

**MULTI-AGENT REINFORCEMENT LEARNING
FOR SWARM ROBOTS FORMATION**

CHRISTINA ANAK BUJANG

**SCHOOL OF AEROSPACE ENGINEERING
UNIVERSITI SAINS MALAYSIA**

2021

**MULTI-AGENT REINFORCEMENT LEARNING FOR SWARM ROBOTS
FORMATION**

by

CHRISTINA ANAK BUJANG

**A thesis submitted in fulfillment of the requirements for the Bachelor Degree of
Engineering (Honours) (Aerospace Engineering)**

June 2021

ENDORSEMENT

I, Christina Anak Bujang, hereby declare that I have checked and revised the whole draft of the dissertation as required by my supervisor.

Christina

(Signature of Student)

Date: 8 July 2021

Ye Zhou

(Signature of Supervisor)

Name: Zhou Ye

Date: 8 July 2021

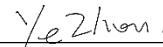
ENDORSEMENT

I, Christina Anak Bujang, hereby declare that all corrections and comments made by the supervisor and examiner have been taken into consideration and rectified accordingly.



(Signature of Student)

Date: 8 July 2021



(Signature of Supervisor)

Name: Zhou Ye

Date: 8 July 2021



(Signature of Examiner)

Name:

Dr. Dr. Ahmad Fadzul Hawary
Senior Lecturer
School of Aerospace Engineering
Engineering Campus, Universiti Sains

Date: 8 July 2021

DECLARATION

This thesis is the result of my investigation, except where otherwise stated and has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any other degree.

Zyza

(Signature of Student)

Date: 8 July 2021

ACKNOWLEDGEMENT

First of all, I would like to express my sincere gratitude and appreciation to my supervisor, Dr. Zhou Ye. She provided an open and free environment that allowed me to choose this topic and provided valuable insight that helped me in completing this final year final project. Without her, this project might not go as well as it is. Thus, I could never summon my courage to enter the world of robotics. The encouragement and motivation that was given during this project are under construction are greatly appreciated. I would also like to acknowledge her exemplary guidance, monitoring, and constant encouragement throughout this thesis.

I would also like to thank my friends and peers, especially Nurfarah Anisah and Nurul Asyikin for their help at the beginning of my stay also for supporting me on my robotics experiment and research initiatives. I would like to express my to gratitude for Ahmad Muaz and Muhammad Noor Sabri for their help in discovering the field of robotics and Reinforcement Learning (RL).

Last but not least, a lot of thanking my family that always give me support, love, and understanding which encourage me to complete this project. I am also happy to present my grateful acknowledgment to anybody who helped me either directly or indirectly in writing this thesis.

MULTI-AGENT REINFORCEMENT LEARNING FOR SWARM ROBOTS FORMATION

ABSTRACT

The project discussed the Multi-Agent Reinforcement Learning (MARL) with an idea to the proposed mobile robot which able to follow the line and avoid the obstacle in a given environment. The reinforcement learning algorithm offers one of the most general frameworks in learning subjects to address some of the control issues in a multi-agent system. The mobile robot is an independent agent that can use sensors, actuators, and control techniques to navigate intelligently based on the specific task required. Specifically, reinforcement learning is employed for developing the training process for the mobile robot to reach the given task as it needs to learn by itself to follow the black line and avoid the obstacle in a given environment based on this project proposed. The reinforcement learning approach presents the algorithm for MARL in a cooperative problem to improve control performance. Experimental and simulation will be carried out to validate the results of the multi-agent control performance. Hence, it should be easy to observe if the control performance shows improvement after learning and can achieve the project proposed. The experiment will therefore indicate the results of the simulation and apply it to the real-time environment as proposed by the project.

PEMBELAJARAN PENGUKUHAN PELBAGAI EJEN UNTUK PEMBENTUKAN ROBOT SWARM

ABSTRAK

Projek ini membincangkan Pembelajaran Pengukuhan Pelbagai Ejen (MARL) dengan idea kepada robot mudah alih yang dicadangkan yang dapat mengikuti garisan dan mengelakkan halangan dalam persekitaran tertentu. Algoritma pembelajaran tetulang menawarkan salah satu rangka kerja yang paling umum dalam subjek pembelajaran untuk menangani beberapa isu kawalan dalam sistem pelbagai ejen. Robot mudah alih adalah ejen bebas yang boleh menggunakan sensor, pengaktif, dan teknik kawalan untuk menavigasi dengan bijak berdasarkan tugas tertentu yang diperlukan. Khususnya, pembelajaran tetulang digunakan untuk membangunkan proses latihan untuk robot mudah alih untuk mencapai tugas yang diberikan kerana ia perlu belajar dengan sendirinya untuk mengikuti garis hitam dan mengelakkan halangan dalam persekitaran tertentu berdasarkan projek ini dicadangkan. Pendekatan pembelajaran tetulang membentangkan algoritma untuk MARL dalam masalah koperasi untuk meningkatkan prestasi kawalan. Eksperimen dan simulasi akan dijalankan untuk mengesahkan keputusan prestasi kawalan pelbagai ejen. Justeru, perlu mudah untuk dipatuhi sekiranya prestasi kawalan menunjukkan peningkatan selepas pembelajaran dan dapat mencapai projek yang dicadangkan. Oleh itu, eksperimen ini akan menunjukkan hasil simulasi dan menggunakannya kepada persekitaran masa nyata seperti yang dicadangkan oleh projek.

