Unmanned Aerial Vehicle Swarm Synchronization Protocol

By

Maikano Kenneth Oganne

14100120

BSc (Software Engineering with Multimedia) (Limkokwing University of Creative Technology Worldwide)

Department of Computer Science and Information Systems

College of Science

Botswana International University of Science and Technology (BIUST)

maikano.oganne@studentmail.biust.ac.bw, +267 73623774

A dissertation submitted to College of Science in Partial Fulfillment of the Requirements for the Award of the Degree of Master of Science in Computer Science of BIUST

Supervisor: Dimane Mpoeleng, Ph.D.

College of Science, BIUST

mpoelengd@biust.ac.bw, +267 72972059

Signature____________________ Date 6/ April /2018

Declaration and Copyright

I, Maikano Kenneth Oganne, declare that this dissertation/thesis is my own original work and that it has not been presented and will not be presented to any other university for a similar or any other degree award.

Signature ....

Acknowledgment

I will first like to thank GOD for the strength and grace he gave me throughout my research period in BIUST.

Secondly, I will like to express my gratitude to my mentor Dr, Dimane Mpoeleng for the valuable remarks, comments and engagement through the learning procedure of this master thesis. Furthermore, I would like to express my gratitude toward Prof, Heiko Schroeder for introducing me to the research topic, as well for the support he offered. Likewise, I thank my friend Mr Zibo Jane for the support he gave me whenever needed.

At last, I should express my significant appreciation to my family for giving me unfailing backing and ceaseless consolation during my time of study. This achievement would not have been conceivable without them.

Dedications

I am dedicating this work to the Almighty GOD, to my family especially my mother, to my friends and my Pastors.

I also dedicate this research work to the department of Computer Science and Information Systems UAV research lab in Botswana University of Science and Technology.

Contents

Table of Figures

List of Abbreviations

API	Application programming interface
BAN	Body Area Network
BLE	Blue Low Energy
DCA	Dynamic Channel Allocation Technique.
UAVSSP	Drone Swarm Synchronization Protocol
FPV	First Person View
FSM	Finite State Machine
GHz	Gigahertz
GPRS	General Packet Radio Service
GPS	Global Position System
GSM	Global System for Mobile communication
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
IR	Infrared
MANET	Mobile Ad-hoc Network
MS	Mobile Station
PAN	Personal Area Network
QoS	Quality of Service
RC	Remote Control
RF	Radio Frequency
ROS	Robot Operating System

SPP	Self-Propelled Particles
TCP	Transfer Control Protocol
UAV	Unmannered Aerial Vehicle
UDP	User Datagram Protocol
VHS	Virtual Heading Sensor
VMC	Vicon Motion Capture
V-Rep	Virtual Robot Experimentation Platform
WLAN	Wide Area Network

Abstract

Unmanned Aerial Vehicle (UAV) swarms are not common due to lack of readily available coordinated protocols and applications for synchronizing UAV swarms in outdoor environments. To solve the problem, this study design, implement, and evaluate a leader-follower Unmanned Aerial Vehicle Swarm Synchronization Protocol (UAVSSP). Unlike other studies that deal with UAV swarm synchronization, this study introduces the concept of pair correlation and back propagation communication scheme amongst UAVs in a swarm. UAVSSP uses Standard Vicsek Model (SVM) Self-Propelled Particle (SPP) functions to predict the velocity and position vectors in next time step (*t+1*) of a UAV within a swarm. Furthermore, the protocol uses Directional pair correlation function to monitor the distance between a UAV and its leader.

For implementation and evaluation of the UAVSSP, Virtual Robot Experimentation Platform (V-Rep) was used to test 1) the cost of wireless transmission range on the synchronization time of UAVs in different shapes using UAVSSP, 2) the cost of networking topology on the synchronization time of UAVs using UAVSSP and 3) communication error rate using UAVSSP. In the cost of wireless transmission range matrix, we were investigating the agility of UAV swarm synchronization in different communication transmission range. The cost of networking topology matrix was dealing with investigating the impacts of UAV networking arrangement in a swarm.

From the experiments conducted, we learned that communication error rate above 3.5% affect negatively the performance of UAVSSP. Adding on the above mentioned observation, a developed protocol recorded high performance when using single-hop MANET routing in V formation. Alteration of casting (unicasting, multicasting and broadcasting) method does not affect UAVSSP performance.

1.0) General Introduction

1.1) Introduction

Unmanned Aerial Vehicle (UAV) commonly known as a flying robot, or a drone, is a small aircraft that flies with no pilot on board. Traditionally, a UAV during mission execution is controlled remotely by a pilot on ground station but with advancement of technology, UAVs can now fly automatically on pre-programmed flight trajectories or can be controlled by more complex dynamic automation systems executing in an onboard computer [1]. A UAV can be used to perform tasks such as hurricane hunting that are otherwise risky to be performed by an actual aircraft or tasks that are tedious to be undertaken by humans [2].In attending to real time problem like search and rescue, it is preferable to use swarm of UAVs than just one UAV [3].

According to Axel *et.al* [4], the real advantage of a swarm of UAVs is that it accelerates the speed at which you can respond to things in the world than just one UAV or an actual traditional aircraft. Furthermore, the advantage and qualities of UAV swarm can be presented by contrasting a solitary UAV and other systems with many members. These qualities are similar to that of nature swarm such as ant colonies, fish shoal and schools [5] [6]. The individual entities in nature swarm show exceptionally poor capabilities in task handling yet swarm of these individuals can handle complicated and complex tasks effectively. It is by mimicking nature swarms that we get to realize that swarm of simple micro-UAVs with deprived resources can solve complicated tasks. In the same light, computer scientists are investigating ways in which the concept of swarming displayed in nature (school of fish, swarm of birds) can be used to build robust and low cost micro-UAV systems for addressing the real world applications. Swarming allows simple autonomous micro-UAVs to work together in goal attainment by interacting together just like social insect’s colonies, swarm of birds and school of fish.

In this research, UAV swarm is defined as a group of UAVs that work together to accomplish specific goals, communicating with each other and assisting other members of the swarm in tasks. Other literature stipulates that UAV swarm should be decentralized and distributed [6]. Decentralization implies that UAVs under the influence of simple rules interacting with each other can complete the task without centralized control and distributed implies that each UAV within a swarm system has the freedom to govern itself but in cooperation with other members [7]. However, in this research context an emphasis is given to intercommunication between UAVs within a swarm and how they synchronize. The exchange of information (such as position) is unavoidable when UAVs has to cooperate with each other. According to Ying *et.al* [6], information exchange with UAV swarm is in three ways: direct communication, communication through environment, and sensing. The procedure can be either same or distinctive for the swarm because of various applications (e.g. a more than one interaction can be used in one swarm). Direct communication uses wireless network to establish UAV to UAV, UAV to UAVs and UAV to Control Station connection. Communication through environment is a concept whereby a robot uses artificial or virtual pheromones to communicate with other robots [6] [8]. Communication through sensing is dependent on sensors onboard to perceive the environment so as to fulfill task such as obstacle avoidance, target search and flocking.

In UAV swarm synchronization, communication allows UAVs to pass valuable information about the environment and the state of other UAVs within the swarm (position vector, velocity vector). Using this information, each individual UAV can perform self-organization using control protocol residing in on-board computer. The control protocols in simple terms are steps that explain how UAVs should execute a given tasks [9]. These steps influence individual behavior of each UAV within a swarm such that through UAVs interaction we might have a collective behavior known as swarm synchronization. UAV swarm synchronization in outdoor environment brings a burden of addressing unstable variables such as communication latency (as a result of temperature, wind) state estimation and location precision when controlling and synching UAV swarm.

Therefore, the aim of this research is to develop and analyze a protocol to keep several programmable small and cheap UAVs compatible with DroneCode in swarm synchrony mode as they are flying outdoor using Global Positioning System (GPS) technology whereby one

UAV acts as a leader. The leading UAV coordinates the communication among UAVs as they exchange their speed, direction, position in terms of three dimensions: Longitude, Latitude, and Altitude to the follower or followers.

1.2) Background

As aforementioned, this research emphasis is on UAV swarm synchronization protocol development. UAV swarm inherits scalability as a requirement from natural swarm systems. Scalability demands swarm synchronization protocol to be tested in all UAVs that partake in swarm activities. However, this requirement might become impractical in cases where a swarm consists of thousands of UAVs because the expense of an individual UAV restricts testing of the protocol in thousands of UAV on the present condition of the robot innovation. Since scalability is a vital point of UAV swarm systems, models (statistical physics, swarm robotics, swarm intelligence) will be required to simulate swarm behavior until less expensive UAVs are manufactured. Models are needed to better comprehend and control the internals of system under investigation. Bayindir *et.al* [10] specified swarm system models in four types which are: sensor-based, microscopic, macroscopic and cellular automata modeling [6].

1.2.1) Swarm Simulation Models

1.2.1.1) Sensor-Based Modeling

Sensor-based modeling is a technique which utilizes the models of sensors and actuators of UAVs or robots and items in the environment as the fundamental segments of the swarm systems. In the wake of modeling these segments, the associations of the robots with the environment and the collaboration between the robots are displayed. This modeling technique is the most utilized and the most used strategy for modeling robotic experiments [10] [6]. The earlier research utilizing sensor-based modeling strategies did not consider the physical restrictions but nowadays researchers bring physical guideline into the model. According to Michael *et.al* [11], they are four main restrictions which are: 1) Flight envelope, 2) radar and communication, 3) onboard processing power, and 4) fuel supply.

The UAV payloads, cruising speed and turning radius are some of the factors considered under flight envelope restriction. Swarms systems using Sensor Based Modeling should account for these factors. As an example, in a scenario whereby the objective of UAV swarm is to fly in predefined formation, the turning radius of each UAV in extreme cases (e.g. minimum and maximum speed, minimum and maximum payloads)should be modeled.

Typically, the size of a UAV determines the onboard processing power, communication and fuel supply abilities. Micro UAVs are highly affected by these restrictions unlike large UAVs and mannered aircraft. Therefore, these restrictions impact the emergent behavior of UAV swarms. For example, the micro UAVs emergent swarm behavior is compromised because wireless communication modules installed in them cannot propagate signals for longer distances. As a result, only a swarm of UAVs flying in close proximity has high emergent behavior unlike swarm of UAVs flying far away from each other.

1.2.1.2) Microscopic Modeling

In the Microscopic modeling, the robots and co-operations are defined as a Finite State Machine (FSM) [6]. The practices of each robot are characterized as a few states, and the exchange conditions depend on the input from communication and sensing. Since the model depends on the practice of every robot, the simulation ought to keep executing for several times in order to attain average behavior of a swarm. According to Ying Tan *et.al* [6] most of swarm systems research under this type is based on probability, since noise (perturbation) can be demonstrated as likelihood in the model. Probabilities are esteemed from the experiments of real robots, and the model is iterated with these probabilities state transfer in the simulation to anticipate the conduct of the swarm.

1.2.1.3) Macroscopic Modeling

Inverse of Microscopic Modeling is Macroscopic Modeling. The gigantic demarcation factor between these two models is; in microscopic model, the swarm behavior is actually mimicked at individual level of entities in a swarm where-else in Macroscopic Modeling, the behaviors/practices are mimicked at swarm level [6]. Along these lines we can conclude that, the Macroscopic Model is more into the description of aggregate conduct of the swarm while Microscopic Model demonstrate the behavior of individuals within a swarm [11].

1.2.1.4) Modeling from Swarm Intelligence Algorithms

Some algorithms have been employed from swarm intelligence into swarm robotics to mimic the natural swarms by many researchers. Particle Swarm Optimization (PSO) which imitates the flocking behavior of birds is one of the algorithms from swarm intelligence that are used in swarm robotics [6]. PSO was introduced by Eberhart and Kennedy in 1995. In this algorithm, each agent within a swarm is treated as a point or particle in three dimensional search spaces. Each particle flies in search space at speed which is progressively balanced by its flying knowledge and its colleague's flying experience [12]. In PSO, particles are kept as individuals from the populace through the course of the run (a run is characterized as the aggregate number of eras of the developmental calculations before end). It is the speed of the particle which is updated according to its past best position and that of its colleague (other particle) [12].

Another swarm intelligence algorithm commonly used in modeling robot swarms besides PSO is Ant Colony Optimization (ACO) algorithm [6] [13]. The Ant Colony Optimization (ACO) models the aggregate scavenging conducts of ants. Adding on PSO and ACO algorithms, Honey Bee and Artificial Fish Swarm are as well applicable in modeling robot swarm behavior [14].

1.2.1.5) Modeling from Statistical Physics

Reynolds *et.al* [15] and Vicsek *et.al* [5] offered an outstanding Self Propelled Particles (SPP) models that has been utilized by some researchers to describe the flocking behaviors of swarm systems such as Bacteria colonies, Fish schools and UAV swarms amongst others. Reynolds models prove that emergent behavior of agents in a swarm (e.g. school of fish) is guided by or can be described by rules such as Cohesion, Separation and Alignment [19]. Cohesion rule allow agents commonly known as particles or boids to be attracted to each other where else separation rule allows agents to maintain space within each other so as to avoid collision and Alignment rule align velocities of each members within a swarm.

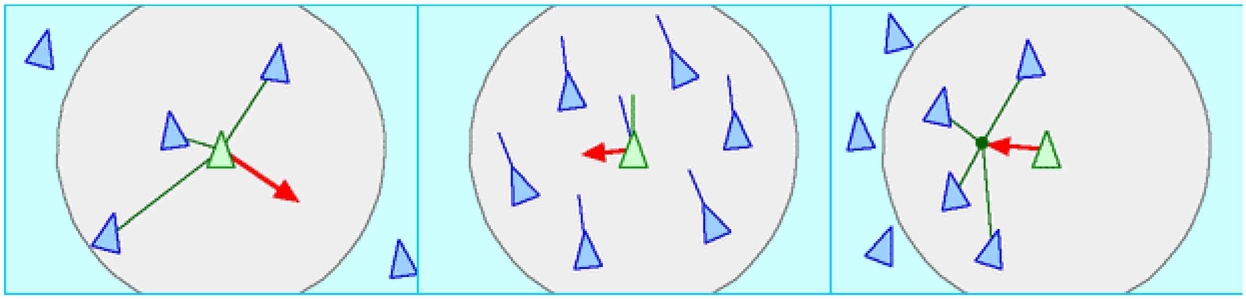


Figure 1(a). Separation: steer to avoid crowding local flock mates [16]

Figure 1(b). Alignment: steer towards the average heading of local flock mates [16]

Figure 1(c). Cohesion: steer to move toward the average position of local flock mates [16]

On the other hand, Vicsek *et.al* [17] introduced a simplified self-propelled particles model to define flocking behavior of swarm systems. This model was presented to study clustering, transport and face move of imbalance or non-equilibrium systems where the speed of particles is dictated by a basic rule and vacillations. The main standard of the model is, at every time a given particle driven with a consistent total speed adopts the normal bearing of movement of particles in its neighborhood in radius *r* with some irregular annoyance (perturbation) included.

