

BIO INSPIRED COOPERATIVE PREY HUNTING FOR MULTI AGENT SYSTEM

BY

MD MUZAKKIR QUAMAR

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

CONTROL AND INSTRUMENTATION ENGINEERING

MAY 2022

This thesis, written by **MD MUZAKKIR QUAMAR** under the direction of his thesis advisor and approved by his thesis committee, has been presented and accepted by the Dean of Graduate Studies, in partial fulfilment of the requirements for the degree of **MASTER OF SCIENCE IN CONTROLS AND INSTRUMENTATION ENGINEERING.**

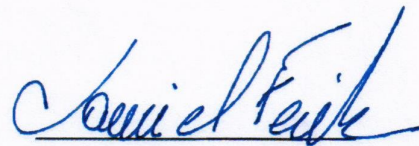


Dr. MUJAHED AL DHAIFALLAH
(Department Chairman)



Dr. SOLIMAN S. AL-HOMIDAN
(Dean of Graduate Studies)

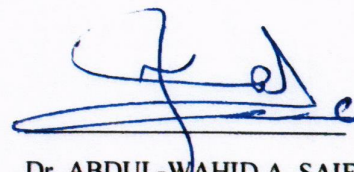
Date:



Dr. SAMI EL FERIK
(Advisor)



Dr. MUJAHED AL
DHAIFALLAH
(Member)



Dr. ABDUL-WAHID A. SAIF
(Member)

© MD MUZAKKIR QUAMAR

2022



In the name of Allah, the Most Gracious, the Most Merciful.

This study is wholeheartedly dedicated to my family who always stood by my side and
have been a source of inspiration and strength.

ACKNOWLEDGEMENTS



All Praise to Almighty Allah for blessing with wisdom, strength, determination, and good health which helped me reach the milestones of my life.

I take this opportunity to thanks the Ministry of Education, Kingdom of Saudi Arabia for giving me an opportunity to pursue my MS in one of the best universities within Kingdom. Likewise, I would also thank the KFUPM imparting world class education, and access to advanced facilities and equipment in class and Laboratory. KFUPM will consistently be recalled in my sweet memories as a home away from home.

I am highly grateful to my thesis advisor and Graduate Coordinator, Prof. Sami El Ferik. I express my wholehearted gratefulness to Prof. Sami for his valuable ideas, direction, and important time from his bustling timetables all through my Masters. I truly value his unassuming nature and generosity which kept me agreeable all through the excursion at KFUPM. I think without his unconditional help this work could not have been conceivable. It was an advantage for me to enter in the realm of exploration under his direction acquiring the abilities needed in this field. Indeed, I say thanks to him for all his help till date and ideally likewise in my prospects. I would wind up saying that it is an honour to work under his supervision, it inspired and benefitted me from his tremendous experience.

I am also grateful to my committee and members, Prof. Abdul Wahed A.Saif and Dr. Mujahed Al Dhaifallah for their valuable suggestions throughout the thesis work and in improvising the quality of my work by exchanging the ideas. I would like to take this

opportunity to thank all the teachers and faculty for their continuous support, guidance, and advice.

I would like to thank my family and friends who encourage me during the hard times for keeping up with this task. Lastly, I am grateful to all respondents and different people for their important collaboration and furthermore to the individuals who directly or by indirectly helped me during this undertaking. Without the assistance of every one of these respected individuals, I would not have had the option to attempt this thesis effectively.

Thanking you.

Md Muzakkir Quamar

TABLE OF CONTENT

ACKNOWLEDGEMENTS	iv
TABLE OF CONTENT	vi
LIST OF TABLES	ix
LIST OF FIGURES	x
LIST OF ABBREVIATIONS	xiii
ABSTRACT.....	xiv
المستخلص.....	xvi
Chapter 1 INTRODUCTION	1
1.1 Problem Statement	4
1.2 Objectives.....	5
1.3 Approach	6
1.4 Organisation	7
Chapter 2 LITERATURE SURVEY	9
2.1 Multi Agent System	9
2.2 Biologically Inspired Network	10
2.2.1 Adaptive Networks	12
2.2.2 Distribution Strategies	13
2.3 Pursuit-Evasion: Swarm of UAVs	13

2.4 Cooperative Prey Hunting	14
2.5 Trajectory controller for Non-Holonomic UAV Models	15
Chapter 3 NONHOLONOMIC UAV MODEL	18
3.1 Introduction	18
Chapter 4 NAVIGATION ALGORITHM FOR UAVS.....	21
4.1 Introduction	21
4.2 The Foraging Phase	22
4.2.1 Fish Prey Algorithm	23
4.3 Evasion Phase.....	35
4.4 Distributed Velocity Estimate for Swarm	37
4.5 Global Estimation of Predator Location and velocity.	38
4.6 Predator Behaviour.....	39
Chapter 5 COOPERATIVE PREY HUNTING BY MULTIPLE PREDATORS.....	40
5.1 Introduction	40
5.2 Motion Control Mechanism for Predators.....	41
5.2.1 State Machine Model	42
5.3 Global Velocity Update for Predator	47
Chapter 6 DESIGN OF TRACKING CONTROLLER.....	48
6.1 Trajectory generation	49
6.1.1 Trajectory Generation in Absence of Threat	50

6.1.2 Trajectory Generation in Presence of Threat	51
6.2 Single UAV Tracking Analysis.....	51
6.2.1 Design of Optimal Fractional Order PID Controller.	52
6.2.2 Control System Implementation.....	57
6.3 Simulation Results	58
6.4 Multiple UAV Tracking Analysis	63
Chapter 7 RESULTS AND ANALYSIS.....	65
7.1 Single Predator Attack	65
7.1.1 Foraging Phase	66
7.1.2 Foraging and Evasion	68
7.2 Single Predator Attack with Obstacle and Wind Turbulence	76
7.2.1 Single UAV Foraging with Obstacle and Turbulence	77
7.2.2 Single UAV Foraging and Evasion with obstacle and Wind Turbulence.	77
7.2.3 Multiple UAV Foraging and Evasion with Obstacle and Wind Turbulence ...	79
7.3 Multiple Predator Attack - Cooperative Hunting	81
7.3.1 Cooperative Hunting Under Wind Turbulence	87
Chapter 8 CONCLUSION AND FUTURE STUDY	88
REFERENCES.....	90
CURRICULUM VITAE	106

LIST OF TABLES

Table 4-1: Assumption Table.....	24
Table 4-2: Estimating node position on the outer boundary of Fragmented groups	34
Table 4-3: Evasion Angle for UAVs in each Quadrant	37
Table 6-1: Gain Parameters FOPID controller.....	58
Table 7-1: Simulation table for single Predator Attack	66
Table 7-2: Simulation table for Multi-Predator Attack- Cooperative Hunting.....	81

LIST OF FIGURES

Figure 1.1: swarming drones, based on the Hunter 2 series of UAS, presented at Unmanned Systems Exhibition and Conference (UMEX 2022) [9].	3
Figure 3.1: UAV Model	19
Figure 4.1 : Air Combat Field	22
Figure 4.2 : Vector representation of Fish nodes and Target	27
Figure 4.3 : Four Regions around Predator based on relative position of fish node	30
Figure 5.1: State Transition Model for Predator	42
Figure 5.2: Predator position with the network center and concept of Outlier	44
Figure 6.1: Control Schematic - Single UAV	48
Figure 6.2: Control decision flowchart for Fleet Navigation	49
Figure 6.3: Generalized Schematic diagram of FOPID controller	53
Figure 6.4: Simple Flow chart structure of particle swarm optimization	56
Figure 6.5: The block diagram of the FOPID control system structure	57
Figure 6.6: Single UAV reference tracking	59
Figure 6.7: Orientation tracking error for single UAV	60
Figure 6.8: Reference tracking for circular trajectory	60
Figure 6.9: Single UAV Reference tracking under wind turbulence	61
Figure 6.10: Orientation tracking error for single UAV under wind turbulence	62
Figure 6.11: Reference tracking for circular trajectory under wind turbulence	62
Figure 6.12: Multiple UAV tracking	64
Figure 7.1: UAV Fleet aligned to forage towards target	66

Figure 7.2: Foraging Behavior 1	67
Figure 7.3: Foraging Behavior 2	67
Figure 7.4: Foraging Behavior 3	68
Figure 7.5: Q1 Attack - Foraging & Evading Behavior 1	69
Figure 7.6: Q1 Attack - Foraging & Evading Behavior 2	69
Figure 7.7: Q1 Attack - Foraging & Evading Behavior 3	70
Figure 7.8: Q1 Attack - Foraging & Evading Behavior 4	70
Figure 7.9: Q1 Attack - Foraging & Evading Behavior 5	71
Figure 7.10: Q1 Attack - Foraging & Evading Behavior 6.....	71
Figure 7.11: Q2 Attack - Foraging & Evading Behavior 1.....	72
Figure 7.12: Q2 Attack - Foraging & Evading Behavior 2.....	72
Figure 7.13: Q2 Attack - Foraging & Evading Behavior 3.....	73
Figure 7.14: Q2 Attack - Foraging & Evading Behavior 4.....	73
Figure 7.15: Q2 Attack - Foraging & Evading Behavior 5.....	74
Figure 7.16: Q2 Attack - Foraging & Evading Behaviour 6.....	74
Figure 7.17: Q3 Attack - Foraging & Evading Behavior	75
Figure 7.18: Q4 Attack - Foraging & Evading Behavior	76
Figure 7.19: Foraging of single swarm in presence of obstacle and wind turbulence.	77
Figure 7.20: Single UAV foraging and evasion with obstacle and wind turbulence- case I	78
Figure 7.21: Single UAV foraging and evasion with obstacle and wind turbulence- case II	78
Figure 7.22: Multiple UAV foraging in the presence of wind and obstacle.....	79

Figure 7.23: Multiple UAV foraging and evasion in the presence of wind and obstacle .	80
Figure 7.24: Case1-Multi-Predator Attack -Frame 1	82
Figure 7.25: Case1-Multi-Predator Attack -Frame 2	82
Figure 7.26: Case1-Multi-Predator Attack -Frame 3	83
Figure 7.27: Case1-Multi-Predator Attack -Frame 4	83
Figure 7.28: Case1-Multi-Predator Attack -Frame 5	84
Figure 7.29: Case1-Multi-Predator Attack -Frame 6	84
Figure 7.30: Case1-Multi-Predator Attack -Frame 7	85
Figure 7.31: Case1-Multi-Predator Attack -Frame 8	85
Figure 7.32: Case1-Multi-Predator Attack -Frame 9	86
Figure 7.33: Case1-Multi-Predator Attack -Frame 10	86
Figure 7.34: Cooperative Hunting Under Wind Turbulence	87

LIST OF ABBREVIATIONS

Abbreviations

UAV	Unmanned Aerial Vehicle
UAS	Unmanned Aircraft System
RPAS	Remotely Piloted Aircraft System
VTOL	Vertically hover, take-off, and landing
DOF	Degree of Freedom
PID	Proportional Integral Derivative
PSO	Particle Swarm Optimization
ATC	Adapt and Then Combine
SLAM	Simultaneous Localization and Mapping
CEP	Complex Event Processing
MAS	Multi Agent System
GPS	Global Positioning System
Fig.	Figure
Eqn.	Equation
PSO	Particle Swarm Optimization
MPC	Model Predictive Controller
AUV	Autonomous underwater vehicle
LMS	Least- Mean Square
CTA	Combine-Then Adapt
FOPID	Fractional order Proportional Integral Derivative

ABSTRACT

Full Name : MD MUZAKKIR QUAMAR
Thesis Title: : Bio-Inspired Cooperative Prey Hunting for Multi Agent
system
Major Field: : Controls and Instrumentation Engineering
Date of Degree: : May 2022

In this thesis, a biologically inspired diffusion adaptation-based algorithm is implemented for swarm of UAVs. The swarm of UAVs in pursuit of food, forage to a desired target exhibiting cooperative behaviour. The swarm is capable of evading in case of a predator attack using Adapt and then combine algorithm. The navigation algorithms are designed to swiftly switch to desired mode of foraging and evasion based on the environmental conditions. Further a cooperative hunting strategy for multi predator is modelled. This is also inspired by the biological behaviour exhibited by some fishes and animals. The hunting strategy is modelled using state transition model.

The swarm and predator are modelled using nonholonomic UAV models knitted in an adaptive network such that the UAVs exhibit biological behaviour during the navigation. This approach is like war of drones in an air combat field. To put in context, in this thesis we aim to use bio inspired algorithms for the navigation of UAVs along with design of an optimal controller to ensure the trajectory tracking of the UAVs. The complete navigation is designed to emulates the scenarios of

warfare, where a group of UAVs are on a mission of accomplish a target and is attacked by enemy UAVs. To approximate the environment with real world scenario we add unstructured obstacle such as buildings or trees, which the agents avoid colliding during the navigation. We also added a wind turbulence zone that emulates storm or typhoon in real world to increase the complexity in the environment. The fleet of UAVs is capable of evading the attack using cognitive evasion algorithm. Similarly, the predator UAVs also form groups and cooperate to trap and attack the swarm considering its enemies. Simulations are performed for both single and multi-predator attack. The results exhibit the effectiveness of the work.

المستخلص

الاسم:

محمد مذكر قمر

عنوان الرسالة:

اصتياد الفريسة الجماعية مستوحيا بيولوجيًا للنظام متعدد العوامل

التخصص الرئيسي:

هندسة الضوابط والأجهزة

تاريخ الشهادة:

مايو 2022

في هذه الأطروحة، يتم تنفيذ خوارزمية قائمة على التكيف مع الانتشار المستوحى بيولوجيًا لسرب الطائرات بدون طيار. سرب الطائرات بدون طيار بحثًا عن الطعام وللبحث عن هدف مرغوب فيه يظهر سلوكًا تعاونيًا. السرب قادر على الهروب في حالة هجوم مفترس باستخدام موازن ثم دمج الخوارزمية. تم تصميم خوارزميات التنقل للتبديل بسرعة إلى الوضع المطلوب للبحث عن الطعام والتهرب بناءً على الظروف البيئية. علاوة على ذلك، تم وضع نموذج لاستراتيجية الصيد التعاوني للحيوانات المفترسة المتعددة. هذا مستوحى أيضًا من السلوك البيولوجي الذي تظهره بعض الأسماك والحيوانات. تم تصميم استراتيجية الصيد باستخدام نموذج انتقال الحالة.

يتم نمذجة السرب والحيوانات المفترسة باستخدام نماذج الطائرات بدون طيار غير المنضبطة المحبوسة في شبكة تكيفية بحيث تظهر الطائرات بدون طيار سلوكًا بيولوجيًا أثناء التنقل. هذا النهج يشبه حرب الطائرات بدون طيار في مجال القتال الجوي. لوضعها في السياق، نهدف في هذه الأطروحة إلى استخدام خوارزميات مستوحاة من الأحياء للتنقل في الطائرات بدون طيار جنبًا إلى جنب مع تصميم وحدة تحكم مثالية لضمان تتبع مسار الطائرات بدون طيار. تم تصميم الملاحظة الكاملة لمحاكاة سيناريوهات الحرب، حيث تقوم مجموعة من الطائرات بدون طيار بمهمة تحقيق هدف ويتم مهاجمتها

من قبل الطائرات بدون طيار للعدو. لتقريب البيئة مع سيناريو العالم الحقيقي، تم إضافة عقبات غير منتظمة مثل المباني والأشجار، والتي يتجنب الوكلاء الاصطناعي بها أثناء التنقل. أضفنا أيضًا منطقة اضطراب الرياح التي تحاكي العاصفة أو الأعاصير في العالم الحقيقي لزيادة التعقيد في البيئة. أسطول الطائرات بدون طيار قادر على التهرب من الهجوم باستخدام خوارزمية التهرب المعرفي. وبالمثل، تشكل الطائرات بدون طيار المفترسة أيضًا مجموعات وتتعاون للقبض على السرب ومهاجمته بالنظر إلى أعدائه. يتم إجراء عمليات المحاكاة لكل من الهجوم الفردي والمتعدد للمفترسات. تظهر النتائج فعالية العمل.

CHAPTER 1

INTRODUCTION

Unmanned aerial vehicles (UAVs) typically refer to an aircraft that does not have a pilot and operates through a combination of technologies including computer vision, artificial intelligence, object avoidance technique, and others. The UAVs have been gaining popularity due to its small size, ease of operation, vertically hover, take-off, and landing (VTOL) low production cost, ease of operation and most importantly unmanned operation in high-risk operation and wars. Due to its ability to perform wide range of civil and military operations such as survey, monitoring, security inspection, search and rescue, delivery operations, disaster management, attacks in warfare, etc it is intensely researched. The researchers are showing interest to develop more advanced UAVs with better control, increased intelligence, and cooperative manoeuvring. According to data extracted from Scopus [1], UAV/UAS/RPAS/drone related research exponentially rose since 2014, while more than 80k papers published with these keywords since 2001.

Today every country is investing heavily in drone technology. As per reports[2] the global drone market is expected to grow by \$21.01 billion at a compound annual growth rate of 14.42% between 2021-2025. A large percentage of this investment is dedicated to military needs. The global military drone market is expected to grow at 6.6% during this period [3]. The military drones are utilised for military use such border surveillance, combat operations, communication, delivery, and anti-terrorism weaponry. Since the military operation is performed in regions with no or minimum information and access. Thus, it is

required for the drones to be robust, precise, adaptive, and intelligent during its operations in such militarised zone. While it is easy to control and operate a single UAV, however, controlling and coordinating the operation of multiple UAVs is complex [4]–[6].

One of the recent advancements in the military use of the UAVs is the swarming technology. Swarming typically refers to a mode of operation, where multiple autonomous systems act as a coherent unit by coordinating their actions. Military applications require networked multi-UAV system to perform practically, optimally, and reliably under changing mission requirements. A 2018 study conducted by U.S Army [7] suggest that “swarming would make attack drones at least 50% more lethal while decreasing the losses they took from defensive fire by 50%”. A UAV combatting in group pose many challenges for smooth control and coordination between the group members. The increasing use of UAV swarming technique integrated with controller design in military operation makes it an active area of research. We envision future military combats will be carried out using swarming of UAVs. In other words, military supremacy will be determined by the potential UAV technology integrated with swarming technique. A statement by Dr. Jemin George of the U.S. Army Research Laboratory bolsters our claim. As per Dr. George, “Future multi-domain battles will require swarms of dynamically coupled, coordinated heterogeneous mobile platforms to overmatch enemy capabilities and threats and hence The US Army is looking to swarming technology to be able to execute time-consuming or dangerous tasks”[8].



Figure 1.1: swarming drones, based on the Hunter 2 series of UAS, presented at Unmanned Systems Exhibition and Conference (UMEX 2022) [9].

A key enabling technology for swarms is the family of algorithms that allow the individual members of the swarm to communicate coordinate and plan trajectories efficiently while accomplishing a mission. The origin of swarm modelling was inspired by the natural behaviour of the biological group's (schools of fish, flocks of birds, swarm of ants, etc.) responses to predators or prey. The collective behaviour of animals has been studied in several literature [10] , bird flocking [11], school of fish [12], bees swarm [13], or wild predators [14]. M. Chen, et al. [15] developed a flocking control for a multi agent systems by identifying team leaders and followers. S. El Ferik et al. [12] [16] proposed a set of biologically inspired control algorithm that enables the multi-agent system to operate by switching through different navigation algorithms. The navigation algorithms are based on concept of adaptive networks along with diffusion and adaptation technique. A combination of path tracking, integrated with diffusion adaptation and potential field

strategies has been implemented to ensure stability while switching, formation maintenance, cohesive motion, mission accomplishment, as well as evasion if under threat. However, when it comes to cooperative hunting, with multiple UAVs we find very few research in literature where trajectory planning, and control is implemented. In [17] the author uses adaptive network model with diffusion adaptation for cooperative prey hunting for point mass object, but without control. Juan Ze in [18] multi-UAV with prey hunting with MPC controller with static and moving obstacle avoidance, but the behaviour of predator or prey is not based on bio inspired technique. A few other research used optimization-based techniques such as PSO [19] or wolf-pack-particle-based model [20] for cooperative hunting. Thus, we propose cooperative hunting for MAS trying to trap and attack mobile UAVs using state transition model along with controller design.

1.1 Problem Statement

Based on our reviews we found a lot of work is done on design of control techniques for tracking nonholonomic UAVs for both single agent and multi agent system. However, when it comes to integration of bio inspired algorithms along with nonholonomic UAVs very little work is done. In [12] the author integrated bio inspired algorithm along with nonholonomic UAV model for swarm of UAVs foraging and evading under threat of single predator. The work was extended in [16] for multi predators. However, when it comes to cooperative hunting there is no current work to the best of our knowledge that integrates the bio inspired algorithms with nonholonomic UAVs for multi predator attack using cooperative hunting strategy along with design of an optimal controller.

Therefore, inspired by the work in [17] and [12], in this thesis we propose a bio inspired cooperative prey hunting for multiple predators. The bio inspired algorithm is based on adaptive network with diffusion and adaptation. The cooperative hunting strategy is modelled using state transition model. An optimal Fractional order PID controller tuned using PSO is designed to ensure nonholonomic model track the trajectories designed using bio inspired algorithm. To add complexity to the system we introduce obstacle and region of wind turbulence in the environment. The controller ensures zero tracking error even in the region of high wind turbulence making the design robust.

1.2 Objectives

The following objectives are achieved in this thesis work,

1. The design of navigation algorithm for nonholonomic group of UAVs based on [21].
2. Design of bio inspired algorithm based on adaptation and diffusion strategy for swarm of UAVs mimicking the behaviour of school of fish, foraging in search of food and evade when under threat based on [12].
3. The design of cooperative hunting strategy based on state transition model as in [17].
4. Integration of Bio inspired algorithm with the non-holonomic UAV models.
5. Design of robust optimal controller to ensure trajectory tracking based on designed navigation algorithm during foraging and evasion.
6. Use of potential field strategy to ensure collision avoidance among UAVs.

7. Design of obstacle avoidance mechanism for both swarm and predator agents during navigation.
8. Design of a wind turbulence region that emulates the storm in real world scenario.
9. Finally, the simulation of the nonholonomic models based on the above designed navigation algorithm and controllers for both single predator attack and multi predator cooperative attack.

1.3 Approach

In this thesis, we propose distributed navigation algorithm for a group of UAVs (swarm). The Navigation algorithm is biologically inspired by the behaviour of school of fishes. The algorithm is based on diffusion and adaptation strategy [21]. The navigation algorithm allows the UAVs to switch to multiple path planning strategy based on the predator attack.

We define two cases for predator attack. A single predator attack and a multi predator attack with cooperative hunting strategy. Under a single predator attack the swarm UAVs foraging on a mission to accomplish a target, are capable of evading. The swarm UAVs are strongly connected adaptive network with learning capability. During foraging the Swarm UAVs move coherently and maintains a safe distance between them, to avoid collision. During evasion the swarm of UAVs switch to cognitive evasion technique to escape danger and once out of danger they reunite and forage towards target. Thus, swarm of UAVs switch between the two modes of navigation, foraging and evasion based on the presence of predator.

Under multi predator attack we design a cooperative hunting strategy for the predators. The cooperative hunting strategy is also biologically inspired. The cooperative hunting strategy

is modelled using state transition model [22]. The predators switch to four different modes - chase, encircle, trap, and attack, during the hunting process. The predator UAVs chase the foraging UAV fleet, encircle it, and then attack. During attack the fleet of UAVs switch to evasion.

The fleet of UAVs for both the predator and swarm is composed of nonholonomic robots under Pfaffian constraint. An optimal FOPID controller is used to control the trajectory of the UAV robots during the navigation. The simulation results have been performed using python programming to demonstrate the results.

1.4 Organisation

The coming work in the thesis is organised as follows. Chapter 2 discusses the literature survey related to the research. Previous work related to multi agent system is discussed. Then the research in past related to bio inspired network is discussed. A subsection for adaptive network and diffusion strategy is discussed as well. We further focus on the work in past for pursuit evasion for MAS. And finally, the literature related to cooperative hunting strategies is discussed. In chapter 3 we discuss the nonholonomic UAV model, its constraint and mathematical formulation.

In chapter 4, we design the navigation algorithm for Swarm of UAVs. First, we discuss the fish prey algorithm for UAVs based on adaptation and diffusion. Then the motion control algorithm for swarm of UAVs during foraging and evasion is discussed.

In chapter 5, we discuss the cooperative prey hunting. We define the state transition model for the predator attack. Then the motion control algorithm for the predator in each state is

modelled. In chapter 6 we design an optimal controller and exhibit the trajectory tracking of single and multiple UAVs. While in chapter 7, we discuss the simulation results for each case. Finally, we conclude our work with the results achieved and future work that can be incorporated in chapter 7.

CHAPTER 2

LITERATURE SURVEY

2.1 Multi Agent System

The notable feature of an UAV is that it cannot perform certain task or accomplish a mission individually in a large area such as performing search operations or surveillance. Therefore, multiple UAVs need to be employed in collaboration among each other to achieve the desired results. The UAVs working in a cooperative manner like a swarm provide powerful capabilities to perform complex tasks, critical mission and can perform challenging operations with easily in a remarkable pace with accuracy compared to individual UAV. These could be the reasons that the group of UAVS as a Multiagent system have started to gain high prominence as the swarm of UAVS in a Multiagent system communicate and collaborate efficiently.

Schmuck et al.[23] used a group of small UAVs to act as Multiagent system that collaborates using a monocular simultaneous localization and mapping (SLAM) application. Tian et al. [24] used a fleet of UAV to perform search operation in dense forest. The UAVs collaborated to explore a thick forest area where GPS is unreliable. They used wireless communication and onboard computations techniques to communicate. Ashraf et al. in [25] proposed an online, collision avoidance and navigation system for swarms of UAVs. They used the geographical locations of the UAVs and designed a system to

successfully detect static or dynamic obstacles and avoid collision. It had three main components: a complex event processing (CEP) and collision prediction module, second, a mutually exclusive locking mechanism, and third, a collision avoidance mechanism.

2.2 Biologically Inspired Network

A network designed to mimic the natural behaviour of animals and fishes is called as Biologically inspired network. The fascinating behaviour exhibited by schooling fish, swarming bees, and flocking birds have always attracted researchers to study them and model solution to complex real-world problems based on that. Xumei Fan et.al in [26] have presented a review on biologically inspired networks and algorithms. They have broadly classified the biologically algorithms into four categories. These are evolutionary based, swarm intelligence, Ecology based, and multi-objective algorithms.

The use of biologically inspired networks is highly diversified and not just limited to computing and robotics, rather it is also used in industry such as transportation and manufacturing. Xiaoge Zhang et. al in [27] designed a transportation network based on the biological behaviours of acellular slime mould *Physarum polycephalum*. A *Physarum* converges to shortest path in the network irrespective of the initial structure of the network or of the initial mass distribution. The author used the foraging behaviour of slime mould *Physarum* to solve the network design problem and optimize the transport flow between two cities. The slime mould model provided the shortest path and optimized the overall transportation cost and traffic congestion thus improving the overall efficiency of transportation model. Paulo Leitao et. al in [28] provided a brief review on the use of

bio-inspired model to solve complex problems in manufacturing process. The bio inspired models for Multiagent system used in manufacturing is mainly focussed. The manufacturing process essentially consist of multi agents, the research explains how bio inspired model integrated with MAS can help design more modular, intelligent, adaptive, and flexible manufacturing process. The author concluded with an experimental case study for bio inspired MAS in a manufacturing process. A full-size manufacturing system with seven workstations (assembly robots, quality control units and load/unload units) connected through a conveyor belt is used. A bio inspired control system using the concept of potential field for task allocation and dynamic pallet routing is used. The experimental results showed an improvement in production time gain of 10% compared to contract-net approach typically used in multi-agent control solutions for resource allocation in manufacturing process.