TABLE OF CONTENTS

ENDORSEMENT	I
ENDORSEMENT	II
DECLARATION	III
ACKNOWLEDGEMENT	IV
ABSTRACT	V
CHAPTER 1 : INTRODUCTION	1
1.1 Context	1
1.2 Reinforcement Learning	1
1.3 Multi-Agent Reinforcement Learning	2
1.3 Problem Statement	4
1.4 Objectives	5
1.5 Thesis Outline	6
CHAPTER 2: LITERATURE REVIEW	7
2.1 Reinforcement Learning	7
2.2 Reinforcement Learning Algorithm Concept Structure	8
2.2.1 Mathematical Foundation	8
2.2.2 Estimating The Optimal Policy	12
2.3 Swarm Robotic Behaviors	13
2.3.1 Spatially Organizing Behaviours	14
2.3.2 Navigational Behaviour	15
2.3.3 Collective Decision Making	15
2.4 Challenge in Multi-Agent Reinforcement Learning (MARL)	16
2.4.1 The Curse of Spatiality	16
2.4.1 The Exploration-Exploitation Trade-Off	16

2.4.3 Reward Shaping	17
CHAPTER 3: METHODOLOGY	18
3.1 Flow Chart	18
3.1.1 Line Follower Flow Chart	18
3.1.2 Obstacle Avoidance Flow Chart	20
3.2 Reinforcement Learning Based Approach	21
3.2.1 Formulate Learning Task	22
3.2.2 Adding Perception	23
3.2.3 Environment Design and Modelling	24
3.3 Kinematic Motion and Characteristics of The Mobile Robot Model	29
3.4 Line Follower Mobile Robot Working Concept	31
3.5 Hardware Component Selection	35
3.5.1 Arduino	35
3.5.2 Motor Driver	36
3.5.3 Sensors	37
3.5.4 Car Chasis	40
3.5.5 Jumper Wire	41
3.5.6 Batteries	42
3.5 Software Implementation (Programming The Robot)	44
CHAPTER 4: RESULTS AND DISCUSSION	46
4.1 Simulation Environment of RL Multi-Agent Swarm Robot	46
4.1.1 Simulation of Map Environment	47
4.1.2 Simulation of the leader-follower formation	47
4.1.3 Simulation of multi-agent leader-followers formation control	50
4.2 Real-Time Environment of RL Multi-Agent Swarm Robot	51
4.2.1 Hardware Setup	51
4.2.2 Experiment Conducted	52

4.3 The Problem and Issues	59
CHAPTER 5: CONCLUSION AND SUGGESTIONS	62
5.1 Conclusion	62
5.2 Future Work	63
REFERENCES	66

LIST OF FIGURE

Figure 1.1: Reinforcement learning decision-making diagram for robot control	4
Figure 2.1: Agent-environment interface	9
Figure 3.1: Line follower flow chart	19
Figure 3.2: Obstacles avoidance flow chart	20
Figure 3.3: The ideal behavior of the cumulative reward	25
Figure 3.4: The ideal behavior of the length of the episodes	26
Figure 3.5: Environment setting	27
Figure 3.6: The whole process for the experimental conduct	28
Figure 3.7: Kinematics model of the mobile robot	29
Figure 3.8: Line follower mobile robot working concept	34
Figure 3.9: Arduino uno	36
Figure 3.10: Motor driver module l298n	37
Figure 3.11: Infrared proximity sensor	38
Figure 3.12: Ultrasonic sensor	39
Figure 3.13: Bluetooth module hc-05	40
Figure 3.14: Car chassis	41
Figure 3.15: Jumper wires	41
Figure 3.16: Batteries	42
Figure 3.17: The assembly schematic of the line follower and obstacle avoidance mobile robot	43
Figure 4.1: Simulation of the given environment	47
Figure 4.2: Leader-follower simulation behavior without the present of obstacles	48
Figure 4.3: Leader-follower simulation behavior shape with the present of obstacles	49

Figure 4.4: The simulation of multi-agent mobile robot in the environment setup	50
Figure 4.5: The mobile robot final design	52
Figure 4.6: Experiment setup	53
Figure 4.7: A-line following task given to the one agent of mobile robot	53
Figure 4.8: The codes adjusting to calibrate the movement of the mobile robot	55
Figure 4.9: An obstacle avoidance task given to one agent of a mobile robot.	55
Figure 4.10: The test for a leader-follower configuration to transmitting and receiving the data.	57
Figure 4.11: The mobile robot is controlled by using a mobile phone via a bluetooth module sensor.	58
Figure 4.12: The experiment of the multi-agent mobile robot is conducted to perform the given task.	59

LIST OF TABLES

Table 3.1: Car chasis components for each mobile robot	40
Table 3.2: The quantity of component used to conduct experiment	43
Table 4.1: The movement of the robot motor direction	54
Table 4.2: The distance from the sensors and the obstacles selection	56

CHAPTER 1

INTRODUCTION

This chapter briefly discussed an introduction to multi-agent reinforcement learning for swarm robots. Particularly, the basic principle and benefit behind the idea of the swarm robot will be introduced. Then, the concept and characteristics, the objectives, and the arguments exposed in this thesis will be briefly presented. Finally, the outline of the thesis will be given.

1.1 Context

The context of this thesis is the field of swarm robot systems with an approach of reinforcement learning. This field gives an exciting basic platform for researchers to get involved and improved with the new ideas to scrutinize their minds in analytical and heuristic approaches. The idea of creating groups of mobile robots that able to collaborate and perform the predefined task given is starting in the early 1980s. The basic principles behind this new approach to robotics cooperation, coordination, and other interactions among themselves were directly inspired by the observation of natural systems.