1.2.2) Organizations that promote UAV research projects

Besides lack of control protocols for synchronizing swarm of outdoor flying UAVs, the wide usage and acceptance of UAVs for commercial purposes have not surfaced because of some laws and policies that prohibit such act. In USA, the regulation governing the use of UAVs for securing air safety was issued on September 2015 by Federal Aviation Administration (FAA) and the regulation has UAVs projects on ice [18] [19] [20]. The regulation is a setback for researchers who want to test UAV control protocols and algorithms in real life environment. In mitigation of the aforementioned challenge, some organization such as Dronecode and Arduino surfaced with the aim of providing affordable hardware and test-bench (simulators) for researches and students who are interested on UAV related projects. The following are some of test benches developed by Dronecode and Arduino.

1.2.2.1) Arduino

Arduino is an organization that focuses on the production of microcontrollers and open-source software for developing electronic devices that can sense and control other physical devices such as UAVs [21]. The main reason Arduino was born is to provide a tool that can be used by students, researchers and hobbyists without background on programming and electronics for prototyping and building scientific projects. The organization encourages competitions so as to cultivate creativity on robotics (UAV, rovers, and humanoid) and electronics projects. On projects relating to this research, Arduino technology was used on AR-DRONE and 1SHEELD- Discover the world around your drone [22], Weather Station for Drones [23] and Control a Parrot AR-Drone with Linino projects [24].

1.2.2.2) DroneCode

DroneCode project is a non-profit making organization under Linux [25] that motivate the improvement of open source client and business UAV software program by means of helping a community of developers and supplying them the resources and tools to help them innovate [26] [27]. A definitive objective is to boost appropriation of the project's code for the advantage of clients with less, better, and more solid UAV programming. Some resources offered by DroneCode includes simulators such as jMavSim, SITL(Software in loop), ROS(Robot Operating System) Gazebo **,** Last_Letter, JSBSim, and X-Plane [28]**.**

1.2.2.3) JMavSim

jMAVSim is a user friendly multi-rotors simulator. The simulator connects with autopilot using MAVLink communication protocol without a ground station. Moreover, jMAVSim can be linked with Hardware in Loop using serial or software in loop using User Datagram Protocol (UDP) [30]. Most researchers have used this simulator in their research projects. As an example, the simulator was used in [31]. The objective of research project in [31] was to develop an algorithm that will enable UAV to track a moving object using image processing technique. In the research, jMAVSim was used in conjunction with Hardware in loop to simulate UAV flight as well as to provide visual scenes of the flight where-else hardware in loop was used to verify and to evaluate the developed protocol without real flight data.

1.2.2.4) SITL(Software in loop)

The SITL simulator offers programmers with a platform to test protocols and algorithms on fixed wing UAV, copters and rovers without any hardware. It consists of an autopilot built on C++. In [32] SITL was used in testing of fussy algorithm developed to control landing behavior of quad-copters in outdoor environments. Another research project where SITL was used can be traced in [33]. The authors of the project in [33] were interested in studying the performance of fixed-wing and multi-rotor unmanned aerial platforms as they are traversing over two different flight paths. In this case STIL was used to evaluate these flight paths and the performance of each vehicle traversing them. The STIL used for the research project in [33] incorporates weather conditions, fixed-wing and multi-rotor frames, the autonomous flight controller, a physics engine to imitate the dynamics of fixed-wing and multi-rotor UAVs, and a ground control station.

1.2.2.5) ROS (Robot Operating System) Gazebo Simulator

ROS and gazebo are two independent packages that run on Linux environment. Gazebo offers the physics of UAVs and the environment (indoor and outdoor) simulation where else ROS represent the operating system of the actual UAV or robot. To integrate ROS and Gazebo, ROS packages named gazebo_ros_pkgs has to be installed. Gazebo_ros_pkgs has been created to grant wrappers around the stand-alone Gazebo. Again, gazebo_ros_pkgs grant the fundamental interfaces that allow Gazebo to simulate UAV using commands from ROS.

1.2.2.6) Last_Letter.

Another simulator besides jMavSim, SITL and ROS Gazebo is last_Letter. Last_letter is a collection of ROS programs suitable for researches and algorithm development. According to [28], last_letter simulator currently is on its early development.

1.2.3) Research Projects

Despite the setback brought in by regulations, UAV swarm projects are ongoing in laboratories. Some of those research projects include a research named 'Towards A Swarm of Agile Micro Quadrotors' in University of Pennsylvania [29], Outdoor flocking and formation flight with autonomous aerial robots in [30], IRobot swarm project in Massauchettes Institute of Technology [31] and the swarm BOTS project in University of Brussels [31].

The intent of 'Towards A Swarm of Agile Micro Quadrotors' research project was to prove that the size of UAVs affects swarm performance. Moreover, the project authors were interested on synchronizing UAV swarm built up of many agents. The experimental result of the project shows that small UAVs can maintain tight formation, can fly in close proximity, they are swift and stable in presences of high perturbations hence improving swarm performance. The swarm synchronization approach they proposed can accommodate more

UAVs than the 20 number used in experiments. However, the project is far from solving the challenges associated with the envisioned applications of UAV swarms because it was developed for indoor test benches only. Figure 2 below shows Towards A Swarm of Agile Micro Quadrotors' indoor experiment whereby a swarm of 20 micro UAVs was used as a test bench.

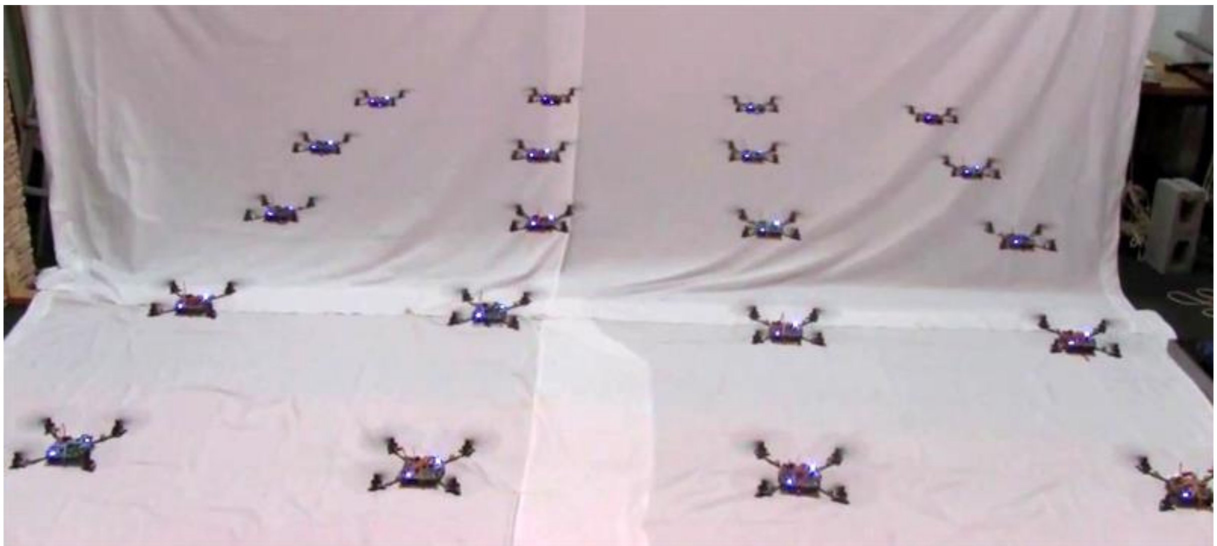


Figure 2: Towards A Swarm of Agile Micro Quadrotors research project indoor experiment. The experiment demonstration swarm of 20 flying micro copters

IRobot Swarm is built up of 100 ground robots synchronized using two algorithms which are; 1) dispersion algorithm and 2) distributed localization and mapping algorithm. The dispersion algorithm is a combination of uniform dispersion and boundary exploration algorithms. Boundary exploration algorithm attracts individual robots to the robots in the boundary of a swarm where-else the Uniform dispersion algorithm helps the individual robots to locate neighbors in close proximity. When robots are in exploration state they use the dispersion localization and mapping algorithm. The algorithm is as well used for mapping the environment as the name connotes. For robot intercommunication, the multi-hop protocol was used.

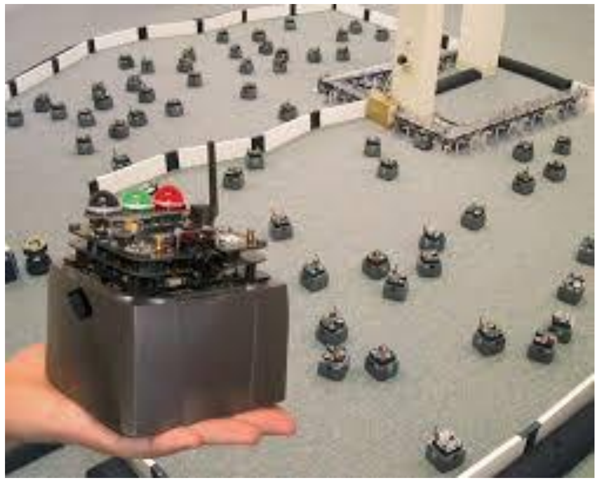


Figure 3: iRobot Swarm dispersed in an open space.

Swarm BOTS is another research project that was carried to synchronize ground robot swarm. The swarm operate under the influence of hole avoidance algorithm and chain formation algorithm. According to [31], the hole avoidance algorithm helps the robots maneuver in unknown and challenging terrain and chain formation algorithm to form chain between destination and source. Robots within a swarm use colors (blue, green and yellow) as mode of communication. During the experimental stage of a project, it was realized that the spontaneous behavior expected was found in a swarm containing five to twenty robots.

a) chain of swarm BOTS robots navigation on un-even terrain

b) Swarm BOTS robots using different colors as their communication mode

Figure 4: Swarm BOTS mini robots

Algorithms used in Swarm BOTS and IRobot swarm projects are tailored for two dimensional work spaces. Therefore cannot address challenges such as wind and communication errors associated with synchronization of robot swarms in three dimensional work spaces especially in outdoor environments.

A research project similar to ours in terms of system application context, communication and positioning technology on the above mentioned projects is ‘Outdoor flocking and formation flight with autonomous aerial robots’. Outdoor flocking and formation flight with autonomous aerial robots project uses physics Self Propelled Particle (SPP) models to define the behavior of each UAVs within a swarm. This project is based on fully decentralized two dimensions algorithm but theoretically the algorithm can be extended into three dimensions. The algorithm allows UAVs to join or leave the swarm at any time without affecting the behavior of a swarm just like natural Aerial flocks of birds. Decentralization implies, the UAV swarm does not have a control station or a central control point. Each and every UAV within a swarm computes its velocity vector and positional vector for the next time step using a mini computer installed in a UAV. UAVs only shares the positional information received from GPS with other UAVs using wireless communication technology. Figure 4 shows a picture of 10 UAVs flying outdoor during the experiment of project in [30]. During the experiments, the minimum difference recorded between UAVs in velocity range of 0-4m/s with GPS precision of ± 2m, 1.5s time delay, and random wind disturbances was 6-10m.

Figure 4. Snapshots of quadcopter flocks of 10 units. Nightlight image: robots are following a target in a grid formation (r0 = 6 m, v0 = 3 m/s). Daylight image: SPP model (r0 = 10 m, vflock = 2 m/s). [32]

1.3) Problem Statement

A considerable number of UAV applications that are envisioned to yield better results are outdoor based. The applications involve hurricane hunting, search and rescue, border patrol and parcel delivery. These kinds of applications with the state of art in micro UAV technology call for employment of micro UAV swarms. The advantages of swarm employment are robustness and scalability. Furthermore, UAV swarm accelerates the speed at which you can respond to things in the world than just one UAV or actual aircraft [4]. According to [33], the gigantic barricade on the actualization of UAV swarms for civil purposes is lack of readily available Coordinated protocols and Applications for synchronizing UAV swarms.

The problem this study addresses is the lack of coordinated protocols that can be implemented to synchronize swarm of flying micro UAVs in outdoor environments.

1.4) Research Objectives

1.4.1) General Objective

Develop ad-hoc swarm synchronizing protocol to keep several outdoor micro UAVs in synchronization mode using GPS technology.

It is not always the case to have reliable pre-existing net-work infrastructure in outdoor environments. Therefore, ad-hoc network infrastructure offers an advantage of ensuring network availability or connectivity in such unfavorable context. UAVs using ad-hoc communication architecture form the network infrastructure by themselves since each UAV within a network act as a mobile router.

Synchronization highlighted in the general objective connotes coordinates synchronization. Since the developed protocol insures that UAVs maintains a pattern (is not a cluster) or formation throughout their flying experience. The easiest way is to ensure that each UAV maintain its position (in terms of coordinates) relative to its leader. UAV are said to be synchronized only when they managed to keep the same distance from each other with tolerable error.

1.4.2) Specific Objectives

- To identify suitable models that can be used in UAV swarm synchronization.

A model is a way of representing real life situations to forecast their future behavior. In other terms, we can define a model as a tool used to represent or depict reality. In most case, models are represented using mathematical equations. Only five types of models which are; Sensor-based, Microscopic, Macroscopic, Swarm Intelligence and Self Propelled Particles (from Statistical Physics) were considered in this study. Not all reviewed models are good to be applied in a given situation. Therefore, on this objective, we wanted to identify suitable models that we can use in synchronizing swarm of homogeneous micro UAVs. We identified a suitable model by considering computational requirement and number of sensors needed for that model to be functional.

