, vol. 70, p. 102077, 2021.
- [30] E. Arican and T. Aydin, "Object detection with rgb-d data using depth oriented gradients," in *Proceedings of the International Conference on Engineering and Natural Sciences (ICENS)*, Hungary - Budapest, 2017.
- [31] A. Paigwar, D. Sierra-Gonzalez, Erkent, and C. Laugier, "Frustum-pointpillars: A multi-stage approach for 3d object detection using rgb camera and lidar," in *Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*. IEEE, 2021, pp. 2926–2933.
- [32] C. Tao, H. Zhang, J. Liu, and J. Li, "F-pvnet: Frustum-level 3-d object detection on point-voxel feature representation for autonomous driving," *IEEE Internet of Things Journal*, vol. 10, no. 9, pp. 8031–8045, 2023.
- [33] Z. Ren and E. B. Sudderth, "3d object detection with latent support surfaces," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2018)*. Salt Lake City, UT, United States: IEEE Computer Society, 2018, pp. 104–112.
- [34] Z. Zhang *et al.*, "H3dnet: 3d object detection using hybrid geometric primitives," in *European Conference on Computer Vision*. Springer, 2020.
- [35] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. San Diego, CA, USA: IEEE Computer Society, 2005, pp. 886–893, Vol. 1.
- [36] P. Viola and M. Jones, "Robust real-time face detection," in *Proceedings of the Eighth IEEE International Conference on Computer Vision (ICCV 2001)*. Vancouver, BC, Canada: IEEE Computer Society, 2001, p. 747.
- [37] X. Zhou and G. Lin, "Review of target detection algorithms," *Frontiers in Computing and Intelligent Systems*, vol. 4, no. 3, pp. 17–19, 2023.
- [38] B. Karbouj, "Comparative performance evaluation of one-stage and two-stage object detectors for screw head detection and classification in disassembly processes," *Procedia CIRP*, vol. 122, 2024, conference: 31st CIRP Conference on Life Cycle Engineering (LCE 2024), Turin. License: CC BY-NC-ND 4.0. Lab: Bsher Karbouj's Lab.
- [39] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision – ECCV 2016. Lecture Notes in Computer Science*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Springer, Cham, 2016, vol. 9905, pp. 21–37.
- [40] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 936–944.
- [41] R. Girshick, "Fast r-cnn," in *2015 IEEE International Conference on Computer Vision*. IEEE, 2015, pp. 1440–1448.
- [42] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2015.

- [43] V. K. Kaliappan, M. S. Shanmugasundaram, L. Ravikumar, and G. B. Hiremath, "Performance analysis of yolov8, rcnn, and ssd object detection models for precision poultry farming management," in *2023 IEEE 3rd International Conference on Applied Electromagnetics, Signal Processing, Communication (AESPC)*. Bhubaneswar, India: IEEE, 2023, pp. 1–6.
- [44] V. K. Kaliappan, R. Thangaraj, P. Pandiyan, K. Mohanasundaram, S. Anandamurugan, and D. Min, "Real-time face mask position recognition system using yolo models for preventing covid-19 disease spread in public places," *International Journal of Adaptive and Autonomous Systems*, vol. 34, no. 1, pp. 73–82, 2023.
- [45] A. Barbadekar, S. Raut, R. Gaikwad, T. Gadad, S. Ghulaxe, and N. Dhabale, "Exploring enhanced localization techniques: Lidar-slam for mobile robots," in *2023 Global Conference on Internet of Things and Cyber-Physical Systems (GCITC)*, 2023.
- [46] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part i," *IEEE Robotics and Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.
- [47] T. Chong, X. Tang, C. Leng, M. Yogeswaran, O. Ng, and Y. Chong, "Sensor technologies and simultaneous localization and mapping (slam)," *Procedia Computer Science*, vol. 76, pp. 174–179, 2015.
- [48] B. Udugama, "Evolution of slam: Toward the robust-perception of autonomy," in *arXiv preprint*, 2023, available at: <https://doi.org/10.48550/arXiv.2302.06365>. [Online]. Available: <https://doi.org/10.48550/arXiv.2302.06365>
- [49] A. Sahoo, S. K. Dwivedy, and P. Robi, "Advancements in the field of autonomous underwater vehicles," in *Ocean Engineering*, vol. 181. Elsevier, 2019, pp. 145–160. [Online]. Available: <https://doi.org/10.1016/j.oceaneng.2019.04.011>
- [50] Y. Bisheng, L. Fuxun, and H. Ronggang, "Progress, challenges, and perspectives of 3d lidar point cloud processing," *DOAJ (Directory of Open Access Journals)*, 2017.
- [51] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.
- [52] B. Langmann, K. Hartmann, and O. Loffeld, "Depth camera technology comparison and performance evaluation," in *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP)*. Barcelona, Spain: SciTePress, 2012, pp. 438–444. [Online]. Available: <https://doi.org/10.5220/0003778304380444>
- [53] C. Peng, "Depth camera point cloud sharpening," in *Proceedings of the IEEE International Conference on Consumer Electronics (ICCE)*. Taipei, Taiwan: IEEE, 2023, pp. 102–106.
- [54] K. Huang, S. Zhang, J. Zhang, and D. Tao, "Event-based simultaneous localization and mapping: A comprehensive survey," *arXiv*, 2023.
- [55] R. Benkis, E. Grabs, T. Chen, A. Ratkuns, D. Čulkovs, A. Ancāns, and A. Ipatovs, "A survey and practical application of slam algorithms," in *2024 Progress In Electromagnetics Research Symposium (PIERS)*, 2024, pp. 1–10.

