CONTROL OF MULTIPLE NONHOLONOMIC UNMANNED AERIAL VEHICLES IN A BIOLOGICALLY INSPIRED ADAPTIVE NETWORK

BY

THOMPSON OLADIPO RAPHAEL

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

MAY 2015

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **THOMPSON OLADIPO RAPHAEL** under the direction of his thesis adviser and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN SYSTEMS AND CONTROL ENGINEERING**.

Thesis Committee

Dr. Sami El Ferik (Adviser)

Dr. Salim Ibrir (Member)

Dr. Fouad Al-Sunni (Member)

Dr. Moustafa Elshafei (Member)

Dr. Uthman Baroudi (Member)

Dr. Adel Ahmed
Department Chairman

Dr. Salam A. Zummo
Dean of Graduate Studies

13/4/15
Date

To God, my family and booski...

ACKNOWLEDGMENTS

I would like to say a special thank you to my thesis advisor, Dr. Sami El Ferik, for his guidance, efforts and encouragement throughout the period spent carrying out this research. You have been taught me to push the limits of knowledge and reasoning, and also to believe in myself and my capabilities. Thank you.

Many thanks to my committee members Dr. Fouad Al-Sunni, Dr. Salim Ibrir, Dr. Moustafa Elshafei, and Dr. Uthman Baroudi for agreeing to be on my thesis committee and generously lending their expertise and precious time. Thank you sirs.

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

Symbol	**Description**
UAV	Unmanned Aerial Vehicle
GPS	Global Positioning System
GAO	Government Accountability Office
AUV	Autonomous Underwater Vehicle
LMI	Linear Matrix Inequality
Fig.	Figure
Eqn.	Equation
Chap.	Chapter
MAS	Multi-Agent System
VLSR	Very Large Scale Robotic
PSO	Particle Swarm Optimization
MILP	Mixed-Integer Linear Program
MPC	Model Predictive Control
ATC	Adapt-Then-Combine

THESIS ABSTRACT

NAME: Thompson Oladipo Raphael

TITLE OF STUDY: Control of Multiple Nonholonomic Unmanned Aerial Vehicles in a Biologically Inspired Adaptive Network

MAJOR FIELD: Systems and Control Engineering

DATE OF DEGREE: April 13, 2015

In this thesis, navigation algorithms for a fleet of nonholonomic UAVs capable of evading a chasing predator and also pursuing a desired target are proposed. The proposed navigation algorithms are used to define path planning trajectories which are tracked by a designed nonlinear backstepping controller. The navigation algorithms are designed to switch path planning strategies based on the prevailing precarious environmental conditions. We implement the group of nonholonomic UAVs in an adaptive network, specifically inspired by the relationship between a school of fish and a predator. The navigation algorithms are thus integrated with a group of nonholonomic UAV models such that the UAVs exhibit the biological behaviors when on a mission. This approach approximately simulates an air combat field. To put this in context, in this

thesis, we aim to use a biologically inspired algorithm along with a designed controller to achieve both target pursuance and effective evasion from a predator. This is equivalent to having a fleet of UAVs on the same mission of attacking a target, while also aware of a predator on pursuit. The UAVs aim to maneuver and evade the predator while also coordinating their movement and behaviors in a cooperative and coherent manner. Analyses of the system dynamics show that the proposed nonlinear tracking control approach guarantee asymptotic stability for the desired navigation paths. Simulations are also carried out to show the performance of the approach in both normal and attack evasion mode.

مستخلص الرسالة

الإسم: طومسون أولاديبو رافييل

عنوان الرسالة: التحكم في الطائرات الآلية متعددة النونهولونوميك في شبكة تكيفية مستوحاه بيولوجيا

التخصص: قسم هندسة النظم

التاريخ: فبراير، 2015

فى هذة الأطروحة لوغاريتمات تصفحية لغواصات عابرة معتمدة على المسار لديها القدرة على الهروب من مطاردة المفترس والأستمرار للوصول للهدف. اللوغاريتمات التصفحية المفترحة تستخدم لتحديد مسارات مخططة بحيث يتم تصميم وتحديد المسار بتحكم الخطوة الرجعى الغير خطى. اللوغاريتمات التصفحية مصممة لتحول مسار التخطيط بناء على ظروف البيئة المتقلبة. نحن ندمج مجموعة من الغواصات المعتمدة على المسار فى شبكة تأقلم مستوحاة من علاقة سرب أسماك وسمكة القرش المفترسة. لوغاريتم التصفح يتكامل مع مجموعة من الغواصات المعتمدة على المسار بحيث ان الغواصات تتسم بالسلوك الأحيائى فى وقت المهمة.

هذا الأسلوب يحاكى تقريبا مجال القتال فى الهواء. لوضع المشكلة فى سياقها نحن نهدف لأستخدام علم الأحياء المستوحى مع التحم المصمم للوصول للهدف وهروب فعال من المفترس. هذا معادل للحصول على غواصات عابرة فى نفس المهمة من مهاجمة هدف ما والهروب من هدف اخر. تهدف الغواصات للمراوغة والهروب من المفترس بينما تنسق حركاتها وسلوكها فى فكر متعاون ومتماسك. تحليل النظام الديناميكى يوضح ان اسلوب تحم المسار يضمن الأستقرار لمسارات التصح المرغوبة. النتائج نفذت لتوضح أداء الأسلوب فى شكل هجوم وهروب طبيعى.

CHAPTER 1

INTRODUCTION

Unmanned Aerial Vehicles (UAVs) are fast becoming a commonplace for military operations. Virtually every developed country with potent military capabilities, and interestingly some developing countries, have Unmanned Aerial Vehicles. The United States Government Accountability Office (GAO), in 2012, [1] released a report estimating that the number of countries in possession of UAVs had almost doubled in the preceding seven years. The unmanned aerial vehicle technology is also catching up with the commercial industry, as the big tech giants - for example Amazon: "Amazon Prime Air" [2] - plan to revolutionize their supply chain systems using unmanned aerial vehicles for home deliveries. The increasing use of UAVs, especially for military applications, continues to make the control design of UAVs an active area of research. We envision a future where military combat would be strictly carried out by unmanned vehicles; in other words, air supremacy would be determined by the UAVs with the best maneuvering and evading skills when engaged in combat.

Unmanned Aerial Vehicles belong to the class of nonholonomic systems. A common constraint attributed this class of models is their inability to be stabilized by a smooth static state feedback controller. This is particularly because such models can usually have a structure with three states and only two inputs to control the system, hence, fail to satisfy [3] [4] ***Brockett's Condition***. Over the years, several control strategies have been developed [5] to tackle this problem. Discontinuous feedback control laws [6] [7] and also continuous feedback laws [4] [8] [9] have been proposed.

Also notable is the use of biologically inspired algorithms based on the collective behavior of animal groups to simulate real world scenarios. These biologically behaviors have been found to be extremely useful in several engineering applications. 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 bird in flight, or the hunting techniques used by carnivores in the wild, include some of the behaviors actively being researched.

1.1 Problem Statement

Our review indicates that previous authors have used many control techniques for tracking nonholonomic UAV models. This is essentially for single UAV models. Scaling it up to a group of nonholonomic UAVs, other authors have considered cooperative control techniques that leverage on modern control theory to achieve desired navigation of a fleet of nonholonomic UAVs. However, to the best of our knowledge, no one is yet to consider integrating the biological behaviors of a school of fish, modeled as a strongly connected graph, with a group of nonholonomic UAVs for navigation purposes. Our integration is achieved using the biological behaviors to define navigation paths while nonlinear tracking controllers, distributed in nature, steer the UAVs in line with the biological behaviors for tracking purposes. In order to solve the stated problem, we carried out the following objectives.

1.2 Objectives

In this thesis, the following were achieved:

1. The design of navigation algorithms for a group of nonholonomic UAVs in flight, taking note of environmental conditions and aerodynamic constraints.

2. The integration of the fish-prey navigation algorithm and a group of nonholonomic UAV models in an adaptive network to mimic the biological foraging behavior of a school of fish.

3. The integration of the evasive behavior of fishes when they are in danger and a group of nonholonomic UAV models.

4. The design of nonlinear backstepping controllers that ensure the nonholonomic UAV model tracks the desired trajectories based on the navigation algorithms.

5. The design of a navigation control that ensures collision avoidance and keeps the UAVs a safe distance from each other.

6. The simulation of the nonholonomic UAV models applying the proposed navigation algorithm and controllers in different scenarios.

1.3 Approach

In this thesis, a distributed navigation algorithm for a group of UAVs capable of evading a chasing predator and also approaching a desired target is proposed. The navigation algorithm is designed to switch path planning strategies based on the prevailing precarious environmental conditions.

We define two environmental conditions: danger and no-danger conditions; defined by the presence or absence of an enemy UAV respectively. Under the no-danger conditions, the navigation technique selects an algorithm based on the foraging behaviors of a school of fish. The group of mobile UAVs move in an adaptive network, specifically inspired by the biological relationship previously stated. Under the danger conditions, the navigation algorithm switches to a cognitive evasive

technique [11] [12] that ensures the UAV under attack evades such danger which could be an enemy UAV (biologically representing a predator for example, a shark) or an approaching missile for military applications.

We propose a nonlinear tracking controller that ensures the paths planned by the navigation algorithm is tracked for either cases of danger and no-danger conditions. The nonlinear tracking controller is designed using backstepping control technique to guarantee the stability of the nonholonomic UAV on the predefined path.

The biologically inspired algorithms used for a fleet of mobile autonomous robots are used to define desired trajectories for the UAVs. The multi-agent UAV robot system has the cognitive abilities to evade an incoming enemy and also follow the bio-inspired navigation trajectory using a designed nonlinear tracking controller. The fleet is composed of nonholonomic homogeneous robots under the Pfaffian constraint and a local robust backstepping controller which is used to ensure path tracking in normal and attack modes respectively. The attacker has different dynamics and is also nonholonomic. The navigation and guidance system takes into account the state of each UAV, the target mission, and environmental conditions. The environmental conditions considered depend exclusively on the presence or absence of an enemy UAV as well as awareness of other friendly UAVs in its surrounding. Analysis of the system dynamics shows that the nonlinear control approach guarantees asymptotic stability. Extensive simulation results attest to the performance of the approach.

1.4 Organization

The rest of this thesis report is organized as follows: chapter (2) describes the literature review of the research carried out. Previous work that has been done on the control of nonholonomic systems is first discussed. Then, the foundation for the bio-inspired behaviors of the navigation algorithm is explored along with the research carried out on evasive techniques used by fighter jets. Finally, we conclude chapter (2) by discussing nonlinear tracking designs used for nonholonomic models. Chapter (3) describes the nonholonomic UAV model and its known constraints. It further elaborates on the controllability of the UAV model and describes the different UAV model forms for control purposes.

In chapter (4), we build on the foundation of the navigation algorithms. First, we discuss the fish-prey algorithm and its application to UAVs. The biological behaviors of foraging fishes are modeled and applied. We then design cognitive evasive strategies for the UAVs in the presence of danger.

The tracking problem is treated elaborately in chapter (5). Control strategies are designed to track the path planned by the navigation algorithm. Both single and multiple UAVs are analyzed to give clarity to the designs employed. Chapter (6) shows the simulations, results and discussions of the research carried out and finally, we conclude in chapter (7).

CHAPTER 2

LITERATURE REVIEW

2.1 Control of Nonholonomic Models

Car-like robots, autonomous underwater vehicles, two-driving wheel mobile robots, and hopping robots are also examples of nonholonomic models. Interestingly, these models are quite similar in structure. They are often represented in either kinematic or dynamic form. Kolmanovsky [5] described a general form for the kinematic model form:

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

where $u_i, i = 1, ..., m$, are control inputs, x are states of the nonholonomic system, defined in the subset of $\mathbb{R}$, $2 \leqslant m < n$, and $g_i, i = 1, ..., m$, are given vector spaces. Expanding (2.1) to the representative form of the unmanned aerial vehicle [13]

yields:

$$\begin{aligned} \dot{x_1} &= u_1 \cos\theta_s \\ \dot{x_2} &= u_1 \sin\theta_s \\ \dot{\theta_s} &= u_2 \end{aligned} \tag{2.2}$$

where (x_1, x_2) represent a location on a plane and u_1 and u_2 represent linear and angular velocity inputs respectively. This model is similar to the unicycle model shown in fig.(2.1).

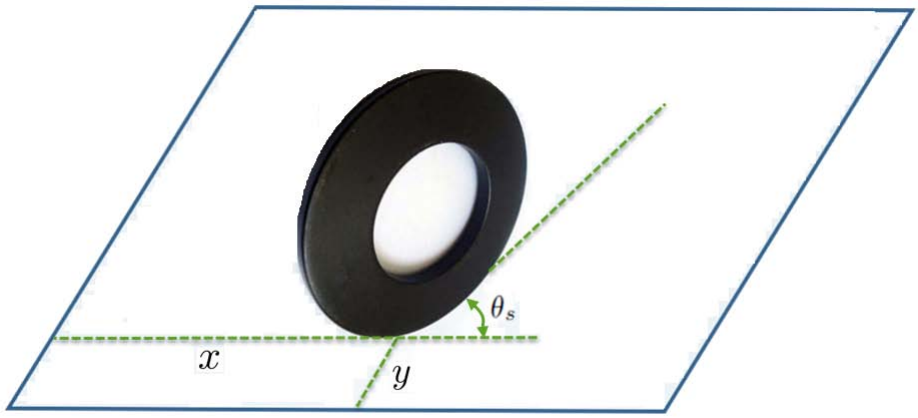

Figure 2.1: Unicycle model

The general dynamic model, defined [5] by the d'Alembert-Lagrange form, is also given by:

$$\begin{aligned} M(x)\ddot{x} + f(x,\dot{x}) &= C\lambda + B(x)\tau \\ J^T(x)\dot{x} &= 0 \end{aligned} \tag{2.3}$$

where $M(x)$ is the positive definite inertia matrix, $x = (x_1, ..., x_n)$ is a vector with

n coordinates, $J(x)$ a $n \times (n-m)$ matrix which is full rank, $B(x)\tau$ a vector force input to the system, $B(x)$ is a matrix whose dimension is $n \times p$ and τ a p-vector control, and λ a Lagrange multiplier vector with $(n-m)$ dimension.

In the design of controllers of this class of models, several techniques have been used in literature.

2.1.1 Robust Control

Yang et al. [14] discussed stabilizing the chained form of a class of perturbed nonholonomic systems. It involved a two stage stabilization process: a subsystem is first established by an adaption law and state feedback; after which the remaining subsystem is also transformed using a coordinate system. A Linear Matrix Inequality (LMI) technique using a robust sliding surface that ensures robustness is then applied to achieve stabilization. The case of local robustness of nonholonomic systems is handled in [6]. A discontinuous control design achieving global asymptotic stability under minor external perturbation is proposed.

2.1.2 Backstepping Controller

Yuan in [15] discusses the use of a Lyapunov-based control in finding an exponential control that ensures stabilization of an autonomous underwater vehicle (AUV), another nonholonomic system. In [16], the authors proposed a backstepping control technique for a spherical robot for use in an unmanned terrain. The backstepping control technique showed asymptotic tracking convergence to the

desired trajectory and was also verified using simulations.

2.1.3 Sliding Mode Control

Researchers have also considered sliding mode control to achieve tracking goals for nonholonomic systems. [7] designed a non-smooth feedback using sliding mode control to track the desired state function of a nonholonomic system. In the case of [17], Chwa combined both the dynamic and kinematic nonholonomic models to design two controllers based on sliding mode technique. By using the polar coordinate form of the kinematic model, he was able to eliminate constraints on the reference velocities. This technique achieved tracking except at an arbitrary small area of the origin. The chattering of the control signal which is common with this technique is also an area of concern.

2.1.4 Time-State Control Form

Time-State Control form presents a way to restructure nonholonomic systems that cannot be represented in chained form. It was presented by Sampei in [9]. This form allows one to overcome the controllability issue associated with nonholonomic systems; however, it requires input switching which, in some cases, may disrupt the stability of the system. The sufficient and necessary conditions for restructuring a system to the time-state form were discussed again by Sampei et al. in [4].

2.1.5 State Transformation and Feedback Linearization

Many authors combined several techniques in dealing with nonholonomic systems. While a lot of them used several state transformation methods first before applying another control technique, others used other techniques first, then applied feedback linearization. A number of them combined both state transformation and feedback linearization.

In [18], Matsuno et al. specifically described an algorithm that converts a class of two inputs and three states nonholonomic systems to their chained forms. This transformation has proved very useful for control analysis of this class of systems. Astolfi [19] on the other hand designed a bounded and discontinuous state feedback control law that ensured exponential convergence. He also took note of measurement errors in his analysis. Luca et al. [20] and Laumond et al. [21] provided elaborate approaches on these subjects. The authors of [20] also provide state transformations that restructure the kinematic models into a chained nonholonomic form. Using the chained form, different feedback techniques were applied to track arbitrary trajectories or stabilize the nonholonomic system.

The authors achieved the control of the nonholonomic system at first by approximate feedback linearization. This technique achieves local asymptotic stability of the chained form system; however, this region of stability was shown to be rather large. The downside though is that the transient response of the system may

degenerate in an unsatisfactory way.

Exact feedback linearization is then used in [20] and also in [8]. Luca et al. considered the possibility of using *input-output feedback linearization* which might compromise some states or use *full-state feedback linearization*. In the first case, the system output is redefined to overcome the problem of matrix singularity, so that a *static feedback* can be designed for the system. The later however includes an integrator to solve the issue of matrix singularity and also adds an auxiliary input to make the system conform.

2.1.6 Non-time Based Tracking Control

This is an interesting technique which avoids the use of time in defining the reference trajectory for a nonholonomic system. Tau et al. [22] proposed an event based controller using a non-time reference to track a desired trajectory. A state to reference projection is used to transform the time based controller to a non-time based controller. Hu et al. [23] however combined the traditional non-time based controller with a biologically inspired additive model. This addition resolves the issue of discontinuous control common with nonholonomic systems and also eliminates the tracking error that exists in [22].

2.1.7 Tracking Control

The tracking problem for nonholonomic systems has been treated by several authors, using different control strategies to investigate it, some of which have been

discussed above. Tian et al. [24] converted the tracking challenge into a two-system stabilization problem by using a transformation and a cascade technique. A Linear Matrix Inequality (LMI) was then designed to stabilize, in other words, track the given reference. In 2010, Cao [25] provided an improvement on [24]. The update allowed for the exponential convergence to a desired trajectory. Yutaka et al. in [26] derived an algorithm that allows a nonholonomic constrained rigid body track a straight line without twirling. The authors derived a function called the steering function by differentiating the trajectory's curvature as a linear combination of the vehicle's position error, orientation error, and current trajectory curvature. They then used Lyapunov's theory to show the asymptotic stabilizability of the system. In [27] the authors solved the problem of tracking and visual servo control for a nonholonomic wheeled robot using a practical approach. They used videos for desired paths and showed convergence by Lyapunov analysis. The authors of [28] also considered the combined tracking and visual problem; however, they included parameter uncertainties in their analysis. Keighobadi et al. in [29] discussed the problem of achieving complete tracking of a wheeled robot using a torque nonlinear controller. They proposed a two fuzzy controllers in solving this problem.

2.1.8 Other Control Methods

Other techniques that were not mentioned above have also been used to control nonholonomic systems. In [30], Hespanha et al. proposed a controller for nonholo-

nomic systems with uncertainty in the model parameters. The authors proposed a hybrid feedback controller using supervisory control. They also used a state transformation to circumvent the singularity issue common to nonholonomic models. Campion et al. [31] describes the nonholonomic system using Euler-Lagrange equations. They showed the derivation in a stepwise manner. Using this model, a smooth state feedback law was devised, it ensures global marginal stability. [32] uses the kinematic model form to achieve control goals. The authors combine a tensor with the gradient vector of a Lyapunov function to design the controller. Exponential convergence is achieved using the designed controller. Lastly, an experimental work was carried out, validating the kinematic model of a nonholonomic "sphericle" in [33], including the nonholonomic constraints and dynamic model.

The control techniques described above are for the general form nonholonomic models and are also applicable to planar unmanned aerial vehicles, a subset of this class of models.

2.2 Biologically Inspired Networks

The collective behavior of animals [34] is ever so fascinating. These behaviors such as schooling fish, swarming bees, and flocking birds continue to intrigue researchers and philosophers as many scientists have tried to understand and even imitate these behaviors. Couzin suggested in [35] that scientists are just beginning to understand the connection between single-level and group-level animal

behaviors, and what parts these relationships play in deciding adaptive responses.

Haque et al. [36] also indicated the growing interest of biological models among engineers. They formulated confinement schemes using the foraging behaviors that bottlenose dolphins adapt to capture their preys. They suggested that the bio-inspired schemes allowed for its implementation in robotic systems.

The manufacturing industry is not left out of this trend. In [37] , the authors noted that most traditional manufacturing systems have the difficulty of adapting to any change or disturbance in the manufacturing processes. They claimed that this situation is largely because these systems are pre-programed and cannot autonomously adapt to these changes. Thus, they proposed that biologically inspired autonomous rescheduling can be attained in the advent of disturbances or changes: equipment failure or malfunction. Specifically, [37] suggests an autonomous control system to deal with such changes in a manufacturing workshop. The authors of [38] highlighted the importance of building adaptive and autonomous manufacturing systems to meet present industry challenges. They suggested that the Multi-Agent System (MAS) concept puts up another way of achieving system autonomy that is biologically inspired. The MAS concept uses distributed and decentralized control of the different parts of the manufacturing system, thus providing autonomy to individual agents or parts of the system. The paradigm can thus be used to achieve biology inspired control of the systems holistically.

Reif et al. in [39] discussed a non-conventional behavioral social control technique for dealing with Very Large Scale Robotic (VLSR) systems. Again their

concern is channeled to industries that use several robots in their daily operations. The proposed a social potential field technique for VLSR systems by defining artificial attraction and repulsion forces between each robot. The approach also ensure the autonomy of each robot in a distributed manner.

The study of the biological behavior of fishes when deciding whether to forage or keep a safe distance from a predator was also studied in [40]. Behavioral models that have been used regarding this subject have been mostly theoretical. The authors thus used data from experiments to develop behavioral models that were used for research. They were developed through statistical model fitting. The study was then carried out using the obtained models.

Qin et al. [41] developed a flocking control for systems with many agents. Their research was targeted towards reference tracking and group motion challenges. They designed controllers that tracked both static and moving targets by identifying team leaders and then, proved the existence and uniqueness of the solution based on the closed loop system. The results were then applied to a group of nonholonomic robots. In 2009, Qin et al. [42] built on [41] by using a decentralized flocking controller when the target to be tracked is fixed.

Moving away from the concept of leaders and followers, in order to give autonomy to each biologically driven agent, [43] presents an idea that *flocks do not need to have leaders*. Olfati-Saber proposed a theoretical structure for formulating distributed flocking algorithms. He used systematically derived objective functions (or cooperative potentials) for the flock members. A Lyapunov stability

equivalent for flocking particles is then proposed.

Last, Dugatkin et al. in [44] studied the way the Guppy fish approaches a possible predator. This is called predator inspection. They observed that the guppies only approached the presumably less precarious part of the predator. On inspection, the guppies modified their behaviors if they perceived a threat. They also foraged less in the presence of danger.

These behaviors are very beneficial to this study and are incorporated in the network behaviors.

2.2.1 Adaptive Networks

Adaptive networks have been used as the backbone of many biologically inspired networks [45] [46] [47] [48]. Sayed Ali in [49] wrote a survey of the study of adaptive networks and the advances in the field. He noted the progress related to adjustments, optimization, and learning over networks. He also wrote about the different *distributed strategies* which allow networked agents to relate locally, learn and adapt to track changes in the data streams they receive and also in the models. He went further to carry out a performance analysis that proved useful in comparing the different network topologies and also the batch versus centralized implementations.

2.2.2 Distribution Strategies

Sayed [50] provides a summary on different distribution strategies and the best strategy to use for certain 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: non-cooperative adaptive strategy and diffusion strategy (cooperative). Sayed concluded, using mean square deviation performance analysis, that diffusion strategies outperform non-cooperative strategies. In Sayed's work with Lopes [51] , an adaptive diffusion least mean square algorithm was formulated to ensure cooperation among 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. [52] extends Lopes' and Sayed's work in [51] by adding data-normalized algorithms and a dynamic topology.

To improve the robustness of diffusion networks in the presence of disturbances and noisy sensor measurements, adaptive combiners are added to the networks as in [53]. 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 [54] Cattivelli et al. Here, diffusion-based adaptive solutions of the least mean square type are proposed

and shown to have improved performance. More recent advances are investigated in [55] [56] [57].

2.2.3 Bio-Inspired Applications

As discussed above, diffusion strategies have been used to describe natural collective behaviors of animals. This section highlights some more applications.

When birds travel long distances, they form a V-shape [58] in order to save energy: by depending on their neighbor's upwash. The authors use diffusion least mean square (LMS) algorithm to imitate the model of the upwash and bird's self organizing formation. The results show that diffusion LMS is capable of accurately describing the V-formation of flocking birds.

In [48] , diffusion algorithm is used to describe the behavior of bees swarming. In their movement to a new location, scout bees who already have the information about the new location lead the remaining bees who do not. The authors show that diffusion adaptation can sufficiently model this behavior.

Sheng-Yuan et al. also described the foraging behaviors [59] of a school of fish in search of food, by defining an objective for each agent in the network. The adaptive network of agents (fishes) is shown to respond adequately in a cohesive and coherent manner. This behavior model was extended in [45] to include collision avoidance and reuniting. In [60] , the authors use diffusion adaptation to describe the foraging behavior of a school of fish, expanding on the previous

work. The proposed algorithm assumes the knowledge of the position of each fish's neighbor. They confirmed the efficiency of diffusion algorithm in describing the fish foraging behaviors.

In further analyzing the behavior of a school of fish, [47] added another dimension to the diffusion algorithm: predation and mobility. In this case, the fishes and their predator (a shark) are mobile. Interestingly, the school of fish have dual objectives: to locate their food source and also evade their predator. Diffusion LMS strategy is shown to be sufficient in dealing with this problem. An extension is proposed in [61] , where the multiple predators were used to attack the school of fish. Once again, diffusion least mean square algorithm was used and confirmed adequate to imitate the biological behavior.

2.3 Trajectory Planning of Nonholonomic Systems

Because of the navigation algorithm which will be proposed in chap. (4), this section reviews researches that have been carried out in the area of path planning of nonholonomic systems. Zhengxiong et al. in [62] proposed the use of particle swarm optimization (PSO) in planning the trajectories of nonholonomic systems. The authors proposed that the path planning problem be converted to an optimal control problem by using the spline approximation method or the Fourier basis method. The authors in [63] also used PSO in the path planning of a wheeled

robot. They focused on the simulation of two goals - to get to a target and to collect balls - for the nonholonomic wheeled robot.