Micro UAVs have limited resources (processing power and battery), therefore cannot carry large payloads (extra sensors). This limitation binds us to identify models that do not require many sensors to be installed in UAV.

- To evaluate algorithms or protocols best suited for synchronizing swarm of outdoor flying UAVs.

- To find which communication technologies, architectures and standards are suitable to support communication between UAVs within a swarm in outdoor environments. They are many wireless technologies, architecture and standards that can be used in UAV to UAV or UAV to base station communication, however, choosing the optimal wireless communication technology, architecture or standard is dependent on the context of use. Some of the technologies, architectures and standards that are commonly used to support UAV swarm wireless communication in indoor environments depending on their attributes (omnidirectional or bidirectional, signal propagation strength) are not best suited for outdoor environments. Therefore, this objective is geared to find the most suited for outdoor environments.

- ❖ To design protocol to synchronize outdoor flying micro UAVs in difference shapes.

  Protocol is a set of rules that governs how a certain task should be carried out. Technically, a protocol is not an algorithm even though sometimes these two words are used interchangeably. A protocol is not sensitive of order where-else an algorithm is. In this study, we use models to describe UAV swarm synchronization protocol. In other words, models help us on coming up with a detailed protocol.

- ❖ Analyze the synchronization performance of the developed swarm synchronizing protocol.

1.5) Research Questions

- ❖ Which models are used in swarm synchronization and how can they be applied in synchronization of micro UAVs?

  They are several mathematical models developed by scientists in order to simulate the movement patterns of living organism swarms such as fish shoals and schools. These living organism swarms are commonly known as natural swarms. Adding on the models developed to study movement patterns of living organism swarms, they are other models such as stochastic models that are as well used to model the behavior of artificial swarms. All these models have advantage and disadvantage over each other, therefore, this research question investigate in theory the most suitable models that can be adopted to be used in synchronizing swarm of micro UAVs.

- ❖ Which control algorithms or protocols are suitable to synchronize swarms of outdoor flying UAV using GPS as a state estimation system?

  Control algorithms (Pseudo- control algorithms) are rules or statements that stipulate how each Unmanned Aerial Vehicle (UAV) should maneuver within a swarm. They are several control algorithms used in artificial swarm synchronization (Robots), however, not all these are good to be used in synchronization of micro UAV swarm. Therefore, this research question investigate strength and weakness of control algorithms used in synchronizing swarm of Aerial artificial swarms (UAV) and ground artificial swarms(Rovers).

- ❖ Which communication techniques are suitable to support communication between UAVs within a swarm in outdoor environments?

  Micro UAVs has deprived resource in terms of processing power and battery life, therefore, not all communication technologies are suitable to enable communication amongst UAVs in a swarm especially in outdoor environments. In this regard, the research question investigates the suitable communication technique that can be employed in a swarm of micro UAVs. Number of communication modules to be installed in UAV, transmission range and communication architecture are variables that are considered in this research question.

- ❖ How to test and analyze the performance of the designed protocol?

1.6) Benefits of the study

The element that differentiates this research and makes it significant from other UAV researches is its context of application on micro UAVs. Micro UAVs unlike large UAVs has limited resources (battery, processing power, payloads) therefore, the intent of the research is to develop a simple protocol that synchronize micro UAVs to follow a leader in a defined formation and in presence of uncertainties such as wind pressure and communication latency using GPS for state estimation. The protocol developed can benefit micro UAVs users in different fields like photography and military.

Moreover, UAV intercommunication Packet Datagram is also designed in addition to the developed outdoor UAV swarm synchronization protocol. The datagram can as well be used by any outdoor UAV swarm system that utilize GPS for state estimation.

1.7) Dissertation Structure

This thesis is divided into eight chapters. Chapter one of the thesis is general introduction, chapter two is UAV swarm networking and technologies, chapter four is Drone Swarm Synchronization Protocol (UAVSSP) Design, chapter five is experimental and results presentation, chapter six is experimental results discussion and chapter seven is conclusion and chapter eight is future work.

Chapter two introduce the reader to technologies used in UAV communication where-else chapter three explain in details methods used to develop and to test Drone Swarm Synchronization Protocol(UAVSSP).

2.0) UAV Swarm Networking and Technologies

Networking UAV provides wireless networking infrastructure to support the quality of service (QoS) needs (bandwidth, latency and reliability) of UAV communication [34]. Unlike conventional manned aircraft or large UAVs, micro UAVs have the highest communication requirement because micro-UAVs need to communicate in order to stay together or to avoid collision when flying together as a swarm [35]. Networking to enable communication amongst micro-UAVs and pilot on ground station comes with constraints because of high mobility, and deprived resources of micro-UAVs. Taking in consideration the aforementioned facts about micro-UAV, this chapter reviews literature in the existing wireless communication technologies, standards, communication mode, protocols and architectures used in UAV systems. The aim is to identifying effective communication protocol, wireless technology and swarm synchronization algorithms or protocols for a swarm of outdoor flying micro UAVs without taxing their resources such as battery and processing power. Furthermore, the chapter is meant to introduce a reader to the basics of UAV wireless communication.

2.1) Communication Architectures

Communication architectures for networking UAVs in a swarm are categorized in two major categories which are centralized and decentralized [34] and each architecture category members has pros and cons. These two categories of architectures specify how information flows between the ground crew and UAVs or between UAVs themselves.

2.1.1) Centralized UAV Communication Architectures

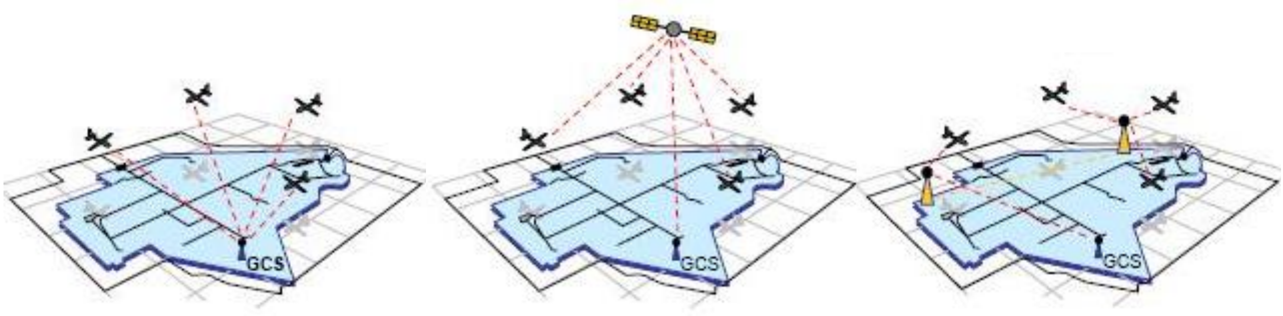
According to Brown *et.al* [36] and Jun *et.al* [37], in Centralized mechanism, a central node is required to connect either UAV to UAV or UAV to a remote pilot. In other words, UAV or UAVs within a swarm are not linked to each other but UAVs are connected directly to ground station, satellite or cellular base station (antenna) to enable communication between UAVs and remote pilots to UAVs. Direct link [36] also called centralized network architecture [37], satellite and cellular architectures are architectures classified as centralized architectures [36] [37].

Direct link network architecture is the most common network architecture because is simple to implement. In the architecture, UAVs are connected to a ground station (GS) on star topology with the assumptions that all connections are maintained over dedicated links and therefore data delivery is reliable with low latency between UAV and a pilot in ground station. However, the direct link architecture has its own downfalls. Firstly high latency is experienced when a UAV communicate with another UAV in a swarm because data has to be routed via ground station first. In this case, UAV to UAV connection will be limited by the characteristics of the link technology and may prove difficult between two highly mobile platforms [35]. Furthermore, communication between UAV and ground station is compromised where line of sight (LoS) is impossible especially in cities where obstacles such as buildings are many. Secondly, the amount of bandwidth scales down with the number of UAVs so that is impossible to have many UAVs operating simultaneously in the same space [35]. Finally, expensive telecommunication hardware such as high-power transmitters, steerable antennas, and significant bandwidth for boasting communication when UAVs are flying far from the ground station [36] is required.

Satellite network architecture offers better coverage than direct link network architecture to control station. However, the connectivity is still provided by routing data through the ground station in the architecture [36] [35]. In simple terms, UAVs are able to connect to ground station via satellite. Satellite network architecture suffers some limitations such as poor data delivery, delayed communication and lack of satellite bandwidth [35] [36].

Finally, in the Centralized UAV Communication Architectures category we have cellular network architecture [35] [36]. The UAV network in cellular network architecture is disseminated over land areas called cells and each cell served by at least one fixed-location radio transceiver or antenna called base station [38] [39]. Base station act as an intermediary between a UAV and a ground station in which all communication is channeled. This architecture has several advantages that can provide good network connectivity and reliable data delivery [36]. Firstly, the architecture allows limited bandwidth to be reused so many times [36]. Secondly, network can be expanded by increasing number of base stations in an area [35] [38]. The disadvantages of using the infrastructure is associated with costs of base stations needed in network realization [35] [36] and hand-offs of frequency channel that occur when a UAV flies into a different cell [36]. During hand-offs, network might fail or frequency channels might not be available in a new cell. However, in [38], the problem was addressed by the introduction dynamic cells. Dynamic cells sometimes referred to as Dynamic Channel Allocation (DCA) technique in cellular network allows a frequency in use to be extended in a foreign cell until another frequency is availed for call handoff to take place [41]. Cellular network infrastructure was used in [39] and [40]. In [39], the architecture was used in networking two Yamaha RMAX helicopter using General Packet Radio Service(GPRS) technology build on GSM standard and the interface of the research experiment was implemented on Java for Ericsson P900 that functions as a control station. The purpose of Ericson P900 is to receive telemetry data from UAVs and present it to the end user, again the Mobile Station (Ericson P900) allows an end user to control the UAVs. Even though the objective of the project or research is to develop highly portable, lightweight ground control station system such as Mobile Stations (MS), the result of the research experiment shows that the usage of cellular network architecture is effective in urban areas where open frequency bands such as wireless Ethernet are unreliable. However, bandwidth using General Packet Radio Service (GPRS) is a problem when streaming real-time videos. In addition to [39], Sky drone is one of the project that utilizes cellular networks(3G or 4G) for the remote control(RC) of UAVs using First Person View (FPV).

According to [41] the sky drone hardware can be connected to MAVlink compatible Autopilot Board such as 3D Robotic APM.



a. Direct link UAV network architecture [42].

b. Satellite UAV network architecture [42].

c. Cellular UAV network architecture [42].

Figure 5: Centralized Communication architectures

2.1.2) Decentralized UAV Communication Architectures / Mobile Ad-hoc Networks

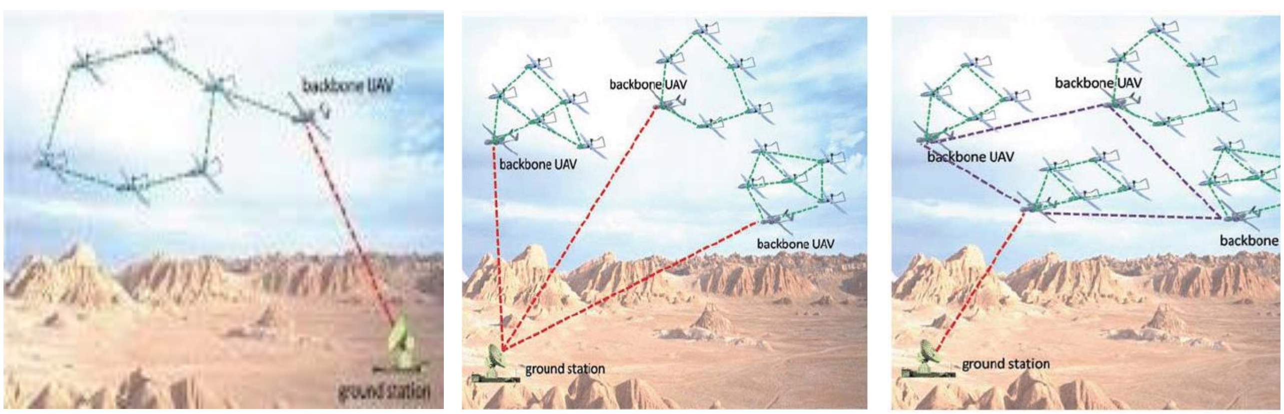
In decentralized communication architectures, central node is not a requirement [37]. Two UAVs can communicate without destining information to the ground station. They are three architectures under this category namely UAV ad-hoc network architecture commonly known as Mobile Ad-hoc Network (MANET) or Mesh in [35], multi-group UAV network and multilayer UAV Ad-hoc network [37].

In UAV ad-hoc network, pre-existing infrastructure is not needed for UAV to communicate. This architecture allows each UAV (node) to operate in distributed peer-to-peer mode [43]. Each and every UAV in a swarm participates in data forwarding for other UAVs within its transmission range [37] using one of transmission modes (Broadcasting, uni-casting and multi-casting) explained in chapter 2.5. In the situation whereby a UAV has to communicate with a UAV or UAVs beyond its communication range, the UAV have to use intermediate nodes to relay the messages and these act is called multi-hopping [43]. In this architecture, only one UAV equipped with two radios (one for communicating with UAVs and the other for communicating with the ground station) is required to act as a gateway of ad-hoc network. Since micro-UAVs cannot carry large payloads, Mobile Ad-hoc Network makes it feasible for low-cost, short range and light-weight radio frequency transceiver to be installed onboard as radio frequency propagator for successful UAV communication. However, it is worth mentioning that low-cost and light-weight transceivers cannot transmit signals for longer distances therefore to optimize connectivity in UAV ad-hoc network architecture, MANET has to be employed using proper wireless standard, and routing protocols. Furthermore, control algorithms and protocols (discussed in chapter 3.0) are needed to make UAV mobility patterns, such as speed and heading directions similar. Sources in [37] [43] [34] [35] [36], highly exalt the usage of Ad-hoc network architecture when networking homogeneous, high mobile node such as UAVs and where network topology is subject to change regularly.

