INVERTED-V SHAPED PLATOON FORMATION AND PATH PLANNING OF A MULTI-ROBOT SYSTEM IN AN OCCLUDED ENVIRONMENT

Thesis submitted for the partial fulfillment of the requirement of the degree

MASTER IN CONTROL SYSTEM ENGINEERING

Submitted by

Arijit Kumar Haldar

Class Roll Number: 002010804015

Examination Roll Number: M4CTL22015

Registration Number: 131409 of 2015-2016

Under the Guidance of

Prof. Madhubanti Maitra

Department of Electrical Engineering

Faculty Council of Engineering and Technology

Jadavpur University

Kolkata-700032

August 2022

JADAVPUR UNIVERSITY

Kolkata- 700032

Certificate of Recommendation

This is to certify that **Mr. Arijit Kumar Haldar (Class Roll No. 002010804015, Examination Roll No. M4CTL22015)** has completed his dissertation entitled "**Inverted-V Shaped Platoon Formation and Path Planning of a Multi-Robot System in an Occluded Environment**", under the direct supervision and guidance of **Prof. Madhubanti Maitra**, Electrical Engineering Department, Jadavpur University. We are satisfied with his work, which is being presented for the partial fulfillment of the degree of **Master in Control System Engineering** from Jadavpur University, Kolkata-700032.

Madhubanti Maitra 05.08.2022

Professor
Electrical Engineering Department
JADAVPUR UNIVERSITY
Kolkata-700 032

Dr. Madhubanti Maitra
Professor, Electrical Engineering Department
Jadavpur University, Kolkata - 700032

Saswati Mazumdar 8.8.22

Head
Electrical Engineering Department
JADAVPUR UNIVERSITY
Kolkata-700 032

Prof. Saswati Mazumdar
Head, Electrical Engineering Department
Jadavpur University, Kolkata - 700032

DEAN
Faculty of Engineering & Technology
JADAVPUR UNIVERSITY
KOLKATA-700 032

Mazumdar 08/08/22

Prof. Chandan Mazumdar
Dean, Faculty of Engineering and Technology
Jadavpur University, Kolkata - 700032

Faculty Council of Engineering and Technology

JADAVPUR UNIVERSITY

Kolkata- 700032

Certificate of Approval

The foregoing thesis is hereby approved as a creditable study of Master in Control System Engineering and presented in a manner satisfactory to warrant its acceptance as a pre-requisite to the degree for which it has been submitted. It is understood that, by this approval the undersigned does not necessarily endorse or approve any statement made, opinion expressed, or conclusion therein but approve this thesis only for the purpose for which it is submitted.

Final Examination for Evaluation of the Thesis

_____

_____

_____

Signature of Examiners

***Only in case the thesis is approved**

Declaration of Originality and Compliance of Academic Ethics

I hereby declare that this thesis contains a literature survey and original research work done by the undersigned candidate, as part of his Master of Control System Engineering studies.

All the information in this document has been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name: ARIJIT KUMAR HALDAR

Examination Roll No.: M4CTL22015

Thesis Title: Inverted-V Shaped Platoon Formation and Path Planning of a Multi-Robot System in an Occluded Environment

Signature: Arijit Kumar Haldar 05/08/2022

ACKNOWLEDGEMENTS

I express my sincere gratitude to my supervisor, Prof. Madhubanti Maitra, Department of Electrical Engineering, Jadavpur University. Kolkata, for her invaluable guidance, suggestion, and constant support, which made it possible for me to complete my thesis successfully.

I especially thank Mr. Dibyendu Roy, TCS Research and Innovation, Kolkata, for being a constant source of encouragement, inspiration and for his valuable suggestions throughout the research work. Not only did his technical expertise in this domain help me steer in the right direction, but also his support whenever I was stuck in any problem. I am highly indebted to him for his inputs and suggestions in this work.

I am also thankful to all the staff of control system laboratory for providing constant encouragement throughout the thesis work and for letting me have a pleasant environment to work in.

I would like to acknowledge all my friends in the Control System Department for their valuable discussions, suggestions and general advices, information and encouragement and all other friends who motivated me all along. Special note of thanks to Mr. Samar Kumar Biswas, A.G.M. (Electrical), M.B.B. Airport, Tripura and my friend, Miss Antara Mal, Counselling Psychologist, BetterLyf Wellness for their constant words of encouragement.

Last but not the least, I extend my words of gratitude to my parents, Mr. Avijit Kumar Haldar and Smt. Mira Haldar, for personally motivating me to carry out the work smoothly.

Arijit Kumar Haldar

Arijit Kumar Haldar
Jadavpur University
Kolkata- 700032

Date: 05/08/2022

ABSTRACT

The domain of robotics has seen major applications in all fields of human life in the recent years. The ability of automated and efficient performance of tasks have improved the quality and speed of completion of tasks. Swarm robotic system has various advantages in regards to other robotic systems, which is faster time of execution, redundancy, efficiency in search and rescue operations, environment sensing, among others. Swarming behaviour is observed among many species of biological creatures. Scavenging for food source, defense against predators and hunting are some of the reasons that swarms of bees, flocks of birds or schools of fishes display swarming behaviour.

Inspired from the swarming of biological agents, a swarm of robotic agents can be used to scan and actuate through an unknown occluded environment efficiently. This work is focused on developing a control logic for platoon formation of a multi-robot system. Mimicking formation flight of birds, an inverted-V formation control is chosen, which has environment sensing and path planning capabilities through an occluded environment and reach a pre-allocated target.

The control algorithm has been tested against multiple simulated environments with different occluded paths. The results presented, highlight the claim that this algorithm is capable of an inverted-V platoon formation, sensing, path planning and actuation. It has been assumed that agents, if capable of forming and maintaining inverted-V platoon like that of birds, the multi-robot system simulated in this work can gain advantages, similar to that obtained by birds; provided that the environment through which the agents actuate has similar lift and drag characteristics as that of air.

CONTENTS

List of symbols

N : Number of agents forming the platoon

N : Set of natural numbers

n : Iteration step number in simulated environment

i : Any agent in the platoon

j : Any other agent other than the i^{th} agent

u_a: Controller effort due to attraction controller

p^i: Current position coordinates of i^{th} agent

p^j: Current position coordinates of j^{th} agent

u_r: Controller effort due to repulsion controller

r_s: Radius of region of repulsion around the agent

b: Wingspan

WTS_{opt}: Optimal wingtip spacing between adjacent birds

δ: Upwash region due to one wing

e_i: Overlap of wingspan with upwash region of another bird

T: Target or goal coordinate

p_o^i: Initial coordinate of i^{th} agent

v^i: Current velocity of i^{th} agent

$\underline{p}^i$: Desired coordinate of i^{th} agent

$\overline{v}$: Mean velocity of all agents

d^i: Distance matrix of sensed obstacles from i^{th} agent

$d_{thre\_}/d_{thre_+}$: Sensor threshold limits for region of obstacle sensing

E: Total energy available in each agent at beginning of simulation

e_1: Energy required for running local controller per second

e_2: Energy required for running global controller per second

E_{thres}: Energy threshold for leader agent after which swapping is necessary

E_{avail}^{leader}: Current energy available in the leader agent

$E_{avail}^{followe}$: Current energy available in any of the follower agents

Ω: Resource profile area through which platoon actuates

u^i: Combined controller effort on i^{th} agent with attractive and repulsive interaction

k_a^i/k_1: Attraction controller gain for i^{th} agent

k_r^i: Repulsion controller gain for i^{th} agent

$\tilde{R}^i$: Predicted obstacle position matrix from sensing data

r: Radius of virtual circle

n_a: Number of agents on one arm of inverted-V formation

n_{a1}: Number of agents on completely filled arm of inverted-V formation with even number of agents

n_{a2}: Number of agents on the other arm of inverted-V with even number of agents

d: Inter-agent distance of separation

err_{goal}: Position error of each agent from goal coordinate

J: Weightage of gradient on resource profile

o^m: Position of sensed m^{th} obstacle

m`: Total number of obstacles sensed

err_{pos}: Position error from desired position

err_{vel}: Velocity error from mean velocity

P: Matrix containing position values of each agent

k_2: Gain in velocity correction

k_v: Velocity damping factor

k_f: Global controller gain

tmp: Temporary variable for agent position swapping action

List of figures

List of tables

CHAPTER 1

INTRODUCTION

1.1 Introduction

A robot is a special kind of device that replicates human activities. It can perform a series of operations (if properly programmed) without the involvement of any human operator [1]. Swarm robotics is a branch of robotics [2] where the coordination among multiple robotic agents is studied. Hence, such kinds of systems, either homogeneous or heterogeneous, work collaboratively to achieve the mission target. Swarming is noticeable in the collective social behaviour of many biological creatures, where platoon formation is a commonly observed phenomenon.

Platoon formation defines a pattern that must be maintained while actuating towards a target. Maintaining platoon formation by a swarm of robots during navigation offers various advantages when compared to moving individually [3]. One such benefit is exposing only a small number of the agents to adversity or improving the group's ability by allowing members to limit their perceptual focus to a specific portion of the environment. Hence, it aids in the system's robustness and efficiency.

If we look at birds in the sky, we would observe that birds, when flying in groups, they form noticeable pattern or formation. The most common formation that we notice among birds of many different species is the inverted-V formation [4]. Inverted-V formation is formed by birds flying in two lines, where the lines intersect each other at an angle, forming a leader bird and follower birds flying behind it with a definite lateral and longitudinal spacing.

Due to flying in this formation, birds gain some advantages over flying without any formation. The first advantage is the aerodynamic advantage [5]. Due to a bird flying through air, the lift generated results in forming downwash and upwash regions in the air [6], [5]. When another bird flies within the upwash region, it gets benefitted from the additional lift it receives, thus helping it to conserve energy. The second advantage is to obtain clear vision in the direction of flying [7]. Due to the formation, none of the birds obstruct vision of the other birds. This helps to birds to see clearly and be alert of any adverse conditions that might arise.

Hence, in this work, we mainly focus on the inverted-V platoon formation of a multi-robot system to mimic the formation flight of birds [8].

1.2 Swarm Robotic System vis-à-vis Single Robot System

Swarm robotics is an approach in which multiple simple robots (multi-robot) work collectively, which are limited by their individual abilities [2]. So, the collective behaviour of the system can be obtained from the interaction of the agents with the environment. A set of simple rules governing the individual agents form complex swarm behaviour [9] which are capable of handling mission goals that any of the individual agents would not be able to perform efficiently, if at all possible.

Single robot system on the other hand consists of one robot that is fully equipped and capable of handling all the mission goals. These robots are usually physically larger [10] and have higher capabilities than any one agent of a swarm robotic system. The control algorithm for a single robot system is complex, since one agent is required to complete the task assigned, and may need multiple hardware components to meet mission needs [9].

Any robotic system designed has to address some issues that are essential for it to be useful.

- Robustness: Successful completion of any task by a robotic system depends on accumulating data from sensors and computing based on some algorithm to do some task. But, real life sensors and working conditions are not ideal [11]. So, there are sensor data fluctuations, erroneous readings or failure in some non-essential components. The system should be able to work accurately even in these uncertainties.
- Time of execution: One of the reasons for using robots instead of doing tasks manually is that robots are faster. In mission-critical situations, the speed of a robotic system is even more crucial, because time of operation may be the result of casualties, if the robotic system is too slow.
- Path planning: When a robot or a system of robots has to reach a certain location, the ability to sense the environment and plan a path through it is essential. The path planning has to be made autonomously, without the need of any manual or external aid for the system to be able to work in any environment.
- Immune to environment conditions: Robots may be deployed in a very harsh environment, where it was never tested [11]. They must be able to perform efficiently while dealing with these new unanticipated difficulties. Otherwise, the mission's performance may suffer.

These issues are addressed in a swarm robotic system. Due to multiple robots, a swarm robotic system is robust, which is discussed in section 1.2.1. Time of execution is reduced due to cooperative work. Inter-agent communication ensures that the swarm robotic system can work in any unknown environment and have path planning capability. Presence of multiple, identical agents means that if one agent fails, any other agent can take its position, and the system is not severely affected in any way [9].

1.2.1 Advantages of a Swarm Robotic System

A swarm robotic system is comprised of many identical robotic agents, each with the capability of sensing its surrounding and communicating with its immediate neighbors, such that these agents can cooperatively work towards a common mission aim. Single robot systems are much more complex than swarm robots and have more points of failure, moreover, single robot systems cannot be used in situations where a large environment has to be sensed to locate a target, with autonomous path planning and sensing capabilities. Lack of redundancy means that single robot systems are more complex and prone to failure.

Some of the reasons which make swarm robotic systems more suited are:

- Robustness: As discussed in section 1.2, fault tolerance and fail-safe system make swarm robotic systems robust [11]. The two main reasons for swarm robotic system to be robust are as follows.
    - Redundancy: As we discussed earlier, all agents in a swarm are identical [12]. Thus, in the event of an agent failure, any other agent can take up the role of the failed agent, without adversely affecting the entire system [9].
    - No single point of failure: Due to the ability for agents to take the position of any other agents in the swarm, small number of agent failure does not affect the outcome of the mission [9]. Thus, this distributed nature of swarm system has no single point of failure [12].

- Flexibility:
    - Since we have identical robots [9] making up agents of a swarm, so, every agent can complete the other's task, if required.
    - Cooperation and coordination in the swarm facilitates a swarm robotic system to adapt [9] to a wide array of use-cases and ability to be deployed in a variety of situations with success.

- Scalability:
    - The swarm does not need to know the exact number of agents in the swarm at any instant of time. This facilitates addition or removal of agents to and from the swarm, with no changes in its control algorithm [13].
    - It allows testing of new algorithms on a small scale with few agents, and then scaling it up during deploying, without any noticeable change in behaviour, if the algorithm is properly designed [9].
    - If a particular swarm is developed keeping some basic functionalities in mind, another mission with similar functionality needs but different in size can deploy the same swarm with lesser or more agents [13], according to needs. This provides reusability.
    - Losing a small percentage of the swarm [13], due to some reason, scales down the swarm; but the swarm continues to function without posing any difficulty in the usefulness of the swarm thereafter.

A swarm robotic system is made up of simple robots, which are easier to fabricate and are also cost-effective. The control rules for such a system are simple, where inter-agent coordination and interaction with the environment bring out the complex swarm behaviour or swarm intelligence, which is not pre-programmed in any way.

This can be demonstrated by a very common use of swarm robotic system, which is search and rescue [14]. The distributed nature of swarm agents helps coordinate and search over a wide area in a very small duration [15], just by having the control rule to search each agent's immediate neighbourhood and communicate the same with its adjoining agents. This can be achieved by a single robot system much more slowly, and require complex search paths and algorithms to effectively cover the complete area. This gives a clear understanding that swarm robotic systems are very much essential when the mission needs simplicity, versatility, flexibility [13] and scalability, which would not be possible with an adequately equipped single robot.

One of the biggest advantages of a swarm robotic system is that it has the shortest execution time [16] compared to other robotic systems. When traversing through an unknown, occluded

environment, path planning ability is most time efficient in case of a swarm robotic system. Due to the presence of multiple agents, environment scanning is performed by each agent simultaneously over a wide area [16]. Inter-agent communication and collaboration helps the swarm robotic system to decide on a path based on the sensing data received from all agents, which ensures faster time of execution.

1.2.2 Theoretical Definition of Swarm

Swarming or aggregation is a cumulative behaviour displayed by a group of agents [12] acting cooperatively to bring out a collective performance as a whole. It lends additional aid to the capabilities already present in the agents [2]. It is the collective motion of a large number of self-propelled agents or entities which make up the swarm. The emergent behaviour arising from simple agent control rules is the defining factor in a swarm since the control algorithm doesn't involve any central coordination [9]. The rules defining the agent motion and communication involve simple motion control of individual agents along with inter-agent communication and environment sensing in a decentralized manner, the interaction of which brings about the behaviour of swarming.

Swarm behaviour can be widely seen in living organisms. From unicellular organisms such as bacteria [17] to insects such as ants [18] and bees [19], small fishes [20] and even birds demonstrate swarming [21]. It can also be seen in big herds of tetrapods such as deers [9], sheep [9] among others. Swarm behaviour in the animal kingdom helps the organisms to be safe from predators [9], ascertain correct direction and location of heading for all while moving over long distances [4], aid in visual communication [5] and be informed about the environment they are in better. This is called collective consciousness in a swarm system [12].

Organisms do not consciously think about forming a swarm [22]. The random motion of these individual creatures helps them synchronize with one another, resulting in visible groups or swarms. For example, locusts have random motion when flying alone, but, when 100 or more locusts come together, chemical receptors in their brain help follow every neighbouring locust, thus forming huge swarms of locusts [12]. This highlights another advantage of swarming behaviour in the animal kingdom. It helps agents for finding and getting proper nutrition or help in attacking its prey, for example in schools of fishes [20].

Swarm Robotics is a field of multi-robotics [2] in which large numbers of robots are coordinated in a decentralized way. Agents in a swarm robotic system are equipped with a set of sensors that allow them to perceive their immediate surroundings, recognize regions that

help or impede mission needs, and convey this information to other agents in their vicinity. The collective information assists the swarm in achieving an organized behaviour [23].

The number of agents to be identified as a swarm system is not strictly defined. As is seen from animal kingdom, with too few agents, swarm behaviour is not present [9], or at least not noticeable. For this reason, H. Hamann [9] states the following points, while deciding the size of a swarm.

i. It is said that the swarm should not be so large that it can be dealt with statistical averages, while also not small enough to be dealt with few-body-problems [9].

ii. These limits can be quantitatively defined as less than the amount for 1 mole and more than 100. The order of which is $10^2 < N < 10^{23}$ [12].

iii. But multi-body dynamics gets very complex from 3 onwards, so, a swarm system can easily be realised with 3 or more robots, while the upper limit stays at 10^{23} [9].

The agents in a swarm are intended to be outfitted with sensors that allow them to sense their immediate environment [12]. Based on this sensing data, they recognise positive and negative artefacts in the environment based on mission requirements [24] and can communicate with nearby agents [25]. This results in a swarm robotic system, which is made up of a collection of sensors and actuators that all work together to achieve a common goal. The requirements for agents to be included in a swarm can be summarized as follows:

i. The dimension or projected ground area required for safe operation for each agent should be known.

ii. Local sensing ability should be present for each agent, so that, they can scan their immediate surroundings and look for obstacles, target, or members of the swarm. The controller then can plan according to the sensor data obtained.

iii. Local communication should be present among the neighbouring agents so that they can distinguish an obstacle from a swarm agent [9]. Also, it is essential for coordination and planning.

One important aspect in a swarm robotic system which is aimed at creating a platoon while actuation, is formation control. It is a swarm control rule that is defined for multi-robot systems which exhibits a platooning behaviour [3].

1.3 Platoon Formation

Formation control in a swarm robotic system is used to describe the shape or pattern the swarm should form during actuation. It provides information about the spatial distances between each agent [26], and the method to maintain it, while navigating through an occluded environment [26], [27]. Formation control is necessary when the agents are supposed to form a defined shape. This might benefit detailed environment sensing or to avoid some adverse conditions in the environment. Different formation topologies are chosen [28] based on mission needs and the controller is designed accordingly.

Platoon, in the context of a multi-robot system refers to a group or swarm of robotic agents having a defined shape or formation, usually maintaining agent hierarchy in the formation during actuation [26]. Platoon formation is the foundation of formation control system in a multi-robot system [3], where the shape or formation is a mandatory criterion for the controller to maintain throughout the duration of the mission.

Platoon formations require agents to occupy specific spatial positions [28]. Because this takes up more space [5] than a swarm system without platoon formation, assembling a platoon with a high number of agents makes navigating an occluded environment challenging. As a result, a multi-robot system with platoon formation contains a lesser number of agents [5]. Figure 1.1 depicts three simple platoon formations: a line, an inverted-V, and a circle.

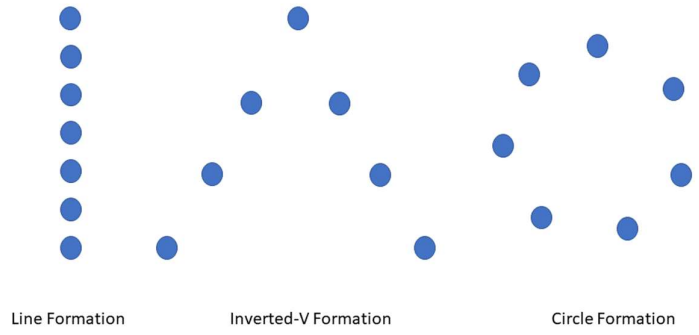

Fig. 1.1 Examples of some multi-robot platoon formation configuration

Now, depending on the mission requirements, a special formation may be necessary in some of the circumstances listed below.

i. Object carrying problem: If the mission needs are such that there is an object of known (or unknown) shape, where the platoon is required to acquire the shape [29] of the

object, and align itself around the object. This would facilitate the object to be carried by the platoon of agents, ensuring that the object is not dropped [30] due to breaking formation and reach an assigned target region.

ii. Zone creation problem: The requirement of the mission might be such that the agents form a platoon, so that the region of interest is divided into one or multiple zones [31]. This might be used for region selection and identification, or due to defensive or offensive enforcement [32]. Such an arrangement is required to prevent undesired entities from entering the militarized zone created by the platoon.

iii. Bio-inspired formation: We find formation flight in birds where birds such as starks, geese, etc., exhibit inverted-V formation while flying over long distances [5]. This formation flight is advantageous to the birds due to energy depletion minimization and visual advantage while maintaining the formation.

When a formation is maintained by the agents in a platoon, the region occupied by the platoon is fixed. Now, if the size of the platoon is large, the region required by the platoon to occupy will be large [26] (depends on the shape and configuration). This would hinder the platoon from going through constricted space [33]. This adds a bottleneck to the applicability of the multi-robot system. Hence, the size of a multi-robot platoon formation system is usually small [33].

The space or region through which the agents traverse, is called the resource profile. It is also called the region of interest, and is henceforth used synonymously. All the agents, obstacles and the goal mentioned in this work, lie on this resource profile.

1.4 Basic Control Proposition

This work employs a hierarchical control strategy with a local controller running on every agent of the platoon, and a global controller running on the leader agent, which occupies the vertex of the inverted-V shape. Local controller carries out the task of sensing the environment and actuating to desired position coordinate at desired velocity during each iteration.

Global controller on the other hand takes a supervisory role and takes in sensing data from all local controllers. Based on this data, desired coordinate and velocity for the next iteration step are calculated. This data is fed back to the local controllers for individual agents to actuate.

1.4.1 Local Controller

Every robotic agent taking part in the platoon formation has a controller running on it. This controller, which is called here as the local controller, is identical for each agent. It is responsible for the agents to have sensing and locomotion capabilities.

Taking any one agent, we see that the local controller does environment sensing by checking for presence of any obstacles around the agent. Sensors are outfitted on each agent, spaced adequately, to allow the agent to scan 360° around it. This forms a circular region of sensed environment around each of the agents, the radius of which depends on the limitations of the sensor used to design the agent. This sensor may be a SONAR [32] sensor, LIDAR or any other non-contact sensing arrangement. The sensed data is communicated to the global controller from each agent, which provides the next desired coordinate for the agents to occupy, based on sensing data received and the present coordinate it is occupying.

Actuation is done by the local controller by calculating error in position between the present coordinate and the coordinate the global controller instructed the agent to occupy. The agent is actuated till this error is zero. Since all agents have their individual local controllers, so, all the agents reach their respective desired coordinates. Then a new set of coordinates are calculated by the global controller which again iterates through these steps.

1.4.2 Global Controller

The global controller runs only on the leader agent, and is a supervisory controller over all the local controllers running on each agent, including the leader agent.

Based on the sensing data received from each agent and their present coordinates, the global controller chooses a region with no obstacles. Next, desired coordinates for each of the agents are calculated by the global controller, such that they form the platoon shape. These coordinates are then communicated to their relevant local controllers, so that the agents can reach these coordinates and form the defined shape.

This continues in an iterative manner, till the distance between the goal or target coordinate, and the center of the virtual circle being fit, is negligible, which means that the platoon has reached the target coordinate.

1.4.3 Reason for hierarchical control strategy

The two-step controller design mentioned in sections 1.4.1 and 1.4.2 are implemented in this work, which simplifies the roles of the leader and the follower agents. Local and global controllers are segregated, because an inverted-V shaped platoon has definite leader and follower agents for the entire duration of operation. Assigning different controllers to agents with different functions in a swarm help us to define simple control rules, yet achieve desired result.

This hierarchical control strategy helps to keep the calculations for individual agent actuation separate from those for platoon handling. It allows the global controller to handle the platoon, without needing to calculate the agent parameters required to be set for the platoon to form and maintain. Close coupling of hierarchical controller efforts brings out the inverted-V platoon behaviour.