1.2 Reinforcement Learning

Reinforcement learning (RL) is a machine learning technique where an agent can learn after making mistakes by interacting with the environment and gain experiences through trial and error. Moreover, the RL objective is not to cluster or mark the data, but to find the best sequence of activities that will deliver the optimal long-term result. By allowing agents to explore, communicate with and learn from the world, RL will solve

the problem easily. Thus, the more agent interacts with, the easier the problem can be solved. Besides, as an example to describe the process of RL, assumes that the world can be described by a set of state S , and that agent can take one from a finite number of actions A . The time is divided into discrete steps, and for each step, the agent observes the state of the world, S_t , and chooses an action. Finally, after taking the action, the reward function (RF) gives a reward to the agent.

Particularly, RL techniques have recently become common for solving multi-agent communication issues (Cui, Liu and Nallanathan, 2020). In RL, tasks are defined indirectly via a cost function, which is generally simpler than directly defining a task model or finding an algorithm for the controller (Hüttenrauch, Šošić and Neumann, 2018). As the cost function is defined, the objective of the RL algorithm is to find a strategy that minimizes the expected cost (Cao *et al.*, no date). However, the implementation of reinforcement learning within the swarm setting is difficult due to the large number of agents that need to be considered (Lillicrap *et al.*, 2016). Compared to single-agent learning, where an agent is faced only with observations of its state (Panait and Luke, no date), each agent in a swarm will make observations of many other agents populating the environment and thus need to process a whole collection of information that might theoretically differ in size.

1.3 Multi-Agent Reinforcement Learning

The MARL field is expanding rapidly, and over the last few years, a wide range of methods have been proposed to harness its benefits and overcome its challenges. These methods combine advances in single-agent RL, game theory, and more broadly, direct policy search techniques. At each phase, the agent is rewarded or punished by interacting

with the environment (Prateek Bajaj, no date). The policy is increasingly refined by experimentation and preparation. Due to the rapid growth of deep learning in recent years, RL has acquired the ability to solve higher-dimensional and higher-difficulty problems (Sajad Mousavi, Schukat and Howley, 2018).

However, applying reinforcement learning in the swarm environment is challenging due to the large number of agents that need to be considered to have already been mentioned (Hüttenrauch, Šošić and Neumann, 2018). Consequently, two key problems can be established in the swarm setting: (1) high status and observation dimensionality, induced by large device sizes and (2) changing the size of the available information collection, either due to addition or removal of agents, or because the number of neighbors observed changes over time (Sajad Mousavi, Schukat and Howley, 2018). Most of the existing multi-agent reinforcement learning methods either concatenate the information obtained from different agents or encode it in a multi-channel image where the image channels contain different features depending on the agent's local view (Hüttenrauch, Šošić and Neumann, 2018). Both forms of techniques, however, have had significant disadvantages.

Instead of finding the solution to the problem, RL measures the performance of the robot. RL assumes that the robot motion describes by a set of states, S and the RL multi-agent can take one of the fixed actions, A . According to the Bellman principles of optimality, being in a state taking an action with the maximum optimal value will lead to the optimal policy that maximizing the return reward, R (Sniedovich, 1978). The RL agent can reach the state optimal value through the sequence of numerical updates based on the interaction. Accordingly, the agent updates the adopted control policy by taking the action that transfers the robot to state the maximum optimal value. The decision-making of RL control is shown in Figure 1.1 (Liu *et al.*, 2021).

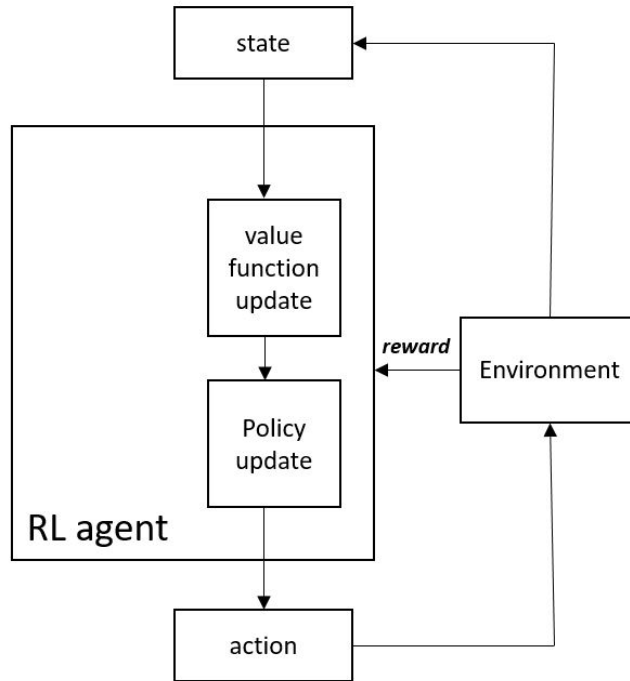


Figure 1.1: Reinforcement Learning Decision-Making Diagram For Robot Control

1.3 Problem Statement

The issues are to conduct the real-time control for the mobile robot by using software such as MATLAB and Arduino IDE. This real-time control application for a mobile robot is to ensure that the multi-agent robot follows the line and avoids obstacles. The track for the line to experiment can be straight or curved while the obstacle will be randomly placed. Since there will be more than one robot involved in this project, the robot can be placed randomly in the experiment track at a setting environment or can be set at one constant place.