Some missions require the deployment of different types of UAVs ranging from large to small UAVs with different payloads, radio transceivers and processing power. In this setup, UAVs of a similar type are usually physically close to each other and therefore, it may not be efficient for these UAVs to form a single Ad-hoc network due to traffic patterns involved in a network control overhead caused [42]. Other Decentralized UAV communication architectures such as Multi-Layer UAV Ad-hoc network and Multi-Group UAV Network maybe considered in mitigating the problem.

Multi-Layer UAV Ad-hoc Network architecture is made up of several UAV ad-hoc networks and is assumed that different UAVs are grouped according to their similarities either by size or type. All these ad-hoc networks are joined by communication channel called backbone. Similar to UAV ad-hoc network architecture, only one backbone UAV in Multi-Layer UAV Ad-hoc Network architecture is directly connected to the ground station. Another architecture that can be used in a network consisting different UAVs that cannot form a single Ad-hoc network is multi-group UAV Network architecture.

Multi-group UAV Network architecture is similar to Multi-Layer UAV ad-hoc Network. The architecture is made up of several UAV ad-hoc networks. Unlike Multi-Layer UAV ad-hoc Network, each ad-hoc networks have their own backbone UAV that link to the ground station. Even-though Distributed UAV network architectures are robust and scalable over Centralized UAV Network architectures according [34] [36] [37], the mobility patterns (speed and direction) of UAVs needs to be controlled and must be in communication range with one another for the Distributed UAV network architectures to be implemented.



a. UAV ad-hoc network architecture [37].

b. Multi-Group UAV network architecture [37].

c. Multi-Layer UAV ad-hoc network architecture [37].

Figure 6: Three types of MANET architectures

2.3) UAV Wireless Communication connection and coverage.

Communication amongst nodes in a network can be implemented either by using wireless connection or by using wired connection [44] [45]. A wired connection, also called the Ethernet, employs physical cables such as twisted pair cable, coaxial cable and fire cables in its implementation. In wireless connection, communication occurs in air interface by using radio frequency technology [34]. Considering the mobility of UAVs, wireless connection is the best option to consider than wired connection in UAV networking. However, pure wireless connection or communication surfers limitations in that, radio signals are subject to interference, low bandwidth, less security, high signal loss and fading [43]. In order to alleviate some of the aforementioned wireless communication problems, [43] and [37] suggest the usage of Mobile ad-hoc network architecture in networking UAVs. Their argument is based on the fact that, Mobile ad-hoc network architecture has the ability to provide multi-hop wireless network without preplaced infrastructure and it provides connectivity beyond line of sight (LoS).

Mobile ad-hoc network communication can be classified under Personal Area Network (PAN), Local Area Network (LAN), Metropolitan Area Network (MAN) and Wide (WAN) area networks depending on their geographical coverage [44]as shown in Figure 7 .

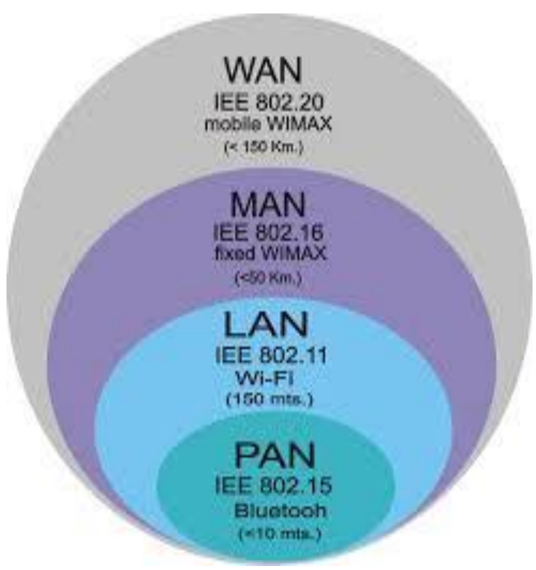


Figure 7. Mobile Ad-hoc network communication classification

Wireless Personal Area Networks (PAN) provides communication range of 1-10m with data rate less than 10 Mbps. PAN is a wireless network that operating on IEEE 802.15.4 and it allows the connection amongst the electronic devices close to each other (e.g., connecting a personal computer to a printer, UAV to UAV connection) [43] [44]. According to [43], most radios PAN are implemented in the 2.4 GHz ISM band to reduce interference and bandwidth reuse. Network type that provides coverage beyond the limits of PAN is Wireless Local Area Network (WLAN). WLAN provide network coverage from 1 – 500m and have data transmission rate up to 100Mbps. This network coverage uses IEEE 802.11 and HiperLAN/2 technology standards. [43] highlight that, WLAN design should address issues such as power consumption, mobility and bandwidth limitations associated with wireless environment. Moreover, the same literature specifies that infrastructure based on Ad-hoc networking should be considered during WLAN implementation or design.

2.4) Wireless Communication Standards and Technology

Wireless communication standards were introduced to govern the employment and manufacture of wireless communication technologies. Currently we have a number of different wireless technologies standards that can be implemented in hardware products for the Internet of Things (IoT) and Machine to Machine (M2M) communication. In IEEE, they are five commonly used standards discussed below.

2.4.1) IEEE 802.15.1: Bluetooth and Blue Low Energy (BLE)

IEEE 802.15.1 family of standards is used in Body Area Network and Personal Area network for short range data transmission. Bluetooth is low tier, ad-hoc and terrestrial wireless standard for short range communication (100m, 10m and 1m). Since Bluetooth uses broadcast communication mode, devices in communication using this technology doesn’t have to be in line of sight with each other [50]. Unlike infrared, Bluetooth signals are omni-directional and can pass through walls. However, Bluetooth has low transmission range. One example where Bluetooth was used as a communication media in UAV research project can be found in [51] . In the research project, Bluetooth technology was used to connect computer workstation with Proxflyer quad-copter via Gumstix Bluetooth connection. Gumstrix is a miniature onboard computer in Proxfler quad-copter.

2.4.2) IEEE 802.15.4: ZigBee

ZigBee is an open worldwide mesh networking standard which permit hubs to be linked together through various pathways. ZigBee is cheap to be implemented, making it a perfect answer for some mechanical applications. Adding on the aforementioned advantage, ZigBee standard uses 128-piece AES encryption. Zigbee wireless technology was used in 'Towards A Swarm of Agile Micro Quadrotors' research project in University of Pennsylvania [29] to establish UAV to UAV and UAV to workstation connection.

2.4.3) IEEE 802.11: WiFi

802.11b, 802.11g, and 802.11n collectively known as WiFi standard are used to interface internet switches to gadgets like PCs, tablets and telephones. This group of standards can run on 2.4GHz ISM, 2.4 UHF and 5GHz SHF ISM radio frequency bands.

2.4.4) IEEE 802.16: WiMax

WiMax is acronym for Worldwide Interoperability for Microwave Access. This wireless innovation permits information to be exchanged at a rate of 30-40 megabits per second in indoor and outdoor environments. The standard was once utilized by a few mobile carriers, outstandingly Sprint, to convey remote information to its clients. Sprint, alongside many other transporters who utilized the standard, later changed over to utilizing quicker LTE 4G network systems for information exchange.

2.5) UAV signal routing and forwarding

The high mobility of UAV in a swarm causes the network topology to change more often, increasing the difficulty in routing amongst UAV. The intensive research has been carried and several routing protocols and algorithms have been proposed to solve MANET routing problem. These proposed protocols are classified either under Unicast, Multicast or Broadcasting [43]. Broadcasting is the simplest routing type on wireless channels. In broadcasting, message transmitted is received by all nodes within one-hop transmission range of the sender. The easiest way to send information to all nodes in network using broadcasting is by flooding [43] [52] . However, flooding may result into broadcast storm problem [52]. In Unicasting, a sender transmits packets to one receiver (one-to-one relationship) using wireless channel explained in chapter 2.4. Multicasting is used in a situation where UAV needs to propagate packets to many destinations (one-to-many relationship) without broadcasting. In multicasting, all the addresses of recipients are known by the sender unlike in broadcasting.

3.0) Related UAV Swarm Synchronization Algorithms and Applications

This chapter identifies models and technology used by other researchers when developing swarm synchronization algorithms and protocols. The motive is to choose the best approach that can be used in this research. Even though the aim is to develop swarm synchronization protocol for outdoor flying UAVs, the related work on non-Aerial swarms is also reviewed simply because the altitude parameter in Aerial UAV swarm modeling normally is treated separately to simplify synchronization algorithms. In other terms, most of swarm synchronization models are developed in 2 dimension (considering only *x* and *y* coordinates) in both Aerial and non-Aerial swarms and the agents in Aerial swarms normally computes altitude (*z* coordinate) separately using altitude sensors. Therefore, the models and technology used in non-Aerial swarm synchronization are also applicable in Aerial swarms. The third dimension (*z* coordinate or altitude) can be computed different as aforementioned or can be added as a variable in 2D models to model the behavior of UAVs in air interface or 3D space.

3.1) Aerial Swarms

The notion of UAV swarm synchronization or collective motion of robots can be dated back in 1994 by Mataric. Mataric [5], in robots swarm synchronization was using robot behavioral rules regarded as aggregation, homing, dispersion and safe-wandering. Robots in the study were able to position themselves using stationary beacons, detect obstacles in their vicinity and they inter communicated with each other using broadcasting communication mode. In recent works computer scientists are employing statistical physics concepts that simulate natural swarms to synchronize UAV swarms and other artificial swarms. Studies in [30] and [15] is based on or has the configuration similar to statistical physics flocking models of Reynolds.

Vasarhelyi *et .al* [30] presented a decentralized control algorithm for synchronizing ten (10) outdoor flying UAVs using GPS-vision. The study was investigating the impacts of all kinds of delays (communication delays) that can affect UAV swarm synchronization. For carrying out the experiments, MK Basicset L4-ME open source self-stabilizing quadcopter from

MikroKopter Germany was used. The quadcopter uses custom processor installed with 3D gyroscope, 3D accelerometer, pressure sensor, GPS receiver, 2.4GHz XBee unit for wireless communication. Wireless communication latency of about 0.4s ± 0.2s was recorded during testing. The wireless signal propagation range was about 10 – 100m using broadcasting mode. Literature in [30] highlights that, despite the accuracy and correctness of mathematical models used in control algorithms, communication latency between agents (UAVs) in swarm can cause some instabilities in swarm synchronization.

Unlike Vasarhelyi *et.al* [30], Sabine *et.al* [15] basing on Reynolds flocking Models developed 2D swarm synchronization algorithm for outdoor flying UAV. The intent of their research was to investigate the impact of communication range and motion constraints on the success of aerial flocking in reality. In addition to Cohesion, Separation and alignment rules used in [30], studies in Sabine *et.al* introduced migration rule to guide the swarm towards their destination point. The experiments of the study truly prove that communication range has an impact on swarm synchronization of flying UAV. According to their findings, small communication range of 80m and low turn rates close to 0.1 *rad/s* cannot fulfill synchronization. UAV swarm synchronization was archived using transmission range of 300m and turn rate of 0.7 *rad/s*. However, the implementation of the synchronization using high transmission range with low battery power usage in micro-UAVs is not possible. In addition, findings in [15] are subjective. Depending on the technology used during experiments, different results are likely to be gathered (e.g using quad-rotor UAV than fixed). Sabine *et.al* [15] was using wing fixed wing (80 cm wingspan) installed with WNDA3180 802.11n (wifi) device transmitting in 5GHz frequency. WNDA3180 wireless device was used for inter UAV communication implemented on Ad-hoc network architecture. Another distributed UAV swarm synchronization algorithm implemented for outdoor test bench is used in CARUS project. However, this project is not based of SPP (Self Propelled Particle) unlike algorithms presented in [5], [15] and [30].

The CARUS project was initiated with the intent of demonstrating the effectiveness of UAV swarm operation in monitoring places where incidents (like robbery) normally occur. CARUS project focus on outdoor swarm synchronization where Global Positioning System (GPS) is used for individual UAV state estimation within a swarm. CARUS is full distributed such that all communication and decision are made locally (on each UAV) without the intervention of controller at the ground station. The swarm consists of five autonomous VTOL multi-rotor micro-UAVs. The mode of communication used in the project is broadcasting and each UAV have a record of every UAV ID.

Another outstanding work was done by Hoffmann *et.al*. [45]. In all research project reviewed, these is the only project tailored to synchronize UAVs in outdoor and indoor test benches. Basing on Nash Bargaining, Hoffmann *et.al* managed to develop a decentralized algorithm for controlling three quad-rotors both indoor and outdoor. The quad-rotors for this research are equipped with sensors that read the current location of a UAV and sensors that observe the UAV vicinity. For the UAV state estimation, GPS is used in outdoor environments where-else overhead USB camera was used in union with hue blob tracking software for indoor environments experiments. In this research, the experimental results show that; UAVs using the algorithm and associative state estimation technology where able to avoid obstacles and maintain formation in tight areas.

Bürkle *et.al* [46] presented centralized swarm of 3 three UAVs. Their swarm is centralized because the UAV mission commands and communication are handled at the ground station unlike in [5], [15] and [30]. Two channels using different frequency band are assigned to each UAVs to establish communication with the ground station. Despite the success of the project in swarm implementation, this work inherits the setbacks associated with direct link networking architecture. The setback includes high communication latency amongst UAVs, the cost of communication in relation to frequency allocation grows exponentially with number of UAVs and the swarm has single point of failure.

Another centralized flocking algorithm for synchronizing UAV swarm system that inherits direct link setbacks was done by Steven *et.al* [47] . This algorithm is developed based on stochastic kinematic model and was tested in outdoor test bench using three fixed wing UAVs equipped with GPS device and initial navigation system. These UAVs were flying at constant speed and fixed altitude.

Adding on [46] and [47], Alex *et.al* [29] used high precision localization system called VMC (Vicon Motion Capture) to record the position of every UAV in a swarm at speed of 100Hz [29]. The high-level control and trajectory planning is executed in desktop base station where else onboard microprocessor is used for low-level control. In wireless communication, MHz and 2.4 GHz Zigbee independent transceivers are used to communicate with all UAVs in the swarm on Multi-Group UAV network architecture. The experimental result of the project shows that a swarm of small UAVs has high agility and can synchronize without and any problems in confined work space. The solution of research project in [29] was tailored for indoor test benches therefore cannot be used to synchronize swarm of outdoor flying UAVs.

3.2) Ground Swarms

Literature reviewed to study swarm communication and synchronization algorithms on ground swarm project includes Kobots swarm [48], the iRobot Swarm [49] and Swarm-bots Project [49].

