BIOLOGICAL INSPIRED ADAPTIVE NETWORK CONTROLLER OF MULTIPLE NONHOLONOMIC UNMANNED AERIAL VEHICLE

BY

MOHAMMED SALEM SALEH MAHDAMI

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

SYSTEMS AND CONTROL ENGINEERING

APRIL 2019

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN- 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **MOHAMMED SALEM SALEH MAHDAMI** 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 fulfillment of the requirements for the degree of **MASTER OF SCINCE IN SYSTEMS AND CONTROL ENGINEERING**

Dr. Samir Al-Amer
(Advisor)

Dr. Hesham Al-Fares
Department Chairman

Dr. Sami El Ferik
(Member)

Dr. Salam A. Zummo
Dean of Graduate Studies

DEANSHIP OF GRADUATE STUDIES

Dr. Abdul-Wahid Al-Saif
(Member)

24/9/19

Date

I dedicate this work to my parents, my brother, my wife, and my lovely son Salem

ACKNOWLEDGMENTS

I would like to say many thanks to my thesis advisor, Dr. Samir Al-Amer for his guidance, his help, his patience and his encouragement.

Also, I would like to express my appreciation to my thesis committee members Dr. Sami El Ferik and Dr. Abdul-Wahd A. Al-Saif for their directions and cooperation.

Furthermore, I appreciate Hadhramout Establishment for Human Development for supporting me with a scholarship.

To my colleagues and friends, thank you!

Finally, I would like to thank the entire Systems Engineering Department for their direct and indirect support all through my time at KFUPM, I am truly grateful.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

UAV	Unmanned Aerial Vehicle
GAO	Government Accountability Office
VTOL	Vertical Take-Off and Landing
DOF	Degree OF Freedom
2D,3D	Two-Dimension, Three-Dimension
LMI	Linear Matrix Inequality
AUV	Autonomous Underwater Vehicle
ATC	Adapt-Then-Combine
CTA	Combine-Then-Adapt
APA	Affine Project Algorithm
MSD	Mean-Square Deviation
IMU	Initial Measurement
PID	Proportional Integral Derivative
GPS	Global Position System

ABSTRACT

Full Name : Mohammed Salem Saleh Mahdami

Thesis Title : Biological Inspired Adaptive Network Controller of Multiple Nonholonomic Unmanned Aerial Vehicle

Major Field : Systems and Control Engineering

Date of Degree : April,2019

In this thesis, we propose algorithms for the navigation of fleet of UAVs in order to evade a chasing attacker during their journey to the target in three-dimensional space. We design these algorithms to transform strategies of path planning depending on the conditions of prevailing movable environment. The relationship between a school of fish and attacker was the inspiration for the implementation of the nonholonomic UAVs in an adaptive network. The navigation algorithms are consequently combined with each other for a group of nonholonomic UAV, so the UAVs display the biological behavior of fish, when they move to the food or when they evade the attacker. To put this in context, the goal of this thesis is to use the biological inspired algorithm to realize the navigation to the target and the evasion from the attacker. The simulations show the performance of this approach for two cases; without attacker and with attacker.

ملخص الرسالة

الاسم الكامل: محمد سالم صالح مهدمي

عنوان الرسالة: المتحكم الشبكي المتكيف مستوحى من النظم البيولوجية للتحكم في مركبات طائرة متعددة بدون طيار

التخصص: هندسة نظم وتحكم

تاريخ الدرجة العلمية: ابريل 2019

في هذه الأطروحة ، نقترح خوارزميات للتنقل لأسطول طائرات بدون طيار من أجل التهرب من العدو المهاجم خلال رحلتهم إلى الهدف في ثلاثة أبعاد .نقوم بتصميم هذه الخوارزميات لتحويل استراتيجيات تخطيط المسار اعتمادًا على ظروف البيئة المنقولة السائدة .كانت العلاقة بين مجموعة من الأسماك يتحركون مع بعض والحيوانات المفترسة مصدر إلهام لتنفيذ الطائرات بدون طيار في شبكة تكيفية وبالتالي من خلال تقليد تصرف الاسماك مع القرش تم دمج خوارزميات التنقل مع بعضها في مجموعة من الطائرات بدون طيار ، لذلك تعرض الطائرات بدون طيار السلوكيات البيولوجية للاسماك ، عندما تنتقل إلى الغذاء أو عندما تتهرب من المفترس .هذه الطريقة تحاكي في مجال الأسلحة الجوية .لوضع هذا في السياق ، فإن الهدف من هذه الأطروحة هو استخدام الخوارزمية المستوحاة من البيولوجيا لتحقيق الملاحة إلى الهدف والتهرب من المهاجم. .تُظهر المحاكاة أداء هذا النهج في حالة وجود المهاجم او عدم وجود المهاجم.

CHAPTER 1

INTRODUCTION

1.1 Introduction

These days, Unmanned Aerial Vehicles (UAVs) are becoming a widespread for many applications. Essential use of UAVs is in the military operations. Military organizations in many countries plan to use Unmanned Aerial Vehicles for some activities. The United States Government Accountability Office (GAO) in 2012, did a research and it estimated that the number of countries owning UAVs had doubled in the previous seven years [1]. Another application of using the UAVs technology is in the commercial industry, as the big tech giants, an example of which is Amazon plans to use "Amazon Prime Air" [2], to revolutionize their product chain system for home deliveries by using UAV. Usage of UAVs makes it possible to gather information in dangerous environments without posing risk to flight crews and humans. For its use in the above-mentioned applications, many different structures, shapes and sizes of vehicles were developed and utilized. The most important feature that the vehicle is supposed to have is Vertical Take-Off and Landing (VTOL). This VTOL capability makes it suitable for applications where space is a problem and indeed the missions and tasks, now-a-days, can make use of this capability, so this makes it superior to other non-VTOL vehicles. This capability is present in only a few vehicles like modern apache helicopters, unmanned helicopters and quad-rotor, octorotors, tilt-rotor, co-axial rotor vehicles. The main disadvantage of VTOL helicopters are the increased complexity and maintenance that comes with the intricate linkages, cyclic control of the main rotor blade pitch, collective control of the main blade pitch, and anti-torque control of the pitch of the tail rotor blades. Also, weighing the benefits and the drawbacks of all the vehicles, the choice of a UAV narrows down to the quad-rotor. The quad-rotor is considered an effective alternative to the high cost and complexity of standard rotorcraft. The quad-rotor UAV became popular in the last few decades, yet a lot of studies on the quad-rotor UAV is being carried out since the beginning of the last century. Etienne Oehmichen first tried six designs for quad-rotor in the 1920s. Then, the second design had four rotors and eight propellers, all driven by a single engine. The rotorcraft exhibited a considerable degree of stability and controllability at that time. A thousand test flights were performed in the middle of 1920s and it was able to remain flying for several minutes in 1923. In 1922, George de Bothezat built a helicopter with four rotors, under the sponsorship of the United States Army. The helicopter had four rotors mounted at the end of X-shape of 18 meters beams and weighed 1700 kg. About 100 flights were executed in 1923 and five meters was the highest altitude it ever reached. Another design for the quad-rotor was Convert Wings Model A quad-rotor in 1956. Employing four rotors to create differential thrust, the craft can hover and move without the complex system of linkages and blade elements present in standard single rotor vehicles. Since then there have been designs and experiments to reduce the size, cost, weight, inclusion of aerodynamic parts and many other factors. Now the size has significantly reduced, and we can even see minute quad-rotors, varying from large to palm sizes. The quadrotor is classified as an underactuated system. This is due to the fact that only four actuators (rotors) are used to control all six degrees of freedom (DOF). The four actuators directly impact z-axis translation (altitude) and rotation about each of the three principal axes. The other two DOF are translation along the x and y-axis. These two remaining DOF are coupled, meaning they depend directly on the overall orientation of the vehicle (the other four DOF). Two more benefits for the quadrotor which are swift maneuverability and increased payload. The increasing demand of Unmanned Aerial Vehicle, for application of military, makes the control design of this UAVs an active area of research.

Unmanned Aerial Vehicles are belonging to the nonholonomic systems class. The main problem in this class of models is that it cannot be stabilized by smooth static state feedback controller. The reason for this problem is that, models can usually have structure with only two inputs and three states to control the system, so the Brockett's Condition is not satisfied [3] [4]. To overcome this problem, a number of control strategies were developed [5] during the past years. These include continuous feedback laws [4] [6] [7] and discontinuous feedback laws [8] [9].

One use of biologically inspired algorithms is based on the collective behavior of animal groups to simulate real world scenarios. Several engineering applications are used in a way similar to biologically behavior. Whittlesey and Dabiri in [10] harnessed the movement flow model of a school of fish to investigate the output power efficiency of vertical axis wind turbines as opposed to the commonly used horizontal axis wind turbines. This behavior, when applied to the real-world wind turbine application showed an optimization of the output power. Several other biological behaviors can be and are being imitated for other applications. Our work highlights one of these applications. We propose the application of the foraging and evading behaviors of a school of fish to a group of mobile UAVs. The idea is easily applicable to rescue missions heading for a target or even attack strategies. The self-organizing formations of a group of birds in flight, or the hunting techniques used by carnivores in the wild, are some of the behaviors actively being researched.

1.2 Problem Statement

From what we found from the former studies, we noted that many researches used single UAV and others worked on using multiple UAV. Some of the latter studies considered that many UAV approach a specified target in 2-diemsion. While the others during their movement to the destination are exposed to attack and they try to avoid it using the same style of school of fish. As a result, our aim is to extend the study of evading the attack using the same principle in 3-Dimensions. In order to solve this problem, we will implement the following objectives.

1.3 Objectives

In this thesis we try to a achieve the following objectives.

1. Design a 3D navigation algorithm for nonholonomic UAVs group in flight, by considering the conditions of the environmental and constraints of aerodynamic.
2. Design a group of nonholonomic UAV models' adaptive network to mimic the behavior of a school of fish for navigation and avoiding the attacker.
3. Design tracking controller for nonholonomic UAV using a series of straight-line paths to avoid an approaching attacker.
4. Design navigation and control of the UAVs to ensure collision avoidance.

1.4 Organization

This thesis is organized as follows: Chapter 2, describes nonholonomic system in 2-dimension. In addition, we describe some techniques of nonlinear controller that are used for nonholonomic systems. Also, the literature review related to this research and some previous work that have been done in nonholonomic systems are discussed in. In Chapter 3, we discuss a single UAV, or a quadrotor and we elaborate an equation of this model. Also, the technique of the controllers, which are used in this algorithm by using two controllers, is discussed in this chapter. Moving to the next chapter, the nonholonomic system algorithm in 3D is elaborated in Chapter 4, with the adaptive network that is used in this algorithm, and we explain how to make the nodes mimic the school of fish for foraging and avoided the enemy when they attack. Chapter 5 shows the simulation results, and discussion of these results. Finally, we end up by the conclusion of our research and suggestion of some future work.

CHAPTER 2

LITERATURE REVIEW

2.1 Nonholonomic System

Nonholonomic systems include many applications such as quadrotors, mobile robot with two-driving wheel, car-like robots, the robot hopping, and autonomous underwater vehicle. Fortunately, the structures of these models are quite similar. These models are often represented in either dynamic or kinematic form. The general kinematic form is described as [5],

$$\dot{x} = g_1(x)u_1 + \cdots + g_m(x)u_m \tag{2.1}$$

where $2 \leq m$, x is the state vector, defined in $\mathbb{R}^n$, u_i, $i = 1, \cdots, m$, are control inputs, and $g_i, i = 1, \cdots, m$, are given vectors. By expanding equation (2.1) to the representative form for the UAV [11],

$$\begin{aligned} \dot{x}_1 &= u_1 \cos\theta \\ \dot{x}_2 &= u_1 \sin\theta \\ \dot{\theta} &= u_2 \end{aligned} \tag{2.2}$$

The system can be described by the D'Alembert-Lagrange form which is a general dynamic model [5] given by,

$$M(x)\ddot{x} + f(x, \dot{x}) = C\lambda + B(x)\tau$$
$$J^T(x)\dot{x} = 0 \qquad (2.3)$$

where $x = (x_1, \cdots, x_n)$ is an n-vector of coordinate, $M(x)$ is the positive definite symmetric inertia matrix, $J(x)$ a full rank $n \times (n-m)$ matrix, λ an $(n-m)$ Lagrange multiplier vector, $B(x)\tau$ a vector forces applied to the system, $B(x)$ is an $n \times p$ matrix, and τ is a control p-vector.

Several control techniques have been used in the literature. Some of them are presented in the following sections.

2.1.1 Robust Control

The chained form of a class of perturbed nonholonomic systems is discussed by Yang et al. [12]. It consists of two-stage stabilization process: firstly, a subsystem is established by state feedback and adaption law; then, the residual subsystem is changed by using a coordinate system. To achieve stabilization, A Linear Matrix Inequality (LMI) technique is applied using a robust sliding surface that ensures robustness. The local robustness case for nonholonomic systems is discussed in [8]. It also proposed design of discontinuous control to achieve global asymptotic stability under external minor disturbance.

2.1.2 Backstepping Controller

The use of a Lyapunov-based control is discussed by Yuan in [13] to find an exponential control to make sure the stabilization of an autonomous underwater vehicle (AUV), another nonholonomic system. The authors, in [14], discussed, the use in an unmanned terrain, a backstepping control technique for a spherical robot. This technique presented the desired trajectory for asymptotic tracking convergence and was confirmed using simulations.

2.1.3 Sliding Mode Controller

To achieve tracking goals for nonholonomic systems, another control technique the researchers have considered is sliding mode control. A non-smooth feedback by using sliding mode control is designed in [9] to track the favorite state function of a nonholonomic system. The dynamic and kinematic nonholonomic models are combined in Chwa [15], to design two controllers using sliding mode control. He was able to remove constraints on the reference velocities by using the polar coordinate form of the kinematic model. This control technique carried out tracking except at an arbitrary small area near the origin. This technique also considered the chattering of the control signal.

2.1.4 Time-State Control form

Some nonholonomic systems cannot be represented by chained form, so the Time-State Control presents a way to change nonholonomic systems that cannot be formulated in chained form. This was presented in [7]. Although this form was able to overcome the controllability problem associated with nonholonomic systems; but it requires input switching which may disrupt the system stability in some cases. Sampei et al. [4] discussed the sufficient and necessary conditions for restructuring a system to the time-state way.

2.1.5 Transformation and Feedback Linearization

Many researchers joined a number of techniques to handle nonholonomic systems. So some authors used some state transformation methods first, then applied another control technique, while others used feedback linearization after applied techniques. While some of them applied feedback linearization and state transformation at the same time.

On one hand, Matsuno et al. [16] exactly elaborated an algorithm that changes a class of three states and two inputs of nonholonomic systems to their chained forms. This transformation was proved to be very useful for control analysis of this class of systems. While on the other hand Astolfi [17] designed a discontinuous and bounded state feedback control law which ensures exponential convergence. He also took in his account note of quantity errors in his analysis. For more elaborations of these subjects, see [18] and [19]. Luca et al. in [18], provided state changes which restructure the kinematic models into a chained form. By using the chained form, alternative feedback techniques were useful to track random trajectories or stabilize the nonholonomic system. The authors used the control of the nonholonomic system firstly by approximation of feedback linearization. While this technique obtained local asymptotic stability of the chained form system, but this region of stability was expected to be extremely large. The disadvantage though is that the passing response of the system may end up in an unsatisfactory method.

Luca [18] and Tian [6] used exact feedback linearization. The possibility of using input-output feedback linearization is considered in [18]. that might use full-state feedback linearization or compromise some states. In the first case, to overcome the problem of matrix singularity, the system output is redefined, so that a static feedback can be designed for the system. A second case to solve the problem of matrix singularity was to include an integrator and add an additional input.

2.1.6 Non-Time Based Tracking Control

This technique avoids the use of time for a nonholonomic system in defining the reference trajectory. To track a desired trajectory, Tau et al. [20] projected an event-based controller by using a non-time reference. But we need to change from the time-based controller to a non-time-based controller. A state to reference projection is used to do this transformation. To solve the problem of discontinuous control common with nonholonomic systems, a biologically inspired additive model and the traditional non-time-based controller are combined with each other by Hu et al. [21]. This addition resolves the issue and eliminates the tracking error that exists in [20].

2.1.7 Tracking Control

Several authors have treated the tracking problem for nonholonomic systems by using various control strategies to explore it, some of these strategies have been elaborated before. The tracking challenge is converted to a two-system stabilization issue by Tian and Cao [22]. They used a cascade and a transformation technique.

To stabilize the system, they designed a Linear Matrix Inequality (LMI) based algorithm, to track the given reference. In 2011, the work in [22] was improved by Cao [23]. The update allowed for the exponential convergence to a desired trajectory. The authors in [24] did something new, they invented an algorithm that makes a nonholonomic constrained rigid body track a straight line without twirling. They differentiate the trajectory's curvature as a linear combination of the vehicle's orientation error, position error, and current trajectory curvature, this derivation or this function is called the steering function. After that they used Lyapunov's theory to improve the stabilization of the system. The authors in [25] give a solution for visual servo control and the tracking problem for a nonholonomic wheeled robot by using a practical approach. By Lyapunov analysis, they used videos for desired paths and showed convergence. Also, the authors in [26] proposed the combined visual problem and tracking; however, in their analysis, they included parameter uncertainties. In [27], Keighobadi et al. solved the problem of achieving complete tracking of a wheeled robot by using a torque of nonlinear controller. They proposed two fuzzy controllers to solve the problem.

2.1.8 Other Control Methods

Other techniques have been used to control nonholonomic systems that were not mentioned above. One of them is presented by Hespanha et al. [28]. That discussed a controller for nonholonomic systems with doubt in the parameters of model. They also proposed using supervisory control and a hybrid feedback controller. They used a state transformation to overcome the problem of singularity common of nonholonomic models. Campion et al. [29] describes the using of Euler-Lagrange equations for nonholonomic system. They presented the derivation in a stepwise manner. A smooth state feedback law using this model was devised, it ensures global marginal stability. The authors in [30] use the kinematic form to reach the goals of control. The authors used gradient vector of a Lyapunov function and a tensor to design the controller. Exponential convergence is obtained by using the designed controller. An experimental work was done, validating the kinematic model of a nonholonomic "spherical" in [31], including the nonholonomic constraints and dynamic model. All the control techniques that were mentioned above are for the general form of nonholonomic models in addition they are appropriate to planar unmanned aerial vehicles, a subset of this class of models.

2.2 The Network of Biological Inspired Robots

The aggregate conduct of animals is interesting [32]. These manners such as a swarm of bees, school of fish, and a flock of birds continue to interest philosophers and researchers. In addition, many scientists have tried to understand and imitate these manners. The author in [33] indicated that the scientists are just starting to know the meaning of the link between group-level and single-level manners of animals, and what part of these connection play in deciding the adaptive response. Likewise, the authors in [34] showed the developing enthusiasm of natural models among engineers. They defined restriction plans using the confinement practices that bottlenose dolphins adjust to catch their preys. They recommended that the bio-inspired plans took into account its usage in mechanical frameworks. The system of bio-inspired, in that time, faced some difficulties with adapting to any disturbance or changes in the process of manufacturing as Park et.al [35] noted. This is because these systems are pre-programmed, and it cannot independently make similar changes. So, it cannot resist these changes and as a result, it will fail. To overcome this problem the author in [35] proposed an independent control system in agreement with these changes in a manufacturing workshop.