1.5 Concept of Agent Swapping

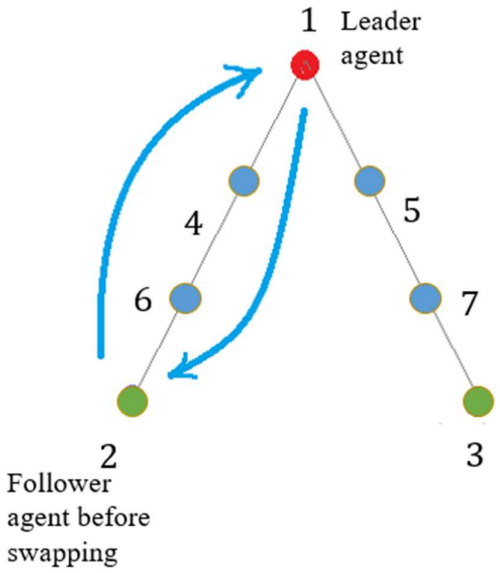

Fig. 1.2 Action of agent swapping

Based on the hypotheses of formation flight in birds, the upwash region generated behind every bird by the flapping of wings is utilized by the birds flying behind it in the formation [5]. Figure 1.2 shows the visualization of an inverted-V shaped platoon with seven agents. The leader agent is the agent occupying the vertex of the inverted-V shape. It is assigned the index of 1.

The last follower agent on each arm of the formation, which are shown here in green, are assigned indices of 2 and 3, as shown.

In this example, we have chosen to swap the positions of the second and the first agents. During swapping, fourth and sixth agents stays in their original position while the first agent and the second agent leave their positions. The first agent actuates towards the position vacated by the second agent and occupies it, while the second agent takes up the leader agent's position. This completes the swapping action.

Since an opposing force, such as drag or friction is not implemented in our work, so the agents actuate unhindered through the region of interest. But, in a hierarchical control strategy, as used in this work, some additional controller runs on the leader agent with respect to the follower agents, while all agents, including the leader agent has a common controller running. This causes more energy usage for leader agent. If the leader agent breaks down or its energy is completely depleted, the entire system will come to a standstill. This can also be handled by agent swapping.

In this work, we have chosen to perform agent swapping when the energy available in the leader agent falls below a pre-determined value. Detailed discussion on this is provided in section 3.5.5.2.

1.6 Contributions of the present work

In this work, we have proposed a control algorithm to create and maintain an inverted-V shaped platoon formation using robotic agents. The platoon will have sensing and path planning ability in an occluded environment.

The contributions of this thesis are as follows:

- The main contribution of the work is to develop and implement a formation control strategy of multi-robot system which mimics inverted-V formation of birds. A two-level hierarchical controller is designed and implemented for a team of robots navigating through an unknown occluded environment. Sensing, path planning, and actuation are autonomously performed by each agent of the group without the involvement of any human operator.
- Swapping of agent position is done to prevent leader agent from getting severely depleted of energy, which would hamper the system performance. According to the actuation rules designed in this work, the leader agent would have highest energy drain

due to two controllers running at the same time. To prevent failure of leader agent, one of the follower agents from the end of the inverted-V formation is brought in the position of leader agent and the global controller is handed over to it. This is done when the energy available to the previous leader agent is low.

- Finally, the algorithm, as mentioned in the previous step is being tested in a simulated environment for validation of the claim that this algorithm can generate and maintain platoon during actuation. Eight different environments are modelled to make the platoon encounter obstacles of different shapes, and their actuation times are recorded for each case. Also, the actuation time for this work is compared against [8] and [73].

1.7 Thesis Organization

The thesis is organized as follows:

- Chapter 1 introduces the concepts of swarm robotics, platoon formation, and agent swapping. The advantages of using a swarm robotic system are discussed here, along with introducing the concept of a platoon and its agent swapping. The control algorithm implemented in this work is briefly introduced here, with detailed discussions in Chapter 3.

- In chapter 2, we have discussed the literature on formation flight in birds that demonstrate V-formation. Here, we have showcased different configurations of the V-formation that is noticeable in nature. A formation model is discussed, which shows the optimal position of agents in the formation to maximize advantage from wind upwash. This chapter establishes the relation between the formation flight model seen in birds to how it can be implemented in a multi-robotic system and explains the basis of fluid drag minimization by achieving a defined formation which this work tries to achieve. It also revisits the agent control algorithms introduced in Chapter 1.

- In chapter 3, we have introduced the control strategy, mathematical definition and interaction of the hierarchical control system with the agents which are intended to form a multi-robot platoon of inverted-V formation. Here, we have described the method of environment sensing, platoon formation, path planning and actuation that is performed in this work. The agent control, calculation of controller effort and hand-over of agent position and velocity decision-making between the local and global controller is

discussed in a stepwise manner till the final actuation stage. This chapter covers the realization of the work in a simulated environment and discusses on the contribution towards the work, in particular.

- In chapter 4, we have compiled and discussed upon the results of the simulation performed, as discussed in chapter 3. The time required for completion of platoon actuation is provided in a tabular format for the different environments in which the algorithm is tested. Finally, we have done an actuation time comparison for this work against two recent papers.

- In Chapter 5 we have summarized the contributions of the thesis towards the work, highlighted the achieved features and pointed out the scope for future work in this regard.

1.8 Chapter Summary

This chapter discusses swarm robotics, platoon formation, and some fundamental concepts that will be employed throughout this study. It goes over the benefits of a swarm robotic system.

Finally, this chapter covers the problem that this work is intended to tackle as well as the motivation behind the use of a hierarchical controller. A systematic summary of this study is provided at the end of this chapter.

CHAPTER 2

Literature Review

2.1 Introduction

The effectiveness of a multi-robot swarm, or a multi-robot platoon in particular, is highly dependent on its control [35] and coordination method. Creating and sustaining a predetermined formation shape necessitates the use of a cooperative control framework [36], which is an attractive area of study.

Scientists from various domains have devised several ways for developing the requisite controller algorithms to produce prescribed patterns utilising multi-robot systems. The defined systems are tested to ensure that they are capable of operating in challenging and unknown terrain with obstacles.

In this chapter, we will look at how the behaviour of biological organisms has led scientists to develop formation control. In addition, various control algorithms from the literature are presented, which deal with multi-robot systems that can move through a territory without colliding with obstacles, guaranteeing formation is maintained.

2.2 Formation Flight in Birds

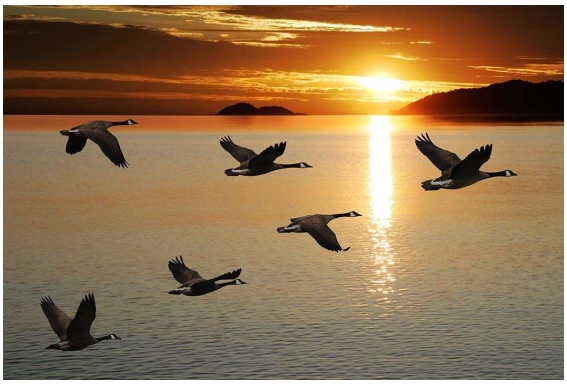

Fig. 2.1 Birds flying in a V-formation [37]

Formation flight has been observed in birds such as the goose, swan, crane, and pelican [4]. When we compare birds that exhibit formation flight behaviour to those that exhibit migratory activity, we discover that the majority [38] of small birds that exhibit migratory behaviour fly in an inverted-V formation. When flying over long distances, larger birds such as flamingos

[39], pelicans, cormorants, cranes, geese [40], swans, and others form very vast flight formations, whereas smaller birds [41] fly in small ordered groups.

This begs the question of what the benefits of flying in a formation are, and why there are disparities in the types of formation presented by birds. Before delving into the formation model, it is necessary to first understand the flight mechanics of birds. Each aspect of bird flight, such as hovering, lifting off, and landing, involves diverse wind flap motions that interact with the air to generate lift. Lift and drag [42] are the two most important aerodynamic characteristics in flight.

The action of air flow on the wing, which serves as an airfoil [43], produces lift force. Because the air above the wing surface is curved [42], it must travel a greater distance than the air below. This means that air moves quicker [44] over the wing surface than beneath it. As a result, the bird experiences a net upward [44] lift on its wings. Wind flapping generates additional lift by creating a downward air thrust. The wings "push" air downward, creating a column of downward flowing air generated by the flap [45] beneath the bird. Due to reaction force, this generates an upward push or lift on the bird's body. The wing creates lift during the downstroke, while the wings are folded for minimal air interaction during the upstroke [45]. The vortices formed by an owl's wing flap [46] are visible in Figure 2.2 due to tiny bubbles suspended in the air.

Fig. 2.2 Bird’s wing flap generating wind vortices [46]

Figure 2.3 shows the cross-section of upwash and downwash region generated due to wing flaps of a bird in a simplified illustration.

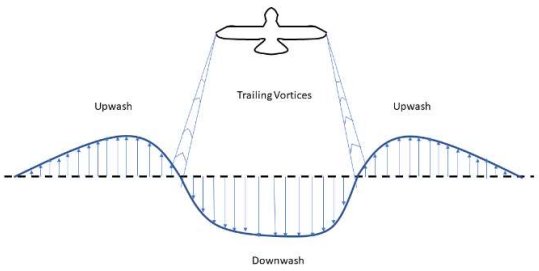

Fig. 2.3 Force distribution profile of a bird's flight path cross section

Aerodynamic drag is the force exerted on the body of the bird in the opposite direction [46] of velocity. This is related to air friction, also known as viscosity. There are two types of drag [4]: lift-induced drag and parasitic drag. Drag is the principal source of energy loss in the bird's flying mechanism.

- Lift-induced drag: The air column now under the wing span is forced down from both sides of the wing during the downstroke [44]. This results in a downward movement of the air column. However, because the wings are small [5], the air column towards the tip of the wings curves outwards due to the tangentially applied thrust. This causes air vortices on both sides of the wings to rotate in opposite directions.
  This reduces the overall lift [44] created by the wing in the first place and is an essential by-product of wing lift. This is referred to as lift-induced drag.
- Parasitic drag: Every bird's body is streamlined [7] to minimize interaction with air when in flight. However, during flying, air hits the front half of the bird [44] and its passage is obstructed by the bird's body and feathers. This exerts an opposing force [7] on the bird's body from the direction of its motion, contributing to energy loss [45]. This drag force caused by air friction is known as parasitic drag.

To decrease the issue of drag induced power drain, birds follow formation flight [41], when flying over long distances.

During long-distance flight or migration, a wide variety of birds fly together in a symmetric V-shaped [37] or a J-shaped [5] coordinated formation. Another variant of the V-formation is commonly known as an "echelon" [5]. The configuration is thought to assist save energy [4] and improve aerodynamic efficiency [4]. As previously explained, a bird flying through air has downward flow of air; nevertheless, due to the vortex region, air flows upwards around both sides of the wings [45]. This zone is known as the upwash region, and it is left in the wake of

a bird that has recently flown through a location [4] and generated the upwash region in the air column with its wing flap.

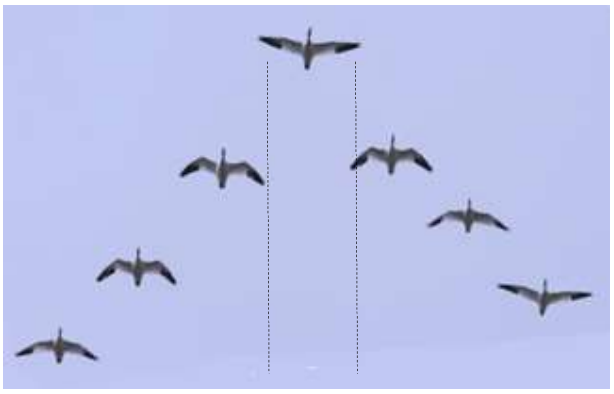

Fig. 2.4 Upwash region in bird formation flight [47]

The zone right outside the leading bird's wingspan (marked by dashed lines) has lift generated by the upwash region [4], as seen in figure 2.4. When another bird takes up that position on both sides of its wings directly behind the leader bird [5], the follower birds benefit from the extra lift. Multiple birds are tied together in this arrangement [4] to gain an aerodynamic advantage.

It is observed that when the V-formation is chained along an excessively long arm, the disruptions [5] created between the formation are magnified, resulting in the formation making and breaking when flying. As a result, the formation size in nature is modest [6] (between 25-30 birds constituting a platoon). This prevents formation breaking due to minor position flaws [4] multiplying with a bigger chain of birds, ensuring benefit from created lift.

For this particular V-shaped platoon formation, two hypotheses [4] are proposed: one for aerodynamic benefit and the other for visual field-of-view advantage. Drawing from [4], we are applying this platoon V-formation to non-biological organisms that can benefit from the similar advantages when crossing a viscous medium such as air or water.

While traversing such a medium, a multi-robot system of agents must oppose fluid drag, which results in significant power depletion [48]. However, travelling the same road while maintaining an inverted-V formation by the platoon reduces [49], [50] the platoon's overall energy drain rate. Because it benefits from flying in the upwash region of the agent directly ahead of it, which offers greater lift.

The control algorithm for platoon formation of a multi-robot system is the emphasis of this work. Agents reposition themselves to form an inverted-V shape and actuate through the resource profile, taking care not to break formation until the desired goal coordinate is achieved. The sensing and path planning algorithms incorporate obstacle avoidance behaviour.

2.3 Types of formation

The inverted-V formation, as mentioned in section 2.2 shows the following formation variations:

- V-formation [49]
- J-formation [5]
- Echelon formation [5]

These are different variations of the inverted-V formation observed in nature, which has nearly identical aerodynamic properties for the birds, but with different flock sizes of interest or configuration.

2.3.1 V-formation

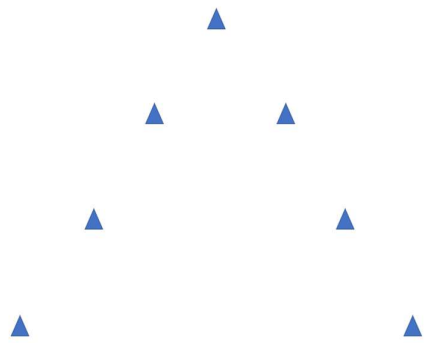

Fig. 2.5 Visual representation of V-formation with 7 agents

The V-formation, as illustrated in figure 2.5, is the platoon formation of agents with agents being equally filled on both arms of the inverted-V symmetrically [5].

2.3.2 J-formation

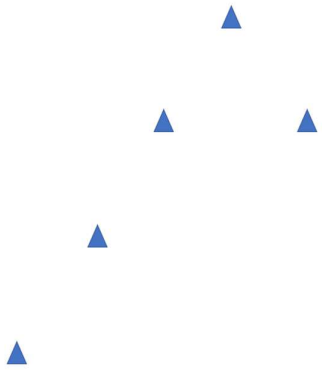

Fig. 2.6 Visual representation of J formation with unequal number of agents on arms of V

J formation, as illustrated in figure 2.6, refers to V-formation of agents but with unequal filling of agents on the two arms of the V-formation [5].

2.3.3 Echelon formation

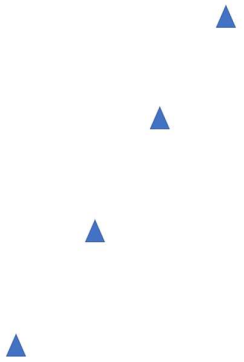

Fig. 2.7 Visual representation of echelon formation with one arm of V completely unfilled

Echelon formation, as illustrated in figure 2.7, is the variation of V-formation where only one arm of the V-formation is occupied by agents, while the other arm is empty [5].

Birds are thought to be able to identify [48] or see the region with the greatest support from the upwash region. This may assist them in occupying the region that provides them with lift [49]. The resulting formation is the physical geometry of the flight path's aerodynamic characteristics, as formed by the birds' wing flaps.

Some birds like to fly to the left, others to the right, and some to the center of any variant of V-formation. The birds flying at the ends and in front are rotated in a timed cyclical pattern [49], distributing flight weariness evenly across the formation members.

It is also notable in birds that they do not maintain the same V-formation during their flight, but rather switch [51] from one arm to the other. This is possibly to take advantage of natural air updrafts [44] caused by hot air rising in the region through which the birds are flying.

Flying in V formation involves not only positioning but also flapping time. The birds behind sync with the leading bird's flapping motion to follow the trail of upwash [44] left by the bird in front. The staggered development of birds described in Hamilton's work supports the visual communication idea. According to [51], staggering the longitudinal distance between birds aids in better visual communication by increasing their unobstructed range of view due to other birds.

According to Lissaman and Schollenberger's [41], a flock of 25 birds flying in formation can improve their range by roughly 71% when compared to a solo bird travelling via the same path. Willis et al. [52] also claim that synchronized flapping of avian wings accounts for up to 20% energy savings.

2.4 Relation between lateral and longitudinal distance for upwash region

From the perspective of control engineers, the principles that steer birds to the V formation are:

- Coalescing: The behaviour of birds to create a flock, in which each bird displays an attractive relationship [4] with each other, causing all birds to flock together instead of flying separately.
- Gap-seeking: The tendency of birds to seek for clear views in the direction of flight is known as gap seeking. It ensures that their range of view is not impeded [4] by the location of another bird. This is necessary during bird flight to assist them maintain coordination with other birds and detect and avoid any predatory birds [48] or other adversities that may be present.
- Stationing: This is the most significant flight control mechanism for forming the formation in which the birds position themselves in the upwash region. The birds form a straight line [4], which follows a rule-based initiation stage; then two lines meet to form an inverted-V, which generates two upwash areas [4] and follows the energy-saving model in steady-flight.

The first and second rules suggest that we distinguish between two modes of behaviour, one succeeding and the other alternating. Birds must first seek the proximity of the group before being allowed a clear view. The second and third rules imply that the sensory input that maintains the bird flying toward a relative position of clear view is not simply visual, but also stems from how easily it keeps up with the group.

We can define a bird by a point, which generally correlates to the centre of buoyancy [49] of the bird in air, by modelling a simplified bird-environment system for a systematic approach to the lateral and longitudinal distances between each agent. Now, from [5], we find two hypotheses that can be found in a variety of different studies.

- **Aerodynamic Advantage Hypothesis**

  As previously stated, aerodynamic advantage is the advantage gained by upwash generated by the leading birds. A pair of trailing vortices that form around the wingtip of a bird's wings. According to a fixed-wing analysis, the distance between these vortices is $\frac{\pi}{4}b$ [5], where b is the wingspan of the bird. The distances are depicted in figure 2.8.

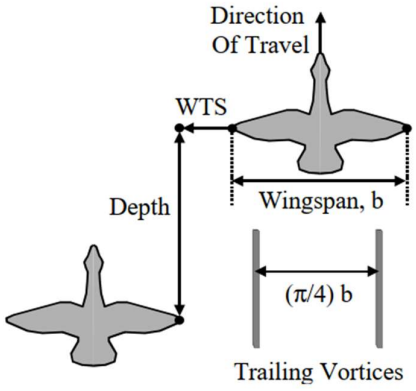

Fig. 2.8 Lateral and longitudinal distance between adjacent birds [5]

Because the vortices are spinning, the area near the wingtip has downwash, which aids the bird in creating lift [53], and the area on the outer region of the vortices has upwash [53] due to the spinning air. The latter also depletes the bird's energy [54]; so, if the wake is not exploited by other birds in the formation, this energy becomes a by-product of the lift produced and is squandered.

To obtain maximum lift from this upwash region, birds has to fly with a wingtip spacing [5] (WTS) of $WTS_{opt} = \left(\frac{\pi}{4} - 1\right) \times \frac{b}{2}$, $WTS_{opt} < 0$. The energy saved is strongly dependent on lateral position of the birds [5], so the trailing birds has to be able to accurately track WTS_{opt}. From Munk's displacement theorem [55], it is observed that the savings are more or less independent of longitudinal position [5].

- **Visual Communication Hypothesis**

As per the visual communication hypothesis, the formation shape or geometry is determined by retinal features and the placement of the eye on the head. According to Heppner [56], the positioning of the eyes limits the birds' field of vision.

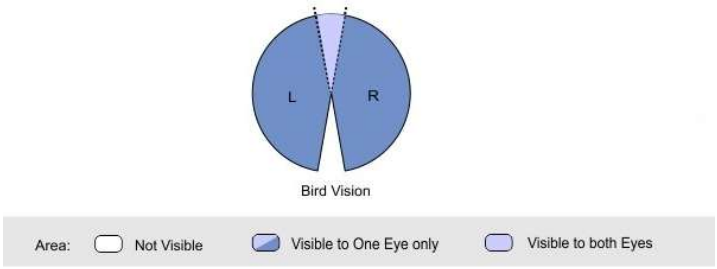

Fig. 2.9 Field of view of bird's eyes [57]

Binocular vision is only attainable on a small field of view angle, as illustrated in figure 2.9. This impairs birds' depth perception; therefore, coordinated movement with other birds helps to improve this problem; consequently, V-formation is recommended, which aids in visual communication [5] while also reducing obscured view. Improved visual communication in birds aids in determining the shortest distance to goal [4], maintaining formation [5], avoiding obstacles on the path [58], and assisting young birds [5] in learning migratory paths and roosting/feeding areas.

We investigate lateral positioning because it has been demonstrated that the longitudinal position of the birds' formation is not [5] highly dependent on energy waste minimization. From the standpoint of systems interpretation, the placement rules can be shown in block-diagram form, as in figure 2.11, with the circumstances shown in figure 2.10.

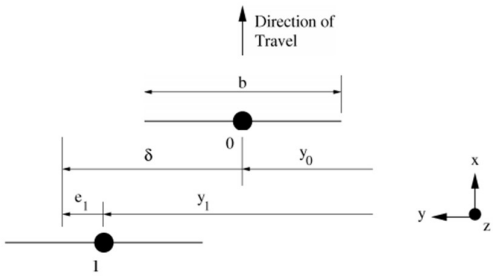

Fig. 2.10 Formation coordinates for generating system block diagram [50]

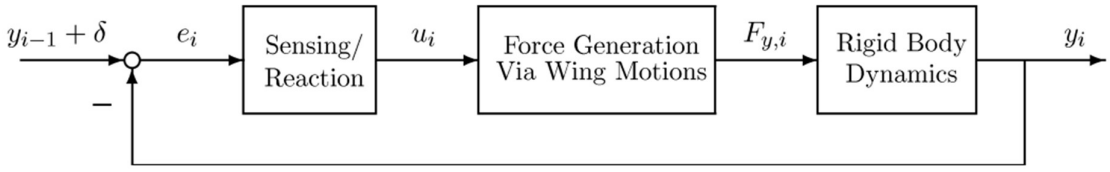

Fig. 2.11 Block diagram depicting dynamics of an individual bird [50]

The feedback path is based on the supposition [50] that the birds can detect the lateral spacing error. Assuming a bird is flying in a formation that maximizes aerodynamic advantage, which is highly dependent on lateral spacing. As a result, any deviation from optimal position means that the net lift [50] required by that specific bird's wings will be greater. Because the additional lift generation is being generated intentionally by the bird, the bird will sense [50] the need for more energy and will realize that it is no longer in the optimal lateral position.

Alternatively, the bird can see the position of the birds in front of it, flying in formation. Any changes from optimal position would mean that the relative displacement [50], [5] between each bird will change, which will be visible to the bird. As a result, the bird will attempt to fix the error and attain the optimal position.

2.4.1 Mechanism for upwash region generation

To understand the mechanism by which the upwash region is generated, we have to discuss the wing flapping motion and the movement of air generated due to it. The flight in birds can be divided into two types [53], namely, gliding and flapping.

Gliding flight is the same as [53] that in a fixed-wing aeroplane. Flapping is a modification that involves the wings flapping up and down in a rhythmic motion [45] to generate increased lift and forward velocity. Birds' wings are slightly inclined [49] to the forward passage of air, allowing them to deflect air downwards and contribute to lift. This is the wing's angle of attack.

Also, the wings of birds are slightly curved [46], which give it a typical airfoil shape, thus making air to travel faster on the top surface than on the bottom. This generates lift. Thus, we see that the bird wing can be compared [53] to a fixed wing flight, and has similar dynamics.

During bird flight, the air close beneath the wing moves downward [59], creating lift. Air moving rearward creates forward thrust [59]. The above-mentioned airflow pattern is now abruptly broken [44] at the extremities of each wing. Due to the limited wing size [60], the downward velocity of air at the wingtip produces a tangential force rather than a downward pull, resulting in a wind vortex on each of these wingtips.

Air flowing beneath the wing surface turns and rises slightly beyond the wingspan. The air is rotated by the torque created by these opposing forces. These vortices begin at the wingtips [60] and funnel outwards due to the bird's flight and downward due to the resulting downward thrust. Due to the conservation of angular momentum, this air vortex can last for two or three minutes [60] before dissipating due to the low air friction.