In [29] Wei Wang et. al presented a bio inspired communication system for underwater robots. Inspired by the behaviour of weakly electric fishes which uses electric field to communicate. The author designed an analytical model for electrocommunication between underwater nodes. The communication was tested with two leaders and a follower in still water, flowing water, and water with obstacle. The underwater leader follower robots were able to successfully communicate and achieved motion synchronisation in a 3D environment.

In [30] Nunzia Palmieri et. al uses bio inspired algorithms for the coordination of mobile robots exploring targets in an unknown environment. The swarm of robots are distributed over a region and use bio inspired metaheuristic approach to explore the target. The state action of robot is defined in 5 states, exploration, coordination, recruitment, waiting and

execution. A robot first explores the target, and once it reaches the target it becomes a coordinator and signals other robots in its vicinity by sending packets to reach the target. The recruited robot does same and move towards the target. Once all the robot reaches the target the execution is performed. This strategy can be best suited for search and rescue operation where robots explore an unknown target in an unknown environment.

2.2.1 Adaptive Networks

Adaptive network is a collection of agents which interact locally and diffuse information through the network to achieve a global target. The repeated interaction and diffusion of information solves the optimisation problem in a decentralised manner. Some of the key features of adaptive network is scalability, robustness to node and link failure and ability to learn through cognitive behaviour. Some of the promising work in the field of adaptive networks can be found in [31]–[36].

Ali Sayed in [37] wrote a detailed survey on adaptive networks. In the survey he discussed in detail stochastic Gradient optimization, risk and loss function, conditions on gradient noise conditions, Mean square error stability and performance. He also discussed the adaptation and learning for single agent and multi agent network individually. The distributed strategies for network interaction, adaptation and learning is also discussed in detail. He further discussed the performance analysis and limitations of distributed solution using statistical tools.

Adaptive network forms the backbone to model bio inspired algorithms. Some of the notable work using adaptive network and diffusion strategy for bio inspired network can be found in [21], [34], [38]– [42].

2.2.2 Distribution Strategies

In distributed mode of operation, the agents share information with their immediate neighbours within the topology. Ali Sayed in [43] discussed several distribution strategies, namely incremental, consensus and diffusion. In the incremental network the information is passed from one node to another in a cyclic path until all agents are covered. This process is repeated in each iteration. However, the incremental strategy has several drawbacks such as it is sensitive to link failure, each node can only receive information from its preceding node and pass to just one succeeding node, and computationally costly as for each iteration N incremental steps is performed. The consensus and diffusion strategy do not suffer from such limitations but in terms of performance diffusion strategy outperforms consensus strategy [44]. In [45] Cattivelli et. al, presented the different diffusion and consensus based LMS algorithms, including the (non-hierarchical) ATC and CTA diffusion LMS algorithm, the consensus-based D-LMS and their performance was analysed. It is observed that the ATC version of diffusion LMS outperforms other algorithms in terms of convergence. Diffusion strategies can be used to solve estimation tasks in cooperative networks with multi nodes with adaptation and learning abilities, indicating that they can be used to mimic the biological behaviours. Sayed in [21], [33] [22] used diffusion strategy to mimic the biological behaviours.

2.3 Pursuit-Evasion: Swarm of UAVs

Pursuit Evasion is a more specific theory where Multiagent system can chase an external agent or evade from an external threat. This Pursuit-evasion theory involving Swarm of UAVs have been studied in detail by Krishnamoorthy et al. [46] where the author uses

multiple UAVs to search a moving target. Ground sensors are used to trigger the search by each UAV. Alexopoulos et al [47], proposed a pursuit-evasion game with multiple UAVs and applied a game-theoretical approach to find a solution. They did experiments using multiple real outdoor flight and successfully validated the proposed solution. Fargeas et al. [48] solved the problem of trajectory planning and navigation for a fleet of UAVs performing surveillance and pursuit. The UAVs rely on communicating with ground sensors to detect potential intruders. They propose a heuristic algorithm to coordinate the UAVs during surveillance & pursuit and show its effectiveness through simulations. Fu et al. [49] proposed a ground target pursuit algorithm using fixed-wing UAVs. They propose a method that generates waypoints and direct the UAV to the latest waypoint. Camci et al [50], implemented ML in pursuit-evasion scenarios. However, they proposed the use of RL to tune type-2 Takagi–Sugeno–Kang fuzzy logic controllers (TSK-FLCs) and not the ML in Navigation or counter evasion.

2.4 Cooperative Prey Hunting

Cooperative prey hunting is an extension to flocking behaviour of swarms. Just as swarms move flock together in search of food, the predator groups also move in group to surround and attack. A bio-inspired algorithm for hunting in a multi robot scenario is presented in [51], where the author uses cooperative hunting strategy for an evading robot. The author uses “Corner and Drag” algorithm to corner the prey to a corner and hunt. The robot uses master slave approach, while hunting. During the search operation each robot independently search for an evader, however the robot that first locates an evader becomes master and flags all the other robots to join it to corner the evader.

Oussama Hamed et. al in [52] presented a cooperative hunting strategy for multi robots based on wolf swarm Algorithm. The multi robot formation is divided into leader, antagonist, and follower, depending upon the location of prey. The antagonist wolf does not follows the leader and has its own cognitive way of searching prey. The antagonist wolf listens to leaders howl and estimates its position, and searches for prey in other direction. The cognitive behaviour of the antagonist reduces the research space and helps to quickly find the target.

In [53] Mingzhi Chen et.al, presented a cooperative hunting for multiple AUVs in 2D and 3D model. They used a hybrid path planning based on GBNN (Glasius bio-inspired neural network) and belief function for hunting a moving target. The AUVs form alliance to hunt target and avoid collision. A few more works where cooperative hunting strategy is implemented can be found in [54] which uses macro mathematical model formulated by rate equations. A potential field based PSO approach for multi robot is used in [55] to search and hunt the targets. A Lyapunov based control strategy is presented in [56] multi robot cooperative hunting. A two-layer Fuzzy inference approach is used in [57] for multi robot hunting. The first layer devices the hunting strategy based on local information of the evasion, and the second layer enables the hunting. A real time cooperative hunting for mobile robots using Bio-inspired neural network approach can be seen in [58].

2.5 Trajectory controller for Non-Holonomic UAV Models

UAVs belong to class of nonholonomic models. A common problem associated with this class of models is their inability to be controlled by a static feedback controller. This is mainly because the nonholonomic models have an underactuated structure and fail to

satisfy Brockett's condition [59], [60]. Therefore, it is important for us to discuss the controller design available in the literature for the trajectory tracking of nonholonomic models. In general, we discuss the control techniques for the trajectory tracking of different nonholonomic models which can be implemented on planar UAVs, a subset of nonholonomic class of models.

Different control strategies can be found in the literature to solve the trajectory tracking of nonholonomic models. In [61] a hybrid Lyapunov based backstepping approach is used to solve the trajectory tracking problem for unicycle model. A Lyapunov based controller is designed to stabilize the robot's orientation angle error while backstepping approach is used to drive the position error to zero. In [12] backstepping control design is used to ensure the trajectory tracking of non-holonomic unicycle mobile robots. An Adaptive backstepping approach for trajectory tracking of unicycle model is used in [62]. A robust sliding mode controller (SMC) is designed in [63] to track the navigation algorithm for the perturbed kinematic model of an Unicycle Mobile Robot. An integral SMC for the trajectory tracking of wheeled mobile robots with uncertainty in parameters is proposed in [64]. An adaptive SMC for trajectory tracking of nonholonomic wheeled mobile robot under kinematic and dynamic uncertainties is designed in [65]. While Feedback linearization-based trajectory tracking of nonholonomic mobile robots is designed and validated with experimental results in [66], [67].

Several researchers have considered using control designs for tracking control of nonholonomic mobile robots using Fuzzy logic control [68], [69]. A super twisting algorithm was designed in [70] where the tracking challenge is converted into two system stabilization. The synthesis of the proposed controller is given in terms of Linear Matrix

Inequalities while the convergence proofs are based on the Lyapunov and Input-to-State Stability theory. A nonlinear disturbance observer-based model predictive control for path following problems of wheeled mobile robots with input disturbances is proposed in [71]. An optimal trajectory tracking control using geometric approach is employed in [72] [73]. The error in the positions and velocities between the trajectory of a non-holonomic system and the desired reference trajectory is minimized by minimizing a cost function. While the conditions for the existence of extrema are determined by the Portraying Minimum Principle.

A few research employed intelligent techniques to optimize the parameters of conventional PID controller. In [74] neuro-PID control and fuzzy-neuro-PID control technique is used to control the trajectory of wheeled mobile robots. An intelligent control design based on optimal control theory, adaptive neural network system and robust control technique is developed in [75]. The trajectory tracking problem is solved using optimal control technique while the neural network approach is employed to approximate the nonlinear function in the optimal control law. The robust adaptive controller is then used to adaptively estimate an unknown upper bound of the time-varying parametric uncertainties, external disturbances, and approximation error of the neural network system. The stability is proved using optimal control theory and Lyapunov function.

CHAPTER 3

NONHOLONOMIC UAV MODEL

3.1 Introduction

In this work the UAVs are nonholonomic moving in a 2D plane. Being nonholonomic means the system has nonholonomic constraints, i.e., velocity constraint cannot be integrated to give positional constraint. While a holonomic system is one where the velocity constraints are integrable. In other words, nonholonomic constraints are constraints on velocity and they do not reduce the configurational space whereas, holonomic constraints are constraint on velocity and reduce the space of the configuration and thus the DOF.

We consider a unicycle mobile robot with nonholonomic constraints. Being nonholonomic UAVs they are constrained by their velocity and cannot make sudden changes in their velocities, and thus positions, and are difficult to control. The kinematic model of planar unmanned aerial vehicle consists of position and orientation and two input velocities. The mathematical modelling for the nonholonomic UAV model on plane is similar to unicycle model [76] and can be expressed as below.

$$\begin{aligned}\dot{x} &= v_1 \cos \theta \\ \dot{y} &= v_1 \sin \theta \\ \dot{\theta} &= v_2\end{aligned}\tag{3-1}$$

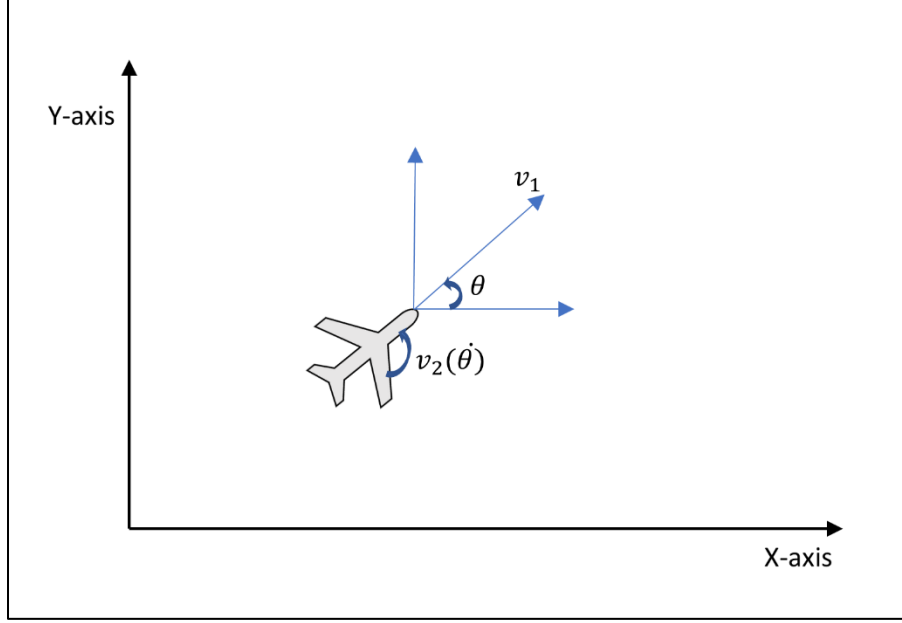


Figure 3.1: UAV Model

where $q = [x \ y \ \theta]^T$ be the generalised coordinate. where (x, y) are the position coordinate of the UAV in plane, θ is the heading angle. And v_1 and v_2 are the forward linear velocity and angular velocity respectively.

The Equation in (3-1) can be rewritten as below,

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta \\ \sin \theta \\ 0 \end{bmatrix} v_1 + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} v_2 \quad (3-2)$$

It can be simply noted that, $\dot{q} = [\dot{x} \ \dot{y} \ \dot{\theta}]^T$ has a one-dimensional null space, which represents the kinematic constraint on the model:

$$[\sin \theta \quad -\cos \theta \quad 0] \dot{q} = 0 \quad (3-3)$$

The constraint is called as Pfaffian constraint [12], such that:

$$D(q)\dot{q} = 0 \quad (3-4)$$

Where $D(q) = [\sin \theta \quad -\cos \theta \quad 0]$

This implies the system has zero lateral velocity, and all available velocities are within the null space of the matrix $D(q)$.

Let the generalised velocities in the null space of $D(q)$ is given by:

$$\begin{bmatrix} \cos \theta \\ \sin \theta \\ 0 \end{bmatrix} v_1, \quad \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} v_2 \quad (3-5)$$

Such that,

$$\begin{aligned} [\sin \theta \quad -\cos \theta \quad 0] \begin{bmatrix} \cos \theta \\ \sin \theta \\ 0 \end{bmatrix} v_1 &= 0, \\ [\sin \theta \quad -\cos \theta \quad 0] \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} v_2 &= 0 \end{aligned} \quad (3-6)$$

Therefore, $\dot{q}$ can be represented as:

$$\dot{q} = \begin{bmatrix} \cos \theta \\ \sin \theta \\ 0 \end{bmatrix} v_1 + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} v_2 \quad (3-7)$$

Which shows the interchangeable relationship between Pfaffian constant and state form model.

CHAPTER 4

NAVIGATION ALGORITHM FOR UAVS

4.1 Introduction

In this section, we discuss in detail the proposed navigation algorithm for the UAVs. The navigation algorithm is designed based on the real time position of the nodes with each other and the external environment. The external environment which is prone to attack from threat or obstacles of the nearness to the target. We will discuss in detail regarding each phase.

We consider a group of UAVs termed as swarm. The individual nodes in the swarm are friendly to each other and move in group maintaining a safe distance. The aim of the swarm is reaching the target while protecting itself from an external attack. In real war scenario, it is similar for a set of drones to reach the target safely. The attacker UAV has the similar kinematics as a single node in swarm. The attacker UAV termed as predator is on its mission to attack the nearest node in the swarm. The swarm is aware of the position of the predator and plans evasion to protect itself from the attack. In real application we will require GPS installed in the nodes and proximity sensors for distributed location calculation and velocity measurements.

Our aim in designing the navigation algorithm for UAVs is to mimic the biological behaviour of the fishes, while foraging in search of food, or evading in case of attack. We

will explore the possible fish behaviours in different modes and design algorithms to define trajectory for the UAVs.

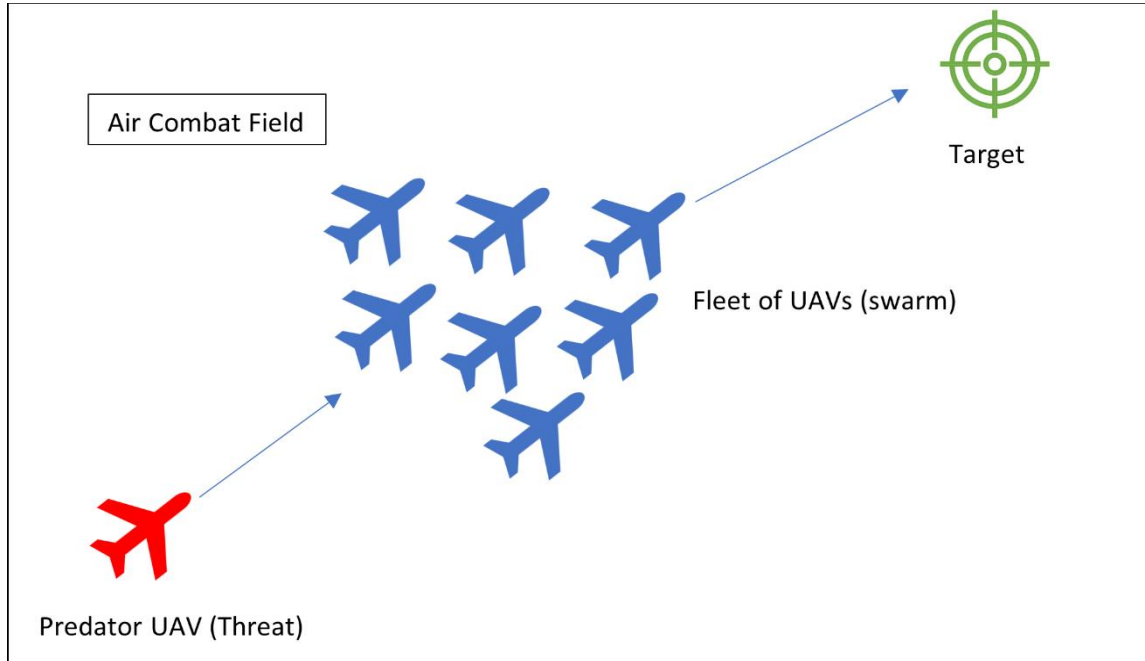


Figure 4.1 : Air Combat Field

The two possible modes of navigation observed from the swarm fishes are:

1. The Foraging Phase
2. The Evasion Phase

We will discuss the two phases for the swarm of UAVs in the following sections.

4.2 The Foraging Phase

The foraging phase corresponds to a scenario where the fishes move in search of food. In this navigation phase there is no immediate danger for the fishes. Thus, we would like the swarm of UAVs to move towards the target in foraging phase just like the fishes move. To model this, we use the fish- prey algorithm as in [77].

4.2.1 Fish Prey Algorithm

Fish prey algorithm is designed based on the concept of adaptive network. An adaptive network consists of closely interactive nodes. The nodes have learning and motion abilities which allows them to interact locally and solve the challenges of distributed inference and distributed processing in real time. The nodes communicate with its neighbours which are mobile while the node itself is mobile. Thus, it creates a network topology that is adaptive. The organizing behaviour of swarm can be easily modelled using adaptive algorithms.

There are various algorithms such as Adapt Then Combine [21], [77], combine Then Adapt [78], probabilistic diffusion algorithms [79], which can be employed for an adaptive network to achieve some cognitive behaviour. In this work we employ diffusion adaptation algorithm [12] namely “Adapt Then combine” to explain how each fish in the swarm contribute to achieve the desire schooling behaviour while foraging or evasion.

A parallel comparison of the nodes in the network can be made with a single fish in the school, and or a UAV in the fleet. In the subsequent subsections we will interchangeably use term the term node, fish, and UAVs to represent the nodes in the adaptive network.

We make the following assumptions as in **Table 4-1** in the proceeding work.

Table 4-1: Assumption Table

<i>S/N</i>	<i>Assumptions</i>
1.	We assume GPS and proximity sensors be used for real experiment.
2.	The UAVs are aware of its neighbourhood, i.e., position and orientation of other UAVs
3.	Each UAV is aware of the target location individually.
4.	Every UAV is locally aware of the position and orientation of the enemy UAV but the evasion occurs when enemy is in the declared vicinity.
5.	The UAVs are assumed to have physical dimensions, with motion governed by physical laws [80], [81].
6.	The food target is assumed to be stationary while the enemy UAV is mobile.
7.	The UAV used is a 2-D model and ignores the aerodynamic considerations of a physical 3-D UAV. And all the action space is in 2-D plane.

A. Diffusion Adaptation

Consider a group of N nodes distributed over a region. Each node denoted by k , is independent and distributed at each instant of time. If node k , shares some information with its neighbouring nodes, then the set of nodes including node k , is called its neighbourhood denoted N_k . Each node at time instant i , evaluates a scalar random process $d_k(i)$, a regression vector correlated to $\bar{u}_{k,i}$. Where, $\bar{u}_{k,i}$ is also a random process representing some information of node k at time i . The objective of each node is to estimate a vector parameter $w_{k,i}$, in real time and in a fully distributed manner. This is achieved using equation (4-1)

$$d_k(i) = \bar{u}_{k,i}w_{k,i} + n_k \quad (4-1)$$

Where, n_k is random disturbance usually zero mean white noise, independent of other variables.

In a distributed network, a node passes data to a subset of nodes in its neighbourhood, which eventually passes data to rest of the connected nodes and so on. Two nodes are said to be connected if there is direct communication between them. The network aims to find the vector parameter $w_{k,i}$, by minimizing the global objective function [77].

$$\mathcal{J}_{global}(w) = \sum_{k=1}^N E|d_k(i) - \bar{u}_{k,i}w|^2 \quad (4-2)$$

Where E represents Expectation of the process. N is the number of nodes.

To solve the diffusion adaptation problem, we have used ATC diffusion algorithm proposed by Ali Sayed in [82], [83]. The ATC algorithm is a two-stage process, the adaptation stage and combination stage. The ATC algorithm assigns a non-negative real

weight, $\alpha_{l,k}$ to communication links set up between the nodes k , and its neighbours l , where $k = 1, 2 \dots N$, and $l \in N_k$. The weight $\alpha_{l,k}$ must satisfy,

$$\sum_{l=1}^N \alpha_{l,k} = 1, \quad \alpha_{l,k} = 0 \quad \forall l \notin N_k \quad (4-3)$$

During adaptation stage each nodes k , interacts within its neighbourhood to adapt locally. Thus, each node updates to current state from its previous state. In our case the state refers to the position vector of the UAVs at time i . while in the combination stage, the node uses the intermediate values of the neighbour nodes obtained during adaptation stage and combines it with the weights obtained in Eqn. (4-3).

The ATC algorithm updates its weight at each instant of time i , according to the equation below given in [83],

$$\begin{aligned} \psi_{k,i} &= w_{k,i-1} + \mu_k \bar{u}_{k,i}^T [\hat{d}_k(i) - \bar{u}_{k,i} w_{k,i-1}] \\ w_{k,i} &= \sum_{l \in N_k} \alpha_{l,k} \psi_{l,i} \end{aligned} \quad (4-4)$$

Where, $w_{k,i}$ is the vector parameter which ATC tries to compute. μ_k is nonnegative step size applied by node k , $\hat{d}_k(i)$ is a linearly regressed vector form of $d_k(i)$, $\bar{u}_{k,i}$ as stated above in Eqn. (4-1).

B. Measurement Model

The measurement model is used to design the vector relationship between the fish nodes and its food target in a Cartesian coordinate system. We then will use the same model for our UAV model. Let the food target be denoted by location vector w^o as shown in **Figure 4.2**. The fish nodes are pursuing the target and move towards it. The distance between the fish node k and the target vector w^o at time instant i , is given by $d_k^o(i)$. While $\bar{u}_{k,i}^o$ is the unit direction vector pointing from the fish location to the target.

$$d_k^o(i) = \bar{u}_{k,i}^o(w^o - x_{k,i}) \quad (4-5)$$

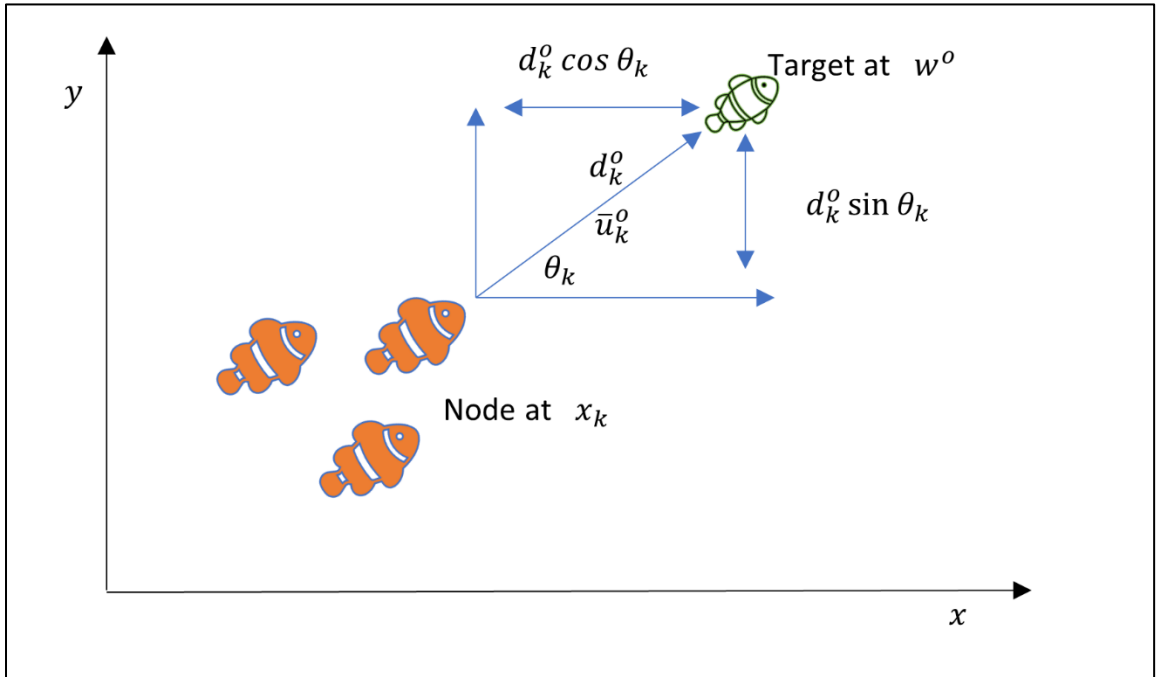


Figure 4.2 : Vector representation of Fish nodes and Target

The location vector of the fish node is given by $x_{k,i}$, which changes with time over the surface. $\bar{u}_{k,i}^o$ is the unit vector direction from fish node $x_{k,i}$ towards target vector w^o . The unit vector $\bar{u}_{k,i}^o$ is defined in terms of azimuth angle $\theta_k(i)$.

$$\bar{u}_{k,i}^o = [\cos \theta_k(i) \quad \sin \theta_k(i)] \quad (4-6)$$

The underwater environment has some disturbances and perturbations in form of waves. Therefore, we introduce noise to bring accuracy in the modelling. When considering the UAV flying in space, the noise can be introduced as wind disturbance.

The noise disturbance can be modelled as in Eq. (4.7)

$$\begin{aligned} \bar{u}_{k,i} &= \bar{u}_{k,i}^o + n_{k,i}^{\bar{u}} \\ d_k(i) &= d_k^o(i) + n_k^d(i) \\ \hat{d}_k(i) &\cong d_k(i) + \bar{u}_{k,i} x_{k,i} \end{aligned} \quad (4-7)$$

Where, $n_{k,i}^{\bar{u}}$ and $n_k^d(i)$ are additive noise terms. While $\hat{d}_k(i)$ is the linearly regressed form of distance $d_k(i)$ changing over time. While $\bar{u}_{k,i}^o$ and $d_k^o(i)$ are same as stated above in **section 4.2.1 (A)**.

Since the nodes in the model calculates distance from two separate targets, the position vector of the food target, and the position vector of the Predator. Therefore, to distinguish between the two targets we use subscripts f for food and p for the predator. Hence going forward, we will use w^f to define the location of the food and w^p to define the location of the predator.

C. Motion Control Mechanism

The nodes in a mobile network would update its location vector over time according to rule defined as below,

$$x_{k,i+1} = x_{k,i} + \Delta t \cdot v_{k,i+1} \quad (4-8)$$

Where i is the time instant, Δt is the sampling time, and $v_{k,i+1}$ is the velocity vector of node k . The node updates its velocity vector based on the surrounding environment and thereby accordingly updates its position vector too. The different modes in which the fish can move while updating its velocity vector are as follows,

1. Foraging towards the food target.
2. Coherent motion and collision avoidance.
3. Self-organisation and Reunion when under attack.
4. Escape the predator attack and evade.

Each node estimates the target location of the predator $w_{k,i}^p$ and the target location of food $w_{k,i}^f$ in real time. We will discuss the estimation of predator and food location in the coming sections. While in the next subsection we will discuss how the nodes computes its velocity to make desired moves.

(1) Foraging towards food source.