The Leapfrog method which is a technique for finding solutions to nonlinear optimal control problems was introduced in [64] as a way to generate trajectories that are optimal. It was shown to be efficient in generating paths which are physically feasible.

Now, coming to a fleet of UAVs, researches have developed several algorithms to ensure groups of UAVs can fly together efficiently and without colliding with each other. The problem of path planning and distributed autonomy of UAVs was considered in [65]. The authors proposed two methods of solving the stated problem. They first transformed the problem into a single Mixed-Integer Linear Program (MILP) problem and solved it using MATLAB and AMPL scripts. They also used an approximate method of computing the costs of several paths in deciding the most efficient and optimal path for each UAV. The second method is however only partially distributed. Shim et al. in [66] proposed the use of Model Predictive Control (MPC) in solving the combined stabilization and trajectory planning problems of multiple helicopters moving together. They showed that their integrated design gives a better result compared to the separate design which the controller and path planner are not integrated.

Desai et al. in [67] also presented a control and path planning strategy for mobile nonholonomic robots moving within an area with obstacles. The robots were to keep a safe distance from each other and also follow a certain formation.

This was achieved using graph theory to identify leaders and followers in the team.

2.4 Evasive Planning Techniques

Shim and Sastry in [68] proposed an MPC for collision avoidance and trajectory planning of UAVs. They ensured each UAV kept a safe distance from each other; however, the UAVs would carry out an evasive maneuver to ensure collision is avoided at all cost. To achieve this, their proposed algorithm calculates the best feasible path based on the future predicted trajectory of its surrounding UAVs. Thus, the evasion is carried out without collision and in the safest way.

In [11], Sprinkle et al. considered evasive techniques for a flying UAV in combat mode. They visualized the possibility of having an adversary attack the UAV. The authors designed a nonlinear MPC that tracks the desired evasive path. This idea is one of the core parts of this thesis as we also consider evasive techniques; however using a cognitive evasive trajectories. Eklund et al. in [12] took the evasive maneuvering results obtained from the simulation in [11] and compared it with the achievable results when flying a piloted F-15 aircraft. They showed the capability of a UAV achieving such maneuvers practically in the nearest future.

Finally, the combined pursuit and evasion problem is consider in [69]. Once again, Sastry et al. used an MPC to achieve tracking of the desired path of an autonomous aircraft which is based either evasion or pursuit. Assumptions are made about the enemy aircraft and predictions are made on the future paths of the enemy. This would help the aircraft decide on its evasive strategy, likewise

its pursuit strategy. This thesis thoroughly deals with these pursuit and evasion strategies as we propose techniques to achieve these objectives.

CHAPTER 3

NONHOLONOMIC UAV MODEL

We consider nonholonomic UAVs moving in a plane. Being nonholonomic, it implies that the UAV model is not completely integrable, thus these robots are constrained by their velocities. A holonomic system is substitutable with a completely integrable system and a nonholonomic system is substitutable with a non-integrable system [15]. In this study and for practical reasons, we assume that the UAVs are equipped with Global Positioning Systems (GPS) to help them analyze and process location data streams; we also assume the UAVs are equipped with proximity sensors. This will be discussed further in chap. (4). The kinematic model of a unmanned aerial vehicle comprises of the orientation and position of the UAV moving within a plane. It is also subject to two input velocities. The motion of planar UAVs is constrained by their inability to make sudden changes in their velocities, therefore positions, and are thus difficult to maneuver. This is directly similar to a unicycle model (shown below), which has a wheel rolling in an upright position on a plane and also rolls without slipping. The constraints imply

the existence of non-integrable first-order differential equations, thus limiting the instantaneous movement which the planar UAV can perform without affecting their controllability. The nonholonomic UAV model is given below and is similar to the unicycle model shown in fig.(2.1):

$$\begin{aligned} \dot{x} &= u_1 \cos\theta_s \\ \dot{y} &= u_1 \sin\theta_s \\ \dot{\theta}_s &= u_2 \end{aligned} \tag{3.1}$$

where x and y represent the coordinate position of the UAV and θ_s is the steering angle.

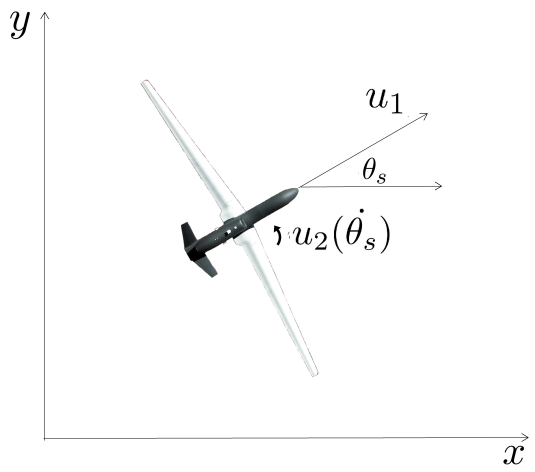

Figure 3.1: UAV model

Let the position and orientation states of the UAV be given by $q = [x \;\; y \;\; \theta_s]^T$. The generalized velocities of the UAV, $\dot{q}$, are not able to presume independent values. They have to fulfill the constraint:

$$\begin{bmatrix} \sin\theta_s & -\cos\theta_s & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta}_s \end{bmatrix} = 0 \tag{3.2}$$

This constraint is referred to as a Pfaffian constraint [70] such that

$$D(q)\dot{q} = 0 \tag{3.3}$$

where $D(q) = \begin{bmatrix} \sin\theta_s & -\cos\theta_s & 0 \end{bmatrix}$. It implies a zero lateral velocity: the linear velocity of the UAV center is in the body plane of the UAV. All practicable velocities must be within the null space of the matrix $D(q)$.

Let the generalized velocities in the null space of $D(q)$ be given by:

$$\begin{bmatrix} \cos\theta_s \\ \sin\theta_s \\ 0 \end{bmatrix} v_1, \quad \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} v_2 \tag{3.4}$$

So that,

$$\begin{bmatrix} \sin\theta_s & -\cos\theta_s & 0 \end{bmatrix} \begin{bmatrix} \cos\theta_s \\ \sin\theta_s \\ 0 \end{bmatrix} v_1 = 0, \quad \text{and} \begin{bmatrix} \sin\theta_s & -\cos\theta_s & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} v_2 = 0 \tag{3.5}$$

Hence, one can conclude that

$$\dot{q} = \begin{bmatrix} \cos\theta_s \\ \sin\theta_s \\ 0 \end{bmatrix} v_1 + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} v_2 \tag{3.6}$$

showing the interchangeable relationship between the Pfaffian constraint and state form model.

Eqn. (3.1) can also be rewritten as

$$\dot{q} = H(q)u \tag{3.7}$$

where

$$H(q) = \begin{bmatrix} \cos\theta_s & 0 \\ \sin\theta_s & 0 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{3.8}$$

Sometimes it is more comfortable to work with the Pfaffian constraint as in (3.3) rather than (3.7). Note that the Pfaffian constraint furnishes another way of representing driftless control affine systems [70].

3.1 Chained form

The canonical or chained form [20] of the nonholonomic model has proved to very useful in terms of exploring control strategies. Below is a transformation that converts the UAV model to its chained form.

Choosing the state transformation below to transform (3.1).

$$\begin{aligned} x_1 &= -\theta \\ x_2 &= x\cos\theta + y\sin\theta \\ x_3 &= -x\sin\theta + y\cos\theta \end{aligned} \tag{3.9}$$

Differentiating (3.9) to obtain new coordinates.

$$\begin{aligned} \dot{x_1} &= -u_2 \\ \dot{x_2} &= u_1\cos^2\theta - \dot{\theta}x\sin\theta + u_1\sin^2\theta + \dot{\theta}y\cos\theta \\ &= u_1 + u_2(-x\sin\theta + y\cos\theta) \\ \dot{x_3} &= -u_1\sin\theta\cos\theta - \dot{\theta}x\cos\theta + u_1\sin\theta\cos\theta - \dot{\theta}y\sin\theta \\ &= (-x\cos\theta - y\sin\theta)u_2 \\ \dot{x_1} &= -u_2 \\ \dot{x_2} &= u_1 + x_3u_2 \\ \dot{x_3} &= -x_2u_2 \end{aligned} \tag{3.10}$$

Again, using the transformation below:

$$\begin{aligned} v_1 &= u_1 + x_3 u_2 \\ v_2 &= -u_2 \end{aligned} \tag{3.11}$$

The canonical or chained form of the UAV is obtained:

$$\begin{aligned} \dot{x_1} &= v_2 \\ \dot{x_2} &= v_1 \\ \dot{x_3} &= x_2 v_2 \end{aligned} \tag{3.12}$$

The chained form has been used in literature especially for higher order nonholonomic systems; thus is important to mention.

Another important representation of nonholonomic systems is in the ***Power Form***. This representation has been shown to be a powerful technique in solving the problem of nonholonomic control and tracking. It employs the use of invariant manifolds [71] in designing controllers for the nonholonomic model.

CHAPTER 4

NAVIGATION ALGORITHM

In this section, we will discuss the path planning algorithms we propose for the UAV model. These navigation algorithms are planned based on the current state of each UAV in relation to each other and prevailing environmental conditions. These environmental conditions could be the presence of an enemy or any form of danger, the absence or presence of other UAVs, or even the goal to reach a target. These concepts would be better elaborated in the next sections.

We will consider a friendly fleet of UAVs flying at a safe distance from each other. The UAVs have a primary goal of reaching a target - this could be a destination or attack target in reality. An enemy UAV with similar nonholonomic constraints described in eqn.(3.1) is also on pursuit; its mission is to attack the closest UAV in the fleet. However, UAVs in the fleet realize the presence of an enemy and plan to evade the predator. The physical application of our algorithm will require the use of GPS in the UAVs for distributed location calculations and other sensors for proximity and velocity measurements.

Our objective is to get these fleet of UAVs flying in a manner that is inspired by how a school of fish move, behave and respond when foraging or escaping danger. We will explore the collective biological behaviors that fishes exhibit and use them in defining the trajectory which the UAVs will fly in a distributed manner.

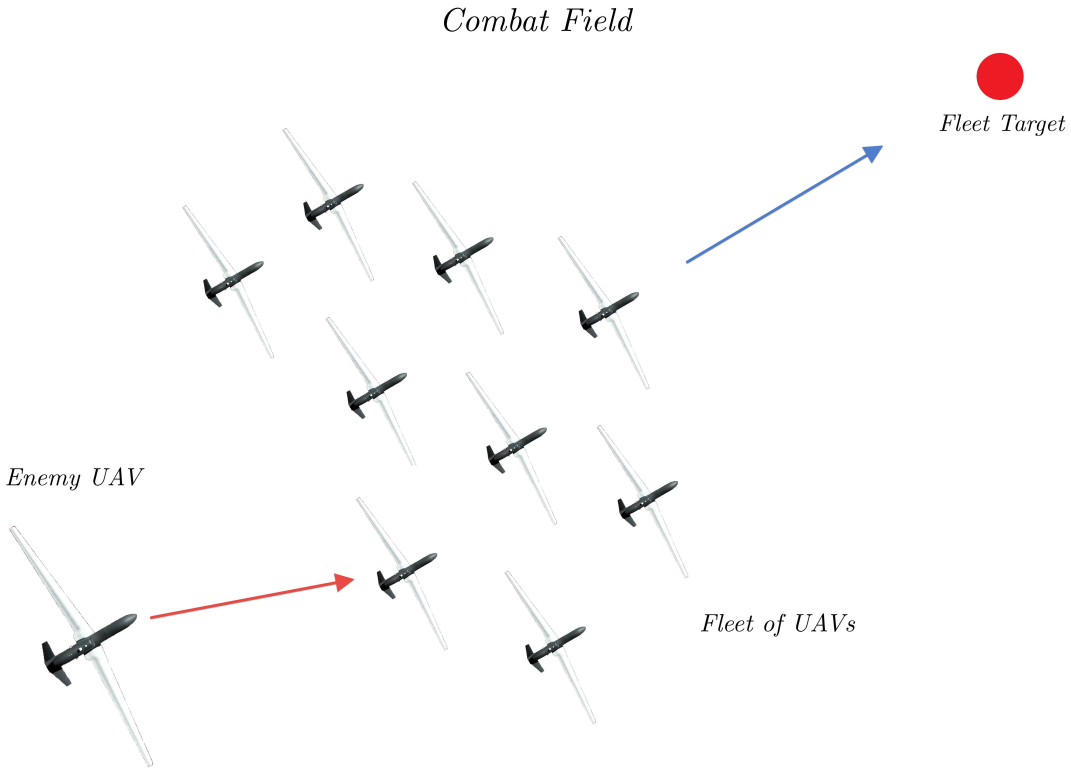

Figure 4.1: Air Combat

There are two main phases of navigation proposed for each UAV are:

1. The foraging phase

2. The evasion phase

4.1 The Foraging Phase

The UAVs, just like fishes, forage when in search of a target. In this case, there is no immediate danger present and the fleet of UAVs would be modeled to behave/respond like a school of fish in search of food. To achieve this, we would employ, as in [47], the fish-prey algorithm explained below.

4.1.1 Fish-Prey Algorithm

The fish-prey algorithm uses the concept of adaptive networks and adds mobility as another attribute. The adaptive network is strongly connected and examined from the viewpoint a group of nodes capable of learning and interacting with one another locally to accomplish distributed inference and processing challenges in real time. The nodes relate locally with their neighbors within neighborhoods which are constantly changing due to the mobility of the nodes. This leads to a network topology which is adaptive in nature. The self-organizing behaviors of a school of fish can thus be efficiently modeled using adaptive algorithms. To draw parallels with the fleet of UAVs, the network nodes represent each fish in the school while they are foraging, and in turn represent each UAV in the fleet. In the ongoing subsections, *we will use the terms UAVs, nodes and fishes interchangeably - each UAV in the fleet refers to a node in the network, and at the same time a fish in the school.* Our aim is to use this algorithm to define desired UAV trajectories that will be used for tracking control purposes in chap.(5).

The following assumptions used in modeling:

S/N	Assumptions
1	We assume a GPS and proximity sensors are in use for practical applications.
2	Each UAV is aware of the position and orientation of other UAVs in its local neighborhood.
3	Every UAV is locally aware of the location of the target.
4	Every UAV is locally aware of the position and orientation of the enemy UAV but the rule of engagement will be active only when the enemy UAV enters its safe region.
5	The UAVs are assumed to have physical dimensions [73], with motion governed by physical laws.
6	The target (or food for fishes) is assumed to be stationary while the enemy UAV is mobile.
7	The adaptive network is strongly connected.
8	The UAV is 2-D and ignores some aerodynamic considerations of a physical 3-D UAV.

Table 4.1: Assumption Table

Diffusion Adaptation

A group of N nodes/UAVs distributed in space is examined. Every node k is independent and distributed at each time i such that each one evaluates a scalar random process $d_k(i)$, a regression vector matching the actualization of a random process $\bar{u}_{k,i}$, where $d_k(i)$ and $\bar{u}_{k,i}$ are correlated. Each node k tries to calculate some vector parameter w^o using the random processes $[d_k(i), \bar{u}_{k,i}]$. These processes and parameters are related using the equation:

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

where n_k is a perturbation usually taken as zero-mean white noise; independent of remaining variables.

In the distributed algorithms, data is disseminated to each node such that it is in communication with its local environment - a subset of the nodes - which distribute the information to the remaining nodes in the network. Two nodes are linked or connected when communication is direct between them. Each node is always linked to itself.

The neighborhood of node $\mathcal{N}_k$ refers to the set of nodes linked to that node including the node. The network aims to compute the parameter w^o such that it minimizes the global objective function [45]:

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

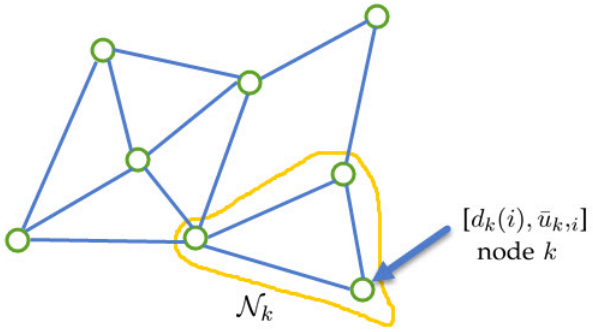

Figure 4.2: Node depicting its neighbors

where E represents the expectation of the process.

Sayed in [51] [54] proposed different diffusion adaptation techniques. In this research we employ the Adapt-Then-Combine (ATC) diffusion algorithm to solve

the distributed adaptation problem. The ATC algorithm is a two stage process: first an adaptation stage and then a combination stage. The algorithm uses a real positive coefficient $a_{l,k}$ to assign weights to the communication links between each node k and its neighbors l. The weighting coefficient must satisfy the equation below.

$$\sum_{l=1}^{N} a_{l,k} = 1, \qquad a_{l,k} = 0, \ \forall \ l \notin \mathcal{N}_k \tag{4.3}$$

During the adaptation stage, each node k obtains the local information from its neighbors to adapt locally. This stage also allows for an update of the state being evaluated from its previous value. In our application, the states represent the location of the UAV/node at the time i. The combination stage uses the intermediate values of neighbors obtained during adaptation and combines them together using the weights $a_{l,k}$. The ATC algorithm is described below.

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

where $w_{k,i}$ denotes the vector parameter which the ATC tries to compute (a UAV's or target's desired location at time i), μ_k is the non-negative step-size applied by node k, $\hat{d}_k(i)$ is the regression form of $d_k(i)$, and $\bar{u}_{k,i}$ is as described above. These variables would be given physical meanings in the next few subsections.

Measurement Model

The measurement model tries to estimate and model the school of fish and prey behavioral relationship. This behavioral model is thus eventually applied to our UAV model. Consider the vector parameter w^o, showing the location of the fish's food source or target that the networked school of fish want to locate. The figure below, fig.(4.3), gives this clarity.

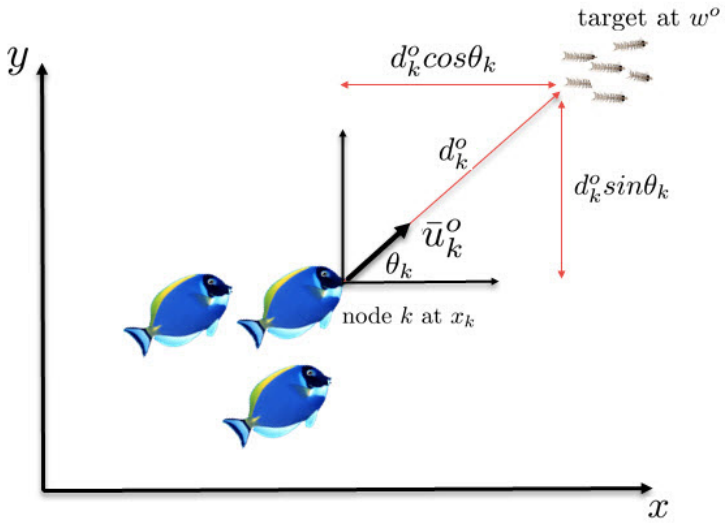

Figure 4.3: Fishes advancing towards the food source at w_o

It shows that the distance between the fish/node k and the food target w^o is represented by $d_k^o(i)$.

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

The fish/node k is located at $x_{k,i}$, which is constantly changing relative to a plane. *Note that $x_{k,i}$ used in this section represents a point located on the plane and not the x-direction of a coordinate system.* The unit direction vector connecting the

location of the fish to the food source is given by $\bar{u}^o_{k,i}$ and is defined in terms of θ_k thus:

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

The fishes live in a noisy environment, thus introducing some disturbance to measurement accuracy. This is synonymous to the UAVs flying against the wind - disturbance. This disturbance is taken into account.

$$\begin{aligned} \bar{u}_{k,i} &= \bar{u}^o_{k,i} + n^{\bar{u}}_{k,i} \\ d_k(i) &= d^o_k(i) + n^d_k(i) \\ \hat{d}_k(i) &\cong d_k(i) + \bar{u}_{k,i}x_{k,i} \end{aligned} \tag{4.7}$$

where $\bar{u}^o_{k,i}$ and $d^o_k(i)$ represent true values, $\hat{d}_k(i)$ is a linear regression model for the changing distance $d_k(i)$, and $n^{\bar{u}}_{k,i}$ and $n^d_k(i)$ are additive disturbance terms. In mimicking the foraging behavior of a school of fish, we assume that each fish represents a node in the network, where its goal is to locate two different targets simultaneously: the food source's location w^f and also that of the predator w^p.

Motion Control Algorithm

Each node's (k) location x_k in the mobile network is defined by the rule:

$$x_{k,i+1} = x_{k,i} + \triangle t.v_{k,i+1} \tag{4.8}$$

where i represents the time/instant, $\triangle t$ represents the sampling time for the node to move from one location to the next and $v_{k,i+1}$ represents the node's velocity. The velocity of the fish/node is determined by the following fish behaviors:

1. Its movement to the food source.
2. Self organizing behavior of the node with others.
3. Coherent motion.
4. Evasion when in danger.

Since we assume that the enemy/predator is moving, hence each node would calculate a *local* estimate of the food source $w^f_{k,i}$ and the predator $w^p_{k,i}$ in real time. To further illustrate how the nodes cognitively compute their velocities, see fig.(4.4) and illustrations below.

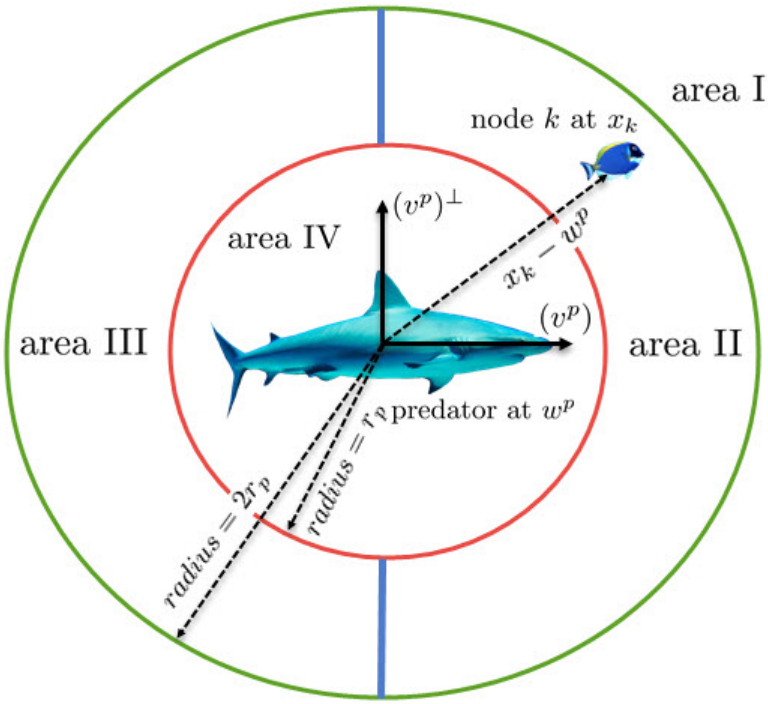

Figure 4.4: Fish cognitive behavior

Movement to the Food Source

For the UAV, this is synonymous to the approaching its desired target. In [47], similar situations are considered. Distinct areas identified in fig.(4.4) such that the response of a node relies on its location in its environment. The concentric circles shown have a radii of r_p and $2r_p$ from the predator. These separations - Area I, II, III, and IV - would help define rules for the next action by each node.

Area I

In this area, the node k is far away from the predator; this implies that the distance between k and the predator is more than $2r_p$. The node continues its foraging behavior by advancing towards its target. The velocity of the node is defined by:

$$v^a_{k,i+1} = \frac{w^f_{k,i} - x_{k,i}}{||w^f_{k,i} - x_{k,i}||} \quad \text{(area I)} \tag{4.9}$$

Area II

When a node is located a distance between radii r_p and $2r_p$, and to the right hand side of fig.(4.4), i.e. the predator is advancing to the node but not close enough, the node ends its search for the food target and joins its neighbors. The velocity of the node is defined by:

$$v^a_{k,i+1} = 0 \quad \text{(area II)} \tag{4.10}$$

Area III

Here the node is located a distance between radii r_p and $2r_p$, and to the left hand side of fig.(4.4), i.e. the predator is traveling away from the node. The node's

response would be to move in the reverse direction of the predator. The velocity is defined by:

$$v^a_{k,i+1} = -\frac{v^p_{k,i}}{||v^p_{k,i}||} \quad \text{(area III)} \tag{4.11}$$

where $v^p_{k,i}$ is node k's approximation of the predator's velocity.

Area IV

In this final case, the node is within the danger area of the predator; the node's distance from the predator is less than r_p. In this case, we will develop cognitive evasive control strategies that would evade the predator. The control strategies would be discussed in section(5) and the navigation trajectories for evasion would be elaborated in the next few sections.

Self Organizing Behavior

Once a predator attacks the school of fish, the network is broken to smaller groups. For their reunion and biological organization to take place, each node at the outer boundaries of the smaller groups calculate the position of the other groups and advances to them. First, each node determines if it is on the edge of the broken down school. Three types of edges are defined namely; the front, the right and the left edges. The node k enumerates its neighbors l in each direction using:

$$x^k_{l,i} = W(v_{k,i})^T(x_{l,i} - x_{k,i}) \tag{4.12}$$

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

$$W(v) = \begin{bmatrix} v_1/||v|| & -v_2/||v|| \\ v_2/||v|| & v_1/||v|| \end{bmatrix} \tag{4.13}$$

which is an orthonormal matrix defining the local coordinate system for a fish advancing with the vector velocity $v = (v_1, v_2)$.

After eqn.(4.12) is evaluated, if the value of $x^k_{l,i}$'s first coordinate is more than zero, then l's position is in front of k. If the value of $x^k_{l,i}$'s second coordinate is more than zero, then l's position is left of k. Last, if the value of $x^k_{l,i}$'s second coordinate is less than zero, then l's position is right of k.