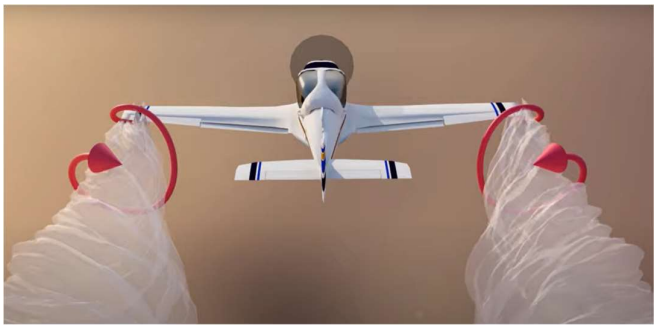

Fig. 2.12 Vortex generated at the wingtips of fixed-wing aircraft [61]

Taking one of the two vortices formed by a fixed wing flight and inspecting its geometry, we can see that air flow has a downward direction close to the wing tip, while it has an upward direction on the diagonally opposite side of the vortex. The air flow direction at the top and bottom of the vortices is parallel to the wing surface. As a result, they do not transfer [54] enough force in the sideways direction.

It is evident from this that the zone on the outer border of each vortex has air flowing upward, resulting in the upwash region. This is the upwash wake path, which weakens [60] as we get closer to the bird. The size and speed of these vortices are directly proportional [60] to the weight of the bird forming the vortices and inversely proportional [60] to the forward velocity.

2.4.2 Positioning rules for agents

From section 2.4.1, the positioning rules for each agent is derived based on the two criteria

- Obtain maximum advantage from upwash

  The upwash zone in a multi-robot system can be built in a variety of ways to imitate the vortex region created in a fluid. The horseshoe vortex model [5] with a Burnham-Hallock profile, which features a center downwash zone with upwash regions on both sides of the downwash region, is the most prevalent. This establishes the lateral gap [4] between agents in order to gain the most benefit from upwash (aerodynamic advantage).

- Provide unhindered field of view

  The birds' possible field of view is hampered [4] as a result of flying behind agents. This also applies to agents' sensing regions. As a result, agents must be arranged in such a way that they have an unobstructed field of view in the forward direction. This defines the longitudinal spacing [5], [62] between agents, allowing for sideways views that would otherwise be impeded by the presence of another agent if not optimally situated. This supports the visual communication concept.

2.5 Swarm Control Algorithms

A swarm system comprised of N number of agents, $N \epsilon N$ can have many types of swarm control algorithms which govern their actuation and behavior of the system. The control algorithm chosen is based on the system requirements, available sensors and agent configuration and system complexity requirements.

2.5.1 Leader-Follower

Leader-Follower swarm control algorithm is designed where a particular robot is chosen as the leader agent [63]. This agent moves through the region of interest according to some defined control rules. All other robots, which are known as the follower agents [63], follow the leader agent while keeping inter-agent separation. The follower agents sense the surroundings and communicate [64] with each neighbouring agent. But actuation is based on the information provided by the leader agent [64], [65]. The leader agent accepts the sensor data from each agent and decide the path [63] to be taken by the swarm.

A leader agent after path planning actuates through the decided path, while the follower agents “follow” the leader agent, while going through the resource profile. The leader agent may not

be statically defined, but dynamically allocated based on current most optimal position or any other characteristics polled between the agents of the swarm [66], [64]. In this situation, the leader may/will change during the time-frame of swarm actuation. Also, there might be more than one leader agent in a system. If for some reason the swarm has to be split [67] into two or multiple parts, each part may choose its own leader and continue, or converge together [67] to give-over leader responsibilities to one leader again when converged.

This work is based on a defined leader agent where there is no platoon splitting and the leader changes only after an agent swapping action is performed.

2.5.2 Attraction-Repulsion Controller for Flocking

Attraction-Repulsion controller is a very common swarm control algorithm where there are two, very simple control rules that act on every agents of the swarm [68]. This is commonly known as virtual potential gradient, which defines the attraction and repulsion magnitudes.

i. Attraction controller: Attraction controller is a controller that generate controller input, u_a , based on inter-agent separation. The aim of attraction controller is to bring agents close together. When an agent i is placed with many other agents j surrounding the i^{th} agent, attraction control makes sure that the i^{th} agent exerts attraction controller input on each of the j^{th} agents. This attraction controller when applied on all agents, the collective behavior of the agents is that the agents would congregate together after actuation. The magnitude of attraction input is directly proportional to the distance of separation between the agents [69]. Here p^i and p^j are the position coordinates of i^{th} and j^{th} agents, respectively [69].

$$u_a \propto \left|p^i - p^j\right|$$

ii. Repulsion controller: Repulsion controller generates controller inputs to prevent agents coming too close together. The magnitude of repulsion defined is based on the distance of agent i with any agent j, and the position of the j^{th} agent with respect to a circle around agent i known as the region of repulsion in which repulsive action is present [69].

$$u_r \propto \frac{\left|p^i - p^j\right|}{r_s}$$

Here, u_r is the repulsion controller input, p^i and p^j are the position coordinates of i^{th} and j^{th} agents, respectively and r_s is the region of repulsion for the agent under consideration. Repulsion controller defines the minimum separation between two agents, thus preventing inter-agent collision [69].

The attraction and repulsion controller together define the swarm behavior that agents would come together from any arbitrary location they are initialized in, while maintaining safe distance from each other when close together, so as to prevent inter-agent collision [70]. These two control rules make up the control behavior known as flocking. In flocking, the swarm forms a congregation, but has no defined formation to adhere to, while preventing collision between agents [66].

2.6 Chapter Summary

This chapter addresses the formation rules found in nature when birds fly great distances in V-formation. The formation requirements are explored in light of the hypotheses presented in the literature.

Second, this chapter addresses the V-formation strategy used in some studies with comparable goals. The positioning rules used in those studies are briefly addressed, and the work in this thesis is motivated by them.

CHAPTER 3

Platoon Formation and Actuation

3.1 Introduction

Actuation of multi-robot systems that retain a defined formation requires the controller of the system to generate and maintain the formation. Achieving an inverted-V shaped platoon and maintaining it, while moving through an unknown, occluded region is the main aim of this work.

This chapter focuses on the control method used in this study for platoon formation and actuation till the defined goal is reached.

3.2 Problem Statement

The objectives of this thesis work are to create an inverted-V shaped platoon with N number of robotic agents, where N $\in N$, where N is the set of natural numbers, $\{1,2,\ldots,\infty\}$. At $t = 0$, the agents are in arbitrary positions. The desired behaviour of the system is to form and maintain an inverted-V shaped platoon during navigation towards the target while avoiding barriers of different shapes and sizes. Here, we have only considered obstacles that remain stationary for the entire time duration of actuation.

Assuming that the platoon's intended destination is known, the first difficulty is that the robots are unaware of the barrier placements. As a result, we must devise a method to sense the region through which the platoon moves. This will make it easier for the controller to decide on a safe path for agents to gather and form the platoon.

Following that, we must form the platoon shape in the sensed safe region. The platoon must have an inverted-V shape. As a result, we make a triangle within the safe region, with two sides forming the required inverted-V shape.

However, using this virtual triangle creation approach [71], we can see that in order to orient the tip of the inverted-V shape towards the platoon's direction of motion, we must first orient the triangle. To begin, the controller determines the intended position for each agent,

guaranteeing proper orientation of the inverted-V form. The controller then pushes each agent from their current position to the target position while avoiding impediments.

The sensing method mentioned earlier handles obstacle avoidance. Path planning and actuation efforts assist the platoon reach the target by iterating through these processes.

3.3 Hierarchical Control Block Diagram

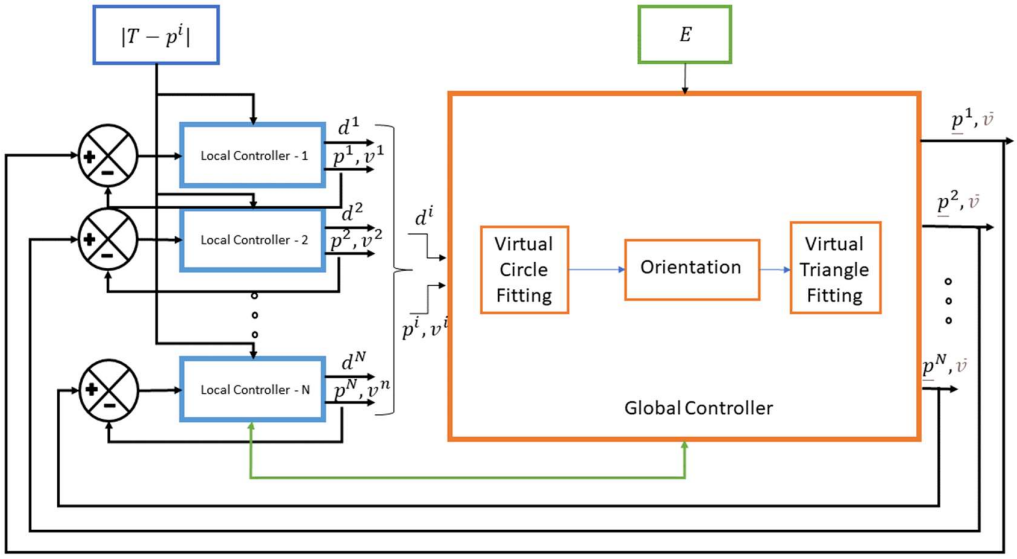

Fig. 3.1 Hierarchical block diagram summarising the working simulation

The local controller (shown in blue of Fig. 3.1) runs on each of the agents 1 through N (N ∈ N), however, the global controller (illustrated in orange) operates on the agent, called henceforth as the leader, which occupies the vertex of the inverted-V shaped structure. p^i represents the position of agents at the start of each iteration. Each of these agents has a local controller operating. As a result, the local controller of every agent knows its respective beginning location, $p^i \; \forall \; i = 1, 2, 3, \dots,$ N, N ∈ N. The desired position, that the agents should occupy, is provided by the global controller to each local controller. The latter calculates the distance between the present position to the desired position, which is the error in position for that particular agent under consideration. Now, obstacles around each agent is detected by its local controller. Distances of these barriers from the agent which detected it, d^i, are sent by the

local controller on that agent to the global controller, running on the leader. Similarly, all local controllers provide its sensing data to the global controller.

These values are used by the global controller for locating a circular zone with no obstructions. Within that region, it fits a virtual triangle after determining the right orientation to point the triangle in. The two sides of this triangle form the required inverted-V configuration, hence the desired coordinates $\underline{p}^i$, that the agents must achieve are determined from this triangle. The current velocity of each agent is used to compute the average velocity $\overline{v}$. These values are given back to the local controller, which computes the difference between the current and desired position and velocity values. This difference in position and velocity is the error term for the local controllers, which are minimized during each iteration, thereby actuating the agents.

At each iteration step, the total power, E, available in the leading agent, is tallied against a threshold value that we set as the critical level, after which agent's position swapping should occur. When it falls below the defined threshold, an agent swapping operation is performed. The follower agent that comes up as the new leader agent handles the global controller, while the previous leader agent is relieved of its duties to run the global controller and only handles the local controller, after the swapping operation is completed.

3.4 Local Controller: Formation Creation

The hierarchical control as discussed earlier is adopted in this work to make the actuation control of individual agents simpler. The local controller acts on each agent and considers the local attraction and repulsion controller for agent actuation.

At the beginning, we assume a team of N robotic agents in an unknown 2-dimensional workspace $\Omega \epsilon \Re^2$ are approaching towards a predefined target location $T \epsilon \Re^2$. The motion dynamics of the i^{th} agent is decided by the current position coordinates $p^i \epsilon \Re^2$ [34].

$$\dot{p}^i = u^i \qquad (3.1)$$

where, u^i = Local attraction controller input on agent i + Local repulsion controller input on agent i.

3.4.1 Attraction Controller

The local attraction controller controls each individual agent to generate actuation controller inputs so that the agents come together from their initial arbitrary position to a predefined inverted-V formation and maintain the formation till the target coordinate is reached [34].

$$u_a^i = k_a^i \left(p^i(t) - \underline{p}^i(t) \right) \qquad (3.2)$$

where k_a^i is the attraction gain for i^{th} agent, p^i the present coordinate the agent and $\underline{p}^i$ the desired position for the agent where it should reach. When actuation starts, $p^i(t)$ is the position coordinate on that instant of time. $\underline{p}^i(t)$ is the desired position where the i^{th} agent is desired to occupy [34], [69].

The attraction controller on the local loop acts on each agent independently of the resource profile through which the agents must transit. The controller effort for an attraction controller is proportional to the position difference between an agent's present location $p^i(t)$ and its desired position $\underline{p}^i(t)$. According to eq. 3.2, as this error term increases, so does the magnitude of u_a^i.

As a result, we can conclude, that the magnitude of attraction force on each agent grows as the error in position increases. Figure 3.2 illustrates this.

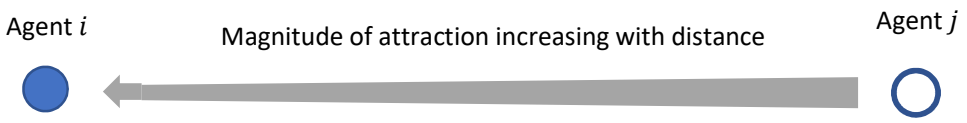

Fig. 3.2 Visualization of attractive interaction between two agents

The block diagram in figure 3.3 can be used to describe the operation of the repulsion controller.

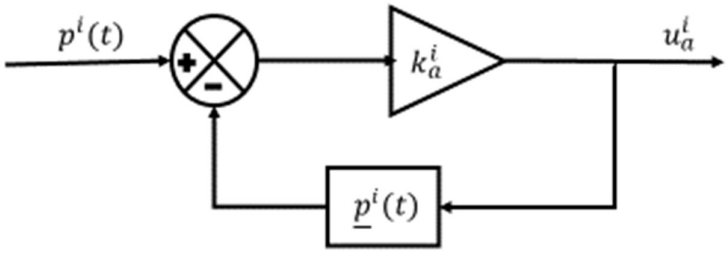

Fig. 3.3 Block diagram representation of attraction controller

3.4.2 Repulsion Controller

The repulsion controller is a part of the local controller in this work. The aim of this controller is to prevent inter-agent collision during actuation. It applies a repulsive force on any agent that comes within a circle around an agent, the radius of which is defined by r_s^i. The direction of the force applied is radially outwards from the i^{th} agent, and is applied on the robot that comes within the region of repulsion of the former [34].

$$u_r^i = k_r^i \, exp\left(\frac{-\frac{1}{2}\left|\left|p^i(t) - p^j(t)\right|\right|^2}{{r_s^i}^2}\right) \tag{3.3}$$

where k_r^i is the repulsion gain of the i^{th} agent, p^j is position of any neighbouring agent around position of agent p^i and r_s^i is the radius of region of repulsion of the i^{th} agent, which is a circular region with radius r_s^i around the i^{th} agent within which the repulsion controller takes precedence over the attraction controller [69], [70], [34].

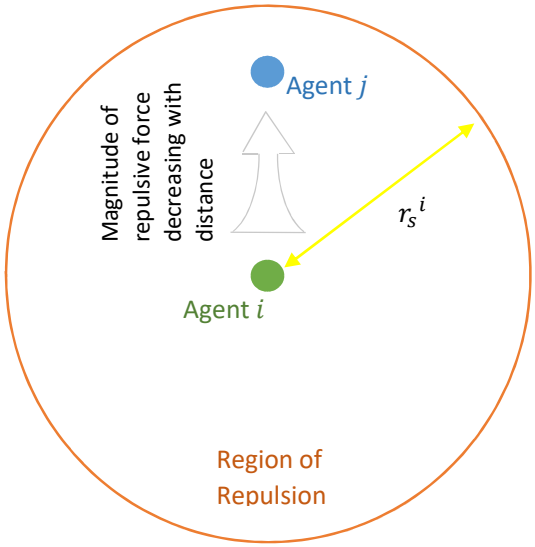

Fig. 3.4 Visualization of repulsive interaction between agents

Figure 3.4 depicts two agents, designated by i and j. The agent indicated by green circle is the i^{th} agent, while the orange ring represents its repulsion region. As indicated in the image, the agent shown in blue circle, which is designated as the j^{th} agent, is within the repulsion region of the i^{th} agent. The grey arrow represents the repulsive force imparted on the j^{th} agent by the

i^{th} agent. The repulsive interaction increases as the distance between the two agents decreases, and vice versa.

The reason for this behaviour can be explained from equation 3.3. When j^{th} agent is outside the region of repulsion of the i^{th} agent, then $\left|p^i - p^j\right| > r_s^i$. Thus, u_r^i becomes $k\exp(-c_1)$, where $c_1 \geq 0.5$, equal to 0.5 only when j^{th} agent on the boundary of the repulsion region of i^{th} agent. When j^{th} agent is within the region of repulsion, $\left|p^i - p^j\right| < r_s^i$. Then, u_r^i becomes $k\exp(-c_2)$, where $0 < c_2 < 0.5$. From the nature of exponential function, we can then validate that as $\left|p^i - p^j\right|$ decreases, the magnitude of u_r^i increases. This creates the repulsive force [34].

3.5 Global Controller: Sensing Action

Every agent in the platoon is equipped with sensors, with which they scan the environment for presence of obstacles. They are outfitted [34] with SONAR sensors separated by z radians, such that each agent covers 2π radians. This ensures that all agents can scan thoroughly around themselves.

Every agent's local controller gathers information about the distance of observed impediments from itself. The local controllers provide the global controller with the position information about barriers that they collect. It is then utilised to locate an area free of obstacles. That location is eventually used for platoon formation. The controller dynamics are constructed with the following assumptions in mind:

Assumption 1: Considering all the agents are identical in nature, presume that the associated attraction-gain, repulsion-gain and the region-of-repulsion are similar [34] $\forall i \in 1,2,\dots,N$, which can be defined as: $k_a^i = k_a$, $k_r^i = k_r$ and $r_s^i = r_s$.

Assumption 2: Each agent in the group is equipped with m number of distance sensors (such as Lidar [127] or Sonar [126]) spaced at a certain (z) radian [34] apart.

Assumption 3: The agents can be approximated by a point coordinate, where the physical dimensions of the agent may be such that the centre of mass of the agent coincides with the point considered for simulation, and the agent is no bigger than the repulsion region [34] r_s defined for the agent.

For a successful path-planning, each agent needs to sense the nearby obstacles efficiently. The following section will describe the overall sensing mechanism followed by the virtual boundary fitting strategy within which the inverted-V formation is to be placed.

3.5.1 Sensing resource profile

The sensing algorithm employed in this work is used to detect the presence of obstacles in the resource profile. We have assumed that the obstacles remain stationary throughout the duration of the simulation. Since all the agents are furnished with the distance sensors (assumption 2), at the t^{th} instance, the sensed obstacle's distance $\boldsymbol{d}^i(t)$ around the i^{th} agent will be [34]

$$\boldsymbol{d}^i(t) = \left\{\boldsymbol{d}^i(t) \middle| d_{thre+} \geq \boldsymbol{d}^i(t) \geq d_{thre-}\right\} \tag{3.4}$$

where $\boldsymbol{d}^i(t) = [d_1^i \; d_2^i \; \ldots \; d_m^i]_t$ is the distance-matrix of the m sensors, d_{thre+} and d_{thre-} are the sensor specific threshold limits such that $d_{thre-} > 2r_s$. From this information, the predicted obstacles' position $\widetilde{\boldsymbol{R}}^i(t) = \left[\widetilde{\boldsymbol{R}}_x^i(t) \quad \widetilde{\boldsymbol{R}}_y^i(t)\right]^T$ with respect to the global coordinate frame of reference, will be [34]

$$\begin{bmatrix}\widetilde{\boldsymbol{R}}_x^i(t)\\ \widetilde{\boldsymbol{R}}_y^i(t)\end{bmatrix} = \begin{bmatrix}p_x^i(t)\\ p_y^i(t)\end{bmatrix} + \sum_{j=1}^{m} d_j^i(t) \times \begin{bmatrix}cos\left[(j-1)z + \theta^i(t)\right]\\ sin\left[(j-1)z + \theta^i(t)\right]\end{bmatrix} \tag{3.5}$$

where $\left[p_x^i(t) \quad p_y^i(t)\right]$ is the present location [32] of the i^{th} agent and $\theta^i(t)$ is the orientation of the agent with respect to the global coordinate frame of reference at the t^{th} instance. In a specific situation, it may happen that the i^{th} agent's field of view is obstructed by the j^{th} neighbouring agent, such that $\left\|p^i - p^j\right\| > 2r_s$ and that may cause a sensing error. To avoid this scenario, the coordinates of the detected obstacles will be updated according to the rule [32]

$$\widetilde{\boldsymbol{R}}_{(x,y)}^i(t) = \left\{\widetilde{\boldsymbol{R}}_{(x,y)}^i(t) | \widetilde{\boldsymbol{R}}_{(x,y)}^i(t) \neq p^j(t)\right\} \quad \forall j \epsilon N_i \tag{3.6}$$

Therefore $\forall i \; \epsilon \; [1,2,\ldots,N]$, the predicted obstacles' locations [34] $\widetilde{\boldsymbol{R}}(t) = \left[\widetilde{\boldsymbol{R}}_x(t) \quad \widetilde{\boldsymbol{R}}_y(t)\right]^T$ will be

$$\widetilde{\boldsymbol{R}}_{(x,y)}(t) = \left[\widetilde{\boldsymbol{R}}_{(x,y)}^1(t) \quad \widetilde{\boldsymbol{R}}_{(x,y)}^2(t) \quad \cdots \quad \widetilde{\boldsymbol{R}}_{(x,y)}^N(t)\right] \tag{3.7}$$

will provide the coordinate information of nearby sensed obstacles (within the range of (d_{thre+}, d_{thre-}) [34] from the present location of the agents) with respect to the global

coordinate frame of reference. After sensing our next aim is to fit in an arbitrary-boundary from the detected obstacles' coordinates. For this work, the shape of the boundary is chosen as a circle.

3.5.2 Virtual Circle fitting within the sensed region

The coordinate information of nearby sensed obstacles is obtained [34] from $\widetilde{\boldsymbol{R}}_{(x,y)}(t)$, as mentioned in section 3.5.1. The global controller can use this information to determine the position of obstacles in the agents' vicinity. This data is used to find a region within the sensed area that has no obstacles.

This selected region is sufficient to fit a virtual circle with no obstacles within or on the circle's periphery. The platoon size determines the radius of this virtual circle, which is discussed in Section 3.5.3.2.

Because there are no sensors in the simulated environment, we have used a slightly modified way of finding safe region within the resource profile, through which the platoon should actuate. The shape of platoon required for this work is an inverted-V, which, geometrically, we use a virtual triangle. Now, this triangle should be placed in such a way in the resource profile, that no obstacles are present that can collide with this triangle.

For this reason, we have fit a virtual circle before fitting the triangle inside the circle, making sure that no obstacles lie inside or on the circumference of the virtual circle. The virtual circle approach is beneficial, because it gives an area within the region of interest without any obstacles, within which we can fit the virtual triangle. The circle shape is chosen, because it ensures a safe region for any orientation of the triangle, that might be required, for the platoon to orient in the direction of motion.

If we did not choose a circular region, then we had to determine the orientation required for each iteration of the platoon first, before ensuring that the agents occupying the vertices of the triangle do not collide with an obstacle. The virtual circle approach allowed us to freely orient the platoon, as needed.

3.5.3 Virtual Triangle fitting within sensed virtual circle

The shape of the inverted-V formation is defined by the virtual triangle that fits within the virtual circle. The angle between the arms of the inverted-V shape must be chosen, such that

we can fit maximum number of agents within the platoon formation while still benefiting from flying within the upwash zone. The inverted-V shape generated within the virtual circle is seen in Figure 3.5. The geometry of inverted-V shape is displayed in blue, while the virtual triangle and circle are shown in black.

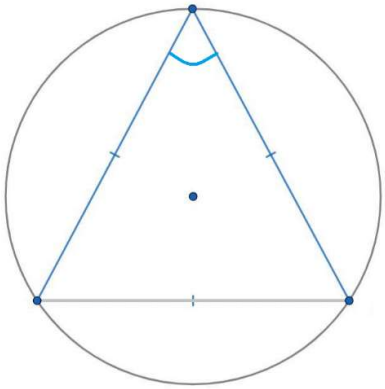

Fig. 3.5 Inverted-V formation formed from inscribed triangle

The minimum inter-agent spacing must be computed in order to determine the maximum number of agents that may fit in a particular triangle. This is done to ensure that the local repulsion controller can avoid inter-agent collisions. The inter-agent spacing sets the desired angle between arms of the inverted-V shape. This offers an upper limit to the total number of agents that can be accommodated in a specific triangle within the sensing circle while preserving the formation.