When anyone studies the biological behavior of the animal as a group, when it wants to search for food or other supplies, they should consider keeping the safe distance from a danger or attacker. This idea is also investigated in [36]. The behavior model which is used in this paper is mostly theoretical. The author of this research has some experimental data and used it for research. The model of statistical fitting is produced then that study was done to the model which was obtained. The system of flocking control with many agents were developed in [37]. The controllers are designed to pursue both moving and nonmoving targets by recognizing the leader of the team. Then the uniqueness and existence of a solution were proved in this paper. This research is extended in [38]. The authors made the target fixed and they used decentralize flocking controller.

On the other hand, the idea of followers and leaders is a traditional idea. [39] suggested another concept in order to give self-moving to each biological driven agent in which there is no need for a leader of the flock. For formulating distributed flocking algorithm, Olfati-Saber [40] suggested a theoretical structure. He used systematically derived objective functions (or cooperative potentials) for the flock members.

2.2.1 Adaptive Network

Adaptive networks are used as the cornerstone of many biologically inspired networks [41] [42] [43] [44]. A survey of the study of adaptive networks and the improvement in the field was written by Sayed Ali in [45]. He noted the progress related to adaptation, optimization, and learning over networks. He also wrote about the different distribution strategies which allow networked agents to interact locally, learn and adapt to track changes in the data streams they receive and in the models.

2.2.2 Distribution Strategies

Sayed [46] provides a summary of various distribution strategies and the best strategy to employ for many applications. The summary furnishes details on diffusion strategies for agent learning and adaptation over networks, showing that they imitate some biological behaviors. The topologies linking agents in the network can either be dynamic or static. The author goes further to describe the network model, similar to the models used in previous papers. Two classifications of distributed strategies are given: noncooperative adaptive strategy and diffusion strategy (cooperative). He concluded, using mean square deviation performance analysis, that diffusion strategies outperform non-cooperative strategies. In Sayed's work with Lopes [47], an adaptive diffusion least mean square algorithm was framed to ensure collaboration between each node. Each node calculates the local estimates for necessary information and shares it with its neighbors. The algorithm is thus both conjunctive and distributed. [48] developed the work of Lopes and Sayed's [47] by adding data-normalized algorithms and a dynamic topology. To advance the robustness of diffusion networks in the existence of noisy and disturbances sensor measurements, adaptive combiners are added to the networks as in [49]. The authors showed that including the adaptive combiners with the diffusion least mean square algorithm makes the algorithm perform better. A similar case of disturbance was analyzed by Cattivelli et al. [50]. Here, diffusion-based adaptive solutions of the least mean square type were proposed and was shown to have improved performance.

2.3 Related Research

There are many researches done in this area. The diffusion adaptation algorithm has been applied in [43]. This algorithm is Adapt-Then-Combine (ATC). It explained how the fish cooperative pursues a source of food meanwhile stay away from attack from the attacker. The algorithm, used in this paper, is implemented in a real-time and distributed method. Each node just communicated with its close neighbors.

The fixed step size of the (ATC) diffusion algorithm is used in the update for vectors of location and velocity. So, in [51] some improvement in (ATC) were done. They suggested that when there is a long distance between the nodes and the target, a large number of iterations are required to reach the target as fast as possible. To do this improvement they suggested two modifications to the ATC. Firstly, it adjusted the distance based on variable step size at diffusion algorithm to update the location and velocity. Secondly, it reduced the number of the communication by taking the best node at each iteration to use a selective cooperation.

The authors in [52] focused on the number of neighbor nodes. It studied a mobile diffusion adaptive network with the cooperation of selectivity. By this algorithm, each node selected a subset of its neighbors so that its steady-state performance be as close as possible to the traditional mobile diffusion network. The authors used Affine Project Algorithm (APA) as the learning rules at the nodes. They concluded that the result by using APA is the same result by using Mean-Square Deviation (MSD) but with lower communication per iteration per node.

The mobile adaptive network diffusion algorithm, can transfer reasonably to a target with an unknown target, is developed in [54]. The authors in [53] increased the number of attackers. These attackers cooperate with each other to do two missions; firstly, surrounding a school of fish, then trapping the school while attack. This paper used three-dimension.

The authors in [41] investigated cognitive abilities and self-organization of adaptive network if the individual agents can move to pursuit a target. It developed diffusion strategies for the mobile network which involved two steps; firstly, for estimating the location of the target. Secondly, for tracking the center of the network mass. The strategies also include location and velocity control mechanism to dedicate the node motion. The result of this paper showed that this way can bypass regions of bad signal or obstacles which was described in this paper by high noise region.

A model for food foraging was developed in [55]. This paper also explained how a school of fish move with each other, as a group, if each fish were to use a distributed strategy which is diffusion adaptation. This paper also compares maneuvers of fish school with and without diffusion. The result of this paper indicated that without diffusion algorithm, the fish foraging does not result in schooling.

The number of targets is increased in [56] in which the network nodes from distinct cluster to study following more than one targets, all while moving in a cohesive collision-free method. It assumed each node is initially assigned a random target at the same time and node did not know how many targets in their location.

A multi-UAV cooperative hunting algorithm by using the neural network is investigated in [30] for the three-dimensional underwater environment. It added the obstacle in this environment where the hunting UAV goes toward the hunting point directly and avoiding collision with the obstacle and with each other. Then they surround and catch the mobile target.

The backstepping tracking controller is tracked in [31] to define path planning trajectories. It implemented the number of nonholonomic UAVs in an adaptive network exactly inspired by the relationship between the attacker and school of fish.

The authors in [32] used limit cycle algorithm by using a neural oscillator with phase difference for a fleet of robot's navigation and coordination to intercept a target in the long distance. The new thing in this paper, in addition to using obstacle, is that it uses virtual targets. The robots, instead of going to the actual target, they go to the virtual target. While the actual target moves to the virtual target.

CHAPTER 3

MODELING OF QUADROTOR UNMANNED AERIAL VEHICLE

A number of researches have discussed the general dynamic model of a quadrotor. One of them was by H. Voos [60]. We consider a fixed frame and an initial frame in which the center of the mass of the quadrotor is origin as shown in Figure 3.1.

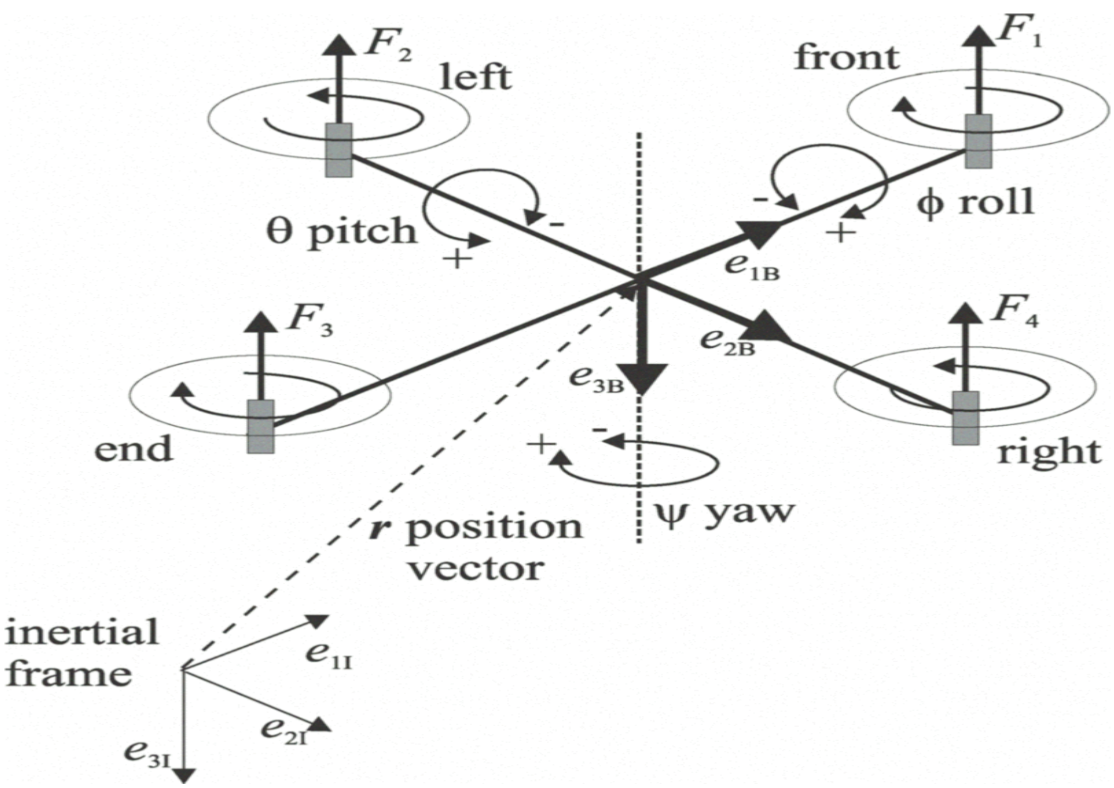

Figure 3.1 Configuration Initial and Body Fixed Frame of the Quadrotor [60]

From Figure 3.1 we note that there are 4 rotors and three angles which are orientation angles. These Euler angles are yaw angle ψ, pitch angle θ and roll angle ϕ that together form the vector $\Omega^T = (\phi, \theta, \psi)$. Furthermore, the vector of position of the vehicle in the inertial frame is $\boldsymbol{r}^T = (x, y, z)$.

3.1 Rotation Matrices

Rotation (also called elemental rotation) is a rotation about one of the axes of a coordinate system. The following three basic rotation matrices rotate vectors by an angle about the x, y, or z-axis, in three dimensions, using the right-hand rule which codifies their alternating signs. (The same matrices can also represent a clockwise rotation of the axes).

A vector $r = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$ when subjected to a rotation of angle ψ with respect to x-axis will be transformed to a new position vector

$$r_{new} = R_x(\psi)\, r \tag{3.1}$$

where

$$R_x(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{bmatrix} \tag{3.2}$$

$R_x(\psi)$ is known as rotation matrix. Similarly, rotation angle θ about y-axis give rise to the rotation matrix

$$R_y(\theta) = \begin{bmatrix} cos\,\theta & 0 & sin\,\theta \\ 0 & 1 & 0 \\ -\,sin\,\theta & 0 & cos\,\theta \end{bmatrix} \tag{3.3}$$

And rotation by angle ϕ with respect to z-axis can be obtained using the rotation matrix

$$R_z(\phi) = \begin{bmatrix} cos\,\phi & -sin\,\phi & 0 \\ sin\,\phi & cos\,\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

When a vector r is subjected to sequence of rotation ψ with respect to x-axis, θ with respect to y-axis and ϕ with respect to the z-axis.

The new position can be obtained

$$r_{new} = R_x\, R_y\, R_z\, r = R\, r$$

where R is given by

$$R = \begin{bmatrix} C_\psi C_\theta & C_\psi S_\theta S_\phi - S_\psi C_\phi & C_\psi S_\theta C_\phi + S_\psi S_\phi \\ S_\psi C_\theta & S_\psi S_\theta S_\phi + C_\psi C_\phi & S_\psi S_\theta C_\phi - C_\psi S_\phi \\ -S_\theta & C_\theta S_\phi & C_\theta C_\phi \end{bmatrix} \tag{3.5}$$

where $S_{(.)}$ and $C_{(.)}$ denotes $\sin(.)$ and $\cos(.)$ respectively.

3.2 Quadrotor Dynamic Model

The quadrotor has four motors Let the rotor speed be ω_i, $i = 1, 2, 3, 4$. The thrust force generated by the rotor $F_i = b\, \omega_i^2$ where ω_i the rotor speed and b is thrust factor.

$$\ddot{\boldsymbol{r}} = g \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} - R\, \frac{b}{m} \sum_{i=4}^{4} \omega_i^2 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{3.6}$$

Let the inertial matrix I be a diagonal matrix with I_x, I_y, and I_z on the main diagonal, J_R is the rotor inertia, the torque vector, denoted by M, is applied to the body of vehicle, and gyroscopic torques denoted by the vector M_G. So, we obtain the differential equation

$$I\ddot{\Omega} = -\left(\dot{\Omega} \times I\ddot{\Omega}\right) - M_G + \mathrm{M} \tag{3.7}$$

From the Figure 3.1

$$M = \begin{bmatrix} Lb(\omega_2^2 - \omega_4^2) \\ Lb(\omega_1^2 - \omega_3^2) \\ d(\omega_1^2 + \omega_3^2 - \omega_2^2 - \omega_4^2) \end{bmatrix} \tag{3.8}$$

where d is the drag factor, L is the length of the lever. The gyroscopic torque is

$$M_G = I_R \left(\dot{\Omega} \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right) . (\omega_1 - \omega_2 + \omega_3 - \omega_4) \tag{3.9}$$

The inputs here are the rotational velocities of the rotors ω_i which is the real vehicle input variables. A new artificial inputs variable are introduced and they are defined as,

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} b(\omega_1^2 + \omega_3^2 + \omega_2^2 + \omega_4^2) \\ b(\omega_2^2 - \omega_4^2) \\ b(\omega_1^2 - \omega_3^2) \\ d(\omega_1^2 + \omega_3^2 - \omega_2^2 - \omega_4^2) \end{bmatrix} \tag{3.10}$$

Or by rearrange equation (3.10).

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ 0 & b & 0 & -b \\ b & 0 & -b & 0 \\ d & -d & d & -d \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \tag{3.11}$$

We note that the gyroscopic torques depend on the rotors rotational velocities and so is the vector $u^T = (u_1, u_2, u_3, u_4)$. Let us assume that $\omega_1 - \omega_2 + \omega_3 - \omega_4 = q(u)$

From equations (3.6) and (3.7) the overall dynamic model of the quadrotor is

$$\ddot{x} = -(\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi)\frac{u_1}{m}$$

$$\ddot{y} = -(\cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi)\frac{u_1}{m}$$

$$\ddot{z} = g - (\cos\phi \cos\theta)\frac{u_1}{m}$$

$$\ddot{\phi} = \dot{\theta}\dot{\psi}\left(\frac{I_y - I_z}{I_x}\right) - \frac{I_R}{I_x}\dot{\theta}\, q(u) + \frac{L}{I_x}u_2 \tag{3.12}$$

$$\ddot{\theta} = \dot{\phi}\dot{\psi}\left(\frac{I_z - I_x}{I_y}\right) + \frac{I_R}{I_y}\dot{\phi}\, q(u) + \frac{L}{I_y}u_3$$

$$\ddot{\psi} = \dot{\phi}\dot{\theta}\left(\frac{I_x - I_y}{I_z}\right) + \frac{1}{I_z}u_4$$

In this research we will assume there is no roll angle or $\phi = 0$. So, the equation (3.12) will be simplified to,

$$\ddot{x} = -(\sin\theta \cos\psi)\frac{u_1}{m}$$

$$\ddot{y} = -(\sin\theta \sin\psi)\frac{u_1}{m}$$

$$\ddot{z} = g - (\cos\theta)\frac{u_1}{m} \tag{3.13}$$

$$\ddot{\theta} = \frac{L}{I_y}u_3$$

$$\ddot{\psi} = \frac{1}{I_z}u_4$$

The state variables of this model $\dot{x} = f(X, u)$ where $X \in \mathbb{R}^7$ is

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} \cong \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \theta \\ \psi \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{3.14}$$

by using equations (3.13) and (3.14) we find,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \end{bmatrix} = \begin{bmatrix} -(\sin x_4 \cos x_5)\frac{u_1}{m} \\ -(\sin x_4 \sin x_5)\frac{u_1}{m} \\ g - (\cos x_4)\frac{u_1}{m} \\ x_6 \\ x_7 \\ \frac{L}{I_y} u_3 \\ \frac{1}{I_z} u_4 \end{bmatrix} \tag{3.15}$$

In our simulation we will take initial account the desired velocity to be the reference variable not the position. From (3.15) we can separate the equations into two subsystems. First subsystem describes the angles or attitude and the second describes the UAV transition. Let us denote the first subsystem by Sub_1 which represents the rate of quadrotor angular velocities as the following,

$$\begin{bmatrix} \dot{x}_6 \\ \dot{x}_7 \end{bmatrix} = \begin{bmatrix} \frac{L}{I_y} u_3 \\ \frac{1}{I_z} u_4 \end{bmatrix} \tag{3.16}$$

The quadrotor Euler angles, which are the variables x_4 and x_5, can after that be calculated by using pure integration. The input of the second subsystem denoted by *Sub₂* are these Euler angles and u_1 and it is described by:

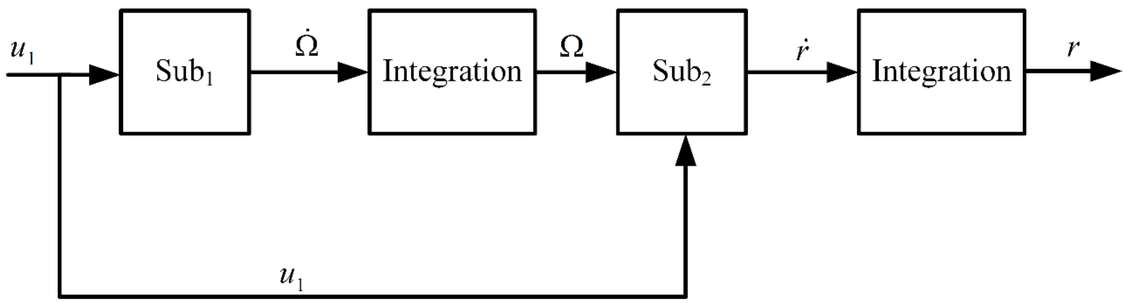

Figure 3.2 Overall Quadrotor Dynamics Model

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -(\sin x_4 \cos x_5)\dfrac{u_1}{m} \\ -(\sin x_4 \sin x_5)\dfrac{u_1}{m} \\ g - (\cos x_4)\dfrac{u_1}{m} \end{bmatrix} \tag{3.17}$$

The position of the vehicle is obtained by the integration. Figure 3.2 shows the structure of the overall model including the subsystems. We use the MATLAB/Simulink to implementation the dynamic model.

3.3 Controller Design For The UAV

The UAV system can be considered to have two main control loops. The first control loop is the vehicle control loop. This control loop is responsible for the generation and stabilization of a currently required movement of the UAV. The second main loop is the mission control loop that computes the desired flight path and control the UAV to follow it. To achieve that we need to determine the commands to be given to the control loop. Direction position control is often used in [54], [62].

For that reason, we assume in this approach that the mission control system commands a desired velocity vector to the vehicle control system. This required velocity vector then must be established and stabilized. In order to obtain the necessary measurements for this velocity control, the vehicle control loop must be equipped with a suitable inertial measurement system (IMU). This IMU delivers the accelerations and angular rates that can be used to further estimate velocities and Euler angles with the help of a Kalman filter. The default command from the mission system is the zero-velocity vector, i.e. the quadrotor UAV should hover at the current position. In this research the main challenge and focus is on the vehicle control loop, i.e. the control of a required velocity vector of the UAV.

The structure of vehicle control is shown in Figure 3.3. In order to achieve and maintain a desired velocity vector, first the necessary attitude of the UAV must be stabilized. Therefore, we propose a decomposition of the control system in an outer-loop velocity

control and an inner-loop attitude control system. In this structure, the inner attitude control loop must be much faster than the outer loop and stabilizes the desired angles that are commanded by the outer loop as shown in Figure 3.3.