Node l's Position w.r.t. k ($x^k_{l,i}$)	**Value**	**Implication**
First Coordinate	$x^k_{l,i} > 0$	Node l is in *front* of k
Second Coordinate	$x^k_{l,i} > 0$	Node l is *left* of k
	$x^k_{l,i} < 0$	Node l is *right* of k

Table 4.2: Estimating nodes in outer boundaries of fragmented groups

Node k is at the front edge if its front neighbors are less than one. Same goes for the right and left edges. The edge nodes k thus search for others and move towards them using:

$$\hat{l} = arg \min_{l} ||x^k_{l,i}|| \; |l \in \mathcal{N}^F_k \setminus \mathcal{N}_k \tag{4.14}$$

$$v^b_{k,i+1} = \begin{cases} 0, & \text{if } \hat{l} = \text{empty set} \\ \dfrac{x_{\hat{l},i} - x_{k,i}}{||x_{\hat{l},i} - x_{k,i}||}, & \text{otherwise} \end{cases} \tag{4.15}$$

$\mathcal{N}_k^F$ refers to the enumerated nodes that are in the front edge, $\mathcal{N}_k^L$ and $\mathcal{N}_k^R$ would also refer to that of the left and right respectively.

Coherent motion

For the nodes to adequately mimic the school of fish biological behaviors, they must move collectively to confuse the predator and also keep a safe distance r from each other to prevent collision with their neighbors. This introduces the concept of ***Potential Fields***. Each node k satisfies the eqn. below to fulfill the anti-collision goal.

$$r - \varepsilon \leq ||x_k - x_l|| \leq r + \varepsilon \;\; \forall \; l \in \mathcal{N}_k \setminus \{k\} \tag{4.16}$$

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

The cohesion and anti-collision objectives are attained using the objective function [59]:

$$\mathscr{J}_k(v_k) = \sum_{l \in \mathcal{N}_k \setminus \{k\}} \left[||(x_k + \triangle t.v_k) - (x_l + \triangle t.v_l)|| - r^2 \right] \tag{4.17}$$

The minimization problem tries to reduce the difference between r and the distance between the updated nodes in eqn.(4.17). We find the derivative of eqn.(4.17) to obtain the optimal value of v_k. [59] shows that the approximate optimal value is given by:

$$v_k = \frac{1}{|\mathcal{N}_k| - 1} \sum_{l \in \mathcal{N}_k \setminus \{k\}} \left[v_l - \left(1 - \frac{r}{||x_l - x_k||} \right) \frac{x_l - x_k}{\triangle t} \right] \tag{4.18}$$

Eqn.(4.18) is composed of two terms: the first computes the mean velocity of node k's neighbors (except itself), and the second stimulates attractive and repulsive

forces - *potential field* - among the nodes. The second term calculates the mean displacement vectors and implies that the nodes should keep a distance of r from each other and also adapt their velocity direction to the direction of the average displacement vector.

This second term is extracted and applied to maintain the potential field forces $\delta_{k,i}$ in our mobile network.

$$\delta_{k,i} = \frac{1}{|\mathcal{N}_k| - 1} \sum_{l \in \mathcal{N}_k \setminus \{k\}} \left(1 - \frac{r}{||x_{l,i} - x_{k,i}||}\right)(x_{l,i} - x_{k,i}) \tag{4.19}$$

The velocity of the node k due to coherent motion can thus be defined as:

$$v^c_{k,i+1} = v^g_{k,i} + \gamma \delta_{k,i} \tag{4.20}$$

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

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

Since our algorithm is distributed, $v^g_{k,i}$ should also be obtained in a distributed way. We will apply the *diffusion adaptation* technique in section (4.1.1). Let the objective function be defined thus:

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

Applying the ATC diffusion algorithm with eqn.(4.4)'s structure yields:

$$\begin{cases} \psi_{k,i} & = (1-\mu_k^v)v_{k,i-1}^g + \mu_k^v v_{k,i} \\ v_{k,i}^g & = \sum\limits_{l\in\mathcal{N}_k} a_{l,k}^v \ \psi_{l,i} \end{cases} \tag{4.23}$$

where superscript v show a relation with v^g and $a_{l,k}^v$ satisfies eqn.(4.3).

Evasion

The evasive behaviors will be treated elaborately in the next few sections to ensure continuity.

Distributed Velocity Estimate

Combining all the identified biological behaviors, the nodes will adapt their velocities according to the combined velocity estimate:

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

γ, β, α and λ are nonnegative weighting elements, while $I_{k,i}$ is a switching function that is either zero if $v_{k,i+1}^a$ and $v_{k,i+1}^b$ are zero or equals 1 otherwise.

To ensure the distributive velocity estimate meets the nonholonomic constraints of a UAV, a maximum velocity v_{max} is set for the nodes.

It is important to note that the estimated velocity for each node obtained from the fish-prey algorithm is used in chap.(5) to define desired trajectories for the UAVs during the foraging phase. We will, in that chapter, propose a controller to

tracks the desired path, thus synchronizing the UAVs with the navigation algorithm.

4.2 The Evasion Phase

As UAVs fly together on a mission, they might encounter some resistance or an attack that would require them to make acrobatic maneuvers in order to prevent such attacks from damaging or destroying them. We term this phase of flight the evasion phase. Sastry in [11] also envisioned the days when UAVs will no more be remotely controlled. He thought about a fully autonomous UAV that would be capable of executing evasive strategies when challenged by an adversary. This is the essence of this flight phase. We propose evasive strategies whereby each UAV would track a certain/defined straight line trajectory in order to maneuver and evade an adversary. The trajectory generation process is defined and elaborated with diagrams in the coming sections.

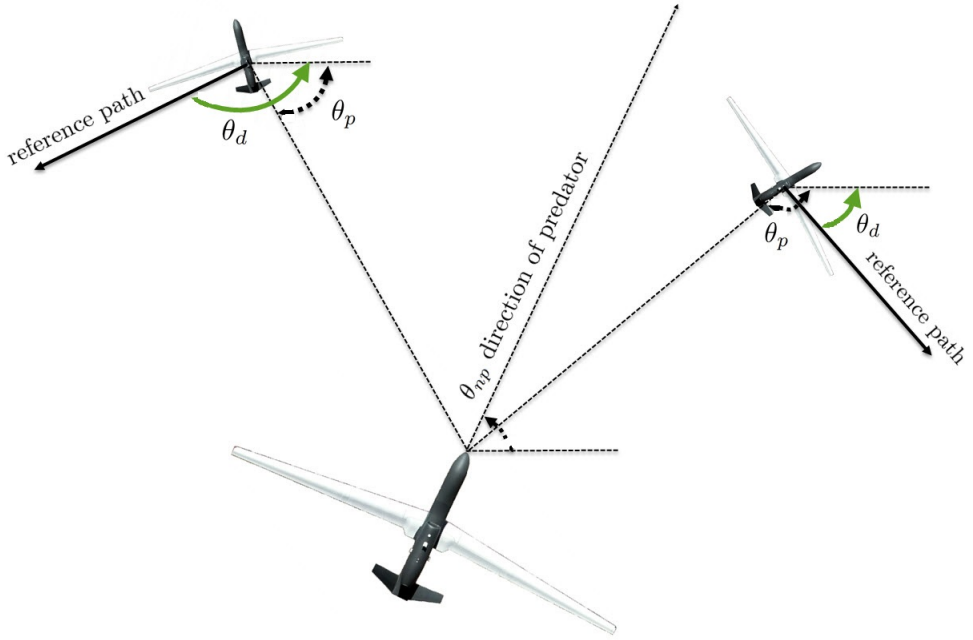

Figure 4.5: UAV Evasion Paths

4.2.1 Trajectory Generation

Our desired trajectory once a UAV senses danger is a straight line path, at an angle θ_d adjacent or perpendicular to the direction of the UAV. Consider a Cartesian plane, the line direction which the UAV would track depends on its current location and also direction which the adversary/enemy UAV is coming. To analyze this further, we will break the plane into the four trigonometric quadrants to consider all the possibilities. First, let's define the desired trajectory during evasion.

The UAV computes the distance d_k^p between the adversary and itself in real-time. At time (iteration) t_0, once $d_k^p < r_p$, the UAV senses the danger within the

defined danger-zone and it switches its planned foraging trajectory and behavior to evasion mode. The position of the UAV $(x(t_0), y(t_0))$ at the instant when the danger is sensed is stored, as it would serve as a reference point during the trajectory generation.

The straight line evasion path is defined using a set of parametric equations:

$$\begin{aligned} x_d(t) &= x(t_0) + \hat{\lambda}(n)\cos(\theta_d(t)) \\ y_d(t) &= y(t_0) + \hat{\lambda}(n)\sin(\theta_d(t)) \end{aligned} \tag{4.25}$$

where

$$\hat{\lambda}(n) = \triangle t + (n-1)\triangle t \tag{4.26}$$

The variable $\hat{\lambda}(n)$ is called the path parameter. It is set up as an arithmetic progression that increases to define the tracked trajectory. n is the iteration number during the execution of the evasion.

4.2.2 Evasive Strategies

We will now analyze the different evasion strategies and the desired straight paths. Consider a UAV on a plane, the desired trajectory will be shown for all four quadrants of a plane.

Here are some definitions which are crucial to the analysis of the proposed evasive strategies:

- We will be using the four-quadrant inverse tangent "$a\tan 2$" in our computations of the angles. For navigation purposes, $a\tan 2$ gives an angle value

between $-\pi$ and π. This is useful in calculating the angles during navigation.

$$-\pi \leq a\tan 2(y, x) \leq \pi \tag{4.27}$$

- θ_{np} is defined as actual angular direction of the enemy UAV. It is useful in deciding which direction the UAV turns to evade the enemy.
- θ_d is the desired orientation for the evasion of the UAV to a safe region. θ_d is obtained using the four-quadrant inverse tangent $a\tan 2$ and thus satisfies eqn.(4.27).
- θ_p is the UAV's approximation of the adversary's approaching angle.

The evasive angle of each UAV is used to define quadrants. The desired angles θ_d are based on the trigonometric quadrants. In deciding the orientation/angle which the UAV will choose to evade, we compare the angles θ_p and $\theta_{np} - 180$, and also θ_p and $\theta_{np} + 180$. This will be more obvious in the next figures.

Quadrant 1

In the first quadrant, once danger approaches i.e. $d_k^p < r_p$ and $\theta_p > \theta_{np} + 180$, the desired evasive angle is given:

$$\theta_d = \theta_p - \theta_c \tag{4.28}$$

where θ_c is the tangential angle (for example 90^o) desired for evasion. Here, the adversary UAV approaches at an angle between $-\pi$ and $-\frac{\pi}{2}$; the UAV on the other hand evades the enemy to track an orientation between 0 and π.

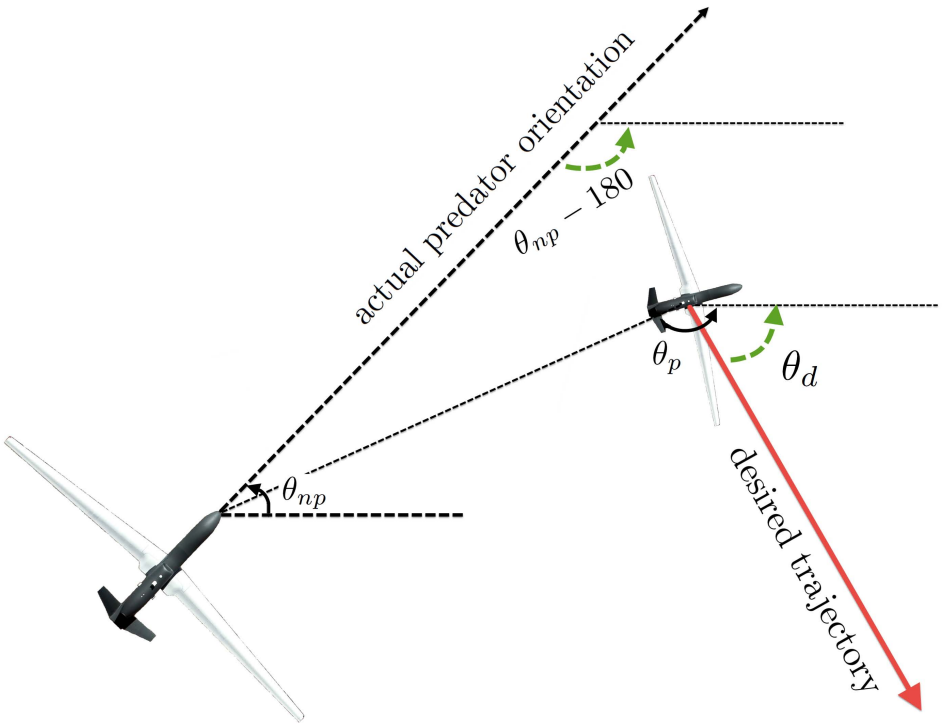

Figure 4.6: First Quadrant Evasion

Quadrant 2

In the second quadrant, once danger approaches i.e. $d_k^p < r_p$ and $\theta_p \leq \theta_{np} + 180$, the desired evasive angle is given:

$$\theta_d = \theta_p + \theta_c \tag{4.29}$$

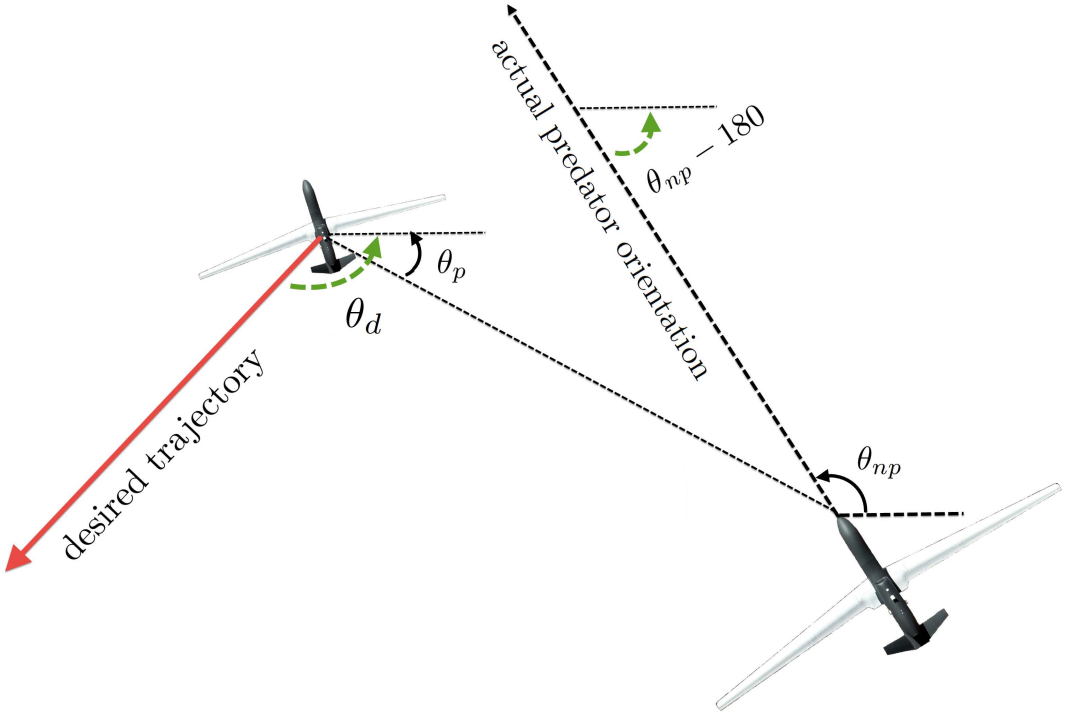

Figure 4.7: Second Quadrant Evasion

Quadrant 3

In the third quadrant, once danger approaches i.e. $d_k^p < r_p$ and $\theta_p \geq \theta_{np} - 180$, the desired evasive angle is given:

$$\theta_d = \theta_p - \theta_c \tag{4.30}$$

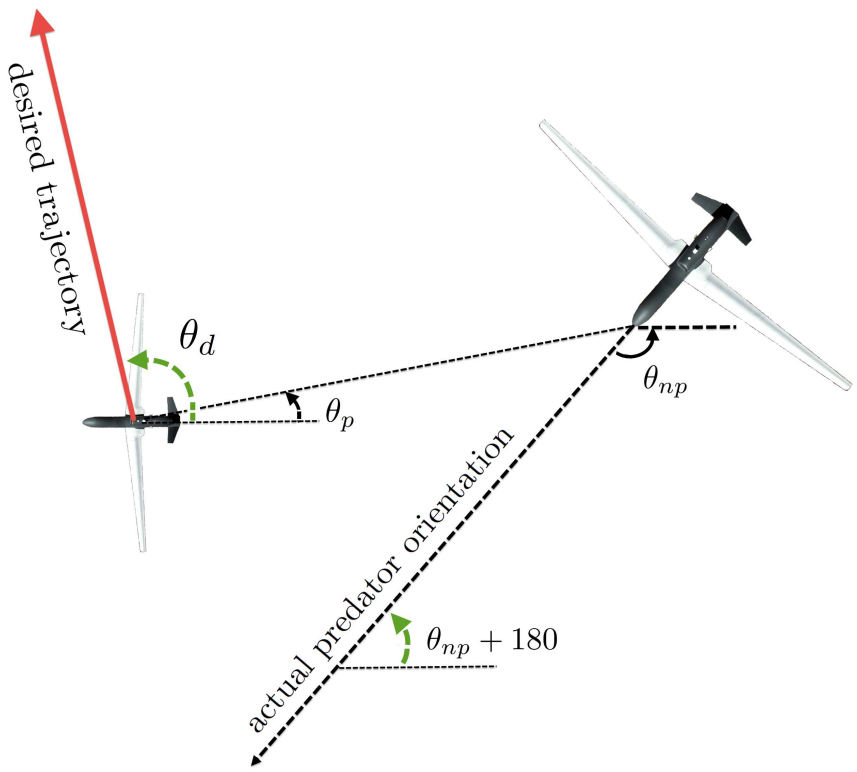

Figure 4.8: Third Quadrant Evasion

Quadrant 4

In the fourth quadrant, once danger approaches i.e. $d_k^p < r_p$ and $\theta_p < \theta_{np} - 180$, the desired evasive angle is given:

$$\theta_d = \theta_p + \theta_c \tag{4.31}$$

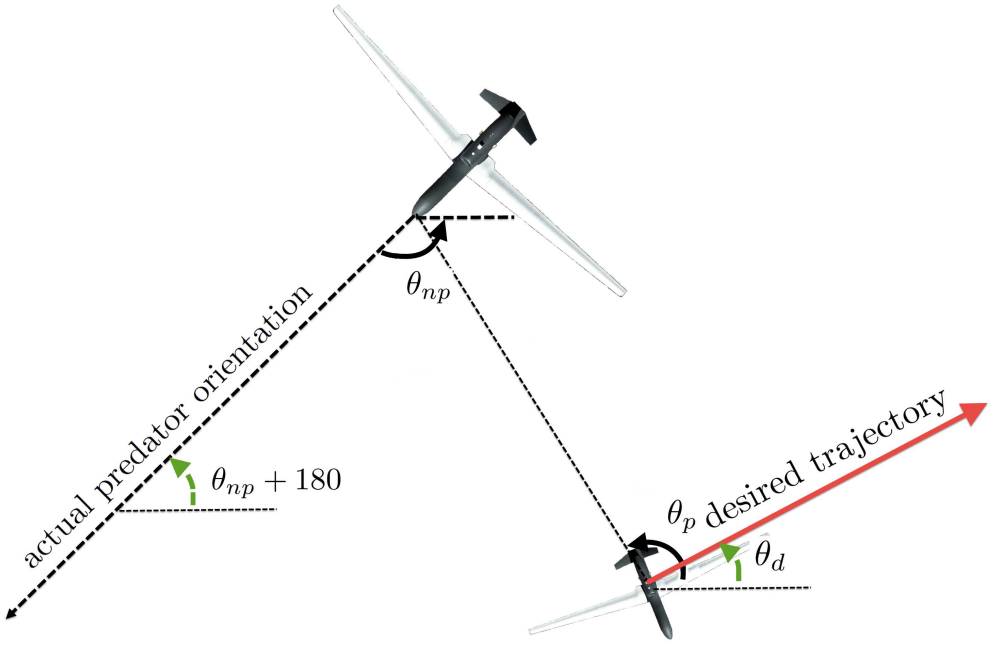

Figure 4.9: Fourth Quadrant Evasion

Here is a table showing the evasion rules and conditions:

Quadrant	**Conditions**	**Desired Evasion Angle**
1	$d_k^p < r_p$ and $\theta_p > \theta_{np} + 180$	$\theta_d = \theta_p - \theta_c$
2	$d_k^p < r_p$ and $\theta_p \leq \theta_{np} + 180$	$\theta_d = \theta_p + \theta_c$
3	$d_k^p < r_p$ and $\theta_p \geq \theta_{np} - 180$	$\theta_d = \theta_p - \theta_c$
4	$d_k^p < r_p$ and $\theta_p < \theta_{np} - 180$	$\theta_d = \theta_p + \theta_c$

Table 4.3: Evasion Table for Danger

4.3 Estimating the Velocity & Location of Predator & Food

As mentioned previously, it is assumed in this application that the predator is mobile and the food source is stationary. These variables are global variables. The goal is for the UAVs/nodes the compute the predator's location w^p that minimizes the objective function:

$$\mathscr{J}^{\,p}(w^p) = \sum_{k=1}^{N} E|\hat{d}_k^p(i) - \bar{u}_{k,i}\, w^p|^2 \tag{4.32}$$

In reality, the location of the predator or target is obtained using a GPS. For our purpose, this estimate is suitable and is a substitute. Using the ATC diffusion algorithm discussed in eqn.(4.4), the global location estimates of the predator $w_{k,i}^p$ and food $w_{k,i}^f$ are calculated with respect to each node. Note that the same computation applies for estimating the location of the target food w^f, simply replace p with f. The velocity of the predator with respect to each node is estimated as:

$$v_{k,i}^p = \frac{1}{\triangle t}(w_{k,i}^p - w_{k,i-1}^p) \tag{4.33}$$

4.4 Predator Behavior

The predator used for our application is also a nonholonomic UAV. It searches for the closest UAV and tracks only that node at a time. The estimated node location

w_i^n is obtained using the ATC diffusion algorithm in eqn.(4.4). The velocity and position of the predator are thus updated using the obtained location estimation:

$$\begin{cases} v_{i+1}^p & = c \,.\, v_{max} \dfrac{w_i^n - w_i^p}{||w_i^n - w_i^p||} \\ w_{i+1}^p & = w_i^p + \triangle t \,.\, v_{i+1}^p \end{cases} \tag{4.34}$$

c is a positive number that governs the predator's speed.

CHAPTER 5

TRACKING CONTROL OF MOBILE UAVS

In this chapter, we will integrate all the concepts proposed in the previous chapters by developing control techniques that ensure the desired trajectories projected from the navigation algorithms are tracked. We discuss the tracking problem for a single UAV then explore implementing the control algorithm on a fleet of UAVs. We also shed more light into the trajectory generation process and trajectory switching conditions. A flowchart showing the control decision process of the UAVs is shown in fig.(5.1). The control schematic for a single UAV closed loop system is shown in fig.(5.2). It is important to note once again that the UAV is 2-dimensional and ignores some aerodynamic considerations of a physical 3-dimensional UAV.

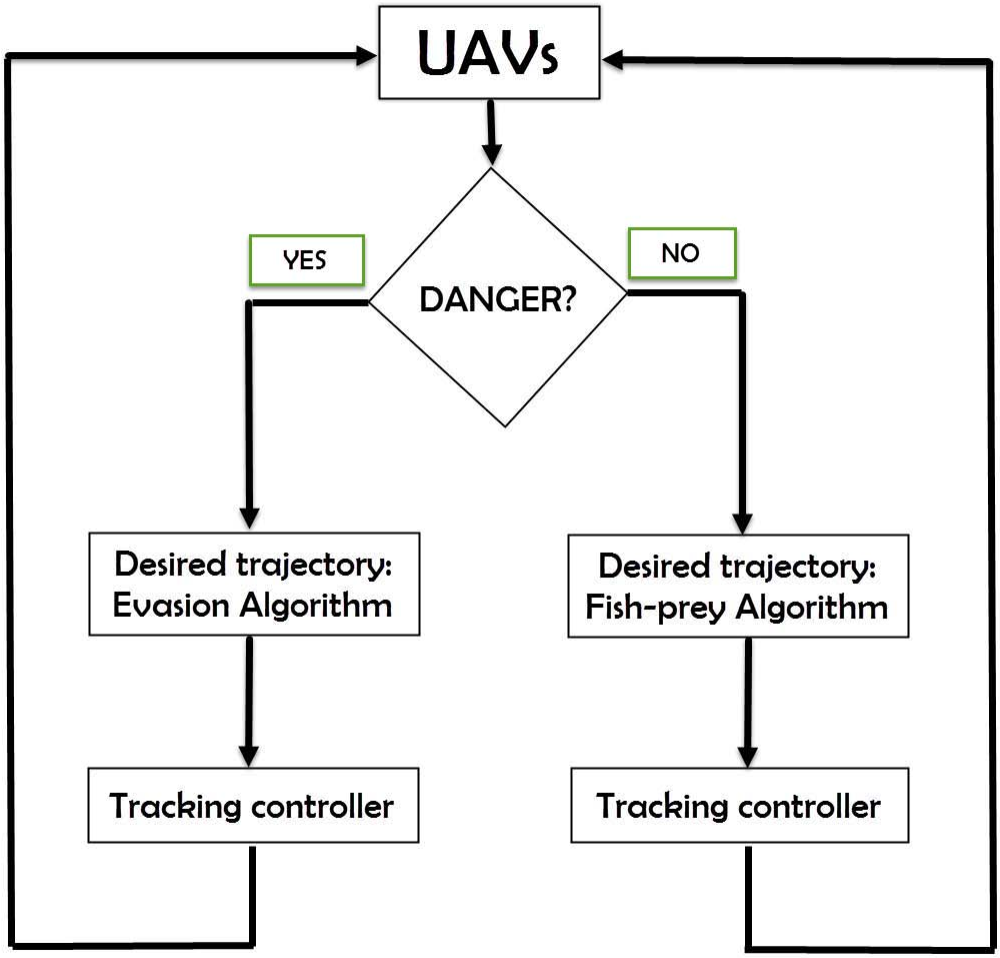

Figure 5.1: Fleet control decision flowchart

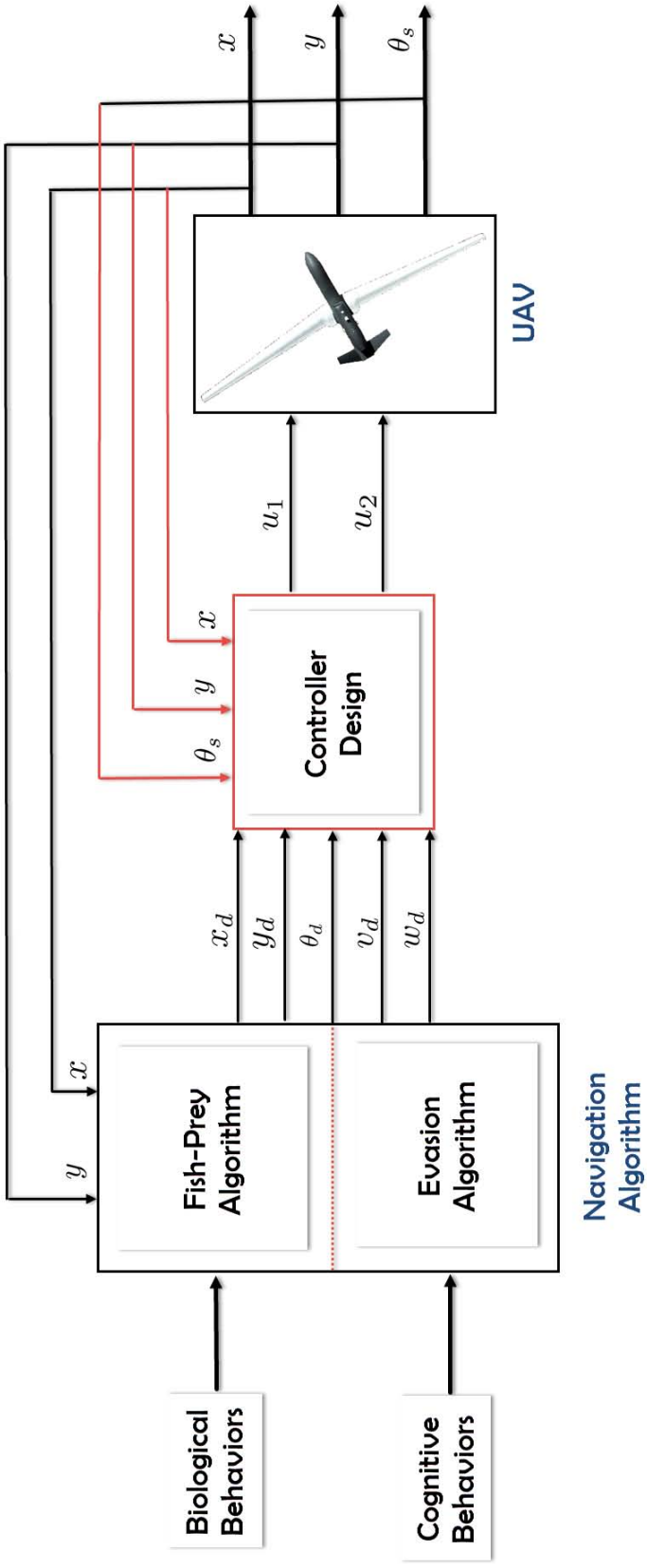

Figure 5.2: Control Schematic - Single UAV

5.1 Trajectory Generation

Based on the work done in chap.(4), the trajectory generation algorithms are put together in a concise manner. Let the desired states from the trajectory planning be defined thus:

- x_d – desired x coordinate position.
- y_d – desired y coordinate position.
- θ_d – desired steering angle θ_s.
- $\tilde{v}_d$ – desired UAV linear velocity in the presence of danger, obtained from the evasive control technique.
- V_d – desired UAV linear velocity in the absence of danger, obtained from the fish-prey algorithm with dynamics.
- w_d – desired UAV angular velocity in either the presence or absence of danger.