Finally, we want the triangle to fit within the circle, with its vertex always facing the direction of actuation. Because the virtual triangle that is fit determines the shape and position of the inverted-V shape on the resource profile. For each iteration, the triangle is suitably aligned while fitting within the circle.

3.5.3.1 Determining vertex-angle of the virtual triangle

The angle formed by the two arms of the inverted-V is selected such that we can include a large number of agents while creating platoon formation. This must be done while ensuring that enough inter-agent distance is maintained, and the formation will not get distorted or skewed.

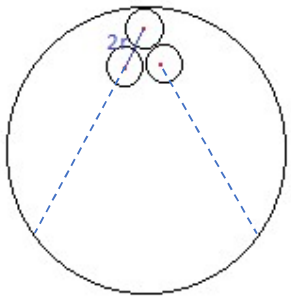

Fig. 3.6 Three agents occupying the vertex of inverted-V formation

Figure 3.6 shows three agents occupying the vertex of the inverted-V formation, which lies within a virtual circle. Each agent has a region of repulsion around itself, which is a circle of radius r. From section 3.4.2, we know that agents cannot be placed within the region of repulsion of another agent.

This means, that the condition for minimum separation between agents would result in agents being placed at a distance twice the radius of region of repulsion. Considering the dimension for region of repulsion is identical for all agents, the separation between two adjacent agents would be twice the radius of region of repulsion, which is $2r$ in this case. This can be visualized from the figure 3.6.

Figure 3.7 shows the situation when three agents, forming the tip of the inverted-V formation, are placed such that there is minimum distance of separation between agents. The regions of repulsion for each agent are shown by grey circles. The circumferences of these circles are touching, which is the least distance of separation, before repulsive force comes into effect.

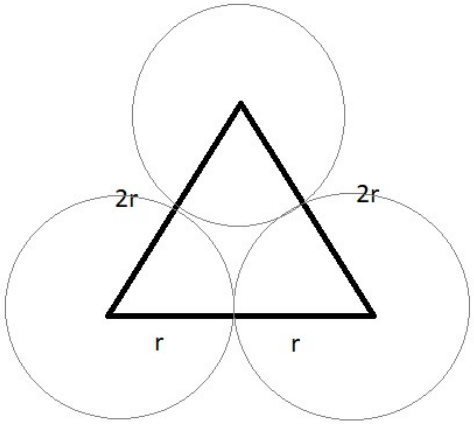

Fig. 3.7 Magnified view of region of repulsion of leader agent and two follower agents when there is least possible separation between agents

As we can see from the above figure, the triangle forming from the radii of the circles denoting the repulsion region of the 3 agents at the center of the circles, becomes an equilateral triangle. This is because, each agent's region of repulsion has radius r. The triangle forming from the distance of separation of these three agents has sides $2r$ each. Now, inner angle of each vertices of an equilateral triangle is $60°$.

Thus, we can conclude that the minimum angle of the vertex of the triangle has to be 60 °. In other words, the triangle that has to be fit within the sensed circle has to be an equilateral triangle.

3.5.3.2 Calculating inter-agent distance

Without considering the feasibility of N agents to fit in the equilateral triangle determined, total number of agents in either one of the arms of the inverted-V formation, including the leader agent is as follows:

Let the radius of the sensed circle is r. So, the largest equilateral triangle that can be inscribed in the circle has side of length $s = r\sqrt{3}$, which is derived in section 3.5.4.2.

Case I: N is odd

Total number of agents on any one arm of the inverted-V shape along with the leader agent is

$$n_a = \lceil N/2 \rceil \tag{3.8}$$

where, $\lceil x \rceil = \min(a \in Z\,;\, a \geq x)\,, x \in \mathbb{R}$

Case II: N is even

For this condition, one both arms of the inverted-V is filled with agents equally until all the agents are filled and one arm is short of just one agent to be equal to the other arm.

Total number of agents in this case for the arm that is completely filled can be shown as

$$n_{a1} = \frac{N}{2} + 1$$
$$n_{a2} = \frac{N}{2} \tag{3.9}$$

since the other arm will be short of one agent.

It has to be noted here that the agents in one arm of the inverted-V platoon formation here includes the leader agent too for each of the above options. So, total number of agents will be $n_a \times 2 - 1$ or $n_{a1} + n_{a2} - 1$.

Here, we defined the total number of agents in a single arm of the inverted-V by n_a. Thus, the inter-agent distance between the agents will be

$$d = \frac{r\sqrt{3}}{(n_a - 1)} \tag{3.10}$$

It should be noted here, that in this work, when N is even, the distance between each agent is calculated by assuming if N = $N_{original}$+1 and letting any one of the arms of the inverted-V to have one agent less than on the other arm, to accommodate for the even number of agents in the platoon. Here, $N_{original}$ is the original number of agents initialised for the given case. Detailed explanation for equation 3.10 is given in section 3.5.4.2.

3.5.3.3 Determining maximum number of agents within the virtual triangle

From section 3.4.3.2, we obtained the expression for inter-agent separation for a desired platoon size. Now, radius of repulsion region, defined as ${r_s}^i$ for the i^{th} agent is the radius of closest approach for each agent. So, we can calculate the minimum distance of separation between agents.

$$d_{min} = 2{r_s}^i \tag{3.11}$$

Now, for all agents to fit into the triangle, the determining condition will be

$$d \geq d_{min} \tag{3.12}$$

This determines the maximum number of agents that may be filled in a particular triangle without the agents straying from the straight line represented by the triangle's sides. This is solved by ensuring proper inter-agent spacing.

3.5.3.4 Navigating the virtual triangle towards target

To accomplish biomimicry of birds, the orientation of the inverted-V formation must be corrected such that its vertex is always points towards the direction of actuation. Any change in direction of actuation should cause the platoon to immediately reorient.

The virtual triangle is fitted within a circle, as detailed in section 3.5.3. The circle is the region without any hurdles where the formation would begin to assemble on the two sides of an inscribing triangle inside the circle during iteration.

From section 3.5.2, we see that the global controller defines a safe region for agent actuation, in which a virtual circle is fit. Since the parameters for this circle is set by the global controller, so, we have information about both its radius and the coordinate C of the circle's centre, as illustrated in figure 3.8.

A line drawn connecting the global origin O and the centre of the circle C intersects the circle at points A and B. As shown in the diagram, calculating the distance AT and BT, where T is the target position, reveals that AT is less than BT. Taking the point with least distance from the target, we can choose the vertex coordinate for the inverted-V shape during first iteration.

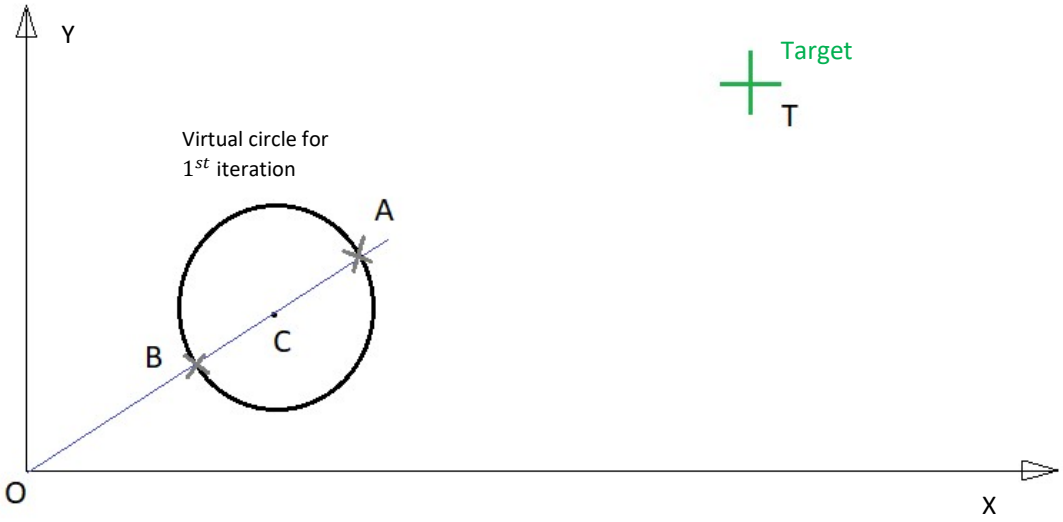

Fig. 3.8 Visualization of sensed circle being placed in resource profile for first iteration step

So, the equilateral triangle inscribing the given circle is placed with its vertex at point A. The remaining two coordinates are known since the triangle is equilateral and is enclosed by a circle. As a result, the three coordinates where the triangle intersects the circle become the initial three coordinates of the required inverted-V configuration.

At this stage, we have one virtual circle, in which our platoon has already formed. For the next iteration, the global controller fits a new virtual circle, within which the platoon has to actuate. We have illustrated this on figure 3.9. The black-dashed circle is the circle which was fit in the previous $(n-1)^{th}$ iteration, while the grey circle is for the n^{th} iteration.

Again, the parameters for both of these circles are determined by the global controller, and hence we have the coordinates C ` and $C$, which are the centres of the previous and present virtual circles, respectively. A line segment drawn through $C$ ` and C intersects the grey circle at points A and B.

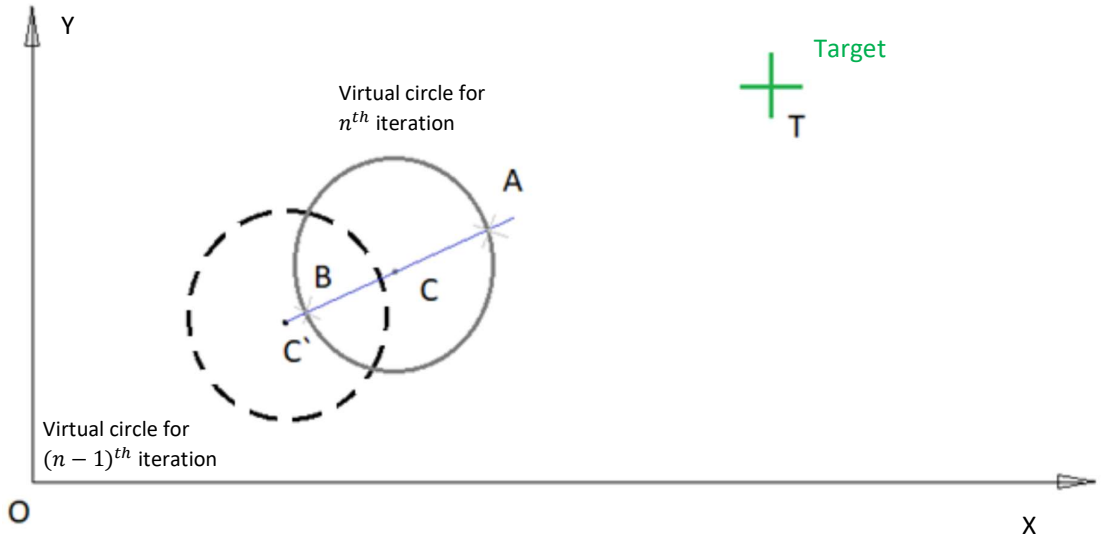

Fig. 3.9 Visualization of sensed circle being placed in resource profile for any iteration step except the first

Now, we determine which of the two points is closest to the target T. As in the previous case, we calculate the distance between the points AT and BT. Here, we can see that distance AT is smaller than BT. Thus, the coordinate for vertex of the inverted-V formation is chosen as A.

This ensures that the leader agent is always placed in the forward direction of actuation throughout the path planning and actuation steps of the inverted-V shaped platoon. While the follower agents occupy the correct position indices ensuring correct formation shape and orientation until the platoon reaches target coordinate T.

3.5.4 Pushing agents on the two sides of the virtual triangle

After determining the triangle coordinates with exact dimension and orientation, the agents are pushed on the two sides of the virtual triangle to form the platoon formation. In section 3.5.3.4, we obtained the coordinate of the vertex of inverted-V formation.

From section 3.5.3.1, we determined that the angle between the arms of the inverted-V shape would be 60°. So, we can obtain the coordinates of the other two vertices of the virtual triangle

from the information that we already have one vertex, and the triangle would be equilateral, which is circumscribed by the virtual circle.

3.5.4.1 Agent indexing and placing leader agent on virtual triangle vertex

Before placing agents in the virtual triangle, N number of agents are indexed from 1 to N, where N $\in N$. The global controller decides which agent should occupy which coordinate during each iteration. For this, the global controller has to send the desired position for each agent to their appropriate local controller. This is facilitated by agent indexing, which is maintained for the entire duration of actuation. It is also required during swapping, which is discussed later.

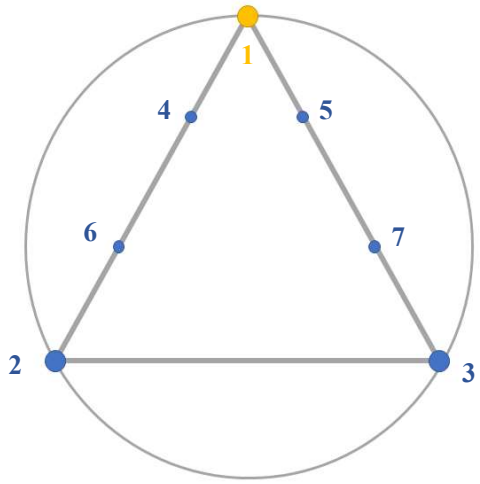

Fig. 3.10 Visualization of circumscribed virtual circle and virtual triangle with 7 agents

The leader agent has an index of one and is positioned at the vertex of the virtual triangle (denoted by orange circle, as shown in figure 3.10). The agents farthest from the vertex of the inverted-V shape are assigned index 2 and 3, respectively. These three agents are given their chosen coordinates to occupy the virtual triangle's corners.

3.5.4.2 Filling agents on both sides of inverted-V arms

Three agents are assigned their index and position in section 3.4.4.1. The remaining (N-3) agents are now to be filled on both sides of the inverted-V arms. The number of agents to be filled on each arm is first established. Given that the available number of agents is less than or

equal to the maximum number of agents that can be filled in the virtual triangle, the distance between each agent for uniform spacing between agents is computed as

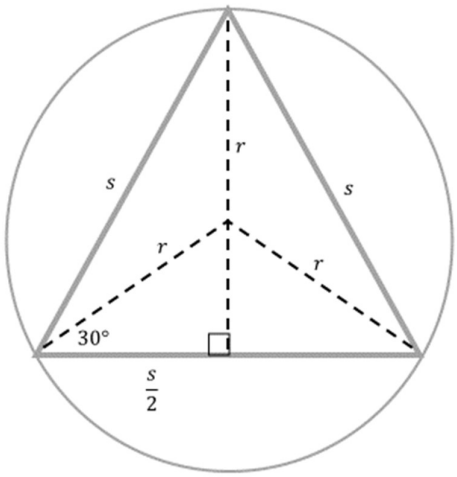

Fig. 3.11 Dimension of sides of virtual triangle

From figure 3.11, we can see

$$r\cos 30° = \frac{s}{2}$$

$$\Rightarrow 2r \times \frac{\sqrt{3}}{2} = s$$

$$\Rightarrow s = r\sqrt{3} \tag{3.13}$$

when N is odd:

Considering any one arm of the inverted-V formation, including the leader agent, the total number of agents in that arm is $\frac{N+1}{2}$. So, the total number of gaps between agents in that arm is $\frac{N-1}{2}$.

Thus, distance between agents would be

$$d = \frac{s}{\left(\frac{N-1}{2}\right)}$$

$$\Rightarrow d = \frac{r\sqrt{3}}{\left(\frac{N-1}{2}\right)} \tag{3.14}$$

When N is even:

Considering any one arm of the inverted-V formation, including the leader agent, the total number of agents in that arm is $\frac{N+2}{2}$ (considering total number of agents as (N+1) and filing agents unevenly on the arms). So, the total number of gaps between agents in that arm is $\frac{N}{2}$.

Thus, distance between agents would be

$$d = \frac{s}{\left(\frac{N}{2}\right)}$$

$$\Rightarrow d = \frac{r\sqrt{3}}{\left(\frac{N}{2}\right)} \tag{3.15}$$

where r is the radius of the virtual circle that is fit in the sensed region with no obstacle and in the forward direction of actuation, and N is the total number of agents in the platoon.

3.5.4.2.1 Condition when number of agents is odd

When the total number of agents is odd, the agents are divided equally between the two arms of the inverted-V shape. From section 3.5.3.4 and 3.5.4, we have obtained the coordinates (x_1, y_1), (x_2, y_2) and (x_3, y_3), as illustrated in figure 3.12. Also, the magnitude for d is obtained from equation 3.14.

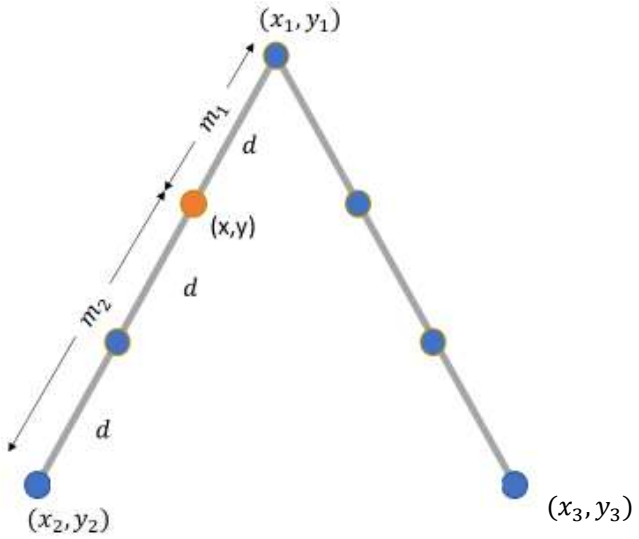

Fig. 3.12 Agent distribution in platoon when total number of agents is odd

Now, to obtain the coordinates of any of the other agents, we can use equation 3.16.

$$(x, y) = \frac{m_1 x_2 + m_2 x_1}{(m_1 + m_2)}, \frac{m_1 y_2 + m_2 y_1}{(m_1 + m_2)} \quad (3.16)$$

where (x_1, y_1) and (x_2, y_2) are the coordinates of the two ends of the line segment describing one of the arm of the inverted-V formation, and m_1, m_2 are the distances of separation from (x, y) to (x_1, y_1) and (x, y) to (x_2, y_2) respectively.

For this example, we have taken a situation where we have seven agents, and we are calculating the coordinates of the agent depicted in orange. m_1 in this case is d and m_2 is $2d$. Solving the equation, we obtain the coordinates of the agent under consideration. For the other agents, we change the values of m_1 and m_2 and perform the same calculations.

3.5.4.2.2 Condition when number of agents is even

When the total number of agents is even, we cannot distribute equal number of agents on the arms of the inverted-V shape. Since we cannot reject an agent that is already being initialised, we have set the positioning rules to assume the total number of agents as N+1 when N is even.

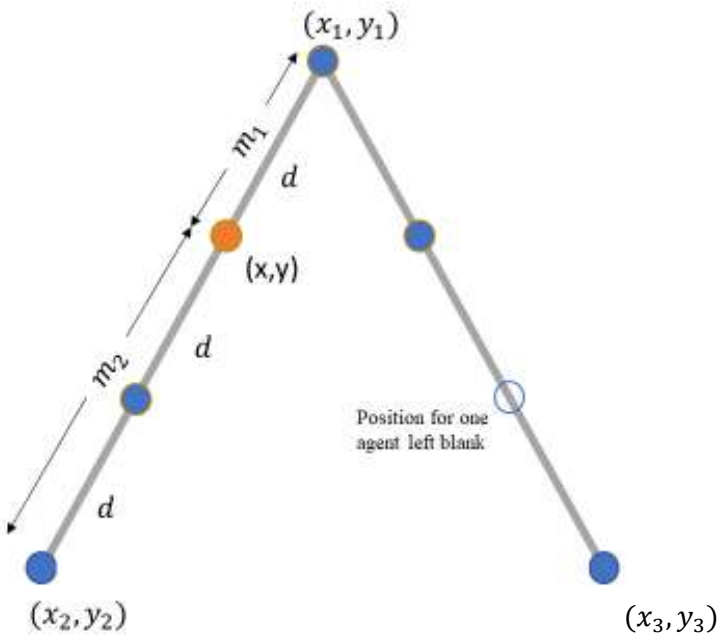

Fig. 3.13 Agent distribution in platoon when total number of agents is even

As shown in figure 3.13, the agents are filled according to the indexing rules discussed in section 3.5.4.1, until we arrive at the last agent. Since this agent is absent, and was only considered for ease of calculations, so, this position is left unoccupied, which is shown here by an unfilled blue circle.

The coordinates for the agents are obtained in an identical manner, as described in section 3.5.4.2.1 using equation 3.16 and separation between agents is computed using equation 3.15.

3.5.5 Agent actuation during each iteration step

Agent actuation is dependent on position coordinates calculated by the global controller, which contributes to position and velocity inaccuracy. The local controller, on the other hand, is tasked with sensing the agents' immediate surroundings.

This is required for safe and efficient path planning. As a result, the hierarchical controller collaborates to ensure that each agent is properly actuated.

3.5.5.1 Calculating desired position and velocity of each agent

The desired position of each agent $\underline{p^i}$ is calculated as described in section 3.4.4.2. The local controller calculates the error in position for each agent and generates an error matrix e^i. This is a matrix containing distances of each agent with assigned index $I \in 1,2,$ ...N, where N $\in N$, from their current positions to the positions determined by the global controller, where each agent should occupy in the next iteration.

At the beginning of the iteration steps, we assign a velocity vector for each agent $v^i = 0$ for the current iteration. This is because, at the beginning, all agents are stationary. Taking discrete time t_{step}, we perform the following:

i. Calculate the position error of each agent from the goal or target coordinate T

$$err_{goal} = p^i - T \tag{3.16}$$

ii. Now, for actuation, the present coordinates of each agent i has to be updated with a new value which aims to reduce the magnitude of the error in position and velocity. v^i multiplied by the discrete time step chosen for each iteration gives magnitude of displacement vector that i^{th} agent has to go through to move forward towards its desired coordinate.

iii. The above steps are iterated for each step, updating the value of position and velocity with new value in each step.

$$err_{pos} = p^i - \underline{p^i} \tag{3.17}$$
$$err_{vel} = v^i - \bar{v}$$

iv. The velocity error is calculated as the deviation from the mean velocity because relative velocity of each agents should be zero, otherwise the platoon cannot maintain formation.

During iteration, if any agent falls behind or temporarily break formation, then error in position for that particular agent increases. To bring the agent back into the platoon, velocity of that robot is increased, which helps it to actuate faster and reach its correct position in the platoon. However, when agents are in formation, their velocity will not diverge from the mean velocity of all the agents, preventing formation from breaking during actuation.

v. The position and velocity for the next step is calculated using Euler's method

$$P_{new} = P_{old} + v^i \times T_{step}$$
$$v^i_{new} = v^i_{old} + \frac{u^i}{m^i} \times k \times T_{step} \quad (3.19)$$

where, P is the matrix containing position values of each agents, v^i is the velocity of agent i, u^i is the input from the local controller, the magnitude of which is defined by the combined effort of attraction and repulsion forces. m^i is the mass of the i^{th} agent.

3.5.5.2 Agent swapping

Agent swapping is the exchange of two platoon agents' location coordinates. Agent swapping is undertaken, in the context of this work, between the leader and the final agent from any one arm of the inverted-V configuration.

Birds, when flying in an inverted-V formation occasionally exchange positions within the formation [40]. The reason for this is found to be related to aerodynamic advantage [44], [72]. The leader bird experiences highest air-drag [44], thus gets tired quicker than the other birds. To prevent fatigue, it swaps position with the last bird in the formation at regular intervals, which has maximum advantage from lift generated by other birds.

For the same reason, this work employs agent swapping. The leader agent is running two controllers, local and global, while the other agents are just running the local controller. This signifies that the leader agent's power drain is greater than that of the other agents. This is a concern because the platoon system requires a global controller to work. When the power is totally depleted, the entire system comes to a halt. Agent switching is performed to prevent this, and the global controller is moved to the new leader agent.

This is achieved by swapping agent indices based on the amount of energy drain. Since energy drain is a function of time, so, after a certain time, $agent_1$ will have much lower remaining energy than $agent_N$. Desired position coordinates of any agent i is defined as $\underline{p}^i$, where matrix $\underline{p}$ has the matrix index as the agent index and desired coordinate values as matrix values.

Now, when we perform

$$\begin{aligned} tmp &= \underline{p}^i \\ \underline{p}^i &= \underline{p}^N \\ \underline{p}^N &= tmp \end{aligned} \tag{3.21}$$

The initial and last values of the required position matrix are then swapped. The local attraction controller now decreases the position error introduced by the current position, completing agent swapping.