While foraging for the food, the nodes update its velocities based on the surrounding environment. This can be synonymous to movement of UAVs towards a defined target where it updates its velocity based on the surrounding threat. The nodes position with the relative position of predator is broken in four different regions. The four regions I-IV can

be illustrated as in **Figure 4.3**. Two concentric circles with origin as predator. The inner circle has radius r_1 while outer circle has radius $r_2 = 2r_1$.

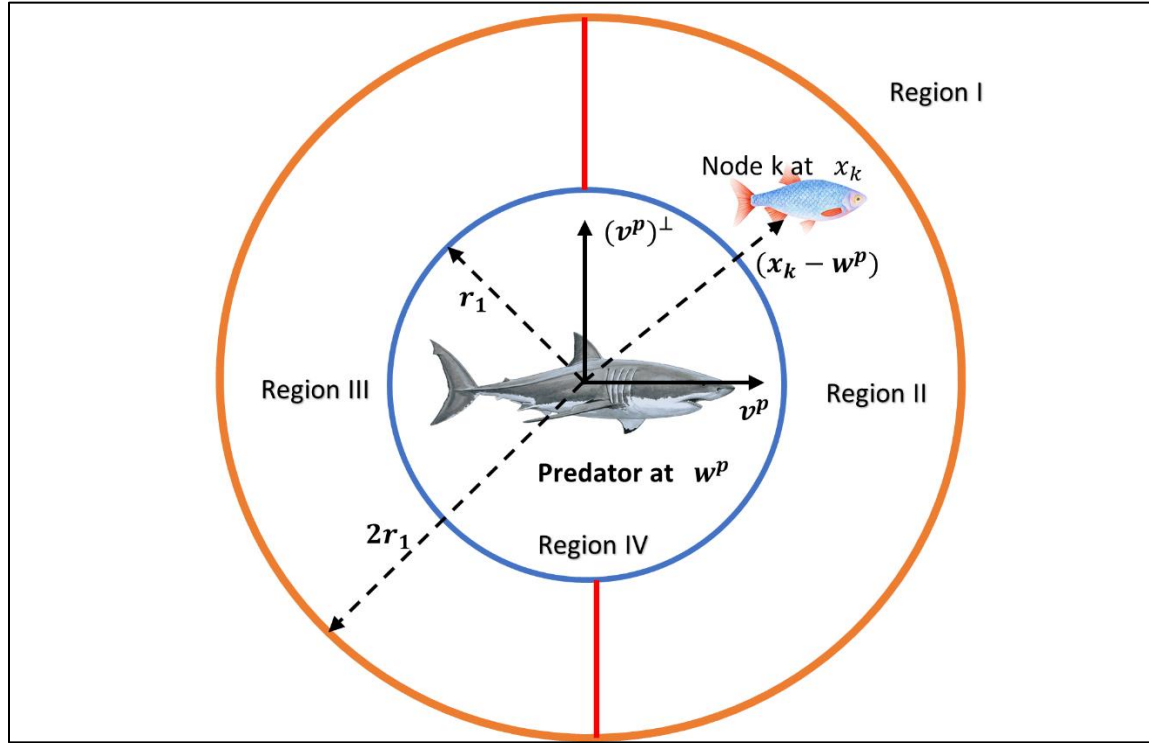


Figure 4.3 : Four Regions around Predator based on relative position of fish node

The nodes update their velocities in each region based on below velocity vector defined.

Region I

In this region, the distance between the node k and the predator is more than radius $2r_1$, and hence the node doesn't sense any threat. The node in region 1 continues to cruise towards the target and remains in foraging mode. The velocity update for region I is defined as below.

$$v_{k,i+1}^a = \frac{w_{k,i}^f - x_{k,i}}{\|w_{k,i}^f - x_{k,i}\|}, (region I) \quad (4-9)$$

Region II

This is the region defined when the node k is at a distance between the radii r_1 and r_2 and on the right-hand side of the predator. The node senses the danger and stops foraging and follows its neighbours to move in coherence. The velocity vector thus becomes zero.

$$v_{k,i+1}^a = 0, (region II) \quad (4-10)$$

Region III

This is the region defined when the node k is at a distance between the radii r_1 and r_2 but on the left-hand side of the predator. The node in this case senses the threat if it continues to move forward. Thus, the node tries to move in opposite direction and maintain a safe distance from the predator. The velocity vector is defined as,

$$v_{k,i+1}^a = -\frac{v_{k,i}^p}{\|v_{k,i}^p\|}, (region III) \quad (4-11)$$

Region IV

This is the region when the UAV node is within the inner circle of the predator with radius r_1 . The node in this case senses danger and tries to escape using evasion strategy discussed in the upcoming subsection “**Escape the predator attack and evade**”

(2) Coherent motion and Collision Avoidance.

The node while moving towards the food source want to confuse the predator and hence move coherently in a group. At the same time, they also wish to maintain harmony with

the neighbouring nodes by avoiding any collision. Thus, the nodes maintain a safe distance r from its neighbours.

Consider two nodes k and l in neighbourhood with position vector x_k and x_l . Therefore, to avoid collision the nodes must satisfy the below equation.

$$r - \epsilon \leq \|x_k - x_l\| \leq r + \epsilon \quad \text{for all } l \in N_k \setminus \{k\} \quad (4-12)$$

Where ϵ is a small positive number, and r is predefined safe distance.

The cohesion and anti-collision is achieved by minimizing the cost function [84] below.

$$J_k(v_k) = \sum_{l \in N_k \setminus \{k\}} [\| (x_k + \Delta t \cdot v_k) - (x_l + \Delta t \cdot v_l) \| - r]^2 \quad (4-13)$$

The minimization problem is solved to obtain,

$$\delta_{k,i} = \frac{1}{|N_k| - 1} \sum_{l \in N_k \setminus \{k\}} \left(1 - \frac{r}{\|x_{l,i} - x_{k,i}\|} \right) (x_{l,i} - x_{k,i}) \quad (4-14)$$

Where $\delta_{k,i}$ is a cohesive term which recommends the nodes to update their velocity direction consistent with the average of displacement vectors $(x_{l,i} - x_{k,i})$, while maintain a safe distance r from its neighbouring nodes.

Finally, the nodes update their velocity vector according to eqn. (4-15) to move coherently without collision.

$$v_{k,i+1}^c = v_{k,i}^g + \gamma \delta_{k,i} \quad (4-15)$$

Where γ is a positive weight and $v_{k,i}^g$ is the average velocity estimate of the centre of gravity of the network and is defined as,

$$v^g \cong \frac{1}{N} \sum_{i=1}^N v_i \quad (4-16)$$

Since the network is distributed thus, we need to obtain the velocity estimate of the centre of gravity of the network in a distributed manner. We will use the diffusion adaptation technique, here as well. We define a global cost function as below.

$$j^g(v^g) = \sum_{k=1}^N \|v_{k,i} - v^g\|^2 \quad (4-17)$$

Applying the solved ATC algorithm structure from eqn. (4.4) we estimate the velocity of centre of gravity.

$$\begin{aligned} \psi_{k,i} &= (1 - \mu_k^v) v_{k,i-1}^g + \mu_k^v v_{k,i} \\ v_{k,i}^g &= \sum_{l \in N_k} a_{l,k}^v \psi_{l,i} \end{aligned} \quad (4-18)$$

Where $\{a_{l,k}^v\}$ is positive real weight as described in eqn. (4.3). While subscript v is used for estimation of v^g .

(3) Self-organization and Reunion when under attack.

When a predator attacks a school of fish, the fishes' splits into small fragments to escape the threat. Once the threat is gone the smaller groups try to reconnect and form a single group to forage. The reunion of the small group occurs when fishes at the outer boundaries of each group moves towards each other. Drawing parallel with the nodes in our network,

each node on the outer boundary of the small, fragmented groups estimates the location of the neighbouring nodes and advance towards them. But for nodes to advance towards other group, it must identify its position and if it is positioned at the edges. Therefore, we define three kinds of edges, the frontal edge, the right edge, and the left edge. A node k computes its neighbours l in each direction using

$$x_{l,i}^k = \mathbb{W}(v_{k,i})^T (x_{l,i} - x_{k,i}) \quad (4-19)$$

Where, $x_{l,i}^k$ is the position vector of l with respect to k and $\mathbb{W}(v)$ is an orthonormal matrix for the local coordinate system of a node moving with velocity vector

$$\mathbb{W}(v) = \begin{bmatrix} v_1/\|v\| & -v_2/\|v\| \\ v_2/\|v\| & v_1/\|v\| \end{bmatrix} \quad (4-20)$$

Once the value of $x_{l,i}^k$ is evaluated, the nodes confirm their edge location based on the values of $x_{l,i}^k$. If the value of first coordinate of $x_{l,i}^k$ is greater than zero, then node l is positioned in front of k . And if the value of second coordinate of $x_{l,i}^k$ is greater than zero, then node l is positioned left to k . while if the value of second coordinate of $x_{l,i}^k$ is less than zero, then node l is positioned right to k . This can be summarised as in the Table 4-2 below.

Table 4-2: Estimating node position on the outer boundary of Fragmented groups

Node l 's position w.r.t k ($x_{l,i}^k$)	Value of $x_{l,i}^k$	Inference
First Coordinate	$x_{l,i}^k > 0$	Node l is positioned in Front of k
second Coordinate	$x_{l,i}^k > 0$	Node l is positioned Left to k
second Coordinate	$x_{l,i}^k < 0$	Node l is positioned Right to k

If the neighbours in the front of node k is less than zero, then we can infer that k is in the frontal edge of the fragmented group. Similarly, it goes for left and right edges. Once the nodes identify their position, they try to find the edge nodes outside its neighbourhood and then moves towards it. This is done using the below equations.

$$\hat{l} = \arg \min_l \|x_{l,i}^k\| \mid l \in N_k^F \setminus N_k \quad (4-21)$$

$$v_{k,i+1}^b = \begin{cases} 0, & \text{if } \hat{l} = \text{empty set} \\ \frac{(x_{l,i} - x_{k,i})}{\|x_{l,i} - x_{k,i}\|}, & \text{Otherwise} \end{cases} \quad (4-22)$$

Where N_k^F is the set of nodes that lie in the front of node k . Similarly, it can be found for left and right edge nodes.

And the velocity of the predator with respect to each node is estimated as below.

$$v_{k,i}^p = \frac{1}{\Delta t} (w_{k,i}^p - w_{k,i-1}^p) \quad (4-23)$$

(4) Escape the predator attack and evade.

We will discuss the evasion in upcoming section 4.3 in detail.

4.3 Evasion Phase

When UAVs fly on a mission they might be attacked by enemies. The UAVs in this case should be able to swiftly manoeuvre its path and evade. In our case when a UAV node lies in region IV it swiftly changes its trajectory from foraging mode to evasion mode. Thus, we propose the evasion strategy, where each UAV will manoeuvre through a defined trajectory and angle to evade the attack.

The defined trajectory is a straight-line path at an angle θ_d , which is adjacent or tangential to the direction of UAVs.

Consider a UAV moving senses a danger at time instant t_0 . And let the position of the UAV at this instant be given as $(x(t_0), y(t_0))$. The distance between the UAV and the enemy at this instant is given by d_k^p , and $d_k^p < r_1$. The UAV in this case switches to evasion mode using the set of defined trajectories. The position of the UAV at t_0 is stored and is reference point for the new trajectory during evasion. The straight-line path during evasion is given by [12],

$$\begin{aligned}x_d(t) &= x(t_0) + \hat{\lambda}(n) \cos(\theta_d(t)) \\y_d(t) &= y(t_0) + \hat{\lambda}(n) \sin(\theta_d(t)) \\ \hat{\lambda}(n) &= \Delta t + (n - 1) \Delta t\end{aligned}\tag{4-24}$$

Where $\hat{\lambda}(n)$ is called as path parameter. It is set as an arithmetic progression that increase to set the defined trajectory. And n is the iteration number during the execution.

The angle of evasion θ_d is obtained using four quadrants inverse tangent atan2. The inverse tangent atan2 gives the angle between $-\pi$ and π . The desired angles of evasion θ_d are based on the trigonometric quadrants and is updated as stated in **Table 4-3**. The UAV approximates the approaching angle of enemy as θ_p . While the actual angle of approach of the enemy is given by θ_{np} . In deciding the angle of evasion, we compare the angles θ_p and $\theta_{np} - 180$ and $\theta_{np} + 180$. For example, in quadrant I if the danger approaches and $d_k^p < r_1$ and $\theta_p \geq \theta_{np} - 180$, the desired angle of evasion is given as,

$$\theta_d = \theta_p + \theta_c$$

Where θ_c is the tangential angle (say 90°) desired for evasion. Similarly, for Quadrant II, III, IV can be obtained as in **Table 4-3**.

Table 4-3: Evasion Angle for UAVs in each Quadrant

Quadrants	Condition for Evasion	Desired Angle of Evasion (θ_d)
I	$d_k^p < r_1$ and $\theta_p \geq \theta_{np} - 180$	$\theta_d = \theta_p + \theta_c$
II	$d_k^p < r_1$ and $\theta_p < \theta_{np} - 180$	$\theta_d = \theta_p - \theta_c$
III	$d_k^p < r_1$ and $\theta_p \leq \theta_{np} + 180$	$\theta_d = \theta_p + \theta_c$
IV	$d_k^p < r_1$ and $\theta_p > \theta_{np} + 180$	$\theta_d = \theta_p - \theta_c$

4.4 Distributed Velocity Estimate for Swarm

Combining all the velocity updates for nodes under each scenario, we propose a global velocity update for the nodes during foraging mode as below.

$$v_{k,i+1} = \lambda \cdot I_{k,i} (\alpha \cdot v_{k,i+1}^a + \beta v_{k,i+1}^b) + (1 - \lambda \cdot I_{k,i}) v_{k,i}^g + \gamma \delta_{k,i} \quad (4-25)$$

Where $\{ \lambda, \alpha, \beta, \gamma \}$ are positive scalar weights. While $I_{k,i}$ is an indicator function and $I_{k,i} \in \{0,1\}$. If both $v_{k,i+1}^a$ and $v_{k,i+1}^b$ are equal to zero, i.e., node k is in region II and does not need to reunite. Otherwise $I_{k,i}$ is equal to one.

We bound the upper velocity limit of the nodes by v_{max} , to meet the nonholonomic constrains of the UAVs.

4.5 Global Estimation of Predator Location and velocity.

The fish nodes need to estimate the location of the predator and the food. The estimation for predator location denoted by w^p and food location denoted by w^f . The algorithm to estimate w^p and w^f is same and hence we just mathematically present the estimation for w^p . The same can be replicated to obtain the estimation for food location w^f .

At any time, instant (i) a node k has the measurement of the location vector $\{\hat{d}_k^p(i), u_{k,i}^p\}$ as in Eqn. (4-7). The nodes estimate the predator location w^p by minimizing the cost function.

$$j^p(w^p) = \sum_{k=1}^N E |\hat{d}_k^p(i) - u_{k,i}^p w^p|^2 \quad (4-26)$$

Applying the ATC algorithm from Eqn. (4-4), we can get estimate $w_{k,i}^p$

$$\psi_{k,i}^p = w_{k,i-1}^p + \mu_k^p u_{k,i}^{p\ T} [\hat{d}_k^p(i) - u_{k,i}^p w_{k,i-1}^p] \quad (4-27)$$

$$w_{k,i}^p = \sum_{l \in N_k} a_{l,k}^p \psi_{k,i}^p$$

Where $\{a_{l,k}^p\}$ satisfies

$$\sum_{l=1}^N a_{l,k}^p = 1, \quad a_{l,k}^p = 0 \quad \forall l \notin N_k \quad (4-28)$$

For the sake of simplicity, we can make an approximation in Eqn. (4-27) as below,

$$(w_{k,i-1}^p - x_{k,i}) \approx \| w_{k,i-1}^p - x_{k,i} \| u_{k,i}^{p\ T} \quad (4-29)$$

Therefore, we can rewrite Eqn. (4-27) as,

$$\psi_{k,i}^p = (1 - \mu_k^p)w_{k,i-1}^p + \mu_k^p q_{k,i}^p \quad (4-30)$$

$$w_{k,i}^p = \sum_{l \in N_k} a_{l,k}^p \psi_{l,i}^p$$

Where $q_{k,i}^p$ is the measured location of the target by the predator node, and can be expressed as,

$$q_{k,i}^p = x_{k,i}^p + d_k^p(i)u_{k,i}^{p^T} \quad (4-31)$$

The update in Eqn. (4-30) is a convex combination of the current measurement $\{q_{k,i}^p\}$ and the last estimate of the target location $w_{k,i-1}^p$.

The same estimation procedure can be followed for w^f as well.

4.6 Predator Behaviour

The predator used for attack is also a nonholonomic UAV model. The predator tracks the nearest swarm node and attack it. The node location (w_i^n) is estimated using Eqn. (4-4)

The velocity and position of the predator at each instant is thus updated by estimating the relative position of the nodes.

$$\begin{cases} v_{i+1}^p &= c \cdot v_{max} \frac{w_i^n - w_i^p}{\|w_i^n - w_i^p\|} \\ w_{i+1}^p &= w_i^p + \Delta t \cdot v_{i+1}^p \end{cases} \quad (4-32)$$

Where c is a positive scalar governing the predator's speed.

CHAPTER 5

COOPERATIVE PREY HUNTING BY MULTIPLE PREDATORS

5.1 Introduction

In this chapter we will discuss the cooperative strategy used by predators for hunting. Just like the swarm of fishes move in groups in search of food, the predator nodes hunt in groups as well. The cooperative behaviour for hunting is exhibited by many land and underwater animals, where they collaborate and form group to strategies their attack and hunt. For example, animals like tigers would attack herd of sheep and dears from different directions. Similarly, among the fish family dolphins encircle their prey and take turn to attack and feed [85]. Drawing parallel to air combat field, the same strategy of cooperative attack can be used by enemy UAVs to attack their enemies on a mission.

We use the same diffusion adaptation technique for the predators as we used for the swarm. The kinematic model of the predator remains the same. Each predator in the group is aware of the position of its neighbour predators. The predator group also moves cooperatively maintaining a safe distance from each other. All the assumptions made for the swarm group in **Table 4-1** remains same for the Predator group as well. The motion control mechanism and cooperative hunting strategy will be discussed in the subsequent section in detail.

5.2 Motion Control Mechanism for Predators

Consider a group of M predators collectively hunting a swarm of fish. From our perspective we say a group of M UAVs are on a mission to hijack and attack enemy UAVs foraging on a mission.

Let the location of any predator at time instant i be defined as $x_{l,i}^p$ where $l \in 1 \dots M$. The predators coordinate their motion influence by two factors from the swarm network. First is the location of the centre of gravity of the network of swarm. Second is the location of the node of interest from the network of swarm.

Let $x_{l,i}^g$ be the estimated location of the centre of gravity of the network by a predator l at time i . The predators cooperatively estimate the location of the centre of gravity $x_{l,i}^g$ using the ATC algorithm discussed in Eqn. (4.4). while the location of the node of interest $w_{l,i}$ is estimated individually. This is because the predators are spread across the network in different direction and hence their node of interest are different. Thus, each predators estimate the position of node of interest $w_{l,i}$ using below equation.

$$w_{l,i} = w_{l,i-1} + \nu u_{l,i}^T [d_l(i) - u_{l,i}(w_{l,i-1} - x_{l,i}^p)] \quad (5-1)$$

Where, ν is a step size. While $d_l(i)$ is the measured distance as defined in Eqn. (4-5) and $u_{l,i}$ is the measured direction of the node as defined in Eqn. (4-6) by any predator l at time i .

The predators collectively hunt to increase their efficiency. The strategy of hunting can be divided into multiple modes or states based on the action of the predators [22].

Therefore, we will model the hunting behaviour of predator as state machine model in the next section.

5.2.1 State Machine Model

The hunting behaviour of the predator is divided into four possible states based on the action involved in the hunting process. We denote these states as S_0, S_1, S_2 and S_3 . The state S_0 is termed as chase. State S_1 is termed as Encircle. State S_2 is termed as Trap. While the final state S_3 is termed Attack. The state transition diagram for the predator is shown in **Figure 5.1** below.

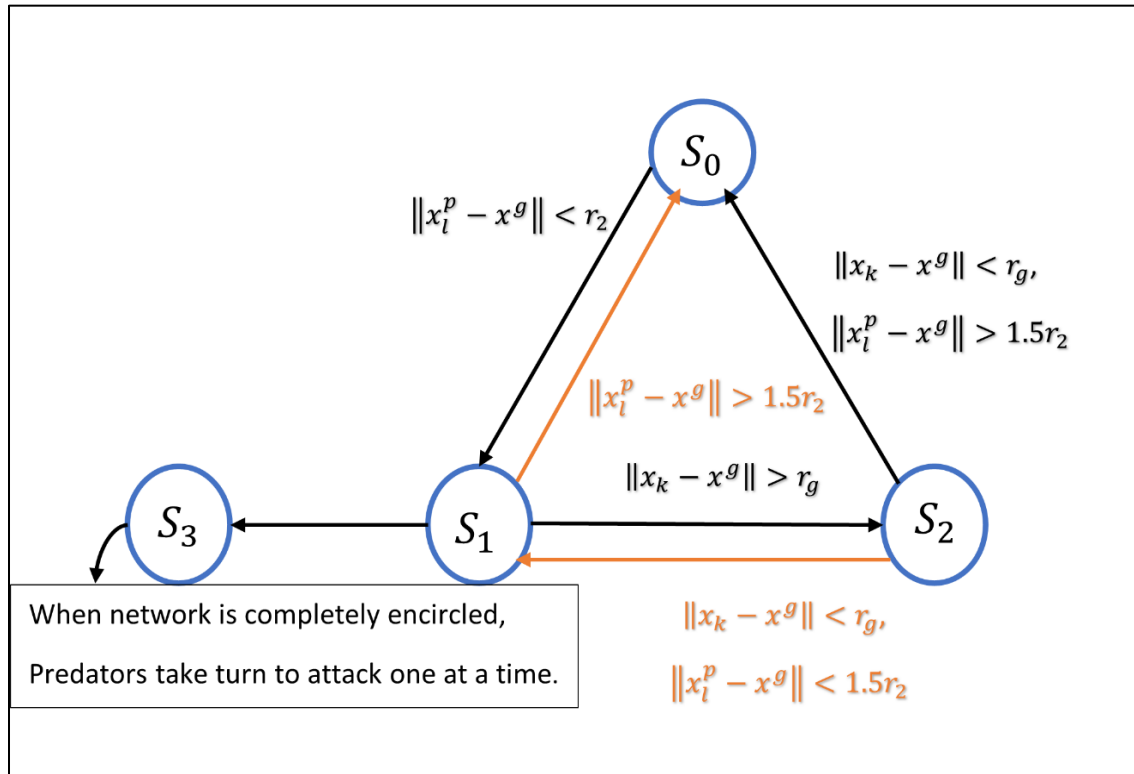


Figure 5.1: State Transition Model for Predator

The cooperative hunting of the predators' starts with chase mode. Predators would move towards the enemies or swarm network in our case. The predator continues to move in chase mode until they reach near to the network. We set this nearness as, the distance

between the network's centre of gravity and the predator location being less than r_2 . In other words, the predator continues to chase until $\|x_{l,i}^p - x_{l,i}^g\| < r_2$. Once the predator reaches the circumference of the defined circle, it switches to next state S_1 . In the encircle mode the predator group will encircle the network while keeping a uniform distance between them.

The predator continues to make an orbital motion around the network centre while monitoring the position of the nodes in the network. The predator looks for the farthest node from the network centre and checks if it is within distance $1.5r_2$. If a node is outside the network centre i.e., $\|w_{l,i} - x_{l,i}^g\| > r_g$, it is termed as an outlier. As an outlier is detected the predator switches to Trap state S_2 and pushes back the outlier into the network centre. Once the predator successfully pushed back the outlier, it may switch to state S_0 or S_1 based on its distance from the network centre. If $\|x_k - x^g\| < r_g$ and $\|x_l^p - x^g\| < 1.5r_2$, the predator switches to S_1 , whereas, if the distance $\|x_k - x^g\| < r_g$ and $\|x_l^p - x^g\| > 1.5r_2$, the predator switches to S_0 as shown in **Figure 5.1**. Once the predator is back into state S_0 it again chases the network and switches to S_1 . And once the predators in state S_1 encircle the swarm network and there is no outlier the predator take turn to attack the Swarm. As one predator takes turn to attack the network the rest of the predator adjust their position to encircle and ensure the swarms in the network do not escape.

In next subsection we will discuss how the predator UAVs update their velocities in each state [17].

State S_0 : Chase

When the distance between the predator and the network is large the predator moves in chase mode. The objective of the predator group is to get closer to the network centre before it can switch to next state. Thus, the predator updates its velocity vector towards the network as in Eqn. (5-2).

$$v_{l,i+1}^d = u(x_{l,i}^g - x_{l,i}^p) \quad (5-2)$$

Where $x_{l,i}^g$ and $x_{l,i}^p$ are same as defined above.

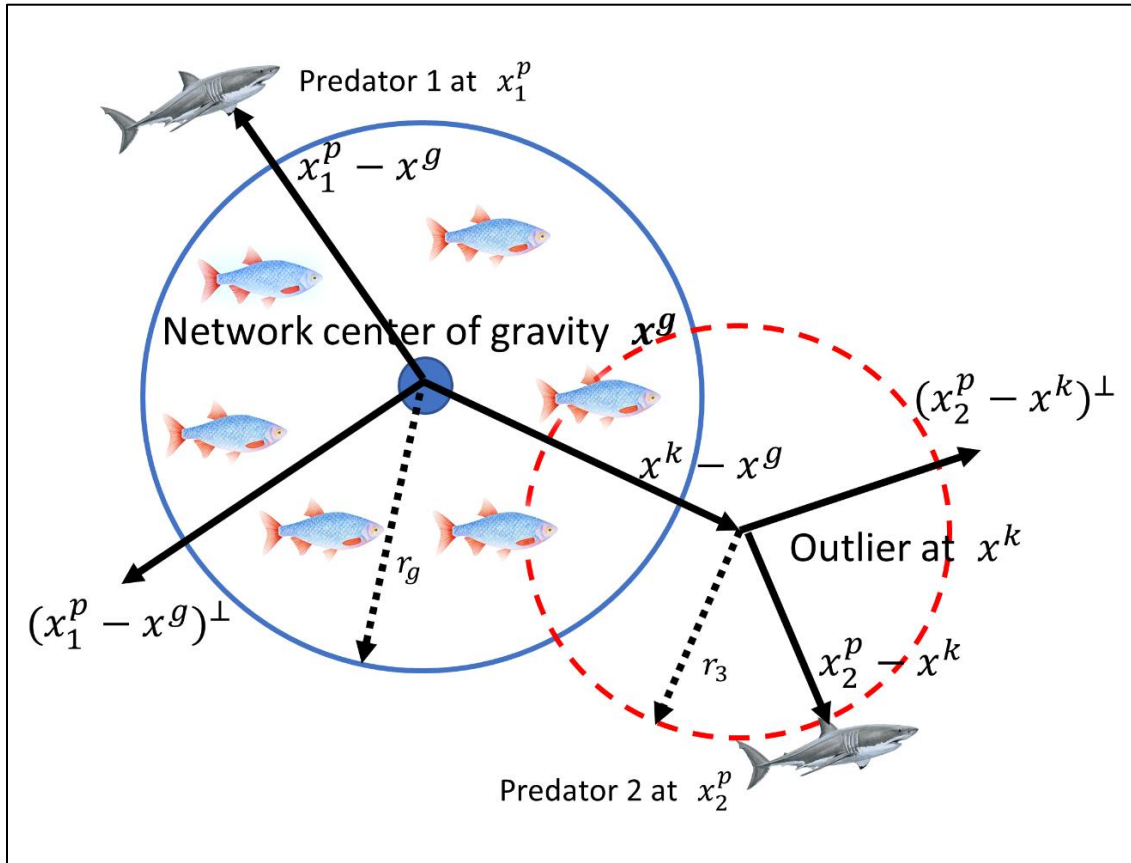


Figure 5.2: Predator position with the network center and concept of Outlier

State S_1 : Encircle

The encircle state is enabled once the predator reaches closer to the swarm network. The predator then starts to encircle the network in a radius r_2 with origin at centre of gravity of network. The predators at this point first align their position and evenly distribute themselves on the circumference of the encircling orbital. We used the simple mathematical formula of chord to place the predators equidistantly on the circumference. So, if we have 6 predators for example, the chord length is $2r \sin(\theta/2)$, and $\theta = 2\pi/6$, thus the chord length is equal to $2r \sin(\pi/6)$. This even distribution of the predator makes it difficult for the swarms to escape. The predator also checks the distance between its closest neighbours using the chord length to avoid collision and maintain uniformity in position. If the distance between its closest neighbours is less than the measured chord length the predator moves in a way to adjust its position and realign itself. The predator also decides its direction of rotation based on the neighbour predator position. Say predator j and l are neighbours within the chord distance, then the inner product of l and j with respect to network centre x^g is obtained as $(x_{j,i}^p - x_{l,i}^g)^T (x_{l,i}^p - x_{l,i}^g)^\perp$. If the inner product is greater than zero then predator j is said to be in the right semicircle of predator l . This can be illustrated as in **Figure 5.2**. The predator l then sets a velocity vector towards empty semicircle. In case there is no neighbour within the set distance then the predator l , randomly chooses a direction of rotation. The velocity vector is set as in Eqn. (5-3).

$$v_{l,i+1}^d = c_2 \cdot u[(x_{l,i}^p - x_{l,i}^g)^\perp] \quad (5-3)$$

Where $c_2 = \pm 1$, (+1 for counter clockwise and -1 for clockwise). If right semicircle is empty, $c_2 = +1$ and if left semi-circle is empty $c_2 = -1$, else c_2 can be randomly +1 or -1.