A more elaborate definition would be given for the desired velocities in the next section based on the presence or absence of an adversary UAV.

5.1.1 Absence of an Adversary UAV

When there is no enemy within the prescribed safe area, each UAV mimics the behaviors from the bio-inspired fish-prey algorithm. Thus, they move at the velocity inspired by the fish-prey algorithm. This velocity, obtained from the motion

control behavior with similar dynamical properties as the position subsystem of the nonholonomic UAV model, is used to obtain the next/predicted position of the desired trajectory. It is important to note that this trajectory also satisfies the nonholonomic constraints of the UAV. The trajectory generation model in the absence of an adversary UAV are given by:

$$\begin{aligned} x_d(t+1) &= x_d(t) + \triangle t \cdot V_d \cos\theta_d(t) \\ y_d(t+1) &= y_d(t) + \triangle t \cdot V_d \sin\theta_d(t) \\ \theta_d(t+1) &= a\tan 2d\ (y_d(t+1), x_d(t+1)) \\ w_d(t+1) &= \frac{\theta_d(t+1) - \theta_d(t)}{\triangle t} \end{aligned} \tag{5.1}$$

5.1.2 Presence of an Adversary UAV

In this case, the UAV will track a desired straight line, which was described in chap.(4), based on its orientation when it first senses the enemy UAV. The desired trajectory generation equations are given in eqn.(4.25).

5.2 Single UAV Tracking Analysis

Before considering a fleet of UAVs, we first develop a control technique that ensures the tracking of a desired path. A full state feedback linearization controller is developed for the nonholonomic UAV model; after which a backstepping controller is also developed for the nonholonomic model.

5.2.1 State Feedback Linearization Controller

We propose a state linearized controller that tracks a ***desired straight line trajectory*** through dynamic feedback.

Theorem 5.1 *Given the nonholonomic UAV model in eqn.(3.1), excluding aerodynamic physical constraints, there exists a dynamic feedback that, after the events of isolated discontinuities, ensures the UAV model exponentially tracks a desired straight line trajectory, such that the feedback is given by:*

$$\begin{bmatrix} u_3 \\ u_2 \end{bmatrix} = \begin{bmatrix} \cos\theta_s & -x_3\sin\theta_s \\ \sin\theta_s & x_3\cos\theta_s \end{bmatrix}^{-1} \begin{bmatrix} -a_1(x_3\cos\theta_s) - a_2(x(t) - x_d(t)) \\ -a_3(x_3\sin\theta_s) - a_4(y(t) - y_d(t)) \end{bmatrix} \tag{5.2}$$

where u_3 is a new auxiliary input to the system, x_3 is a new state defined as

$$x_3 = u_1 \quad and \quad u_3 = \dot{u_1} \tag{5.3}$$

a_1, a_2, a_3, and a_4 are positive feedback gains that make up a Hurwitz polynomial and $x_3 \neq 0$.

Proof. Let ζ_x represent the error between the coordinate position x and the desired x position x_d, ζ_y represent the error between the coordinate position y and the desired y position y_d. Thus,

$$\begin{aligned} \zeta_x &= x(t) - x_d(t) \\ \zeta_y &= y(t) - y_d(t) \end{aligned} \tag{5.4}$$

From eqn.(5.4), you can infer that:

$$\begin{aligned} \dot{\zeta_x} &= u_1 \cos\theta_s - \dot{x}_d(t) \\ \dot{\zeta_y} &= u_1 \sin\theta_s - \dot{y}_d(t) \end{aligned} \tag{5.5}$$

Since $\theta_d(t)$ for a straight line is constant, solving $\dot{x}_d(t)$ and $\dot{y}_d(t)$ using eqn.(4.25) yields zero.

$$\begin{bmatrix} \dot{\zeta_x} \\ \dot{\zeta_y} \end{bmatrix} = \begin{bmatrix} \cos\theta_s & 0 \\ \sin\theta_s & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{5.6}$$

As seen, the matrix above is singular and the input that would make the angular velocity error zero does not appear. To address this we will add an integrator and augment the system as we shall see in following equations. Differentiating further:

$$\begin{aligned} \ddot{\zeta_x} &= -u_1\ \dot{\theta_s} \sin(\theta_s(t)) + \dot{u_1} \cos(\theta_s(t)) \\ \ddot{\zeta_y} &= \quad u_1\ \dot{\theta_s} \cos(\theta_s(t)) + \dot{u_1} \sin(\theta_s(t)) \end{aligned}$$

Let $u_3 = \dot{u_1}$ represent a new auxiliary input and $x_3 = u_1$ represent a new state. Substitute for $\dot{\theta_s}$ as in eqn.(3.1).

$$\begin{aligned} \ddot{\zeta_x} &= u_3 \cos\theta_s - x_3 \sin\theta_s \cdot u_2 \\ \ddot{\zeta_y} &= u_3 \sin\theta_s + x_3 \cos\theta_s \cdot u_2 \end{aligned}$$

$$\begin{bmatrix} \ddot{\zeta_x} \\ \ddot{\zeta_y} \end{bmatrix} = \begin{bmatrix} \cos\theta_s & -x_3 \sin\theta_s \\ \sin\theta_s & x_3 \cos\theta_s \end{bmatrix} \begin{bmatrix} u_3 \\ u_2 \end{bmatrix} \tag{5.7}$$

As seen above, the matrix in always nonsingular if $x_3 \neq 0$. Considering that we are tracking a trajectory, the linear velocity, $x_3 = u_1$, cannot be equal to zero since that would imply stabilizing at a point.

To obtain a feedback that assures the tracking objective, let a synthetic input $\hat{f} = \ddot{\zeta}$ such that:

$$\begin{bmatrix} u_3 \\ u_2 \end{bmatrix} = \begin{bmatrix} \cos\theta_s & -x_3 \sin\theta_s \\ \sin\theta_s & x_3 \cos\theta_s \end{bmatrix}^{-1} \begin{bmatrix} \hat{f}_1 \\ \hat{f}_2 \end{bmatrix} \tag{5.8}$$

$\hat{f}_1$ and $\hat{f}_2$ are designed to stabilize the UAV in the desired path. Let

$$\begin{aligned} \hat{f}_1 &= -a_1 \dot{\zeta_x} - a_2 \zeta_x \\ \hat{f}_2 &= -a_3 \dot{\zeta_y} - a_4 \zeta_y \end{aligned} \tag{5.9}$$

$$\begin{aligned} \hat{f}_1 &= -a_1(x_3 \cos\theta_s) - a_2(x(t) - x_d(t)) \\ \hat{f}_2 &= -a_3(x_3 \sin\theta_s) - a_4(y(t) - y_d(t)) \end{aligned} \tag{5.10}$$

Thus the feedback linearized controller that ensures tracking of the desired straight line is given by:

$$\begin{bmatrix} u_3 \\ u_2 \end{bmatrix} = \begin{bmatrix} \cos\theta_s & -x_3 \sin\theta_s \\ \sin\theta_s & x_3 \cos\theta_s \end{bmatrix}^{-1} \begin{bmatrix} -a_1(x_3 \cos\theta_s) - a_2(x(t) - x_d(t) \\ -a_3(x_3 \sin\theta_s) - a_4(y(t) - y_d(t)) \end{bmatrix} \tag{5.11}$$

The above result provides exponential tracking of the required reference line; however, is dependent on the linear feedback $\hat{f}$ provided by eqn.(5.10). The main drawback of using this feedback linearization technique is because of the possibility of the UAV velocity crossing zero during an initial transient. For these reasons, we will now propose a nonlinear feedback controller which is independent on an auxiliary input to track a straight line trajectory.

5.2.2 Backstepping Controller

A nonlinear controller based on backstepping method is designed to achieve asymptotic tracking of the UAV to a desired trajectory.

Theorem 5.2 *There exists a control input based on the backstepping design for a nonholonomic UAV model excluding aerodynamic physical constraints, that ensures the UAV model asymptotically tracks a desired trajectory, such that the control input is given by:*

$$\begin{aligned} u_1 &= \tilde{v}_d \cos(\theta_d - \theta_s) + k_1 \cdot ((x_d - x)\cos\theta_s + (y_d - y)\sin\theta_s) \\ u_2 &= k_2 \tilde{v}_d \cdot (-(x_d - x)\sin\theta_s + (y_d - y)\cos\theta_s) + w_d + k_3 \tilde{v}_d \sin(\theta_d - \theta_s) \end{aligned} \tag{5.12}$$

where k_1, k_2, and k_3 are positive scalars.

Proof. The UAV model is given as:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta}_s \end{bmatrix} = \begin{bmatrix} \cos\theta_s & 0 \\ \sin\theta_s & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{5.13}$$

We desire to make the UAV to track a desired nonholonomic trajectory given as:

$$\begin{bmatrix} \dot{x}_d \\ \dot{y}_d \\ \dot{\theta}_d \end{bmatrix} = \begin{bmatrix} \cos\theta_d & 0 \\ \sin\theta_d & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{v}_d \\ w_d \end{bmatrix} \tag{5.14}$$

Let the tracking error be defined thus [74]:

$$\begin{bmatrix} \zeta_x \\ \zeta_y \\ \zeta_\theta \end{bmatrix} = \begin{bmatrix} \cos\theta_s & \sin\theta_s & 0 \\ -\sin\theta_s & \cos\theta_s & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d - x \\ y_d - y \\ \theta_d - \theta_s \end{bmatrix} \tag{5.15}$$

The error states go to zero when the desired and current states are equal. Differentiating the tracking error:

$$\begin{aligned} \zeta_x &= (x_d - x)\cos\theta_s + (y_d - y)\sin\theta_s \\ \zeta_y &= -(x_d - x)\sin\theta_s + (y_d - y)\cos\theta_s \\ \zeta_\theta &= \theta_d - \theta_s \end{aligned} \tag{5.16}$$

$$
\begin{aligned}
\dot{\zeta_x} = &\quad (\dot{x_d} - \dot{x})\cos\theta_s - \dot{\theta_s}(x_d - x)\sin\theta_s + (\dot{y_d} - \dot{y})\sin\theta_s + \dot{\theta_s}(y_d - y)\cos\theta_s \\
\dot{\zeta_y} = &-(\dot{x_d} - \dot{x})\sin\theta_s - \dot{\theta_s}(x_d - x)\cos\theta_s + (\dot{y_d} - \dot{y})\cos\theta_s - \dot{\theta_s}(y_d - y)\sin\theta_s \\
\dot{\zeta_\theta} = &\quad \dot{\theta_d} - \dot{\theta_s}
\end{aligned}
\tag{5.17}
$$

$$
\begin{aligned}
\dot{\zeta_x} = &\quad \tilde{v}_d\cos\theta_d\cos\theta_s - u_1\cos^2\theta_s + \tilde{v}_d\sin\theta_d\sin\theta_s - u_1\sin^2\theta_s \\
&+ u_2(-(x_d - x)\sin\theta_s + (y_d - y)\cos\theta_s) \\
\dot{\zeta_y} = &-\tilde{v}_d\cos\theta_d\sin\theta_s + u_1\cos\theta_s\sin\theta_s + \tilde{v}_d\sin\theta_d\cos\theta_s - u_1\sin\theta_s\cos\theta_s \\
&- u_2((x_d - x)\cos\theta_s + (y_d - y)\sin\theta_s) \\
\dot{\zeta_\theta} = &\quad w_d - u_2
\end{aligned}
\tag{5.18}
$$

For $\dot{\zeta_x}$

$$
\begin{aligned}
\tilde{v}_d\cos\theta_d\cos\theta_s + \tilde{v}_d\sin\theta_d\sin\theta_s &= \tilde{v}_d\cos(\theta_d - \theta_s) \\
u_1\sin^2\theta_s + u_1\cos^2\theta_s &= u_1 \\
-(x_d - x)\sin\theta_s + (y_d - y)\cos\theta_s &= \zeta_y
\end{aligned}
\tag{5.19}
$$

For $\dot{\zeta_y}$

$$
\begin{aligned}
-\tilde{v}_d\cos\theta_d\sin\theta_s + \tilde{v}_d\sin\theta_d\cos\theta_s &= \tilde{v}_d\sin(\theta_d - \theta_s) \\
u_1\cos\theta_s\sin\theta_s - u_1\cos\theta_s\sin\theta_s &= 0 \\
(x_d - x)\cos\theta_s + (y_d - y)\sin\theta_s &= \zeta_x
\end{aligned}
\tag{5.20}
$$

Thus eqn.(5.18) is reduced to:

$$\begin{aligned} \dot{\zeta_x} &= \tilde{v}_d \cos\zeta_\theta + u_2\zeta_y - u_1 \\ \dot{\zeta_y} &= \tilde{v}_d \sin\zeta_\theta - u_2\zeta_x \\ \dot{\zeta_\theta} &= w_d - u_2 \end{aligned} \tag{5.21}$$

In order to track the required reference, we aim to reduce the posture tracking error to zero. To achieve this, we define a Lyapunov function and design a backstepping controller with control inputs $u = [u_1 \;\; u_2]^T$ that ensures the tracking of the desired states x_d, y_d, and θ_d such that:

$$\lim_{t\to\infty} [|\zeta_x| + |\zeta_y| + |\zeta_\theta|] = 0 \tag{5.22}$$

Let V_L be defined as Lyapunov function:

$$\begin{aligned} V_L = \frac{1}{2}((x_d - x)\cos\theta_s + (y_d - y)\sin\theta_s)^2 + \frac{1}{2}(-(x_d - x)\sin\theta_s + (y_d - y)\cos\theta_s)^2 \\ + \frac{1}{k_2}(1 - \cos(\theta_d - \theta_s)) \end{aligned} \tag{5.23}$$

where k_2 is a positive scalar. Using eqn.(5.16), the Lyapunov function can be simplified.

$$V_L = \frac{1}{2}\zeta_x^2 + \frac{1}{2}\zeta_y^2 + \frac{1}{k_2}(1 - \cos(\zeta_\theta)) \tag{5.24}$$

Differentiating further to prove the convergence to the desired trajectory:

$$
\begin{aligned}
\dot{V}_L &= \dot{\zeta_x} \cdot \zeta_x + \dot{\zeta_y} \cdot \zeta_y + \frac{\dot{\zeta_\theta}}{k_2} \sin(\zeta_\theta) \\
&= (\tilde{v}_d \cos\zeta_\theta + u_2\zeta_y - u_1) \cdot \zeta_x + (\tilde{v}_d \sin\zeta_\theta - u_2\zeta_x) \cdot \zeta_y + \frac{w_d - u_2}{k_2} \sin(\zeta_\theta) \\
&= (\tilde{v}_d \cos\zeta_\theta - u_1) \cdot \zeta_x + \tilde{v}_d \sin(\zeta_\theta) \cdot \zeta_y + u_2\zeta_x \cdot \zeta_y - u_2\zeta_x \cdot \zeta_y + \frac{w_d - u_2}{k_2} \sin(\zeta_\theta) \\
&= (\tilde{v}_d \cos\zeta_\theta - u_1) \cdot \zeta_x + \tilde{v}_d \sin(\zeta_\theta) \cdot \zeta_y + \frac{w_d - u_2}{k_2} \sin(\zeta_\theta) \\
&= (\tilde{v}_d \cos\zeta_\theta - u_1) \cdot \zeta_x + (\tilde{v}_d \cdot \zeta_y + \frac{w_d - u_2}{k_2}) \sin(\zeta_\theta) \\
&= (\tilde{v}_d \cos\zeta_\theta - u_1) \cdot \zeta_x + (k_2\tilde{v}_d \cdot \zeta_y + w_d - u_2)\frac{1}{k_2} \sin(\zeta_\theta)
\end{aligned}
$$

To ensure the tracking errors go to zero, $\dot{V}_L < 0$. Thus let

$$
\begin{aligned}
u_1 &= \tilde{v}_d \cos(\zeta_\theta) + k_1 \cdot \zeta_x \\
u_2 &= k_2\tilde{v}_d \cdot \zeta_y + w_d + k_3\tilde{v}_d \sin(\zeta_\theta)
\end{aligned}
\tag{5.25}
$$

where k_1 and k_3 are positive scalars. This implies that

$$
\begin{aligned}
\dot{V}_L &= -k_1 \cdot \zeta_x^2 - \frac{k_3}{k_2} \cdot \tilde{v}_d \sin^2(\zeta_\theta) \\
\dot{V}_L &< \quad 0
\end{aligned}
\tag{5.26}
$$

We can deduce that applying the input u

$$
\begin{aligned}
u_1 &= \tilde{v}_d \cos(\theta_d - \theta_s) + k_1 \cdot ((x_d - x)\cos\theta_s + (y_d - y)\sin\theta_s) \\
u_2 &= k_2\tilde{v}_d \cdot (-(x_d - x)\sin\theta_s + (y_d - y)\cos\theta_s) + w_d + k_3\tilde{v}_d \sin(\theta_d - \theta_s)
\end{aligned}
\tag{5.27}
$$

will cause the tracking error to reduce asymptotically to zero and make the UAV's position and orientation track the desired straight line reference. ∎

5.3 Multiple UAV Analysis

We have considered the case of a single UAV tracking a straight line reference for evasive purposes. Here, we consider the case of multiple nonholonomic UAVs tracking individually defined straight line references and also tracking trajectories inspired by the fish-prey algorithm when no danger is present.

5.3.1 Tracking the Evasion Trajectory

When many UAVs are evading along their defined paths, the possibility of collision arises. It is our desire to ensure that even as the UAVs evade an enemy, they are aware of the safe distance between themselves and try to maintain it in order to avoid collision.

To ensure collision avoidance during evasion, we introduce the concept of potential fields previously discussed. Each evading UAV satisfies the equation below in order to achieve the objective.

$$r - \varepsilon \leq ||x_k(t) - x_l(t)|| \leq r + \varepsilon \;\; \forall\, l \in \mathcal{N}_k \setminus \{k\} \tag{5.28}$$

where k represents a particular evading UAV and l indicates its neighbors. r and ε maintain their descriptions. The potential field $\delta_k(t)$ derived from the chap.(4)

is used.

$$\delta_k(t) = \frac{1}{|\mathcal{N}_k| - 1} \sum_{l \in \mathcal{N}_k \setminus \{k\}} \left(1 - \frac{r}{||x_l(t) - x_k(t)||}\right)(x_l(t) - x_k(t)) \tag{5.29}$$

Thus, including potential field into each UAV's linear velocity input, represented as u_1, yields:

$$\tilde{u}_1^k(t) = u_1^k(t) + \gamma_2 \delta_k(t) \tag{5.30}$$

where $u_1^k(t)$ is the input linear velocity on UAV k at instant t obtained from the closed loop controller and γ_2 is a positive scalar.

5.3.2 Tracking the Foraging Trajectory

As seen in fig.(5.2), the designed controller accepts inputs based on different trajectories that are inspired by the presence and absence of an enemy UAV. Now, we propose to use the designed controller to track the bio-inspired trajectories generated in the absence of an enemy UAV. The desired bio-inspired trajectory is given in eqn.(5.1).

Theorem 5.3 *In order to track the bio-inspired desired trajectories of a non-holonomic UAV model, excluding aerodynamic physical constraints, there exists a control input based on the backstepping design that ensures the UAV model asymp-*

totically tracks the bio-inspired trajectory, such that the control input is given by:

$$\begin{aligned} u_1 &= V_d \cos(\theta_d - \theta_s) + k_4 \cdot ((x_d - x)\cos\theta_s + (y_d - y)\sin\theta_s) \\ u_2 &= k_5 V_d \cdot (-(x_d - x)\sin\theta_s + (y_d - y)\cos\theta_s) + w_d + k_6 V_d \sin(\theta_d - \theta_s) \end{aligned} \tag{5.31}$$

where V_d is the distributed velocity estimate obtained from the fish-prey algorithm, k_4, k_5, and k_6 are positive scalars.

Proof. The theorem is easily proved using the proof of theorem (5.2). ∎

5.4 Limiting Velocity & Acceleration

During path planning of a nonholonomic UAV, the objective of tracking certain trajectories is also achievable using non-smooth paths, theoretically. Unfortunately, such non-smooth paths usually occur as a result of a large velocity input (acceleration input in the case of a feedback linearized controller). Such inputs are to large for the nonholonomic model to achieve in real applications due the nonholonomic constraints. Thus, to attain smooth tracking of reference paths, a maximum value is used to limit the velocities and accelerations.

CHAPTER 6

RESULTS AND DISCUSSIONS

Here we will simulate the UAVs to observe their behaviors when approaching a target and also when evading. We first analyze a single UAV under the evasion algorithm and then observe the entire fleet carrying out the foraging and evasion process together. The table (6.1) shows the variables used for the simulation carried out. The parameter values chosen are subjective, based on how you wish to design your network: for example, much distance you chose to maintain between the UAVs and also between the UAV and the enemy or the radius of communication of each UAV.

6.1 Single UAV Tracking Simulation

The tracking abilities of a single UAV is simulated to observe the effect of the evasion phase of the navigation algorithm. All four quadrants are considered in terms of the desired evasion or escape angle. Theorem(5.2) is applied to achieve tracking purposes. The controller parameters are chosen as: $k_1 = 1$, $k_2 = 20$, and

Variables	Meaning	Value
$\mathcal{N}$	Number of UAVs	49
$\mathcal{N}_k^{max}$	Maximum number of neighbors of UAV "k"	6
$w_{k,1}^p$	Initial Position of Predator	(50,50) or (-30,-30)
$w_{k,1}^f$	Initial Position of Food Target	(118,118)
$\triangle t$	Sampling Time	0.3
r_p	Predator Radius	12
r	Ideal distance between UAVs	4
R	Range of communication among UAVs	6
μ_k	Non-negative step-wise scalar	1
α	Non-negative weighting element	1
β	Non-negative weighting element	2
λ	Non-negative weighting element	0.5
γ	Non-negative weighting element	3
γ_2	Non-negative weighting element	0.8

Table 6.1: Simulation Table

$k_3 = 90$.

6.1.1 Quadrant I Evasion

The orientation which the UAV wishes to track is between 0^o and 90^o. The initial position and orientation of the UAV is chosen as $x(1) = 4$, $y(1) = 2$,and $\theta_s(1) = 100^o$. The desired initial states are given as $x_d(1) = 1.1$, $y_d(1) = 1.1$, $\theta_d = 60^o$, $\tilde{v}_d = 1$, and $w_d = 0$.

Evasion Simulation

With the desired orientation in the first quadrant, the figure below shows the UAV evading an approaching predator.

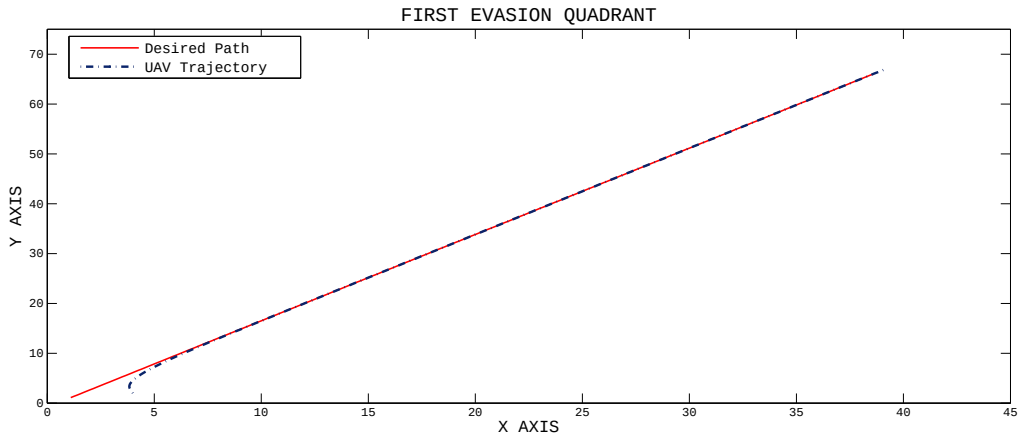

Figure 6.1: First Quadrant Evasion Simulation

The UAV is seen tracking the desired straight line trajectory. The tracking sequence is better observed in the zoomed-in figure below.