Though aerodynamic advantage is the major reason for agent swapping in a V-formation multi-robot system, in this work, due to lack of a modelled fluid drag in the environment model, there is no resistance to the motion of the agents. For this work, we take the energy usage required for the agents to be a part of the platoon. The energy required is in two major areas

- Energy usage during actuation
- Energy usage for running the controllers

Since actuation in this work is unhindered by any resistive forces, energy requirement for all agents will be same, and is a function of time. Based on the previous assumption that all agents are identical in their capabilities, and can take any other agent's place, we can say that the total energy available for each agent in the form of individual agent's power supply is same.

Let us assume this as E. Now, let us take the energy required for running the local controller as e_1/s while energy required running the global controller as e_2/s. After a time t_o, taking E, $e_1, e_2 > 0; E > e_1, E > e_2; t_o > 0$,

Energy available in leader agent

$$E_{avail}^{leader} = E - (e_1 + e_2)t_o \tag{3.22}$$

Energy available in any follower agent

$$E_{avail}^{follower} = E - e_1 t_o \tag{3.23}$$

Now, subtracting $E_{avail}^{follower}$ from E_{avail}^{leader}, we get

$$E - e_1 t_o - e_2 t_o - E + e_1 t_o = -e_2 t_o \tag{3.24}$$

Since e_2 and t_o are both positive, and energy drain is taken as negative, we can conclude that the leader agent has higher energy drain at t_o by $e_2 t_o$ units, and that the energy depletion increases over time. This would cause the leader agent to lose energy faster and fail, rendering the entire system unworkable because the global controller would no longer be able to function.

The control algorithm is designed such that if

$$E_{avail}^{leader} < E_{thres}$$

The leader agent then swaps positions with one of the extreme-end agents on the inverted-V shape's arms. When the swapping is complete, the former agent's global controller stops operating and the new leader agent begins running the global controller. During iterations, the local controller remains active on all agents. Thus, the position and role of leader and follower are swapped between the agents engaging in the swapping.

3.6 Chapter Summary

This chapter discusses the hierarchical controller operation and method of implementation of controllers in this work to achieve inverted-V platoon formation, sensing, path planning, actuation to a predefined goal coordinate T, while avoiding obstacles and minimising energy depletion while maintaining formation.

This chapter's main objective is to demonstrate the contribution of this work towards forming a platoon of definite shape, using a collection of homogeneous robots. The algorithms used in this work are presented in a systematic manner, from platoon formation to reaching the goal.

CHAPTER 4

Simulation Results

4.1 Introduction

This chapter contains a compilation of all simulation findings achieved in this effort. Starting from arbitrary points, a set of robots independently achieves and maintains an inverted-V shaped arrangement. We have designed different environments with various forms of obstacles that this platoon must avoid in order to reach a pre-set destination. The simulation results as well as time comparisons obtained from these test cases are provided and discussed in this chapter.

The complete simulation is run in MATLAB and evaluated on two separate operating systems. The simulation times in this paper are for MATLAB R2018a running on a Windows 10 laptop equipped with an Intel Haswell family i3-4005U CPU and 4GB DDR3 1600MHz RAM. The identical simulation code was tested on the same laptop running Arch Linux and same version of MATLAB for GNU/Linux. The simulation timings were comparable, although a few seconds faster on the latter.

4.2 Environments with different obstacles

The control technique suggested in this work is to achieve and maintain inverted-V shaped robot formation. We created seven distinct scenarios with varying shapes of static obstacles and one environment without any obstacles to see how the algorithm performs under different environmental situations.

This section contains the trajectory plots of the agents as they move through these environments and reach a pre-allocated target. Time vs position and deviation from mean velocity plots for these agents are shown, taking a platoon of 5 agents in each case.

A comparison table of time of actuation for each of these environments is given in Table 4.1.

4.2.1 Obstacle-less environment

The first environment taken is modelled without any obstacles. The agents create inverted-V formation and maintain this platoon while actuating towards a predefined target.

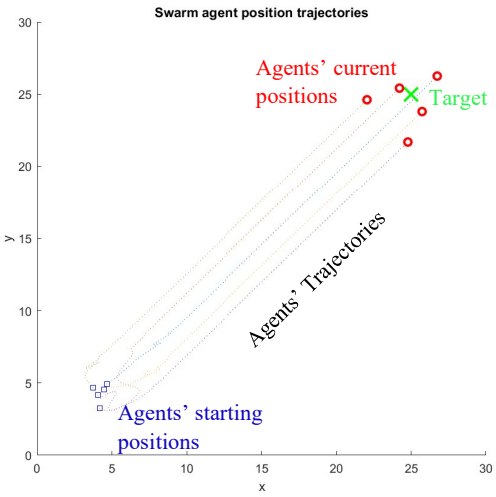

Fig. 4.1 Trajectories of agents with no obstacles in the environment

Figure 4.1 shows the trajectories of each agents at $t = 49.559957$s, when the platoon has reached the target. Blue squares represent the coordinates at which each agent was initialized, and the red circles represent the present position of the agents. The green cross represents the location of the target coordinate where the platoon is designated to reach. The dotted lines represent the trajectories for each agent.

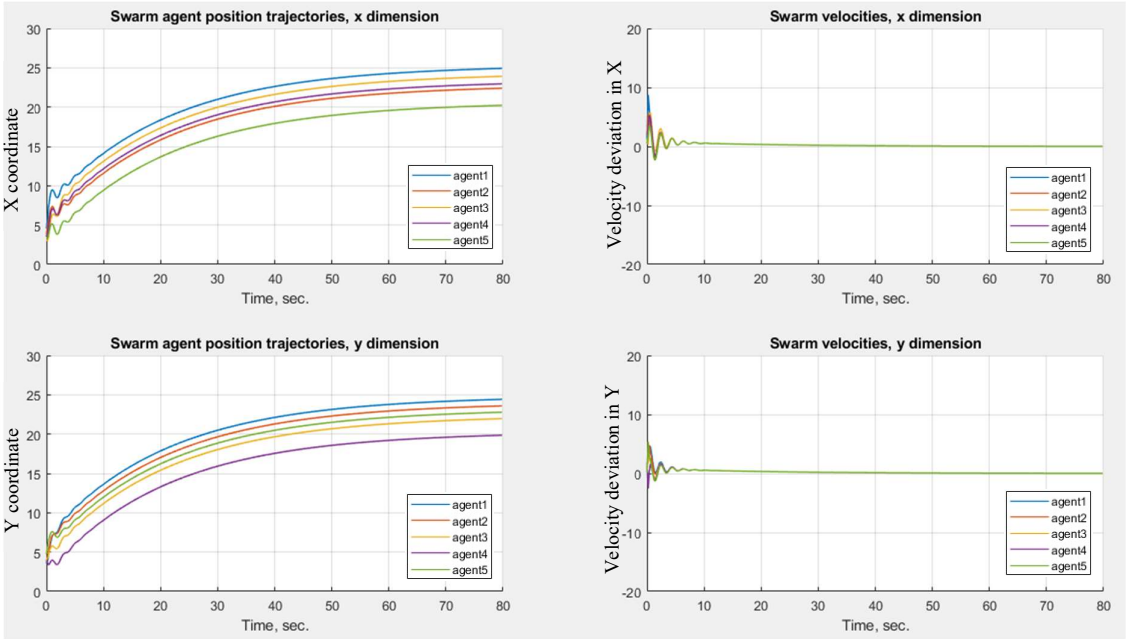

Fig. 4.2 Plot of time vs. agents' position & deviation from agents' mean velocity for obstacle-less environment

Figure 4.2 depicts the time vs. position and deviation from mean velocity plots for each agent. The location and velocity plots show that when there are no obstacles, the agents actuate towards the goal, unhindered, with velocity decreasing as the platoon gets closer to the destination. When the platoon arrives at the destination, the location of the agents stops changing. The position graphs of both the X and Y coordinates of agents, which have become parallel to the time axis, demonstrate this.

Once platoon is formed, all agents have the same velocity, hence deviation from mean velocity becomes zero, which prevents breaking of formation after platoon is formed.

4.2.2 Environment with Inverted-Z shaped obstacle

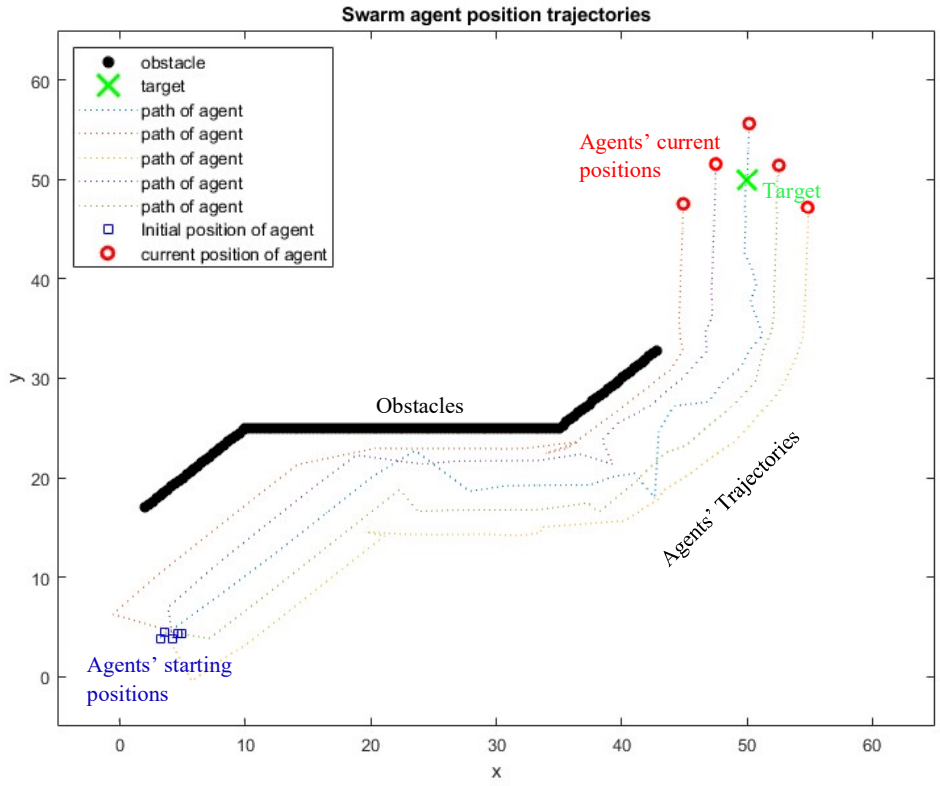

Fig. 4.3 Plot of paths taken by each agent to traverse through inverted-Z shaped obstacle and reach goal

Figure 4.3 shows the trajectory plots of agents in inverted-V platoon when moving through environment with inverted-Z shaped obstacle. Agents are shown as red circles, the target as green cross and the obstacle is shown in black thick line. The trajectories are shown in different coloured dotted lines to show paths taken by each agents.

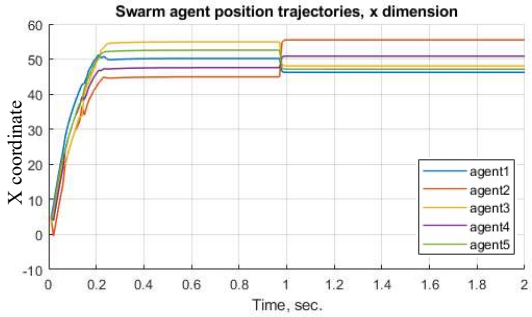

Fig. 4.4(a) Time vs. X-Position plot for inverted-Z obstacle region

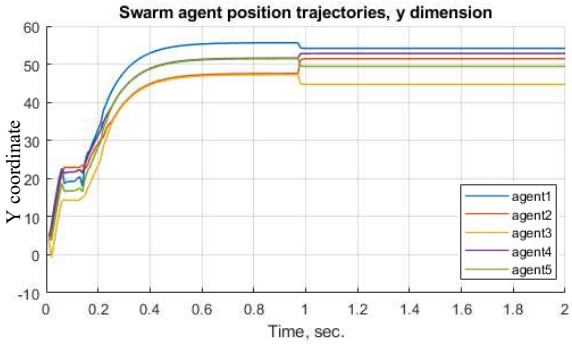

Fig. 4.4(b) Time vs. Y-Position plot for inverted-Z obstacle region

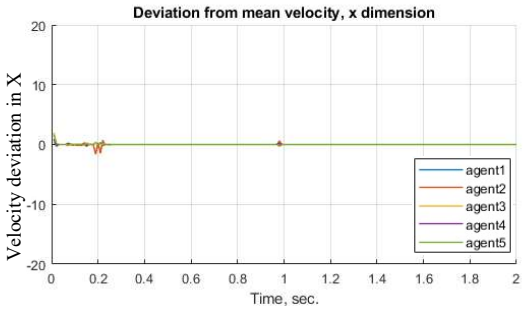

Fig. 4.4(c) Time vs. variation in X from mean velocity plot for inverted-Z obstacle

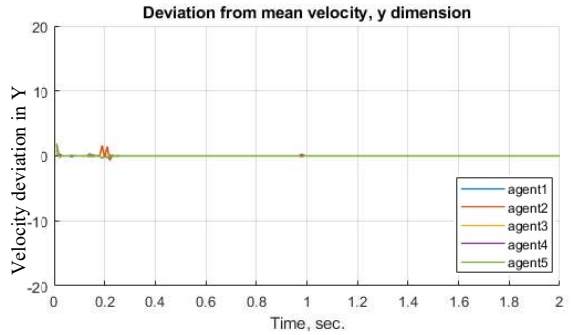

Fig. 4.4(d) Time vs. variation in Y from mean velocity plot for inverted-Z obstacle

Fig. 4.4 Plot of time vs. agents' position & deviation from agents' mean velocity for environment with inverted-Z shaped obstacle

Time vs. position and deviation from mean velocity plots are shown in figure 4.4. Time of simulation run is taken as 2 seconds, which is only meant for using as an independent variable in the simulation. Time of actuation, as compiled in table 4.1, is the actual time needed for the platoon to reach target coordinate. Here, 5 agents were taken for platoon formation.

4.2.3 Environment with Y shaped obstacle where agents initialized near base of obstacle

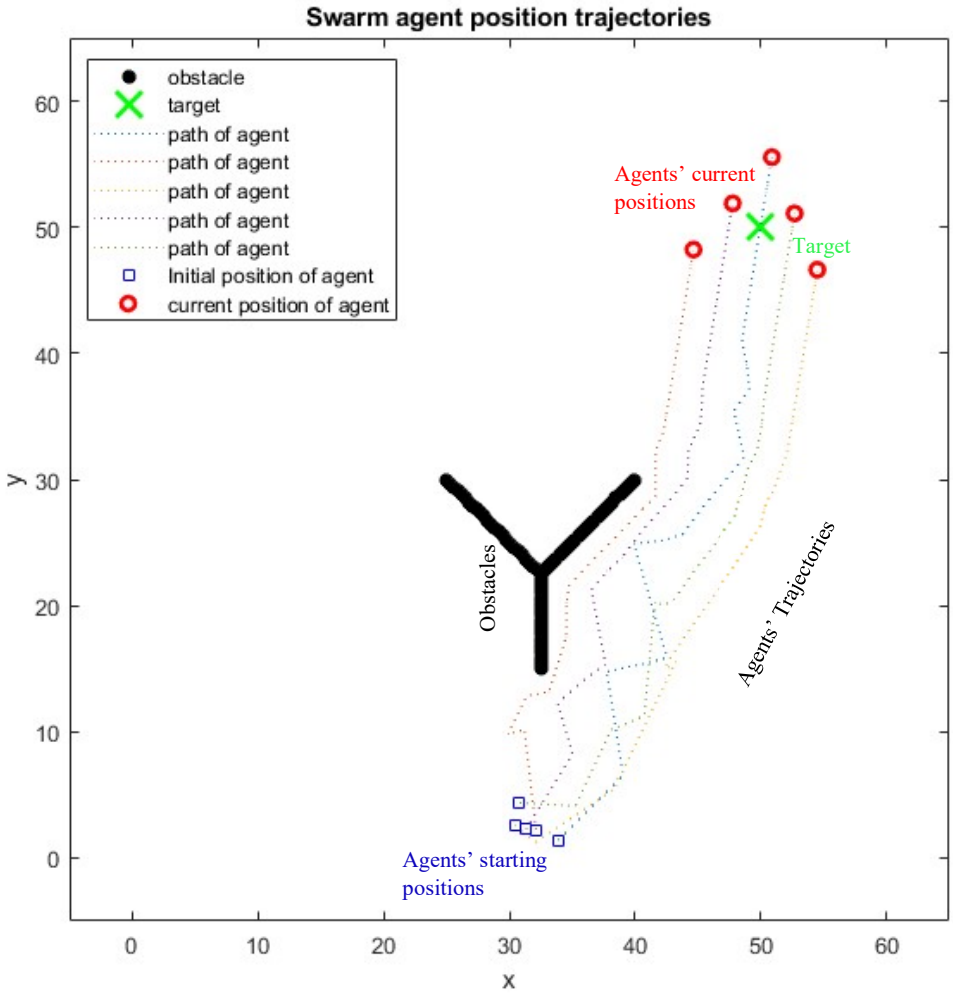

Fig. 4.5 Plot of paths taken by each agent to traverse through region with Y-shaped obstacle when agents defined near base of Y

Figure 4.5 shows the trajectory plots of agents in inverted-V platoon when moving through environment with Y-shaped obstacle. Agents are shown as red circles, the target as green cross and the obstacle is shown in black thick line. The trajectories are shown in different coloured dotted lines to show paths taken by each agents.

The path chosen is in the direction of the goal, without considering the position of obstacles, at the beginning of the simulation. Because the agents were initiated at the base of the Y-shape,

when an obstacle appears, the platoon deviates away from it while continuing to progress towards the goal.

4.2.4 Environment with Y shaped obstacle where agents initialized near origin

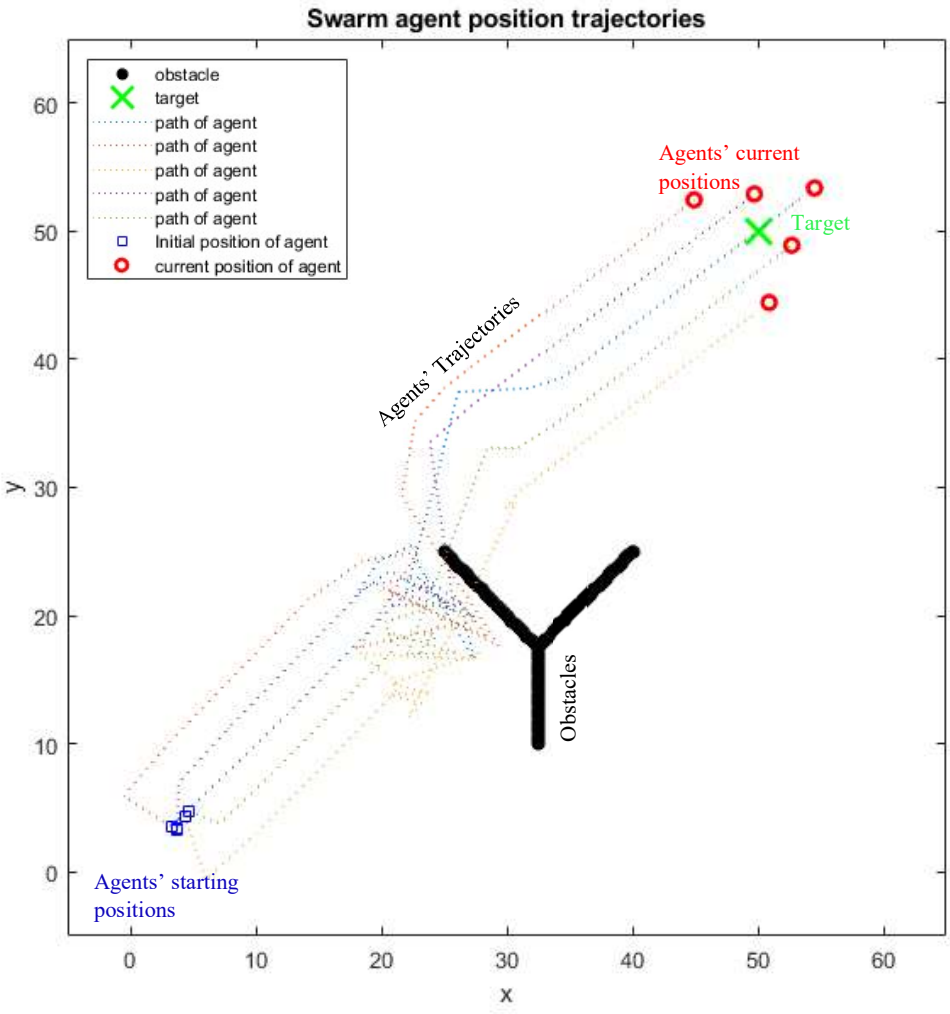

Fig. 4.6 Plot of paths taken by each agent to traverse through Y-shaped obstacle region with agents defined near origin

The same obstacle zone as in figure 4.5 is chosen here, but the agents are initialized near to the global origin. The right-most agent in the platoon encounters a blind-end in the direction of the destination coordinate due to the geometry of the obstacle at that moment.

As a result, that agent temporarily breaks the formation to go around the obstacle before rejoining the platoon to reach the target. The trajectory plots are shown in figure 4.6.

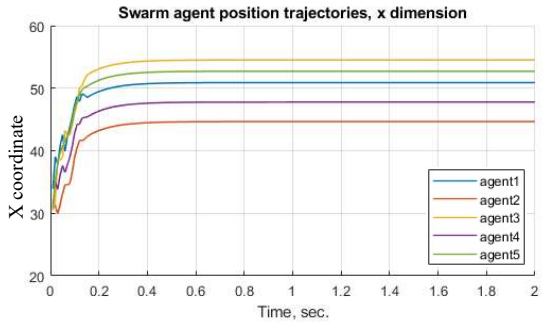

Fig. 4.7(a) Time vs. X-position plot for Y-shaped obstacle with agents starting near origin

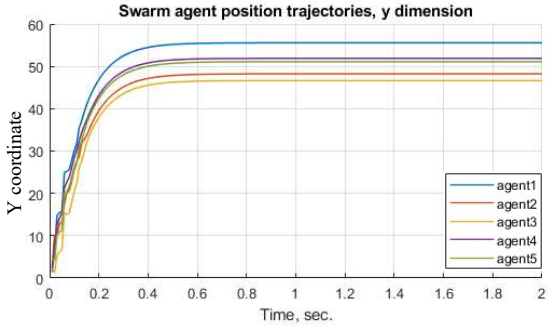

Fig. 4.7(b) Time vs. Y-position plot for Y-shaped obstacle with agents starting near origin

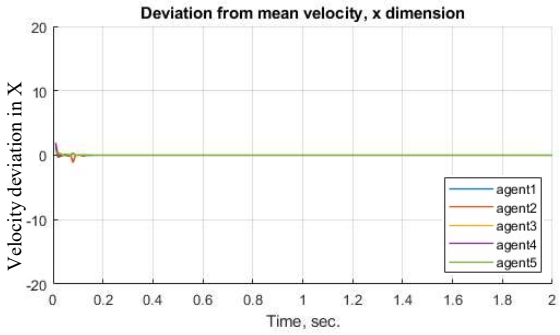

Fig. 4.7(c) Time vs. variation in X from mean velocity plot for Y-shaped obstacle with agents starting near origin

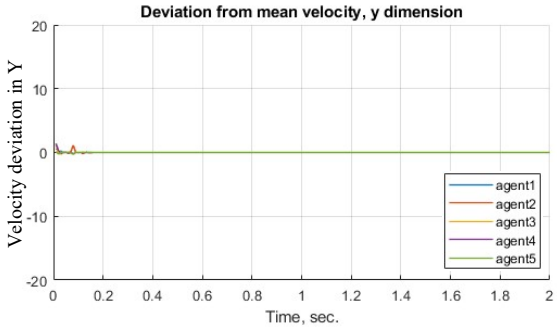

Fig. 4.7(d) Time vs. variation in Y from mean velocity plot for Y-shaped obstacle with agents starting near origin

Fig. 4.7 Plot of time vs. agents' position & deviation from agents' mean velocity for environment with Y shaped obstacle

Figure 4.7 shows the time vs. position and deviation from mean velocity plots for inverted-V platoon formation actuating through region with Y-shaped obstacle.

The agent initialization is close to origin, and 5 agents are used to create the platoon.