The most important task for the project is to train the mobile robot a lot so that the robot can improve themselves by correcting their own mistake. The training task needs to conduct several times to make sure that the mobile robots can keep on moving along the line also can avoid any obstacle in front of them. The possible solution to train the robot is by leveraging reinforcement learning. Reinforcement learning is particularly

well-situated for using on the mobile robot. It is a machine learning technique where an agent is trying to find an optimal control policy for a certain task by continuously interacting with the environment and improving the policy with the gained experience.

1.4 Objectives

This project contributed to the investigation for multi-agent swarm robot formation control in a predetermined environment. This project consists of two main objectives such as:

1. To study the algorithm of the multi-agent system that moving to the target point with and without the existing of the obstacles by applying the reinforcement learning.
2. To investigate the behaviour of multi-agent mobile robot to detect the line and avoiding the obstacles in the given environment.

1.5 Thesis Outline

This thesis is based on the main steps taken for the real-time control system in a multi-agent mobile robot. The outline of the thesis is structure as follows:

Chapter 2 will contain two literature reviews. The review is an analysis of primary collective behaviors in swarm robotic followed by the review of successful mobile robots. **Chapter 3** will describe the approaches employed and the methodology of the whole process of this project. The design and implementation of the simulation and real-life environment along with the reinforcement learning algorithm will be discussed in detail in this chapter. Next, **chapter 4** will contain the experimental results from the implementation of the RL algorithm on a mobile robot using software for real-life environment trials. The corresponding experimental results will be presented with a general discussion at the end. Finally, **chapter 5** is the conclusion for the whole project along with the suggestions, and any possible future works are drawn.

CHAPTER 2

LITERATURE REVIEW

In this chapter, the first review will discuss the RL algorithm concept. It is the most important part of the RL framework. The mathematical model of the general RL problem is explained, and the main elements in the RL framework, which are Markov Decision and Q-learning, are discussed afterward. The swarm robotic behaviors will also be discussed in detail. It is the primary collective behavior used by swarm robots. Next, the kinematics model of the mobile robot project will be also briefly introduced in this section. The literature review is not exhaustive for all available projects, but it shows the most commonly used swarm robot platforms.

2.1 Reinforcement Learning

The use of Q-learning is normally for navigation and obstacle avoidance. The sparse reward strategy such that the agent receives the signal while reaching the target. For example, when the reaching target is 1, the hitting target would be -1 and otherwise 0. The robot is trained several times and the learned policy has evaluated by exposing the robot to the same environment with a static goal which is to follow the line and avoid the obstacle. The robot was able to complete the task by using the RL approach for mobile robot navigation. However, the time taken to complete the task was relatively long.

There are several ways of the reward function to solve the navigation problem in the autonomous robot. The following is the ways that can be taken to overcome the problem (Hutabarat *et al.*, 2020):

Binary reward

This strategy is commonly used in reinforcement learning problems that do not include dynamical behavior. This method makes the convergence rate very slow in highly-dimensional state spaces where the probability of finding the goal is low.

Sparse reward

This approach is to shape the reward function which has been used very intensively. In addition, this reward is similar to binary reward. As the negative reward is hitting the obstacles then otherwise is zero.

Potential-based reward

The purpose of the rewards is to find the transformation to the sparse reward function that gains pieces of knowledge about the surrounding in the design of the reward function. This potential-based reward is determined based on the distance between the goal and the obstacles.

2.2 Reinforcement Learning Algorithm Concept Structure

2.2.1 Mathematical Foundation

RL drawback may be model as a style of mathematically perfect for such drawback. Hence, a certain theoretical statement may be created. The mathematical object takes place to captures the set of states, the character of the transitions between the states, and therefore the rewards related to such transition.

Markov Decision Process

A Markov Decision Process (MDP) may be a discrete-time stochastic control process. It provides a mathematical framework for modeling decision-making wherever true depends partly on the taken decision (Khan, 2019).

In MDP, it consists of an associate degree Agent-Environment interface that is that the typical reinforcement learning cycle. Agent-Environment interface is wherever the (Hutabarat *et al.*, 2020):

- Agent: it is a package program that learns and makes intelligent selections. This agent can act with the setting by actions and receives the reward supported by the action taken.
- Environment: it is the simulated or real-world setting that interact with the agents.

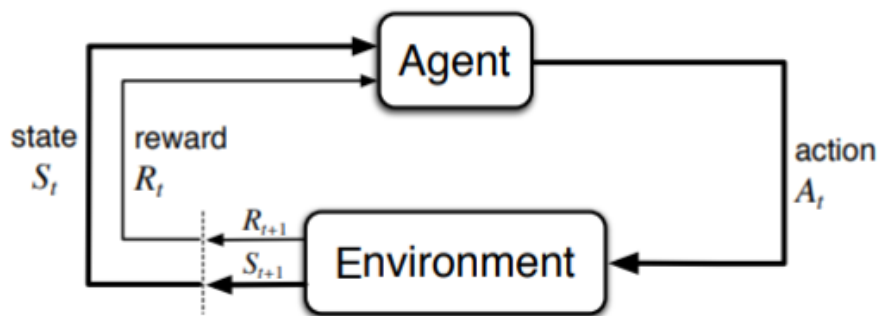


Figure 2.1: Agent-Environment Interface

The interaction between the agent and its setting is outlined a lot specifically. At any time step, the agent receives an illustration of its setting state, $S_t \in S$. To supported that state, the agent selects associate action $A_t \in A(S_t)$, where $A(S_t)$ is the set of possible actions at the state S_t . As a result of the chosen action, the agent can receive the numerical reward $R_t \in R(S_t)$ at just the once later. Thus, it realize itself as a replacement state S_{t-1} .