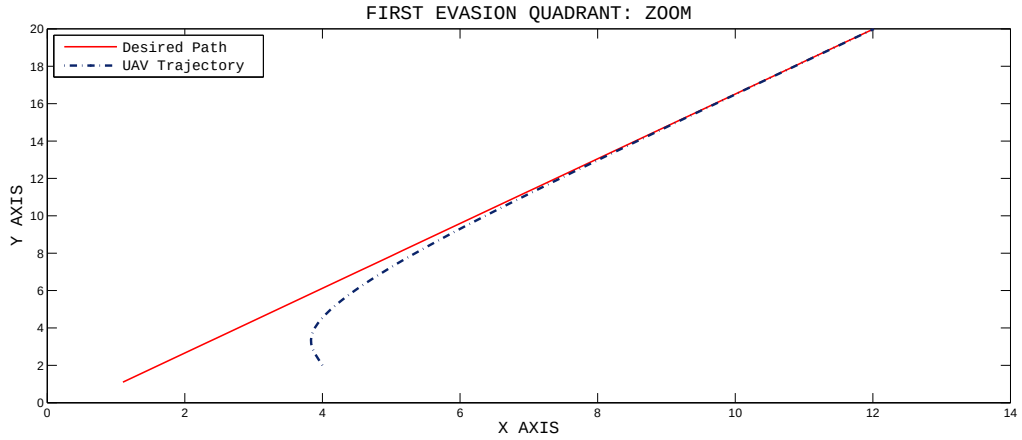

Figure 6.2: First Quadrant Evasion - Zoomed View

The nonholonomic UAV tracks the desired straight line path at a angle of $\theta_d = 60^o$ in the first quadrant. The nonholonomic nature of the system also becomes evident.

Tracking Error

The tracking error for position states - ζ_x and ζ_y - are plotted. The tracking orientation error - ζ_θ - is also plotted.

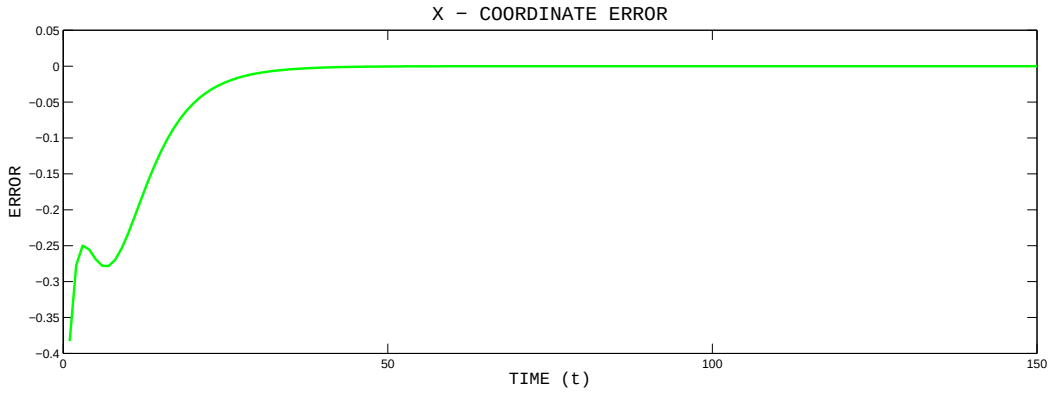

Figure 6.3: X-Coordinate Error - ζ_x

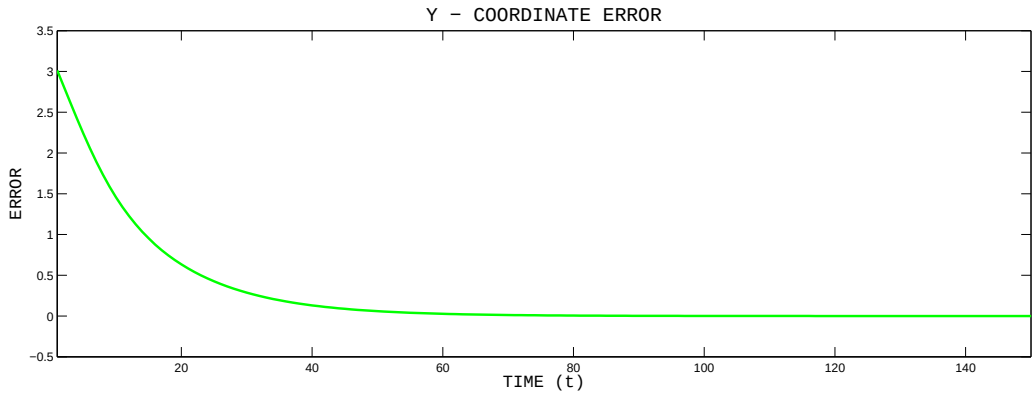

Figure 6.4: Y-Coordinate Error - ζ_y

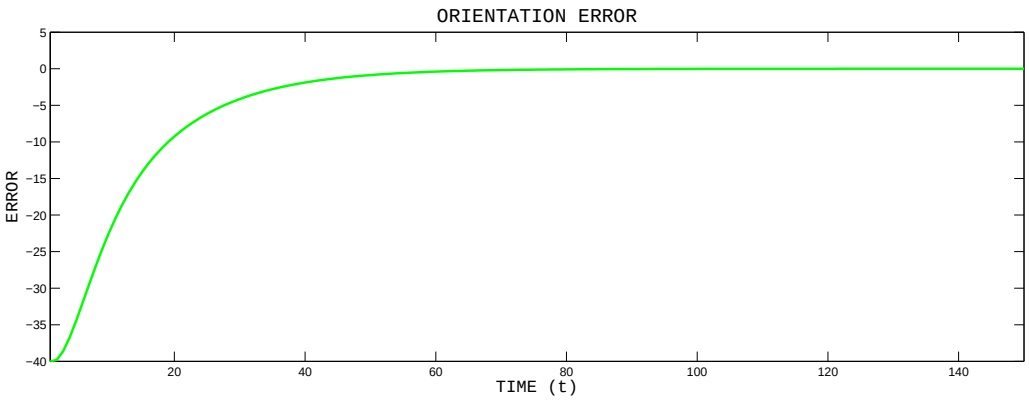

Figure 6.5: Orientation Error - ζ_θ

The tracking error states ζ_x, ζ_y and ζ_θ are seen to asymptotically approach zero, thus showing tracking ability of the controller. It is obvious that the further away the initial states of the UAV are from the desired states, the longer it takes the UAVs to converge to the desired trajectory.

Velocity Tracking

The desired linear and angular velocities we wish to track are $\tilde{v}_d = 1$, and $w_d = 0$ respectively. The figures below show the velocity tracking.

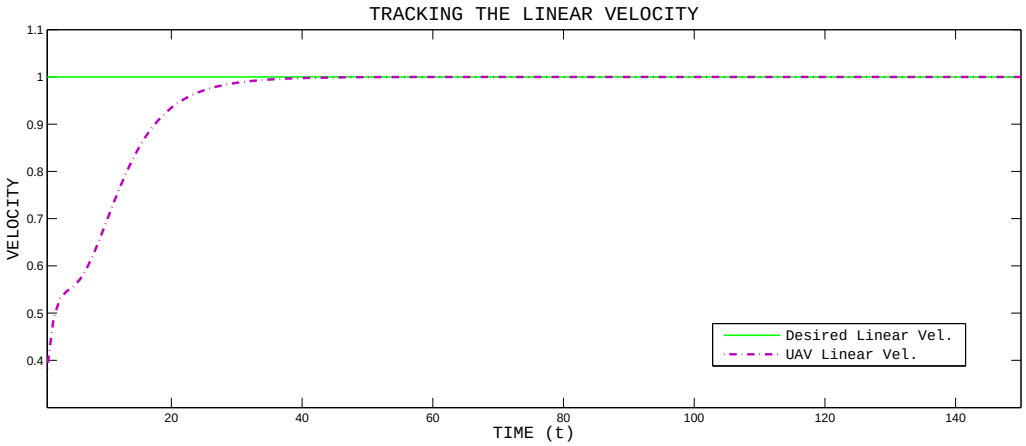

Figure 6.6: Linear Velocity Tracking - v_d

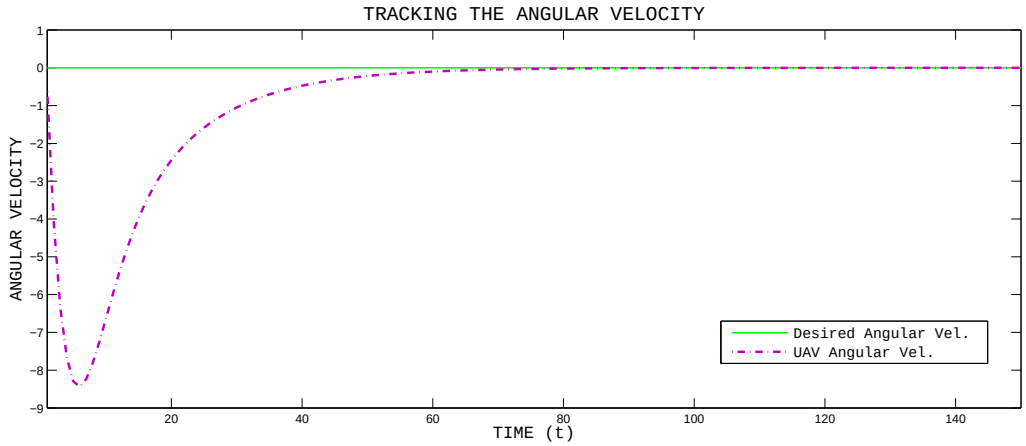

Figure 6.7: Angular Velocity Tracking - w_d

The desired linear velocity is constant and away from zero. This is logical since tracking would imply converging to a moving trajectory. The angular velocity on the other hand approaches zero asymptotically. This is because the desired trajectory is a straight line path, thus the rate of change of the UAV's orientation has to be zero since the desired orientation is constant.

6.1.2 Quadrant II Evasion

Now the orientation which the UAV wishes to track is between 90^o and 180^o. The initial position and orientation of the UAV is chosen as $x(1) = 2.5$, $y(1) = 2$,and $\theta_s(1) = 160^o$. The desired initial states are given as $x_d(1) = 1.1$, $y_d(1) = 1.1$, $\theta_d = 105^o$, $\tilde{v}_d = 1$, and $w_d = 0$.

Evasion Simulation

The desired orientation is in the second quadrant, the figure below shows the UAV evading an approaching predator.

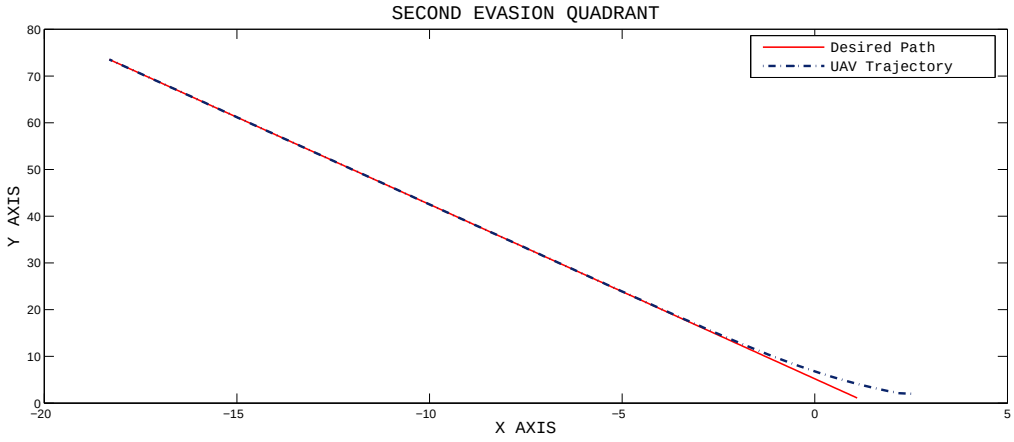

Figure 6.8: Second Quadrant Evasion Simulation

The figure shows that UAV tracks the desired straight line trajectory when subjected to the proposed controller. The tracking sequence is expanded in the zoomed-in version below.

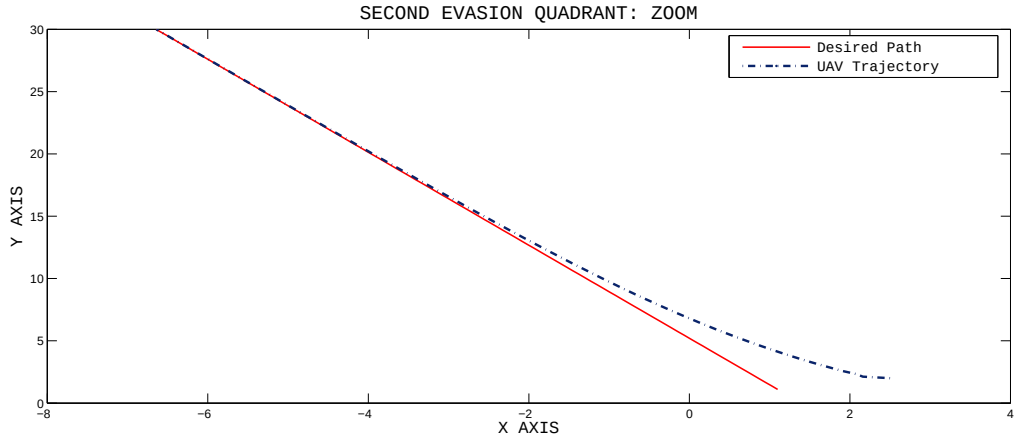

Figure 6.9: Second Quadrant Evasion - Zoomed View

The nonholonomic UAV is seen tracking the desired straight line path at a angle of $\theta_d = 105^o$ in the second quadrant. The nonholonomic nature of the system again becomes evident.

Tracking Error

The tracking error states - ζ_x, ζ_y, and ζ_θ - are plotted below.

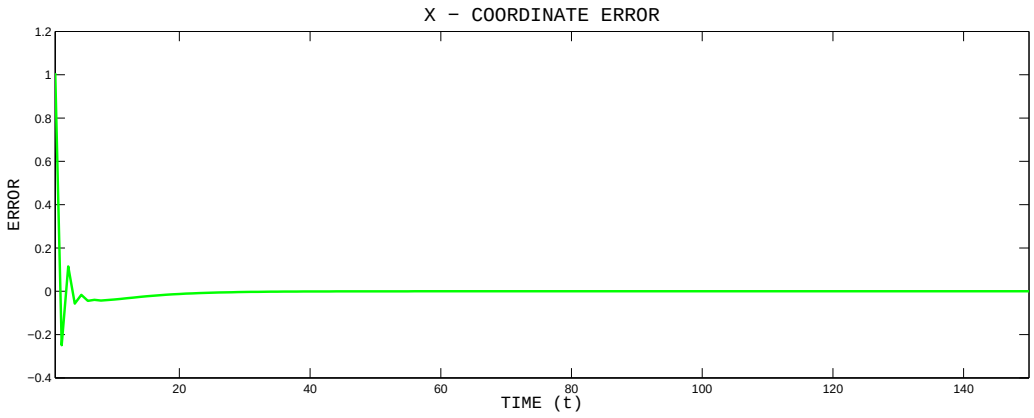

Figure 6.10: X-Coordinate Error - ζ_x

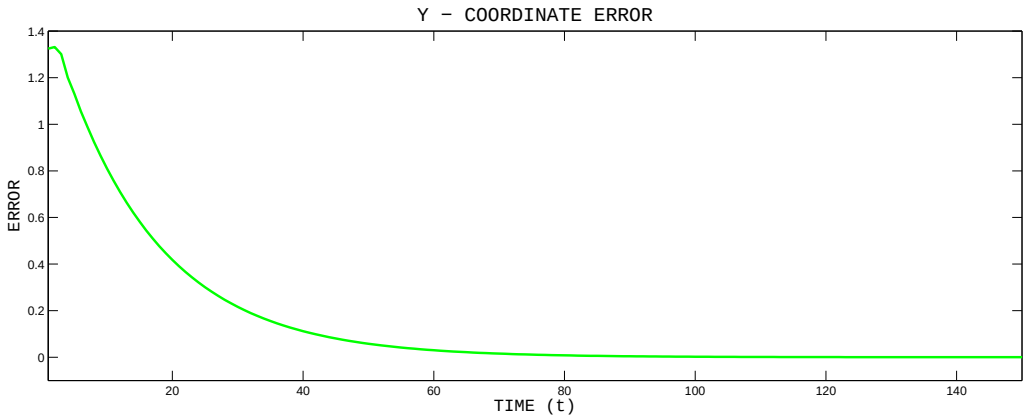

Figure 6.11: Y-Coordinate Error - ζ_y

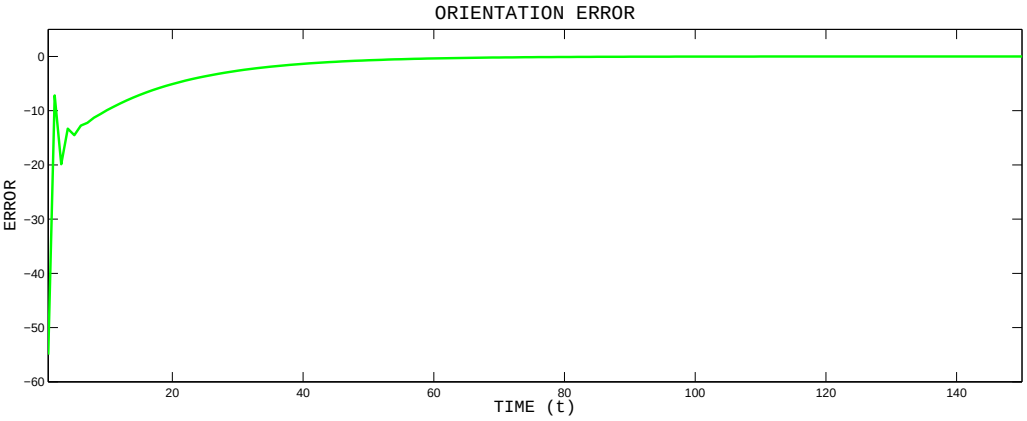

Figure 6.12: Orientation Error - ζ_θ

The tracking error states ζ_x, ζ_y and ζ_θ are seen to asymptotically approach zero, thus showing tracking ability of the controller. It also shows that the further away the initial states of the UAV are from the desired states, the longer it takes the UAVs to converge to the desired trajectory.

Velocity Tracking

The desired linear and angular velocities we wish to track are $\tilde{v}_d = 1$, and $w_d = 0$ respectively. The figures below show the velocity tracking.

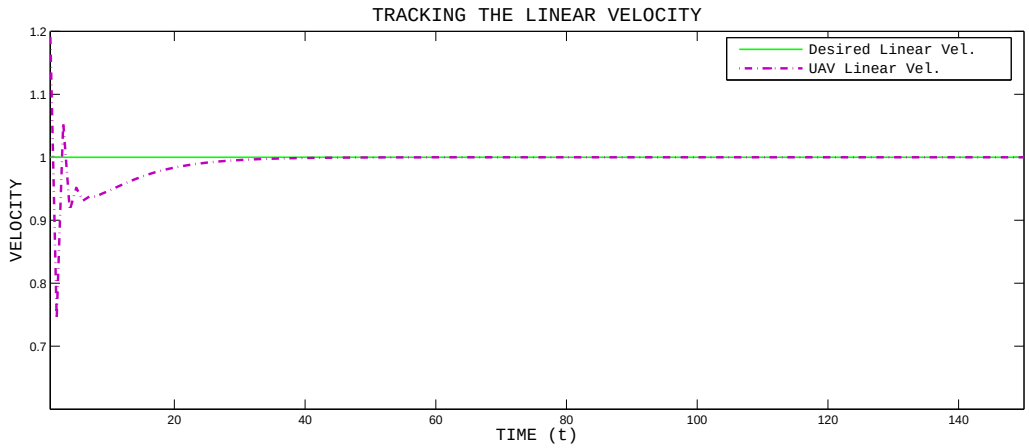

Figure 6.13: Linear Velocity Tracking - v_d

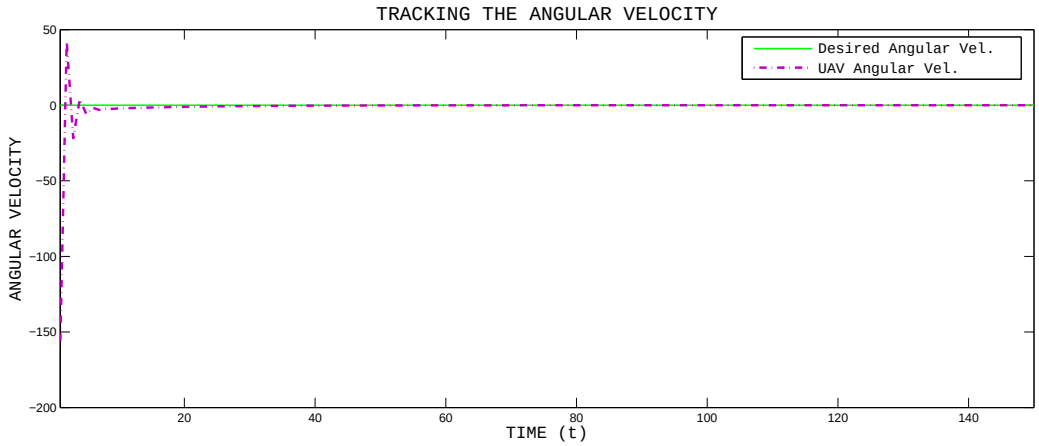

Figure 6.14: Angular Velocity Tracking - w_d

The desired linear and angular velocities are tracked asymptotically, showing the effectiveness of the controller.

6.1.3 Quadrant III Evasion

The desired orientation of the UAV in the third quadrant is between -180^o and -90^o. The initial position and orientation of the UAV is chosen as $x(1) = 2.5$, $y(1) = 2$,and $\theta_s(1) = -85^o$ respectively. The desired initial states are given as $x_d(1) = 2.5$, $y_d(1) = 1.1$, $\theta_d = -105^o$, $\tilde{v}_d = 1$, and $w_d = 0$.

Evasion Simulation

The desired orientation is in the third quadrant, the figure below shows the UAV evading an approaching predator.

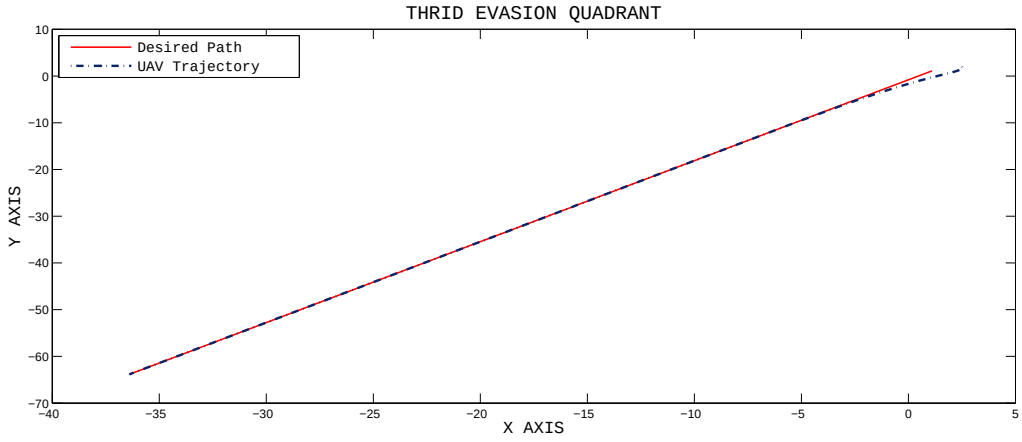

Figure 6.15: Third Quadrant Evasion Simulation

The UAV is seen tracking the desired straight line trajectory under the influence of the proposed controller. The tracking sequence is expanded in the zoomed-in version below.

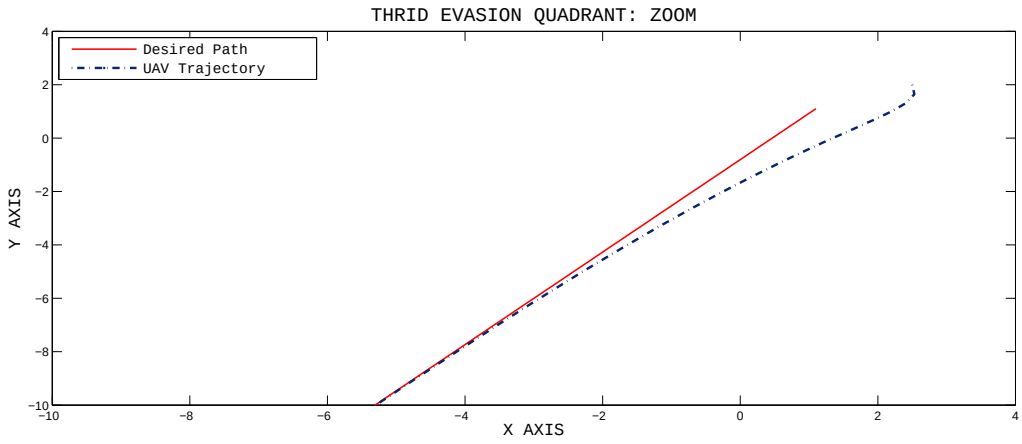

Figure 6.16: Third Quadrant Evasion: Zoomed View

The inability of UAV to directly assume its desired trajectory as seen in the figure emphasizes the effect of the UAV's nonholonomic constraints.

Tracking Error

Plots of the error states versus time - ζ_x, ζ_y, and ζ_θ - are drawn below.