4.2.5 Environment with rectangular obstacle

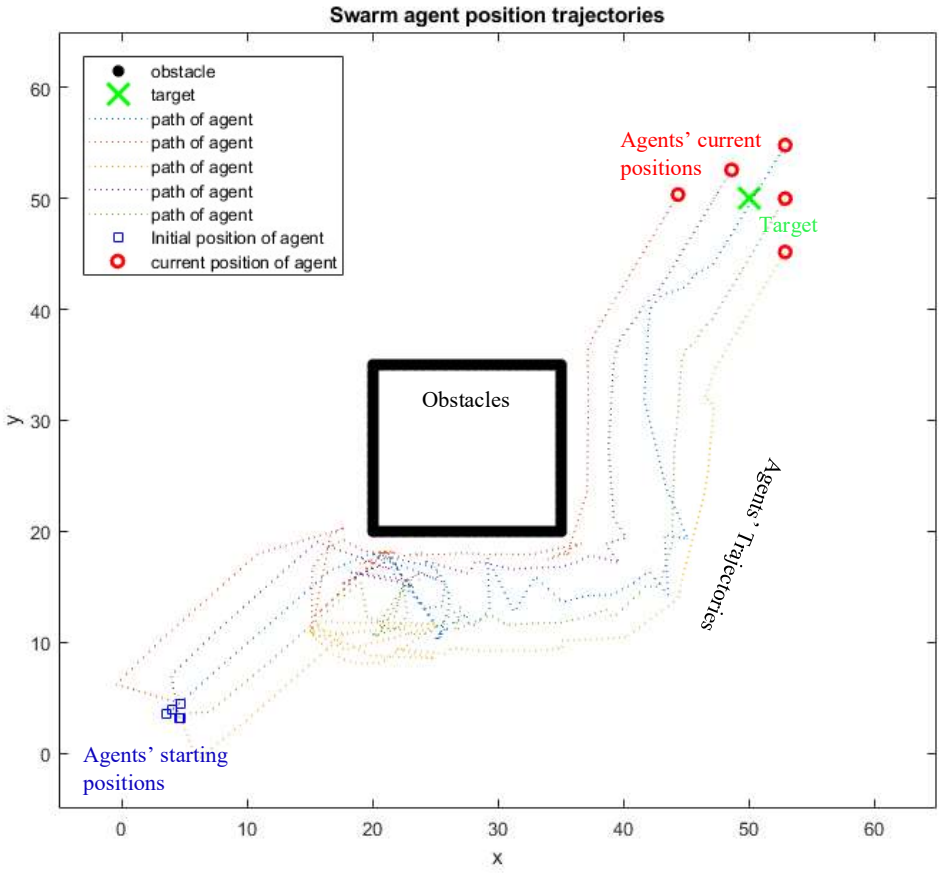

Fig. 4.8 Plot of paths taken by each agent to traverse through rectangular obstacle region

Figure 4.8 shows a rectangular obstacle path (depicted by a thick black line) which lies in the path of the agents that would have been taken to reach the destination coordinate (depicted by green cross). Each agent's initialization coordinate is shown by blue squares, from which the agents begin actuation. The dotted coloured lines represent the paths that each agent takes to actuate near the goal coordinate, avoid obstacles, and preserve formation.

The platoon actuates in the direction of line-of-sight towards goal, until it senses the obstacle. Then it reorients itself and continue to move towards the target, while keeping away from the obstacle, till it reaches its destination.

The time position and deviation from mean velocity plots are given below.

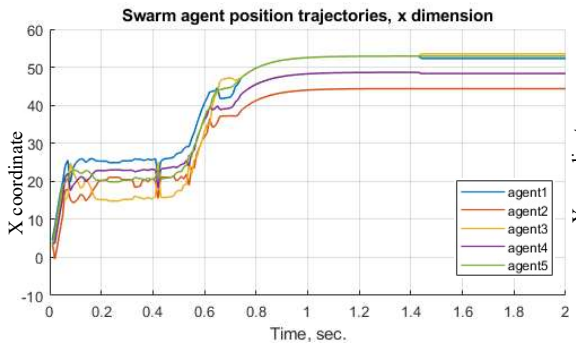

Fig. 4.9(a) Time vs. X-position plot for rectangular obstacle region

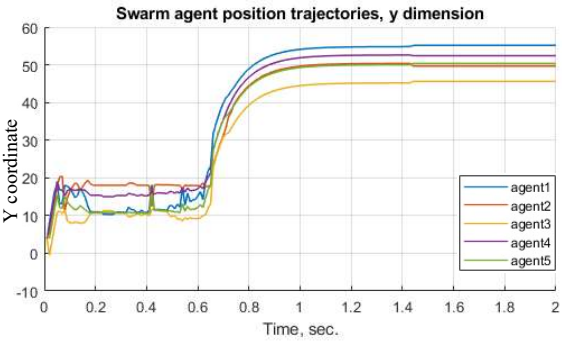

Fig. 4.9(b) Time vs. Y-position plot for rectangular obstacle region

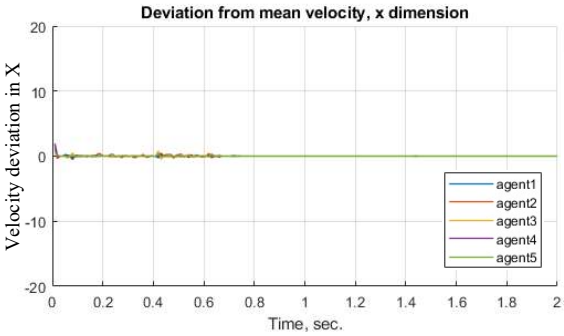

Fig. 4.9(c) Time vs. variation in X from mean velocity plot for rectangular obstacle

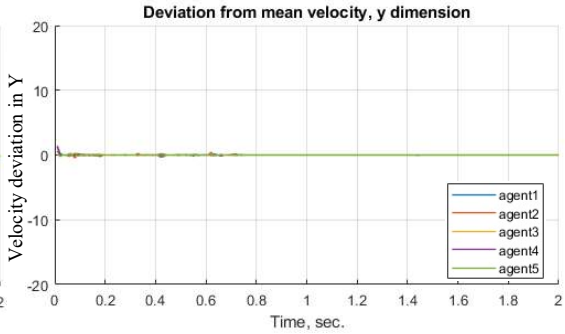

Fig. 4.9(d) Time vs. variation in Y from mean velocity plot for rectangular obstacle

Fig. 4.9 Plot of time vs. agents' position & deviation from agents' mean velocity for environment with rectangular obstacle

Figure 4.9 shows the time vs. position and deviation from mean velocity plots, obtained when simulated for environment with rectangular obstacle.

4.2.6 Environment with vertical line-shaped obstacle

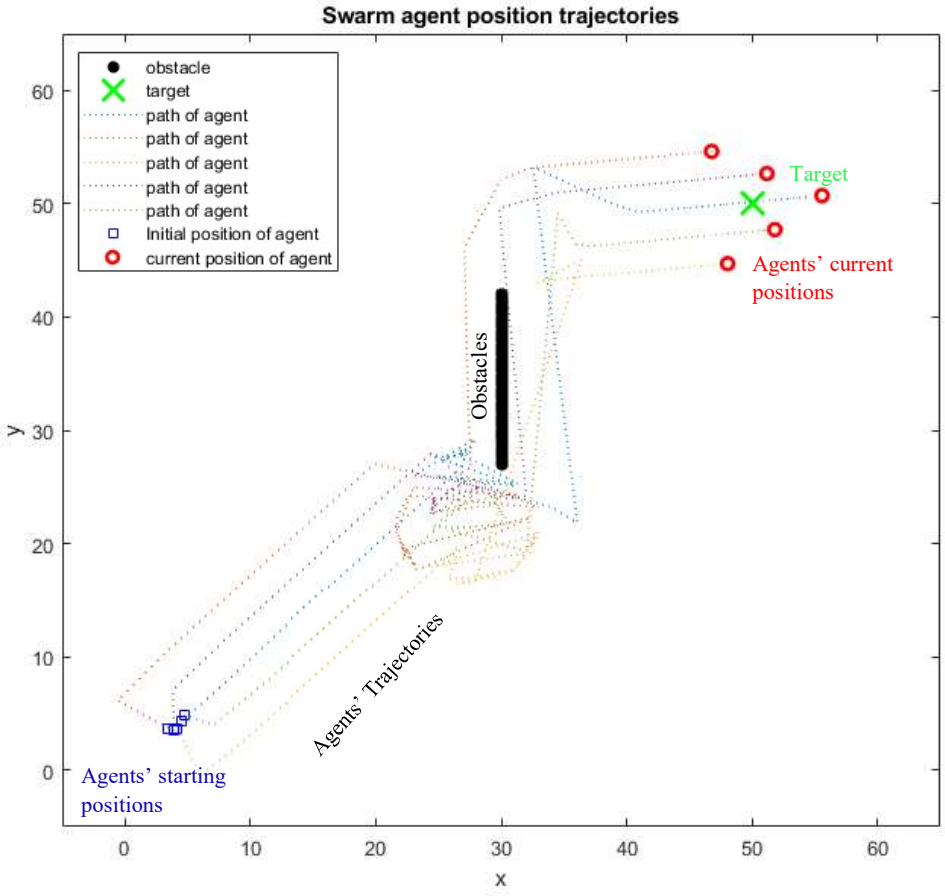

Fig. 4.10 Plot of paths taken by each agent to traverse through region with vertical line obstacle

Figure 4.10 shows a vertical line obstacle path is taken (depicted by black, thick line) which lies in the path of the agents that would have been taken to reach the destination coordinate (depicted by green cross). Each agent's initial coordinate is shown by blue squares, from which the agents begin actuation. The dotted coloured lines represent the paths that each agent takes to actuate near the goal coordinate, avoid obstacles, and preserve formation.

The platoon moves in the direction of its line of sight toward the goal until it detects an obstacle. Then it reorients itself and continues to advance towards the target while avoiding the obstruction until it arrives at its destination.

The time vs. position and deviation from mean velocity plots are shown below.

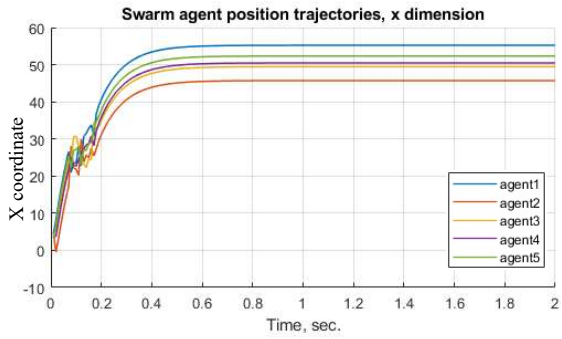

Fig. 4.11(a) Time vs. X-position plot for vertical line obstacle region

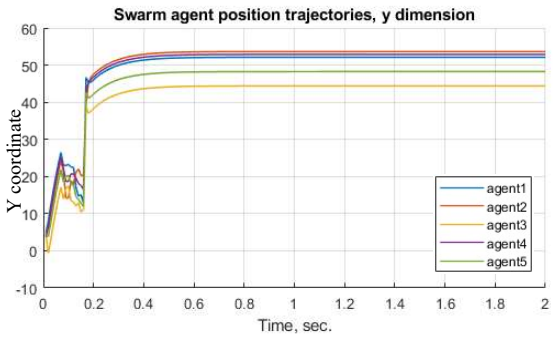

Fig. 4.11(b) Time vs. Y-position plot for vertical line obstacle region

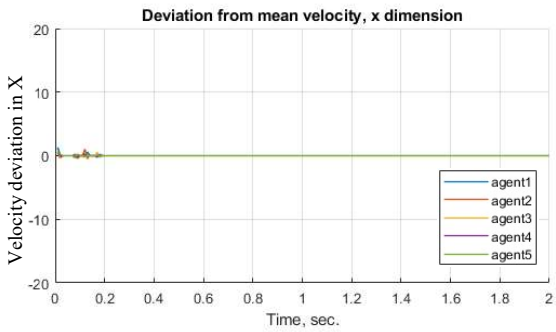

Fig. 4.11(c) Time vs. variation in X from mean velocity plot for vertical line obstacle

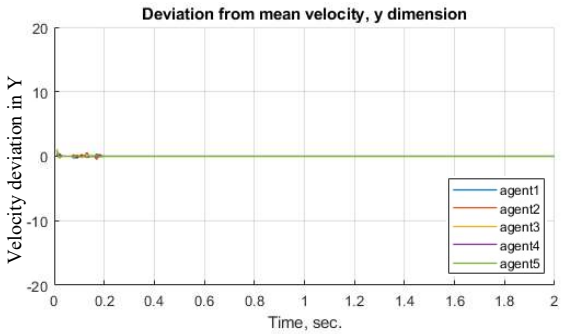

Fig. 4.11(d) Time vs. variation in Y from mean velocity plot for vertical line obstacle

Fig. 4.11 Plot of time vs. agents' position & deviation from agents' mean velocity for environment with vertical line shaped obstacle

Figure 4.11 shows the time vs. position and deviation from mean velocity plots, obtained when simulated for environment with vertical line-shaped obstacle.

4.2.7 Environment with horizontal line-shaped obstacle

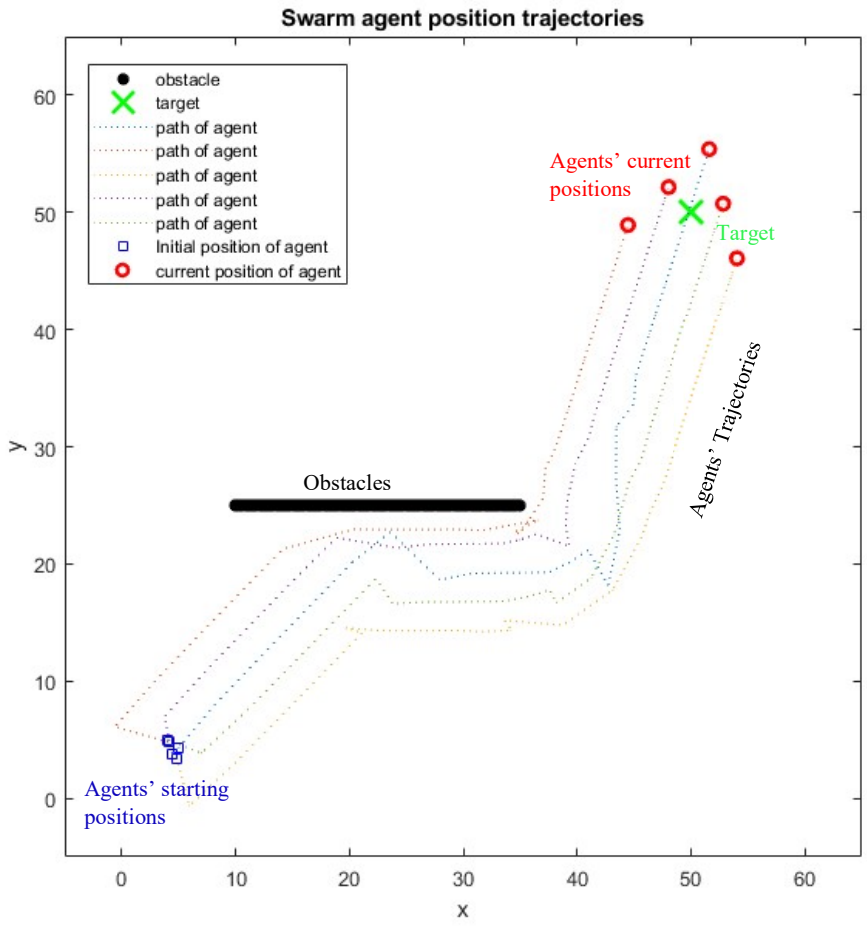

Fig. 4.12 Plot of paths taken by each agent to traverse through region with horizontal line obstacle

Figure 4.12 shows a vertical line obstacle path is taken (depicted by black, thick line) which lies in the path of the agents that would have been taken to reach the destination coordinate (depicted by green cross). Each agent's initial coordinate is shown by blue squares, from which the agents begin actuation. The dotted coloured lines represent the paths that each agent takes to actuate near the goal coordinate, avoid obstacles, and preserve formation.

In this case, the platoon senses a "wall" of obstacles, hence it actuates along its edge till line-of-sight is re-established with the target. Then it reorients and reaches the target.

The time vs. position and deviation from mean velocity plots are given below.

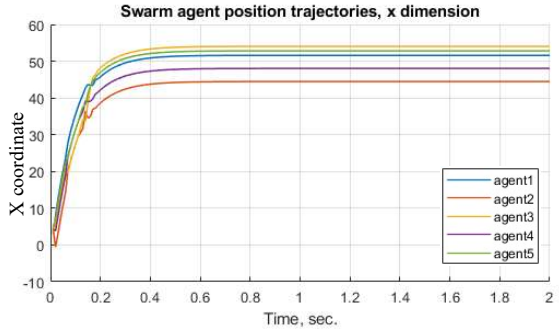

Fig. 4.13(a) Time vs. X-position plot for horizontal line obstacle region

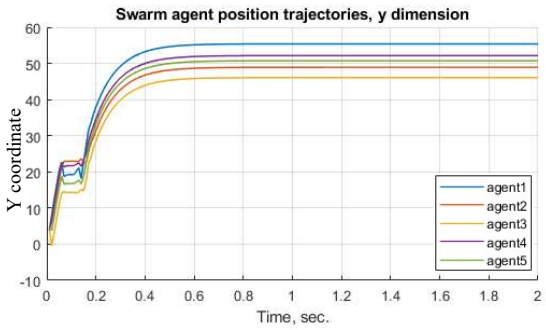

Fig. 4.13(b) Time vs. Y-position plot for horizontal line obstacle region

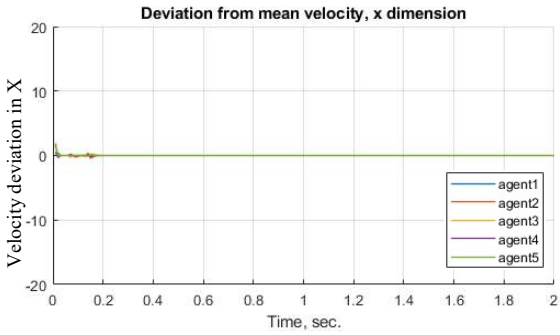

Fig. 4.13(c) Time vs. variation in X from mean velocity plot for horizontal line obstacle

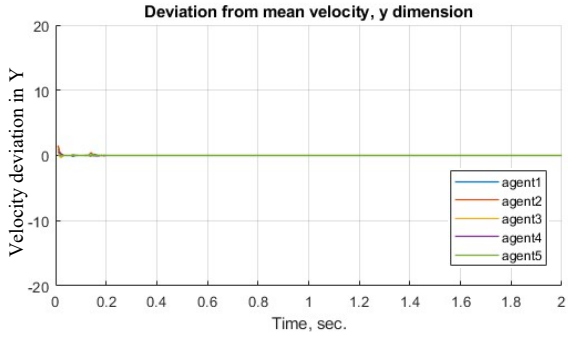

Fig. 4.13(d) Time vs. variation in Y from mean velocity plot for horizontal line obstacle

Fig. 4.13 Plot of time vs. agents' position & deviation from agents' mean velocity for environment with horizontal line shaped obstacle

Figure 4.13 shows the time vs. position and deviation from mean velocity plots, obtained when simulated for environment with horizontal line-shaped obstacle.

4.2.8 Environment with circular obstacle

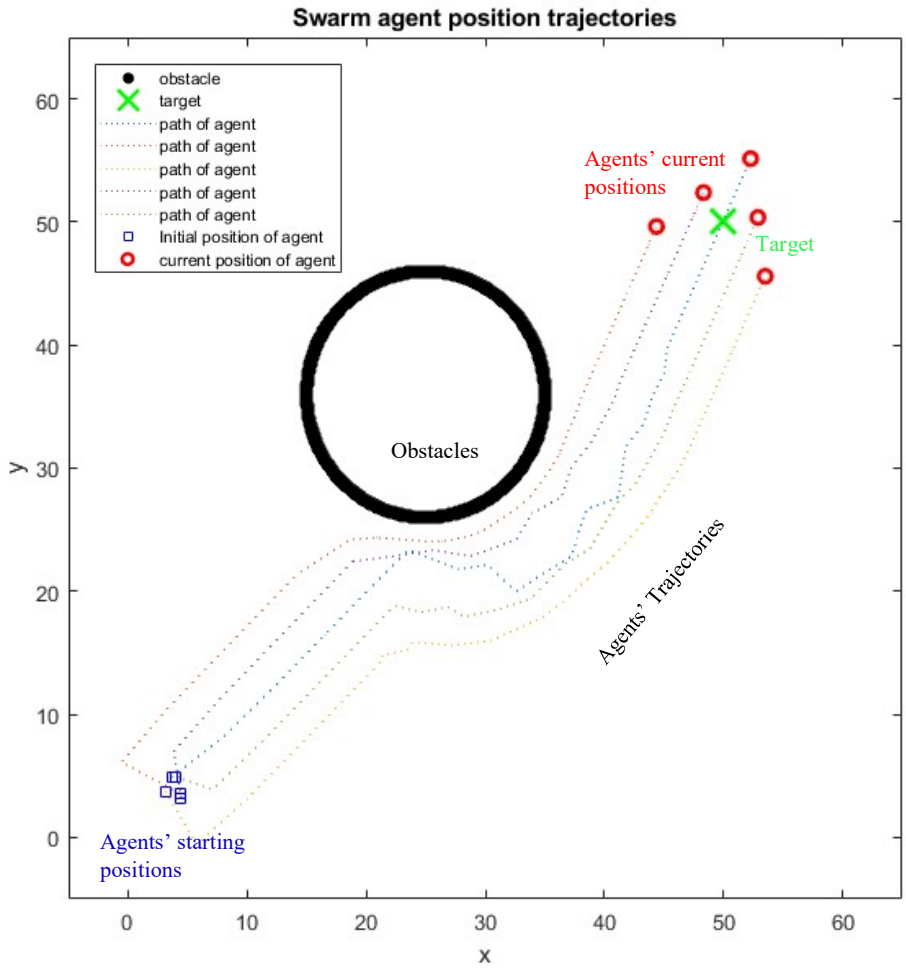

Fig. 4.14 Plot of paths taken by each agent to traverse through region with circular obstacle

Figure 4.14 shows a circular obstacle path is taken (depicted by black, thick line) which lies in the path of the agents that would have been taken to reach the destination coordinate (depicted by green cross). Each agent's initial coordinate is shown by blue squares, from which the agents begin actuation. The dotted coloured lines represent the paths that each agent takes to actuate near the goal coordinate, avoid obstacles, and preserve formation.

In this case, the circular obstacle is avoided by the platoon by going around the obstacle, tracing its edge, till it reaches the region from which the inverted-V shaped platoon can reach the target along line-of-sight path.

The time vs. position and deviation from mean velocity plots are given below.

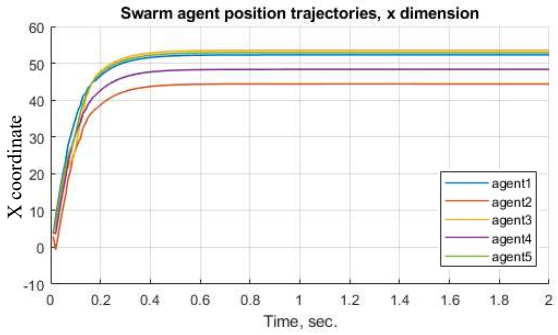

Fig. 4.15(a) Time vs. X-position plot for circular obstacle region

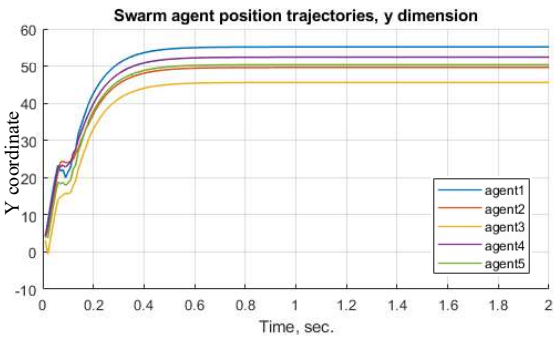

Fig. 4.15(b) Time vs. Y-position plot for circular obstacle region

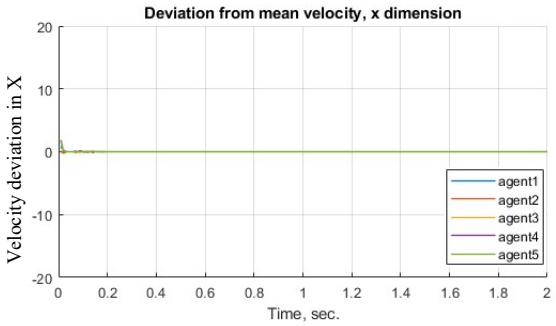

Fig. 4.15(c) Time vs. variation in X from mean velocity plot for circular region

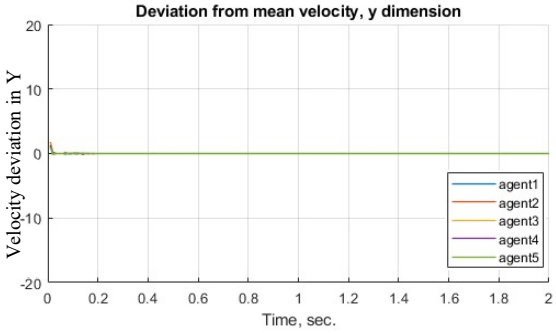

Fig. 4.15(d) Time vs. variation in Y from mean velocity plot for circular region

Fig. 4.15 Plot of time vs. agents' position & deviation from agents' mean velocity for environment with circular obstacle

Figure 4.15 shows the time vs. position and deviation from mean velocity plots, obtained when simulated for environment with circular obstacle.