Now firstly, let us consider the controller of inner control loop Con_1 or the attitude control loop which stabilize the desired pitch and yaw angles or the desired vector $\Omega_d^T = (\theta_d, \psi_d) = (x_{4,d}, x_{5,d})$. The last two equations of state variable model (3.13). we note that

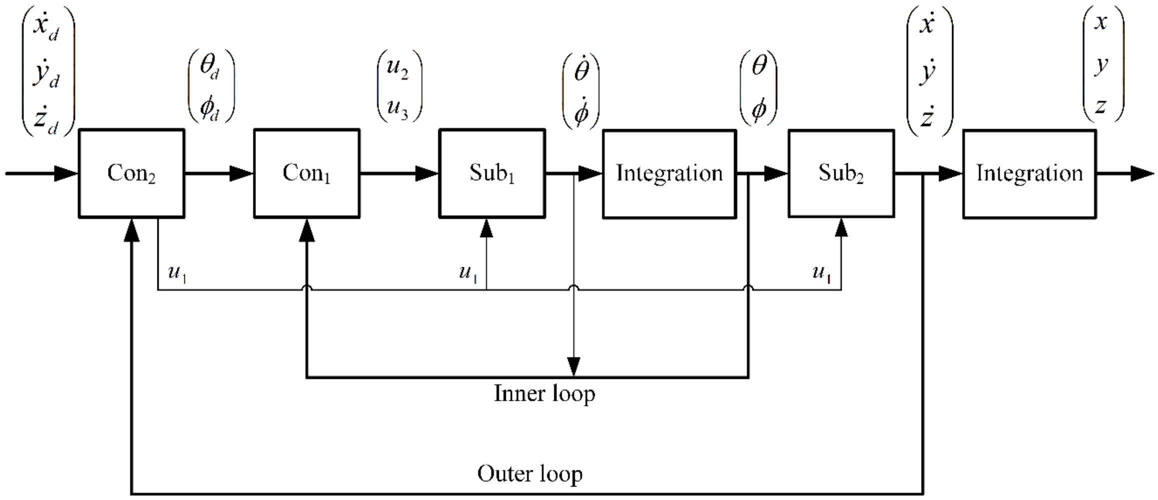

Figure 3.3 Structure of the UAV Vehicle Control

these equations are linear for an integrator and subsystem *Sub_1*. After that we will derive the equations of desired vector of velocity which is the outer loop *Con_2*.

3.4 Inner Loop or Attitude Control System (Con_1)

For the design of the attitude control system *Con_1*. As mention before we note the equations of the attitude is linear. In this research we will use (PID) controller.

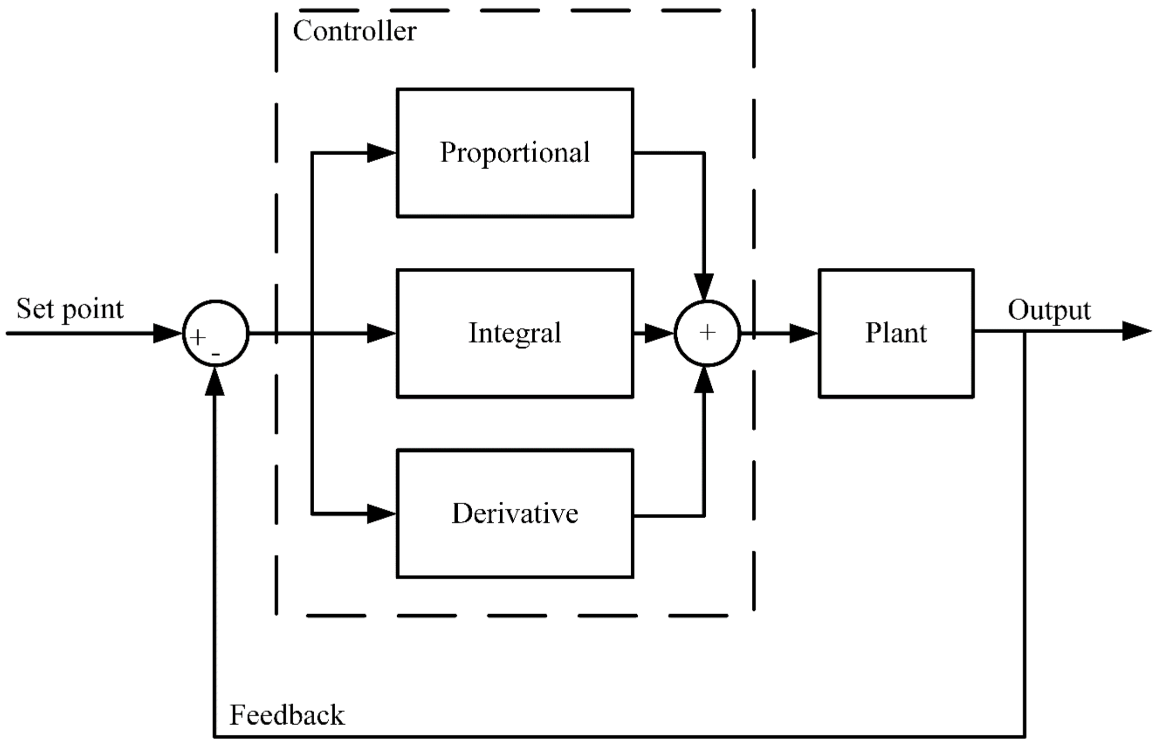

Figure 3.4 PID Block Diagram

3.4.1 PID Controller

PID controller is a combination of three components the proportional, the integral and the derivative as shown in Figure 3.4. The output equation of the PID controller is given as

$$u(t) = k_p\, e(t) + k_i \int e(t)dt + k_d\, \frac{de(t)}{dt} \tag{3.18}$$

The process of getting ideal response from the PID controller by PID setting is called tuning of controller. For desired output, this controller must be properly tuned. One need to get the optimal value of proportional gain k_p integral gain k_i, and derivative gain k_d.

In our research we use MATLAB\Simulink so the tuning will be easy, just by click on tune, the MATLAB will make autotune and give the best k_p,k_i and k_d values.

3.4.2 Outer Loop or Velocity Control System (Con$_2$)

If the attitude control or inner loop is fast enough as we designed before, let us suppose that the desired values of the pitch and the yaw angles, which are $x_{4,d}$ and $x_{5,d}$ are faster than the outer loop or velocity control loop. So, the closed loop system, inner loop, can be considered as a static block, which transmit the calculated values of desired of pitch angles to the following subsystem *Sub$_2$*. From the equation (3.17), the subsystem model *Sub$_2$* will be described as the deferential equation of the nonlinear as the following

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -\frac{u_1}{m}(\sin x_{4d} \cos x_{5d}) \\ -\frac{u_1}{m}(\sin x_{4d} \sin x_{5d}) \\ g - (\cos x_{4d})\frac{u_1}{m} \end{bmatrix} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} \tag{3.19}$$

where x_{4d}, $x_{5d}$and $u_1$are the input variables where we can explain equation (3.19) as the following form

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} \tilde{u}_1 \\ \tilde{u}_2 \\ \tilde{u}_3 \end{bmatrix} = \begin{bmatrix} f_1(x_{4d}, x_{5d}, u_1) \\ f_2(x_{4d}, x_{5d}, u_1) \\ f_3(x_{4d}, x_{5d}, u_1) \end{bmatrix} \tag{3.20}$$

where $\tilde{u}_1$, $\tilde{u}_2$ and $\tilde{u}_3$ are the new input variables in nonlinear form which depend on the other three inputs. Related to these new input variables, the task of control is not difficult because it involves the control of first order independent system that can be solve by using pure proportional controller,

$$\begin{bmatrix} \tilde{u}_1 \\ \tilde{u}_2 \\ \tilde{u}_3 \end{bmatrix} = \begin{bmatrix} k_1(x_{1d} - x_1) \\ k_2(x_{2d} - x_2) \\ k_3(x_{3d} - x_3) \end{bmatrix} \tag{3.21}$$

where k_1, k_2 and k_3 are the controller parameters. We can use these parameters in a way to make the outer loop fast but not as fast as the inner loop. Now we must find the real input variables x_{4d}, $x_{5d}$and u_1 by using the variables of transformed input $\tilde{u}_1$, $\tilde{u}_2$ and $\tilde{u}_3$ by using equation (3.19). To solve these nonlinear equations let us suppose that

$$\alpha = \sin x_{4d} \Rightarrow \cos x_{4d} = \pm\sqrt{1-\alpha^2}, \beta = \sin x_{5d} \Rightarrow \cos x_{5d} = \pm\sqrt{1-\beta^2} \tag{3.22}$$

By substituting (3.22) in (3.19) we find that

$$\begin{bmatrix} \tilde{u}_1 \\ \tilde{u}_2 \\ \tilde{u}_3 \end{bmatrix} = \begin{bmatrix} -(\frac{\alpha\, u_1\sqrt{1-\beta^2}}{m}) \\ -\frac{\alpha\,\beta\, u_1}{m} \\ g - \left(\sqrt{1-\alpha^2}\right)\frac{u_1}{m} \end{bmatrix} \tag{3.23}$$

By solving these equations, we find that

$$\alpha = -\tilde{u}_2 \frac{m}{\beta\, u_1} \tag{3.24}$$

$$\beta = \pm \frac{\tilde{u}_2}{\sqrt{\tilde{u}_1^2 + \tilde{u}_2^2}} \tag{3.25}$$

$$u_1 = m \times \sqrt{(\tilde{u}_3 - g)^2 + \tilde{u}_1^2 + \tilde{u}_2^2} \tag{3.26}$$

Now we can calculate x_{4d} and x_{5d} by inverse equation (3.22). $x_{4d} = \arcsin\alpha$ and $x_{5d} = \arcsin\beta$.

3.4.3 The Overall Control System

The whole control system involves two subsystems (two closed loop systems); inner loop which is for attitude and outer loop which is for velocity control loop. The whole control system is shown in Figure 3.3. From the two-control system (the two loops) we can find the three input variables u_1, u_2 and u_3. In addition, the equation (3.10) can be used to find the rotors angular rates which are ω_1, ω_2, ω_3 and ω_4. The essential benefit of the whole system is that the controllers and feedback linearization are not difficult to do.

3.5 Summary

In summary the quadrotor model is given by the equation (3.13). This equation is separated into two subsystems; the first sub system is given by the equation (3.16) and the other subsystem is given by the equation (3.17). we use the PID controller for the first subsystem while we use the backstepping controller for the second subsystem. This model will be used in the simulation later

CHAPTER 4

THREE-DIMESIONAL NONHOLONOMIC UAV MODEL

We consider a quadrotor model of UAV. In this chapter, we will propose the path planning algorithms in three-dimension for the UAV. These navigation algorithms are planned based on the current state of each UAV and with relationship to each other in general environmental conditions. These environmental conditions could be the presence of an enemy or any form of danger, the presence or absence of other UAVs or even the target to reach. All these concepts will be discussed in the next sections.

In this research we will take a fleet of UAVs. The fleet are flying with each other and with a safe distance between each other. All these UAVs have a main goal, which is looking for the target. The UAVs are attacked by enemy that also have nonholonomic constraints. Its mission is to attack the nearest UAV in the fleet. In the presence of an enemy, the UAV tries to evade this attacker. The physical applications of our research would call for the use of GPS in the UAVs in order to distribute location calculations. In addition, other sensors are needed in order to collect proximity and velocity measurements.

Our goal in this research is to make a fleet of UAVs to fly in three dimensions in a way, which is inspired by movements of school of fish, when they are foraging and when they are evading the attacker.

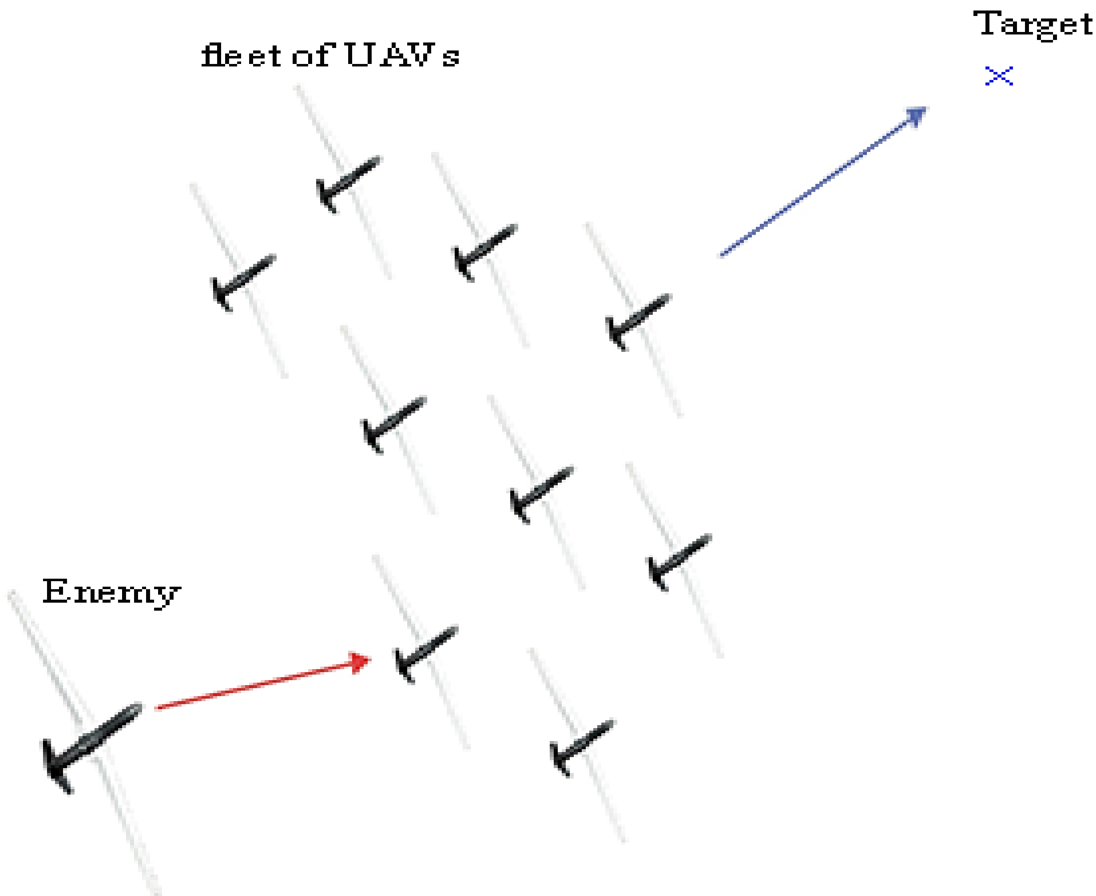

Figure 4.1 Air Combat [58]

For each UAV, there are two behaviors of navigation; foraging behavior and evading behavior.

4.1 Foraging Behavior

The behavior of the UAVs like fishes, they forage when they are looking for a target. In this case the danger or attacker is absent, as result the fleet of UAVs will be modeled to conduct food searching as a school of fish. To realize this, we will use an approach similar to the one proposed in [47], the Fish-Prey Algorithm that will be discussed in more details in the next section.

4.1.1 Fish-Prey Algorithm

Adaptive networks concept is used in the Fish-Prey Algorithm and the mobility is added as another attribute. We use the adaptive network because the connection between nodes will be strong and it is examined from the point of view of nodes group that are able to learn and interact with each other to overcome distributed inference and processing challenges in real time. All nodes are locally related with each other, with their neighbors, which are continuously changing because the movement of nodes. The school of fish behavior in self-organization could be modeled using adaptive algorithm. To draw parallels between a school of fish and the fleet of UAVs, each node in the network in fact represent a fish in the school while they are looking for food, and in turn represent a UAV in the fleet. So, from now on we will say the nodes, to mean UAV in the fleet or a fish in the school. To do this model we have to take into account the assumptions below;

- For practical application, we suppose that sensors of proximity and GPS are used.
- Every UAV is knowing the orientation and position of all UAV in neighborhood.
- Each UAV is knowing where the target is (target location).
- Every UAV is knowing the orientation and position of the UAV enemy but once the enemy enters the safe distance of the UAVs, they will activate the engagement rule.

- We assume all UAVs have physical dimension [63], and physical rule motion govern.
- We assume the target for nodes (food) is stationary, but the enemy UAV is moving.
- The adaptive network is strongly connected.
- The UAVs are moving in 3-D, but some aerodynamic are ignored.

4.1.2 Diffusion Adaption

The N UAVs (nodes) group, which is distributed in a space, is examined. Each node k is distributed and independent at every time instant i, everyone evaluates a scalar random process $d_k(i)$ a regression vector connecting the realization of a random process $\bar{u}_{k,i}$, where $\bar{u}_{k,i}$ and $d_k(i)$ are correlated. By using the random process $\left[d_k(i),\ \bar{u}_{k,i}\right]$, every node k tries to find some vector parameter w^0. These parameters and processes are related by using the following equation

$$d_k(i) = \bar{u}_{k,i}\, w^0 + n_k \tag{4.1}$$

Where n_k is a perturbation independent of other variables usually taken as a zero-mean white noise.

The data is spread, in distributed algorithm, to every node that means it is in communication with its local environment – the nodes subset – that provide the information to other nodes in the network. When the communication is direct between them, two nodes are connected or linked. Every node is always connected to itself.

The node N_k neighborhood indicates the group of nodes which are connected to that node including the node itself. The goal of the network is to find the w^0 parameter which minimizes the global objective function [41].

$$g_{global}(w) = \sum_{k=1}^{N} E\left|d_k(i) - \bar{u}_{k,i}\, w\right|^2 \tag{4.2}$$

where E denotes to the expectation operator.

There are different types of diffusion adaptation technique. Many papers explain these techniques such as [47]-[50]. In this research we will use the Adapt-Then-Combine (ATC) algorithm to solve the problem of distributed adaptation. The algorithm of ATC has two steps. The first step is adaptation and the second step is combination. A real positive coefficient $a_{l,k}$ is used by this algorithm to specify weights to the communication connection between every node k and its neighbors l. This weighting must satisfy the following equation

$$\sum_{l=1}^{N} a_{l,k} = 1 \quad a_{l,k} = 0, \forall\, l \notin\, N_k \tag{4.3}$$

Each node k calculates the local information from its neighbors during the step of adaptation to adapt locally. The updates of the states will be evaluated from its previous value. This also happen in this step. To apply the ATC in our research we assume that the location of the node (UAV) represents the states at the time i. And in the combine step we use the intermediate values of neighbors calculated during adapt step and combine them together using the weights $a_{l,k}$. We can describe the ATC by the following equations

$$\begin{cases} \psi_{k,i} = w_{k,i-1} + \mu_k \, \bar{u}_{k,i}^T [\hat{d}_k(i) - \bar{u}_{k,i} \, w_{k,i-1}] \\ w_{k,i} = \sum_{l \in N_k} a_{l,k} \, \psi_{l,i} \end{cases} \tag{4.4}$$

where $w_{k,i}$ is the parameter vector which the ATC tries to calculate of a UAV's or target's desired location at time i, μ_k is the non-negative step-size which is applied by node k, $\hat{d}_k(i)$ is the regression form of $d_k(i)$. The physical meaning of all these variables will be explained in the next subsection.