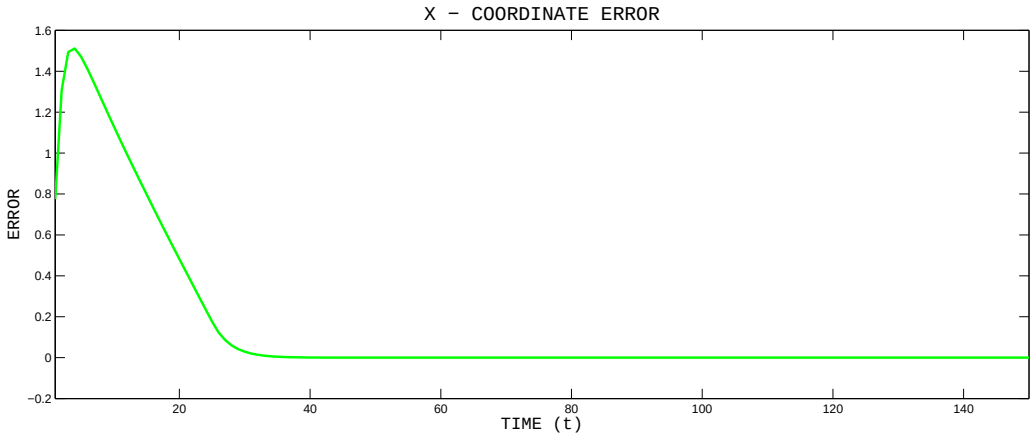

Figure 6.17: X-Coordinate Error - ζ_x

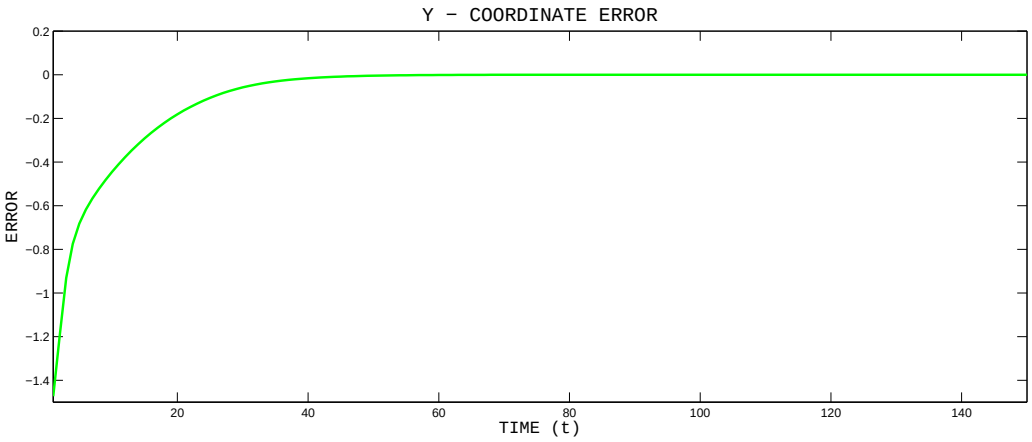

Figure 6.18: Y-Coordinate Error - ζ_y

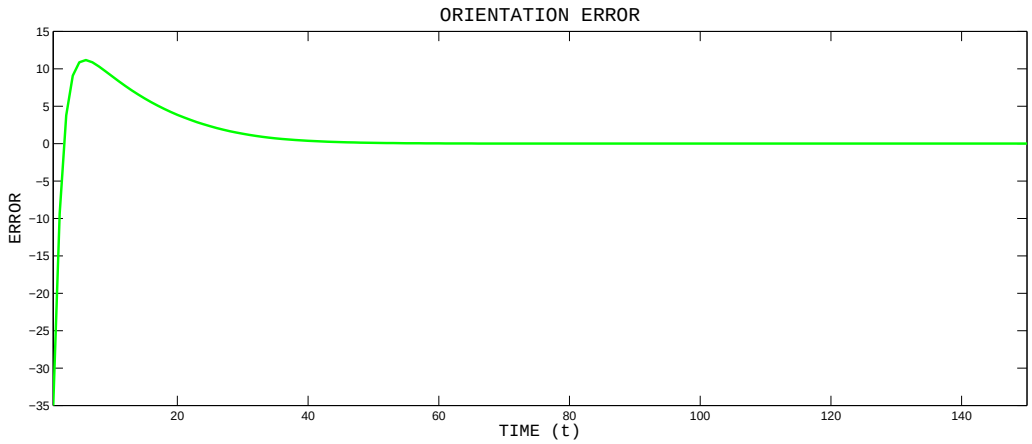

Figure 6.19: Orientation Error - ζ_θ

The tracking error states ζ_x, ζ_y and ζ_θ are seen to asymptotically approach zero, again showing tracking ability of the controller.

Velocity Tracking

The desired linear and angular velocities we wish to track are $\tilde{v}_d = 1$, and $w_d = 0$ respectively. The figures below show the velocity tracking.

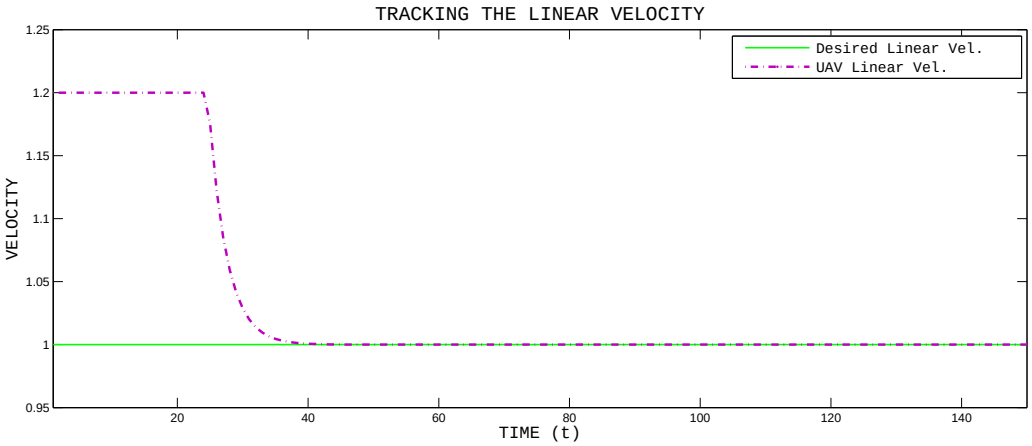

Figure 6.20: Linear Velocity Tracking - v_d

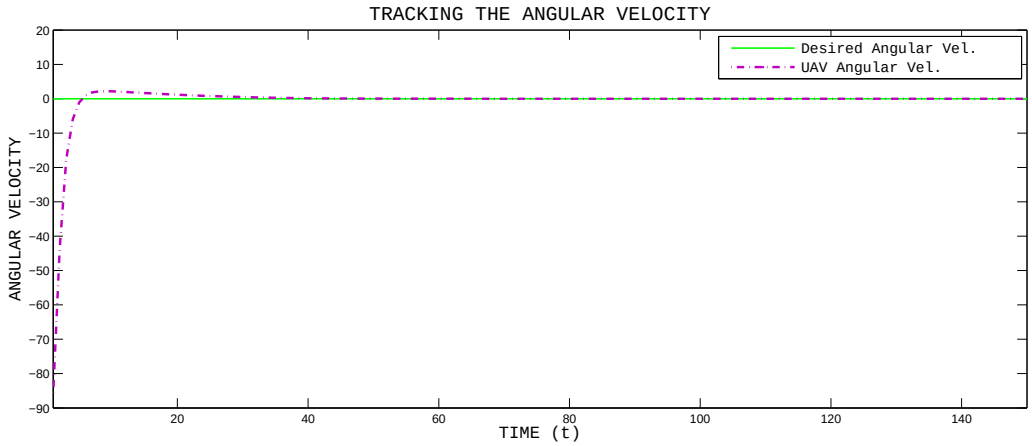

Figure 6.21: Angular Velocity Tracking - w_d

The desired linear and angular velocities are tracked asymptotically, showing the effectiveness of the controller.

6.1.4 Quadrant IV Evasion

Last, in the fourth quadrant, the UAV tracks a desired angle between -90^o and 0^o. The initial position and orientation of the UAV is chosen as $x(1) = 2.5$,

$y(1) = 2$,and $\theta_s(1) = 160^o$. The desired initial states are given as $x_d(1) = 1.1$, $y_d(1) = 1.1$, $\theta_d = -60^o$, $\tilde{v}_d = 1$, and $w_d = 0$.

Evasion Simulation

The figure below shows the UAV evading an approaching predator.

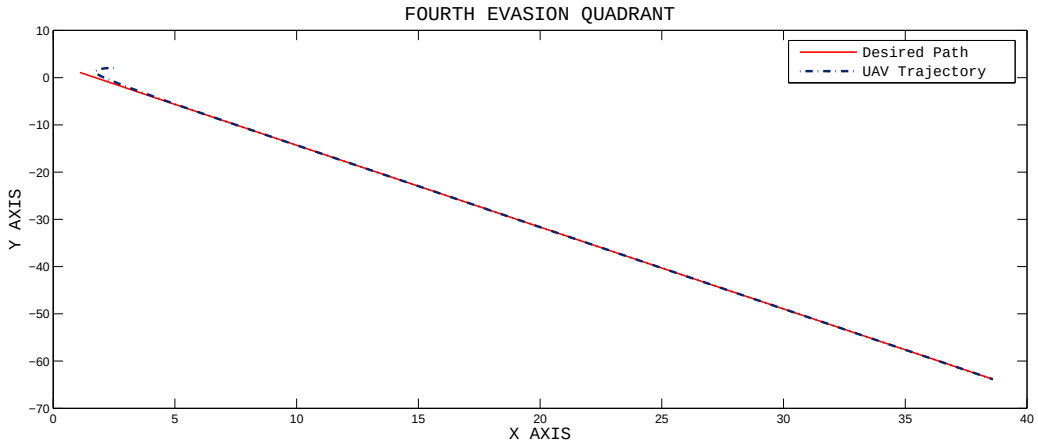

Figure 6.22: Fourth Quadrant Evasion Simulation

The UAV tracks the desired straight line trajectory. The tracking sequence is better observed in the zoomed-in figure below.

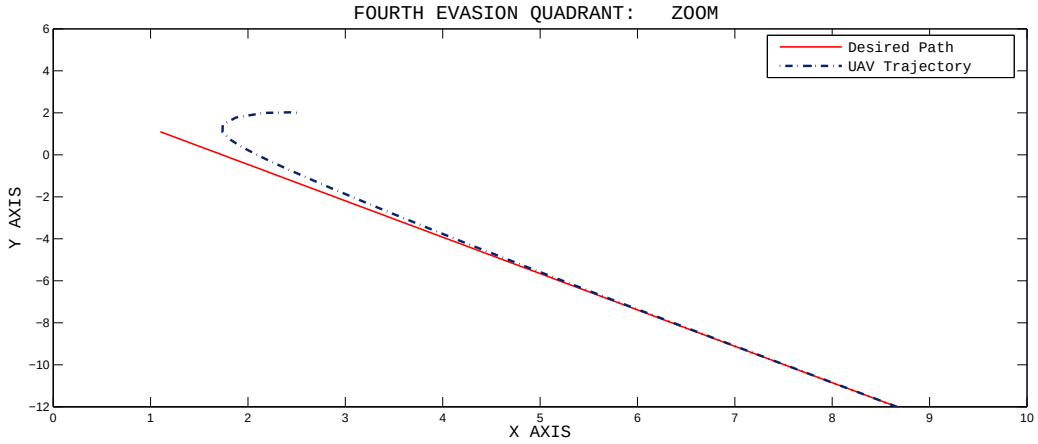

Figure 6.23: Fourth Quadrant Evasion - Zoomed View

The nonholonomic UAV is seen tracking the desired straight line path at a angle of $\theta_d = -60^o$ in the fourth quadrant. The nonholonomic nature of the system becomes evident.

Tracking Error

The tracking error for position states - ζ_x and ζ_y - in the fourth quadrant are plotted. The tracking orientation error - ζ_θ - is also plotted.

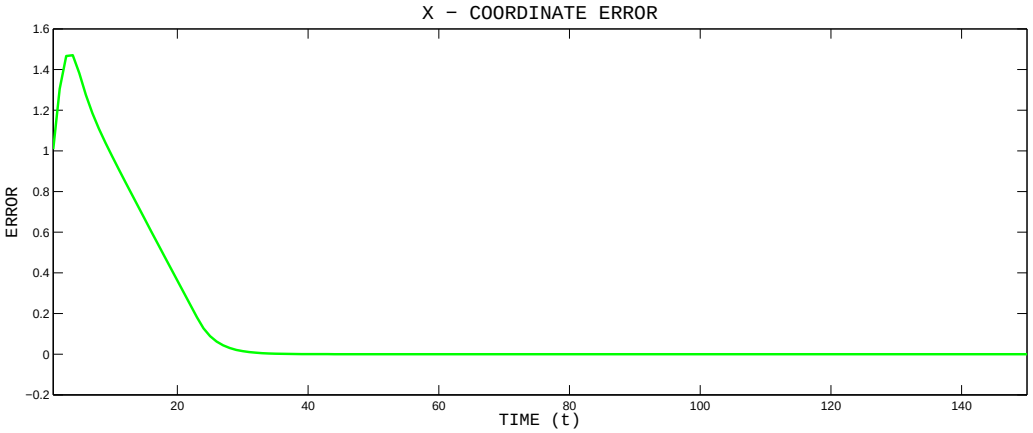

Figure 6.24: X-Coordinate Error - ζ_x

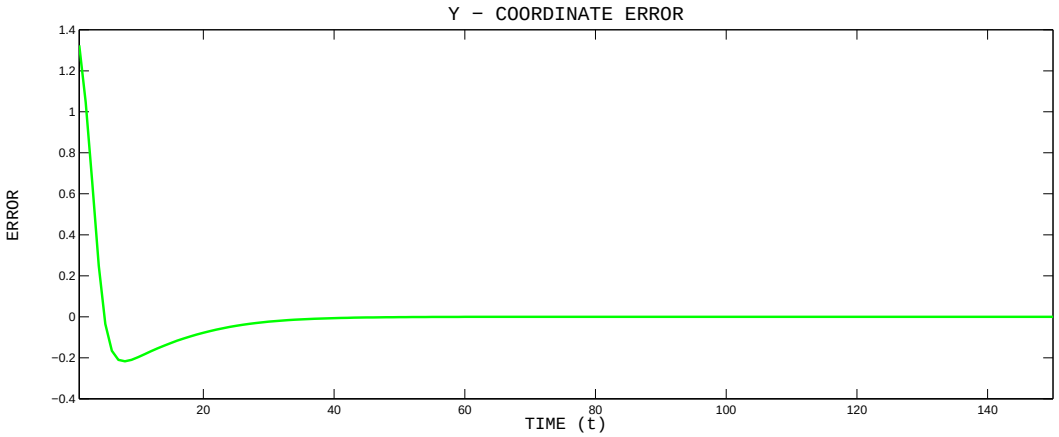

Figure 6.25: Y-Coordinate Error - ζ_y

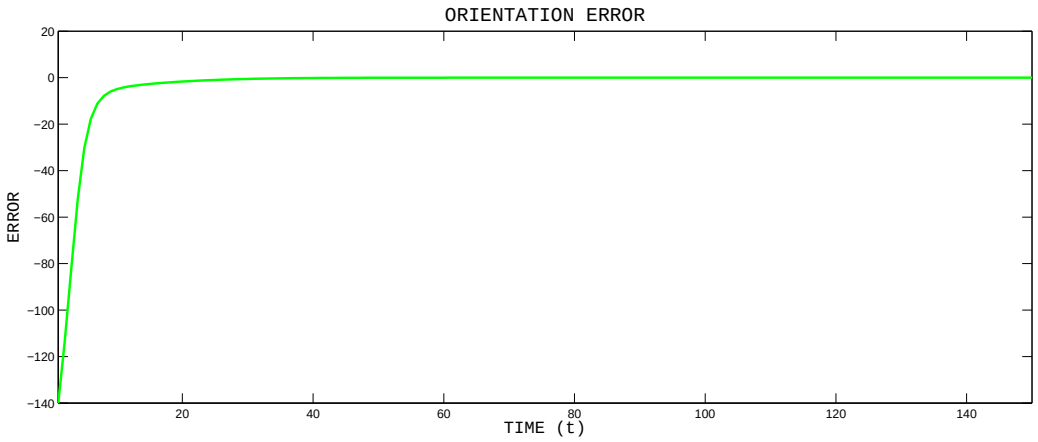

Figure 6.26: Orientation Error - ζ_θ

The tracking error states ζ_x, ζ_y and ζ_θ are seen to asymptotically approach zero.

Velocity Tracking

The desired linear and angular velocities we wish to track are $\tilde{v}_d = 1$, and $w_d = 0$ respectively. The figures below show the velocity tracking.

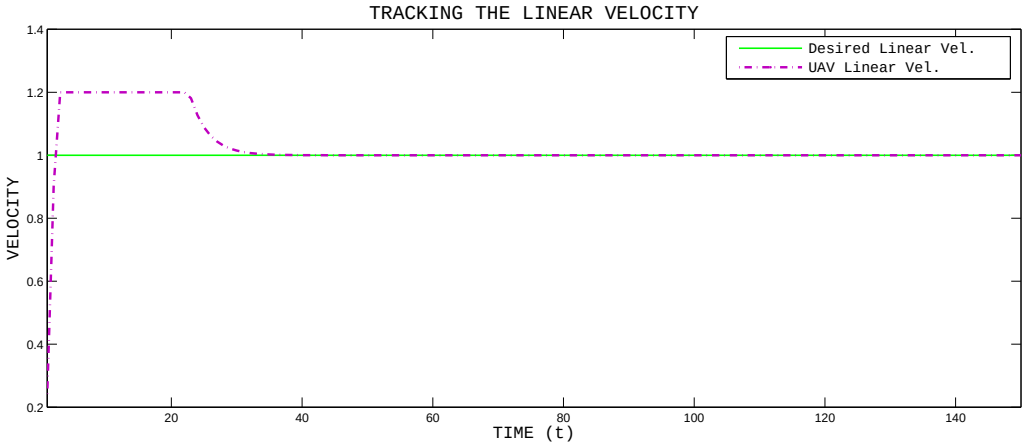

Figure 6.27: Linear Velocity Tracking - v_d

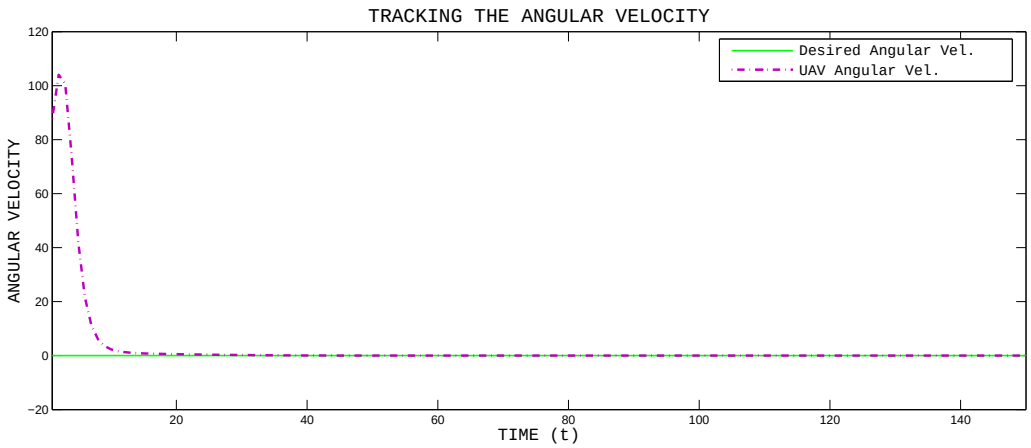

Figure 6.28: Angular Velocity Tracking - w_d

Once again, the linear and angular velocities are shown to converge to the desired velocities.

The single UAV analysis is used to show the effectiveness of the controller on each UAV. The convergence of the error states to zero and also the system states to their desired states as seen in the figures makes the previous statement obvious. We can infer that each UAV will successfully track the desired straight line trajectory when in an environment with many other UAVs. We will show this in the next simulations.

6.2 Two Sample UAVs Foraging & Evasion

This simulation is done to make the foraging and evasion processes of the UAVs imaginative. We show two UAVs ○ approaching a target △. The adversary UAV □ approaches the foraging UAVs and causes them to evade at different orientations. The enemy decides to attack UAV 1, causing UAV 1 to continue evading. UAV

2 on the other hand is out of danger, thus it re-tracks the target and focuses on approaching the target.

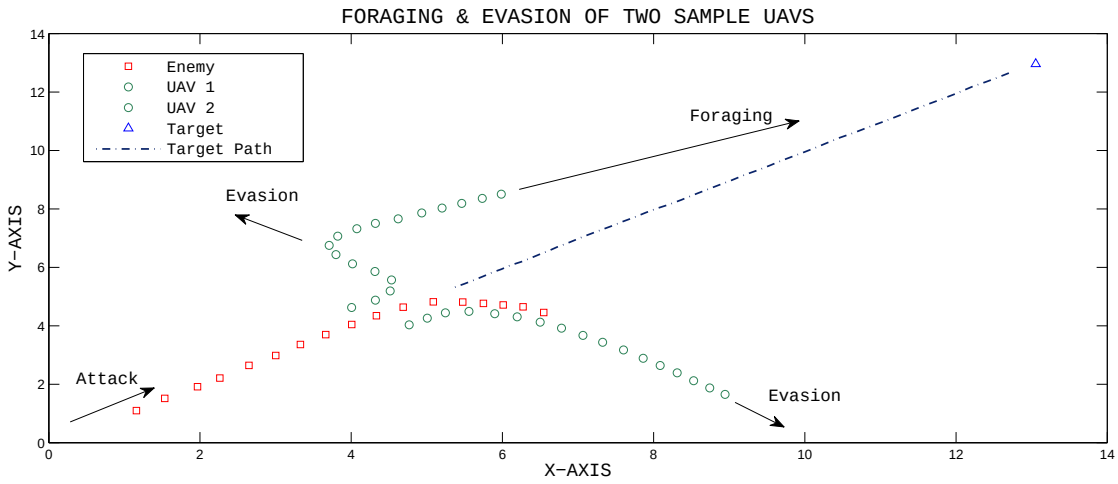

Figure 6.29: Foraging & Evasion of Two UAVs

6.3 Multiple UAV Simulation

In this section, we consider the entire closed loop system schematic shown in fig.(5.2). However, we consider the closed loop with all the forty-nine UAVs moving towards a target by tracking the path generated by the fish-prey algorithm. These fleet of UAVs are also ready to evade an adversary using the designed closed loop controller. First, we show trajectory tracking of the fleet of UAVs influenced by on the fish-prey algorithm. Then, we include a predator to observe the evasion strategies of the fleet in the presence of danger. Table(6.1) shows some the values used during the simulation. The controller parameters using during evasion remain the same as in the single UAV evasion case while $k_4 = 0.5$, $k_5 = 75$, and $k_6 = 20$ are used during the foraging phase i.e. when the UAV is not in danger.

6.3.1 Foraging Phase only

The figures below show the series of behaviors exhibited by the fleet of UAVs in the absence of danger. Here's the legend used for the figures.

◯ ≡ Fleet of UAVs

□ ≡ Enemy UAV

△ ≡ Target

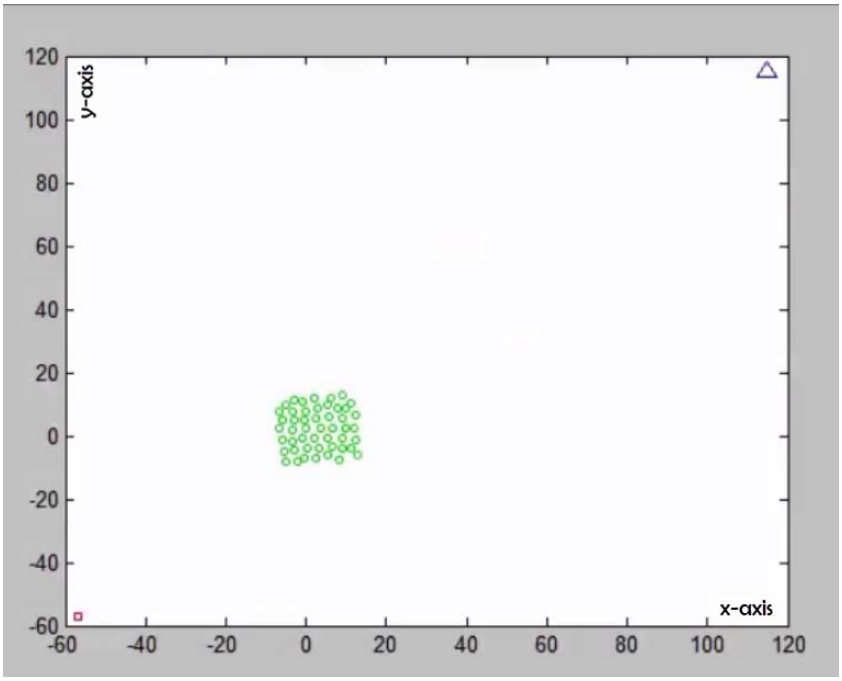

Figure 6.30: Foraging Behavior 1

The fleet of UAVs are initialized, all 49 in number.

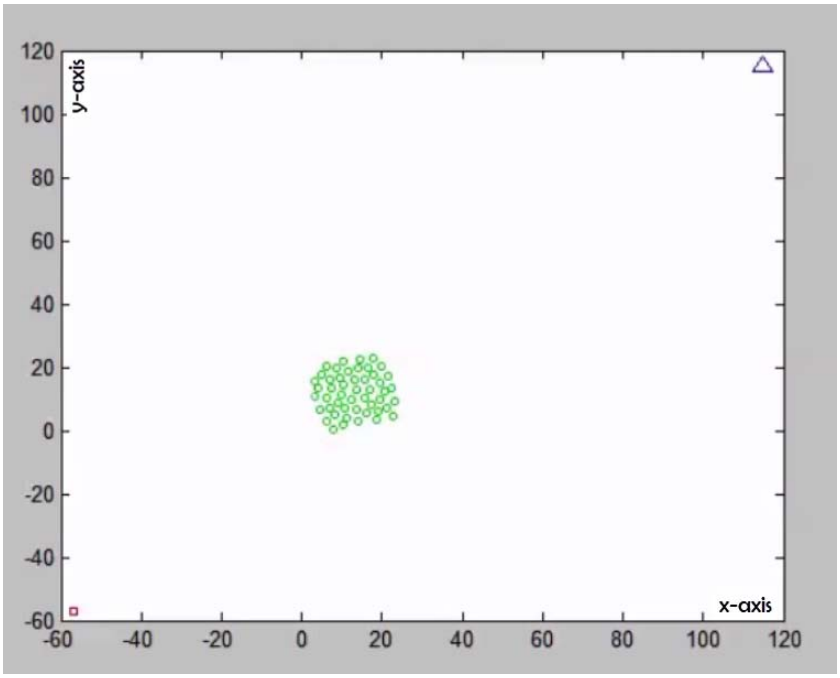

Figure 6.31: Foraging Behavior 2

The UAVs are seen advancing towards the target, located in the top right corner of the figure.