4.2.9 Environment with obstacles forming an occluded path

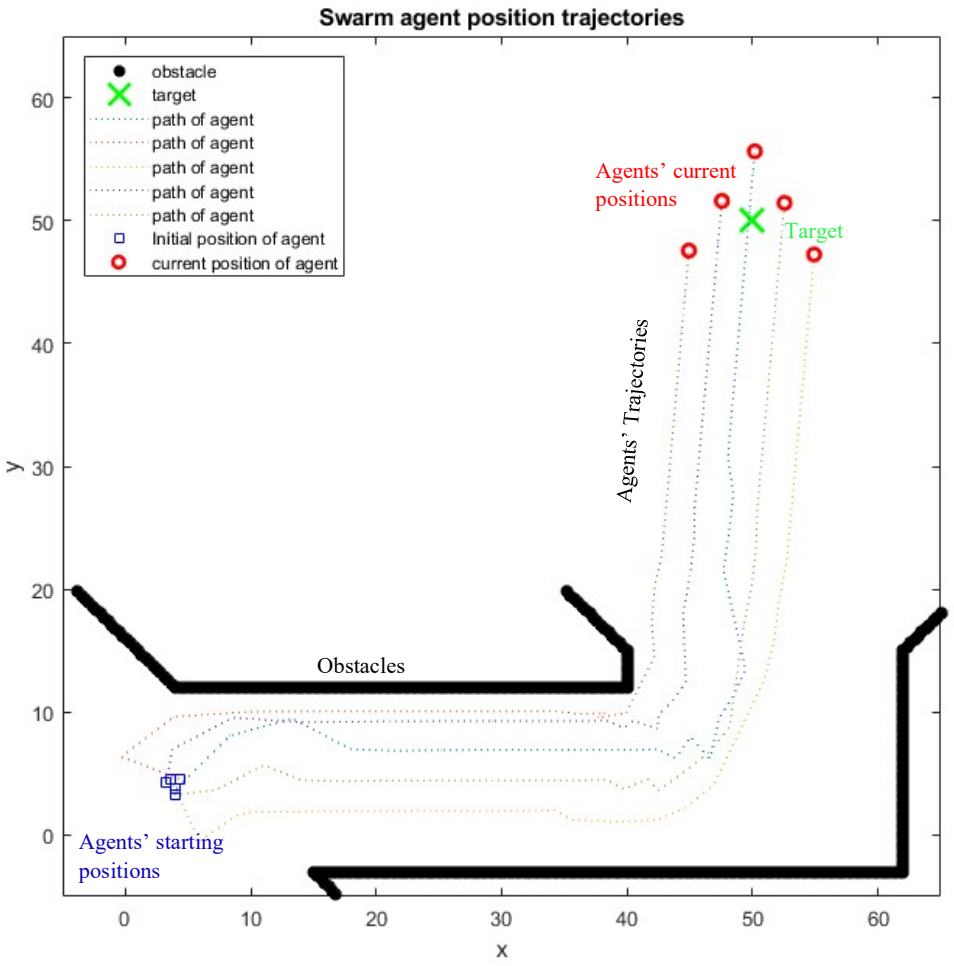

Fig. 4.16 Plot of trajectories taken by each agent to traverse through an occluded path region

Figure 4.16 shows occluded obstacle path is taken (depicted by black, thick line) which lies in the path of the agents that would have been taken to reach the destination coordinate (depicted by green cross). Each agent's initial coordinate is shown by blue squares, from which the agents begin actuation. The dotted coloured lines represent the paths that each agent takes to actuate near the goal coordinate, avoid obstacles, and preserve formation.

In this situation, the platoon senses obstacles on two sides, and hence has only a narrow path through which it can move. The platoon actuates till (45,10) until line-of-sight with target is established, at which point, it orients towards the target and reach the defined region.

The time vs. position and deviation from mean velocity plots are given below.

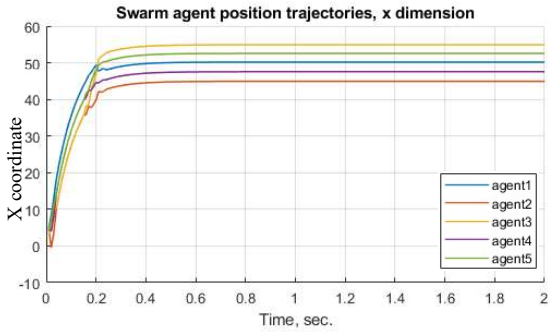

Fig. 4.17(a) Time vs. X-position plot for occluded path region

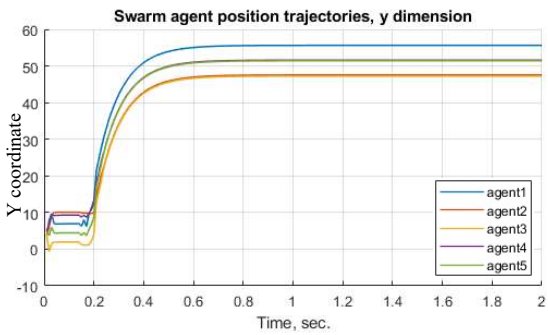

Fig. 4.17(b) Time vs. Y-Position plot for occluded path region

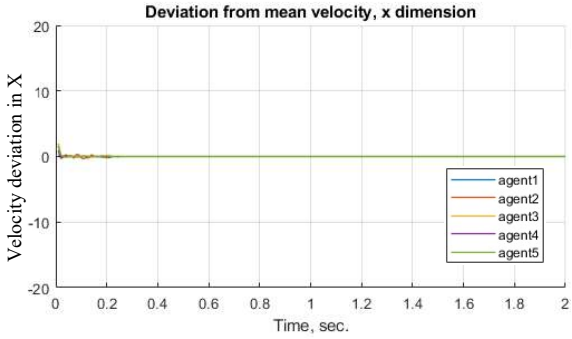

Fig. 4.17(c) Time vs. variation in X from mean velocity plot for occluded path region

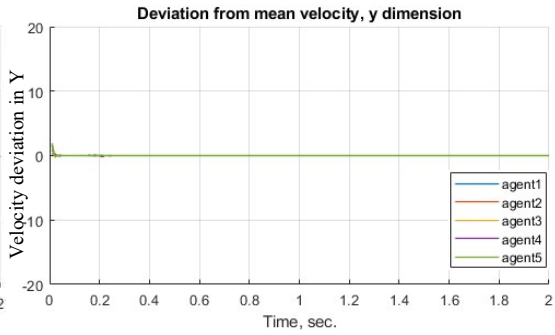

Fig. 4.17(d) Time vs. variation in Y from mean velocity plot for occluded path region

Fig. 4.17 Plot of time vs. agents' position & deviation from agents' mean velocity for environment with obstacle forming an occluded path

Figure 4.17 shows the time vs. position and deviation from mean velocity plots, obtained when simulated for environment with obstacles forming an occluded path.

4.3 Visualization of obstacle avoidance and path planning

Figure 4.18 depicts different time instants from actuation through the environment described in section 4.2.2.

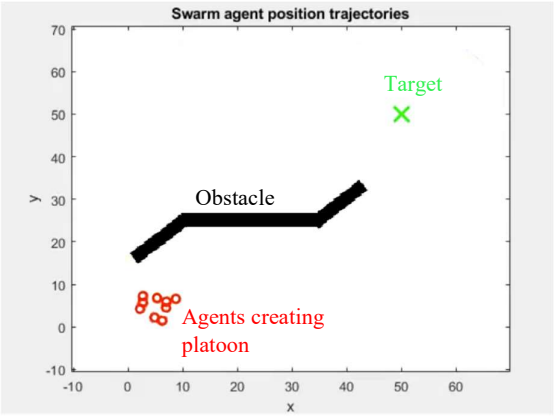

Fig 4.18(a) At $t = 6$ seconds

Fig. 4.18(b) At $t = 16$ seconds

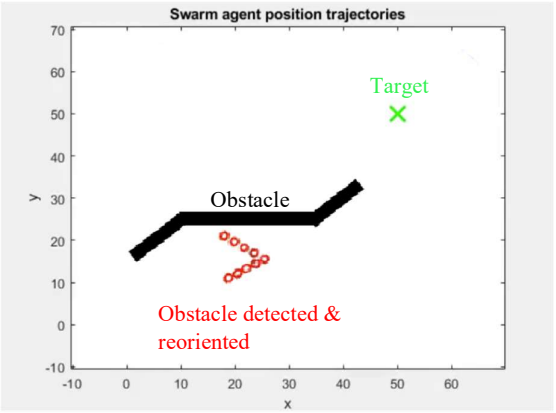

Fig. 4.18(c) At $t = 63$ seconds

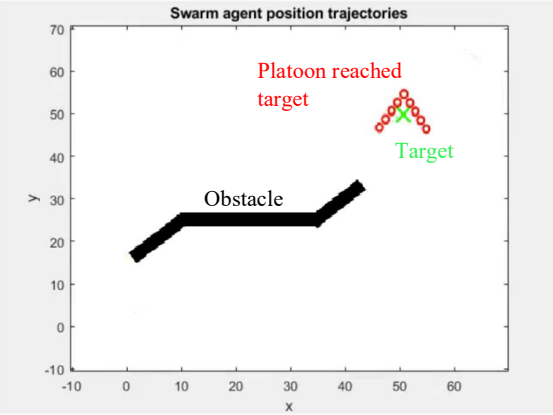

Fig. 4.18(d) At $t = 316$ seconds

Fig. 4.18 Plot of platoon actuation at four different time instances of simulation

The figures above show four instances from the start and end of the simulation when tested with obstacles

Figure 4.18 depicts the agents reorganizing themselves to create the platoon for occupying the first virtual triangle coordinates at simulation time $t = 6$ seconds. After $t = 16$ seconds, the agents have completed the inverted-V platoon formation by passing across two detected virtual circle zones. The controller detects the obstacles at $t = 63$ seconds, and the platoon moves away from them and reorients itself to actuate parallel to the X-coordinate along the obstacle wall, until the global controller detects a clearance to advance in the Y-coordinate towards the

goal. Finally, at $t = 316$ seconds, the platoon has arrived at the target location and has reoriented itself to point in the direction of the target when clearing was detected.

The time specified here is for the simulation's visualization to execute, which is limited by the available video memory for the CPU of the device on which the simulation was performed. Calculations of time of actuation are performed later in the job with the visualization turned off to off-load the processor for an accurate computation. However, it is restricted by device characteristics and is merely a comparison value.

4.4 Agent Swapping

From section 3.5.5.4, the agent swapping algorithm implemented in this work is visualised.

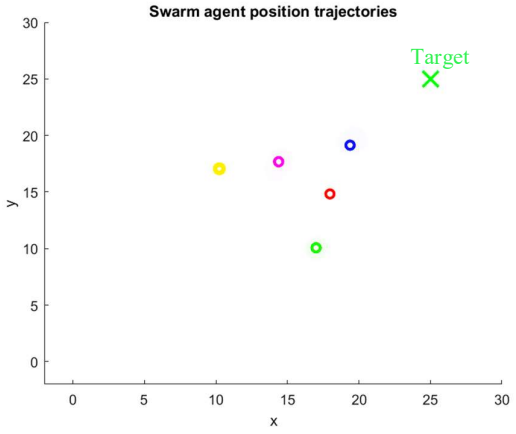

Fig. 4.19(a) Before agent swapping

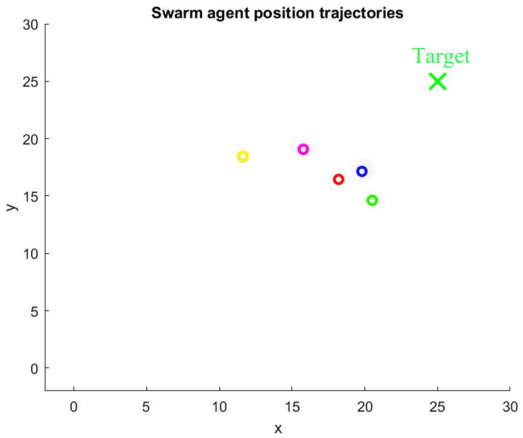

Fig. 4.19(b) During agent swapping : Right-arm agent getting swapped

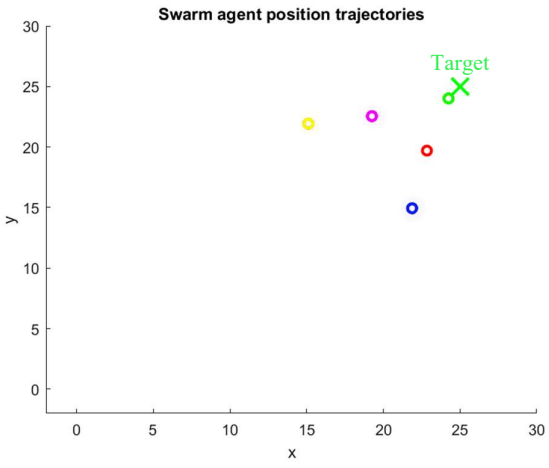

Fig. 4.19(c) After agent swapping

Fig. 4.19 Plots visualizing agent swapping action

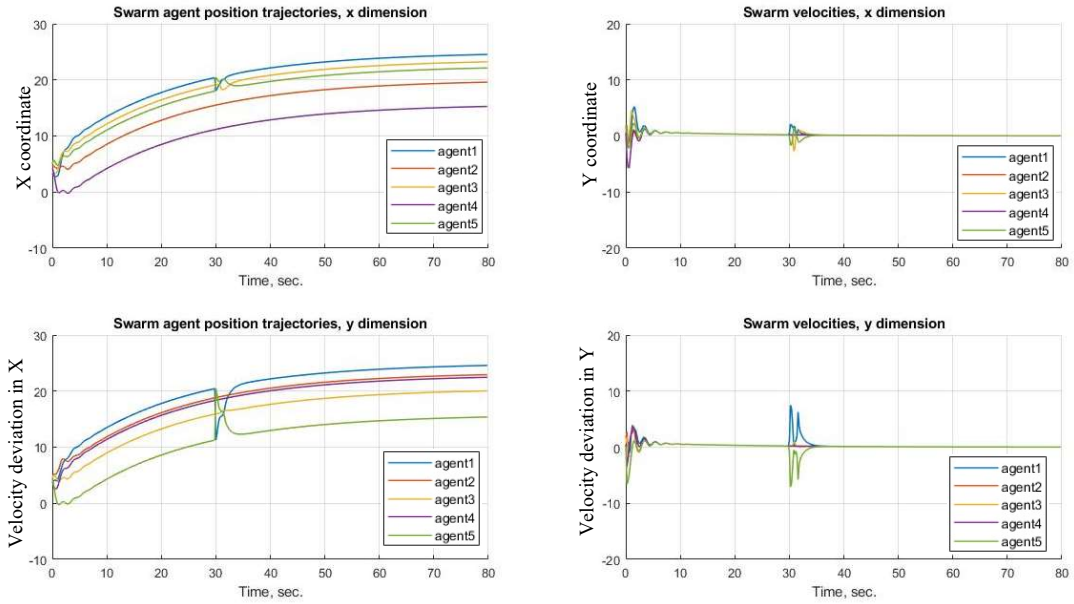

Fig. 4.20 Time vs. position & Deviation from mean velocity plot showing agent swapping

Figure 4.19 shows swapping of agent positions during execution of the simulation. Agents are depicted in different colours to aid in visualization of the swapping action. Time vs. position and deviation from mean velocity plots are shown in figure 4.20.

4.5 Comparative Study

This section presents a comparative study of different scenarios with actuation time. The chosen parameters are studied and their effect on the platoon model thus created in compared.

4.5.1 Maximum number of agents in platoon

A multi-robot system should have a large number of agents. The number of agents to participate in inverted-V platoon formation in this study is determined by how many agents can fit within the sensed region while preserving formation. Because the detected region is circular and a V-shape is made up of two line segments, the diameter is the longest line that can fit within a circle. In terms of the angle between the arms of the V, the maximum number of agents can be filled when the angle between the arms of the V is 0°.

4.5.2 Effective desirable angle

Allowing for inter-agent separation between agents, we obtain from section 3.5.3.1 that the minimum angle between the two arms of the inverted-V formation must be 60°. Because extending the angle any further would enable fewer agents in the platoon, the optimal angle between the platoon's arms is determined at 60°.

The selection of this angle allows for the highest number of agents in the platoon, which provides redundancy and may be required for mission demands; also, it favours maximum advantage from the upwash zone without reducing platoon size.

Increasing the sensing zone to accommodate a platoon with a V-shaped arm angle greater than 60° results in the following

Table 4.1 Time comparison with varying angle between arms of V-formation

V-formation angle	Simulation Time (seconds)	
	Without Agent Swapping	With Agent Swapping
60°	49.559957	53.7428174
95°	1216.099226	1318.746680

The perceived virtual circle radius must be extended to accommodate the same number of agents with a wider angle between two arms of the inverted-V form. As a result, each repetition requires a larger region to be sensed in order to fit the platoon. The table 4.1 validates the intuitive logic that iterating through a bigger region for sensing takes longer, and will be even longer if obstacle patterns are used, due to longer paths that must be followed to assure avoiding all obstacles while keeping the formation.

Figure 4.21 shows a platoon of 5 agents, which are denoted by red circles, actuating towards target coordinate, denoted by a green cross. It is circumscribed by a blue circle, which is the sensed region for that iteration. The angle between the arms of the inverted-V formation taken in this case is 95°.

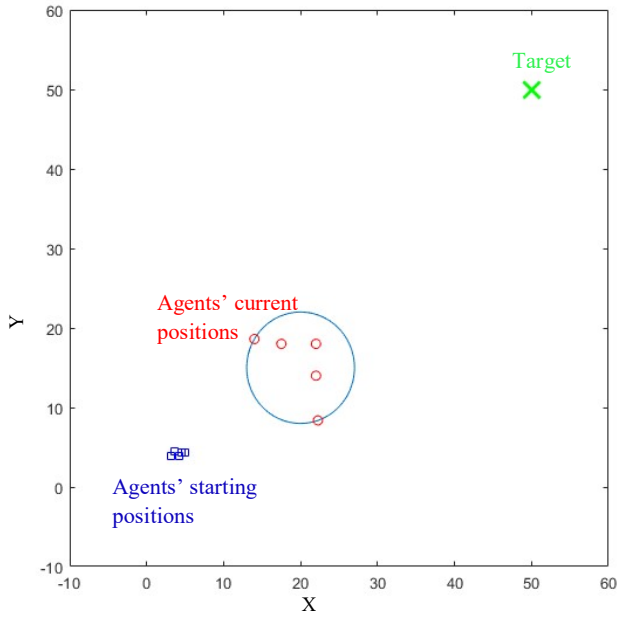

Fig. 4.21 Platoon placed inside virtual circle actuating with angle = 95° towards target coordinate

4.5.3 Comparison of actuation time when actuating through environments with different static obstacles

Table 4.2 shows the time of actuation required to reach a predefined goal, while actuating through regions with different obstacles, as discussed in section 4.2. The timings are recorded for the simulated agents to reach target autonomously, starting from arbitrary coordinates.

Table 4.2 Actuation-time comparison with different obstacle paths

Obstacle Type	Time of Reaching Target (seconds)	
	Without Agent Swapping	With Agent Swapping
No obstacle	49.559957	53.7428174
Horizontal Line	206.043675	223.433761
Vertical Line	153.438995	166.389246
Inverted-Z	316.648365	343.373487
Rectangle	465.173959	504.434641
Y-shaped with agents at origin	205.129419	222.442342
Y-shaped with agents at base	201.002364	217.966964
Circle	1001.828620	1086.382960
Occluded Path	622.649531	675.201151

It has to be noted here that the goal or target coordinate defined in this work is placed at $(50,50)$ and the agents are defined randomly within a rectangular region, the diagonal coordinates of which are $(2,2)$ and $(5,5)$. Obstacles placed are in the line of sight between the initialized coordinates and the target coordinate.

According to the table above, the time of actuation necessary to reach the target coordinate is consistently higher when agent swapping is conducted vs actuation without agent swapping. As a result, if agent switching is required, it comes at the expense of increased actuation time. It aids in the balancing of energy depletion in the system, hence the switching activity is a good attribute for the platoon to have.

However, it can be observed that the swapping action required more time if performed in the proximity of an obstacle, where the breaking and making of formation must be done while taking the obstacles into account. The longer the time necessary for actuation, the closer or more intricate the path that must be traversed. Because energy depletion is directly proportional to the time of actuation, the swapping action in this study is determined by an energy threshold limit.

It is preferable not to increase the actuation time unduly if the platoon can reach the destination without agent swapping. However, if the distance to be travelled by the platoon is great enough that the time necessary for actuation makes available energy critically lower than the threshold established, the leader agent may fail, rendering the global controller useless. In that instance, the extra time spent exchanging is a necessary sacrifice that must be made.

4.5.4 Comparison of time of actuation with some recent papers

This work is based on inverted-V platoon formation with agents initially placed at random coordinate points, sense and avoid obstacles, finally plan a path and reach a defined goal coordinate while maintaining formation. Previous similar works found has focused mainly on inverted-V platoon formation and commented on the time to form stable formation using their algorithm.

In the paper [73], the author has reported 20 seconds as the time to stable V-formation, whereas, the paper [8] has reported formation stabilization time, approximately 3 seconds.

Recording the time to form stable V-formation for this work yields as 1.54 seconds. From this result, we can conclude that the proposed method outperforms than the existing state of art.

4.6 Chapter Summary

This chapter presents the simulation results gained from the work done in this thesis, summarises the data created during the simulation run, and benchmarks the timing necessary for simulation run in various scenarios.

Then, a comparison is performed based on the results obtained from different environments, against which the algorithm is tested, and the outcome is presented. Finally, a timing comparison with an approximate estimate against other works is presented.

CHAPTER 5

Conclusion

5.1 Contributions to the thesis

i. Platoon formation control of a multi-robot system is presented and a point of limitation is highlighted which can be solved using biomimicry of inverted-V formation flight in birds. The formation details observed in birds is studied in detail, which is then modelled for a multi-robot system.

ii. A multi-robot formation control algorithm is designed and the method of implementation and obtained results are compiled which achieves an inverted-V formation of agents in a simulated environment from any arbitrary configuration, senses the region of interest, plans a suitable path avoiding obstacles and reach a defined target coordinate while maintaining the formation.

iii. The hierarchical controller efforts generated are demonstrated in a stepwise manner which collectively achieve the formation control behaviour of the multi-robot system. The control algorithm is tested against eight different environments with stationary obstacles of different shapes. Time of actuation comparison is also done with two recent papers.

5.2 Scope of future work

i. Agent failure handling for the platoon may be incorporated and dynamic obstacle handling for the model can be tested for controller performance.

ii. The simulated controller algorithm may be translated to real robot platoon actuation, thus getting a scope to study merits and demerits in the controller design from a practical point of view.

iii. Region of repulsion for agents may be designed to be decided dynamically based on physical dimension of agents, which would make the system scalable.

iv. Detailed environment modelling with fluid drag may be designed to validate claim of energy minimization from upwash region.

v. The controller may be upgraded to include the z-dimension, thus making the controller usable for flying agents such as drones, which might be used to compare bird flapping vs rotor thrust on upwash region generated.

vi. Controller may be further modified to include a smooth actuation mechanism to eliminate slight oscillation during actuation steps.

vii. Algorithm may be further improved to take advantage of parallel processing wherever possible to decrease sensing time required for each iteration in simulated environment.

Bibliography

[1] E. Garcia, M. A. Jimenez, P. G. De Santos and M. Armanda, “The evolution of robotics research,” in *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 90-103, 2007.

[2] L. Bayindir and E. Sahin, “A review of studies in swarm robotics,” in *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 15, no. 2, pp. 115-147, 2007.

[3] S. Monteiro and E. Bicho, “A dynamical systems approach to behavior-based formation control,” in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2606-2611, 2002.

[4] A. Nathan and V.C. Barbosa, “V-like Formation in Flocks of Artificial Birds,” in *Artificial Life*, vol. 14, no. 2, pp. 179-188, 2008.

[5] P. Seiler, A. Pant and J.K. Hedrick, "Analysis of bird formations," in *Proceedings of the 41st IEEE Conference on Decision and Control*, vol. 1, pp. 118-123. IEEE, 2002.

[6] I.L. Bajec and F. H. Heppner, "Organized flight in birds," *Animal Behaviour*, vol. 78, no. 4, pp. 777-789, 2009.