4.1.3 Measurement Model

The aim of measurement model is to estimate the behavior relationship of the school of fish model. We apply this behavioral model in the UAV model. Figure 4.2 explains this. The vector parameter w^0 which is considered as the target location (fish's food source location) that is the target of network of school of fish. This figure also shows the distance between the node (fish) k and the food target w^0 is denoted by $d_k^0(i)$ and can be calculated as follows

$$d_k^0(i) = \bar{u}_{k,i}^0 (w^0 - x_{k,i}) \tag{4.5}$$

The node (fish) k is located at $x_{k,i}$ that is continuously changing, and $\bar{u}_{k,i}$ is the unit direction vector which connects the location of nodes to the target. This vector contains two angles; azimuth angle $\theta_k(i)$ and elevation angle $\phi_k(i)$ thus

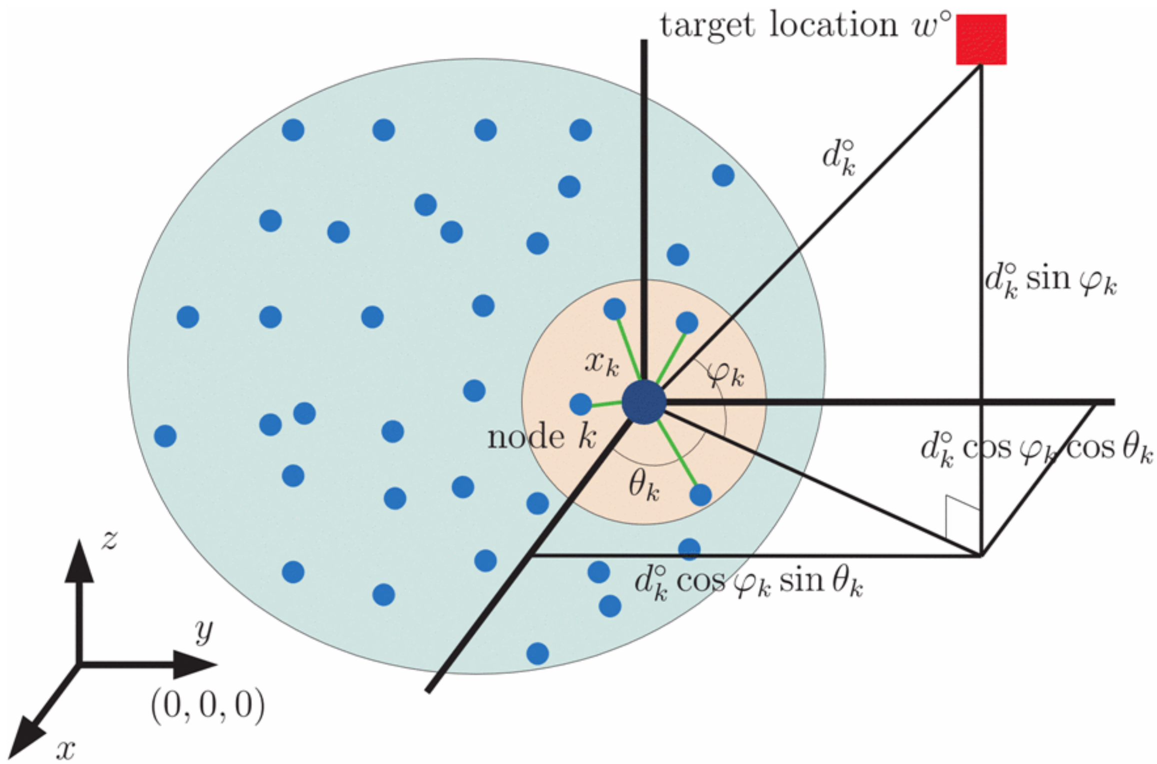

Figure 4.2 System Model in 3D

$$\bar{u}_{k,i}^{0} = [\cos\theta_k(i)\cos\phi_k(i) \quad \sin\theta_k(i)\sin\phi_k(i) \quad \sin\phi_k(i)] \tag{4.6}$$

Because the fishes live in a noisy environment, we should consider some disturbance to measurement accuracy. In the case of UAV, the disturbance come from flying against the wind. So, we should consider this disturbance in our calculation as;

$$\bar{u}_{k,i}^{k} = \bar{u}_{k.i}^{0} + n_{k,i}^{\bar{u}}$$

$$d_k(i) = d_k^0(i) + n_k^d(i) \tag{4.7}$$

$$\hat{d}_k(i) \cong d_k(i) + \bar{u}_{k,i}\, x_{k,i}$$

where $\bar{u}^0_{k.i}$and $d^0_k(i)$ are the true value, $\hat{d}_k(i)$ is a linear regression model for the distance changing $d_k(i)$, and $n^{\bar{u}}_{k,i}$ and $n^d_k(i)$ are additive disturbance terms.

To behave as a school of fish for foraging, let us suppose that every fish denotes a node in the network. The goal for this representation to determine two different targets simultaneously; location of the food source w^f and the location of the attacker w^p.

4.1.4 Control Motion Algorithm

The location x_k for each node's k is calculated in the mobile network as

$$x_{k,i+1} = x_{k,i} + \Delta t \; v_{k,i+1} \tag{4.8}$$

where i denotes the time instant, Δt denotes the sampling time for the node movement to move from one location to the next, and $v_{k,i+1}$ denotes the velocity of the node (fish). This velocity is determined by the following behaviors of fish;

1. The movement of the fish (node) to the food source.
2. Self- organizing behavior of the node with others.
3. Coherent motion.
4. Evasion, in the case of danger.

Because the movement of the attacker (enemy), every node will determine a local estimate of source of food $w^f_{k,i}$ and the attacker $w^p_{k,i}$ in real time. Figure 4.3 elaborates computing velocity of the nodes cognitive.

Movement to The Food Source

We will use as in [43], Figure 4.3 to identify distinct zone such that response of node relies on it. The concentric circles shown have a radii of r_p and $2r_p$ from the attacker. These separations- Zone I, Zone II, Zone III, and Zone IV will help to define rules for the next action by each node.

<u>Zone I</u>

The attacker in this zone is far away from the node k; this means that the distance between the attacker and the nodes is more than $2r_p$. As result, the nodes continue its foraging behavior towards the target. In this case the node velocity will be;

$$v_{k,i+1}^{a} = \frac{w_{k,i}^{f} - x_{k,i}}{\left\| w_{k,i}^{f} - x_{k,i} \right\|} \qquad \text{(Zone I)} \tag{4.9}$$

<u>Zone II</u>

When the location of the node between radii r_p and $2r_p$, and to the right-hand side in Figure 4.3. The attacker is moving toward the node, but it is not too close. As a result, the node stop researching for the food target and joins its neighbors. In this case the node velocity is;

$$v_{k,i+1}^{a} = 0 \quad \text{(Zone II)} \tag{4.10}$$

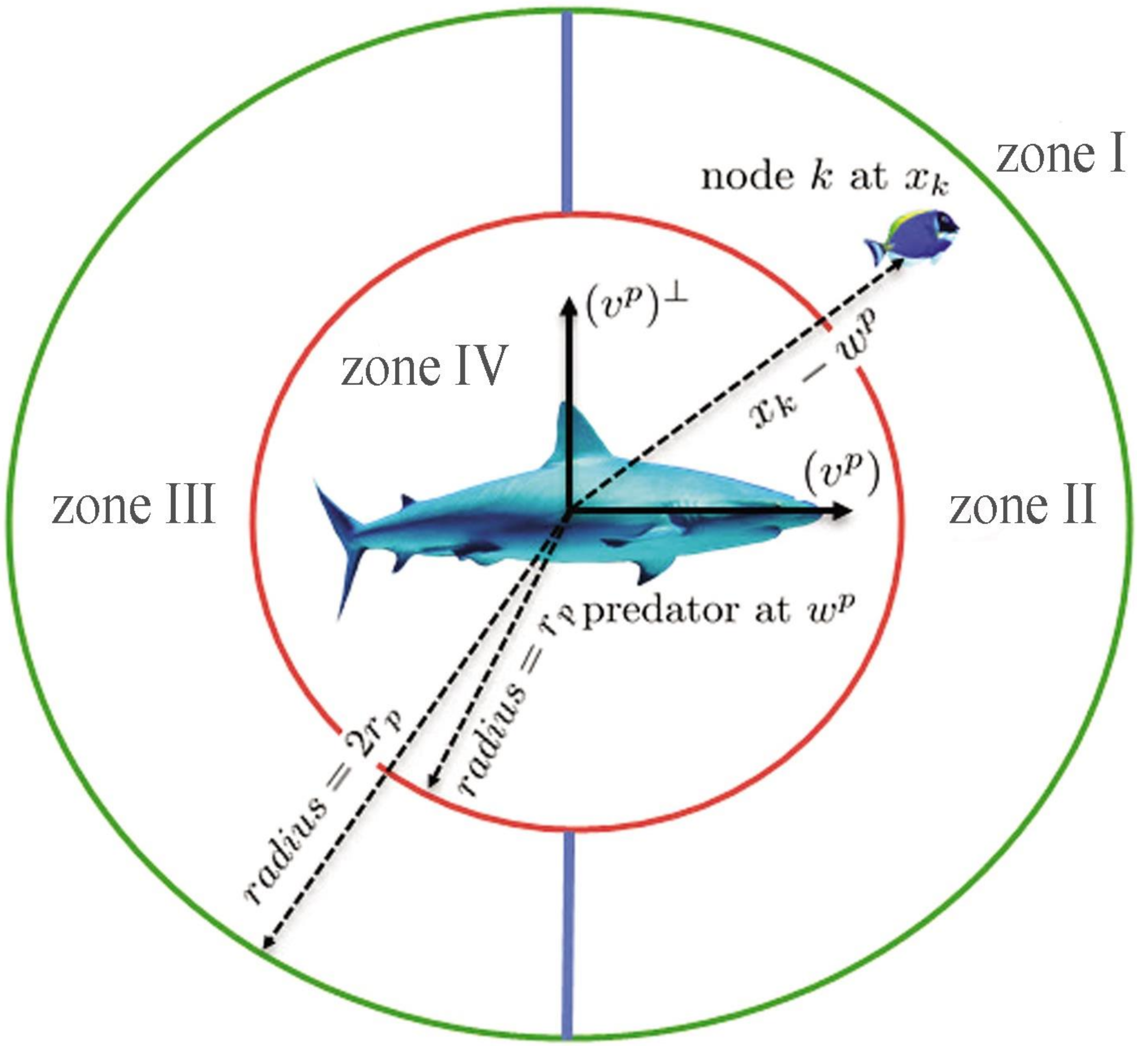

Figure 4.3 The Behavior of Fish Cognitive

<u>Zone III</u>

In this zone the location of the node between radii r_p and $2r_p$ and to the left-hand side as shown in Figure 4.3, as result the attacker is going away from the node. In this case the response of the node will move in the reverse direction of the attacker. So, node velocity will be

$$v_{k,i+1}^{a} = -\frac{v_{k,i}^{p}}{\left\| v_{k,i}^{p} \right\|} \quad \text{(Zone III)} \tag{4.11}$$

Zone IV

This is the final case, where the node is located in the danger zone; the distance between the attacker and the node is less than r_p. In this case the cognitive evasive control strategies will be developed as result to escape or evade the attacker. The node velocity will be.

$$v_{k,i+1}^{a} = \left(\frac{r_p}{\left\| x_{k,i} - w_{k,i}^{p} \right\|} - 1 \right) \left(x_{k,i} - w_{k,i}^{p} \right) \quad \text{(Zone IV)} \tag{4.12}$$

Self-Organization Behavior

Once the attacker enters the danger zone of the fishes, the network between them is broken and they are divided into smaller groups. To make reunion with each other and biological organization to take place, each node at the boundaries of the smaller groups calculate the position to the other groups and move to them. First, every node must determine if it is on the edge of the broken-down school. There are three edges defined; the front, the right, and the left edges. The node k enumerates its neighbors l in each direction by using the following equation

$$x_{l,i}^{k} = W\left(v_{k,i}\right)^{T} (x_{l,i} - x_{k,i}) \tag{4.13}$$

where $x_{l,i}^{k}$ is the position of the node l with respect to k and

$$W(v) = \begin{bmatrix} {v_1}/{\|v\|} & {v_2}/{\|v\|} & -{v_3}/{\|v\|} \\ {v_2}/{\|v\|} & {v_1}/{\|v\|} & {v_2}/{\|v\|} \\ {v_3}/{\|v\|} & {v_2}/{\|v\|} & {v_1}/{\|v\|} \end{bmatrix} \tag{4.14}$$

is an orthogonal matrix defining the local coordinate system for fish movement with the vector velocity $v^T = (v_1, v_2, v_3)$.

From the equation (4.13) if the value of $x_{l,i}^k$ in the first coordinate is more than zero, then the l position is in front of k. If the value of $x_{l,i}^k$ second coordinate is more than zero, then l position in left of k. Finally, if the value of $x_{l,i}^k$ is less than zero then l position in the right of k. Table 4.1 summarize all these coordinates.

Table 4.1 Estimating Nodes in Outer Boundaries of Fragmented Groups

Condition	**Implication**
First coordinate of $x_{l,i}^k > 0$	Node l is in front of k
Second coordinate of $x_{l,i}^k > 0$	Node l is in left of k
Second coordinate of $x_{l,i}^k < 0$	Node l is in right of k

We can say, if the neighbor number in the front less than one, then the node k belongs to the frontal edge of the fragment groups. The same thing in the right and left edge. After that, the nodes in the edge search for another group. For instance, nodes on the front edge will find the nearest frontal node outside its neighborhood and advance toward that nodes. this, node k will perform the following equation.

$$\hat{l} = \arg\min_{l} \left\| x_{l,i}^{k} \right\| \ \mid l \in N_k^F \backslash N_k \tag{4.15}$$

$$v_{k,i+1}^{b} = \begin{cases} 0, & if\ \ \hat{l}\ \ is\ empty \\ \dfrac{x_{\hat{l},i} - x_{k,i}}{\left\| x_{\hat{l},i} - x_{k,i} \right\|}, & otherwise \end{cases} \tag{4.16}$$

N_k^F refers to the enumerated nodes that are in the front edge, N_k^L and N_k^R will also refer to that of the left and right respectively.

Figure 4.4 shows two fragment group, the connection between the fishes are specified by lines. The highlight indicated that, one fish at the fragmental edge (left group) and one fish on the left group (right group). They will make regrouping by movement along the arrow direction

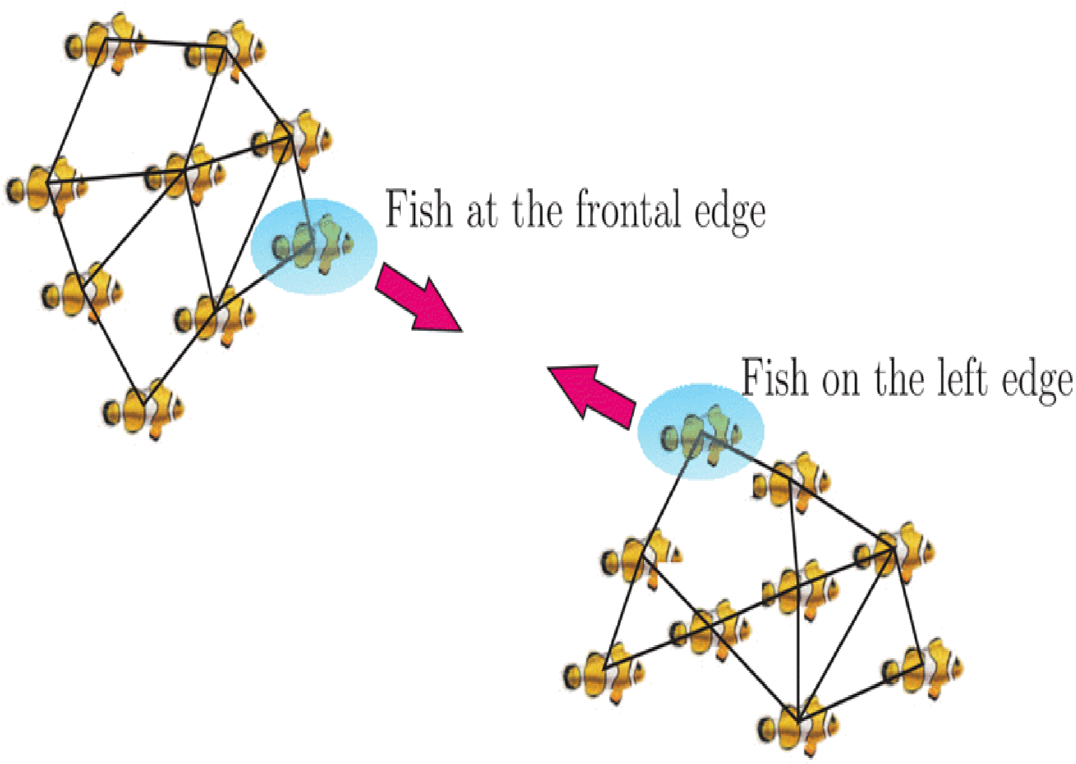

Figure 4.4 Reunion Between Two Groups

Coherent Motion

The behavior of the nodes is like school of fish, they do not only move to reach to the food source and avoid the attacker, but also advance collectively to confuse the attacker. To prevent the collision between each other, they have a safe distance r from their neighbors. This concept is called potential field. For nodes to avoid collision with each other, each node k satisfies the following equation

$$r - \epsilon \leq \|x_k - x_l\| \leq r + \epsilon \quad \forall \ l \in N_k\{k\} \tag{4.17}$$

where r is the ideal distance between the nodes that ensure coherent motion and ϵ is a small positive number.

The following objective function is used for the objectives of anti-collision and cohesion [54];

$$g_k(v_k) = \sum_{l \in N_k\{k\}} [\|(x_k - \Delta t \ v_k) - (x_l - \Delta t \ v_l)\| - r^2] \tag{4.18}$$

We need to minimize the above objective function. The minimization problem tries to reduce the difference between r and the distance between the update nodes in equation (4.17). So, we find the derivative of equation (4.18) to find the optimal value of v_k. [54] shows that the approximate optimal value which is

$$v_k = \sum_{l \in N_k\{k\}} [v_l - (1 - \frac{r}{\|x_l - x_k\|}) \frac{x_l - x_k}{\Delta t}] \tag{4.19}$$

Because of the coherent motion, the velocity of the node is

$$v_{k,i+1}^{c} = v_{k,i}^{g} + \gamma\ \delta_{k,i} \tag{4.20}$$

where $v_{k,i}^{g}$ is the local velocity value of the network's center of gravity, γ is non-negative number and v^{g} will be

$$v^{g} \cong \frac{1}{N}\sum_{l=1}^{N} v_{l} \tag{4.21}$$

Because of our algorithm is distributed, $v_{k,i}^{g}$ should also be calculated in a distributed method. Here we will satisfy the diffusion adaptation technique, which was explained in Section (4.1.2). Let the objective function be

$$g^{v}(v^{g}) = \sum_{k=1}^{N} \left\| v_{k,i} - v^{g} \right\|^{2} \tag{4.22}$$

By applying the ATC diffusion algorithm with the equation (4.4)'s structure will be

$$\begin{cases} \psi_{k,i} = (1-\mu_{k}^{v}) v_{k,i-1}^{g} + \mu_{k}^{v}\ v_{k,i} \\ v_{k,i}^{g} = \sum_{l \in N_{k}} a_{l,k}^{v}\ \psi_{l,i} \end{cases} \tag{4.23}$$

Where superscript v shows a relative with v^{g} and $a_{l,k}^{v}$ that satisfies equation (4.3)

Distributed Velocity Estimate

By combining all the specified biological behaviors, the nodes will adjust their velocities as

$$v_{k,i+1} = \lambda\ I_{k,i}\left(\alpha\ v_{k,i+1}^{a} + \beta\ v_{k,i+1}^{b}\right) + \left(1 - \lambda\ I_{k,i}\right) v_{k,i}^{g} + \gamma\ \delta_{k,i} \tag{4.24}$$

where λ, α, β and γ are nonnegative weighting elements and $I_{k,i}$ is a switching function given by

$$I_{k,i} = \begin{cases} 0 & \text{if } (v^a_{k,i+1} = 0 \text{ and } v^b_{k,i+1} = 0) \\ 1 & \text{otherwise} \end{cases} \tag{4.25}$$

when $I_{k,i} = 0$ th is means that the node k is in the zone II and does not need to reunion. Additionally, the maximum speed of the nodes is bounded by v_{max}, the magnitude of $v_{k,i+1}$ will be scaled. Moving forward we suppose that each node in the network is setting its speed according to the equation (4.24).