State S_2 : Trap

In this mode the predator checks for an outlier. A node outside the network centre of gravity is considered as an outlier as shown in **Figure 5.2**. The Predator suspects the node trying to escape and hence pushes it back into the network centre. The predator does not attack an outlier, rather it only blocks its position. The attack would likely make the node escape using evasion technique, and hence the predator moves around the outlier at a finite distance and ensures the node is pushed back to the network centre. The predator updates its velocity vector as in Eqn. (5-4).

$$v_{l,i+1}^d = \begin{cases} u(w_{l,i} - x_{l,i}^p) & \text{if } \|w_{l,i} - x_{l,i}^p\| > 1.5r_3 \\ -u(w_{l,i} - x_{l,i}^p) & \text{if } \|w_{l,i} - x_{l,i}^p\| < r_3 \\ c_3 \cdot u[(w_{l,i} - x_{l,i}^p)^\perp] & \text{otherwise} \end{cases} \quad (5-4)$$

Where $c_3 = \pm 1$ depending upon the inner product of $(w_{l,i} - x_{l,i}^g)^T (x_{l,i}^p - w_{l,i})^\perp$. If the inner product is greater than zero c_3 is 1, else -1.

State S_3 : Attack

In this mode the predator after completely encircling the swarm network takes turn to attack the swarm. The predator tracks the location and velocity vector of the nearest swarm and moves towards it to attack. We update the velocity vector as in Eqn. (5-5)

$$v_{l,i+1}^d = u[w_{l,i} + \Delta t \cdot v_{l,i} - x_{k,i}^p] \quad (5-5)$$

where $v_{l,i}$ is the velocity estimate of the swarm node by any predator l at time instant i . $v_{l,i}$ is estimated as in Eqn. (4-23).

5.3 Global Velocity Update for Predator

Combining all the behaviour of the predator we propose a global velocity update and update for the location vector of the predator. The predator also wishes to maintain a safe distance from its fellow predator and avoid any collision. Thus, each predator while updating its velocity also takes care of collision avoidance.

$$\begin{aligned} v_{l,i+1}^p &= \lambda^p v_{l,i+1}^d + \gamma^p \delta_{l,i}^p \\ x_{l,i+1}^p &= x_{l,i}^p + \Delta t \cdot v_{l,i+1}^p \end{aligned} \quad (5-6)$$

where $\{\lambda^p, \gamma^p\}$ are non-negative weights and $\delta_{l,i}^p$ can be expressed as,

$$\delta_{l,i}^p = \frac{1}{|M_l| - 1} \sum_{l \in M_l \setminus \{k\}} (r_p - \|x_{l,i}^p - x_{j,i}^p\|) \cdot u \cdot (x_{l,i}^p - x_{j,i}^p) \quad (5-7)$$

Where M_l , is the set of predator nodes, within the radius r_p . While the overall speed of the predator $v_{l,i+1}^p$ is bounded by v_{max}^p .

CHAPTER 6

DESIGN OF TRACKING CONTROLLER

In this chapter we will integrate the designed controller with the navigation algorithms designed in previous chapters. The aim of the controller design is ensuring that the UAVs completely follow the trajectory planned by the navigation algorithm with error. Therefore, we design an optimal Fractional order PID (OFOPID) controller to ensure zero error in the navigation. The designed controller is first integrated with a single UAV and a single UAV tracking analysis is done. A Schematic diagram for a closed loop single UAV tracking controller is shown in **Figure 6.1**. Further the same control technique is implemented for the fleet of UAVs.

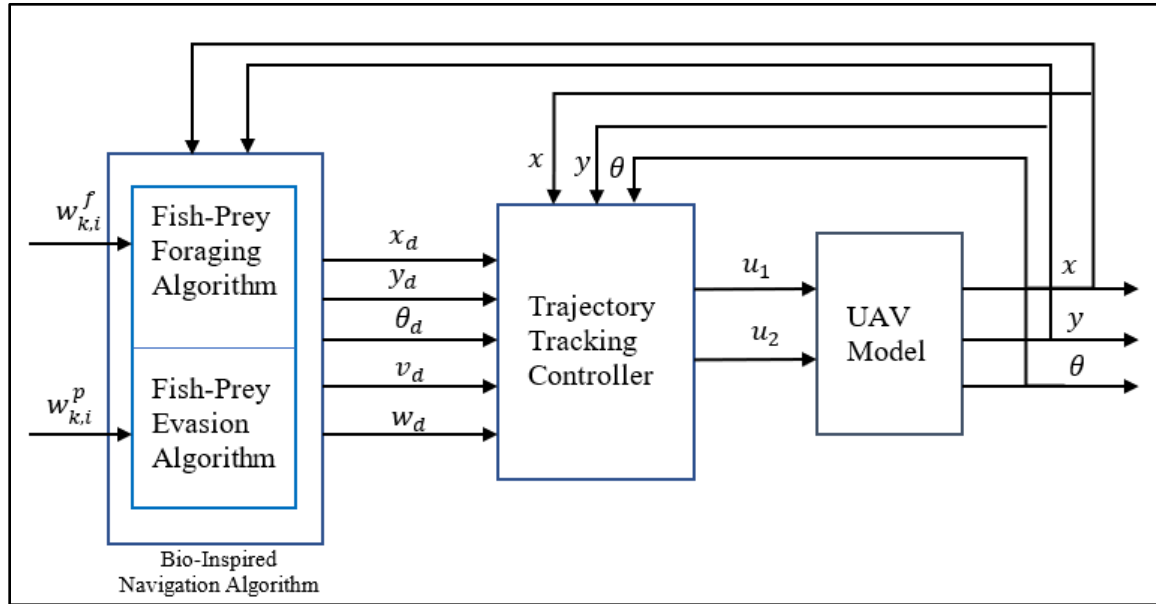


Figure 6.1: Control Schematic - Single UAV

In the upcoming section we will shed some lights on trajectory generation process and switching conditions. Here once again we reiterate that the UAV in consideration is planar and ignores the aerodynamic consideration of a 3D physical model.

6.1 Trajectory generation

In this section, we summarise the trajectory generation algorithm for the UAVs based on work done in chapter 4 and 5. The UAVs navigate in two modes depending upon the presence of danger. A simple schematic diagram as shown in **Figure 6.2** in form of flowchart represents the control decision process of the UAVs during navigation.

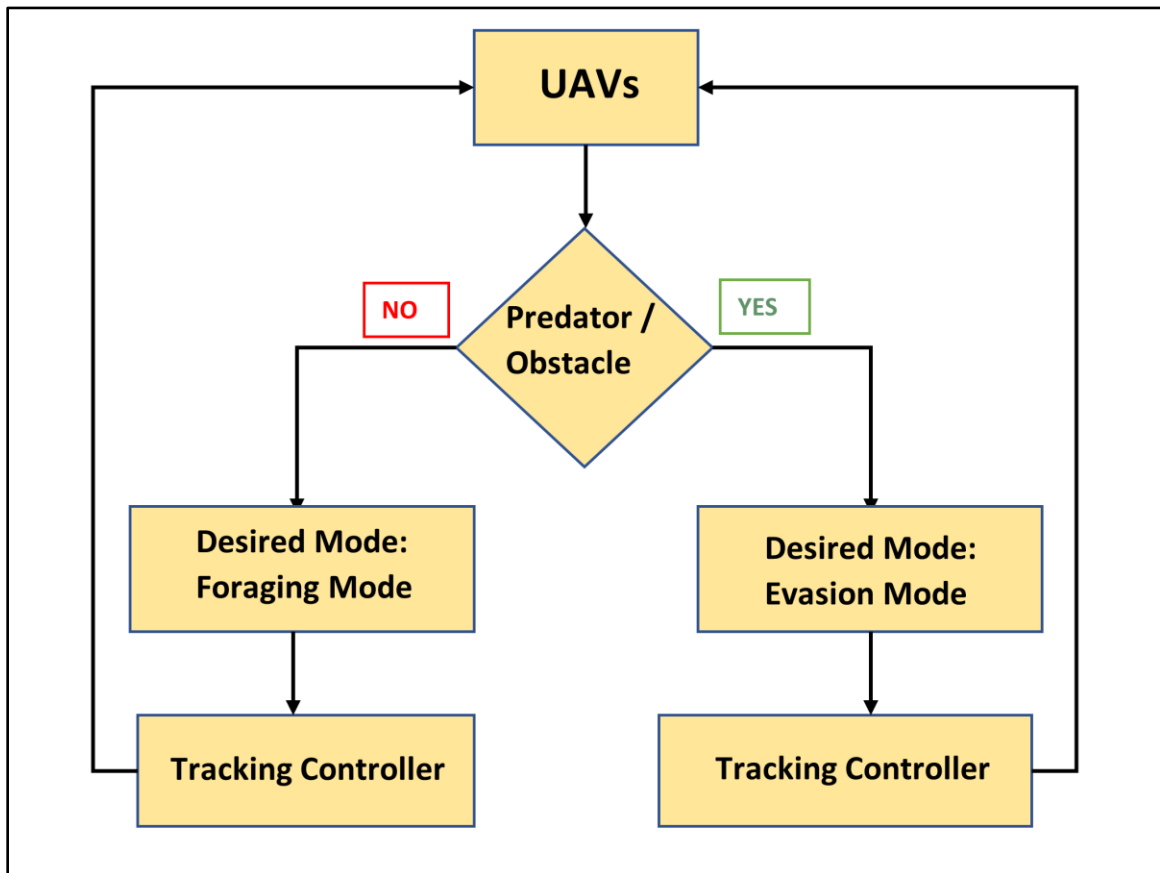


Figure 6.2: Control decision flowchart for Fleet Navigation

The navigation mode of the UAVs depends upon the presences and absence of predator and obstacle. In the absence of predator/obstacle the UAVs move in foraging mode else they switch to evasion mode until they are far away (defined distance) from threat. In each case the trajectory tracking is controlled using the designed FOPID controller.

The desired state for the UAV's during navigation is defined as follows.

- x_d : desired x coordinate position
- y_d : desired y coordinate position
- θ_d : desired steering angle of UAV
- V_d : desired UAV linear velocity. This is obtained from distributed velocity estimate for swarms in the fish prey algorithm, **Chapter 4 section 4.4**.
- w_d : desired UAV angular velocity, in presence or absence of predator.

In the next subsection we discuss the desired velocity of the UAVs in detail, based on the presence or absence of threat.

6.1.1 Trajectory Generation in Absence of Threat

When there is no threat, the UAVs navigate in the foraging mode. They mimic the behaviour defined in the bio-inspired fish prey algorithm. while they move at a velocity inspired by the fish prey algorithm. The velocity is used to obtain the next desired position of the designed trajectory. It is also worth noting that the trajectory satisfies the nonholonomic constraint.

The trajectory generated in absence of threat is given as below.

$$\begin{aligned}
x_d(t+1) &= x_d(t) + \Delta t \cdot V_d \cos \theta_d(t) \\
y_d(t+1) &= y_d(t) + \Delta t \cdot V_d \sin \theta_d(t) \\
\theta_d(t+1) &= \text{atan2d}(y_d(t+1), x_d(t+1)) \\
w_d(t+1) &= \frac{\theta_d(t+1) - \theta_d(t)}{\Delta t}
\end{aligned} \tag{6-1}$$

6.1.2 Trajectory Generation in Presence of Threat

In case of a predator attack the UAVs evade using straight line path as defined in chapter 4, under section 4.3 Evasion Phase.

6.2 Single UAV Tracking Analysis

We first design a tracking controller for a single UAV, before implementing it for the fleet of UAVs. The controller will ensure trajectory tracking for the UAV in both the foraging and evasion mode.

The kinematic model of a planar UAV (unicycle model) is given by Eqn. (6-2)

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta}_s \end{bmatrix} = \begin{bmatrix} \cos \theta_s & 0 \\ \sin \theta_s & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{6-2}$$

Let the desired trajectory that we wish the UAV must follow is given by Eqn. (6-3)

$$\begin{bmatrix} \dot{x}_d \\ \dot{y}_d \\ \dot{\theta}_d \end{bmatrix} = \begin{bmatrix} \cos \theta_d & 0 \\ \sin \theta_d & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V_d \\ w_d \end{bmatrix} \tag{6-3}$$

The tracking error [86] is given by Eqn. (6-4)

$$\begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} \cos \theta_s & \sin \theta_s & 0 \\ -\sin \theta_s & \cos \theta_s & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d - x \\ y_d - y \\ \theta_d - \theta_s \end{bmatrix} \quad (6-4)$$

The error state goes to zero when the desired and current are equal i.e., if the designed controller is able to force the error states e_x , e_y and e_θ to zero, the UAV will be able to track the desired trajectory.

We therefore proceed to design an Optimal Fractional Order PID controller for the UAV tracking. The parameters of the FOPID will be tuned using PSO technique. The robustness of the designed controller will be illustrated by introducing a wind turbulence region in the environment that would affect the dynamics of the UAV. However, it will be observed from the simulation results that the designed OFOPID is effective in controlling the UAVs to follow the desired trajectory. The details of this shall be discussed in upcoming sections.

6.2.1 Design of Optimal Fractional Order PID Controller.

In recent years the fractional order controller is gaining momentum over the conventional techniques. A detailed review on the industrial acceptance of fractional order system can be found in [87]. One of the main reasons is the fractional order control system decreases the steady state error and increases the speed of the response, and relative stability. This is because the additional “tuning knobs” in the fractional order allows swift and coherent adjustments of the dynamics of the control systems.

The integral-differential equation that defines the control action of a fractional order system is given by Eqn. (6-5) below.

$$u(t) = k_p e(t) + k_i D^{-\lambda} e(t) + k_d D^{\mu} e(t) \quad (6-5)$$

Where $e(t)$ is the error between the desired and current state. And $u(t)$ is the control input. While $\{ k_p, k_i, k_d, \lambda \text{ and } \mu \}$ are five control parameters. A generalised FOPID controller is shown in **Figure 6.3**. To design an optimal controller, the parameters of the FOPID needs to be tuned using some optimization technique. A number of optimization algorithms exists such as Genetic algorithm, Artificial Bee Colony, Ant colony Optimization, etc. However, we will be using Particle Swarm optimization (PSO) technique to optimize the parameters because of the superior performance of the PSO compared to other optimization techniques. PSO is regarded as of the best [88], [89] technique due to simple algorithm, low computational cost, stability in finding optimal solution and better performance.

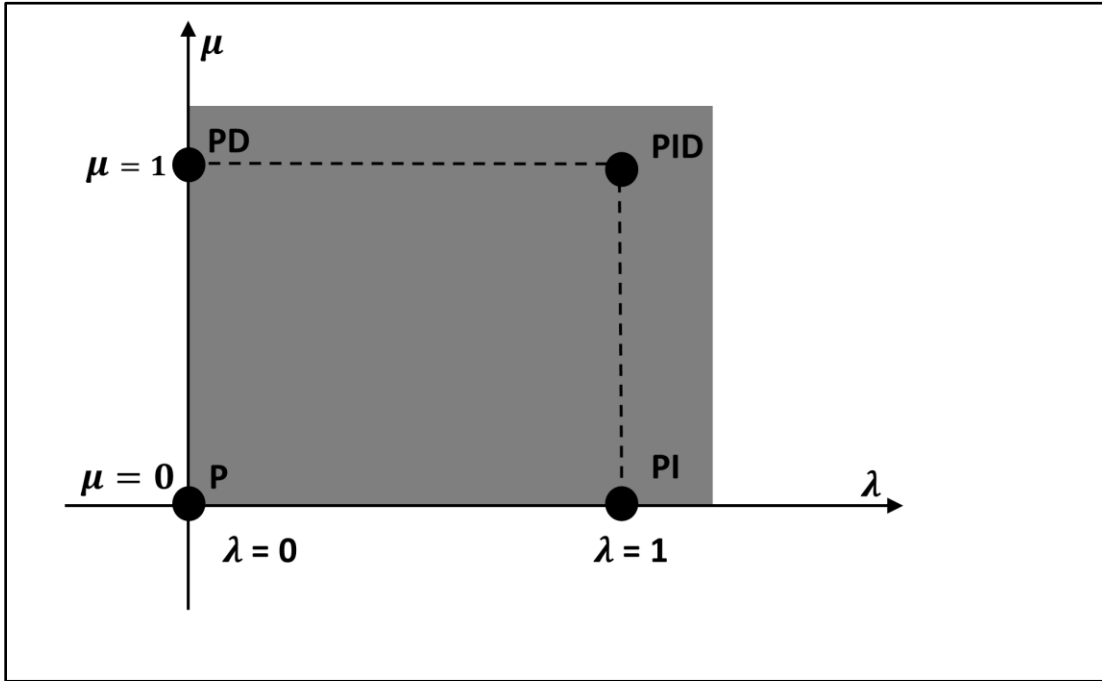


Figure 6.3: Generalized Schematic diagram of FOPID controller

Particle Swarm Optimization

Particle swarm optimization is a stochastic optimization technique based on the principle of natural selection and search algorithm. It is also inspired by the natural behaviour of fish swarm and birds flocking moving towards a target. The optimization technique involves a set of particles (swarm agents) that move around in the search space, looking for the best solution for a defined problem. Each swarm in the space represents a solution to the problem, and it is defined using its position and velocity. Each particle in the swarm tries to keep a track of its best position in the solution space, associated with the best solution that has been achieved so far and is known as P_{best} or personal best. Similarly, the PSO tracks the global best position g_{best} , this is the best possible value obtained so far by any particle in the swarm. In a D dimensional space, the position and velocity of the i^{th} particle of the swarm can be represented by a vector, $S_i = (X_{i1}, X_{i2} \dots X_{iD})$ and $V_i = (V_{i1}, V_{i2} \dots V_{iD})$. whereas the personal best position reached by the i^{th} particle can be represented by a vector as $P_{best_i} = (P_{i1}, P_{i2} \dots P_{iD})$ and global best as the index of particle that visited the best position in the swarm.

The general steps involved in the working of PSO algorithm is further explained using the below steps [90], [91].

Step 1. The parameters of the algorithm such as swarm size, maximum iterations, inertia weight (min and max), cognitive constant (C_1), social constant (C_2), initial position of the particles, and initial velocity of the particles are initialized. The inertia weight function(W) is calculated as below,

$$W = w_{\max} - \frac{w_{\max} - w_{\min}}{\text{iter}_{\max}}$$

Step 2. The values of each gain parameter $\{ k_p, k_i, k_d, \lambda \text{ and } \mu \}$ are initialized randomly within the optimal range of values.

Step 3. The fitness of each particle is evaluated using the integral of the square of the error (ISE) given by Eqn. (6-6)

$$\text{ISE} = \int_0^{\infty} [e(t)]^2 dt \quad (6-6)$$

Step 4. P_{best} and g_{best} of the particles are obtained based on the fitness value calculated in Step 3.

Step 5. The velocity and position of each particle is updated using the below set of Equations given by Eqn. (6-7) and Eqn. (6-8) respectively.

$$V_i^{k+1} = W V_i^k + C_1 R_1 (P_{\text{best}_i} + S_i^k) + C_2 R_2 (g_{\text{best}_i} + S_i^k) \quad (6-7)$$

$$S_i^{k+1} = S_i^k + V_i^{k+1} \quad (6-8)$$

Where W is the inertia weight, and it is a positive constant which plays an important role in balancing the global search i.e., it influences the exploration. while R_1 and R_2 are random numbers between 0 and 1. C_1 is an individual learning parameter. It weighs the importance of the particle's own previous experiences. And C_2 is a social learning parameter. It weighs the importance of the global learning of the swarm.

Step 6. The procedures from 2 to 5 are repeated until the maximum iterations reached or the best solution is obtained.

The simple flowchart representation of the PSO is given in **Figure 6.4**.

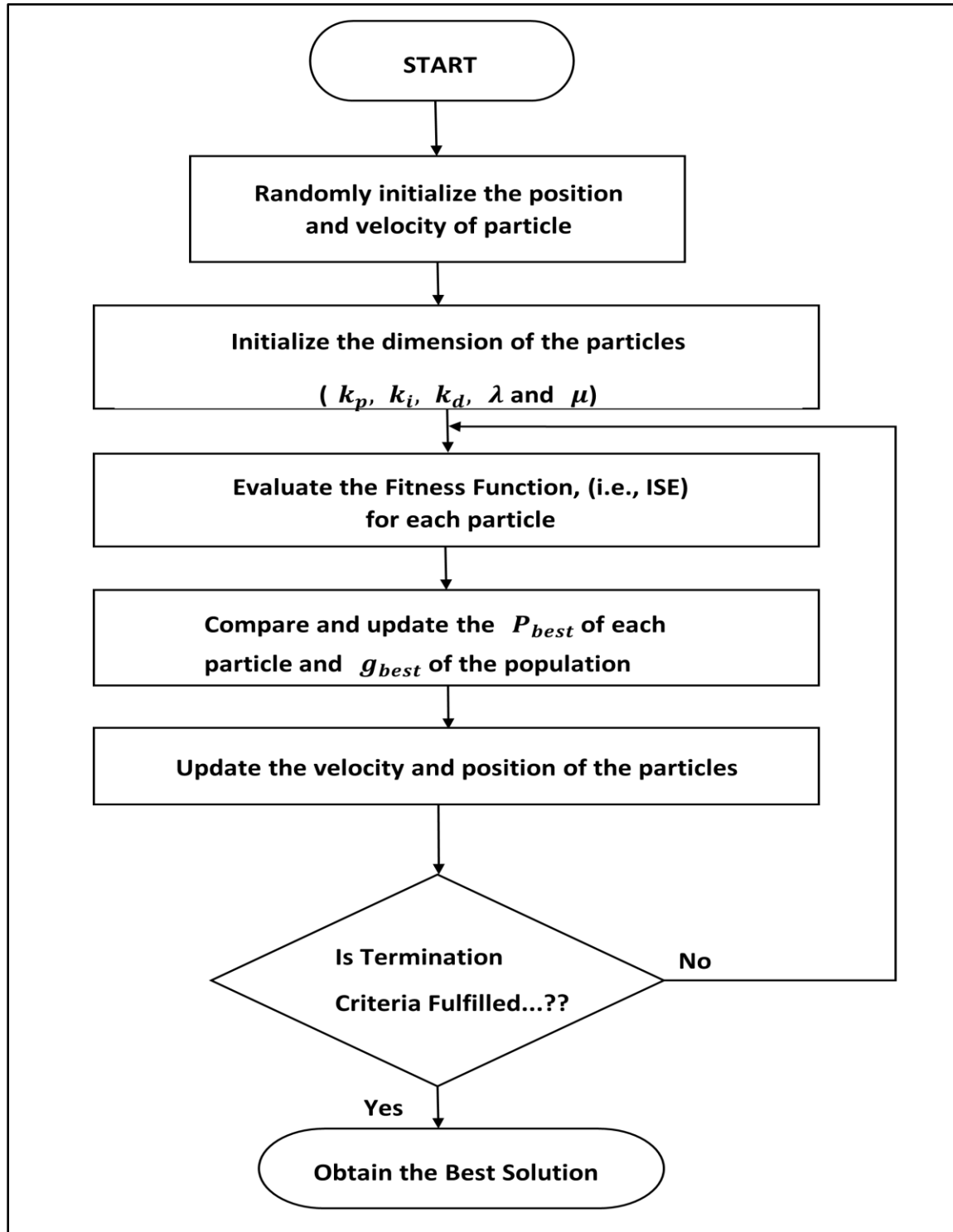


Figure 6.4: Simple Flow chart structure of particle swarm optimization

6.2.2 Control System Implementation

The block diagram for control system design of single UAV agent (unicycle model) tracking a desired trajectory is given in **Figure 6.5**. Two FOPID controllers are designed to drive the position error of the unicycle model to zero. In other words, the two controllers track the desired trajectory of x and y position coordinate. Since the orientation of the model is given by $\theta = \text{atan2}(\frac{y}{x})$ and desired heading $\theta_d = \text{atan2}(\frac{y_d}{x_d})$, therefore when the position error $e_y = (y_d - y)$ and $e_x = (x_d - x)$ goes to zero respectively, the orientation error $e_\theta = (\theta_d - \theta)$ also goes to zero. Thus, the controller is able to track the desired trajectory.

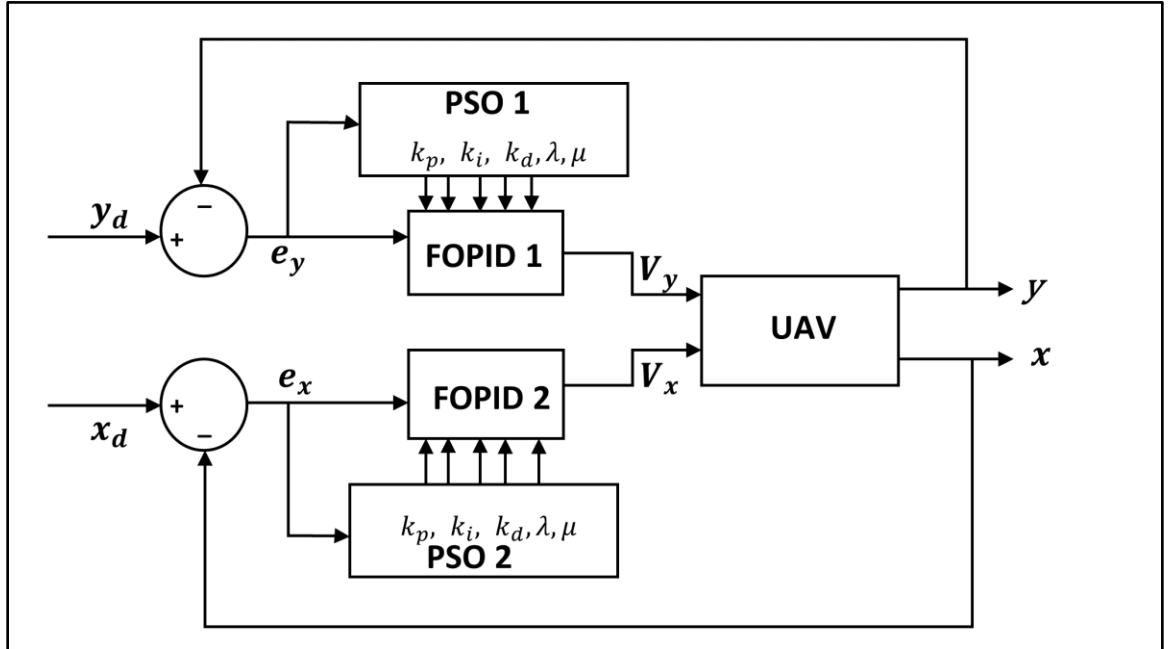


Figure 6.5: The block diagram of the FOPID control system structure

6.3 Simulation Results

The simulation results are carried out to illustrate the effectiveness of the FOPID controller.