[7] J.T. Emlen, “Flocking behavior in birds," in *The Auk*, vol. 69, pp. 160-170, 1952.

[8] X. Li, Y. Tan, J. Fu and I. Mareels, "On V-shaped flight formation of bird flocks with visual communication constraints," in *13th IEEE International Conference on Control & Automation (ICCA)*, pp. 513-518, 2017.

[9] H. Hamann, *Swarm robotics: A formal approach*, 1st ed., Springer International Publications, 2018.

[10] S. C. Nair, E. M. Coronado, M. T. Frye and Y. Qin, "Swarm intelligence for the control of a group of robots," in *10th System of Systems Engineering Conference (SoSE)*, pp. 205-207, 2015.

[11] Y. Wang, Y. Song, and F. L. Lewis, “Robust adaptive fault-tolerant control of multiagent systems with uncertain nonidentical dynamics and undetectable actuation failures,” in *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3978-3988, 2015.

[12] G. Beni, “From swarm intelligence to swarm robotics,” in *International Workshop on Swarm Robotics*, pp. 1-9, Springer, Berlin, Heidelberg, 2004.

[13] J. D. Bjerknes and A. F. T. Winfield, “On fault tolerance and scalability of swarm robotic systems,” *Distributed autonomous robotic systems*, pp. 431-444, Springer, 2013.

[14] S. Saxena, R. Jais and D. R. Sona, "Search and Rescue Operations Using Robots Demonstrating Swarm Behaviour," in *5th International Conference on Communication and Electronics Systems (ICCES)*, pp. 115-119, 2020.

[15] J. Kennedy and R. Eberhart, “Particle swarm optimization,” *Proceedings of ICNN’95 International Conference on Neural Networks*, Perth, WA, Australia, vol. 4, pp. 1942-1948, 1995.

[16] X. Chen, P. Zhang, G. Du, and F. Li, “A distributed method of dynamic multi-robot task allocation problems with critical time constraints,” *Robotics and Automation Systems*, vol. 118, pp. 31-46, 2019.

[17] J. Cheng and R. Li, "A novel particle swarm optimization based on bacteria quorum sensing mechanism," in *IEEE Fifth International Conference on Advanced Computational Intelligence (ICACI)*, pp. 485-488, 2012.

[18] Z. Yang, M. Liu, J. Xiu and C. Liu, "Study on cloud resource allocation strategy based on particle swarm ant colony optimization algorithm," in *IEEE 2nd International Conference on Cloud Computing and Intelligence Systems*, pp. 488-491, 2012.

[19] A. Fünfhaus, J. Göbel, Julia Ebeling, H. Knispel, E. Garcia-Gonzalez, and E. Genersch, “Swarming motility and biofilm formation of paenibacillus larvae, the etiological agent of american foulbrood of honey bees (apis mellifera),” *Scientific reports*, vol. 8, no. 1, pp. 1–12, 2018.

[20] I. Ashraf, H. Bradshaw, T. T. Ha, J. Halloy, R. Godoy-Diana, and B. Thiria, “Simple phalanx pattern leads to energy saving in cohesive fish schooling,” *Proceedings of the National Academy of Sciences*, vol. 114, no. 36, pp. 9599-9604, 2017.

[21] F. Zou, H. Jones, D. Jiang, T. M. Lee, A. Martinez, K. Sieving, M. Zhang, Q. Zhang, E. Goodale, et al., “The conservation implications of mixed-species flocking in terrestrial birds, a globally-distributed species interaction network,” *Biological conservation*, vol. 224, pp. 267-276, 2018.

[22] V. Gazi and K. M. Passino, “Stability analysis of social foraging swarms: combined effects of attractant/repellent profiles,” *Proceedings of the 41st IEEE Conference on Decision and Control*, vol. 3, pp. 2848-2853, 2002.

[23] V. Gazi, “Stability Analysis of Swarms,” Ph.D. dissertation, The Ohio State University, 2002.

[24] C. C. Cheah, S. P. Hou, and J. J. E. Slotine, “Region-based shape control for a swarm of robots,” in *Automatica*, vol. 45, no. 10, pp. 2406– 2411, 2009.

[25] R. Haghighi and C.-C. Cheah, “Multi-group coordination control for robot swarms,” in *Automatica*, vol. 48, no. 10, pp. 2526–2534, 2012.

[26] D. Li, S. S. Ge, W. He, G. Ma, and L. Xie, “Multilayer formation control of multi-agent systems,” in *Automatica*, vol. 109, 2019.

[27] G. Vasarhelyi, C. Viragh, G. Somorajai, N. Tarcai, T. Szorenyi, T. Nepusz, and T. Vicsek, “Outdoor flocking formation flight with autonomous aerial robots,” *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3866-3873, 2014.

[28] A. Prayitno and I. Nilkhamhang, "V2V Network Topologies for Vehicle Platoons with Cooperative State Variable Feedback Control," *Second International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP)*, pp. 1-4, 2021.

[29] R. Chen, Y. Shen and C. Wang, "Ant Colony Optimization Inspired Swarm Optimization for Grid Task Scheduling," *International Symposium on Computer, Consumer and Control (IS3C)*, pp. 461-464, 2016.

[30] A. T. Hoang, V. T. Le and N. G. Nguyen, "A Novel Particle Swarm Optimization-Based Algorithm for the Optimal Communication Spanning Tree Problem," *Second International Conference on Communication Software and Networks*, pp. 232-236, 2010.

[31] J. Shin and J. Woo, "Test and evaluation for the autonomous multi-zone navigation coverage test in IEC 62885-7," *17th International Conference on Control, Automation and Systems (ICCAS)*, pp. 1949-1951, 2017.

[32] H. Li and J. Xing, "Combat effectiveness evaluation method of photoelectric defense system based on SVM optimized by particle swarm algorithm," *Proceedings of*

International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC), pp. 2680-2684, 2013.

[33] G. H. Elkaim and M. Siegel, "A lightweight control methodology for formation control of vehicle swarms," *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 191–196, 2005.

[34] D. Roy, A. Chowdhury, M. Maitra, and S. Bhattacharya, "Virtual region based multi-robot path planning in an unknown occluded environment," In 2019 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 588-595, 2019.

[35] Z. Qu, "Cooperative Control of Dynamical Systems: Applications to Autonomous Vehicles," Springer Science and Business Media, 2009.

[36] M. A. Hsieh, V. Kumar and L. Chaimowicz, "Decentralized controllers for shape generation with robotic swarms," *Robotica*, vol. 26, no. 5, pp. 691-701, 2008.

[37] James MacDonald, Metro Creative, "Why Do Geese Fly in V Formation," in JSTOR, Nov 2015. [Online]. Available: https://daily.jstor.org/geese-fly-v-formations/

[38] T.C. Williams, T.J. Klonowski, and P. Berkeley, "Angle of canada goose V flight formation measured by radar," *Auk*, vol. 93, pp. 554-559, July 1976.

[39] J.R. Speakman and D. Banks, "The function of flight formations in Greylag Geese Anser anser; energy savings or orientation?," *Ibis*, vol. 140, pp. 280-287, 1998.

[40] C.J. Cutts and J.R. Speakman, "Energy savings in formation flight of pink-footed geese," *J. of Experimental Biology*, vol. 189, pp. 251-261, 1994.

[41] P. B. S. Lissaman, and C. A. Shollenberger, "Formation flight of birds," *Science*, vol. 168, no. 3934, pp. 1003-1005, 1970.

[42] P.F. Major and L.M. Dill, "The three-dimensional structure of airborne bird flocks," *Behavioral Ecology and Sociobiology*, vol. 4, pp. 111-122, 1978.

[43] F.H. Heppner, "Avian flight formations," *Bird-Banding*, vol.45, no.2, pp.160-169, 1974.

[44] B. Parslew, "Simulating avian wingbeats and wakes," Ph.D. dissertation, The University of Manchester (United Kingdom), 2012.

[45] D. D. Chin, D. Lentink, "Flapping wing aerodynamics: from insects to vertebrates," *Journal of Experimental Biology*, vol. 219, no. 7, pp. 920–932, April 2016.

[46] J. R. Usherwood, J. A. Cheney, J. Song, S. P. Windsor, J. P. J. Stevenson, U. Dierksheide, A. Nila, and R. J. Bomphrey. "High aerodynamic lift from the tail reduces drag in gliding raptors," *Journal of experimental biology*, vol. 223, no. 3, 2020.

[47] Getty, "Birds' stunning V-formation, which is far more tightly controlled than scientists thought," Mirror, Jan 2014. [Online]. Available: https://www.mirror.co.uk/news/technology-science/mystery-birds-fly-v-formation-3025188

[48] H. Weimerskirch, J. Martin, Y. Clerquin, P. Alexandre, and S. Jiraskova. "Energy saving in flight formation," *Nature*, vol. 413, no. 6857, pp. 697-698, 2001.

[49] J. P. Badgerow, and F. R. Hainsworth. "Energy savings through formation flight? A re-examination of the vee formation," *Journal of Theoretical Biology*, vol. 93, no. 1, pp. 41-52, 1981.

[50] P. Seiler, A. Pant, and J. K. Hedrick. "A systems interpretation for observations of bird V-formations," *Journal of theoretical biology*, vol. 221, no. 2, pp. 279-287, 2003.

[51] W. J. Hamilton, "Social aspects of bird orientation mechanisms," *Animal orientation and navigation*, pp. 57-71, 1967.

[52] D. Willis, Peraire, J., and K. Breuer, "A computational investigation of bio-inspired formation flight and ground effect," In 25th *AIAA Applied Aerodynamics Conference*, pp. 4182, 2007.

[53] T. Liu, "Comparative scaling of flapping-and fixed-wing flyers," *AIAA journal*, vol. 44, no. 1, pp. 24-33, 2006.

[54] J. Gao, C. Gu, H. Yang, and M. Wang, "A flight formation mechanism: The weight of repulsive force," *Communications in Nonlinear Science and Numerical Simulation*, Elsevier B. V., vol. 95, pp. 105648, 2021.

[55] Mishima, Shigenori, and Spyros A. Kinnas, "Application of a numerical optimization technique to the design of cavitating propellers in nonuniform flow," *Journal of ship research*, pp. 93-107, 1997.

[56] M. Moškon, F. H. Heppner, M. Mraz, N. Zimic, and I. L. Bajec, "Fuzzy model of bird flock foraging behavior," In *Proceedings of the 6th EUROSIM Congress on Modelling and Simulation*, vol. 2, pp. 1-6. by B. Zupancic, R. Karba & S. Bla zic, 2007.

[57] S. P. J. Perera, “Peripheral Vision of Eyes,” The IFOD, Feb 2019. Accessed: Jun 1, 2020. [Online]. Available: https://www.theifod.com/stereo-vision/

[58] J. P. Badgerow, “An analysis of function in the formation flight of Canada geese,” *The Auk*, vol. 105, pp. 749-755, 1988.

[59] C. M. Sewatkar, A. Sharma and A. Agrawal, “A first attempt to numerically compute forces on birds in V formation,” *Artificial life*, vol. 16, no. 3, pp. 245-258, 2010.

[60] W. J. Devenport, M. C. Rife, S. I. Liapis, and G. J. Follin. "The structure and development of a wing-tip vortex," *Journal of fluid mechanics*, vol. 312, pp. 67-106, 1996.

[61] M. Sabogal, Pilot Effect, “Wingtip Vortices | Pilot Tutorial”. (Jan 18, 2020). Accessed: Jun 4, 2020. [Online Video]. Available: https://www.youtube.com/watch?v=FFgbUx1GdxM

[62] H. Kang, J. Joung, and J. Kang, “Power-efficient formation of UAV swarm: Just like flying birds?,” In *GLOBECOM 2020-2020 IEEE Global Communications Conference*, pp. 1-6. IEEE, 2020.

[63] L. Consolini, F. Morbidi, D. Prattichizzo, and M. Tosques, “Leader-follower formation control of nonholonomic mobile robots with input constraints,” *Automatica*, vol. 44, no. 5, pp. 1343-1349, 2008.

[64] Z. Qiao, J. Zhang, X. Qu and J. Xiong, "Dynamic Self-Organizing Leader-Follower Control in a Swarm Mobile Robots System Under Limited Communication," in *IEEE Access*, vol. 8, pp. 53850-53856, 2020.

[65] H. Wang, D. Guo, X. Liang, W. Chen, G. Hu, and K. K. Leang, “Adaptive vision-based leader-follower formation control of mobile robots,” *IEEE Transactions on Industrial Electronics*, vol. 64, no. 4, pp. 2893-2902, 2016.

[66] M. C. Lee and M. G. Park, “Artificial Potential Field Based Path Planning For Mobile Robots Using A Virtual Obstacle Concept,” In *Proceedings 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003)*, vol. 2, pp. 735-740. IEEE, 2003.

[67] S. Dasgupta, V. Raghuraman, A. Choudhury, T. N. Teja and J. Dauwels, "Merging and splitting maneuver of platoons by means of a novel PID controller," *IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1-8, 2017.

[68] V. Gazi and K. M. Passino, “A Class Of Attractions/Repulsion Functions For Stable Swarm Aggregations,” *International Journal of Control*, vol. 77, issue 18, pp. 1567-1579, 2004.

[69] Y. Liu and K. M. Passino, “Stable social foraging swarms in a noisy environment,” *IEEE Transactions on Automatic Control*, vol. 49, no. 1, pp. 30-44, 2004.

[70] D. Roy, M. Maitra and S. Bhattacharya, “Study of formation control and obstacle avoidance of swarm robots using evolutionary algorithms,” *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 003154-003159, 2016.

[71] X. Wei, J. Yang, X. Fan, L. Xiao and C. Wang, "Distributed Adaptive Parameter Cooperative Guidance Laws for Vee Formation Flight," *Chinese Automation Congress (CAC)*, pp. 3153-3158, 2018.

[72] F.R. Hainsworth, “Induced drag savings from ground efiect and formation flight in brown pelicans," *J. of Experimental Biology*, vol. 135, pp. 431-444, 1988.

[73] F. S. Cattivelli and A. H. Sayed, “Self-Organization in Bird Flight Formation Using Diffusion Adaptation,” *3rd International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pp. 49-52, 2009.

[74] P. Waldron, "Why Birds Fly in a V Formation," in Science, Jan 2014. [Online]. Available: https://www.science.org/content/article/why-birds-fly-v-formation-rev2

[75] V. Pratt, “Direct least-squares fitting of algebraic surfaces,” *ACM SIGGRAPH computer graphics*, vol. 21, no. 4, pp. 145-152, 1987.

[76] Y. Chen, and Z. Wang, "Formation control: A Review and Consideration," In *Proceedinds of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3181-3186, 2005.

[77] F. S. Cattivelli, and A. H. Sayed, “Modeling bird flight formations using diffusion adaptation,” *IEEE transactions on signal processing*, vol. 59, no. 5, pp. 2038-2051, 2011.

[78] H. Wang and M. Rubenstein, “Shape formation in homogeneous swarms using local task swapping,” *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 597-612, 2020.

[79] Y. Liu, “Cohesive Behaviors of Cooperative Multiagent Systems with Information Flow Constraints,” Ph.D. dissertation, The Ohio State University, 2004.

[80] J. A. Oroko and G. N. Nyakoe, "Obstacle Avoidance and Path Planning Schemes for Autonomous Navigation of a Mobile Robot: A Review," *Proceedings of the 2012 Mechanical Engineering Conference on Sustainable Research and Innovation*, vol. 4, pp. 314-318, 2012.

[81] D. Roy, A. Chowdhury, M. Maitra and S. Bhattacharya, "Multi-robot virtual structure switching and formation changing strategy in an unknown occluded environment," In 2018 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4854-4861, IEEE, 2018.

[82] V. Gazi and K. M. Passino, "Swarm stability and optimization," 1st ed., Springer

Science & Business Media, Berlin, Heidelberg, 2011.

[83] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pp. 25-34, 1987.

[84] X. Liang, L. Li, J. Wu and H. Chen, "Mobile robot path planning based on adaptive bacterial foraging algorithm," *Journal of Central South University*, vol. 20, no. 12, pp. 3391-3400, 2013.

[85] D. Jayaraj, L. T. Vishwanath and O. V. S. Sarma, "Path Planning In Swarm Robots Using Particle Swarm Optimization On Potential Fields," *International Journal of Computer Applications*, vol. 60, no. 13, 2012.

[86] Z. Hong, Y. Liu, Z. Gao, and C. Yi, "The Dynamic Path Planning Research For Mobile Robot Based On Artificial Potential Field," In 2011 *International Conference on Consumer Electronics, Communications and Networks (CECNet)*, pp. 2736-2739. IEEE, 2011.

[87] Y. Origane, Y. Hattori, and D. Kurabayashi, "Control Input Design for a Robot Swarm Maintaining Safety Distances in Crowded Environment," *Symmetry*, vol. 13, no. 3, pp. 478, 2021.

[88] H. Shi, and G. Xie, "Collective dynamics of swarms with a new attraction/repulsion function," *J. Sun, Ed., Mathematical Problems in Engineering*, pp. 13, Hindawi, 2011.

[89] D. Hummel, "Formation flight as an energy-saving mechanism," *Israel Journal of Zoology*, vol. 41, pp. 261-278, 1995.

[90] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithm and theory," *IEEE Transactions in Automatic Control*, vol. 51, pp. 401-420, 2006.

[91] I. Vine, "Risk of visual detection and pursuit by a predator and selective advantage of flocking behaviour," *Journal of Theoretical Biology*, vol. 30, pp. 405-422, 1971.

[92] D. Hummel, "Aerodynamic aspects of formation flight in birds," *Journal of Theoretical Biology*, vol. 104, pp. 321-347, 1983.

[93] D. Swaroop, "String Stability of Interconnected Systems: An Application to Platooning in Automated Highway Systems," Ph.D. thesis, University of California at Berkeley, 1994.

[94] D. Zhou, Z. Wang, and M. Schwager, "Agile coordination and assistive collision avoidance for quadrotor swarms using virtual structures," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 916-923, 2018.

[95] T. Gustavi and X. Hu, "Observer-based leader-following formation control using onboard sensor information," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1457-1462, 2008.

[96] X. Dong, B. Yu, Z. Shi, and Y. Zhong, "Time-varying formation control for unmanned aerial vehicles: Theories and applications," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 1, pp. 340–348, 2015.

[97] J. C. Derenick and J. R. Spletzer, "Convex optimization strategies for coordinating large-scale robot formations," *IEEE Transactions on Robotics*, vol. 23, no. 6, pp. 1252–1259, 2007.

[98] N. Michael and V. Kumar, "Planning and control of ensembles of robots with non-holonomic constraints," *The International Journal of Robotics Research*, vol. 28, no. 8, pp. 962–975, 2009.

[99] L. C. Pimenta, M. L. Mendes, R. C. Mesquita, and G. A. Pereira, "Fluids in electrostatic fields: An analogy for multirobot control," *IEEE Transactions on Magnetics*, vol. 43, no. 4, pp. 1765–1768, 2007.

[100] X. Yan, J. Chen, and D. Sun, "Multilevel-based topology design and shape control of robot swarms," *Automatica*, vol. 48, no. 12, pp. 3122– 3127, 2012.

[101] M. Kloetzer and C. Belta, "Temporal logic planning and control of robotic swarms by hierarchical abstractions," *IEEE Transactions on Robotics*, vol. 23, no. 2, pp. 320–330, 2007.

[102] V. G. Santos and L. Chaimowicz, "Hierarchical congestion control for robotic swarms," in 2011 *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4372–4377, 2011.

[103] P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer, "Arbitrary pattern formation by asynchronous, anonymous, oblivious robots," *Theoretical Computer Science*, vol. 407, no. 1-3, pp. 412–447, 2008.

[104] V. Gazi, "Swarm aggregations using artificial potentials and sliding-mode control," *IEEE Transactions on Robotics*, vol. 21, no. 6, pp. 1208–1214, 2005.

[105] J. C. Bezdek and I. M. Anderson, "An application of the c-varieties clustering algorithms to polygonal curve fitting," *IEEE transactions on systems, man, and cybernetics*, no. 5, pp. 637–641, 1985.

[106] D. Shah and L. Vachhani, "Swarm aggregation without communication and global positioning," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 886–893, 2019.

[107] A. Okubo, "Dynamical aspects of animal grouping: swarms, schools, flocks, and herds," *Advances in Biophysics*, vol.22, pp.1-94,1986.

[108] T. Vicsek, A. Czirk, E. Ben-Jacob, I. Cohen and O. Shochet, "Novel type of phase transition in a system of self-driven particles," *Physical Review Letters*, vol.75, no.6, pp.1226, 1995.

[109] F. Liao, R. Teo, J.L. Wang, X. Dong, F. Lin and K. Peng, "Distributed formation and reconfiguration control of vtol uavs," *IEEE Transactions on Control Systems Technology*, vol.25, no.1, pp.270-277, 2017.

[110] X. Li and Y. Xi, "Distributed connected coverage control for groups of mobile agents, *International Journal of Control*," vol.83, no.7, pp. 1347- 1363, 2010.

[111] T. Sugimoto, "A theoretical analysis of formation flight as a nonlinear selforganizing phenomenon," *IMA Journal of Applied Mathematics*, vol.68, no.5, pp. 441-470, 2003.

[112] H.P. Thien, M.A. Moelyadi and H. Muhammad, "Effects of leaders position and shape on aerodynamic performances of V flight formation," *ICIUS 2007*, pp. 43-49, 2007.

[113] J.B.E.O'Malley and R.M.Evans, "Flock formation in white pelicans," *Canadian Journal of Zoology*, vol.60, no.5, pp.1024-1031, 1982.

[114] M. Klotsman and A. Tal, "Animation of flocks flying in line formations," *Artificial Life*, vol.18, no.1, pp.91-105, 2012.

[115] I.L. Bajec, N. Zimic and M. Raz, "Simulating flocks on the wing: the fuzzy approach," *Journal of Theoretical Biology*, vol.233, no.2, pp.199- 220, 2005.

[116] F. R. Hainsworth, "Precision and dynamics of positioning by Canada geese flying in formation," *Journal of Experimental Biology*, vol.128, no.1, pp.445-462, 1987.

[117] Y. Cao, W.Yu, W. Ren and G.Chen, "An overview of recent progress in the study of distributed multi-agent coordination," *IEEE Transactions on Industrial Informatics*, vol.9, no.1, pp.427-438, 2013.

[118] Z. Chen, M. Fan, and H. Zhang, "How much control is enough for network connectivity preservation and collision avoidance?," *IEEE Transactions on Cybernetics*, vol.45, no.8, pp.1647-1656, 2015.

[119] S. Camazine, J. L. Deneubourg, N. R. Franks, J. Sneyd, G. Theraulaz, and E. Bonabeau, "Self-Organization in Biological Systems," Princeton Univ. Press, 2003.

[120] D. Merkle and M. Middendorf, "Swarm intelligence and signal processing," *IEEE Signal Process. Mag.*, vol. 25, no. 6, pp. 152–158, Nov. 2008.

[121] J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1465–1476, 2004.

[122] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3122–3136, Jul. 2008.

[123] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1035–1048, Mar. 2010.

[124] L. Li and J. A. Chambers, "Distributed adaptive estimation based on the APA algorithm over diffusion networks with changing topology," in *Proc. IEEE Statist. Signal Process. Workshop*, Cardiff, Wales, Sep. 2009, pp. 757–760.

[125] N. Takahashi, I. Yamada, and A. H. Sayed, "Diffusion least-mean squares with adaptive combiners: Formulation and performance analysis," *IEEE Trans. Signal Process.*, vol. 8, no. 9, pp. 4795–4810, Sep. 2010.

[126] P. Seiler, "Coordinated Control of Unmanned Aerial Vehicles," Ph.D. thesis, University of California, Berkeley, 2001.

[127] A. Pant, P. Seiler, and K. Hedrick, "Mesh stability of look-ahead interconnected systems," in *IEEE Transactions in Automatic Control*, vol. 47, no. 2, pp. 403-407, February 2002.

[128] G. Antonelli and S. Chiaverini, "Fault tolerant kinematic control of platoons of autonomous vehicles," in *IEEE International Conference on Robotics and Automation*, 2004. Proceedings. ICRA '04, pp. 3313-3318, vol.4, 2004.

[129] R. M. C. Santiago, A. L. De Ocampo, A. T. Ubando, A. A. Bandala, and E. P. Dadios, "Path planning for mobile robots using genetic algorithm and probabilistic roadmap," in *IEEE International Conference on humanoid, nanotechnology, information technology, communication and control, environment management (HNICEM)*, pp. 1-5, 2017.

[130] A. Pandey and D. R Parhi. "Optimum path planning of mobile robot in unknown static and dynamic environments using fuzzy-wind driven optimization algorithm," in *Defence Technology*, vol. 13, no. 1, pp. 47–58, 2017.

Arijit Kumar Halder
05/08/2022