4.2 Global Variable Estimation

In this section we will discuss the estimation of location and velocity of the attacker and location of the food, in addition the estimation of the center gravity velocity.

4.2.1 Velocity and Location of The Attacker and Food Estimation

As we mention before, we assume that the attacker is moving while the food source is fixed. These variables are global. Because the algorithms of estimating attacker location $w^p$and food location w^f are the same just by replacing p instead of f, we will elaborate only on the estimation of attacker location w^p. The goal for the nodes (UAVs) is to calculate the location of the attacker w^p that minimizes the following objective function,

$$g^p(w^p) = \sum_{k=1}^{N} E\left|\hat{d}^p_k(i) - \bar{u}_{k,i}\, w^p\right|^2 \tag{4.26}$$

By applying ATC diffusion algorithm, we find that

$$\begin{cases} \psi_{k,i}^p = w_{k,i-1}^p + \mu_k^p \, \bar{u}_{k,i}^{p\,T} [\hat{d}_k^p(i) - \bar{u}_{k,i}^p \, w_{k,i-1}^p] \\ w_{k,i}^p = \sum_{l \in N_k} a_{l,k}^p \, \psi_{l,i}^p \end{cases} \tag{4.27}$$

where $a_{l,k}^p$ is defined as follows

$$\sum_{l=1}^{N} a_{l,k}^p = 1 \qquad a_{l,k} = 0, \forall \, l \notin \, N_k \tag{4.28}$$

Since our research model in (4.7) is a geometry-bearing (see (4.5)), we can use this fact to simplify the first equation of (4.27) under reasonable approximation,

$$w_{k,i-1}^p - x_{k,i} \approx \left\| w_{k,i-1}^p - x_{k,i} \right\| \bar{u}_{k,i}^{p\,T} \tag{4.29}$$

Now by rewriting equation (4.27) we will find

$$\begin{cases} \psi_{k,i}^p = (1 - \mu_k^p) w_{k,i-1}^p + \mu_k^p \, s_{k,i}^p \\ w_{k,i}^p = \sum_{l \in N_k} a_{l,k}^p \, \psi_{l,i}^p \end{cases} \tag{4.30}$$

This update is a combination of the current measurement $s_{k,i}^p$ and the previous target location $w_{k,i-1}^p$. The estimation of the attacker velocity with respect to every node is,

$$v_{k,i}^p = \frac{1}{\Delta t} (w_{k,i}^p - w_{k,i-1}^p) \tag{4.31}$$

4.2.2 Velocity of Center Gravity Estimation

We will estimate the center gravity v^g in a distributed way. The velocity of the center gravity v^g is the average velocities of all nodes in the network as in equation (4.21). To estimate v^g we consider the global cost function as

$$g^{v}(v^{g}) = \sum_{k=1}^{N} \left\| v_{k,i} - v^{g} \right\|^{2} \tag{4.32}$$

By using the arguments in [50] and the same structure in equation (4.27), we can calculate $v_{k,i}^{g}$ by the following diffusion algorithm

$$\begin{cases} \psi_{k,i} = (1-\mu_k^{v}) v_{k,i-1}^{g} + \mu_k^{v}\, v_{k,i} \\ v_{k,i}^{g} = \sum_{l \in N_k} a_{l,k}^{v}\, \psi_{k,i} \end{cases} \tag{4.33}$$

where $a_{l,k}^{v}$ is a set of nonnegative real coefficient, and the superscript v is used for estimating of v^{g}.

4.3 The Attacker Behavior

The attacker spoor one node location at i time. We suppose that the attacker keeps tracking of the close nodes. The attacker calculates the distance $d^{n}(i)$ and the direction $\bar{u}_{i}^{n}$, at every time instant i of the closest node. Then these observations are used to calculate the node measured location as the following

$$s_i^{n} = w_i^{p} + d^{n}(i)\, \bar{u}_i^{n\,T} \tag{4.34}$$

After that, the attacker updates the node location according to

$$w_i^{n} = (1-\mathcal{V}) w_{i-1}^{n} + \mathcal{V}\, s_i^{n} \tag{4.35}$$

After the attacker estimate the nodes location, the attacker advances toward them. The attacker velocity and position are updated using the following estimation.

$$\begin{cases} v_{i+1}^{p} = c\ v_{max} \dfrac{w_i^n - w_i^p}{\left\| w_i^n - w_i^p \right\|} \\ w_i^p = w_i^p + \Delta t\ v_{i+1}^{p} \end{cases} \tag{4.36}$$

where c is a positive number to control the attacker speed.

CHAPTER 5

RESULTS AND DISCUSSIONS

In this chapter, we present simulation of the quadrotor systems in different cases. The parameters used follow that of the experimental setup developed in [61]. The simulation shows the movement of the UAV in 3-D. Here we consider the behavior of the UAV in the case with no attacker and the case with a single attacker.

5.1 Simulation of The Quadrotor Control

As we explained the derivations in Chapter 3. The quadrotor model is implemented by using the parameters in Table 5.1 and by using the MATLAB\SIMULINK. As we mention before this model has two loops. First loop which is internal loop we use the PID controller. There are many methods to find the k_p, k_i and k_d parameters (gains) of this controller. They include trial and error, by this method we have to suppose some values for gains to give us the perfect results. But this way takes time and may not give us the ideal results. Another method is Ziglar-Nichols and others. Matlab also provide auto tuning for the PID controller by using auto tuning. In MATLAB\SIMULINKE we find in the PID window the tuning button. By clicking to this button, the MATLAB calculate the perfect values. We use this way to find the gains.

From equation (3.16) in the block of $Con._1$ we use two PID controller for each equation. The gains are k_p=0.5, $k_i = 0.1$, and $k_d = 0.3$.

Table 5.1 The Quadrotor Parameters

Variable	Meaning of the variable	Value
g	Gravity constant	9.81 m/s^2
m	Mass of Quadrotor	0.5 kg
L	Length of the Lever	0.2 m
$I_x = I_y$	Inertia Matrix	$4.85 \times 10^{-3}\ kg.m^2$
I_z	Inertia Matrix	$8.81 \times 10^{-3}\ kg.m^2$
I_R	Inertia Matrix	$3.36 \times 10^{-5}\ kg.m^2$
b	Thrust Factor	$2.92 \times 10^{-6}\ kg.m$
d	Drag Factor	$1.12 \times 10^{-7}\ kg.m^2$

The parameters for the velocity controller $k_1 = k_2 = k_3 = 5$.

5.1.1 Angular Velocity

Figure 5.1 show the angular velocities during the simulation. We note that angular velocity of pitch angle $d\theta/dt$ has more fluctuation which reach more than 45 rad/s. While the angular velocity of yaw angle $d\psi/dt$ has less fluctuation. However, all the angular velocities have fast response.

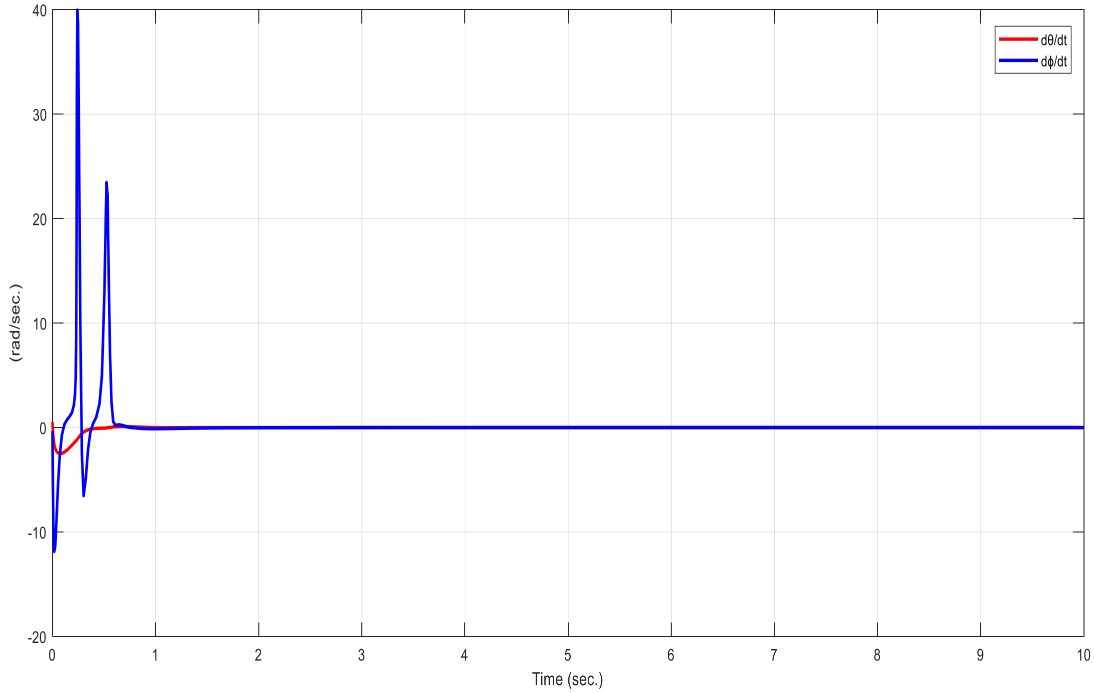

Figure 5.1 Angular Velocities

5.1.2 Control Velocity

Another simulation in this section is the control of given desired velocity vector. So, we suppose that the hovering is the starting state of our simulation (this means all velocities are zero). The task of the control here to involve the desired velocity stabilization, which we assume that these desired assumed velocities are $\dot{x}_d = \dot{y}_d = \dot{z}_d = 0.4\, m/s$. And to generate linear movement. Figure 5.2 show the plot of all velocities when the control is working.

From this plot we note that the desired state is reached after short transition phase and the quadrotor is going to the steady state rate and advancing with fixed velocity. When the quadrotor flight in fixed velocity, the angles also are staying at steady state case 0.4 m/s which is the desired value for all velocities that we assume before. The angular rates are all zero after some initial movement.

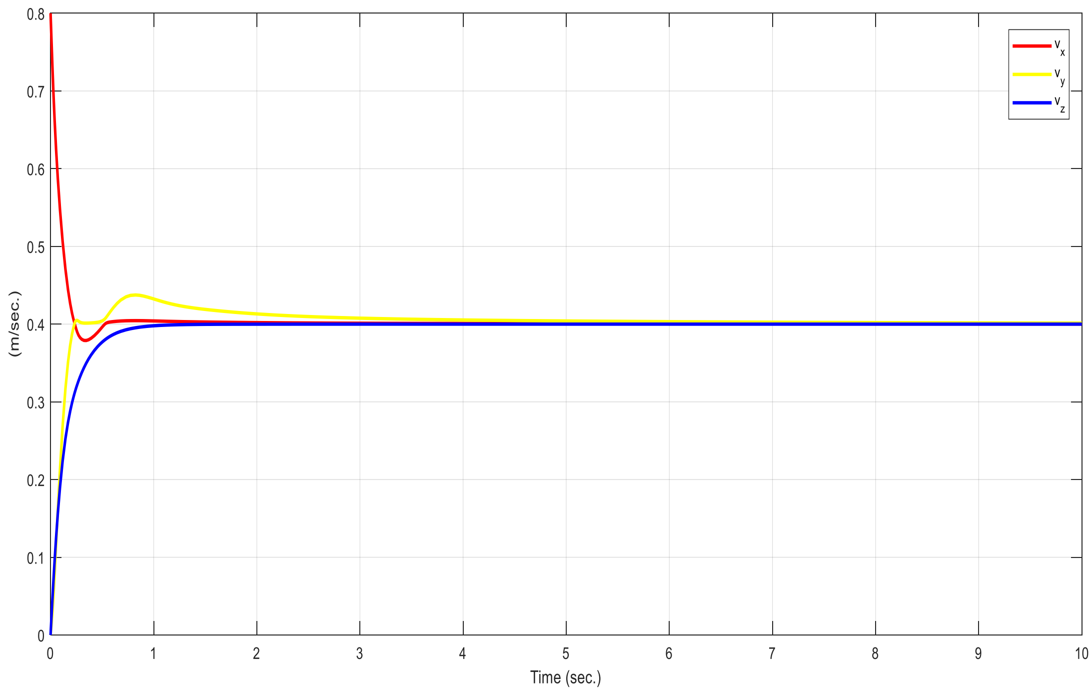

Figure 5.2 Velocity Control

5.2 Simulation of Multiple of UAVs

This section is the essential section in our research, we will elaborate in this section and discuss the simulation of the multiple UAVs. We will simulate two cases; first case without attacker and second case with attacker.

Table 5.2 The Parameters of Simulation of Multiple UAV

Variable	**Meaning of the variable**	**Value**
N	UAVs number	27
N_k^{max}	Maximum number of neighbors of UAV "k"	3
$w_{k,1}^{p}$	Initial location of the attacker	(5,5,5)
$w_{k,1}^{f}$	Initial location of the food target	(50,50,50)
Δt	Sampling time	0.2 sec
r_p	Attacker radius (attacker zone)	10 m
r	The ideal distance between the node	6 m
R	Rang of communication among UAV	4
μ_k	Non-negative step-wise scaler	0.5
α	Non-negative weighting element	1
β	Non-negative weighting element	2
γ	Non-negative weighting element	2
λ	Non-negative weighting element	0.5

5.2.1 Simulation with No Enemy

Firstly, we will simulate without attacker, which means no evasion. From the Figure 5.3 we have to identify the shape of attacker, UAV, and the target as the following

O ≡ Nodes (fleet of UAVs)

× ≡ Target (food)

The following Figures (5.3-5.8) show the position for successive instant.

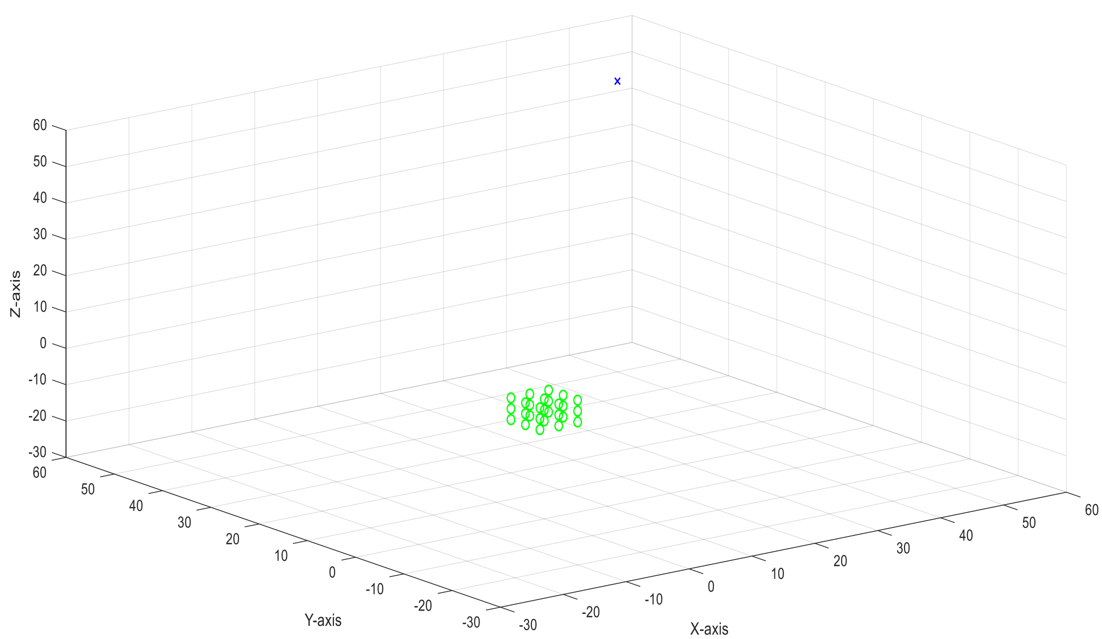

Figure 5.3 Foraging Behavior 1

In the case without attacker, nodes (UAVs) will be in foraging case. They will move in coherent motion, and they will keep safe distance between them to avoid collision with any one of the nodes.

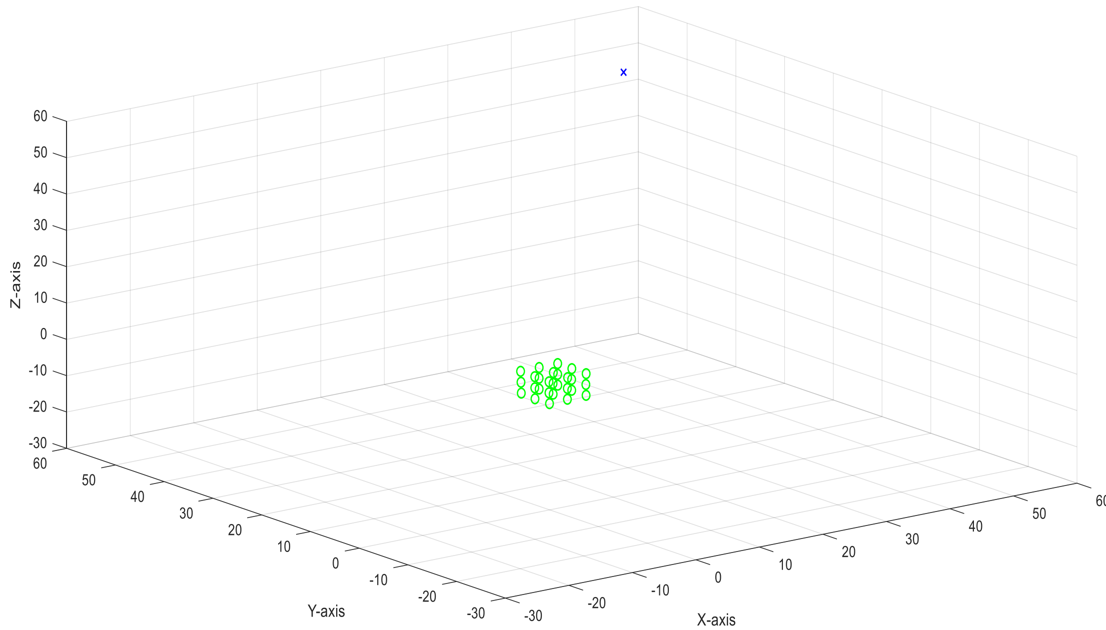

Figure 5.4 Foraging Behavior 2

Also, we note that from the figures, the nodes\UAV are moving directly from their location up to reach to the target and they stop moving and stay there which means their mission is finished.