The optimized gain parameters obtained for the two FOPID controllers are shown in below

Table 6-1. The other PSO parameters used in the simulation are as follows: $D = 5$, $w_{\max} = 1$, $w_{\min} = 0.25$, swarm size = 30, Itermax = 20, and finally both C_1 & $C_2 = 0.5$.

Table 6-1: Gain Parameters FOPID controller

Parameters	k_p	k_i	k_d	λ	μ
FOPID 1	0.52482476	0.01456812	0.01315661	2.97737203	2.09678112
FOPID 2	0.52482476	0.01456812	0.01315661	2.97737203	2.09678112

The simulation result for single UAV trajectory tracking is shown in **Figure 6.6**. The selected $[x_r, y_r] = [0, 0]$ and the reference linear velocity, $v_r = 0.03$.

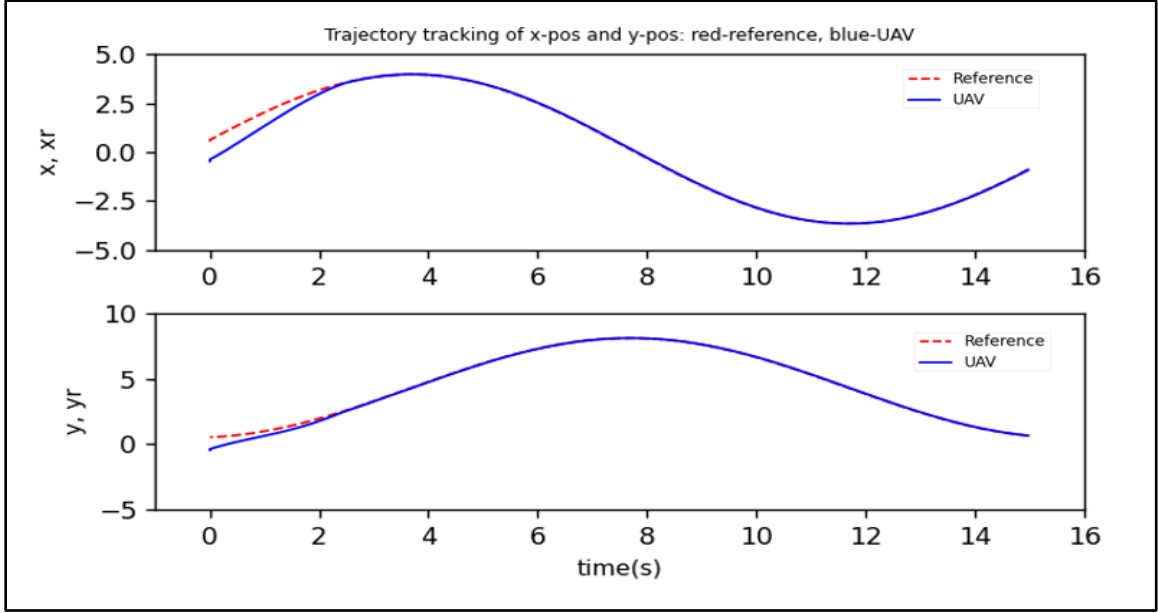


Figure 6.6: Single UAV reference tracking

The tracking error e_y and e_x is plotted as shown in **Figure 6.7**. Further a circular smooth trajectory with initial heading direction $\theta_d = 5 \text{ rad}$ and desired linear velocity $v_r = 0.03$ units and desired angular velocity $w_r = 0.45 \text{ rad/iteration}$ was generated to illustrate the trajectory tracking by designed controller as shown in **Figure 6.8**.

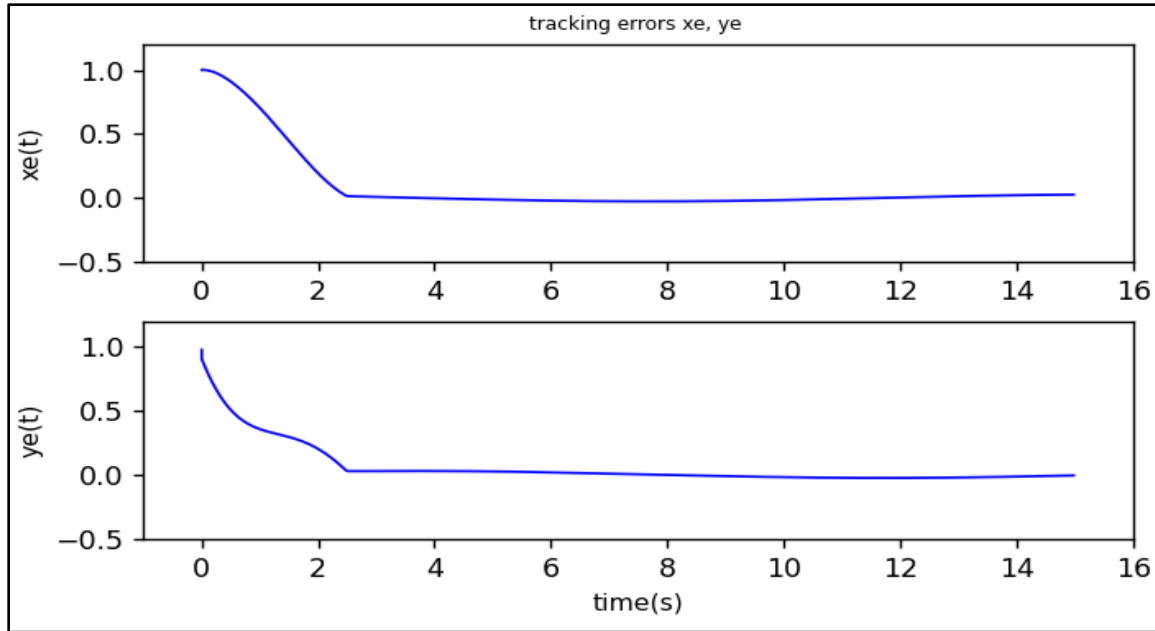


Figure 6.7: Orientation tracking error for single UAV

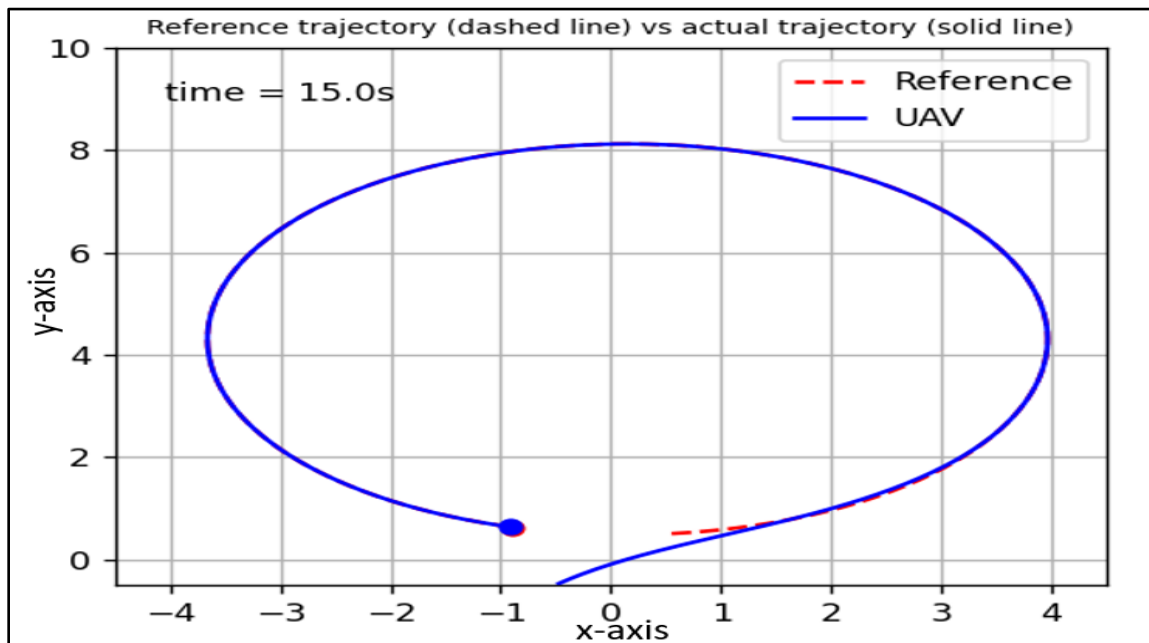


Figure 6.8: Reference tracking for circular trajectory

The reference tracking of the agent in the presence of wind turbulence region is shown in **Figure 6.9**. The agent passes through the region of turbulence in between 7 to 9 seconds.

A disturbance in the trajectory can be observed in the error tracking as shown in **Figure 6.10**. Similarly, in **Figure 6.11** the disturbance in the trajectory is observed however the controller successfully tracks the reference in the region of turbulence.

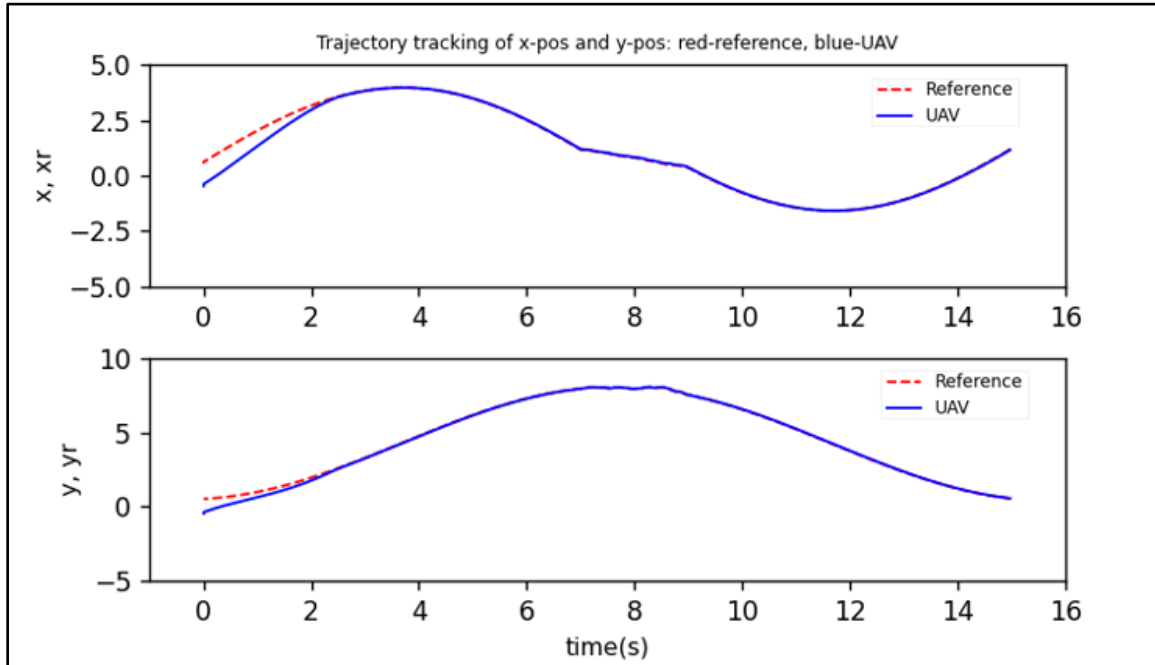


Figure 6.9: Single UAV Reference tracking under wind turbulence

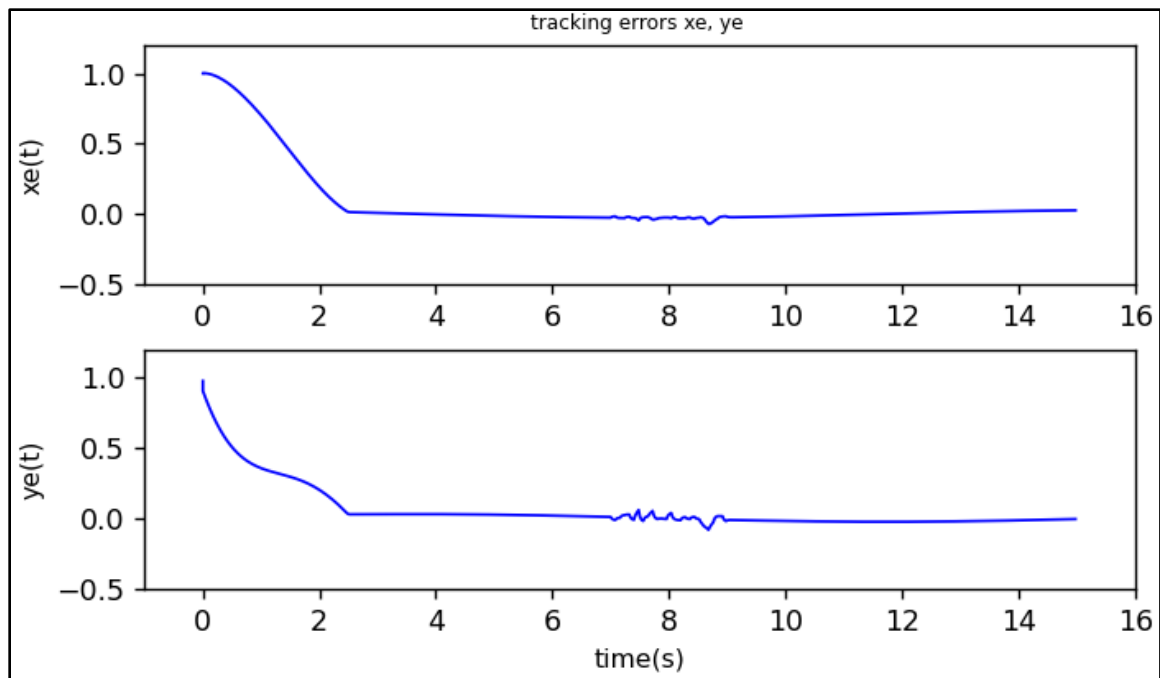


Figure 6.10: Orientation tracking error for single UAV under wind turbulence

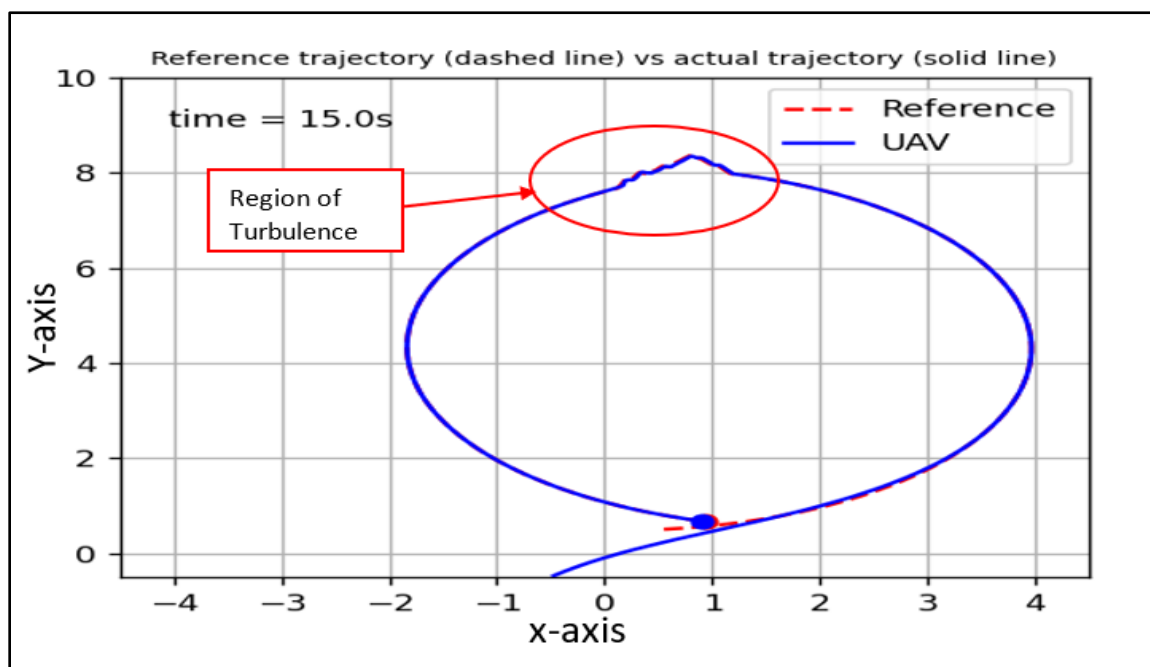


Figure 6.11: Reference tracking for circular trajectory under wind turbulence

From the above results we can say that the designed controller is able to track the reference even in the region of turbulence which proves designed controller is robust.

6.4 Multiple UAV Tracking Analysis

In this section we discuss the multi agent tracking. During multiple UAV navigation, the agents while foraging and evasion wish to move coherently and avoid any collision among themselves. Since the agents know the position of their neighbouring UAV agents, therefore they maintain a safe distance between themselves. To do so we use the concept of potential field strategy discussed in **Chapter 4, section 4.2.1 (C)**.

The Simulation results for multiple UAVs tracking a desired reference is shown in **Figure 6.12**. It can be seen that multiple UAVs from distinct initial position track the straight-line reference to reach the target. The UAVs move towards the reference trajectory however they maintain a safe distance between them and coherently move towards the target without any collision.

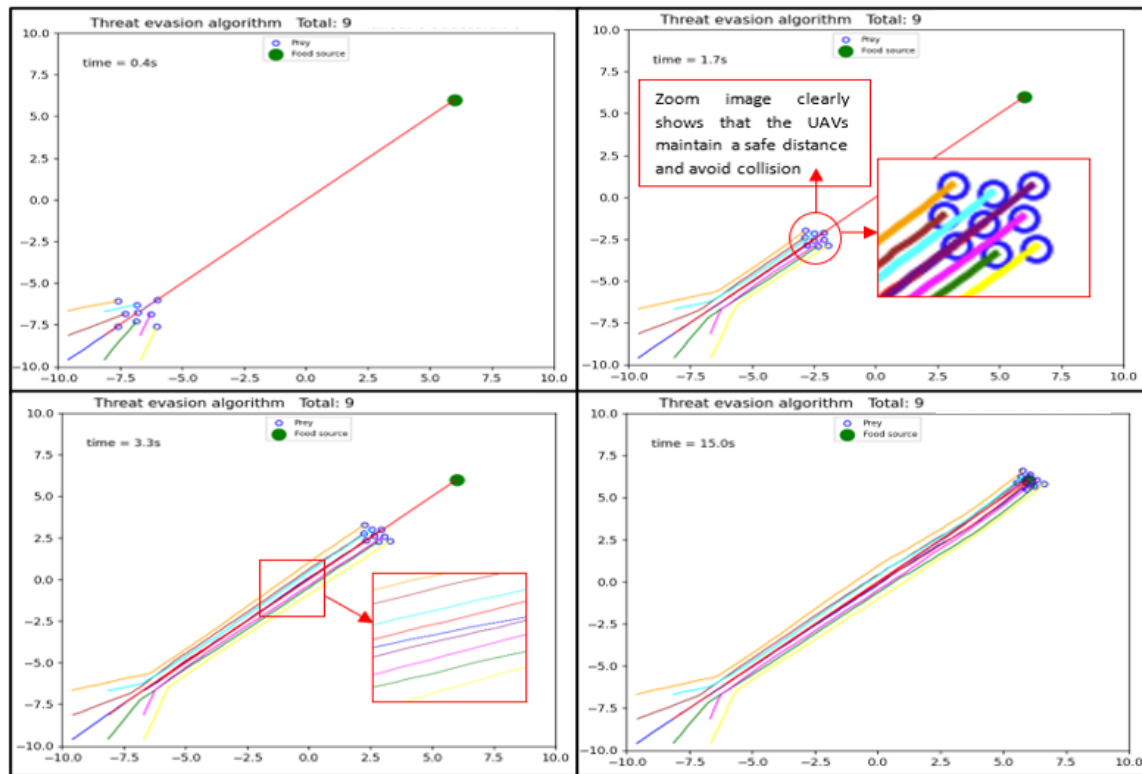


Figure 6.12: Multiple UAV tracking

CHAPTER 7

RESULTS AND ANALYSIS

In this chapter we will discuss the simulation and results for single predator attack and multi predator attack with cooperative hunting. First, we will exhibit the single predator attack with foraging and evasion for nodes in absence and in presence of threat respectively. Then we show the result for multi predator attack with cooperative hunting strategy where the predators will hunt the UAV swarm using the state transition model. The variable and parameters for each case will be updated in the coming sections. The parameters chosen are subjective and can be varied depending how you wish you design your network. For example, the radius can be changed, the minimum and maximum distance between the nodes can be changed to avoid collision and realign if separated.

7.1 Single Predator Attack

In this section we will showcase the scenario where a single predator attacks the group of UAVs foraging towards the food target. Since the predator can attack from any region and hence all four quadrants are considered in terms of the desired evasion or escape angle. The variables and parameters used for simulation is given in **Table 7-1**. The legends used for fleet of UAVs, predator and target is as under.

-  : Fleet of UAVs
-  : Predator
-  : Target

Table 7-1: Simulation table for single Predator Attack

Variables	Meaning	Values
N	Number of UAVs	36
N_k^{max}	Maximum Number of Neighbors of UAV "k"	6
$w_{k,l}^p$	Initial Position of Predator	(-6, -6) or (5,5)
$w_{k,l}^f$	Initial Position of Food Target	(6,6)
Δt	Sampling Time	0.1
r_1	Predator Inner Radius	1.5
r_2	Predator Outer Radius	3.0
$r \pm \epsilon$	Upper and Lower Threshold for Swarm Nodes	0.5 ± 0.1
μ_k	Non-negative Stepwise Scalar	1.0

7.1.1 Foraging Phase

In this section we exhibit the fleet of UAVs foraging towards the target under no threat.

The initial number of the Swarm is 36 as shown in **Figure 7.1**.

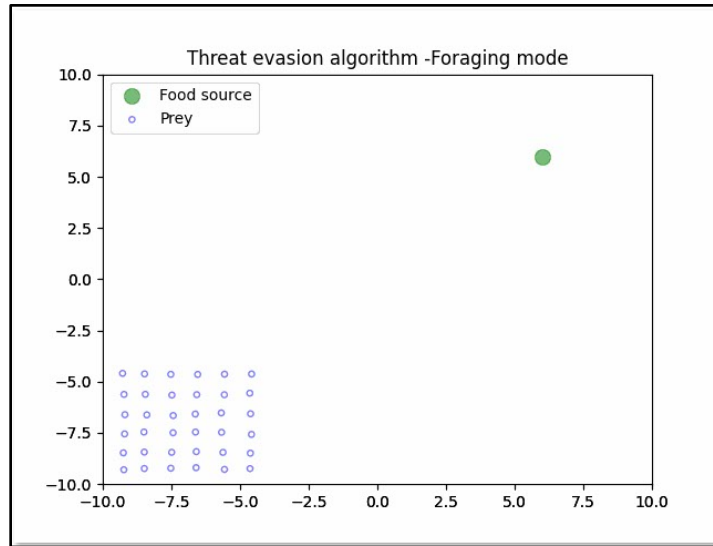


Figure 7.1: UAV Fleet aligned to forage towards target

The UAVs can be seen advancing towards the target in a coherent manner as shown in

Figure 7.2.

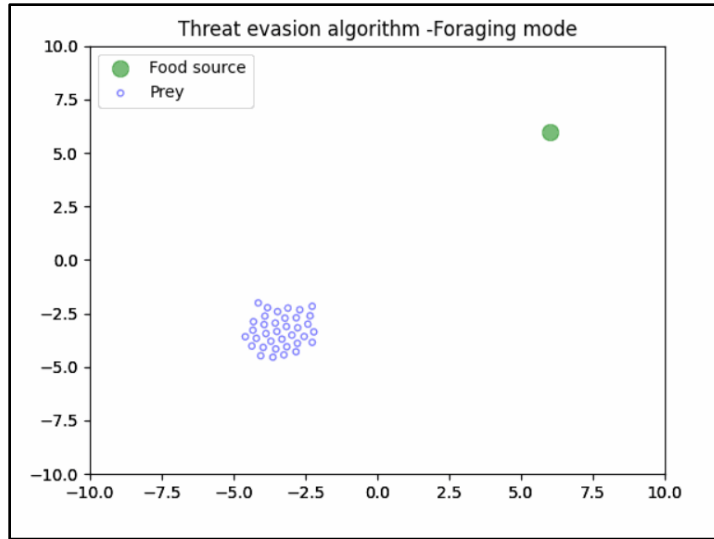


Figure 7.2: Foraging Behavior 1

The Fleet of UAVs maintain a safe distance between them and avoid collision while foraging as shown in **Figure 7.3**.

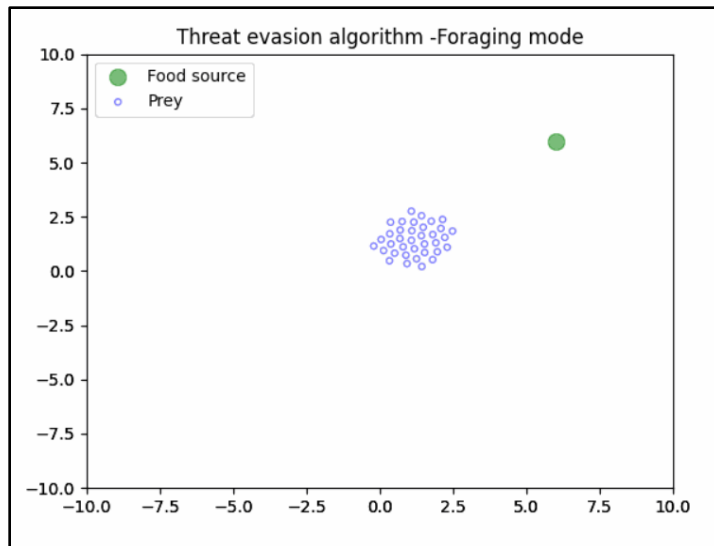


Figure 7.3: Foraging Behavior 2

The fleet can be seen reaching the target in a self-organised manner, mimicking the biological behaviour of school of fishes as in **Figure 7.4**. The UAVs can be seen strictly knitted in an adaptive network while foraging. They do not collide, nor do they get

separated with time. This is obvious from shape transition of the fleet from the start to end until it reaches the target.

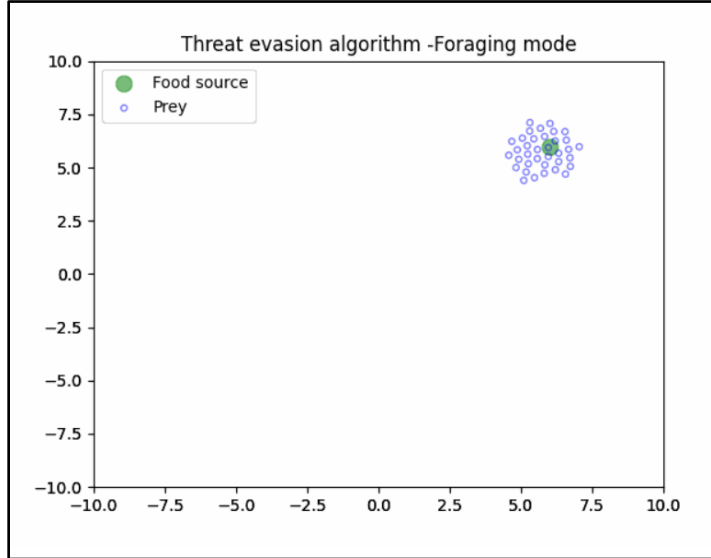


Figure 7.4: Foraging Behavior 3

7.1.2 Foraging and Evasion

In this section we add mobility to threat. The fleet of swarm can be seen foraging and evading simultaneously in absence of and in presence of threat respectively. We will showcase the attack from each quadrant to exhibit the evasion of swarm in different scenario.