MDPs consists of states, actions, transitions between states, and reward perform (Fard and Pineau, 2011):

$$S_0, A_0, R_1, S_1, A_1, R_2, \dots, S_T, A_T, R_{T+1} \quad (1)$$

This trajectory is not fully random activity. A selected action at a selected state sometimes influences the reward and new state that arise. R_t and S_t are unit random variables that likelihood distribution depends on the continuing state and action. For any continuing state $s \in S$ and action $a \in A(s)$, the likelihood of reaching a state $s' \in S$ and getting a reward $r \in R$ ensuing step is given by the likelihood distribution function p (2) (Fard and Pineau, 2011):

$$S_0, A_0, R_1, S_1, A_1, R_2, \dots, S_T, A_T, R_{T+1}$$

$$p : S' \times R \times S \times A \rightarrow |R \quad (2)$$

$$(s', r, s, a) \rightarrow \Pr(S_t = s', R_t = r | S_{t-1} = s, A_{t-1} = a)$$

It is price noting that this perform p fully defines the dynamics of the MDP. The RL coaching task, therefore, consists of estimating p by interacting with the setting and observant the transitions that ensue.

From p , the helpful functions for the RL task are computed. As an example, associate estimation on smart associate action a is at a selected state s by computing the expected immediate reward $r(s, a) := S \times A \rightarrow R$ (3),

$$r(s, a) := E [R_t | S_{t-1} = s, A_{t-1} = a] = \sum_{r \in R} r \sum_{s' \in S} p(s', r | s, a) \quad (3)$$

Policies and Reward

A policy π could be a performance that evaluates this state of the setting. Formally, a settled policy perform is outlined as $\pi : S \rightarrow A$. Otherwise, it additionally potential to outline a random policy as $\pi : S \times A \rightarrow [0, 1]$. Each state $s \in S$, it holds $\pi(s, a) \geq 0$ and $\sum_{a \in A} \pi(s, a) = 1$.

The RL agent must learn the optimum policy that maximizes some life of the expected total reward. There are many choices to outline this measure. Maximizing the immediate reward R_{t+1} would end in a short-sighted greedy policy that prioritizes short rewards because of the value of long-run losses. For a better choices, maximize the add of rewards G_t (4),

$$G_t := R_{t+1} + R_{t+2} + R_{t+3} + \dots \quad (4)$$

However, G_t still has some inconvenience. Firstly, within the case of infinite episodes, G_t might fail to converge. It does not appear logical to contemplate that every one rewards area unit is equally vital. Thus, discounting the addition of rewards can solve each problem and provides a place to a replacement discounted accumulative add of reward G_t (4.1) (Fard and Pineau, 2011),

$$G_t := R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}, \quad (4.1)$$

where

γ is the discount factor and it is usually in the range of $0.9 < \gamma < 0.999$.

2.2.2 Estimating The Optimal Policy

Now that the optimal policy has been mathematically defined, an approximation algorithm is needed to estimate such policy by interacting with the environment to collect experience (Fard and Pineau, 2011). Deep neural networks are a good option for such tasks due to their effectiveness at approximating functions. Hence, $\pi_\theta: S \rightarrow A$ is defined as a neural network with parameter, θ .

Q-Learning Algorithm

Q-learning is considered the first successful deep reinforcement learning (DRL) implementation. The basic rule of the Q-learning algorithm (5) can be expressed as (Saadatmand *et al.*, 2020)

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha (r_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)), \quad (5)$$

where s_t, a_t and r_t are the state, action, and the rewards at time t , respectively. In addition, γ, α and $Q(s_t, a_t)$ are the discount factor $\gamma \in [0, 1]$ to guarantee the divergence of the value function at time t with state s_t and the chosen action a_t , respectively.

The main idea of Q-learning is to learn a function that estimates the value of pairs (Jang *et al.*, 2019). Then, the optimal estimated policy consists simply of selecting the action a to predict the maximum value at state s . Formally, the value function of Q_π^* (6) as (Saadatmand *et al.*, 2020),

$$Q_\pi^*: S \times A \rightarrow R \quad (6)$$

$$(s, a) \rightarrow E_\pi [G_t | S_t = s, A_t = a]$$

Note that Q_{π}^* depends on the associated policy π , as this policy defines the behavior of the agent during the episode. Therefore, the continuous future states and rewards arise under such policy (Jang *et al.*, 2019).

Furthermore, the pure exploitation approach is used in pure Q-learning. To select the action, only the optimal policy is allowed. However, this method can be inefficient when it gets shocked in local minima (Jang *et al.*, 2019). There is sufficient exploration in learning to allow the agent to select a nonoptimal action. To overcome the exploration concern in ϵ -greedy methods, the action can be decreases exploration with increasing the learning process. Then, the policy becomes the effectiveness of the controller (Jang *et al.*, 2019). In other words, the exploration needs to decrease by increasing the learning process.

2.3 Swarm Robotic Behaviors

In this section, the approach used to evaluate collective behavior is to incorporate the idea of behavioral series. To understand the role of subgroups in a robotic swarm, behavioral sequence analysis shows the transformation of robotic action from the point of view of specialization. By applying this approach, the collective actions of multi-agent robots can be observed. These group actions are classified into three main categories: (1) spatial behavioral organization, (2) navigational behavior, and (3) collective decision-making. It is based on the grouping by (Brambilla *et al.*, 2013). Following is a brief description of collective behavior and a description of the problems involved in the robotic swarm.