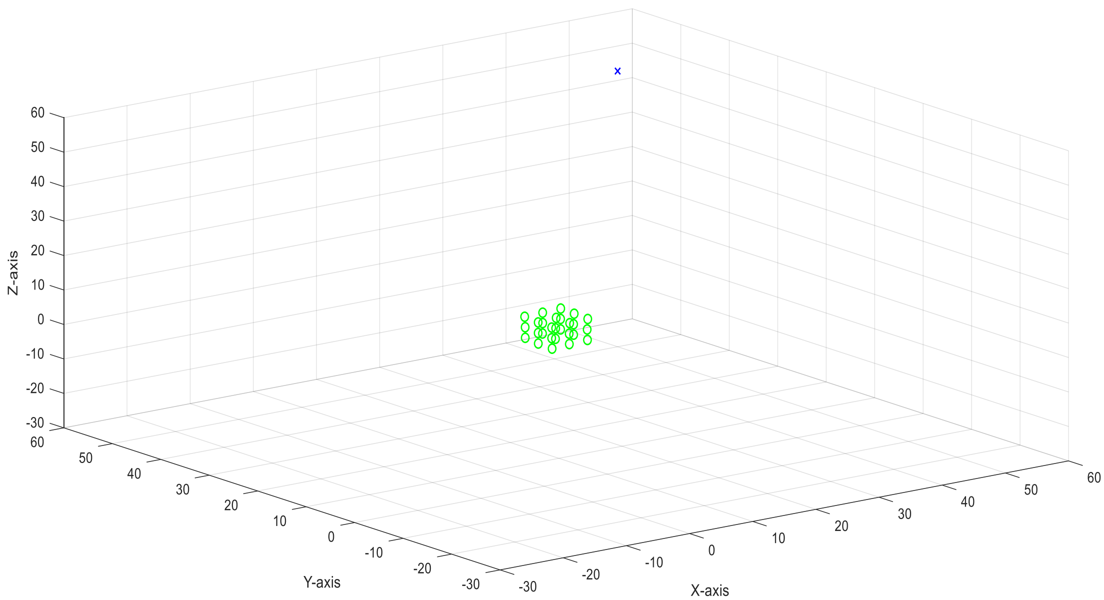

Figure 5.5 Foraging Behavior 3

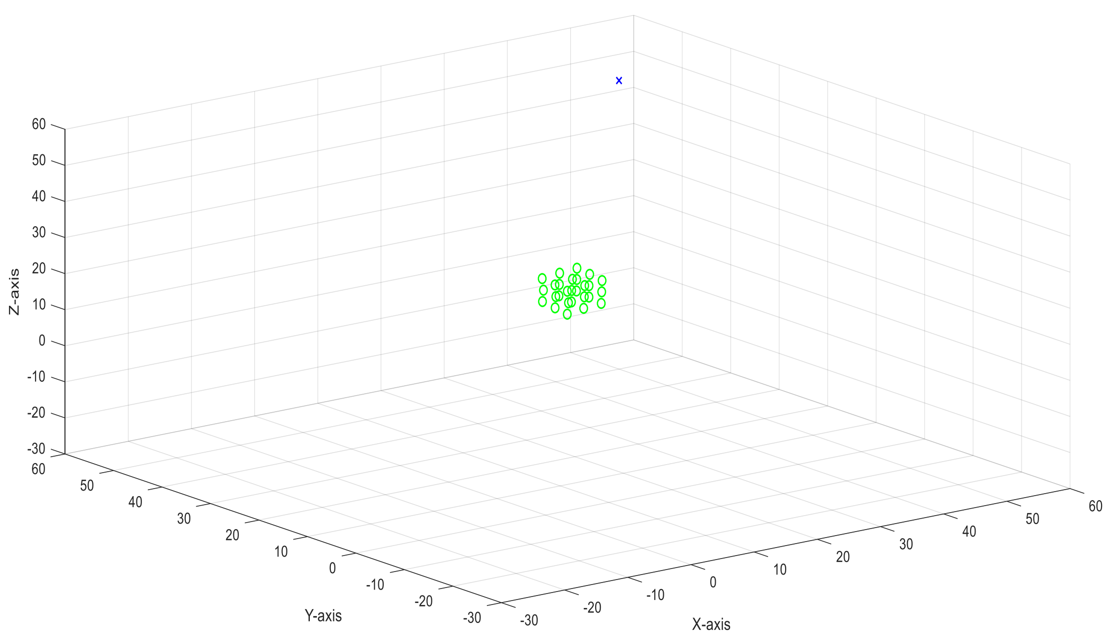

Figure 5.6 Foraging Behavior 4

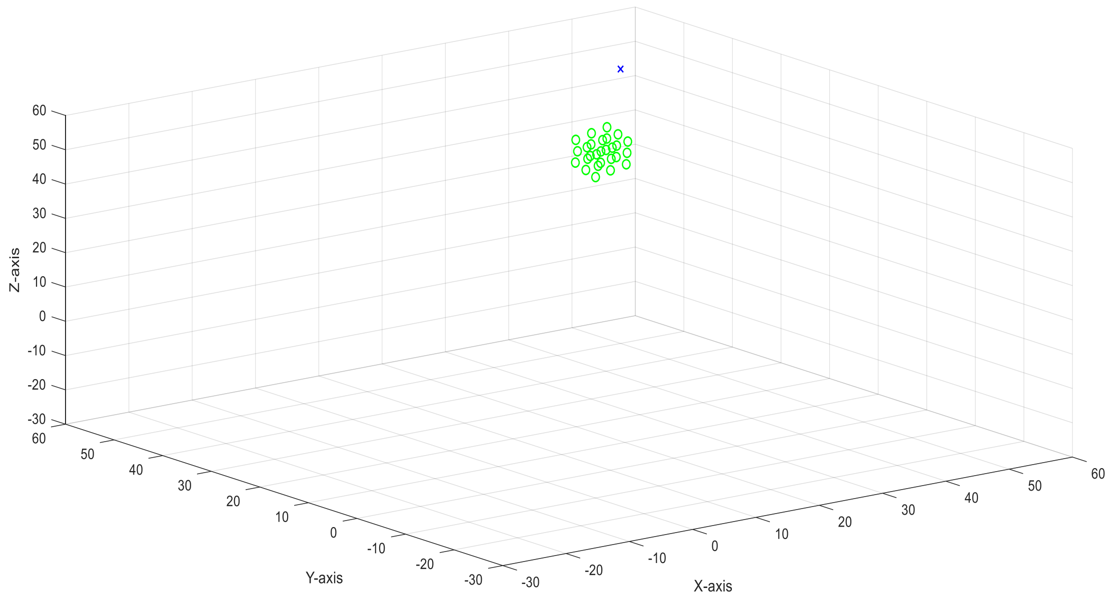

Figure 5.7 Foraging Behavior 5

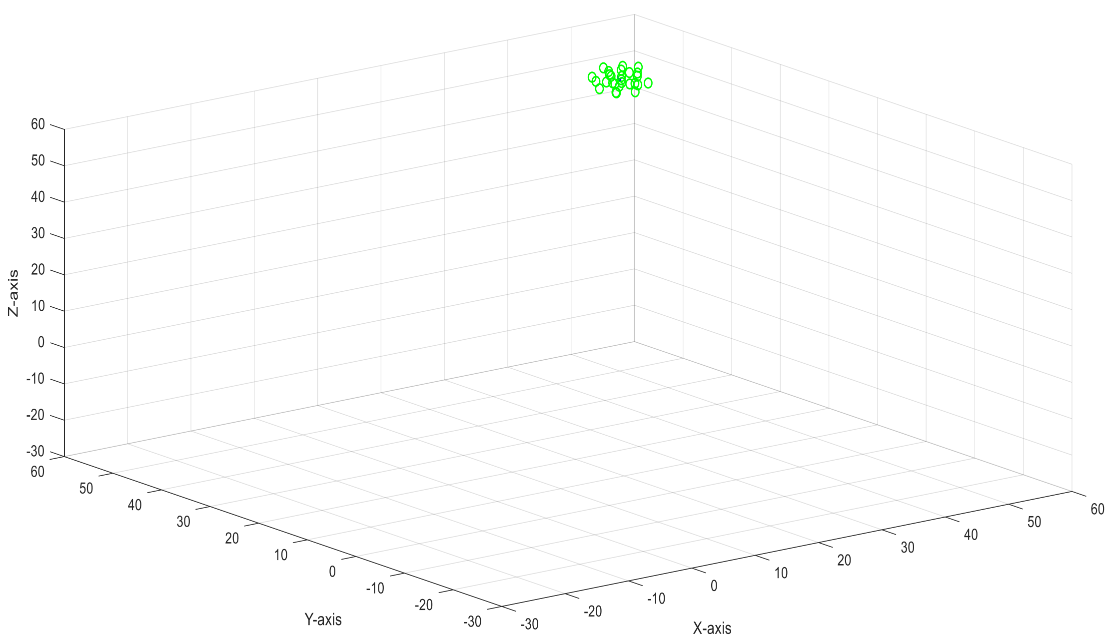

Figure 5.8 Foraging Behavior 6

5.2.2 Simulation with Enemy

In this section we will add the attacker, which is presented by the symbol □ , and we will make the attacker movement. The attacker is in the center of the preys or UAVs. The nodes in the beginning will run away from the attacker, because the attacker is entering to the danger zone r_p of the UAVs after that they will try to reunion with their near other group in their way to the target.

Sometimes the attacker tries to catch the nearest prey, but they increase their speed up to reach to the food or target as shown in the figures below