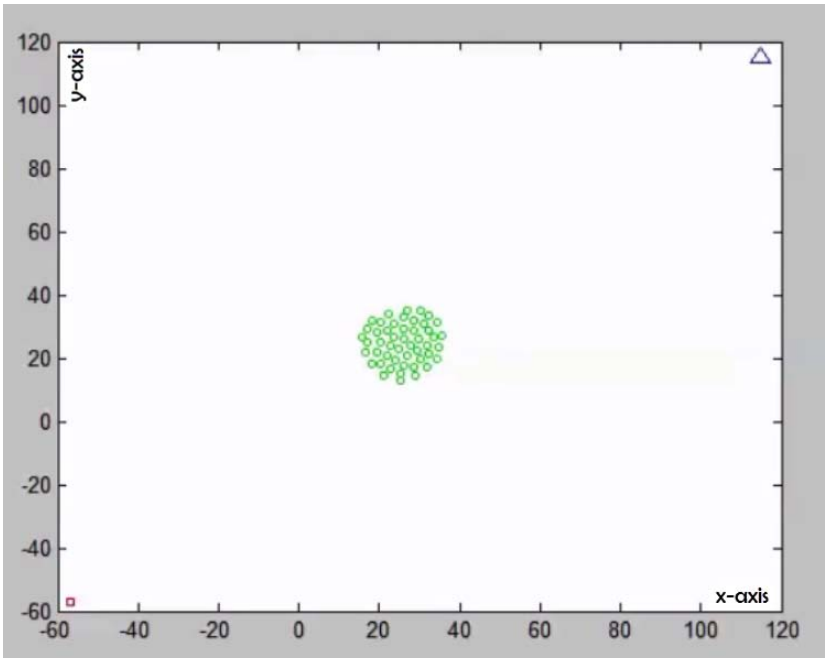

Figure 6.32: Foraging Behavior 3

The fleet self-organize themselves using the fish-prey algorithm. It is seen that they maintain a safe distance from each other, while moving coherently.

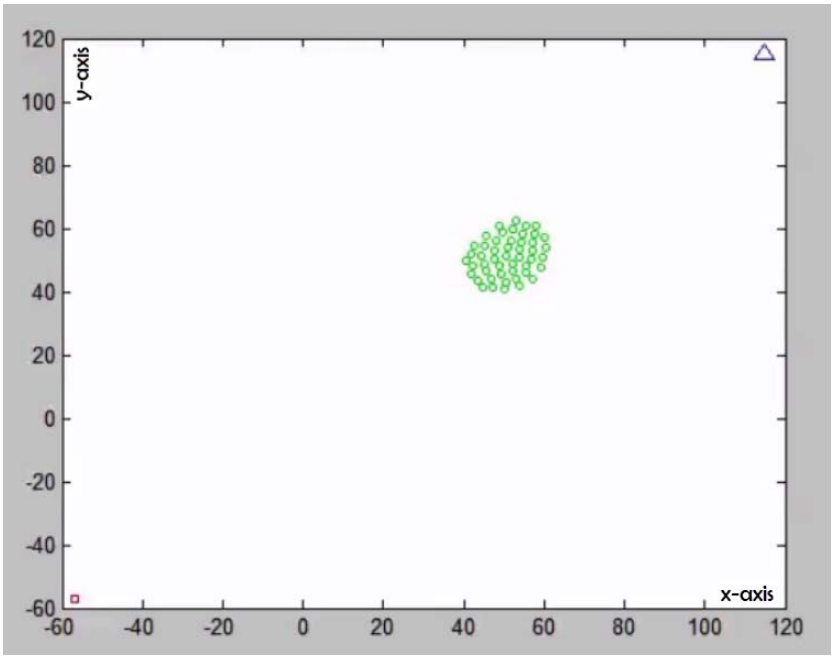

Figure 6.33: Foraging Behavior 3

This last figure shows the fleet of UAVs in a well organized formation. The effect of the biological foraging behaviors of a school of fish are imposed on the movement of a fleet of UAVs to a desired target. It also shows that the self organizing behaviors are more emphasized as the UAVs advances with time. This is obvious from the shape transition of the entire fleet formation.

6.3.2 Foraging and Evasion

We will now add mobility of the adversary to the simulation. The biologically inspired foraging and evading behaviors are observed simultaneously.

The predator/enemy UAV approaches the fleet of UAVs with its approach orien-

tation in the first quadrant. The following figures show the responses of the UAVs when approached by danger.

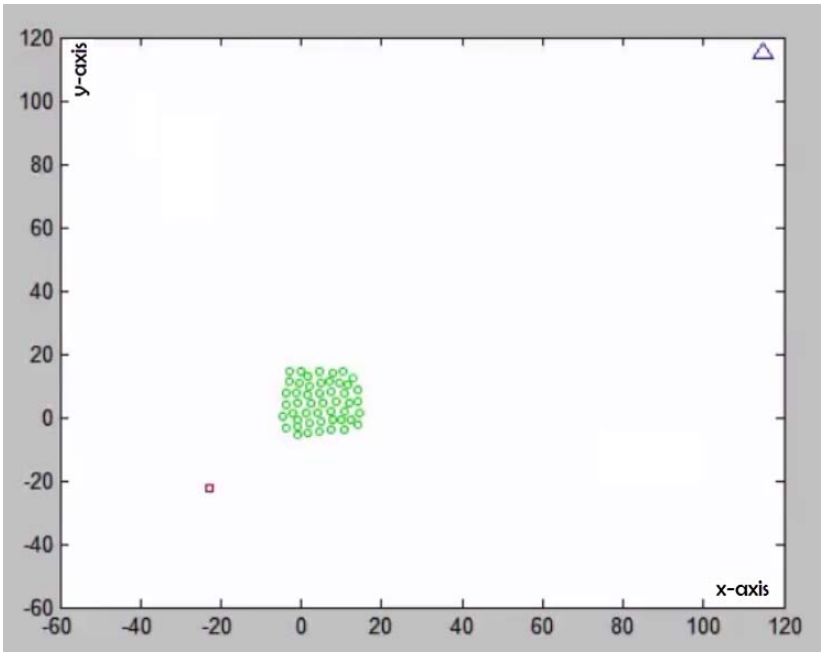

Figure 6.34: Q1 - Foraging & Evading Behavior 1

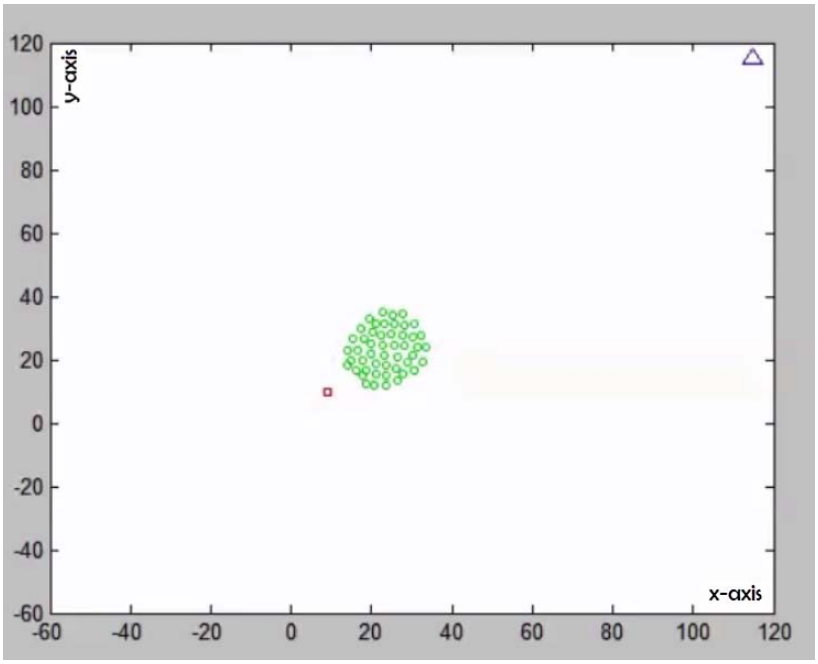

Figure 6.35: Q1 - Foraging & Evading Behavior 2

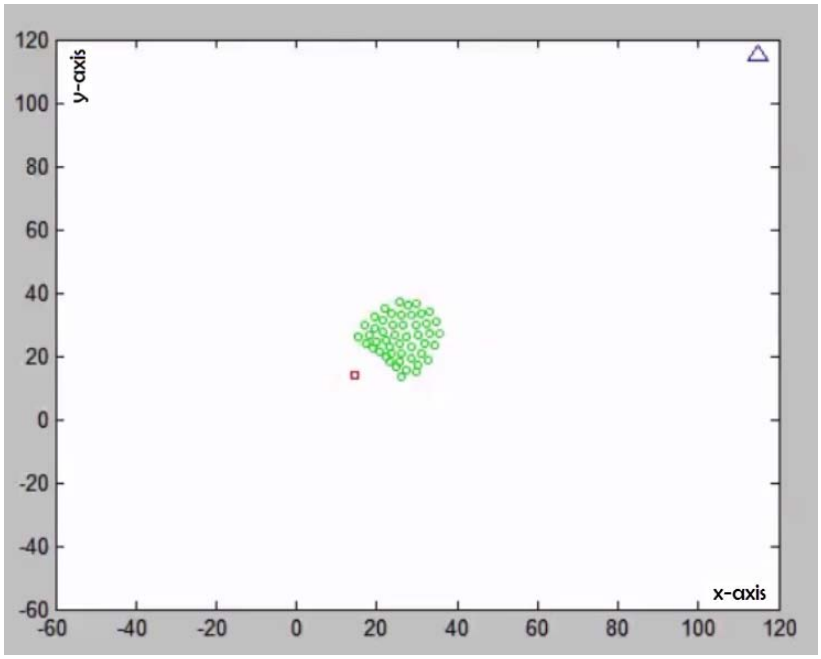

Figure 6.36: Q1 - Foraging & Evading Behavior 3

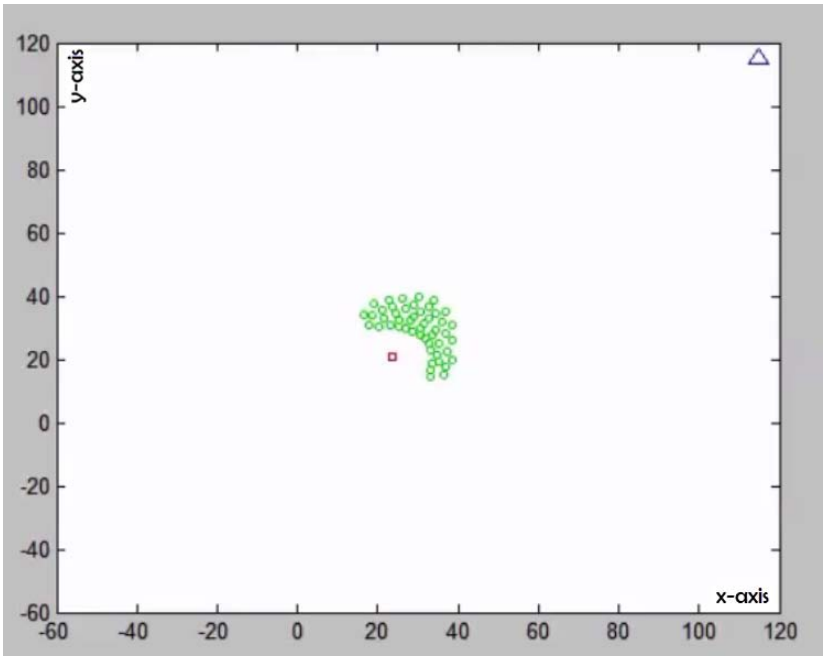

Figure 6.37: Q1 - Foraging & Evading Behavior 4

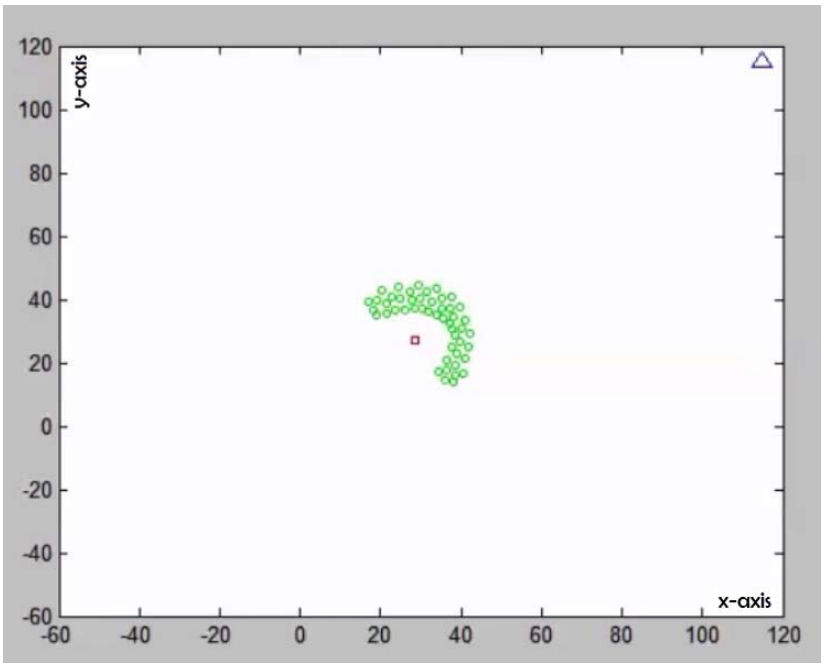

Figure 6.38: Q1 - Foraging & Evading Behavior 5

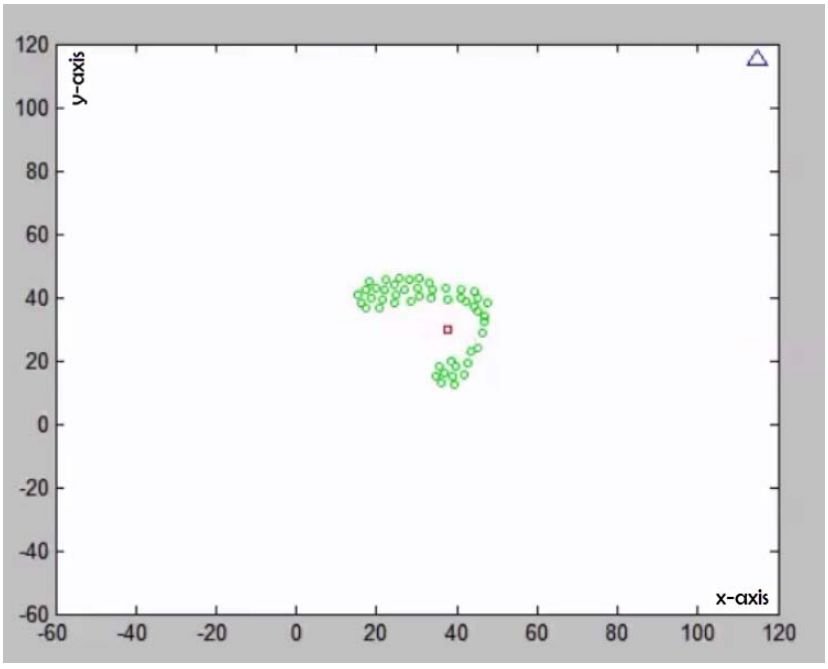

Figure 6.39: Q1 - Foraging & Evading Behavior 6

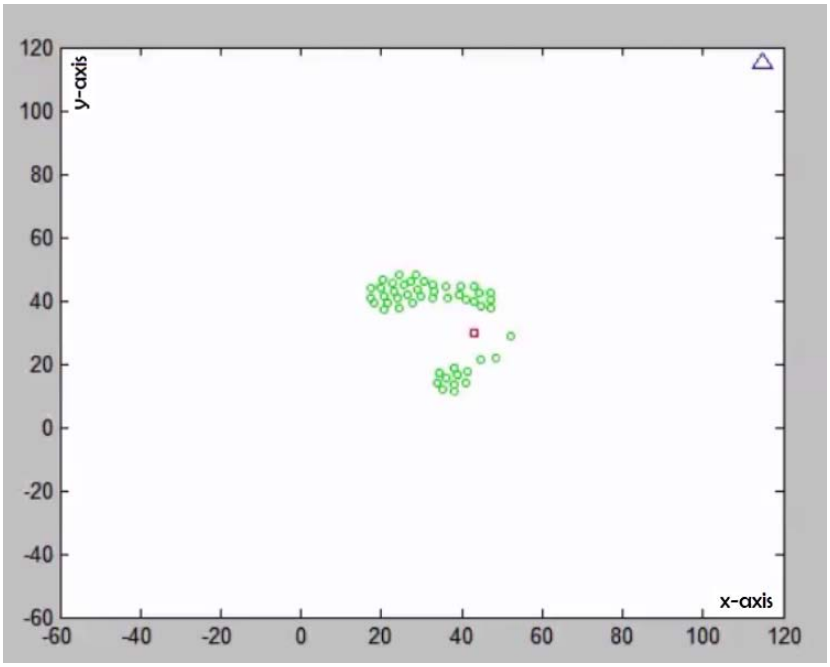

Figure 6.40: Q1 - Foraging & Evading Behavior 7

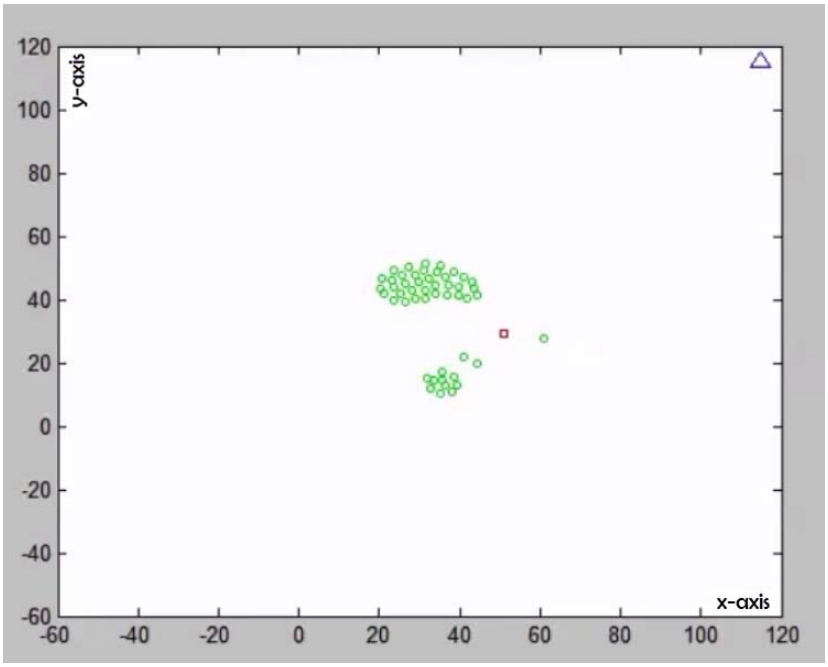

Figure 6.41: Q1 - Foraging & Evading Behavior 8

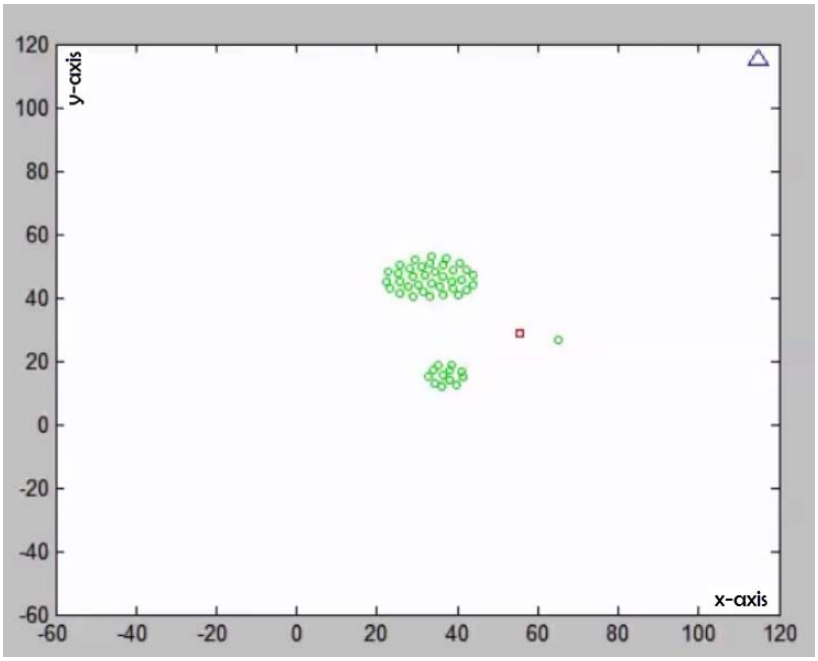

Figure 6.42: Q1 - Foraging & Evading Behavior 9

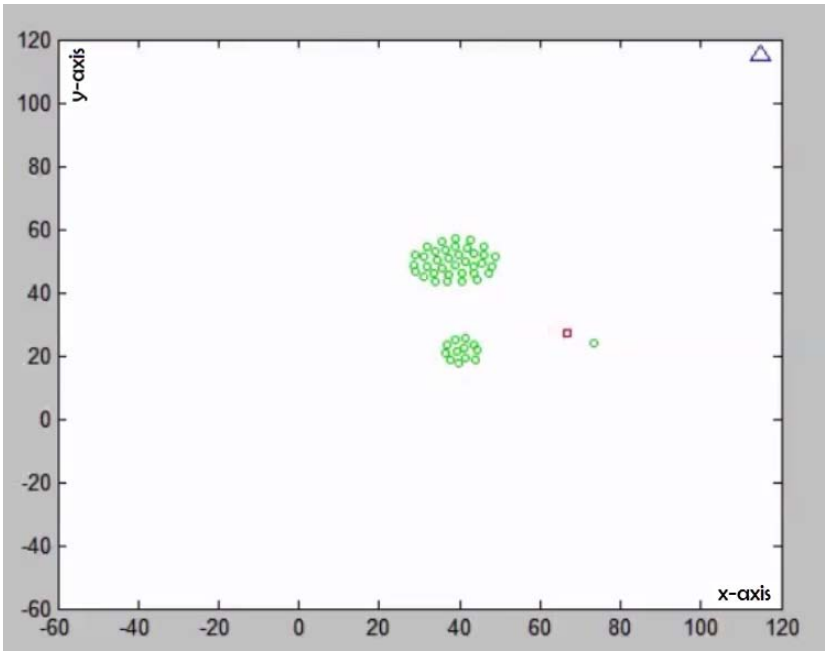

Figure 6.43: Q1 - Foraging & Evading Behavior 10

The adversary UAV approaches the fleet, locating the closest UAV of fleet. In the fig.(6.35) the UAVs detect the presence of the enemy and begin to evade.

The spread out and also avoid collision during the process. In fig.(6.39) the fleet becomes fragmented to ensure that it is away from the enemy UAV. As some UAVs become detacted from the fleet (fig.(6.40)), once they are away from danger they begin to regroup and move in coherent motion as seen in figures (6.41) and (6.42). Finally the last shows the enemy UAV on the pursuit of only one UAV while the others in the fleet have successfully evaded and are approaching the food.

Another simulation showing the evading and foraging behaviors is carried out; however this time, the enemy vehicle approaches from the third quadrant angle.