2.3.1 Spatially Organizing Behaviours

These behaviors make it possible for robots to travel in a swarm in the world to spatially arrange themselves or objects. These behaviors can be classified into several possible ways included: aggregation, pattern forming, chain formation, and self-assembly. Moreover, robots can also physically move objects to create clusters and structures. The more description for this behaviors classification will be an introduction as follow (Christensen *et al.*, 2020):

- *Aggregation* (Soysal and Şahin, 2005): In a particular region of the environment, aggregation pushes the individual robots to congregate spatially. This enables the swarm's individuals to get close to each other spatially for more interaction.
- *Pattern forming* (Spears *et al.*, 2004): The swarm of robots in a particular shape is organized by pattern forming. Chain forming is a special case where robots form a line, usually to create multi-hop communication between two points.
- *Chain formation* (Khaldi and Cherif, 2015): In the behavior of chain formation, robots need to position themselves to link two points. The chain that they shape can then be used as a navigation or surveillance guide.
- *Self-assembly* (Trianni and Nolfi, 2011): To create structures, self-assembly links the robots. Via communication connections, they can either be linked physically or electronically. Morphogenesis, where the swarm forms into a predefined form, is a special case.
- *Clustering of objects* (Durrant-Whyte *et al.*, 2012): Object clustering and assembly help the robotic swarm to control objects that are spatially distributed. For construction processes, clustering and assembly of objects are important.

2.3.2 Navigational Behaviour

These behaviors allow the organized movement of robot swarms in the environment. Following is such an example of navigation behaviors (Christensen et al., 2020):

- *Collective Discovery* (Ducatelle et al., 2014): To discover it, collaborative exploration navigates the swarm of robots cooperatively through the environment. It can be used to get an overview of the situation, scan for objects, track the environment, or create a network of communication.
- *Collective Motion* (Turgut et al., 2008): The swarm of robots in a formation is moved by synchronized motion. The formation may have a shape that is well defined. For example, a line, as in flocking, or be arbitrary.
- *Collective Transport* (Baldassarre, Parisi and Nolfi, 2006): Collective movement by swarming robots allow items that are too heavy or too large for individual robots to be transported collectively.

2.3.3 Collective Decision Making

These behaviors allow the robots to take a common decision on a given problem in a swarm. Following is such an example of collection decision making (Christensen et al., 2020):

- *Consensus Achievement* (Garnier and Navas, 2012): Consensus helps the individual robots in the swarm to agree on or converge from several alternatives to a single common preference.
- *Task Allocation* (Pini et al., 2011): Task allocation allocates evolving tasks to the swarm's robots dynamically. It aims to optimize the whole swarm system's

efficiency. If the robots have heterogeneous capabilities, to further improve the system's efficiency, the tasks can be spread accordingly.

2.4 Challenge in Multi-Agent Reinforcement Learning (MARL)

2.4.1 The Curse of Spatiality

The curse of spatiality refers to numerous phenomena that arise once analyzing and organizing information in high-dimensional areas that don't occur in low-dimensional settings like the three-dimensional physical area of everyday expertise. It means the error will increase with the rise within the range of options. It refers to the fact that algorithms are tougher to style in high dimensions and sometimes have a period exponential within the dimensions. Within the basic RL formula, the calculable values for every attainable distinct state or state-action could result in a directly exponential increase in procedure complexness (Buşoniu, Babuška and De Schutter, 2010). The complexness of dirt is a result of the number of agents that have its variables. Thus, every agent facing the moving-target learning drawback.

2.4.1 The Exploration-Exploitation Trade-Off

The key challenge that arises in coming up with reinforcement learning systems is in leveling the trade-off between exploration and exploitation. The exploration-exploitation trade-off needs the RL formula to balance between the exploitation of the agent's current data and exploration of the gathering action taken to enhance knowledge (Buşoniu, Babuška and De Schutter, 2010). In RL, if the agent is driven by maximizing setting rewards, the coaching method can renounce to explore higher actions that are incorrectly calculable to be less profitable. On the opposite hand, powerfully motivating

it to explore may result in failing coaching convergence, because the actions dead may well be too random to extract helpful data concerning the setting. For advanced environments, the state area is simply too broad. It's not computationally possible to explore the complete state and action area, the agent should prohibit itself to a comparatively little set of areas. Leveling what proportion the agent ought to explore is essential to satisfactory coaching.

2.4.3 Reward Shaping

Reward shaping is a good technique for incorporating domain data into reinforcement learning (RL) (Servin and Kudenko, 2008). Existing approaches like potential-based reward shaping usually modify the use of a given shaping reward performance. However, since the transformation of human data into numeric reward values is commonly imperfect because of reasons like human psychological feature bias, fully utilizing the shaping reward performance could fail to enhance the performance of RL algorithms (Hu *et al.*, no date). For instance, achieving the human goal from words to the same reward is difficult. Thus, if the reward isn't aligned with the first human goal, the agent may or otherwise to maximize the reward while not fulfilling its original task (Buşoniu, Babuška and De Schutter, 2010).