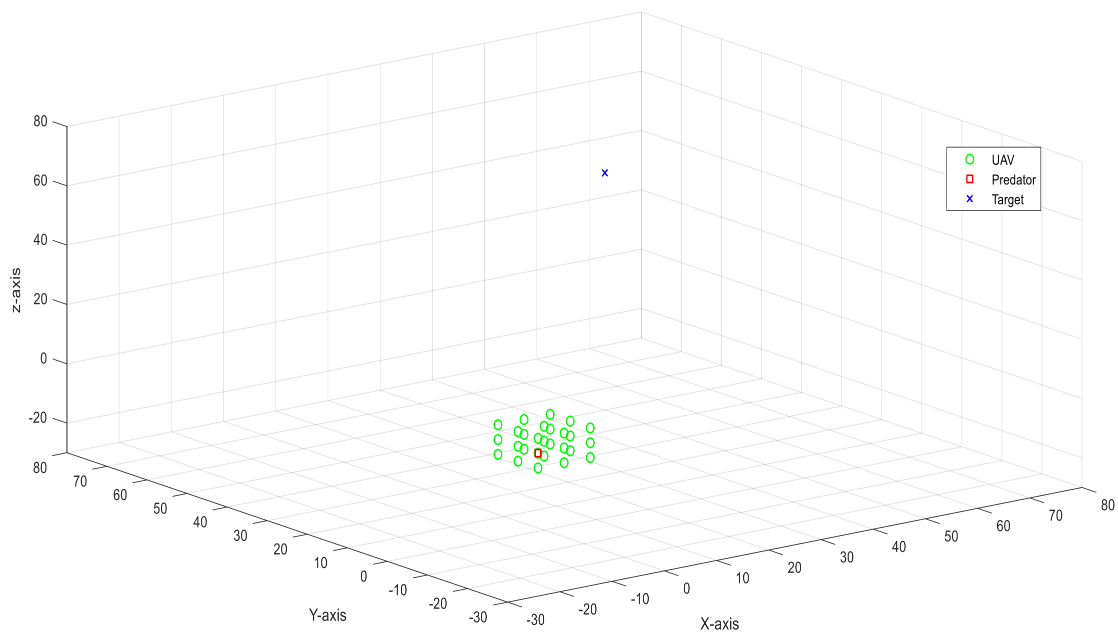

Figure 5.9 Foraging Behavior with Evasion 1

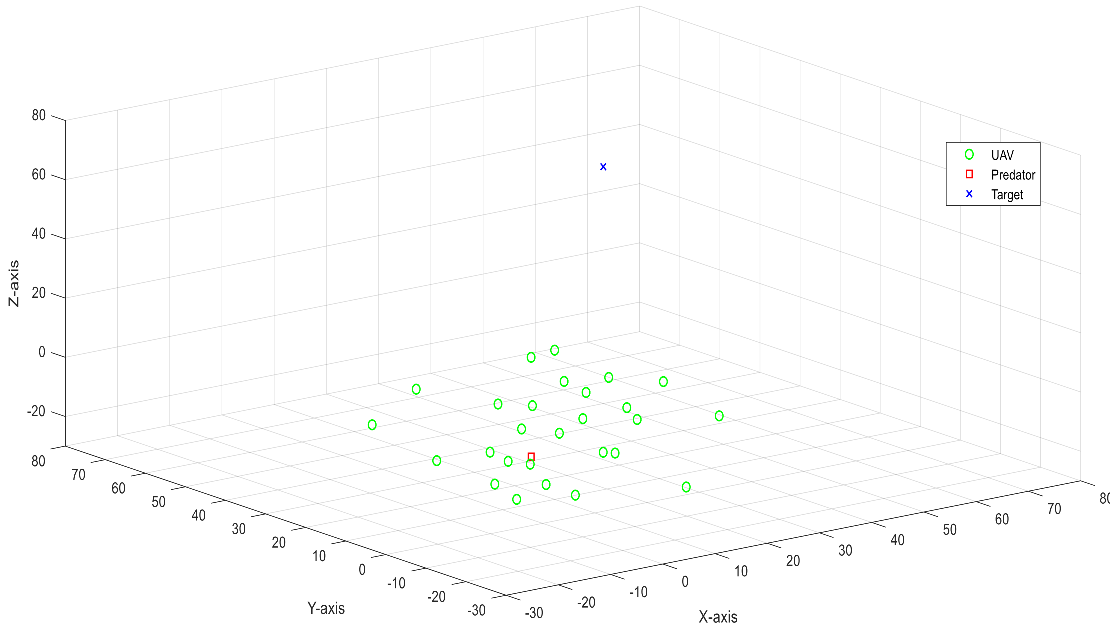

Figure 5.10 Foraging Behavior with Evasion 2

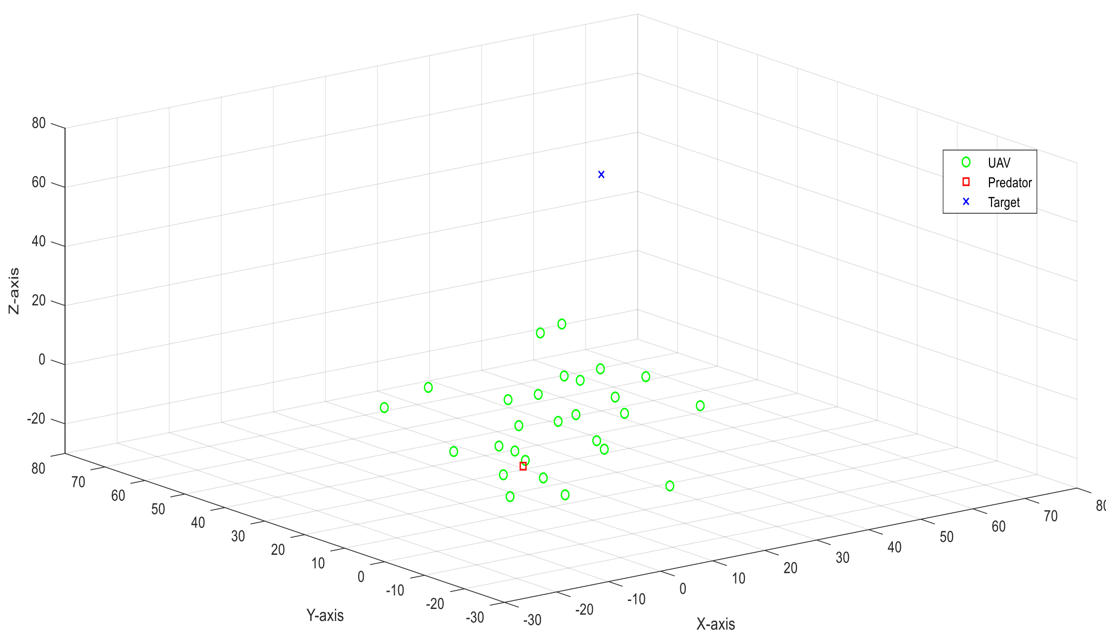

Figure 5.11 Foraging Behavior with Evasion 3

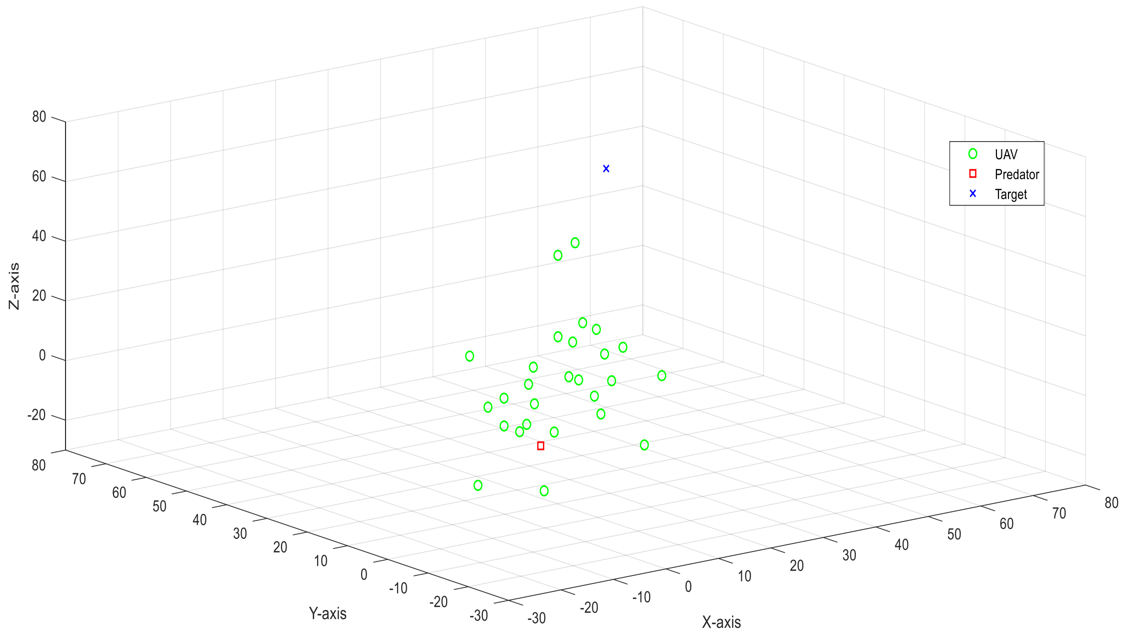

Figure 5.12 Foraging Behavior with Evasion 4

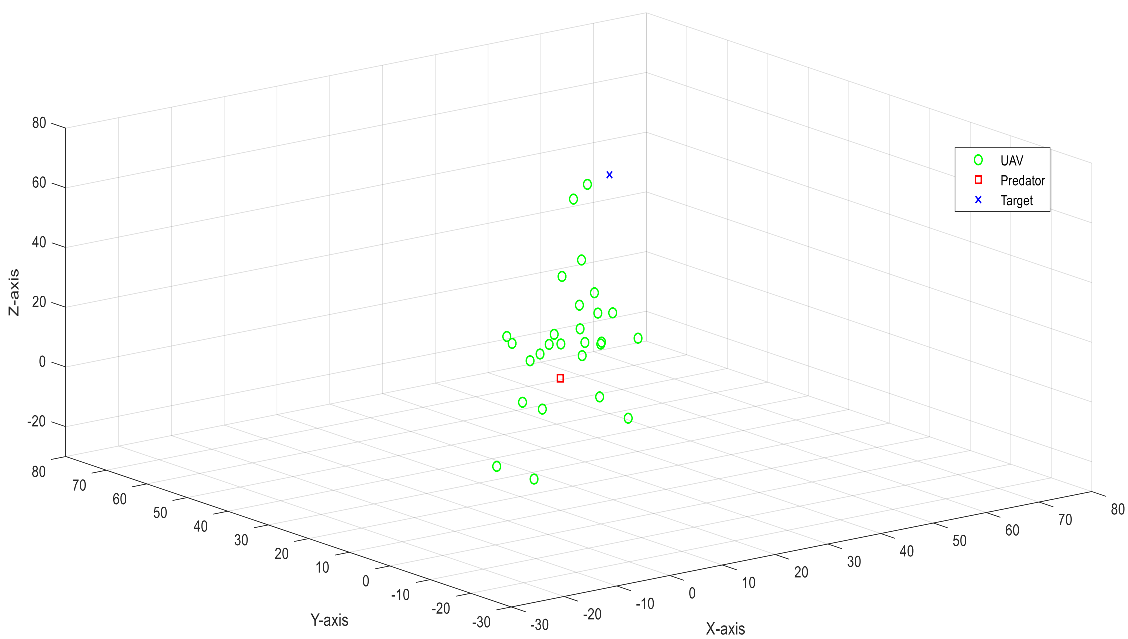

Figure 5.13 Foraging Behavior with Evasion 5

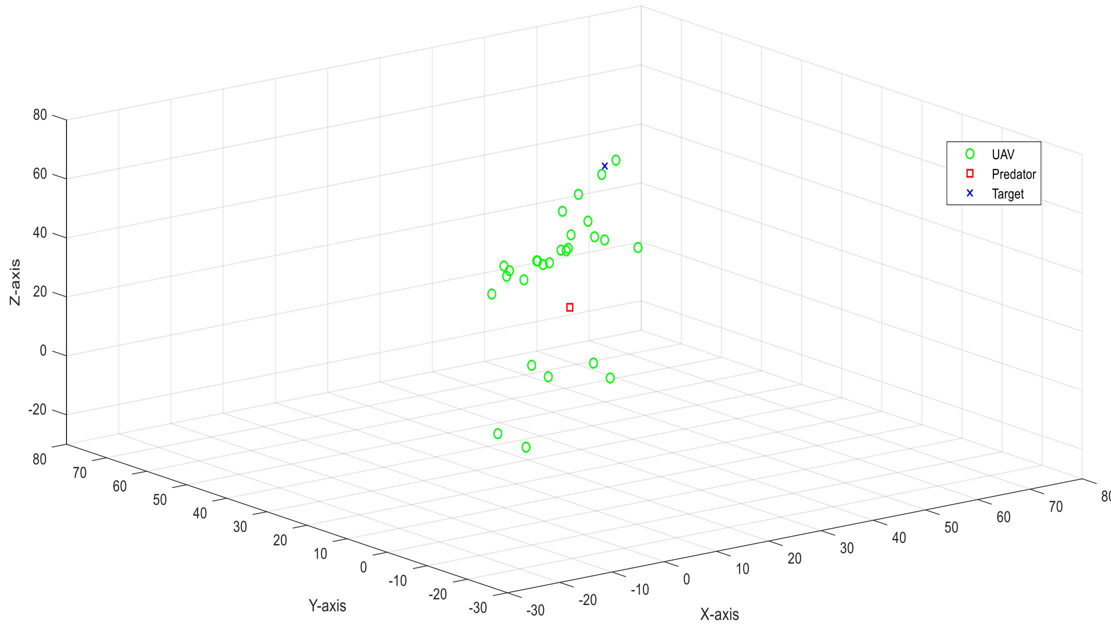

Figure 5.14 Foraging Behavior with Evasion 6

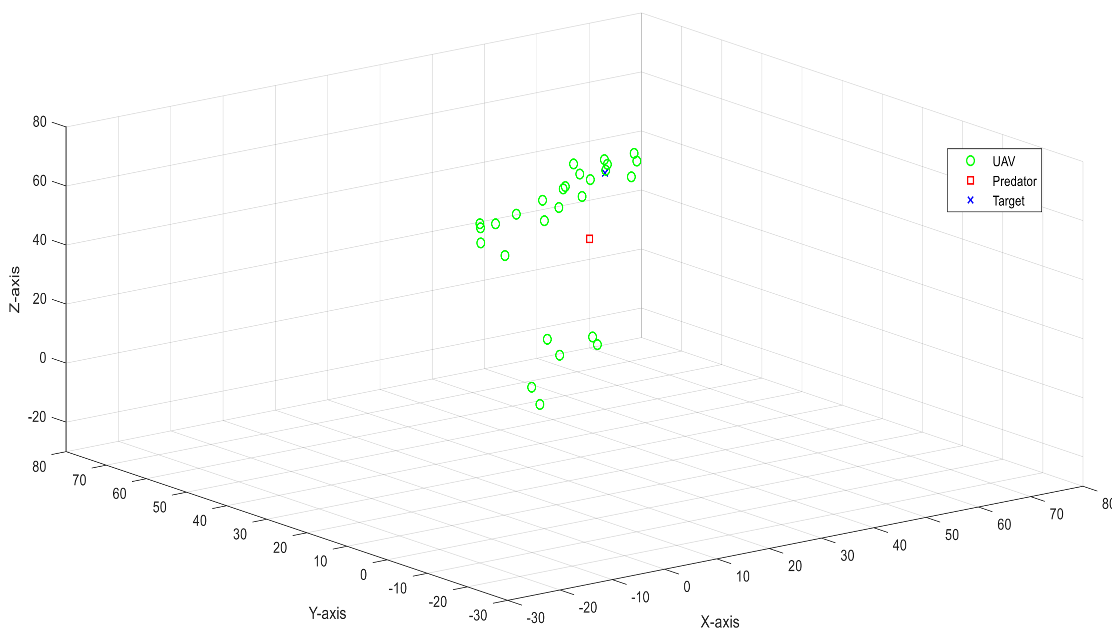

Figure 5.15 Foraging Behavior with Evasion 7

5.2.3 Simulation with Not Very Close Enemy

In this case the attacker is located far away from the nodes. The initial location of the attacker is (-20, -20, -20). In this case, as we see from the figures below, the attacker chases the nodes and try to catch the nearest one, but the nodes also try to evade this attacker.