Turgut *et.al* on their study were investigating the properties that influence self-organization of robots in two dimensional space using Kobots [48]. Even though the research was conducted indoor, investigated properties provide technical guidance in outdoor Aerial UAV swarm synchronization. The investigated properties where; (1) the amount and nature of the noise encumbering the sensing systems (2) the number of neighbors each unit had, and (3) the range of the communication. Adding on [30] and [15], Turgut *et.al* [48] found that the range of wireless communication determines the swarm size. During system testing, it was found that Kobots beyond wireless communication range could not be synchronized. Furthermore, the research revealed that, the motion of robot swarms can be influenced by externally guiding some of their members towards a desired direction [5].Figure 5 shows a picture of Kobots used in the research. Kobot is equipped with infrared (IR) sensor, IEEE 802.15/Zigbee wireless module having transmission range of about 20m and a Virtual Heading Sensor (VHS). VHS coupled with digital compass and Zigbee module is used to sense the absolute alignment of its neighboring robots where-else the IR module is used to distinguish between obstacles and other robots in the vicinity.

Figure 8. Picture of one Kobot (left) and of a group of seven Kobots (right). These robots were designed by the KOVAN research lab for use in swarm robotic studies.

Similar to [48] is iRobot Swarm project [49], iRobot Swarm project uses IR technology in local communication, obstacle avoidance and localization. The dispersion Algorithm, distributed mapping and localization algorithms are utilized in the control and communication of every robots in a swarm. Dispersion Algorithm defines how robots should maneuver with respect to its neighboring robots. If the robot is close to its neighbor, then the algorithm governs the robots to repel from each other. Another segment of the algorithm is used when robots are to occupy a space in an ordered manner. In this phase, the robots are divided into three groups which are frontier robots (adjacent to open space), wall robots (up against the wall) and interior node (neither frontier nor wall). The signal is sent to other robots by frontier robots in multi-hop manner. The recipient robots calculate the shortest path to the frontier robots and use it to align its self to the frontier robot. Distributed localization and mapping Algorithm is used by robots in target searching and map plotting. Using the dispersion Algorithm, distributed mapping and localization algorithms, 100 robots were able to distribute themselves in a space, form different shapes and plot the map successfully.

University of Brussels developed a swarm system named ‘The Swarm Bots Project’. Swarm Bots Project uses color communication system unlike Kobots [48] and iRobot [49] swarms.

The intent of researchers during Swarm Bots project implementation was to simplify control algorithms by focusing on emergent behavior of the swarm than complicated ad-hoc patterns. By using Co-operative hole avoidance algorithm, Swarm Bots are able to maneuver the holes and skip holes on the ground. Chain formation algorithm helps robots to form a chain between a nest and a prey. Pattern in chain always follows blue, green and yellow repeatedly. S-bot emits BGYBGYBGYBGY when moving away from the nest and BYGBYGBYG, when moving towards the nest. BGY is acronym for Blue Green and Yellow colors where–else BYG is acronym for Blue Yellow and Green.

3.4) Conclusion

The above literature has highlighted the significance of communication in swarm synchronization algorithms. Despite the accuracy and correctness of mathematical models used in control algorithms, failure to have reliable and robust communications may cause instabilities and lead to collision amongst UAV in a swarm [32]. In order to achieve the highest degree of synchronization, a collection of UAVs requires effective communication with one another and external control points. Mobile ad-hoc architectures are preferred over centralized architectures to address the problem of communication in outdoor micro UAV swarm systems. Ad-hoc architecture offers intercommunication amongst UAVs with low latency and boasts signal propagation on wireless cheap lightweight communication modules installed in UAVs. Furthermore, multi-hop wireless mobile ad-hoc network is suitable for transient network topology [50] , can minimize UAV power consumption and maximize UAV loitering time [51]. However, the effect of mobile ad-hoc network architecture on synchronization time has not been investigated. Another factor that needs to be addressed is lack of precise localization system in outdoor environments. This factor makes it difficult to consider tight formations synchronizing of UAVs in outdoor environments in presence of perturbations. Therefore, an optimistic outdoor flying UAV swarm synchronization algorithm with feed forward terms can help on synchronizing UAV swarms operating in this disadvantage [32]. Feed-forward in this context refers to the ability of a UAV to predict its velocity vector and position vector in next time step. Moreover, it is advantageous to mimic nature swarms such as birds and bees when developing swarm synchronization algorithms for outdoor test benches. Mimicking nature swarms allows the UAVs with deprived resource to interact together using simple rules without centralized control to produce complex swarm behavior [3]. The development of algorithms for synchronizing swarm of UAVs for outdoor test benches is at infancy and far to be used in practical application [6]. Therefore, this study focuses on developing a swarm synchronization protocol for outdoor flying UAVs using GPS technology. The SPP (self-propelled particles) models are used in the algorithm to predict system behavior at swarm level.

4.0) Unmanned Aerial Vehicle Swarm Synchronization Protocol (UAVSSP) Design

In order to answer the aforementioned limitation of outdoor Unmanned Aerial Vehicle swarm synchronization protocols, this thesis present a fully distributed leader-follower UAVSSP that can be used in real life outdoor UAV swarm applications. The protocol model the behavior of a UAV swarm using SPP (Self Propelled Particle) functions and MANET (Mobile Ad-hoc Network) is used to network UAVs within a swarm.

The SPP models are used to calculate the directional correlation between UAVs, to compute velocity vector and position vector of a UAV relative to a leading UAV. The models do not arrange UAVs into a formation .UAVs should be arranged in a desired formation prior to flight.

4.1) Protocol Requirements

- **Each leader is associated with only one follower and a follower can be a leader to another UAV**. The developed protocol is based on pair correlation approach whereby UAVs within a swarm allies in pairs. Each UAV within a swarm communicates with its leader and ignores other UAVs.

- **A leading UAV forwards its GPS coordinates(x, y, z), id and velocity to its follower**. In packet forwarding the protocol we developed uses back propagation scheme. The back propagation (top - down) scheme demands data to be transmitted by a leading UAV to its wing (follower) UAV. The data is used by wing (follower) UAV to calculate its position in a formation relative to its leader.

- **The UAVs executing the algorithm should be equipped with GPS module**. The GPS module is used for UAV state estimation (position and orientation).

- **Only one UAV can act as a backbone or gateway**. Only one UAV is required to connect UAV swarm to the ground control station.

- **A backbone or a gateway UAV is allowed to have more than one follower (wing UAVs).** This requirement is effective in formations (V or J) whereby backbone UAV is positioned in the middle. In this case, a backbone UAV will be able to communicate with a UAV on its left (left wing UAV) and right(right wing UAV).

4.2) Unmanned Aerial Vehicle Swarm Synchronization Protocol (UAVSSP) models (Self Propelled Particles)

Self-Propelled Particle functions are used in different professions to simulate swarm behavior. In biology, the models are used in simulation of bacteria colony, cell migration (e.g. tumor), where-else in chemistry the models are used in simulation of molecule migrations (osmosis and diffusion), in physics to study the relationship of variable and how relationship affect swarm behaviors [5]. The Unmanned Aerial Vehicle Swarm Synchronization Protocol (UAVSSP) presented in this thesis uses directional pair correlation function, and SVM(Standard Vicsek Model) SPP functions adopted from [5] to simulate the UAV swarm of outdoor flying UAVs.

$$\boldsymbol{C_{ij}\ (\tau)} = \langle \vec{v_i}(\boldsymbol{t}) \cdot \vec{v_j}(\boldsymbol{t} + \boldsymbol{\tau}) \rangle$$

Equation 1

The directional pair correlation function above (equation 1) shows to what degree particle $\boldsymbol{i}$ at time $\boldsymbol{t}$ is related to particle $\boldsymbol{j}$ at time $\boldsymbol{t} + \boldsymbol{\tau}$.The symbol $\vec{v_i}(\boldsymbol{t})$ denotes the normalized velocity of the *ith* at time $\boldsymbol{t}$ and $\langle \ldots \rangle$ denote averaging over time where else $\vec{v_j}(\boldsymbol{t} + \boldsymbol{\tau})$ represent normalized velocity of the *jth* particle at time $t + \boldsymbol{\tau}$.Tau $\boldsymbol{\tau}$ in the above equation 1 is a correlation time between particles (SPP) and *t* is an actual time. The result of the equation is used to govern the leader-follower relationship in a swarm. According to the equation 1, if the result of the computation is a negative value, it means *ith* particle velocity is dropping. Therefore, in this situation *jth* particle will be confirmed as a leader of the swarm.

In this thesis, the directional pair correlation function in equation 1 is used to govern the relationship of a leading UAV and its follower (wing), whereby, a leading UAV takes a position of the *ith* particle and a follower takes a position of the *jth* particle. In an occurrence whereby a negative value is returned by the function, UAVSSP has to force a wing UAV (follower or *jth* particle) to reduce speed. Since one wing (follower) UAV is associated with one leader in the Unmanned Aerial Vehicle Swarm Synchronization Protocol (UAVSSP), the relationship of a wing UAV and rest of other UAVs in a swarm is not implemented. The diagram below (Figure 9) shows a pictorial representation of equation 1.

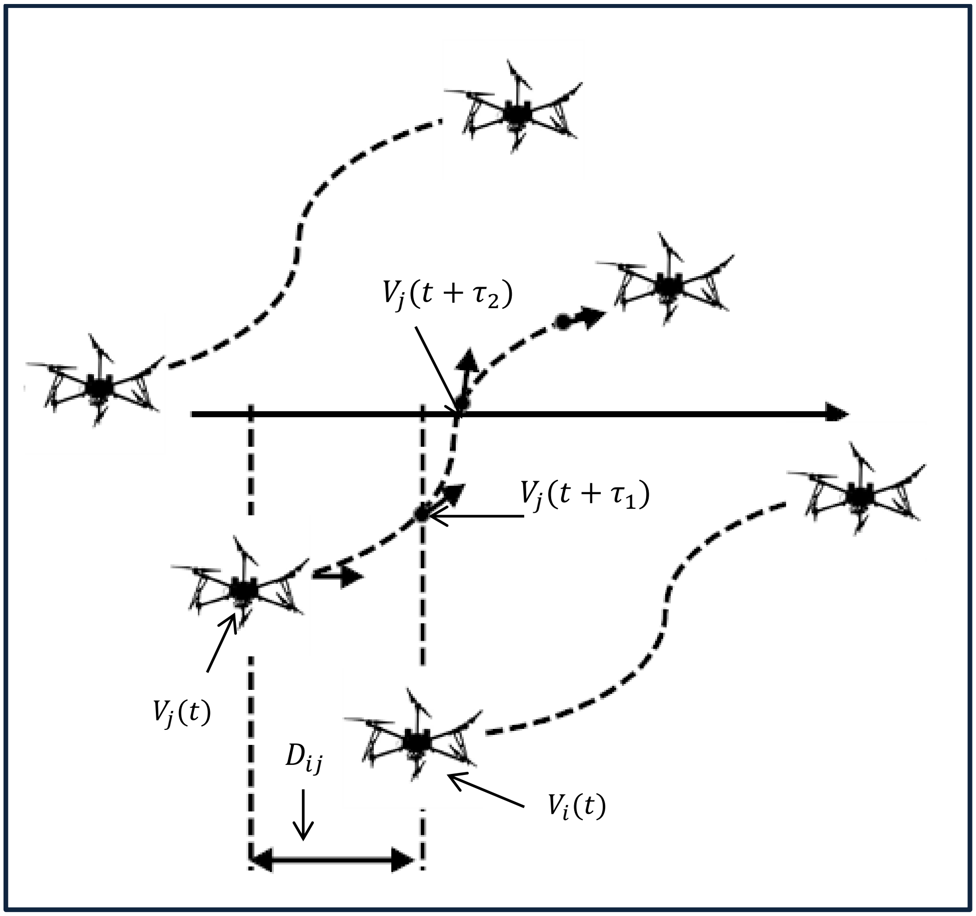


Figure 9: Diagram showing relationship between two UAVs under influence of the directional correlation function

Dij = ‖ d ‖ determines the projected distance between a leading UAV and its wing (follower) UAV whereby ‖ d ‖ = $\sqrt{\left(x_i - x_y\right)^2 + \left(x_i - y_y\right)^2}$. (x_i , y_i) represent *ith* UAV (leading UAV) coordinates and (x_y , y_y) represents coordinates of jth UAV(follower / wing UAV).

The SPP function in equation 5 and equation 6 in page 38 help in *feed-forward* calculation of velocity and position vectors in the next time step of the *jth* UAV. Unlike in [5] whereby the focus was on simulating flocking behavior of non-formation oriented swarm agents, this thesis focus on a swarm with defined formation. Therefore, as shown in equation 5 and equation 6, the functions are altered to calculate the position and velocity vector of *jth* UAV for the next time step in presence of perturbation.