CHAPTER 3

METHODOLOGY

In this section, the distributed multi-robot system will be introduced in detail. The design procedure of the mobile robot model for the real-time control system will also be introduced in this section. The design procedure starts from the chosen software and application to hardware. Moreover, the system has been divided into two main sections for testing and verification, the line follower mobile robots and the obstacle avoidance under a given environment. The goal is to design a mobile robot that follows the black line during the experimental test. It should also navigate the multi-agent mobile robot in a given environment without colliding with obstacles or with other mobile robots. To achieve the objective of this project, RL based approach will be introduced in this section.

3.1 Flow Chart

The flowchart for this project is divided into 3 different flow charts which are the line follower flow chart, the obstacles avoidance flow chart, and the combination of the line follower and obstacles avoidance flowchart. This flow chart is divided into three to analyze the whole process of the robots to complete one task at each time. Plus, it is easy to identify the problem if the robot not on the path asset. The flow chart explains as follow:

3.1.1 Line Follower Flow Chart

Figure 3.1 shows that the flow chart of line follower mobile robots. The system for the line follower is designed using a flow chart defining. Hence, it will be following

the designated line that on its path. The flow chart contains the decision of the robot taking to follow the line path. The line follower robot depends on the sensor system the process takes time. The robot is made to be able to reach its destination so that it becomes speedier and more effective for its work.

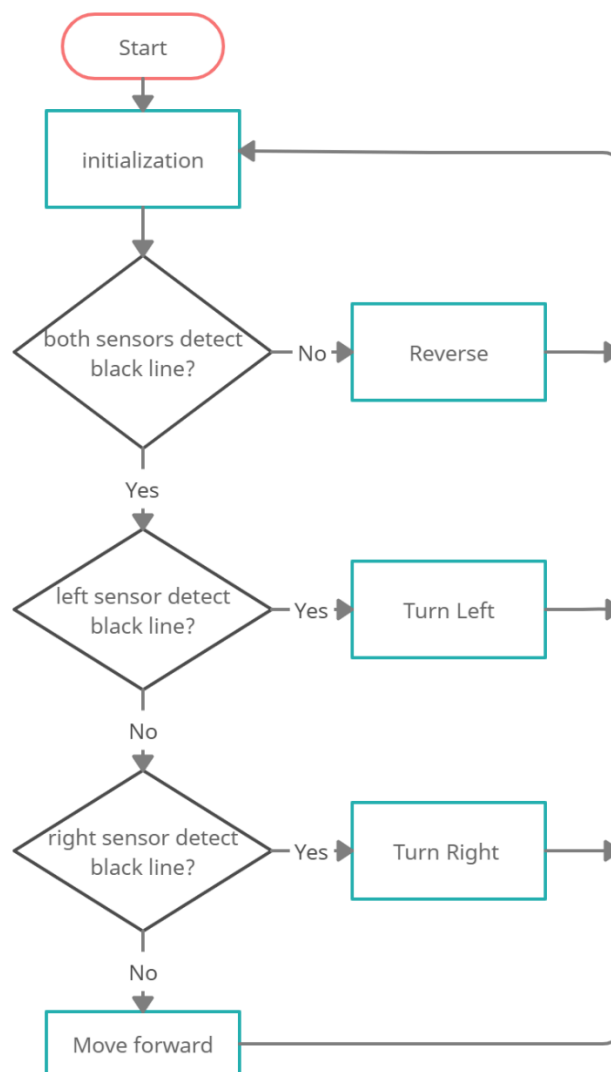


Figure 3.1: Line Follower Flow Chart

3.1.2 Obstacle Avoidance Flow Chart

Figure 3.2 shows that the flow chart of obstacles avoidance mobile robots. Based on the flow chart, if the distance ahead is less than 10cm, the controller will prompt the motor to turn at a 90-degree angle and move in the forward direction. The IR sensor will send out its signal once the ultrasonic part is clear. However, if it also detects the obstacles, the motor is prompted to rotate in an anti-clockwise direction for a reverse of the mobile robot to take place. Then, it will turn to the right and continue in the forward direction.