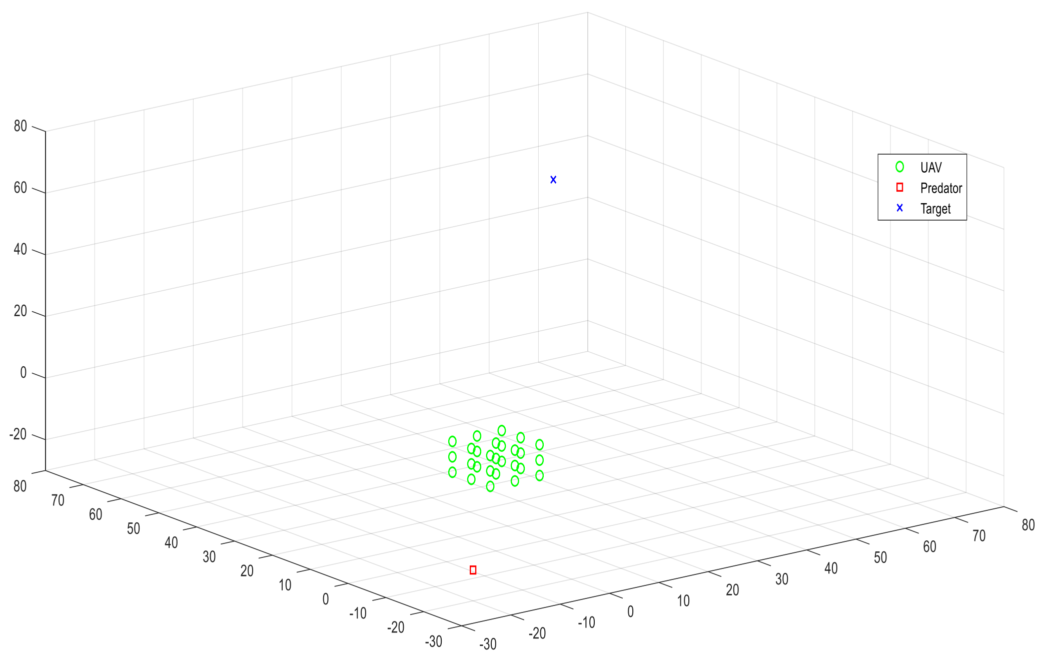

Figure 5.16 Foraging Behavior with Evasion the Chasing Attacker 1

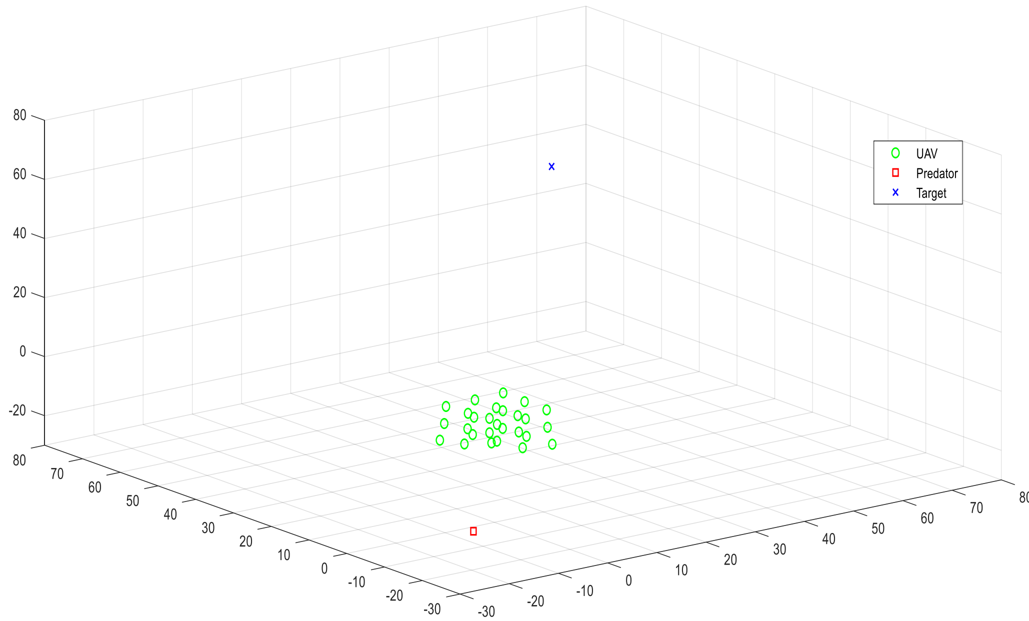

Figure 5.17 Foraging Behavior with Evasion the Chasing Attacker 2

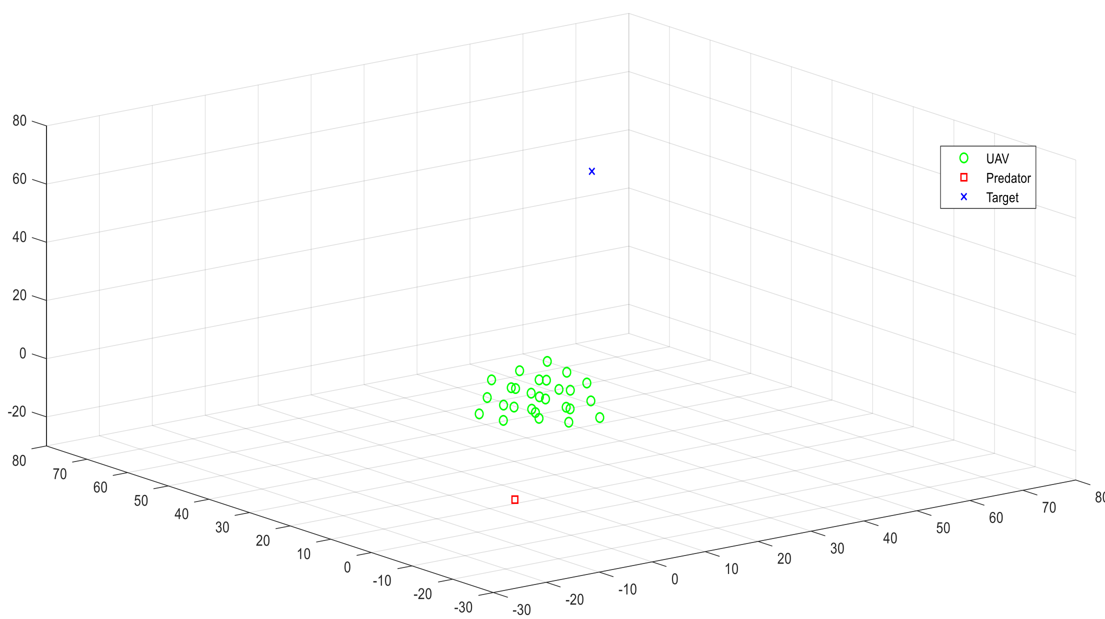

Figure 5.18 Foraging Behavior with Evasion the Chasing Attacker 3

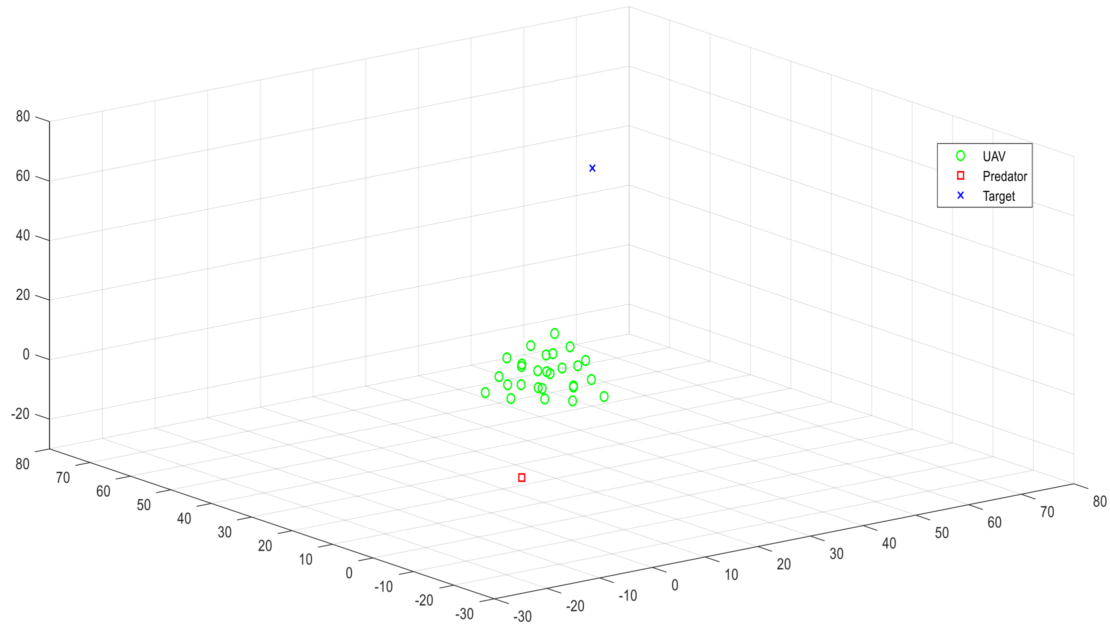

Figure 5.19 Foraging Behavior with Evasion the Chasing Attacker 4

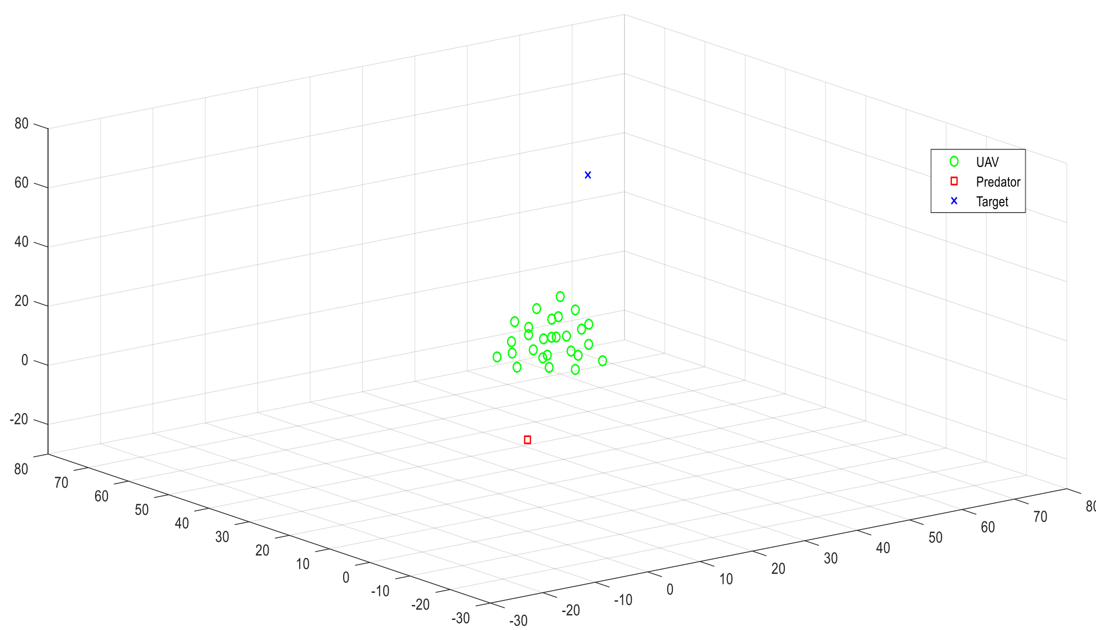

Figure 5.20 Foraging Behavior with Evasion the Chasing Attacker 5

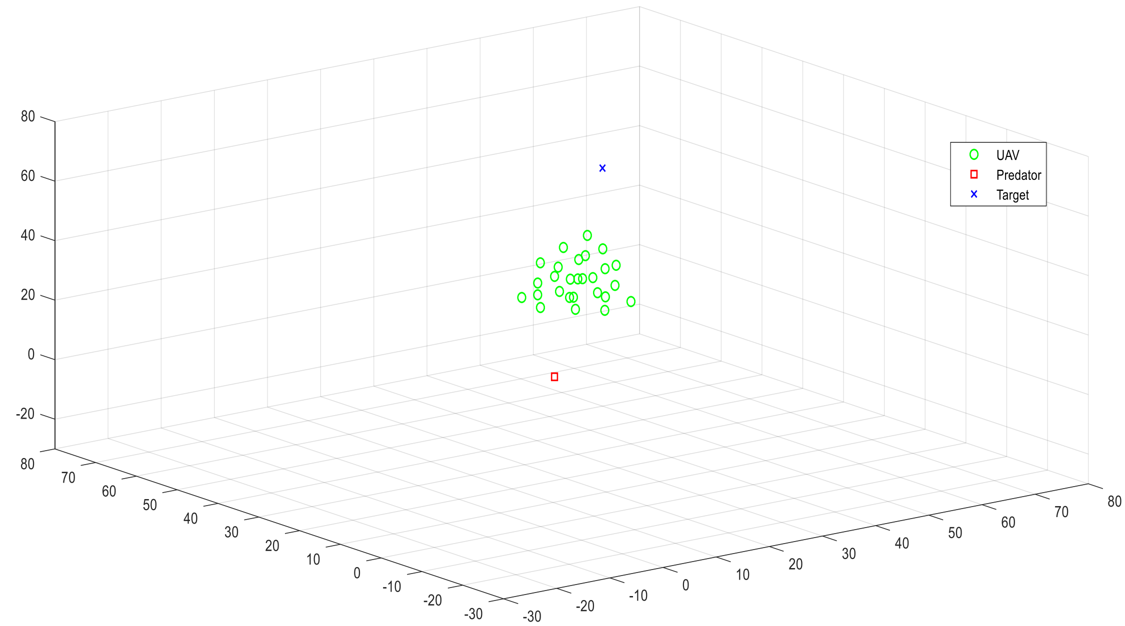

Figure 5.21 Foraging Behavior with Evasion the Chasing Attacker 6

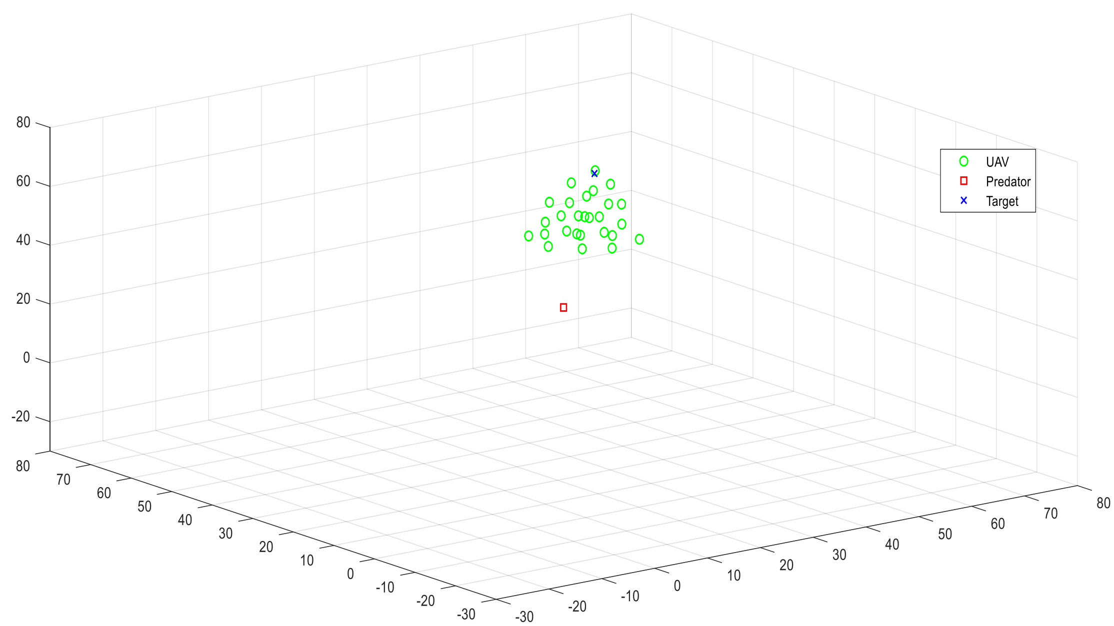

Figure 5.22 Foraging Behavior with Evasion the Chasing Attacker 7

CHAPTER 6

CONCLUSION AND FUTURE STUDY

In this research we presented two things. On one hand, vehicle control and biological inspired technology. The vehicle control system for quadrotor UAV depend on two combine control strategies. These controllers' strategies are PID controller and feedback linearization controller. A PID controller is used in the inner-loop while feedback linearization is used in the outer-loop. We focus here to control the velocity and the angular velocity of the UAV A last simulation presented here covers the control of a given desired velocity vector. Therefore, we assume that the quadrotor starts in the hovering state (i.e. all velocities are zero). The control task now comprises the stabilization of a desired velocity vector with $x_d = y_d = z_d = 0.4$ and to generate a linear movement. The desired state is achieved after a short transition phase and the quadrotor is moving with constant velocity. During that constant flight the angles are also kept constant and hence the angular rates are all zero after the initial transition.

For future work there are many control strategies that can be done. Intelligence control or and other controller strategies instead of what we use in this research can be used to get excellent result with faster response.

On the other hand, the foraging of multiple UAVs is presented in this research. The main goal of these UAVs is to reach to the target. While they are foraging, they faced attack from the enemy. As result, they try to evade this attack and complete their mission. We use the biological inspired technology to mimic the school of fish for foraging to go to the food and for evading the attacker.

We simulate this in 3-dimension and using the network so that each UAV communicate with its neighbors. Also, we are using the ATC for distrusted manner.

In this research the stationary target is used, so we recommended to use the mobile target, in addition we recommended to add some obstacles in the environment to add more difficulties. We recommended also to use the controller to get a good response and to ensure the tracking of the UAV.

References

[1] Melito T., "*Agencies Could Improve Information Sharing and End-Use Monitoring on Unmanned Aerial Vehicle Exports*," United States Government Accountability Office, Washington, United States, Tech. Rep., July 2012.

[2] Misener P., "*FAA: Amazon Petition for Exemption*," Amazon, United States, Tech. Rep., July 2014.

[3] Brockett, Roger W., Richard S. Millman, and Hector J. Sussmann, eds. "*Differential geometric control theory*" proceedings of the conference held at Michigan Technological University, June 28-July 2, 1982. Vol. 27. Birkhauser, 1983.

[4] Sampei, Mitsuji, et al. "*Necessary and sufficient conditions for transformation of nonholonomic system into time-state control form*." Proceedings of 35th IEEE Conference on Decision and Control. Vol. 4. IEEE, 1996.

[5] Kolmanovsky I. and N. McClamroch, "*Developments in Nonholonomic Control Problems*," IEEE Control Systems, vol. 15, no. 6, pp. 20–36, 1995.

[6] Tian Y.-P. and S. Li, "*Exponential Stabilization of Nonholonomic Dynamic Systems by Smooth Time-Varying Control*," Automatica, vol. 38, no. 7, pp. 1139–1146, 2002.

[7] Sampei, Mitsuji. *"A control strategy for a class of nonholonomic systems-time-state control form and its application."* Proceedings of 1994 33rd IEEE Conference on Decision and Control. Vol. 2. IEEE, 1994.

[8] Valtolina E. and A. Astolfi, "*Local Robust Regulation of Chained Systems*," Systems and Control Letters, vol. 49, no. 3, pp. 231–238, 2003.

[9] Bloch, A., and S. Drakunov. *"Tracking in nonholonomic dynamic systems via sliding modes*." Proceedings of 1995 34th IEEE Conference on Decision and Control. Vol. 3. IEEE, 1995.

[10] Whittlesey R. W., S. Liska, and J. O. Dabiri, "*Fish Schooling As A Basis for Vertical Axis Wind Turbine Farm Design*," Bioinspiration & Biomimetics, vol. 5, no. 3, p. 035005, Sep. 2010. [Online]. Available: http://iopscience.iop.org/1748-3190/5/3/035005

[11] Chang, Young Hwan, Claire Tomlin, and Karl Hedrick. *"Biologically-inspired coordination of multiple UAVs using sliding mode control."* Proceedings of the 2011 American Control Conference. IEEE, 2011.

[12] Yang, Liang, and Jianying Yang. *"Stabilization for a class of nonholonomic perturbed systems via robust adaptive sliding mode control."* Proceedings of the 2010 American Control Conference. IEEE, 2010.

[13] Yuan H., "*Control of Nonholonomic Systems*," Ph.D. dissertation, University of Central Florida, Florida – United States, 2009.

[14] Qiang Z., L. Zengbo, and C. Yao, "*A Backstepping Based Trajectory Tracking Controller for A Non-Chained Nonholonomic Spherical Robot*," Chinese Journal of Aeronautics, vol. 21, no. 5, pp. 472–480, Oct. 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1000936108600618.

[15] Chwa D., "*Sliding-Mode Tracking Control of Nonholonomic Wheeled Mobile Robots In Polar Coordinates*," IEEE Transactions on Control Systems Technology, vol. 12, no. 4, pp. 637–644, 2004.

[16] Matsuno, Fumitoshi, and Junji Tsurusaki. *"Chained form transformation algorithm for a class of 3-states and 2-inputs nonholonomic systems and attitude control of a space robot."* Proceedings of the 38th IEEE Conference on Decision and Control (Cat. No. 99CH36304). Vol. 3. IEEE, 1999.

[17] Astolfi A., "*Exponential Stabilization of A Wheeled Mobile Robot Via Discontinuous Control*," Journal of Dynamic Systems, Measurement, and Control,
vol. 121, no. 1, pp. 121–126, 1999.

[18] Luca A. D., G. Oriolo, and C. Samson, *Robot Motion Planning and Control*, ser. Lecture Notes in Control and Information Sciences. LAAS-CNRS, Toulouse: Springer Berlin Heidelberg, ch. Feedback control of a nonholonomic car-like robot, 1998.

[19] Laumond J. P., S. Sekhavat, and F. Lamiraux, *Robot Motion Planning and Control*, ser. Lecture Notes in Control and Information Sciences. LAASCNRS, Toulouse: Springer Berlin Heidelberg, ch. Guidelines in nonholonomic motion planning for mobile robots, 1998.

[20] Tau J., N. Xi, and W. Kang, "*Tracking Control of Nonholonomic Mobile Robots*," in 1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation, pp. 77–82, 1999.

[21] Hu, Eric, Simon X. Yang, and Ning Xi. *"A novel non-time based tracking controller for nonholonomic mobile robots."* Proceedings 2001 IEEE International Symposium on Computational Intelligence in Robotics and Automation (Cat. No. 01EX515). IEEE, 2001.

[22] Tian, Yu-Ping, and Ke-Cai Cao. *"An LMI design of tracking controllers for nonholonomic chained-form system*." 2007 American Control Conference. IEEE, 2007.

[23] Cao K.-C., "*Global K-Exponential Trackers for Nonholonomic Chained-Form Systems Based On LMI*," Intern. J. Syst. Sci., vol. 42, no. 12, pp. 1981–1992, 2011.

[24] Kanayama, Yutaka J., and Fariba Fahroo. *"A new line tracking method for nonholonomic vehicles."* Proceedings of International Conference on Robotics and Automation. Vol. 4. IEEE, 1997.

[25] MacKunis W., N. Gans, A. Parikh, and W. E. Dixon, "*Unified Tracking and Regulation Visual Servo Control for Wheeled Mobile Robots*," Asian Journal of Control, vol. 16, no. 3, pp. 669–678, May 2014. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1002/asjc.826/abstract

[26] Ou M., S. Li, and C. Wang, "*Finite-Time Tracking Control for Nonholonomic Mobile Robots Based on Visual Servoing*," Asian Journal of Control, vol. 16, no. 3, pp. 679–691, May 2014. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1002/asjc.773/abstract.

[27] Keighobadi J. and M. B. Menhaj, "*From Nonlinear to Fuzzy Approaches in Trajectory Tracking Control of Wheeled Mobile Robots*," Asian Journal of Control, vol. 14, no. 4, pp. 960–973, Jul. 2012. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1002/asjc.480/abstract.

[28] Hespanha J. P., D. Liberzon, and A. S. Morse, "*Logic-Based Switching Control of A Nonholonomic System with Parametric Modeling Uncertainty*," Systems and Control Letters, vol. 38, no. 3, pp. 167–177, 1999.

[29] Campion, Guy, Brigitte d'Andrea-Novel, and Georges Bastin. *"Modelling and state feedback control of nonholonomic mechanical systems."* [1991] Proceedings of the 30th IEEE Conference on Decision and Control. IEEE, 1991.

[30] Tsuchiya K., T. Urakubo, and K. Tsujita, "*Motion Control of A Nonholonomic System Based on the Lyapunov Control Method*," Journal of Guidance, Control, and Dynamics, vol. 25, no. 2, pp. 285–290, 2002.

[31] Camicia, Carlo, Fabio Conticelli, and Antonio Bicchi. *"Nonholonomic kinematics and dynamics of the sphericle."* Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)(Cat. No. 00CH37113). Vol. 1. IEEE, 2000.

[32] Camazine, Scott, et al. *"Self-organization in biological systems"*. Princeton university press, 2003.

[33] Couzin I. D., "*Collective Cognition in Animal Groups*," Trends in Cognitive Sciences, vol. 13, no. 1, pp. 36–43, 2009.

[34] Haque M. A., A. R. Rahmani, and M. B. Egerstedt, "*Biologically Inspired Confinement of Multi-Robot Systems*," International Journal of Bio-Inspired Computation, vol. 3, no. 4, pp. 213–224, 2011.

[35] Park H.-S., N.-H. Tran, and J.-W. Park, "*Biologically Inspired Techniques for Autonomous Shop Floor Control*," in New Technologies - Trends, Innovations and Research, C. Volosencu, Ed. South Korea: InTech, 2012.

[36] ˇaSkalski G. T. and J. F.ˇaGilliam, "*Feeding Under Predation Hazard: Testing Models of Adaptive Behavior with Stream Fish*," The American Naturalist, vol. 160, no. 2, pp. 158–172, 2002.

[37] Qin Li Z.-P. J., "*Flocking of Decentralized Multi-Agent Systems with Application to Nonholonomic Multi-Robots*," Proceedings of the 17th World Congress the International Federation of Automatic Control, pp. 9344–9349, 2008.

[38] Q. Li and Z.-P. Jiang, "*Diffusion Strategies for Adaptation and Learning Over Networks: An Examination of Distributed Strategies and Network Behavior*," Kybernetika, vol. 45, no. 1, pp. 84–100, 2009.

[39] Olfati-Saber, Reza. "*Flocking for Multi-Agent Dynamic Systems: Algorithms and Theory*." IEEE Transactions on automatic control 51.3: 401-420 (2006).

[40] Tu, Sheng-Yuan, and Ali H. Sayed. *"Mobile adaptive networks."* IEEE Journal of Selected Topics in Signal Processing 5.4: 649-664 (2011).

[41] Cattivelli F. and A. Sayed, "*Modeling Bird Flight Formations Using Diffusion Adaptation*," IEEE Transactions on Signal Processing, vol. 59, no. 5, pp. 2038–2051, 2011.

[42] Tu, Sheng-Yuan, and Ali H. Sayed. *"Tracking behavior of mobile adaptive networks."* 2010 Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers. IEEE, 2010.

[43] Li, Jinchao, Sheng-Yuan Tu, and Ali H. Sayed. *"Honeybee swarming behavior using diffusion adaptation."* 2011 Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE). IEEE, 2011.

[44] Sayed A., "*Adaptive Networks*," Proceedings of the IEEE, vol. 102, no. 4, pp. 460–497,2004.

[45] Sayed A., 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.

[46] Lopes C. and A. Sayed, "*Diffusion Least-Mean-Squares Over Adaptive Networks*," IEEE Transactions on Signal Processing, vol. 56, no. 7, pp. 3122–3136, 2008.

[47] Lopes, Cassio G., and Ali H. Sayed. *"Diffusion adaptive networks with changing topologies."* 2008 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2008.

[48] Takahashi N., I. Yamada, and A. Sayed, "*Diffusion Least-Mean-Squares with Adaptive Combiners: Formulation and Performance Analysis*," IEEE Transactions on Signal Processing, vol. 58, no. 9, pp. 4795–4810, 2010.

[49] Cattivelli F. and A. Sayed, "*Diffusion LMS strategies for distributed estimation*," IEEE Transactions on Signal Processing, vol. 58, no. 3, pp. 1035–1048, 2010.

[50] Rastegarnia, Amir, Azam Khalili, and Md Kafiul Islam. *"An Improved Self-Organizing Diffusion Mobile Adaptive Network for Pursuing a Target."* arXiv preprint arXiv:1603.08543 (2016).

[51] Pak, Amin Lotfzad, et al. *"A distributed target localization algorithm for mobile adaptive networks."* ECTI Transactions on Electrical Engineering, Electronics, and Communications 14.2: 47-56 (2016).

[52] Tu, Sheng-Yuan, and Ali H. Sayed. *"Cooperative prey herding based on diffusion adaptation."* 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2011.

[53] Tu, Sheng-Yuan, and Ali H. Sayed. *"Mobile adaptive networks with self-organization abilities."* 2010 7th International Symposium on Wireless Communication Systems. IEEE, 2010.

[54] Tu, Sheng-Yuan, and Ali H. Sayed. *"Foraging behavior of fish schools via diffusion adaptation."* 2010 2nd International Workshop on Cognitive Information Processing. IEEE, 2010.

[55] Lin, May Zar, Manohar N. Murthi, and Kamal Premaratne. *"Mobile adaptive networks for pursuing multiple targets."* 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2015.

[56] Huang, Zongrui, Daqi Zhu, and Bing Sun. *"A multi-AUV cooperative hunting method in 3-D underwater environment with obstacle."* Engineering Applications of Artificial Intelligence 50: 192-200 (2016).

[57] El Ferik, Sami, and Olapido Raphael Thompson. *"Biologically inspired control of a fleet of uavs with threat evasion strategy*." Asian Journal of Control 18.6: 2283-2300 (2016).

[58] Manzoor, Sajjad, Sungon Lee, and Youngjin Choi. *"A coordinated navigation strategy for multi-robots to capture a target moving with unknown speed."* Journal of Intelligent & Robotic Systems 87.3-4: 627-641 (2017).

[59] Voos, Holger. *"Nonlinear control of a quadrotor micro-UAV using feedback-linearization."* 2009 IEEE International Conference on Mechatronics. IEEE, 2009.

[60] Voos H., "*Nonlinear and Neural Network-based Control of a Small Four-Rotor Aerial Robot*," in Proc. of the IEEEIASME Int. Conference on Advanced Intelligent Mechatronics, Zurich, CH, 2007.

[61] Couzin I. D., "*Collective Cognition in Animal Groups*," Trends in Cognitive Sciences, vol. 13, pp. 36-43, Jan. 2009.

[62] Albayrak, Okay. *"Line and Circle Formation of Distributed Autonomous Mobile Robots with Limited Sensor Range".* Naval Postgraduate School Monterey CA, 1996.

Vitae

Personal Information:

- Name: Mohammed Salem Saleh Mahdami
- Nationality: Yemeni
- Address: Alhawtah – Hadhramout – Republic of Yemen
- Email: mahdamy2003@gmail.com
- Tel No.: Mobile: (+967) 773880961 / 712919775
- Home Phone: (+967) – 5 – 427714
- Date of Birth: 13th February 1986
- Place of Birth: Hadhramout – Yemen
- Nationality: Yemen
- Marital status: married

Education:

- MSc (Systems and Control Eng.), 2019
  - College of computer science and engineering
  - King Fahd University of petrol and minerals (KFUPM)
  - Dhahran, Saudia Arabia
- BSc (Electronics and Communications), 2012
  - College of Engineering
  - Hadhramout University of Science and Technology
  - Al-Mukalla, Hadhramout, Republic of Yemen

Training Courses:

- Courses in Microsoft Office 2010 (IT – Windows7 – word – Power point – Excel – Internet – Access) during the year of 2012 provided by the Benevolent Fund for Outstanding Students, Al-Mukalla, Hadhramout, Republic of Yemen.
- Course in English language " preparing of TOEFL " (January – March 2012) provided by the Benevolent Fund for Outstanding Students, Al-Mukalla, Hadhramout, Republic of Yemen.
- Courses in Human Development (October – November 2011) provided by the Benevolent Fund for Outstanding Students, Al-Mukalla, Hadhramout, Republic of Yemen.
- Transmission Course (PCM – PDH – SDH) during the month of July 2010 at the General Telecommunication Institute (GTI), Sana’a.

- Summer training at Yemen Mobile Corporation in Sana'a on July to August 2010.
- Summer training at Seiyon Radio Transmission Station on July 2009.
- Summer training at the Hadhramout Power Plant (General Foundation for Power) on July to August 2009.
- Guidance Skills course at Hadhramout University on 2008.
- First Aid course at Hadhramout University on 2008.

Skills:

- Language Skills
    - Arabic – native
    - English – (working knowledge)
- Computer software programs such as Microsoft Office (word – PowerPoint – excel), the internet and matlab
- Ability to work under pressure
- Ability to work with work group