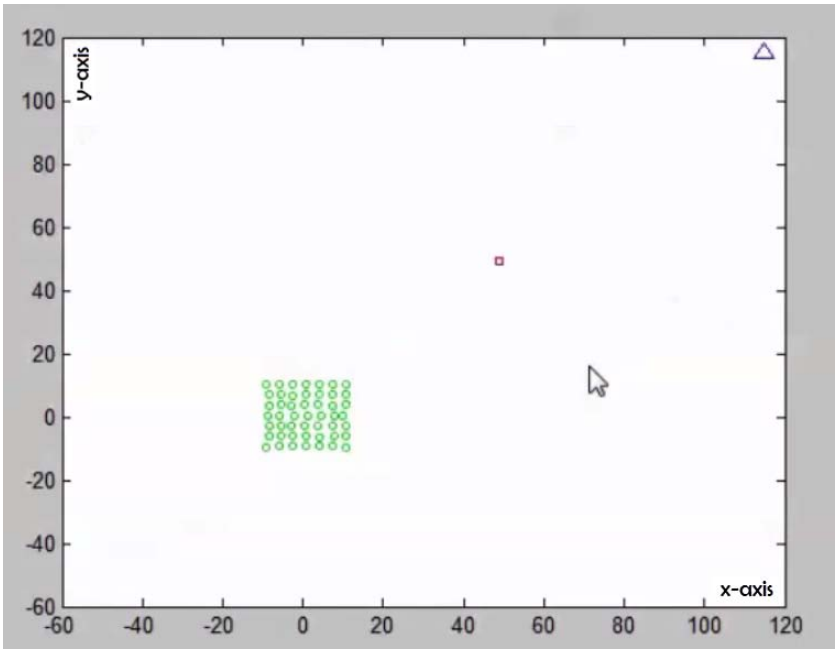

Figure 6.44: Q2 - Foraging & Evading Behavior 1

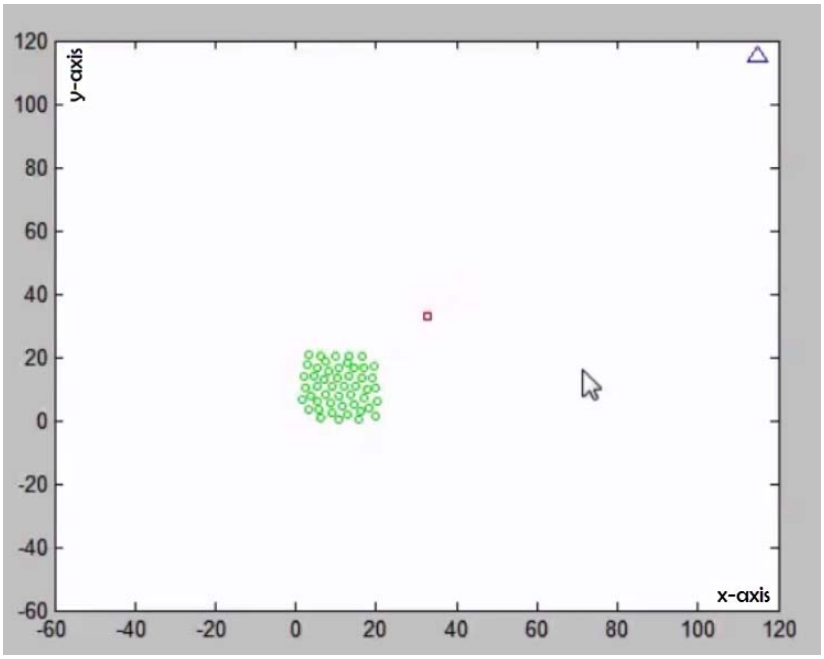

Figure 6.45: Q2 - Foraging & Evading Behavior 2

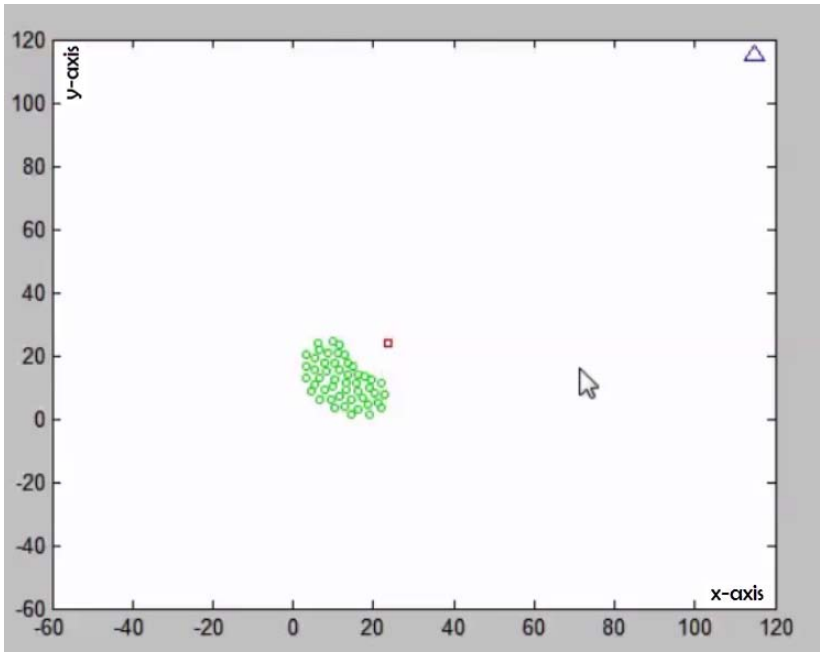

Figure 6.46: Q2 - Foraging & Evading Behavior 3

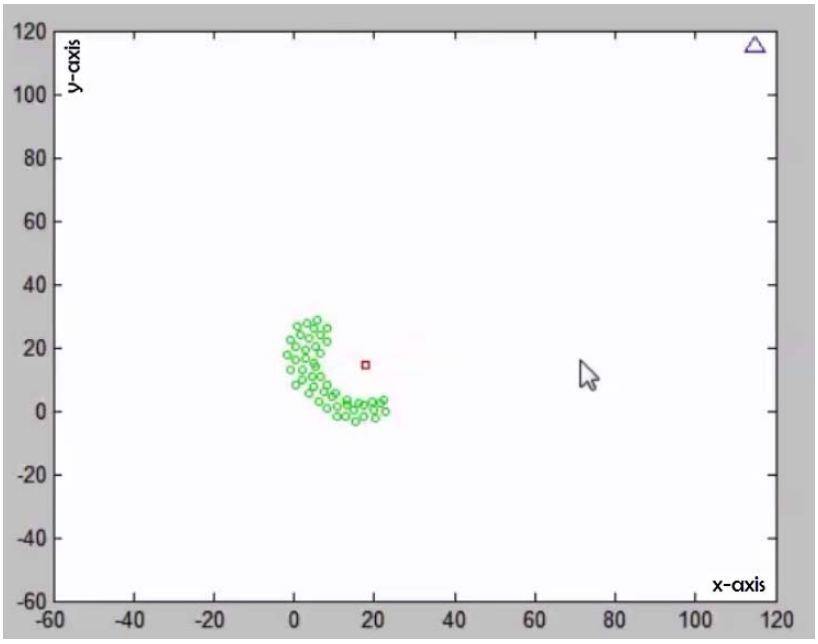

Figure 6.47: Q2 - Foraging & Evading Behavior 4

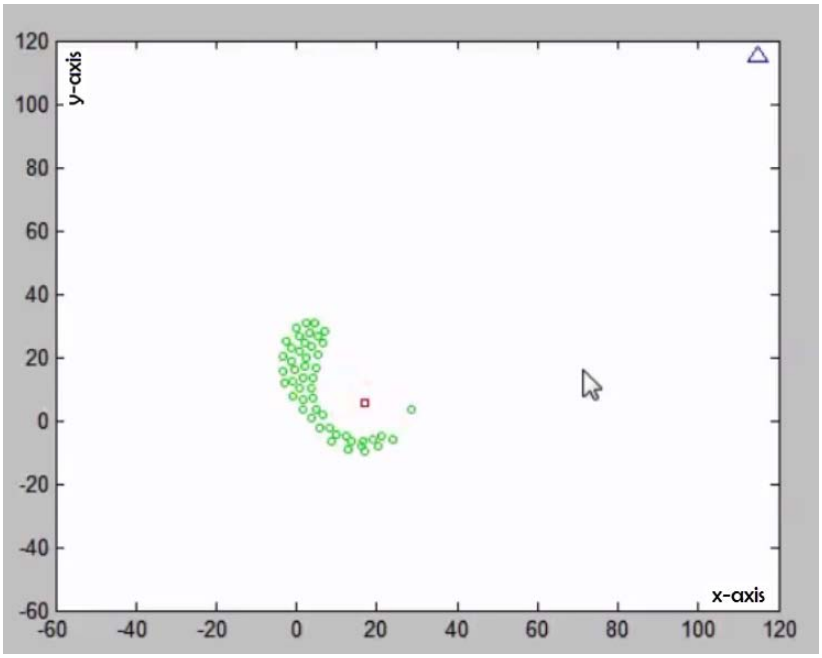

Figure 6.48: Q2 - Foraging & Evading Behavior 5

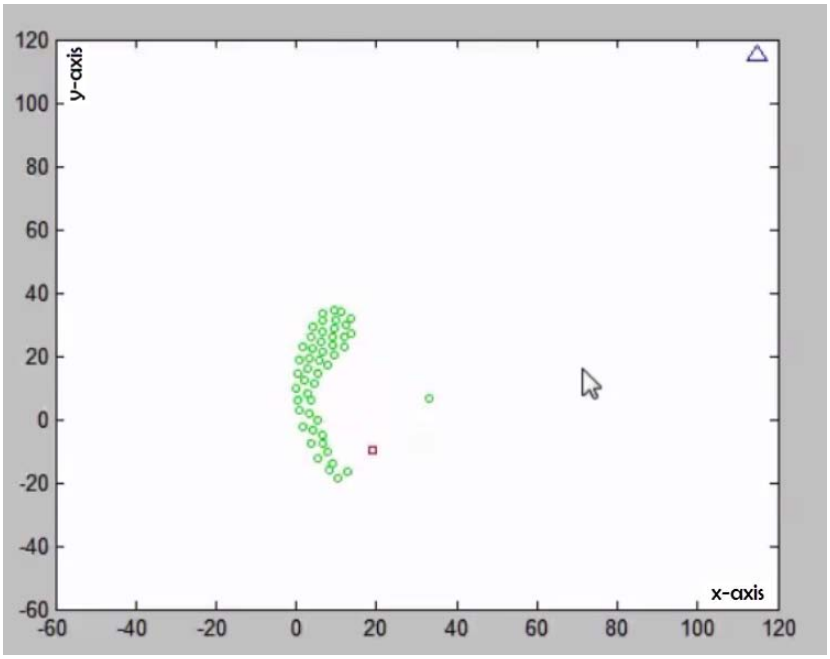

Figure 6.49: Q2 - Foraging & Evading Behavior 6

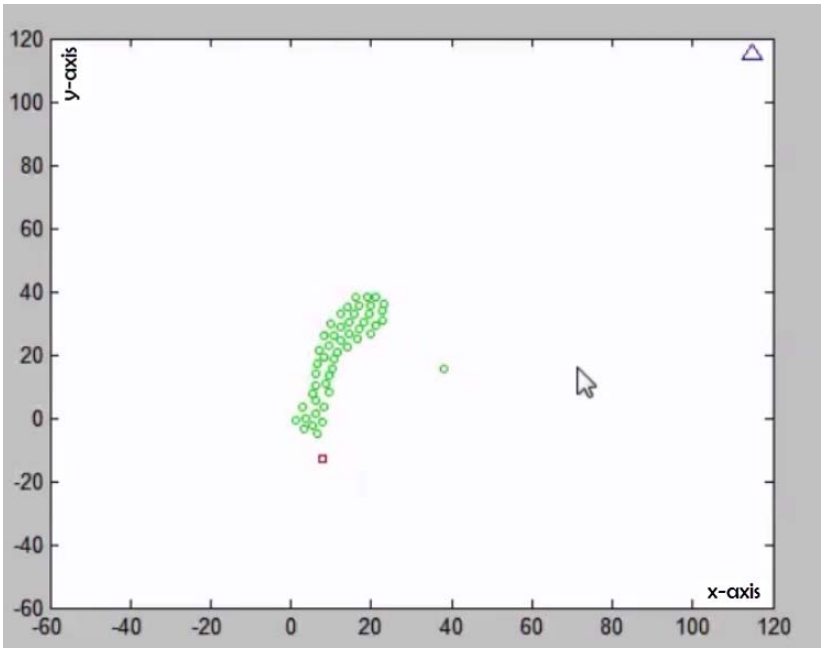

Figure 6.50: Q2 - Foraging & Evading Behavior 7

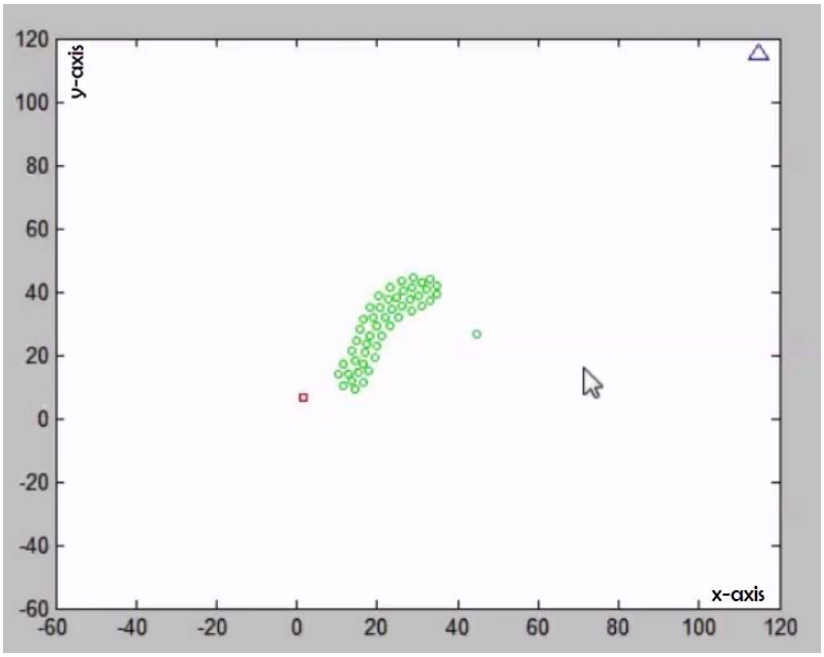

Figure 6.51: Q2 - Foraging & Evading Behavior 8

The enemy UAV approaches the fleet from the opposite direction of the fleet. In fig.(6.46) the fleet begin to evade the enemy after sensing it within their communications radius. Once again, the UAVs spread out by tracking straight lines tangential to the direction of the approaching UAV. By doing this, the UAVs evade as seen in fig.(6.48). One of the UAVs in the fleet however, becomes fragmented from the formation while the others maintain their cohesion and adaptability in the process of evasion. The spread out UAVs are seen to reunite in fig.(6.51) with the detached UAV approaching the fleet to ensure coherence.

CHAPTER 7

CONCLUSION AND FUTURE STUDY

In reality, UAVs are currently not equipped enough to carry out evasion strategies for air combat. In fact, the idea that air combat can be fully autonomous may seem like fiction to many. However, we envision a future of fully autonomous air forces. We proposed a tracking control strategy to achieve evasion in this research. To enhance the evasion strategies, other tracking controllers can be researched and implemented into the fleet of UAVs, then studied for performance.

The fleet of UAVs used the biological behavior of a school of fish to define its trajectory when attacking the target. Other biological behaviors such as the behaviors of swarming birds and bees can be explored for applicability. These behaviors can also be used in future research to define tracking trajectories which a proposed controller would then track.

The fish-prey algorithm used in this research assumes the adaptive network is

strongly connected. In the future, this assumption can be relaxed. The possibility of achieving concurrent attack and evasion when the network is weakly connected can also be explored.

Future research can also include the practical application of the fish-prey distributed biological algorithm to a fleet of UAVs. Such research along with physical technological advancement of UAV designs would put a spot light on the actual viability of the algorithms.

The autonomy of unmanned aerial vehicles is becoming more important in collecting information for surveillance, rescue and military applications. Our research spans out from this. Currently, most UAVs carry out tasks with assigned targets on the ground. They are also capable of staying away from gun fire by aviating at high altitudes. For UAVs to be able to also win against an enemy that is airborne, UAVs will need to be physically redesigned as Sastry et. al [11] mentioned. However, UAVs can be made more intelligent using biologically inspired algorithms such as the one proposed in this research. Further research in expanding these concepts indeed do have a great potential in 3D space unstructured environment and obstacle avoidance capabilities.

REFERENCES

[1] T. Melito, "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] P. Misener, "FAA: Amazon petition for exemption," Amazon, United States, Tech. Rep., July 2014.

[3] R. Brockett, R. Millmann, and H. Sussmann (Eds.), "Asymptotic stability and feedback stabilization, differential geometric control theory," *Progress in Mathematics*, 1983.

[4] M. Sampei, H. Kiyota, M. Koga, and M. Suzuki, "Necessary and sufficient conditions for transformation of nonholonomic system into time-state control form," in *Proceedings of the 35th IEEE Conference on Decision and Control, 1996*, vol. 4. IEEE, December 1996, pp. 4745–4746 vol.4.

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

[6] E. Valtolina and A. Astolfi, "Local robust regulation of chained systems," *Systems and Control Letters*, vol. 49, no. 3, pp. 231–238, 2003.

[7] A. Bloch and S. Drakunov, "Tracking in nonholonomic dynamic systems via sliding modes," in *Proceedings of the 34th IEEE Conference on Decision and Control, 1995.* Singapore: IEEE, December 1995, pp. 2103–2106 vol.3.

[8] Y.-P. Tian and S. Li, "Exponential stabilization of nonholonomic dynamic systems by smooth time-varying control," *Automatica*, vol. 38, no. 7, pp. 1139–1146, 2002.

[9] M. Sampei, "A control strategy for a class of nonholonomic systems - time-state control form and its application," in *Proceedings of the 33rd IEEE Conference on Decision and Control, 1994*, vol. 2. IEEE, December 1994, pp. 1120–1121 vol.2.

[10] R. W. Whittlesey, 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] J. Sprinkle, J. Eklund, H. Kim, and S. Sastry, "Encoding aerial pursuit/evasion games with fixed wing aircraft into a nonlinear model predictive tracking controller," in *43rd IEEE Conference on Decision and Control, 2004. CDC*, vol. 3, Dec. 2004, pp. 2609–2614 Vol.3.

[12] J. Eklund, J. Sprinkle, and S. Sastry, "Implementing and testing a nonlinear model predictive tracking controller for aerial pursuit/evasion games on a fixed wing aircraft," in *American Control Conference, 2005. Proceedings of the 2005*, Jun. 2005, pp. 1509–1514 vol. 3.

[13] Y. H. Chang, C. Tomlin, and K. Hedrick, "Biologically-inspired coordination of multiple UAVs using sliding mode control," in *American Control Conference (ACC), 2011.* IEEE, June 2011, pp. 4123–4128.

[14] L. Yang and J. Yang, "Stabilization for a class of nonholonomic perturbed systems via robust adaptive sliding mode control," in *American Control Conference (ACC), 2010.* Marriott Waterfront, Baltimore, MD, USA: IEEE, June 2010, pp. 1178–1183.

[15] H. Yuan, "Control of nonholonomic systems," Ph.D. dissertation, University of Central Florida, United States – Florida, 2009.

[16] Z. Qiang, L. Zengbo, and C. Yao, "A back-stepping 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

[17] D. Chwa, "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.

[18] F. Matsuno and J. Tsurusaki, "Chained form transformation algorithm for a class of 3-states and 2-inputs nonholonomic systems and attitude control of a space robot," in *Proceedings of the 38th IEEE Conference on Decision and Control, 1999.* IEEE, 1999, pp. 2126–2131 vol.3.

[19] A. Astolfi, "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.

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

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

[22] J. Tau, N. Xi, and W. Kang, "Tracking control of nonholonomic mobile robots," in *1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation, 1999. CIRA '99. Proceedings.* United States: IEEE, 1999, pp. 77–82.

[23] E. Hu, S. X. Yang, and N. Xi, "A novel non-time based tracking controller for nonholonomic mobile robots," in *Proceedings 2001 IEEE International*

Symposium on Computational Intelligence in Robotics and Automation, 2001. Canada: IEEE, July-August 2001, pp. 119–124.

[24] Y.-P. Tian and K.-C. Cao, "An lmi design of tracking controllers for nonholonomic chained-form system," in *American Control Conference, 2007. ACC '07.* IEEE, July 2007, pp. 4512–4517.

[25] K.-C. Cao, "Global k-exponential trackers for nonholonomic chained-form systems based on LMI," *Intern. J. Syst. Sci.*, vol. 42, no. 12, pp. 1981–1992, 2011.

[26] Y. Kanayama and F. Fahroo, "A new line tracking method for nonholonomic vehicles," in *1997 IEEE International Conference on Robotics and Automation, 1997. Proceedings*, vol. 4, Apr. 1997, pp. 2908–2913 vol.4.

[27] W. MacKunis, 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

[28] M. Ou, 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

[29] J. Keighobadi 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

[30] J. P. Hespanha, 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.

[31] G. Campion, B. d'Andrea Novel, and G. Bastin, "Modelling and state feedback control of nonholonomic mechanical systems," in *Proceedings of the 30th IEEE Conference on Decision and Control, 1991.* IEEE, December 1991, pp. 1184–1189 vol.2.

[32] K. Tsuchiya, 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.

[33] C. Camicia, F. Conticelli, and A. Bicchi, "Nonholonomic kinematics and dynamics of the sphericle," in *2000 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings*, vol. 1. IEEE, 2000, pp. 805–810 vol.1.

[34] C. Deneubourg, F. Sneyd, and T. Bonabeau, *Self-organisation in Biological Systems.* Princeton, 2001.

[35] I. D. Couzin, "Collective cognition in animal groups," *Trends in Cognitive Sciences*, vol. 13, no. 1, pp. 36–43, 2009.

[36] M. A. Haque, 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.

[37] H.-S. Park, 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.

[38] P. Leitão, J. Barbosa, and D. Trentesaux, "Bio-inspired multi-agent systems for reconfigurable manufacturing systems," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 5, pp. 934–944, Aug. 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0952197611001886

[39] J. H. Reif and H. Wang, "Social potential fields: A distributed behavioral control for autonomous robots," *Robotics and Autonomous Systems*, vol. 27, no. 3, pp. 171–194, May 1999. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889099000044

[40] G. T. Skalski and J. F. Gilliam, "Feeding under predation hazard: Testing models of adaptive behavior with stream fish," *The American Naturalist*, vol. 160, no. 2, pp. 158–172, 2002.

[41] Z.-P. J. Qin Li, "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.

[42] 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.

[43] G. T. Skalski and J. F. Gilliam, "Flocking for multi-agent dynamic systems: algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.

[44] L. A. Dugatkin and J.-G. J. Godin, "Predator inspection, shoaling and foraging under predation hazard in the trinidadian guppy,poecilia reticulata," *Environmental Biology of Fishes*, vol. 34, no. 3, pp. 265–276, 1992.

[45] S.-Y. Tu and A. Sayed, "Mobile adaptive networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 649–664, 2011.

[46] F. Cattivelli and A. Sayed, "Modeling bird flight formations using diffusion adaptation," *IEEE Transactions on Signal Processing*, vol. 59, no. 5, pp. 2038–2051, 2011.

[47] S.-Y. Tu and A. Sayed, "Tracking behavior of mobile adaptive networks," in *2010 Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*. California, USA: IEEE, November 2010, pp. 698–702.

[48] J. Li, S.-Y. Tu, and A. Sayed, "Honeybee swarming behavior using diffusion adaptation," in *2011 IEEE Digital Signal Processing Workshop and IEEE*

Signal Processing Education Workshop (DSP/SPE). Arizona, USA: IEEE, January 2011, pp. 249–254.

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

[50] A. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. Towfic, "Diffusion strategies for adaptation and learning over networks: an examination of distributed strategies and network behavior," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 155–171, 2013.

[51] C. Lopes and A. Sayed, "Diffusion least-mean squares over adaptive networks," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3122–3136, 2008.

[52] C. G. Lopes and A. H. Sayed, "Diffusion adaptive networks with changing topologies," in *IEEE International Conference on Acoustics, Speech and Signal Processing, 2008. ICASSP*. United States: IEEE, March 2008, pp. 3285–3288.

[53] N. Takahashi, 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.

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

[55] J. Chen, C. Richard, and A. Sayed, "Diffusion LMS for clustered multitask networks," *arXiv:1310.8615 [cs, math]*, 2013.

[56] J. Chen, C. Richard, and A. H. Sayed, "Multitask diffusion adaptation over networks," *IEEE Transactions on Signal Processing*, vol. 62, no. 16, pp. 4129–4144, 2014.

[57] M. Gholami, E. Strom, and A. Sayed, "Diffusion estimation over cooperative networks with missing data," in *2013 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, December 2013, pp. 411–414.

[58] F. Cattivelli and A. Sayed, "Self-organization in bird flight formations using diffusion adaptation," in *2009 3rd IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*. The Netherlands: IEEE, December 2009, pp. 49–52.

[59] S.-Y. Tu and A. Sayed, "Mobile adaptive networks with self-organization abilities," *IEEE Journal of Selected Topics in Signal Processing*, pp. 379–383, 2010.

[60] S.-Y. Tu and A. H. Sayed, "Foraging behavior of fish schools via diffusion adaptation," in *2010 2nd International Workshop on Cognitive Information Processing (CIP)*. Italy: IEEE, June 2010, pp. 63–68.

[61] ——, "Cooperative prey herding based on diffusion adaptation," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Czech Republic: IEEE, May 2011, pp. 3752–3755.

[62] G. Zhengxiong and G. Xinsheng, "A particle swarm optimization for the motion planning of wheeled mobile robot," in *2010 8th World Congress on Intelligent Control and Automation (WCICA)*, Jul. 2010, pp. 2410–2414.

[63] N. Kherici and Y. M. B. Ali, "Bio-inspired algorithm for wheeled robot's navigation," *Int. J. Artif. Intell. Soft Comput.*, vol. 2, no. 4, pp. 353–366, Sep. 2011. [Online]. Available: http://dx.doi.org/10.1504/IJAISC.2011.042716

[64] B. Matebese, D. Withey, and M. Banda, "Application of the leapfrog method to robot path planning," in *2014 IEEE International Conference on Information and Automation (ICIA)*, Jul. 2014, pp. 710–715.

[65] A. Richards, J. Bellingham, M. Tillerson, and J. P. How, "Coordination and control of multiple UAVs," in *AIAA Guidance, Navigation, and Control Conference (GNC)*, Monterey, CA, August 2002, aIAA Paper 2002-4588. [Online]. Available: http://acl.mit.edu/papers/2002_4588.pdf

[66] D. Shim, H. Kim, and S. Sastry, "Decentralized nonlinear model predictive control of multiple flying robots," in *42nd IEEE Conference on Decision and Control, 2003. Proceedings*, vol. 4, Dec. 2003, pp. 3621–3626 vol.4.

[67] J. Desai, J. Ostrowski, and V. Kumar, "Modeling and control of formations of nonholonomic mobile robots," *Robotics and Automation, IEEE Transactions on*, vol. 17, no. 6, pp. 905–908, 2001. [Online]. Available: http://dx.doi.org/10.1109/70.976023

[68] D. Shim and S. Sastry, "An evasive maneuvering algorithm for UAVs in see-and-avoid situations," in *American Control Conference, 2007. ACC '07*, Jul. 2007, pp. 3886–3891.

[69] J. Eklund, J. Sprinkle, and S. Sastry, "Switched and symmetric pursuit/evasion games using online model predictive control with application to autonomous aircraft," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 3, pp. 604–620, May 2012.

[70] S. LaValle, *Planning Algorithm*, 1st ed. Cambridge University Press, 2006.

[71] S. Ibrir and S. Brahim-Belhaouari, "On the control of nonholonomic systems in power form," in *Control '98. UKACC International Conference on (Conf. Publ. No. 455)*, Sep. 1998, pp. 1438–1443 vol.2.

[72] H. Khalil, *Nonlinear Systems*, 3rd ed. Eaglewood cliffs, NJ: Prentice Hall, April 2002.

[73] O. Albayrak, "Line and circle formation of distributed autonomous mobile robots with limited sensor range," Ph.D. dissertation, 1996.

[74] Y. Kanayama, A. Nilipour, and C. Lelm, "A locomotion control method for autonomous vehicles," in *, 1988 IEEE International Conference on Robotics and Automation, 1988. Proceedings*, Apr. 1988.

Vitae

- Name: Thompson Oladipo Raphael
- Nationality: Nigerian
- Date of Birth: October 29, 1986
- Email: *oladipothompson@yahoo.co.uk*
- Permanent Address: 57, Road 809, Area 8, OPIC Estate, Agbara, Ogun State. Nigeria
- BS: First class honors (4.73/5.0) in Systems Engineering, University of Lagos.
- MS: Distinction (3.821/4.0) in Systems & Control Engineering, King Fahd University of Petroleum & Minerals.
- Publications: S. El-Ferik and O. Thompson, "Biologically Inspired Control of a Fleet of UAVs with Threat Evasion Strategy", *Asian Journal of Control.* (Submitted)
- Publications: S. El-Ferik and O. Thompson, "Evasive control of multiple UAVs in a mobile adaptive network". (Under review)