Figure 7.5 show the responses of the UAVs when the predator attacks from Q1. The initial position of predator is $[x, y] = [2.5, 2.5]$. The maximum velocity for swarm is $v_{max} = 0.1 \text{ units}$ while for the predator $v_{max}^p = 0.14 \text{ units}$

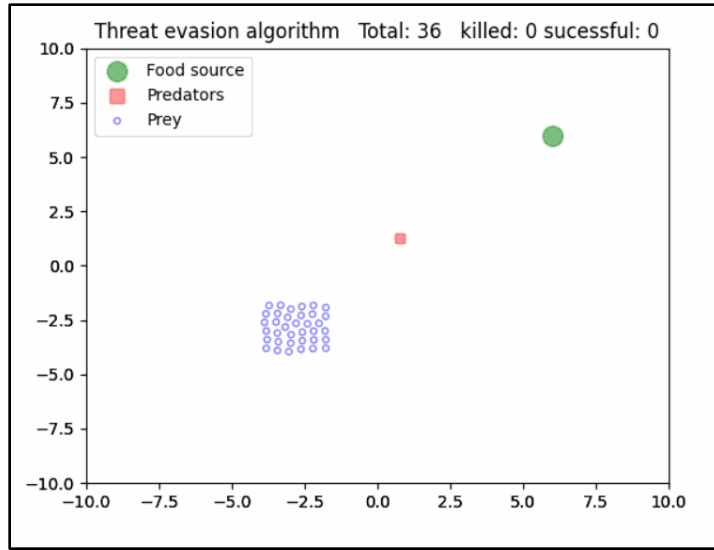


Figure 7.5: Q1 Attack - Foraging & Evading Behavior 1

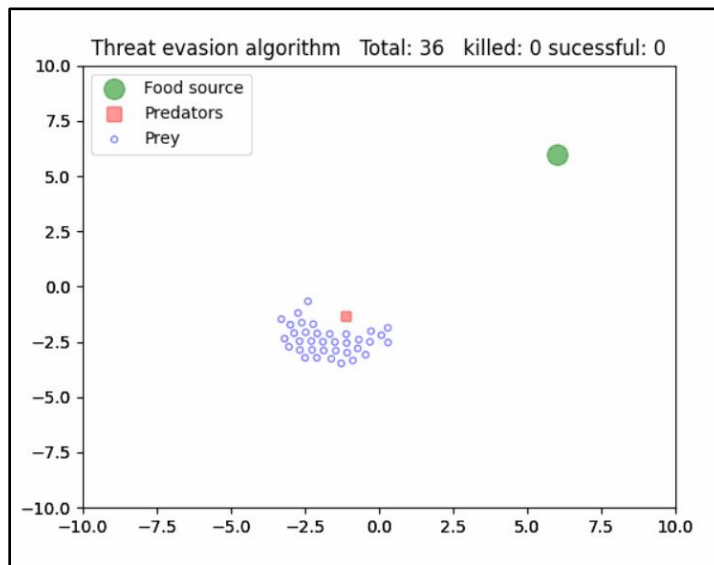


Figure 7.6: Q1 Attack - Foraging & Evading Behavior 2

The predator can be seen attacking the Fleet of UAVs as shown in **Figure 7.6**. The UAVs in the fleet sensed the danger of the threat coming in the threat zone and have started to evade. The Swarm of UAVs can be seen spreading out to escape the attack.

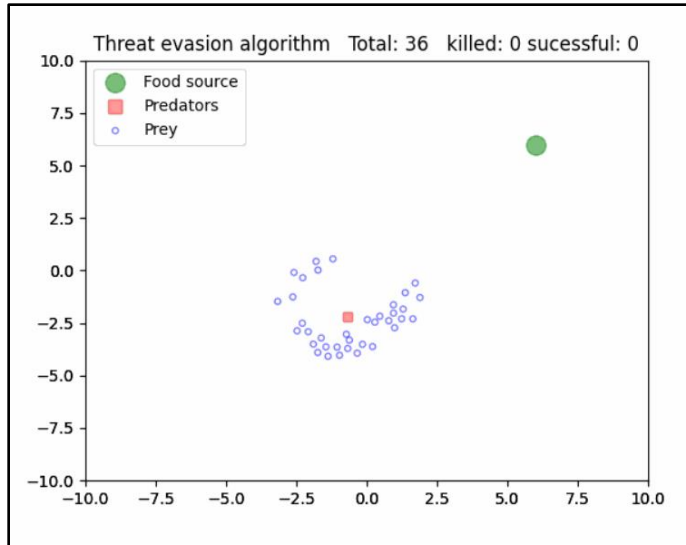


Figure 7.7: Q1 Attack - Foraging & Evading Behavior 3



Figure 7.8: Q1 Attack - Foraging & Evading Behavior 4

The Fleet of UAVs can be seen fragmenting into smaller groups to escape that attack as shown in **Figure 7.7** and **Figure 7.8**. While evasion the swarm of UAVs can be seen maintaining a safe distance amongst each other to avoid collision. The predator can be seen chasing the nearest swarm as in **Figure 7.8**. while the other UAVs can be seen regrouping and foraging towards the target as in **Figure 7.9**. The UAVs which reach the target are marked successful, while ones which comes in direct contact with predator are marked

killed. Towards the end the 30 UAVs successfully reached the target while 6 were killed as shown in **Figure 7.10**.

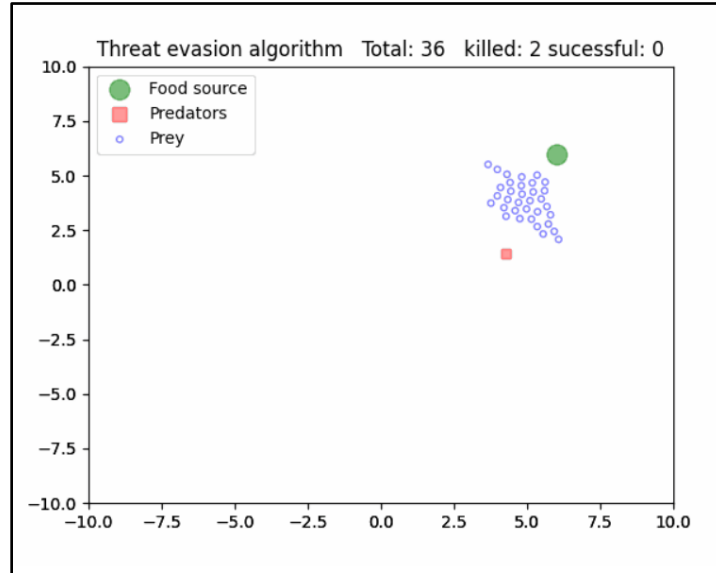


Figure 7.9: Q1 Attack - Foraging & Evading Behavior 5

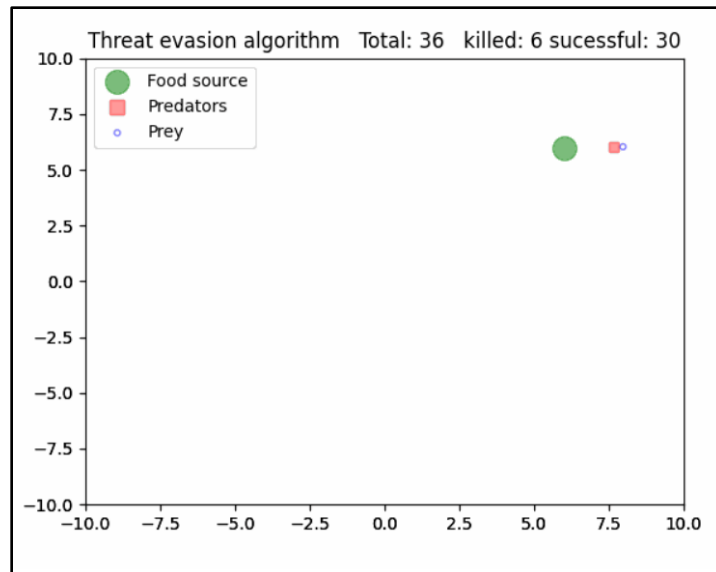


Figure 7.10: Q1 Attack - Foraging & Evading Behavior 6

We now exhibit the attack from the predator from initial position from Q2 with initial position at $[x, y] = [-7.5, 2.5]$.

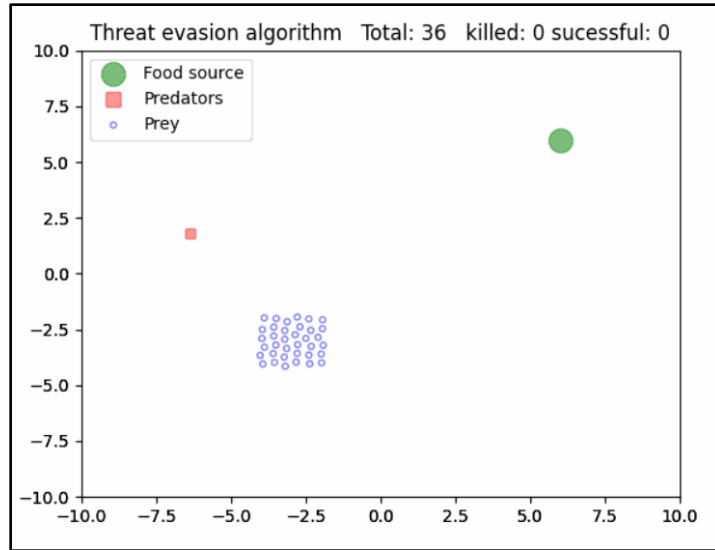


Figure 7.11: Q2 Attack - Foraging & Evading Behavior 1

The threat can be seen advancing towards the fleet of UAVs foraging towards the target as in **Figure 7.11**. The predator attacks the fleet, while the UAV nodes can be seen switching to evasion from foraging mode as in **Figure 7.12**

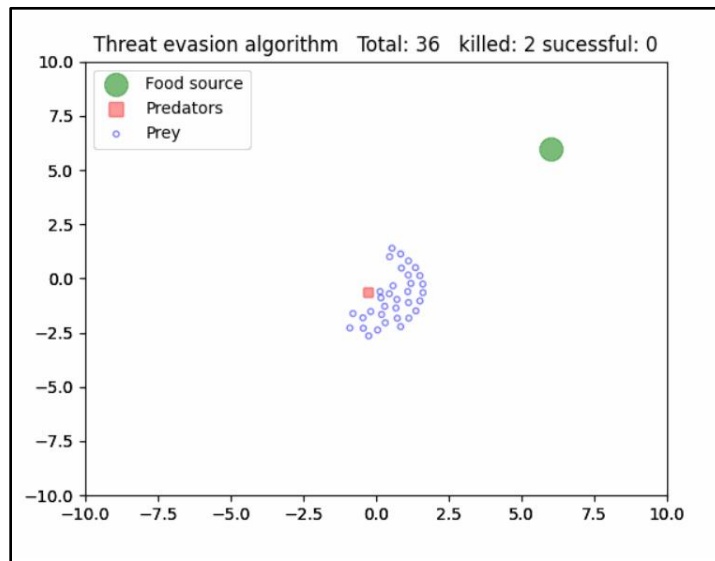


Figure 7.12: Q2 Attack - Foraging & Evading Behavior 2

The fleet of UAVs can be seen evading the attack, at the same time they maintain a safe distance between them to avoid collision as in **Figure 7.13** and **Figure 7.14**. the fragmented

group of UAVs can be seen recombining while foraging towards the target as in **Figure 7.14**.

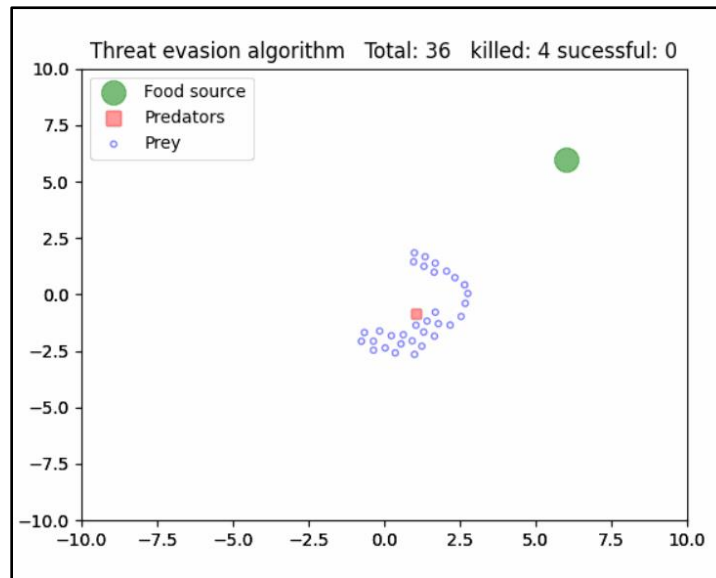


Figure 7.13: Q2 Attack - Foraging & Evading Behavior 3

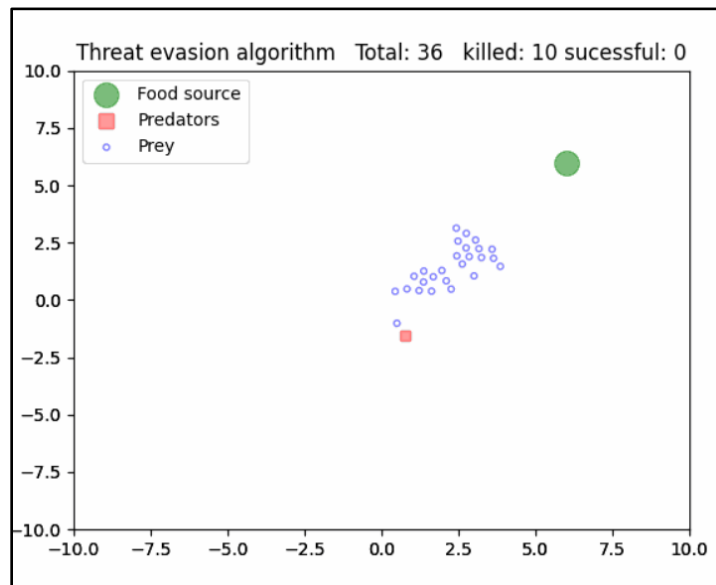


Figure 7.14: Q2 Attack - Foraging & Evading Behavior 4

A group of UAVs from the entire fleet of 36, that reaches the target location are the ones which are successful in the mission and are marked successful. While the ones that comes

in direct contact with the predator UAVs are destroyed and marked as killed as shown in **Figure 7.15** and **Figure 7.16** .

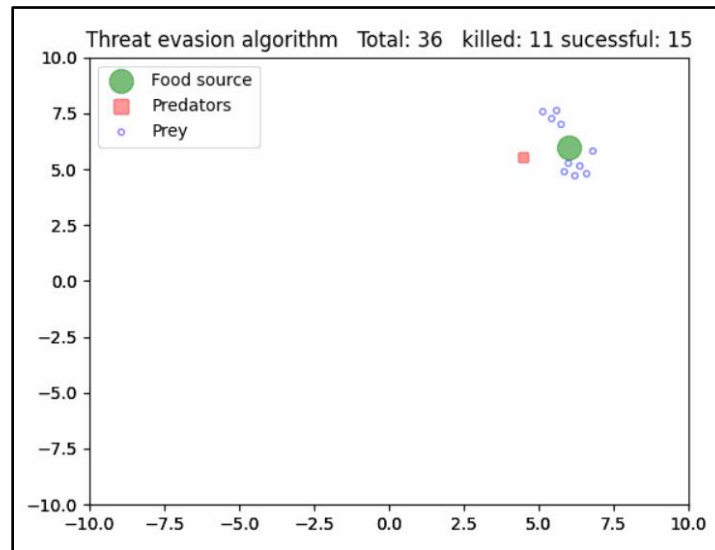


Figure 7.15: Q2 Attack - Foraging & Evading Behavior 5

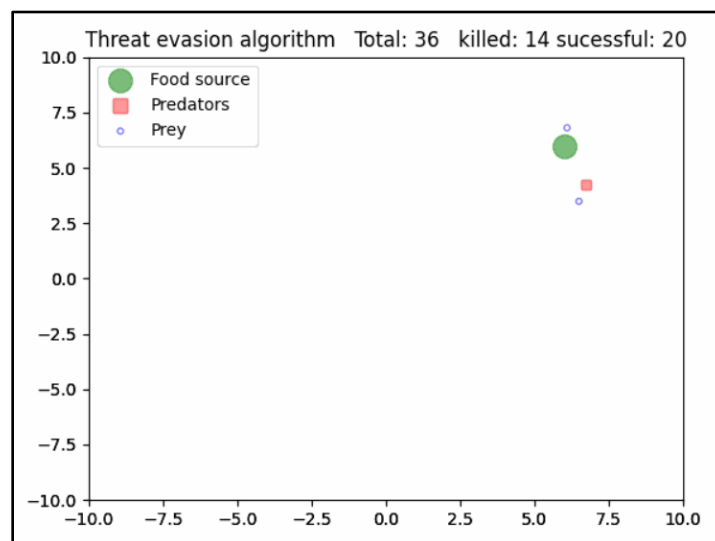


Figure 7.16: Q2 Attack - Foraging & Evading Behaviour 6

Similarly, we exhibit the attack from quadrant 3 with initial position of predator at $[x, y] = [-4, -7.5]$. The foraging and evasion can be seen as shown in **Figure 7.17**. Finally, we

exhibit the attack from quadrant 4 with initial position of predator at and $[x, y] = [6.0, -7.5]$. The foraging and evasion can be seen as shown in **Figure 7.18**.

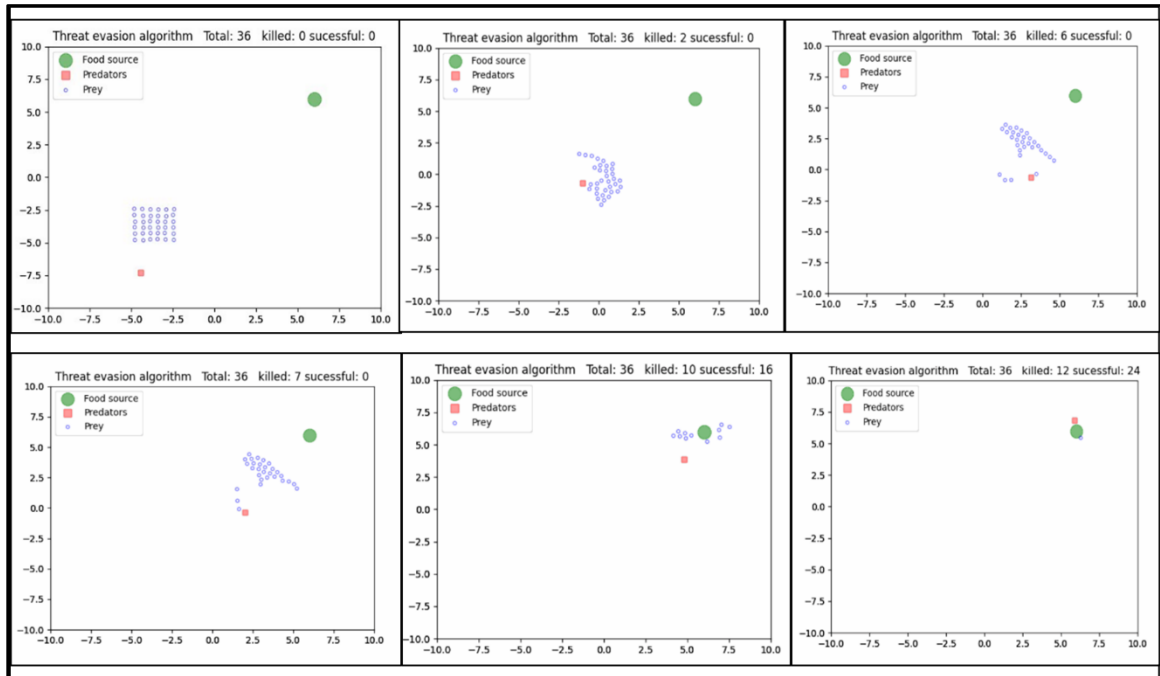


Figure 7.17: Q3 Attack - Foraging & Evading Behavior

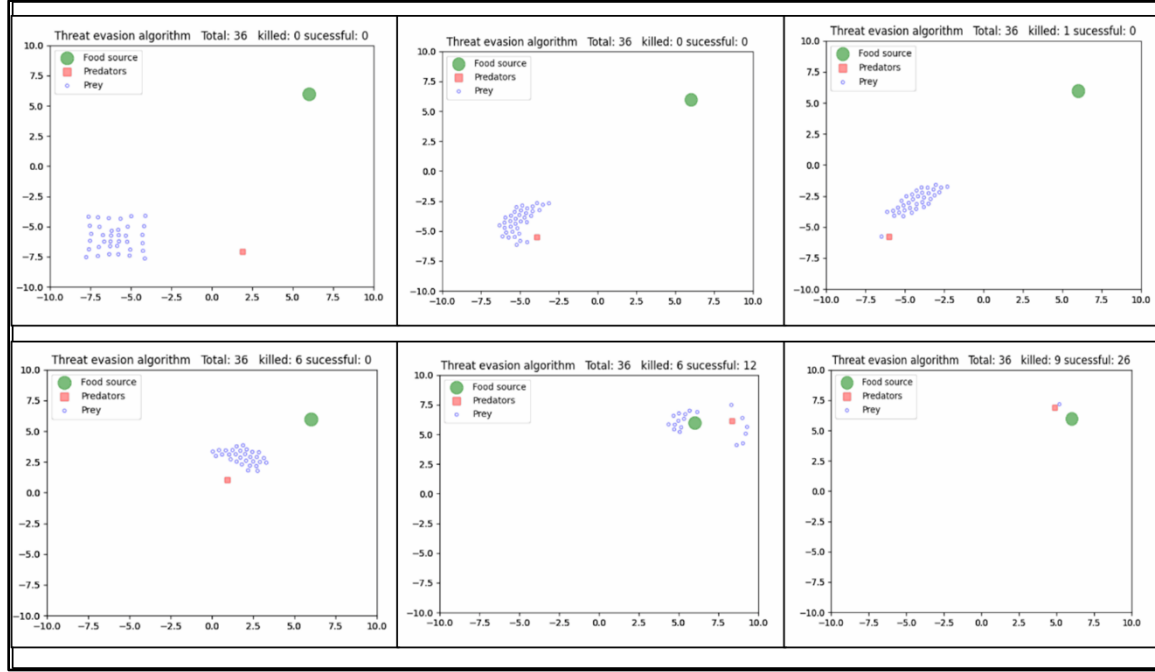


Figure 7.18: Q4 Attack - Foraging & Evading Behavior

7.2 Single Predator Attack with Obstacle and Wind Turbulence

To emulate real world scenario, where the region of action can have obstacle such as buildings and trees, we introduced unstructured obstacle in the environment. The obstacle are circular blocks randomly generated during each simulation. The location of obstacle is unknown to both the swarm and predator. The position of the obstacle is estimated using the ATC diffusion algorithm discussed in **Section 4.2.1 (A)**. Both the swarm agent and predator avoid collision with the obstacle using the concept of potential field. When the agents are close enough to the obstacle, say $(r + \delta)$, where r is the radius of the obstacle and δ is a small distance, the agents switch to evasion algorithms discussed in **section 4.3** until they are away from the obstacle.

To add further complexity a wind turbulence region was created as discussed in section 6.3 that mimics the extreme weather conditions such as storm or typhoon.

7.2.1 Single UAV Foraging with Obstacle and Turbulence

The simulation results for tracking of the navigation algorithm during the foraging mode of a single swarm in presence of obstacle and wind turbulence is shown in **Figure 7.19**. The swarm agent navigates the environment smoothly until it reaches the region of high wind turbulence where it experiences a little turbulence because of wind and high noise. However, the swarm agent is able to successfully navigate through the region of turbulence avoiding the obstacle present in the region and reach the desired target.

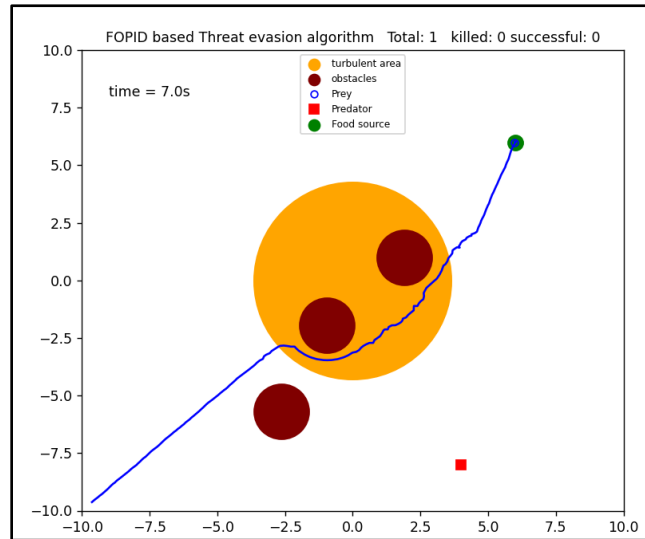


Figure 7.19: Foraging of single swarm in presence of obstacle and wind turbulence.

7.2.2 Single UAV Foraging and Evasion with obstacle and Wind Turbulence.

Similarly, the simulation results for tracking of the navigation algorithm during the foraging and evasion of a single swarm under attack in presence of obstacle and wind turbulence is shown in **Figure 7.20** and **Figure 7.21**.

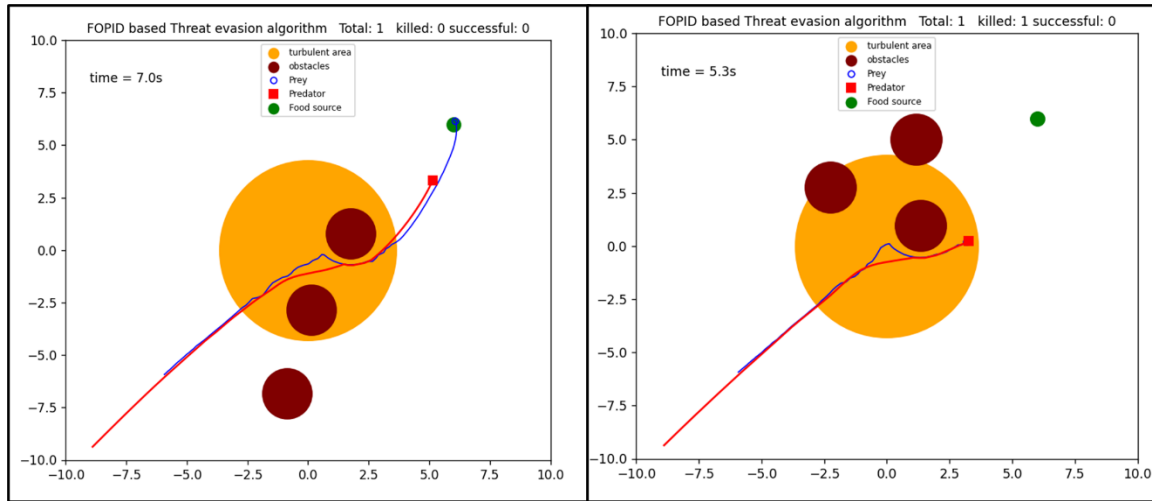


Figure 7.20: Single UAV foraging and evasion with obstacle and wind turbulence- case I

In both the figures we have presented two cases. In both the cases it can be seen the UAV tries to escape the predator attack and reach the target while avoiding collision with the obstacle. The predator also avoids the obstacle while chasing the UAV agent. In one scenario the predator successfully kills the agent whereas in another scenario the Swarm agent makes a successful escape.