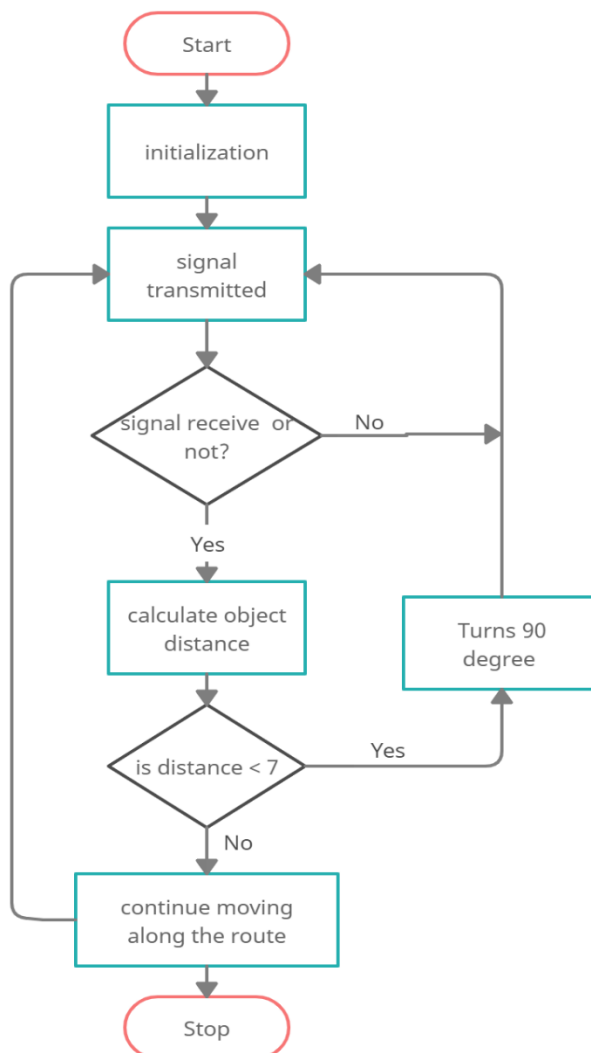


Figure 3.2: Obstacles Avoidance Flow Chart

3.2 Reinforcement Learning Based Approach

The line follower and obstacle avoidance problem in multi-agent environment problems is develop using the suggested RL-based approach. The RL-based approach may be a suggested approach because it won't to teach the robots to speak with one another through the leader-follower method. So that, the multi-agent robots can move together in line also avoid collision during a known environment. Each robot will learn incrementally an efficient decision policy over state-space by trial-and-error where the sole input from the environment is delayed scalar reward (Khan, 2019). The task for every agent will maximize the long-term discounted reward per action. In this thesis, the suggested approach is predicated on simple cooperation among them since the robots are homogeneous. In fact, during this approach, the robots will communicate instantaneous information among them and update an equivalent decision policy. The robots have the benefit by sharing the policies they learned because the robot is going to be faced with an equivalent situation. The robot must update an equivalent policy (Khan, 2019). A policy during this work is represented by a table of Q values, where rows correspond to situations and columns to possible actions. Therefore, public knowledge is learned by all robots. Since all the robots exploit and reinforce the public knowledge, each robot will use undirect experimental knowledge that's acquired by other agents, and quick learning is achieved (Polina *et al.*, 2017).

Moreover, to supply the group behavior of the multi-agent robot, all the autonomous robot system must be ready to follow the straightforward maneuvering control rules as follow (Azouaoui *et al.*, 2006):

- Follow the line and stay together as a group
- Avoid obstacles within the group
- Follow the group based on their speed and heading

The multi-agent robot must be ready to move in line by following the road from the start until the top of the road. As an example, avoid the obstacles while following the road requires the agent to rotate and calculate the space and offset from the obstacles to make sure that collision among the agents is not happening (Ghosh *et al.*, 2017). To realize this sort of task, the suggested Q-learning technique to regulate basic line follower robots. The multi-agent swarm robot is distributed during a predetermined environment and must move by following the road and avoid the obstacles supported by the RL approach. Afterward, this multi-agent swarm robot must navigate during a known environment to realize the target employing a leader-follower technique.

3.2.1 Formulate Learning Task

It is important to define a criterion on whether the robot successfully makes turns while avoiding the obstacles and back to the road again afterward. Consider the road follower robot shown in Figure 3.6, the robot may be a move-in line by the detection from IR and ultrasonic sensor. As mentioned previously, the detection of the sensor sometimes takes time and with the training to make sure the robot to stay follow the road and avoid the obstacle. Therefore, the task of the RL multi-agent swarm robot is formulated as learning by applying a leader-follower technique to make sure the agents ready to move following the road and obstacle avoidance. A leader-follower technique is where the leader must study the environment first and transmit the info to the follower via communication.

3.2.2 Adding Perception

As described within the objective section, the RL-based multi-agent swarm robot formation control should be ready to navigate the agents require to follow the road and turn successfully to avoid obstacles during a given environment. Hence, the agents should have enough perception to remember the environment and therefore the distance from obstacles including space from one another to avoid collision while experimenting.

The important information about the environment is that the distance between the robot and therefore the obstacle. Ideally, all of the agents should keep a minimum distance far away from an obstacle to perform a successful turning. However, to realize this whole perception of the environment, the robot would require to put in multiple sensors. Besides, every sensor has its functionality. Therefore, during this case, there are several sensors are needed to hold different tasks as described within the objective. The most sensors that require to be installed are going to be an IR sensor to detect the road because the objective is to follow the road while an ultrasonic sensor is installed to occupy the space of the obstacles from robots and obstacles within a particular range so that the robot will stop and turns because the obstacle is detected. The Bluetooth module sensor is required to make communicate among the robots. Thus, the leader-follower technique is often conducted.

Based on the formulation of the training task, a minimum requirement is to follow the road and to avoid collision between each of the robots and therefore the obstacles the maximum amount as possible. On the opposite hand, it's acceptable for the robots to not turn perfectly in 90 degrees while avoiding the obstacles as long because the robot doesn't hit the obstacles and successful turns and ready to find the track line back.

3.2.3 Environment Design and Modelling

The simulation and the real-time experimental platforms are designed to validate the proposed algorithm. It is required to determine whether reinforcement learning-based approaches improved the learning performance of multi-agent swarm robot formation control in a given environment. In the literature review, reinforcement learning algorithms are compared based on the following performance measures:

The cumulative reward

One way to show the performance of reinforcement learning is to plot the cumulative reward as a function of the number of steps. The cumulative reward is the total reward accumulated over time by following the actions generated by policy starting from an initial state. Besides, it provides a quantitative measure of the quality of the robot trajectory all over the episode. If the movement trajectory follows the line and avoids obstacles in minimum steps, the effect of the positive reward received dominated the sum of the reward. In contrast, if the major portions of the trajectory surround obstacles, the negative rewards received will dominate the sum. The ideal behavior of the cumulative reward during the training is explained as in Figure 3.3 (Bosch, Seeliger and Van Gerven, no date),