Equation 2 below shows the original SPP function as presented in [5] before alteration.

$$\vec{V_i}(\boldsymbol{t+1}) = \boldsymbol{V_0}\frac{\langle\vec{V_j}(\boldsymbol{t})\rangle_{\boldsymbol{R}}}{\left|\langle\vec{V_j}(\boldsymbol{t})\rangle_{\boldsymbol{R}}\right|} + \boldsymbol{perturbation}$$

Equation 2

The equation 2 above compute the velocity vector of the *ith* particle at time $t + 1$ in cluster (non-formation oriented swarm) whereby the *ith* particle is surrounded by *N* number of particles in its radius denoted by *R*. Since *ith* particle movement is influenced by particles in its surrounding(within radius *r*), the velocity vector in time $t + 1$ of *ith* particle is calculated by multiplying velocity *Vo*(fixed absolute velocity the particles in a swarm move with) with average direction of motion of all particles in its proximity. The unit vector pointing the average direction of motion of particles in close proximity of *ith* characterized by its angle ϑ_j (t) within communication range is returned by $\frac{\langle\vec{V_i}(\boldsymbol{t})\rangle_{\boldsymbol{R}}}{\left|\langle\vec{V_i}(\boldsymbol{t})\rangle_{\boldsymbol{R}}\right|}$ in the above equation 2. The annotation $\langle\ldots\rangle_R$ represents the summation of particle velocities within radius *R* [5]. Perturbation was used in order to represent the unlikelihood that can occur in a system. Perturbation can be considered as a natural consequence of factors like wind affecting the motion of agents within a swarm [5]. Perturbation variable in above equation 2 is calculated by adding random angle to the average direction of swarm in time $t + 1$ as shown in equation 3 and 4 in page 37. Just like equation 1, equation 3 is not altered. The equation 3 is used as written in [5]. In equation 4, we replaced square brackets in order to make the equation comply with standard mathematical convention.

$$\boldsymbol{\vartheta_j(t+1) = \vartheta_j(t) + \Delta_j(t),}$$

Equation 3

Where $\vartheta_j(t)$ is calculated as

$$\boldsymbol{\vartheta_j(t) = arctan\left(\frac{\langle V_{i.x}\rangle_R}{\langle V_{i.y}\rangle_R}\right)}$$

Equation 4

In equation 3, $\Delta_j(t)$ represents perturbation in time *t*. In equation 4, $V_{i.x}$ and $V_{i.y}$ represents x and y coordinates of a leading UAV velocity. Perturbation represented by $\Delta_j(t)$ in equation 4, is a random parameter taken between $-\eta\pi$ and $\eta\pi$ whereby $\eta < 1$. The equation 3 computes the direction of motion of particle *j* at time *t+1* by taking in consideration the uncertainties that may affect the angle of motion. These uncertainties are regarded as perturbation. Perturbation is added to cater for the uncertainty such a wind that affects the direction of motion of agents within a swarm system

Equation 5 below shows the final SPP model after alteration from equation 2. The equation 5 predicts the velocity vector of *jth* UAV (UAV that follows a leader) in the next time step ($\vec{v_j}(t+1)$) in relation to *ith* UAV (leading UAV) whereby $j \neq i$ (in terms of *x* and *y* position coordinates). The UAVs are assumed to be moving at a fixed absolute velocity V_0 and each UAV assume the direction of its leader within its wireless communication range. We changed three things in the equation which are;

1) Removed the summation.

Since a UAV is correlated with only one UAV (a leading UAV) the summation is not needed. The unit vector that represents a leading UAV average direction of motion characterized by its angle ϑ_j(t) within communication range is $\frac{\vec{v_i}(t)}{|\vec{v_i}(t)|}$.

2) Swapped the position of *j* and *i* variables.

The original equation (equation 2) as presented in [5] simulate *ith* particle as a follower of *jth* particle. However, in our developed protocol, an *ith* particle represent a leading UAV where else *jth* particle represent a follower UAV(wing UAV). Therefore, we had to swap the position of *j* and *i* variables in the equation.

3) Removed perturbation.

The author in [5] were simulating the behavior of swarm agents theoretically, therefore, they introduced perturbation in order to represents likelihood in the system. In our study, we were interested in developing a protocol that is used in actual outdoor environments where parameters that are represented by perturbation are present. Therefore, we removed the perturbation from the equation 2.

$$\vec{v_j}(\boldsymbol{t}+\mathbf{1}) = \boldsymbol{V_0}\frac{\vec{v_i}(\boldsymbol{t})}{\left|\vec{v_i}(\boldsymbol{t})\right|}$$

Equation 5

The position of a *jth* UAV (a follower or a wing UAV) in time *t+1* is calculated using equation 5 below. The equation adds the position vector of *jth* particle in time *t* denoted by $\vec{x_j}(\boldsymbol{t})$ with its velocity vector at time *t + 1.*

$$\vec{x_j}(\boldsymbol{t}+\mathbf{1}) = \vec{x_j}(\boldsymbol{t}) + \vec{v_j}(\boldsymbol{t}+\mathbf{1})$$

Equation 6

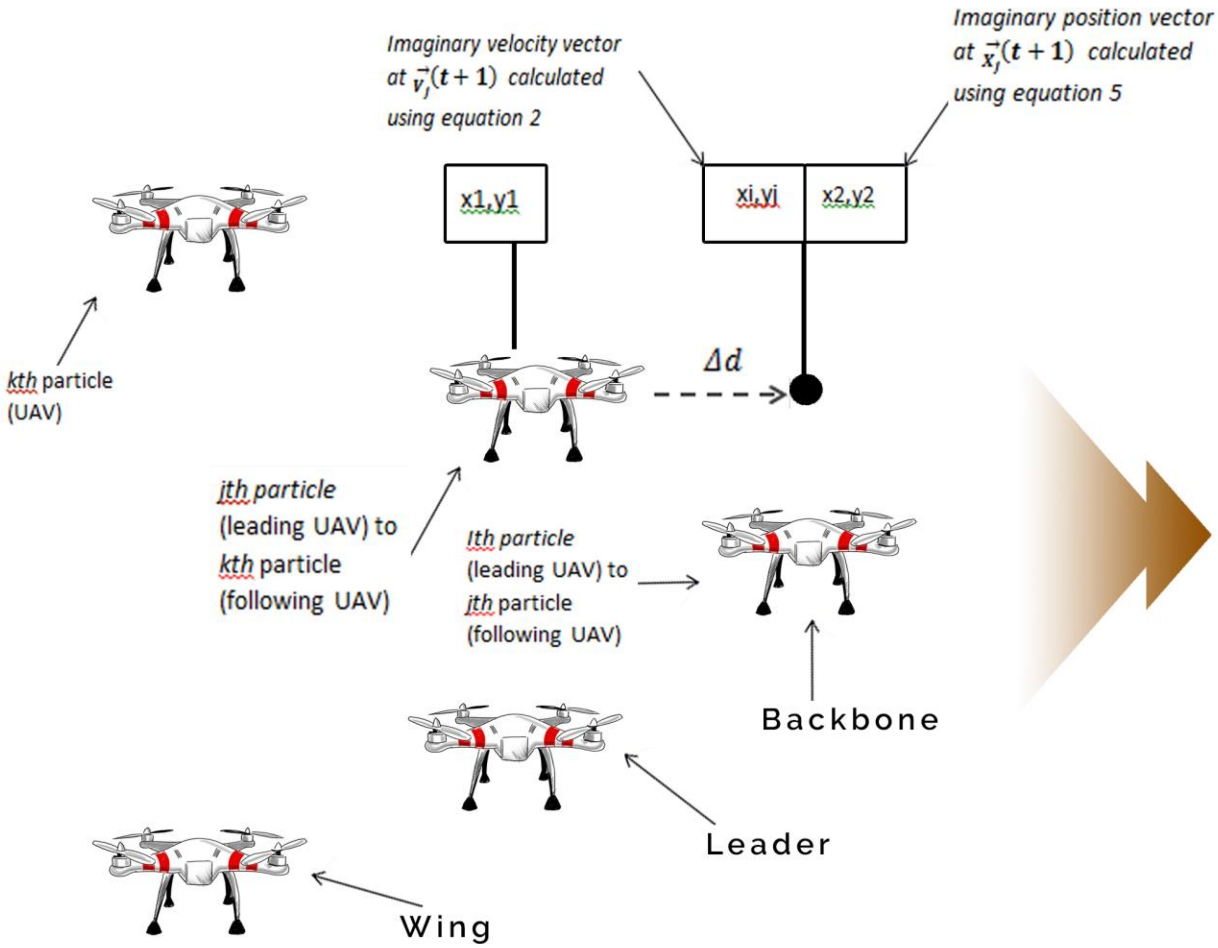


Figure 10 Unmanned Aerial Vehicle Protocol

Backbone UAV in the figure above links the swarm to the ground station. It is the only UAV that is allowed to be a leader of two UAVs in UAVSSP. As presented in figure 10, backbone UAV is a leader to the immediate UAV on its left (top on figure 10) and immediate UAV on its right (bottom on figure 10). UAVs labeled *leader* act as follower and a leader while UAVs labeled *wing* are UAVs without followers.

4.3) Unmanned Aerial Vehicle Swarm Synchronization Protocol (UAVSSP) Pseudo code

Protocol1 below (Algorithm 1) executes once and is used during UAV initialization. Before a UAV start to fly with others in a swarm, it should know its rank and its partner. They are three ranks that can be assumed by a UAV. A UAV can be a *backbone UAV, leader* or a *wing* (follower). Only one UAV can assume a rank of a *backbone* in a swarm. A *backbone UAV* is the main UAV that link a swarm and a controller or a ground station. Sometimes a UAV can act as a *leader* and as a *follower(wing)* at the sametime in a developed protocol. That kind of UAV is called a *leader* UAV. A *Wing* UAV has a leader but it does not have a follower.

The first line in protocol 1(algorithm 1) store coordinates retrieved from GPS module for the first time. Then coordinates are stored in *final.xO, final.yO* and *final.zO*. *final.xO , final.yO and final.zO* coordinates are used for safe landing in case were a UAV lost its communication with its leader or a control station. Line 3 is used by a UAV to request for a connection and a handshake with other UAVs. Handshake is a networking jargon used when two gadgets negotiates communication terms and condition.

If a UAV does not receive a response after requesting connection, a UAV becomes a *backbone UAV* (main leader) immediately then send connection request for five more times. If no response is received, then that means a UAV does not have a follower. Otherwise, responded UAV will be deemed as wing UAV. This procedure is explained in line 6 to 15 of protocol 1 (algorithm 1) below.

Furthermore, 16 through 43 explain what happens when a UAV is not a *backbone UAV* or a *leader*. A UAV send acknowledgement feedback to its leader. An acknowledgement feedback contains UAV unique identification number. A pair correlation will be established at this level. However, a UAV needs to check whether to assume a *leader* or a *wing* rank. In this case, a UAV will request for a connection and handshake, if it does not receive acknowledgement from any UAV then it assumes *wing* rank else *leader* rank.

The LUA source code for the protocol 1(algorithm1) is included in appendices 1. During the experiments, the protocol 1 LUA source code was combined with LUA source code or control algorithm that comes with qudricoper model in V-REP.

Protocol 2 (Algorithm 2) below is the main protocol that controls a UAV. This protocol will be executed as long as a UAV is flying. The first line re-initialize coordinates(*x,y,z*) with new readings from GPS module. The re-initialization is necessary because is assumed that a UAV has made a displacement from first point to another. The protocol then checks whether a UAV executing the protocol is a *backbone UAV* or not. If a UAV is a *backbone UAV* then line 3 is executed. Line 3 broadcast UAV coordinates and velocity at time *t*. How a *backbone UAV* maneuvers is not specified in a protocol because is assumed a leader is taking its mission commands from the ground station.(Visit appendices 2 to see protocol 2 LUA program).

If a UAV has assumed a rank of *leader*, then line 7 to 28 will be executed. Line 7 reads *controlSignal* sent by its leader. Control signal consists of *x,y,z* Coordinates and *velocity* at time *t* . If a *controlSignal* is not sent or if connection cannot be established, then a UAV will attempt to read the signal again for 5 times then a UAV will trigger safe-land mode. Safe-land mode takes a UAV to its initial coordinates which are stored in *final.xO*, *final.yO* and *final.zO* variables. The safe-land mode is crucial to UAVs synchronization since the developed protocol is not fault tolerant.

Line 16 to 28 is executed only when a UAV has received *controlSignal* from its leader. New coordinates will be stored in leader.x2, leade.y2 and leader.z2. These new coordinates together with old coordinates will be used to find the distance displacement of a UAV in time *t*. Condition statement at line 19 uses directional pair correlation function (explained in protocol 4) to test if the direction of a leading UAV is not falling behind. If the value returned by a function is less than zero, then current UAV (UAV running the protocol) velocity will be reduced in line 20.

Line 23 and 24 calculates the velocity and position vectors in time *t + 1*. This prediction of velocity and position vectors in next time step is called *feed-forward terms* in this thesis. Line 25 flies UAV to the predicted position vector in time *t +1*. Since coordinates(leader.x2,leader.y2,leader.z2) will be outdated after computations in line 23, 24 and 29, the coordinates will be stored as old coordinated(leader.x2,leader.y2,leader.z2) in line 27.

Protocol 3(Algorithm 3) below is based on equation 2 to calculate velocity vector of a wing UAV at time t + 1. *fVelocity* is a fixed absolute velocityV_0. Variable *distance* holds the displacement of a UAV between two points. Velocity vectors *xC* and *yC* are calculated in line 4 and line 5.

Algorithm 3: velocity(leader.x, leader.x2, leader.y, leader.y2)

```
1 velocity ← fVelocity
2 answer ← math.sqrt((leader.x2 - leader.x) + (leader.y2 - leader.y))
  // computes distance between vectors
3 distance = math.root(answer) xC ← ((leader.x/distance) ∗ velocity)    // x
  velocity vector at t + 1
4 yC ← ((leader.y/distance) ∗ velocity)          // y velocity vector at t + 1
5 return(xC, yC)
```

```
---Function for calculating velocity vector at time t + 1--
function velocity(leader_x,leader_x2,leader_y,leader_y2,fVelocity)
{
   --fVelocity is a leader's velocity
   velocity = fVelocity
   answer   = ((leader_x2 - leader_x)* (leader_x2 - leader_x)) +
((leader_y2 - leader_y)))
   --Calculate distance between vectors
 distance = math.sqrt(answer)
   -- Calculate velocity vector at t + 1
 xC = (leader_x2 / distance)* velocity
 yC = (leader_y2 / distance)* velocity
 return(xC,yC)
}    ---------------
```

Figure 11: LUA code calculating velocity of a UAV at time t+ 1.

Using velocity vectors calculated in protocol 3,protocol 4(Algorithm 4) below calculate position vector at time *t + 1*. Position vector at time *t +1* is calculated in line 1 and line 2. This protocol was established from equation 7 in sub section 4.2.

Algorithm 4: position(xC, yC,me.x,me.x)

```
1 xP = (me.x + xC)                        // x position vector at t + 1
2 yP = (me.y + yC)                        // y position vector at t + 1
3 return(xP, yP)
```

```
function position(xC,yC,me_x, me_y)
{
      --Calculate x and y cordinates at time t+1
      xP = (me_x + xC)
      yP = (me_y + yC)
      return(xP,yP)
 }
```

Figure 12: Algorithm 4 translated into LUA code. The code was calculating the imaginary position of a drone in time t + 1

The below protocol (Algorithm 5) is a directional pair correlation protocol. Based on equation 1, the protocol is used to model the relationship between a leader and wing (follower) UAV. If the value calculated in line 1 is below zero that means the direction of a leading UAV is falling behind.