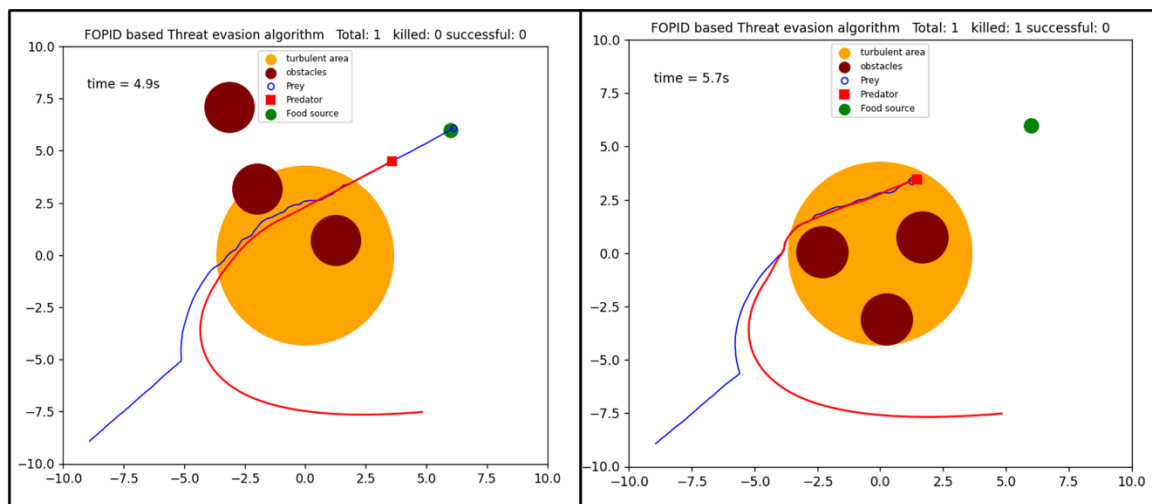


Figure 7.21: Single UAV foraging and evasion with obstacle and wind turbulence- case II

7.2.3 Multiple UAV Foraging and Evasion with Obstacle and Wind Turbulence

While the Simulation results for multiple UAV foraging in the presence of wind and obstacle is shown in **Figure 7.22**. It can be seen that the swarm of UAVs coherently forage towards the target maintaining a safe distance between them. At the same time, they avoid any collision with the obstacle during foraging. The movement of swarm UAVs in the region of wind turbulence is also very smooth. They do not get distracted or randomize their movement in the region of wind turbulence. This is possible due to the design of effective controller.

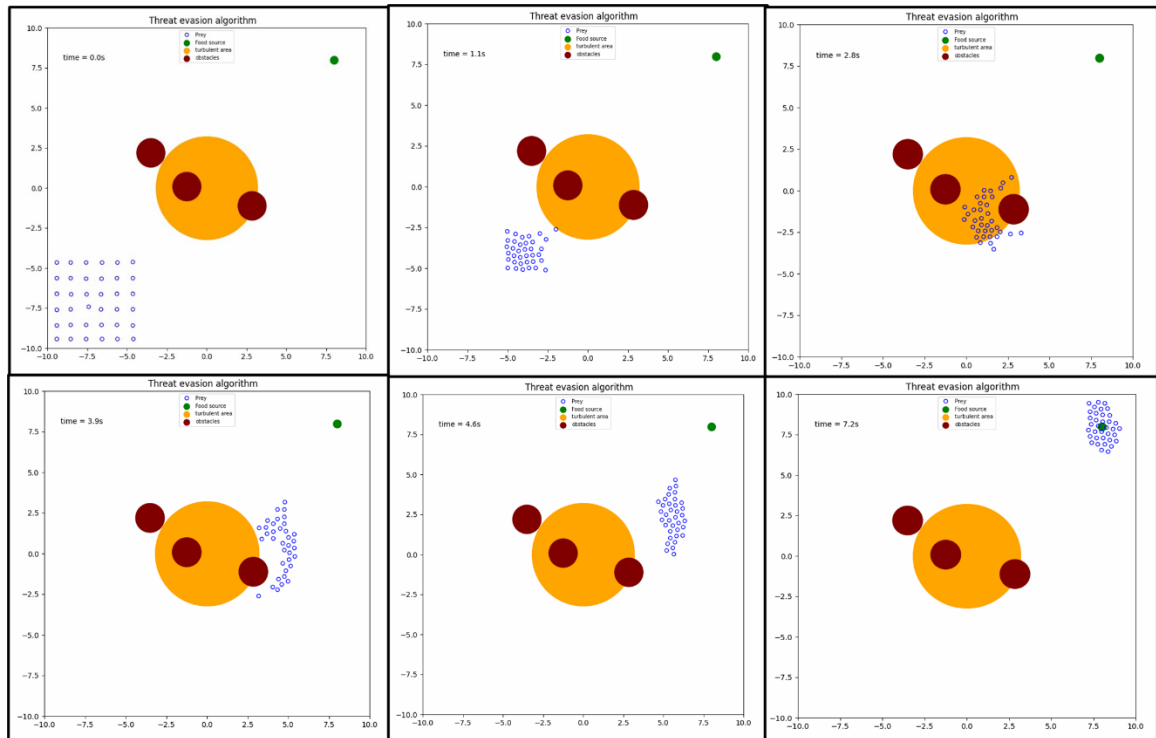


Figure 7.22: Multiple UAV foraging in the presence of wind and obstacle

Further the Simulation results for multiple UAV foraging and evasion when under attack in the presence of wind and obstacle is shown in **Figure 7.23**. It can be seen that the

predator moves towards the swarm agents while the swarm agents forage towards the food target until the predator is close enough and launches attack as in second and third snap of **Figure 7.23**. The swarm agents can be seen avoiding collision with the obstacle as well as trying to escape the predator attack. The predator also avoids any collision with the obstacle while hunting.

The result exhibit smooth foraging and evasion by the swarm agents even in the region of wind turbulence without any collision. Also, the swarm agents avoid any collision with obstacle during foraging as well as during evasion when under attack from the predator agent. Similarly, the predator can be seen attacking the swarm UAVs while avoiding any collision with the obstacle.

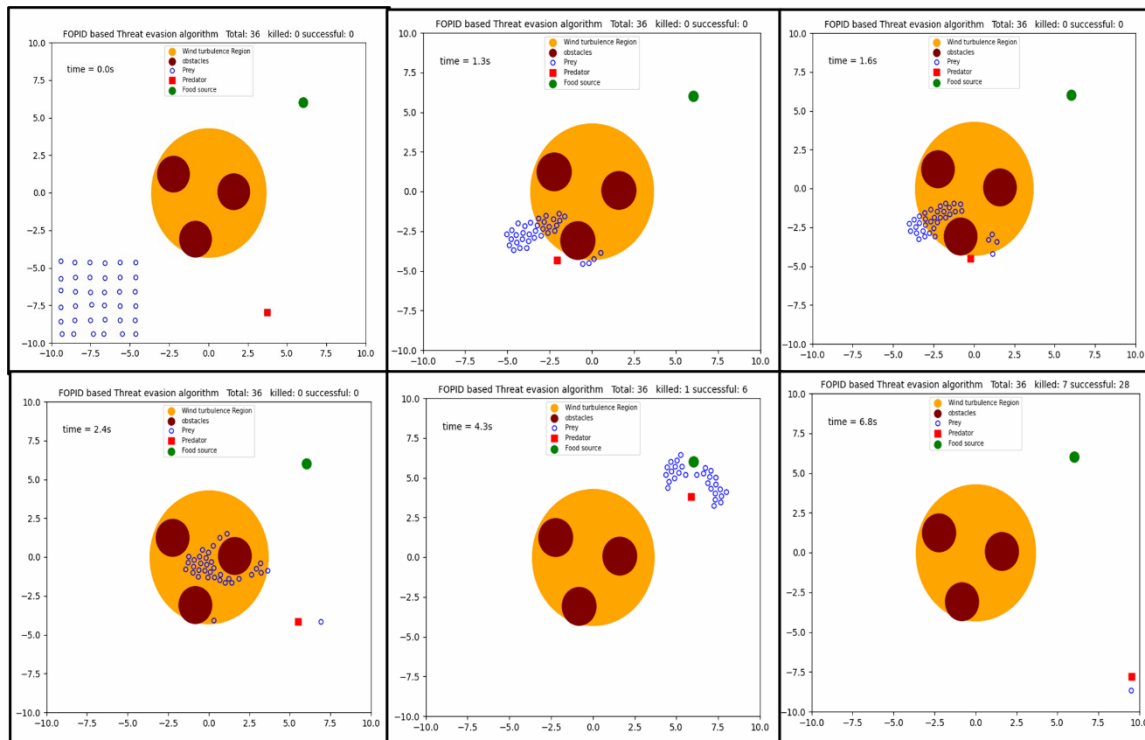


Figure 7.23: Multiple UAV foraging and evasion in the presence of wind and obstacle

7.3 Multiple Predator Attack - Cooperative Hunting

In this section we will exhibit the simulation results for cooperative hunting strategy with multi predator. A group of six predators will attack a group of 36 swarm of UAVs foraging on a mission. The predator can be seen successfully chasing the fleet of UAVs and then encircle them and trap them before launching the attack. While the Swarm of UAVs will try to escape and evade the attack. The simulation results will demonstrate the air combat warfare scenario with multi attacker.

The variables and parameters used for simulation is given in **Table 7-2**.

Table 7-2: Simulation table for Multi-Predator Attack- Cooperative Hunting

Variables	Meaning	Values
N	Number of UAVs	36
N_k^{max}	Maximum Number of Neighbors of UAV "k"	6
M	Number of Predator UAVs	6
$w_{k,l}^f$	Initial Position of Food Target	(6,6)
Δt	Sampling Time	0.5
r_1	Predator Inner Radius	1.5
r_2	Predator Outer Radius	3.0
r_3	Outlier's Radius	0.5
r_g	Encircle Radius	3.0
r_s	Radius of predator during encircling to check other predator in the region	1.2
r_p	Radius of predator to avoid collision	0.5
$r \pm \epsilon$	Upper and Lower Threshold for Swarm Nodes	0.5 ± 0.1
μ_k	Non-negative Stepwise Scalar	1.0
v_{max}	Maximum velocity for swarm Nodes	0.12
v_{max}^p	Maximum velocity for Predator Nodes	0.18

The initial position of the predators has been set in the fourth quadrant as shown in **Figure 7.24**. The Predator UAVs starts to chase the Fleet of UAVs foraging on a mission to reach a stationary target at $[x, y] = [6, 6]$ as shown in **Figure 7.25**. The predator is in State 0 i.e., Chase.

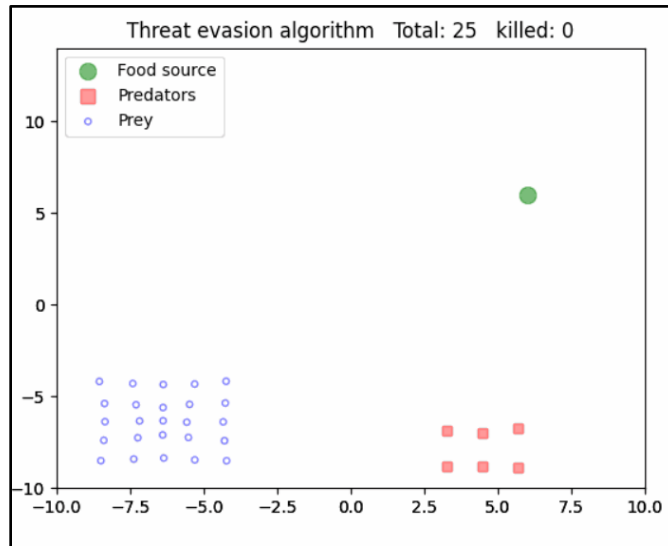


Figure 7.24: Case1-Multi-Predator Attack -Frame 1

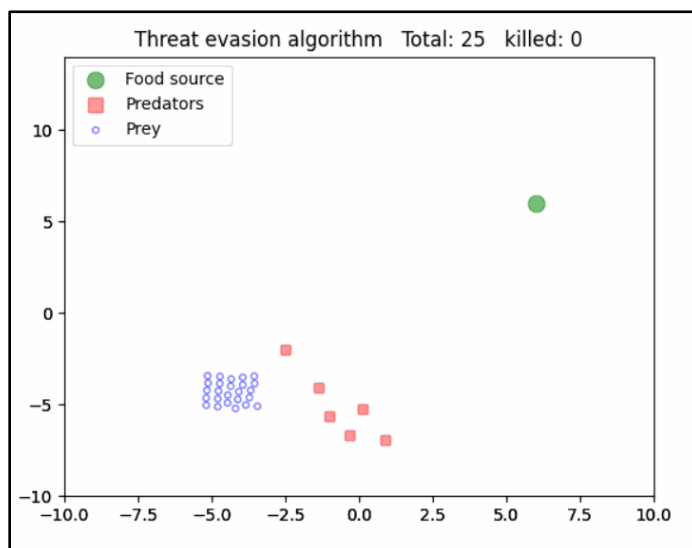


Figure 7.25: Case1-Multi-Predator Attack -Frame 2

The predator group can be seen going across the fleet of UAVs to block them from foraging and trying to encircle them as shown in **Figure 7.26**, while in the next frame the predators can be seen completely encircling the Fleet of UAVs as in **Figure 7.27**.

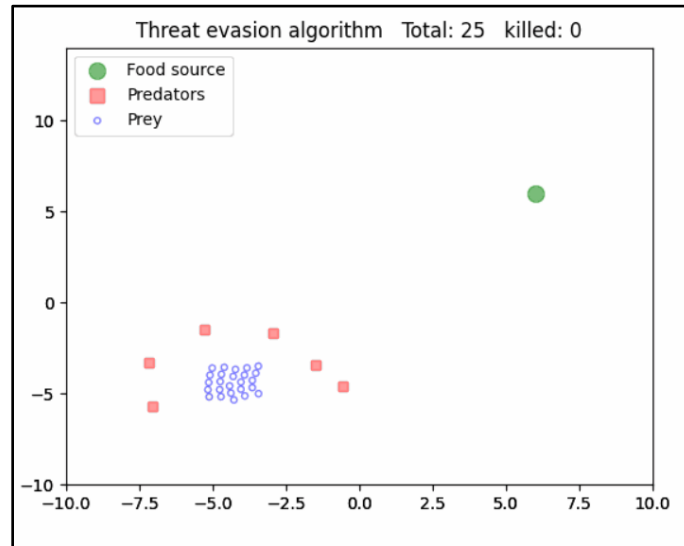


Figure 7.26: Case1-Multi-Predator Attack -Frame 3

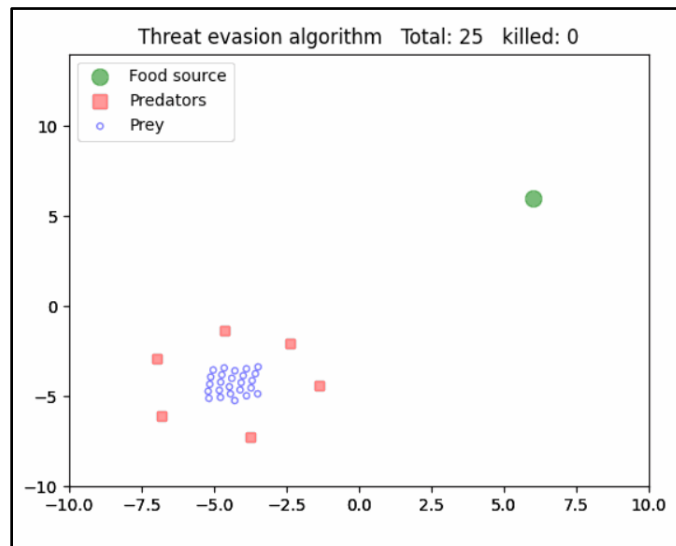


Figure 7.27: Case1-Multi-Predator Attack -Frame 4

The predator in **Figure 7.27** is in State 1, i.e., Encircle Mode. Once the encircle is complete the Predator will launch attack on the trapped fleet of UAVs as in **Figure 7.28**. The predator group is now in State 2 i.e., Attack.

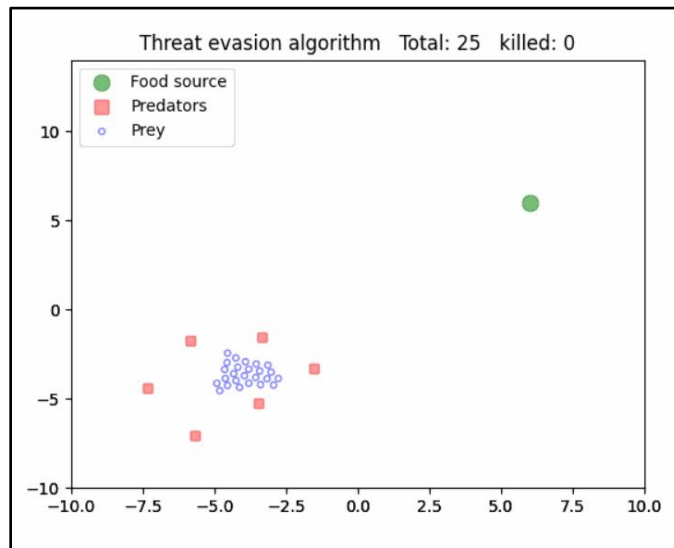


Figure 7.28: Case1-Multi-Predator Attack -Frame 5

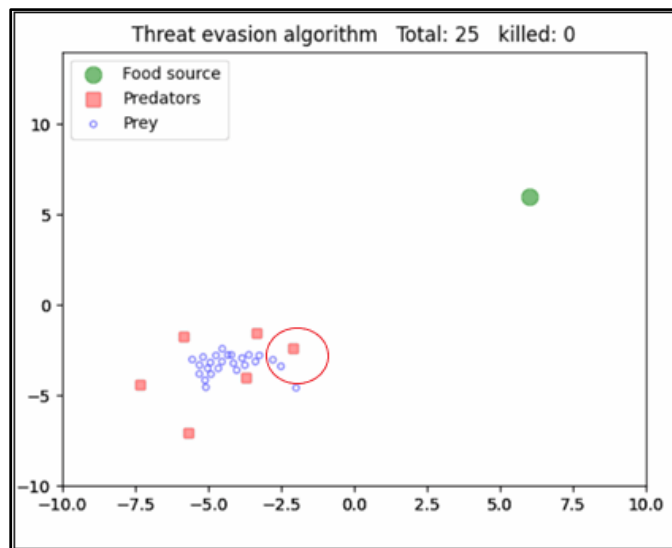


Figure 7.29: Case1-Multi-Predator Attack -Frame 6

Once the attack is initiated the Fleet of UAVs try to evade and escape the attack as shown in **Figure 7.29**. The encircled predator in frame 6 can be seen moved anticlockwise from

its position in frame 5 to push an outlier trying to escape. However, some of the UAVs from the fleet manages to escape as seen in **Figure 7.30**.

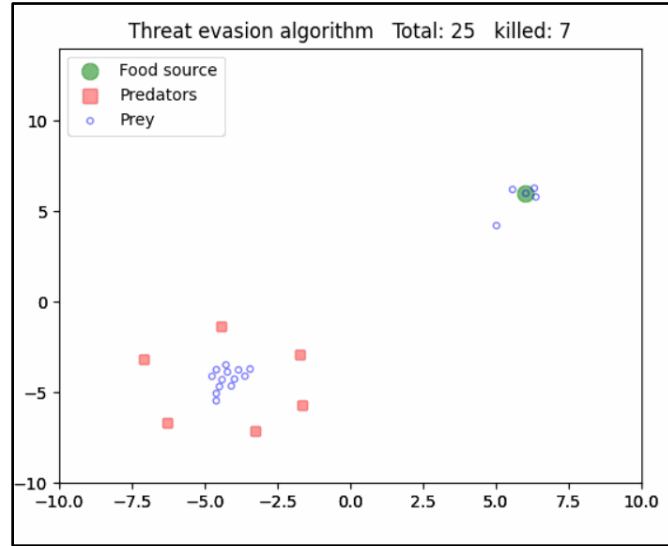


Figure 7.30: Case1-Multi-Predator Attack -Frame 7

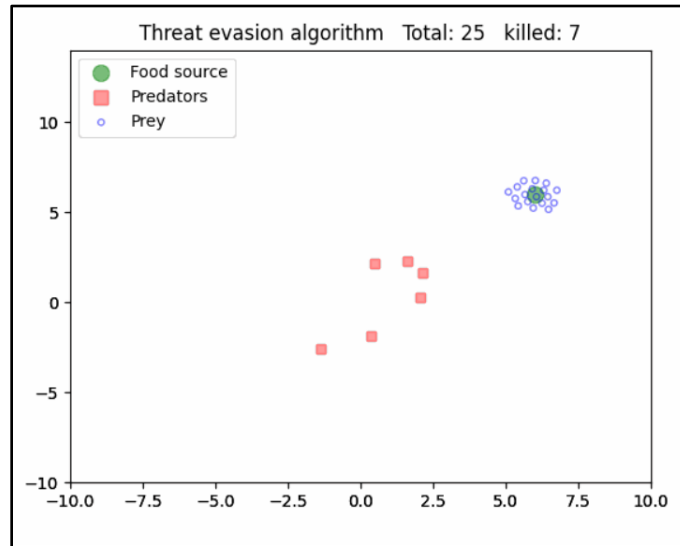


Figure 7.31: Case1-Multi-Predator Attack -Frame 8

Once the complete group from the remaining Fleet of UAVs escapes the predator UAVs can be seen going back to State 0, i.e., chase mode as seen in **Figure 7.31**.

The predator group encircles the Swarm of UAVs around the food target and launch attack again until UAVs escape to some safe destination, out of the environment which the predator is unaware of as can be seen in **Figure 7.32**. Once all the UAVs are either killed or escape to safe zone the predator returns to its origin as shown in **Figure 7.33**.

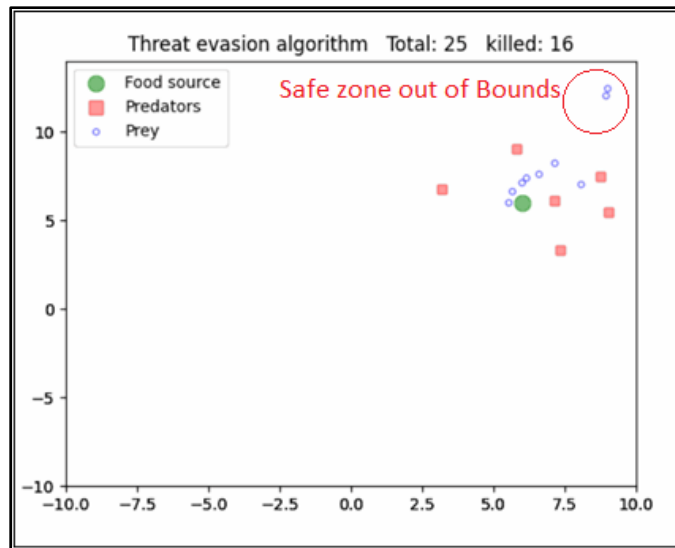


Figure 7.32: Case1-Multi-Predator Attack -Frame 9

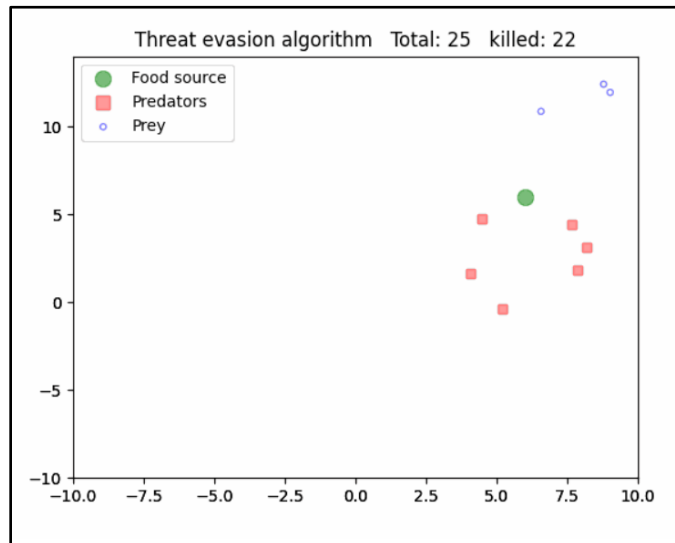


Figure 7.33: Case1-Multi-Predator Attack -Frame 10

7.3.1 Cooperative Hunting Under Wind Turbulence

In this section we introduced a wind turbulence region in the cooperative hunting scenario, to see the predator response under wind turbulence. It can be seen from **Figure 7.34** that while the predator attacks and encircles the swarm agents in the region of high wind turbulence the swarm agent tries to escape the predator attack smoothly. The designed controller is successful in controlling the swarm to navigate as per the designed algorithms. The wind effect does not disturb the escape behavior of swarm even when under attack from predator.

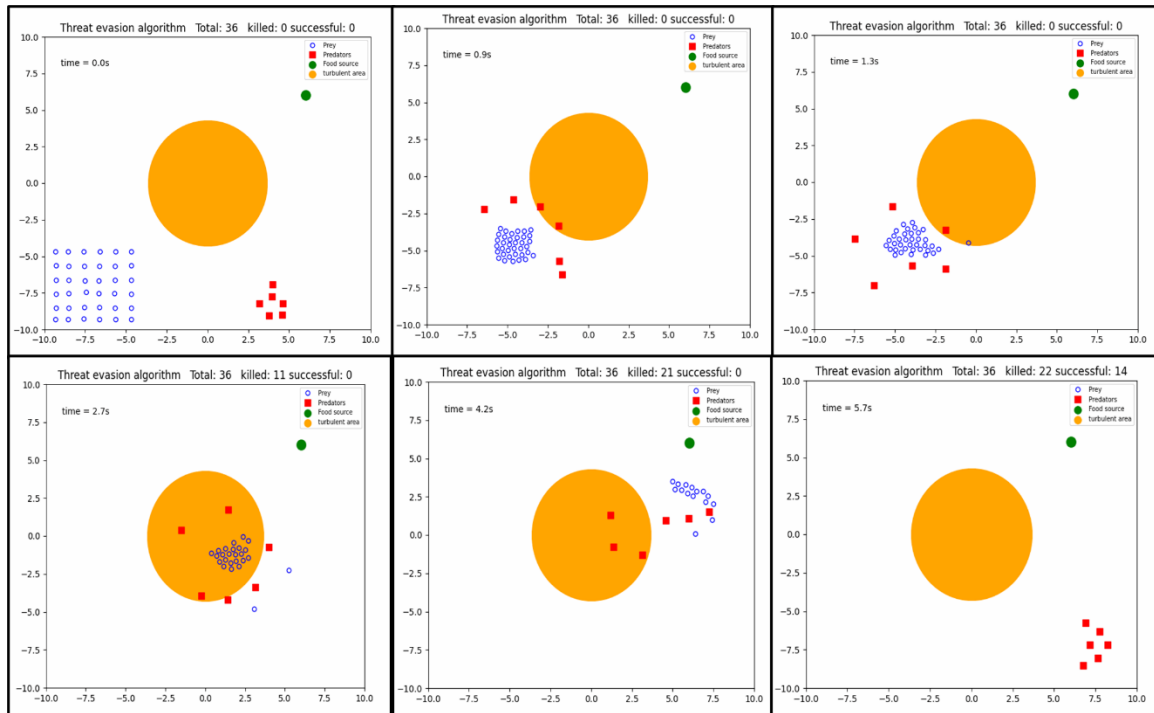


Figure 7.34: Cooperative Hunting Under Wind Turbulence

CHAPTER 8

CONCLUSION AND FUTURE STUDY

In this work we successfully implemented bio inspired algorithms for Multi agent system. The Navigation algorithms were based on diffusion and adaptation model. The simulation results exhibited the effectiveness of the work and demonstrated how biological behaviors can be mimicked to devise strategy for solving complex real-world issues.