- [56] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: A versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.
- [57] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 611–625, 2018.
- [58] X. Gu, X. Wang, and Y. Guo, "A review of research on point cloud registration methods," *IOP Conference Series Materials Science and Engineering*, vol. 782, no. 2, p. 022070, 2020.
- [59] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time," in *Proceedings of Robotics: Science and Systems (RSS)*, 2014. [Online]. Available: <https://doi.org/10.15607/rss.2014.x.007>
- [60] W. Xu and F. Zhang, "Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3317–3324, 2021.
- [61] W. Chen, X. Wang, S. Gao, G. Shang, C. Zhou, Z. Li, C. Xu, and K. Hu, "Overview of multi-robot collaborative slam from the perspective of data fusion," *Machines*, vol. 11, no. 6, p. 653, 2023.
- [62] X. Liu, J. Lei, A. Prabhu, Y. Tao, I. Spasojevic, P. Chaudhari, N. Atanasov, and V. Kumar, "Slideslam: Sparse, lightweight, decentralized metric-semantic slam for multi-robot navigation," *arXiv*, 2024. [Online]. Available: <https://doi.org/10.48550/arxiv.2406.17249>
- [63] P. Lajoie and G. Beltrame, "Swarm-slam: Sparse decentralized collaborative simultaneous localization and mapping framework for multi-robot systems," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 475–482, 2024.
- [64] A. Mehta and A. Modi, "Robust sliding mode protocols for formation of quadcopter swarm," 2024.
- [65] F. Martinez and A. Rendon, "A swarm-based flocking control algorithm for exploration and coverage of unknown environments," 2023.
- [66] E. Bonabeau, A. Sobkowski, G. Theraulaz, and J. L. Deneubourg, "Adaptive task allocation inspired by a model of division of labour in social insects," in *Biocomputing and Emergent Computation: Proceedings of BCEC97*. London, UK: World Scientific, 1997, pp. 36–45.
- [67] M. Brambilla, C. Pinciroli, M. Birattari, and M. Dorigo, "Property-driven design for swarm robotics," in *Proceedings of 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. Richland: IFAAMAS, 2012, pp. 139–146.
- [68] D. St-Onge, J. Le Ny, and G. Beltrame, "Swarm-slam: Sparse decentralised collaborative simultaneous localization and mapping framework for multi-robot systems," *IEEE Robotics and Automation Letters*, vol. 8, no. 3, pp. 1720–1727, 2023.
- [69] G. Beni and J. Wang, "Swarm intelligence in cellular robotic systems," in *Proceedings of NATO Advanced Workshop on Robots and Biological Systems*, 1989.
- [70] S. Thrun, "A probabilistic approach to concurrent mapping and localization for mobile robots," *Autonomous Robots*, vol. 5, no. 3, pp. 253–271, 2003.

- [71] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.
- [72] L. E. Parker, "Alliance: An architecture for fault-tolerant multirobot cooperation," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 2, pp. 220–240, 1998.