Algorithm 5: Cij(xC, yC,leader.x, leader.y)

1 $value \leftarrow (xC * leader.x) + (yC * leader.y)$

2 $return(value)$

```
--A pair correlation function
 function Cij(xC,yC,leader_x,leader_y)
 {
        value = (xC * leader_x) + (yC * leader_y)
        return (value)
 }
```

Figure 13: directional pair correlation LUA function

4.4) Unmanned Arial Vehicle Swarm Synchronization Protocol (UAVSSP) Networking

The protocol uses back propagation communication scheme built on Mobile Ad-hoc Network (MANET) architecture. The scheme is called back propagation because hierarchy is observed when data is transmitted. Required data by a *wing* UAV to compute its position in a swarm is back propagated by its leader but not vice versa. Figure 14 below shows how data is transmitted in back propagation scheme. The three rectangles represent a swarm of three UAVs (UAV 1, UAV 2 and UAV 3). UAV 1 which is a *gateway UAV* transmits its velocity and relative position coordinates(x, y, z) to UAV 2. UAV 2 uses velocity and coordinates sent by *gateway UAV* as inputs to calculate its velocity, angle and position vector at time (t + 1). In the same way, UAV 2 transmits its velocity and relative position coordinates(x, y, z) to UAV 3.

Mobile Ad-hoc network is a local area network (LAN) that is assembled suddenly as mobile devices or gadgets such as UAVs associate. The gadgets in the MANET act as mobile routers, therefore the network coverage increases exponentially with the number of gadget participating in a network. MANET communication is based on multi-hopping strategy in which gadget sends message to its neighboring gadgets then the message will be hopped (forwarded) to other gadgets in likewise manner. MANET is a convenient architecture for the developed protocol simply because it offers better coverage and is cheap to be deployed than other centralized architectures. The protocol developed in this research is assumed to be used by UAVs equipped with lightweight off-shelf wireless modules. Even though these wireless modules are cost effective, they cannot propagate packets for longer distance. In order to maintain reliable UAV inter communication within a swarm, special wireless modules should be used to transmit packets over a long distance or to use MANET architecture as an alternative. Custom wireless modules consume too much battery energy when transmitting signals therefore reduce UAV loitering time. MANET guarantees communication coverage using off-shelf wireless modules. In this case, battery energy is not over used and UAV hovering time is maximized.

In contrast, with traditional MANET implementation that utilizes multi-hop strategy in data transmission, the MANET implementation of the developed protocol (UAVSSP) is single hop. Single-hop transmission implies that, a data from a specific node cannot be retransmitted to other nodes in a network. In case of the developed protocol, a leading UAV back propagate its position(x, y, z) to its follower alone using unicast or broadcasting transmission mode.

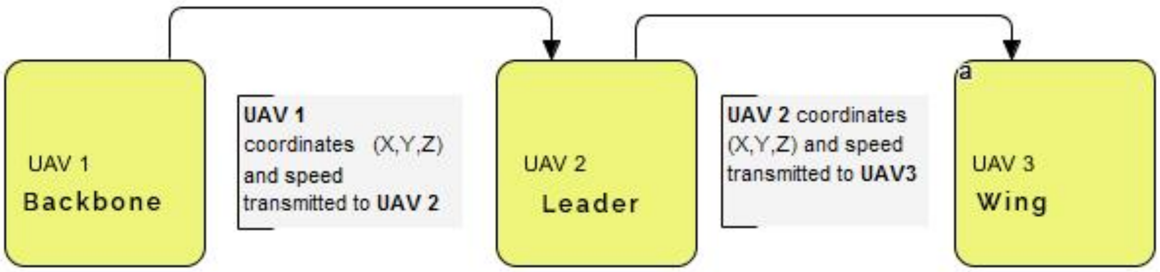


Figure 14

4.4.1) Datagram

Figure 15 below present a pictorial view of Unmanned Aerial Vehicle Swarm Synchronization Protocol (UAVSSP) datagram or data unit. The datagram defines the data structure of a packet that has to be sent during UAV inter communication. It is assumed that the packets are sent using Transmission Control Protocol (TCP) sitting on Internet Protocol (IP).TCP gives dependable packet stream conveyance benefit between nodes. However, the protocol is not time sensitive. In cases where time is critical, UAVSSP should be used in User Datagram Protocol (UDP). The only disadvantage of using UDP is; UDP does not attempt to rebroadcast lost packet as compared to TCP. Figure 16 below shows how UAVSSP datagram sits in TCP/IP or UDP/IP datagrams

← UAVSSP Datagram →					
Header		**Data**			
Sender ID	Receiver ID	X Coordinates	Y Coordinates	Z Coordinates	Velocity

Figure 15

← TCP/IP or UDP/IP Datagram →		
IP Header	TCP / UDP Header	**UAVSSP Datagram**

Figure 16: Shows how Unmanned Aerial Swarm Synchronization Protocol (UAVSSP) sits in TCP/IP or UDP/IP datagram

UAVSSP Datagram in figure 15 above is divided into two main compartments which are header and data. A header compartment contains identification numbers of a leading UAV and its wing(follower). A leading UAV identification number is stored as *sender ID.* Where-else the *Receiver ID* contains identification number of a wing UAV or follower. The provision of *Receiver ID* is optional depending on the casting mode used. When unicast is used in data transmission *Receiver ID* is obligatory but when broadcasting is used *Receiver ID* is not needed.

Data compartment contains information from a leading UAV. The information is used by a *wing* UAV to calculate angle of direction, velocity vector and position vector in time $t + 1$. x in data compartment contains leading UAV latitude number where-else y contains longitude and z contains altitude number.

4.4) Simulation Environment

Free educational licensed Virtual Robot Experimentation Platform (V-Rep) was used in the study to simulate the experiments of the developed UAV swarm synchronizing protocol (UAVSSP). V-Rep is a 3D universal platform developed by Coppelia Robotics [52] [53] for simulating robot systems. In favor of this research, V-Rep offers free educational license which is available for Windows, Linux and Mac OS [52]. The simulator uses less computational resources even in complex simulations, and has the ability to simulate many robots at the same time. Furthermore, V-Rep platform embeds diverse devices that allow fast algorithm coding and the code can be transferred in an actual robot without complications [54], algorithms for controlling V-Rep Models can be implemented in different programming languages(C/C++, Python, Java, Lua, Matlab or Urbi) [52] [55]. Adding on the above mentioned V-REP strength, the platform utilizes distributed control architecture [52]. Distributed control architecture allows models within the simulation to be individually manipulated using plugins, embedded scripts, remote API client and Robot Operating System (ROS) nodes [55]. Furthermore, the platform uses a custom rendering engine and gives bolster for three physics engines which are Bullet, ODE, and Vortex. The three physics engines can be used interchangeably during the simulation.

Robot Operating System (ROS) Nodes offers ROS functionality within V-Rep platform(simulator) that other ROS nodes can communicate with [56].A plugin or a module is a common library that V-Rep loads automatically at start-up or alternatively, plugins can be stacked using *simLoadModule* and discharged using *simUnloadModule*. In V-Rep, Plugins provides an interface to a hardware gadget. ROS Nodes are as well used in functions that require fast computations and can be utilized as a wrapper for executing algorithms written in different dialects (programming languages) or even composed for different microcontrollers [65].

Embedded scripts are another attractive feature of V-REP simulation platform [57].The default script of V-REP also regarded as main-script is LUA. LUA script language was used in this research to code the developed protocol because is simple to be programmed, fast, and easy to be embedded and offers a powerful application programming interface (API) that empowers LUA to communicate with other programming languages. According to [57], LUA is one of the script language gaining niche in robotics, game development and image processing.

Figure 17 shows the V-Rep screen displaying an array of seven floating (flying) UAVs during simulation. The simulation was executed to test the performance of Drone Swarm Synchronization Protocol (UAVSSP) in V formation. During the simulation, UAVs were floating at speed of 50 m/s and altitude of + 0.5180.

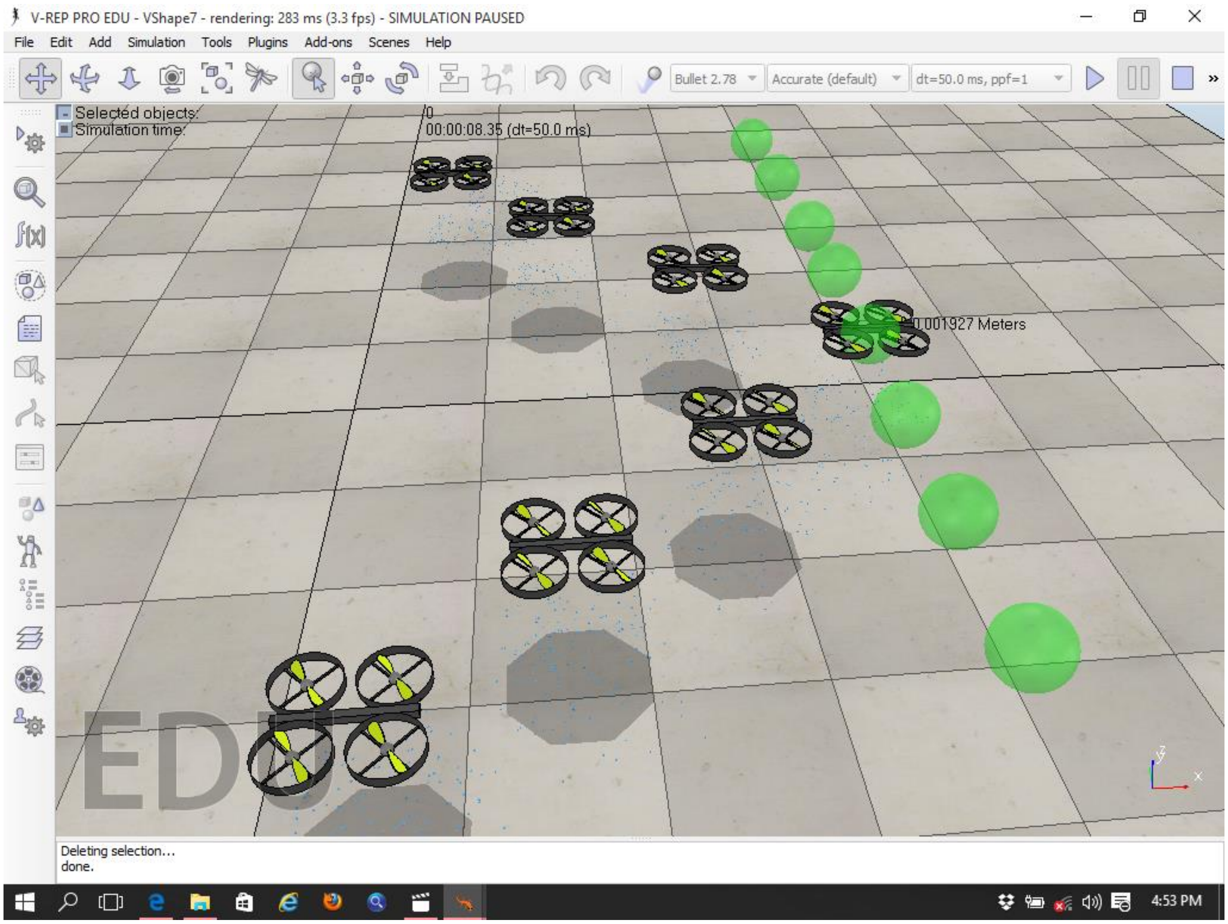


Figure 17: V-Rep screen displaying seven floating quadricopters arranged in V formation

Figure 18 is a presentation of the V-Rep screen displaying an array of eight floating (flying) UAVs during simulation. The simulation was executed to test the performance of UAVSSP in J formation. During the simulation, UAVs were floating at speed of 50 m/s and altitude of + 0.5180.

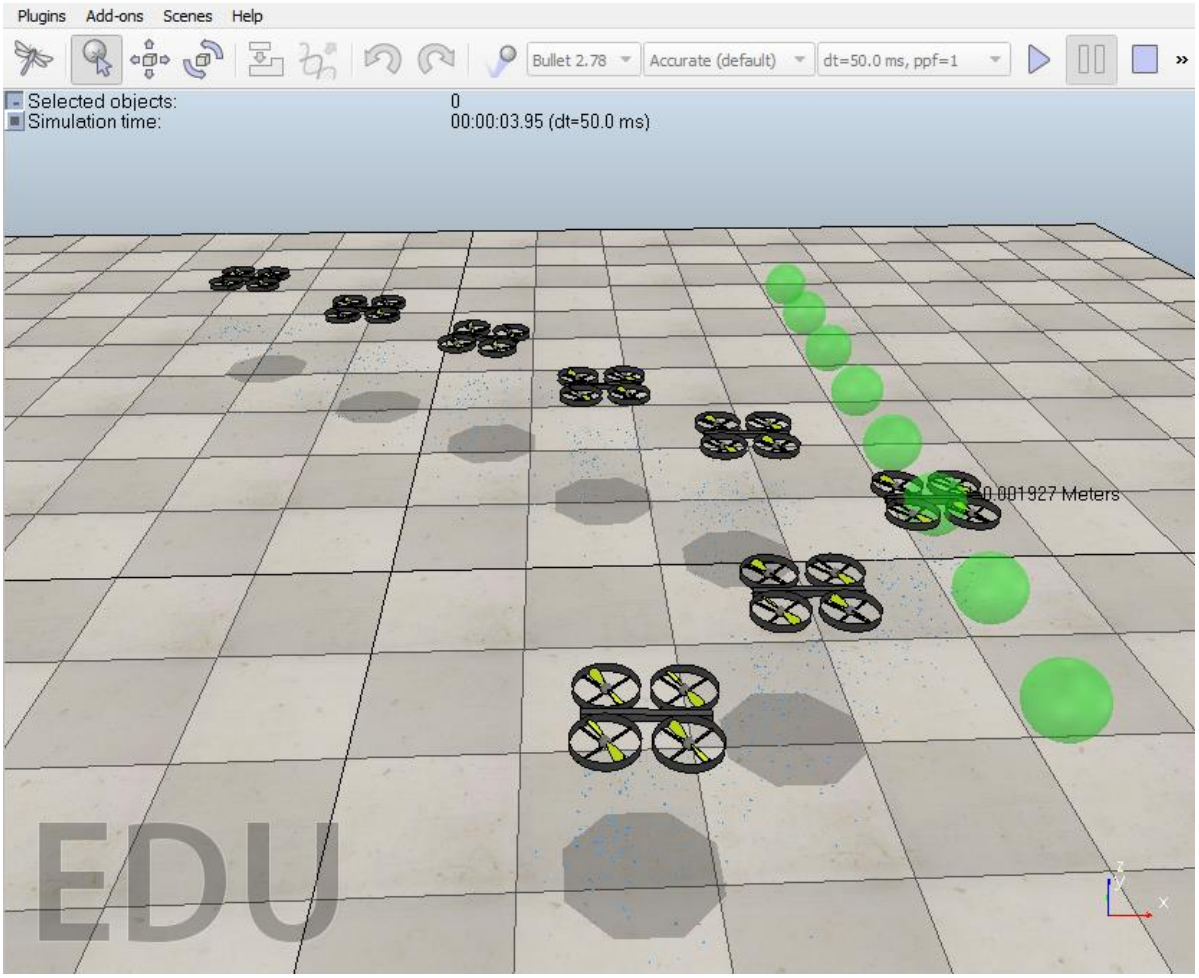


Figure 18: V-Rep screen displaying eight floating quadricopters arranged in J formation