The simulation results for both single predator attack and multiple predator attack were presented. The swarm of UAVs on mission to reach the target while escaping the attack. While some of the UAVs were successful in reaching the target few got killed during the attack. We envisioned this bio-inspired behavior of the swarm of fishes during foraging and evasion, like air combat warfare scenario. However, we restricted our model to a nonholonomic unicycle model. We also designed a robust optimal FOPID controller to ensure the trajectory tracking of the swarm of UAVs. To Emulate the environment with real world we introduced unstructured obstacle mimicking building and tress. The swarm of UAVs and the predators were able to successfully navigate avoiding any collision. Further to add more complexity we also introduced a Wind turbulence region mimicking real world storm or typhoon that could affect the motion of the UAVs and disturb their navigation. However, our designed robust optimal FOPID controller was able to successfully control the trajectory of the UAVs as per the designed navigation algorithm.

In future this could be implemented on a fully autonomous aircraft or drones. The current work in implemented in a 2D plane, in future work this can well be implemented in a 3D

model. To test the effectiveness of the algorithms mobile obstacle can be introduced. Similarly, the target can be made mobile to add further complexity.

Also, a futuristic approach could be using Artificial intelligence or Machine learning on top of the biologically inspired algorithms making UAVs more intelligent during foraging and Evasion.

REFERENCES

- [1] F. Nex *et al.*, “UAV in the advent of the twenties: Where we stand and what is next,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 184, pp. 215–242, Feb. 2022, doi: 10.1016/J.ISPRSJPRS.2021.12.006.
- [2] ResearchAndMarkets.com, “Global Drone Industry Report 2021-2025 | Business Wire.” [Online]. Available: <https://www.businesswire.com/news/home/20220112005718/en/Global-Drone-Industry-Report-2021-2025-Growth-of-21.01-Billion-Projected-Rising-Legal-Acceptance-of-Drone-Operations-for-Commercial-Applications---ResearchAndMarkets.com>
- [3] “Military Drones Global Market Report 2022.” <https://www.globenewswire.com/news-release/2022/02/28/2392912/0/en/Military-Drones-Global-Market-Report-2022.html> (accessed Apr. 23, 2022).
- [4] J. Zhao, F. Gao, G. Ding, T. Zhang, W. Jia, and A. Nallanathan, “Integrating communications and control for UAV Systems: Opportunities and challenges,” *IEEE Access*, vol. 6, pp. 67519–67527, 2018, doi: 10.1109/ACCESS.2018.2879637.
- [5] H. T. Do *et al.*, “Formation Control Algorithms for Multiple-UAVs: A Comprehensive Survey,” *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems*, vol. 8, no. 27, pp. 1–13, 2021, doi: 10.4108/EAI.10-6-2021.170230.

- [6] A. Vashisth, R. Singh Batth, and R. Ward, “Existing Path Planning Techniques in Unmanned Aerial Vehicles (UAVs): A Systematic Review,” *Proceedings of 2nd IEEE International Conference on Computational Intelligence and Knowledge Economy, ICCIKE 2021*, pp. 366–372, Mar. 2021, doi: 10.1109/ICCIKE51210.2021.9410787.

- [7] Maj Sean Williams and School of Advanced Military Studies US Army Command and General Staff College, “Swarm Weapons: Demonstrating a Swarm Intelligent Algorithm for Parallel Attack.,” May 2018. <https://apps.dtic.mil/sti/pdfs/AD1071535.pdf> (accessed Apr. 23, 2022).

- [8] “Hierarchical Reinforcement Learning helping Army advance drone swarms.” <https://www.therobotreport.com/hierarchical-reinforcement-learning-army-drone-swarms/> (accessed Dec. 23, 2020).

- [9] “Swarming Drones Application for UAS Unveiled | Unmanned Systems Technology.” <https://www.unmannedsystemstechnology.com/2022/02/swarming-drones-application-for-uas-unveiled-at-umex-2022/> (accessed Apr. 23, 2022).

- [10] N. F. J. S. E. B. JL Deneubourg, *Self-organization in biological systems*. 2003. Accessed: Dec. 23, 2020. [Online]. Available: <https://books.google.com/books?hl=en&lr=&id=BIqCDwAAQBAJ&oi=fnd&pg=PP7&dq=C.+Deneubourg,+F.+Sneyd,+and+T.+Bonabeau,+Selforganisation+in+Biological+Systems.+Princeton,+2001.&ots=A146G-t6gz&sig=yANHhzHPMF-r9gr1GKX9kIGaL0s>

- [11] 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, May 2011, doi: 10.1109/TSP.2011.2107907.
- [12] S. El Ferik and O. R. Thompson, "Biologically Inspired Control Of A Fleet Of Uavs With Threat Evasion Strategy," *Asian Journal of Control*, vol. 18, no. 6, pp. 2283–2300, Nov. 2016, doi: 10.1002/asjc.1324.
- [13] J. Li, S. Y. Tu, and A. H. Sayed, "Honeybee swarming behavior using diffusion adaptation," in *2011 Digital Signal Processing and Signal Processing Education Meeting, DSP/SPE 2011 - Proceedings*, 2011, pp. 249–254. doi: 10.1109/DSP-SPE.2011.5739220.
- [14] Y. H. Chang, C. Tomlin, and K. Hedrick, "Biologically-inspired coordination of multiple UAVs using sliding mode control," in *Proceedings of the American Control Conference*, 2011, pp. 4123–4128. doi: 10.1109/acc.2011.5991252.
- [15] M. Chen, F. Dai, H. Wang, and L. Lei, "DFM: A Distributed Flocking Model for UAV Swarm Networks," *IEEE Access*, vol. 6, pp. 69141–69150, 2018, doi: 10.1109/ACCESS.2018.2880485.
- [16] S. el Ferik, "Behavioral control of UAVs with multi-threat evasion strategy inspired by biological systems," in *2017 14th International Multi-Conference on Systems, Signals and Devices, SSD 2017*, Dec. 2017, vol. 2017-January, pp. 181–186. doi: 10.1109/SSD.2017.8167018.

- [17] S. Y. Tu and A. H. Sayed, “Cooperative prey herding based on diffusion adaptation,” *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pp. 3752–3755, 2011, doi: 10.1109/ICASSP.2011.5947167.
- [18] “Control, Planning, and Learning for Multi-UAV Cooperative Hunting - ProQuest.” <https://www.proquest.com/openview/c95ce0322169f7547567de4c9cb428ee/1?pq-origsite=gscholar&cbl=18750&diss=y> (accessed Apr. 23, 2022).
- [19] “Multi-UAV Path Planning and Guidance for Cooperative Hunting in a Cluttered Environment | TSpace Repository.” <https://tspace.library.utoronto.ca/handle/1807/98529> (accessed Apr. 23, 2022).
- [20] R. Hu, N. Tan, and F. Ni, “A New Scheme for Cooperative Hunting Tasks with Multiple Targets in Dynamic Environments,” pp. 1816–1822, Mar. 2022, doi: 10.1109/ROBIO54168.2021.9739257.
- [21] S. Y. Tu and A. H. Sayed, “Tracking behavior of mobile adaptive networks,” in *Conference Record - Asilomar Conference on Signals, Systems and Computers*, 2010, pp. 698–702. doi: 10.1109/ACSSC.2010.5757652.
- [22] S. Y. Tu and A. H. Sayed, “Cooperative prey herding based on diffusion adaptation,” *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pp. 3752–3755, 2011, doi: 10.1109/ICASSP.2011.5947167.

- [23] P. Schmuck, “Multi-UAV collaborative monocular SLAM,” in *Proceedings - IEEE International Conference on Robotics and Automation*, Jul. 2017, pp. 3863–3870. doi: 10.1109/ICRA.2017.7989445.
- [24] Y. Tian *et al.*, “Search and rescue under the forest canopy using multiple UAVs,” *The International Journal of Robotics Research*, vol. 39, no. 10–11, pp. 1201–1221, Sep. 2020, doi: 10.1177/0278364920929398.
- [25] A. Ashraf, A. Majd, and E. Troubitsyna, “Online path generation and navigation for swarms of UAVs,” *Scientific Programming*, vol. 2020, 2020, doi: 10.1155/2020/8530763.
- [26] X. Fan, W. Sayers, S. Zhang, Z. Han, L. Ren, and H. Chizari, “Review and Classification of Bio-inspired Algorithms and Their Applications,” *Journal of Bionic Engineering* 2020 17:3, vol. 17, no. 3, pp. 611–631, May 2020, doi: 10.1007/S42235-020-0049-9.
- [27] X. Zhang *et al.*, “A Biologically Inspired Network Design Model,” *Scientific Reports* 2015 5:1, vol. 5, no. 1, pp. 1–14, Jun. 2015, doi: 10.1038/srep10794.
- [28] P. Leitão, J. Barbosa, and D. Trentesaux, “Bio-inspired multi-agent systems for reconfigurable manufacturing systems,” *Engineering Applications of Artificial Intelligence*, vol. 25, no. 5, pp. 934–944, Aug. 2012, doi: 10.1016/J.ENGAPPAI.2011.09.025.
- [29] W. Wang, J. Liu, G. Xie, L. Wen, and J. Zhang, “A bio-inspired electrocommunication system for small underwater robots,” *Bioinspiration &*

- Biomimetics*, vol. 12, no. 3, p. 036002, Mar. 2017, doi: 10.1088/1748-3190/AA61C3.
- [30] N. Palmieri, X. S. Yang, F. de Rango, and S. Marano, “Comparison of bio-inspired algorithms applied to the coordination of mobile robots considering the energy consumption,” *Neural Computing and Applications*, vol. 31, no. 1, pp. 263–286, Jan. 2019, doi: 10.1007/S00521-017-2998-4/TABLES/10.
- [31] B. Widrow, J. McCool, M. G. Larimore, C. R. Johnson, and G. Tacconi, “Stationary and Nonstationary Learning Characteristics of the LMS Adaptive Filter,” *Aspects of Signal Processing*, vol. 1, pp. 355–393, 1977, doi: 10.1007/978-94-010-1223-2_23.
- [32] J. B. Foley and F. M. Boland, “A Note on the Convergence Analysis of LMS Adaptive Filters with Gaussian Data,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 7, pp. 1087–1089, 1988, doi: 10.1109/29.1632.
- [33] S. Y. Tu and A. H. Sayed, “Mobile adaptive networks,” *IEEE Journal on Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 649–664, Aug. 2011, doi: 10.1109/JSTSP.2011.2125943.
- [34] J. Chen and A. H. Sayed, “On the learning behavior of adaptive networks - part II: Performance analysis,” *IEEE Transactions on Information Theory*, vol. 61, no. 6, pp. 3518–3548, Jun. 2015, doi: 10.1109/TIT.2015.2427352.
- [35] C. G. Lopes and A. H. Sayed, “Steady-state performance of adaptive diffusion least-mean squares,” *IEEE Workshop on Statistical Signal Processing Proceedings*, pp. 136–140, 2007, doi: 10.1109/SSP.2007.4301234.

- [36] A. H. Sayed and C. G. Lopes, “Distributed recursive least-squares strategies over adaptive networks,” *Conference Record - Asilomar Conference on Signals, Systems and Computers*, pp. 233–237, 2006, doi: 10.1109/ACSSC.2006.356622.
- [37] A. H. Sayed, “Adaptive networks,” *Proceedings of the IEEE*, vol. 102, no. 4, pp. 460–497, 2014, doi: 10.1109/JPROC.2014.2306253.
- [38] A. H. Sayed, S. Y. Tu, J. Chen, X. Zhao, and Z. Towfic, “Diffusion strategies for adaptation and learning over networks: An examination of distributed strategies and network behavior,” *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 155–171, 2013, doi: 10.1109/MSP.2012.2231991.
- [39] A. H. Sayed, “Diffusion Adaptation over Networks,” pp. 323–453, May 2012, doi: 10.48550/arxiv.1205.4220.
- [40] S. Khawatmi, A. M. Zoubir, and A. H. Sayed, “Decentralized Decision-Making Over Multi-Task Networks,” *Signal Processing*, vol. 160, pp. 229–236, Jul. 2019, doi: 10.1016/J.SIGPRO.2019.01.029.
- [41] S. Khawatmi, X. Huang, and A. M. Zoubir, “Distributed decision-making over mobile adaptive networks,” *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pp. 3864–3868, Jun. 2017, doi: 10.1109/ICASSP.2017.7952880.
- [42] V. Erokhin, T. Berzina, A. Smerieri, P. Camorani, S. Erokhina, and M. P. Fontana, “Bio-inspired adaptive networks based on organic memristors,” *Nano*

- Communication Networks*, vol. 1, no. 2, pp. 108–117, Jun. 2010, doi: 10.1016/J.NANCOM.2010.05.002.
- [43] A. H. Sayed, “Adaptive networks,” *Proceedings of the IEEE*, vol. 102, no. 4, pp. 460–497, 2014, doi: 10.1109/JPROC.2014.2306253.
- [44] S. Y. Tu and A. H. Sayed, “Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks,” *IEEE Transactions on Signal Processing*, vol. 60, no. 12, pp. 6217–6234, 2012, doi: 10.1109/TSP.2012.2217338.
- [45] F. S. Cattivelli and A. H. Sayed, “Diffusion LMS strategies for distributed estimation,” *IEEE Transactions on Signal Processing*, vol. 58, no. 3 PART 1, pp. 1035–1048, Mar. 2010, doi: 10.1109/TSP.2009.2033729.
- [46] K. Krishnamoorthy, D. Casbeer, and M. Pachter, “Minimum time UAV pursuit of a moving ground target using partial information,” in *2015 International Conference on Unmanned Aircraft Systems, ICUAS 2015*, Jul. 2015, pp. 204–208. doi: 10.1109/ICUAS.2015.7152292.
- [47] A. Alexopoulos, B. Kirsch, and E. Badreddin, “Realization of pursuit-evasion games with unmanned aerial vehicles,” in *2017 International Conference on Unmanned Aircraft Systems, ICUAS 2017*, Jul. 2017, pp. 797–805. doi: 10.1109/ICUAS.2017.7991416.
- [48] J. las Fargeas, P. Kabamba, and A. Girard, “Cooperative Surveillance and Pursuit Using Unmanned Aerial Vehicles and Unattended Ground Sensors,” *Sensors*, vol. 15, no. 1, pp. 1365–1388, Jan. 2015, doi: 10.3390/s150101365.

- [49] X. Fu, H. Feng, and X. Gao, "UAV mobile ground target pursuit algorithm," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 68, no. 3–4, pp. 359–371, Dec. 2012, doi: 10.1007/s10846-012-9690-9.
- [50] E. Camci and E. Kayacan, "Game of Drones: UAV pursuit-evasion game with type-2 fuzzy logic controllers tuned by reinforcement learning," in *2016 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2016*, Nov. 2016, pp. 618–625. doi: 10.1109/FUZZ-IEEE.2016.7737744.
- [51] H. Agrawal, P. Agrawal, and V. Potdar, "A novel bio-inspired algorithm for hunting in multi robot scenario," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 15, no. 3, pp. 1553–1563, 2019, doi: 10.11591/ijeecs.v15.i3.pp1533-1563.
- [52] O. Hamed and M. Hamlich, "Improvised multi-robot cooperation strategy for hunting a dynamic target," 2021, doi: 10.4108/eai.8-2-2021.168691.
- [53] M. Chen and D. Zhu, "A Novel Cooperative Hunting Algorithm for Inhomogeneous Multiple Autonomous Underwater Vehicles," *IEEE Access*, vol. 6, pp. 7818–7828, Feb. 2018, doi: 10.1109/ACCESS.2018.2801857.
- [54] Y. Song, Y. Li, C. Li, and X. Ma, "Mathematical Modeling and Analysis of Multirobot Cooperative Hunting Behaviors," *Journal of Robotics*, vol. 2015, 2015, doi: 10.1155/2015/184256.

- [55] X. Cao and C. Sun, "A potential field-based PSO approach to multi-robot cooperation for target search and hunting," *At-Automatisierungstechnik*, vol. 65, no. 12, pp. 878–887, Dec. 2017, doi: 10.1515/AUTO-2017-0080.
- [56] H. Shen, N. Li, S. Rojas, and L. Zhang, "Multi-Robot Cooperative Hunting," pp. 349–353, Mar. 2017, doi: 10.1109/CTS.2016.0069.
- [57] Y. Duan, X. Huang, and X. Yu, "Multi-robot dynamic virtual potential point hunting strategy based on FIS," *CGNCC 2016 - 2016 IEEE Chinese Guidance, Navigation and Control Conference*, pp. 332–335, Jan. 2017, doi: 10.1109/CGNCC.2016.7828806.
- [58] J. Ni and S. X. Yang, "Bioinspired neural network for real-time cooperative hunting by multirobots in unknown environments," *IEEE Transactions on Neural Networks*, vol. 22, no. 12 PART 1, pp. 2062–2077, Dec. 2011, doi: 10.1109/TNN.2011.2169808.
- [59] M. Sampei, H. Kiyota, M. Koga, and M. Suzuki, "Necessary and sufficient conditions for transformation of nonholonomic system into time-state control form," *Proceedings of the IEEE Conference on Decision and Control*, vol. 4, pp. 4745–4746, 1996, doi: 10.1109/CDC.1996.577644.
- [60] M. Fliess and F. Messenger, "Methods of Nonlinear Discontinuous Stabilization," *Nonlinear Synthesis*, pp. 112–131, 1991, doi: 10.1007/978-1-4757-2135-5_8.

- [61] S. Alshamali, “A backstepping design approach to a class of mobile robots,” *2017 Asian Control Conference, ASCC 2017*, vol. 2018-January, pp. 1341–1344, Feb. 2018, doi: 10.1109/ASCC.2017.8287366.
- [62] “Adaptive Backstepping Control for a Unicycle-Type Mobile Robot | Journal of Engineering Research.” <https://kuwaitjournals.org/jer/index.php/JER/article/view/8306> (accessed Jul. 21, 2022).
- [63] M. Mera, H. Ríos, and E. A. Martínez, “A sliding-mode based controller for trajectory tracking of perturbed Unicycle Mobile Robots,” *Control Engineering Practice*, vol. 102, p. 104548, Sep. 2020, doi: 10.1016/J.CONENGPRAC.2020.104548.
- [64] A. Bessas, A. Benalia, and F. Boudjema, “Integral Sliding Mode Control for Trajectory Tracking of Wheeled Mobile Robot in Presence of Uncertainties,” *Journal of Control Science and Engineering*, vol. 2016, 2016, doi: 10.1155/2016/7915375.
- [65] Y. Koubaa, M. Boukattaya, and T. Damak, “Adaptive sliding mode control for trajectory tracking of nonholonomic mobile robot with uncertain kinematics and dynamics,” <https://doi.org/10.1080/08839514.2018.1519100>, vol. 32, no. 9–10, pp. 924–938, Nov. 2018, doi: 10.1080/08839514.2018.1519100.
- [66] A. A. Kabanov, S. Stoyanov, and E. N. Kabanova, “Trajectory-tracking control of mobile robot via feedback linearization,” *Advances in Intelligent Systems and*

- Computing*, vol. 680, pp. 32–41, 2018, doi: 10.1007/978-3-319-68324-9_4/COVER/.
- [67] L. Montoya-Villegas, J. Moreno-Valenzuela, and R. Pérez-Alcocer, “A Feedback Linearization-Based Motion Controller for a UWMR with Experimental Evaluations,” *Robotica*, vol. 37, no. 6, pp. 1073–1089, Jun. 2019, doi: 10.1017/S0263574718001443.
- [68] H. Batti, C. ben Jabeur, and H. Seddik, “Fuzzy Logic Controller for Autonomous Mobile Robot Navigation,” *2019 International Conference on Control, Automation and Diagnosis, ICCAD 2019 - Proceedings*, vol. 2019-January, Jul. 2019, doi: 10.1109/ICCAD46983.2019.9037922.
- [69] M. R. H. Al-Dahhan and M. M. Ali, “Path tracking control of a mobile robot using fuzzy logic,” *13th International Multi-Conference on Systems, Signals and Devices, SSD 2016*, pp. 82–88, May 2016, doi: 10.1109/SSD.2016.7473656.
- [70] P. Rochel, H. Ríos, M. Mera, and A. Dzul, “Trajectory tracking for uncertain Unicycle Mobile Robots: A Super-Twisting approach,” *Control Engineering Practice*, vol. 122, p. 105078, May 2022, doi: 10.1016/J.CONENGPRAC.2022.105078.
- [71] S. Yu, Y. Guo, L. Meng, T. Qu, and H. Chen, “MPC for Path Following Problems of Wheeled Mobile Robots,” *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 247–252, Jan. 2018, doi: 10.1016/J.IFACOL.2018.11.021.

- [72] A. Nayak, R. S. Martín De Almagro, L. Colombo, and D. M. de Diego, “Optimal trajectory tracking of nonholonomic mechanical systems: A geometric approach,” *Proceedings of the American Control Conference*, vol. 2019-July, pp. 1924–1929, Jul. 2019, doi: 10.23919/ACC.2019.8814647.
- [73] L. J. Colombo, D. M. de Diego, A. Nayak, and R. T. Sato Martín De Almagro, “Geometric Optimal Trajectory Tracking of Nonholonomic Mechanical Systems,” <https://doi.org/10.1137/19M1243464>, vol. 58, no. 3, pp. 1652–1675, Jun. 2020, doi: 10.1137/19M1243464.
- [74] G. Abdelhakim and H. Abdelouahab, “A New Approach for Controlling a Trajectory Tracking Using Intelligent Methods,” *Journal of Electrical Engineering & Technology 2019 14:3*, vol. 14, no. 3, pp. 1347–1356, Mar. 2019, doi: 10.1007/S42835-019-00112-1.
- [75] M. Boukens and A. Boukabou, “Design of an intelligent optimal neural network-based tracking controller for nonholonomic mobile robot systems,” *Neurocomputing*, vol. 226, pp. 46–57, Feb. 2017, doi: 10.1016/J.NEUCOM.2016.11.029.
- [76] H. Yuan, “University of Science & Technology of China,” 2002.
- [77] S. Y. Tu and A. H. Sayed, “Mobile adaptive networks,” *IEEE Journal on Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 649–664, Aug. 2011, doi: 10.1109/JSTSP.2011.2125943.

- [78] A. H. Sayed, “Adaptive networks,” *Proceedings of the IEEE*, vol. 102, no. 4, pp. 460–497, 2014, doi: 10.1109/JPROC.2014.2306253.
- [79] S. Guan, C. Meng, and B. Biswal, “Diffusion-Probabilistic Least Mean Square Algorithm,” *Circuits, Systems, and Signal Processing*, vol. 40, no. 3, pp. 1295–1313, Mar. 2021, doi: 10.1007/S00034-020-01518-3/FIGURES/8.
- [80] “[PDF] Line and circle formation of distributed autonomous mobile robots with limited sensor range | Semantic Scholar.” <https://www.semanticscholar.org/paper/Line-and-circle-formation-of-distributed-autonomous-Albayrak/26c9da86ec75eb0aba69fcc4bb1d2dc103eda41c> (accessed May 04, 2022).
- [81] H. Yuan, “Control Of Nonholonomic Systems,” *Electronic Theses and Dissertations*, Jan. 2009, Accessed: May 04, 2022. [Online]. Available: <https://stars.library.ucf.edu/etd/4017>
- [82] N. Takahashi, I. Yamada, and A. H. Sayed, “Diffusion least-mean squares with adaptive combiners: Formulation and performance analysis,” *IEEE Transactions on Signal Processing*, vol. 58, no. 9, pp. 4795–4810, Sep. 2010, doi: 10.1109/TSP.2010.2051429.
- [83] F. S. Cattivelli and A. H. Sayed, “Diffusion LMS strategies for distributed estimation,” *IEEE Transactions on Signal Processing*, vol. 58, no. 3 PART 1, pp. 1035–1048, Mar. 2010, doi: 10.1109/TSP.2009.2033729.

- [84] S. Y. Tu and A. H. Sayed, “Mobile adaptive networks with self-organization abilities,” *Proceedings of the 2010 7th International Symposium on Wireless Communication Systems, ISWCS'10*, pp. 379–383, 2010, doi: 10.1109/ISWCS.2010.5624545.
- [85] K. J. Benoit-Bird and W. W. L. Au, “Cooperative prey herding by the pelagic dolphin, *Stenella longirostris*,” *J Acoust Soc Am*, vol. 125, no. 1, p. 125, Jan. 2009, doi: 10.1121/1.2967480.
- [86] Y. Kanayama, A. Nilipour, and C. A. Lehm, “LOCOMOTION CONTROL METHOD FOR AUTONOMOUS VEHICLES.,” pp. 1315–1317, 1988, doi: 10.1109/ROBOT.1988.12248.
- [87] A. Tepljakov *et al.*, “Towards Industrialization of FOPID Controllers: A Survey on Milestones of Fractional-Order Control and Pathways for Future Developments,” *IEEE Access*, vol. 9, pp. 21016–21042, 2021, doi: 10.1109/ACCESS.2021.3055117.
- [88] M. N. Ab Wahab, S. Nefti-Meziani, and A. Atyabi, “A Comprehensive Review of Swarm Optimization Algorithms,” *PLOS ONE*, vol. 10, no. 5, p. e0122827, May 2015, doi: 10.1371/JOURNAL.PONE.0122827.
- [89] F. D. Wihartiko, H. Wijayanti, and F. Virgantari, “Performance comparison of genetic algorithms and particle swarm optimization for model integer programming bus timetabling problem”, doi: 10.1088/1757-899X/332/1/012020.

- [90] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, pp. 1942–1948, doi: 10.1109/ICNN.1995.488968.
- [91] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm Intelligence 2007 1:1*, vol. 1, no. 1, pp. 33–57, Aug. 2007, doi: 10.1007/S11721-007-0002-0.

CURRICULUM VITAE

Name : Md Muzakkir Quamar
Nationality : Indian
Date of Birth : 25-Nov_1993
Email : muzaik.281@gmail.com
Telephone: : +91-7838586923
Address: : Patna, Bihar, India, 800004.

Education

- **MS in Systems and Controls Engineering** (2022)
 - KING FAHD UNIVERSITY OF PETROLEUM AND MINERALS, KSA
 - GPA: 3.57/4.0.
- **BE in Electronics and Communication Engineering**, (2010-2014)
 - MAGADH UNIVERSITY, Gaya, India
 - Overall Aggregate 78% (Distinction)

WORK EXPERIENCE

- **Lab Instructor** (Mechatronics CISE 412), KFUPM, Term 211.
- **Research Assistant** (part time) at Research Institute, KFUPM, 2021-22.
- **Service Delivery Manager**, Orange Business services, 2018.
- **Analyst- Network Infrastructure Ordering**, Orange Business services, 2016-2018.
- **Analyst- Business Operation**, SIS Prosegur, 2014 – 2016.

ACADEMIC ACCOMPLISHMENTS

- Presented a Research Paper “**Instrumental Variable System Identification for Time-Delayed system with Non-Integer Time-Delay**” in IEEE Conference,” International Multi-Conference on Systems, Signals & Devices (SSD),2020.
- Presented a Research paper on “**Comparative analysis of Identification of Turbojet engine using Parametric, Non-Parametric and Artificial intelligent techniques**” at 10th Annual KFUPM Students Scientific Forum.
- Proposed a Research Model on “**Automation of Indian Railway**” in Inter-College Technical Fest at Maulana Azad College of Engineering and Technology (MACET) in 2011 and secured first position.

RESEARCH-BASED ACADEMIC PROJECTS AT KFUPM

- Term project on “**LQR Design of a Two-link Pendulum**” during coursework in Optimal Control.
- Term project on “**Efficient Controller for controlling a Non-Linear System (CSTR)**” during coursework in Linear Control.
- Term project on “**NB-IOT Based Integrated Framework for monitoring Pipeline Condition in Oil and Gas Industry**” during coursework in Distributed Computer Control and Field Buses.
- Term project on “**Design and control of a Non-Linear Inverted Pendulum Cart System**” during coursework in Non-Linear Control.
- Term project on “**Handwritten Number Recognition using Neural Networks**” during coursework in Neural Network Theory and Applications.