5.0) Experiments and Results

This chapter presents the simulated experimental results of Unmanned Aerial Vehicle Swarm Synchronization Protocol (UAVSSP). 64 bit Intel® core™(@ 2.00GHz) i7-2630QM CPU Toshiba Harman/Kardon running on windows 10 pro with Random Access Memory of 4GB was used during the simulation . The experiments were carried in order to test performance of the developed protocol where maximum of thirteen (13) (UAVs) were used as a test bed. The performance of the protocol was tested as a function of time *t* against the V and L formations. These formations are used by birds when migrating. According to [66], the aforementioned formations has aerodynamics advantage on aerial natural swarms. As an example, literature in [66] results prove that birds are able to save energy, have better visuals and fly for longer distance when using these formations. This thesis however, investigates: 1) the cost of wireless transmission range on the synchronization time of UAVs in different shapes using Unmanned Aerial Vehicle Swarm Synchronization Protocol (UAVSSP), 2) the cost of networking topology on the synchronization time of UAVs using Unmanned Aerial Vehicle Swarm Synchronization Protocol (UAVSSP) and 3) communication error rate using Unmanned Aerial Vehicle Swarm Synchronization Protocol (UAVSSP).

The cost of wireless transmission range on the synchronization time of UAVs factor was tested using the transmission range of 1.5m (minimum) and 3m(maximum) in the experiments. The intent is to understand the relationship between wireless transmissions coverage and the formation of the swarm as well as, how these two constructs (transmission and formation) affect the swarm size, swarm synchronization time and swarm stability time on the developed swarm synchronization protocol. Swarm stability in this thesis is defined as a degree at which an agent (UAV) is able to position its-self within group of other UAVs or in a swarm.

The significance of matrix 2 is to test whether networking topology used in the developed protocol has any positive impact on the UAV swarm synchronization time or the overall performance of Unmanned Aerial Vehicle Swarm Synchronization Protocol (UAVSSP). To test this, the Unmanned Aerial Vehicle Swarm Synchronization Protocol (UAVSSP) topology was tested against star networking topology. Topology brings positive impacts only if it grantees less communication error in a limited wireless transmission range, less swarm synchronization time and if the topology has the ability to host many UAVs.

In the star topology, only one UAV amongst other UAVs in a swarm is a leader. In other terms, the topology has *one is to many* relationship implementation. Where-else in Unmanned Aerial Vehicle Swarm Synchronization Protocol (UAVSSP) topology, the relationship between a follower (wing UAV) and a leader is *one is to one.* One follower is assigned to one leader, and a follower can be a leader to another UAV. A leader in both topologies transmits its position(x, y, z) and velocity to its follower or followers. Other UAVs (follower or followers) calculates their positions using inputs from a leader.

Matrix 3 (communication error rate) in run time is associated with communication failure [60] occurred during UAV intercommunication or communication between a UAV and ground station. The matrix tests to what degree communication failure affects the performance of the developed swarm synchronization protocol (UAVSSP). In protocol presented by this thesis, communication error occurs only when a UAV fails to receive packet transmitted by its leader. They are several factors that can result communication error in UAV intercommunication. Communication error might occur when a UAV is loitering or flying outside communication range of its leader. This kind of communication error is called *out of range communication error*. Sometimes communication error occurs due to defect in wireless communication module. For this thesis, communication error rate measured in the experiments was based on *out of range communication error*. The error rate (ε_r) was calculated using equation 8 where *r* is the number of received (by a follower) packets, *s* is number of sent (by a leader) packets in time *t* and *n* represents the total number of UAVs in a swarm excluding a *backbone / gateway* UAV. The result of the equation is an averaged error rate of communication failures occurred in all UAVs partaking in a swarm at time *t*. $\left[\frac{\sum_{i=1}^{n}\left(\frac{r}{s}*100\right)}{n}\right]$ Portion of the equation gives us the communication success rate in percentages. Therefore, success rate is subtracted from 100 in order to have the averaged error rate. *100* represent the overall packets sent by a leader to its follower or followers at time *t* in percentage.

$$\varepsilon_r = 100 - \left[\frac{\sum_{i=1}^{n}\left(\frac{r}{s} * 100\right)}{n}\right]$$

Equation 6

The figure below shows communication log of thirteen UAVs at simulation ran time. Communication log captured displays the number of packet sent by a leader and number of UAVs sharing data as well their positions with respect to each other. The '*data received'* and '*data not received'* statements next to the UAV position indicate that the packet is received or not received. If simulation environment indicate '*data not received'* that shows communication error has occurred.

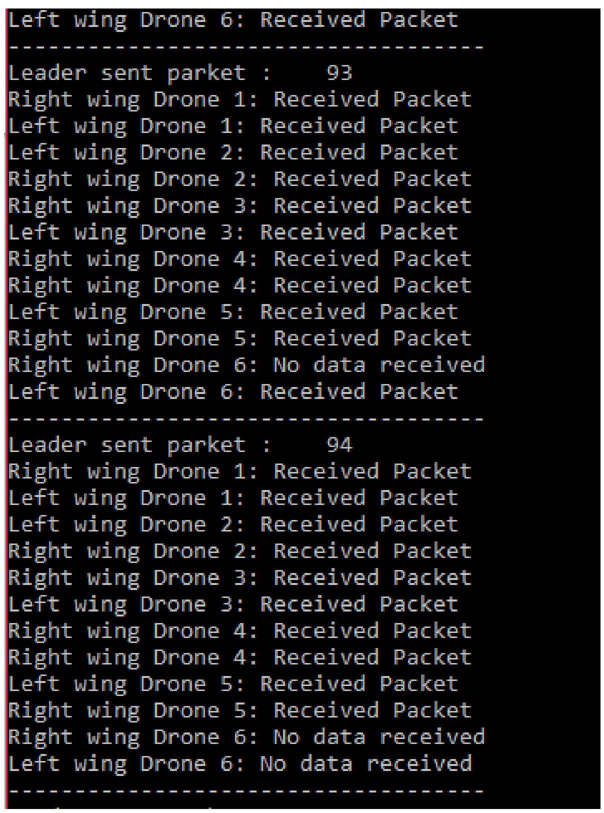
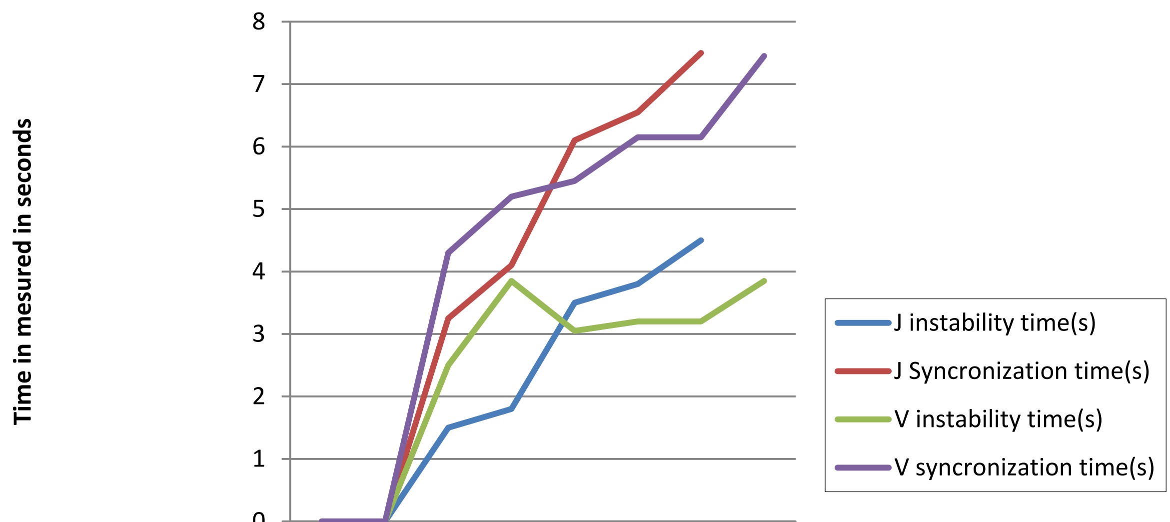


Figure 19

Figure 19 above shows that unreliable communication was experienced starting from packet 93 on right wing drone number 6 and in packet number 94, left wing drone number 6 six started to experience communication breakdown. Right wing and left wing state the position occupied by a UAV in relation to a leading UAV.

5.1) Experimental results 1

The results presented in figure 20 and 21 below shows the correlation of communication error rate, swarm instability rate and synchronization time factors. Figure 20 and 21 again shows how the correlation affects the overall performance of the Unmanned Aerial Vehicle Swarm Synchronization Protocol. The experiment was implemented using single hop MANET routing and minimum wireless coverage range of 1.5m.



Swarm Size ->	1	2	3	4	5	6	7	8
J instability time(s)	0	0	1.5	1.8	3.5	3.8	4.5	
J Syncronization time(s)	0	0	3.25	4.1	6.1	6.55	7.5	
V instability time(s)	0	0	2.5	3.85	3.05	3.2	3.2	3.85
V syncronization time(s)	0	0	4.3	5.2	5.45	6.15	6.15	7.45

Figure 20: Wireless communication range = 1, Routing = single hop

Figure 20 above show a minimum of 3.25s and a maximum of 7.50s time taken in J formation for Unmanned Aerial Vehicle Swarm Synchronization Protocol (UAVSSP) to synchronize a swarm size of seven UAVs as compared to V formation with a minimum of 4.30s and maximum of 7.45s to synchronize a swarm size of eight UAVs. In swarm size of three to four UAVs, V formation was out performed by J formation. The poor performance was resulted by high instability rate of UAVs experienced in V formation. The V formation swarm stability rate improved beyond swarm size of four hence increasing the performance of V formation against J formation .The results show that UAVSSP was not able to synchronize UAV swarm with communication error rate of 4.6+ % in both V and J formations (see figure 21 below). This implies that swarm instability affects swam synchronization negatively.

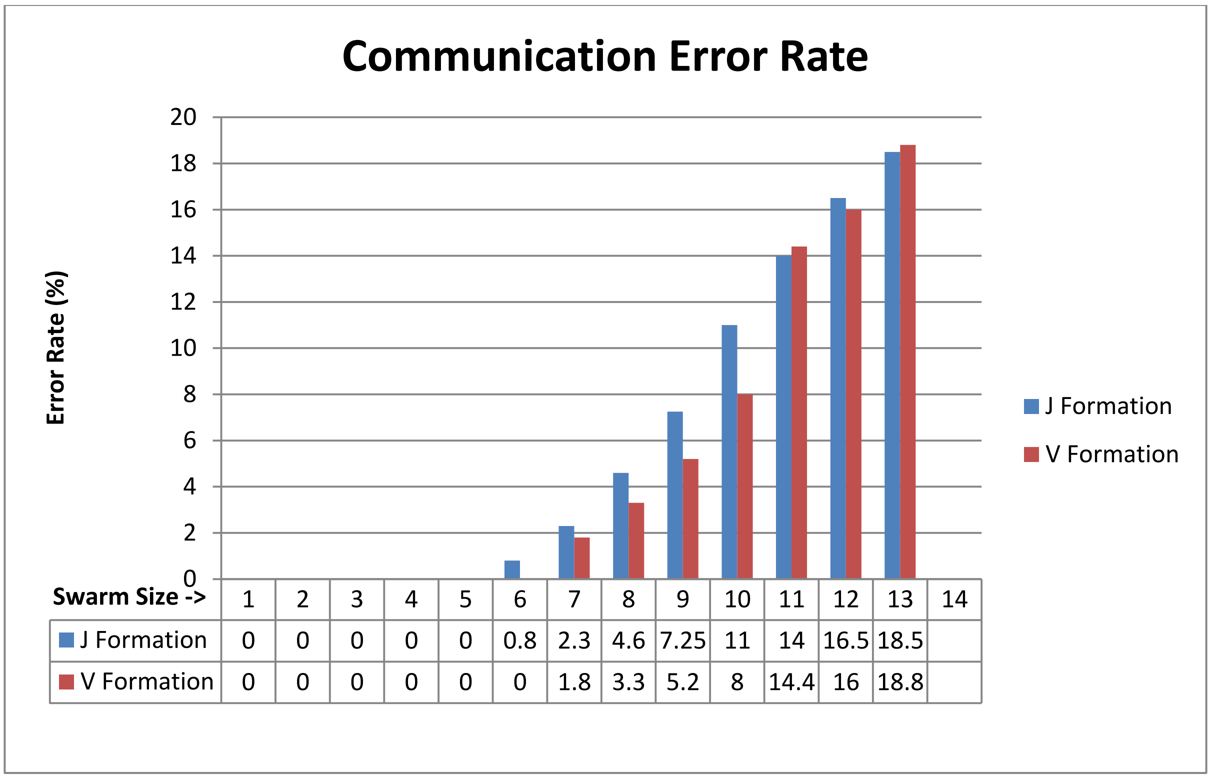
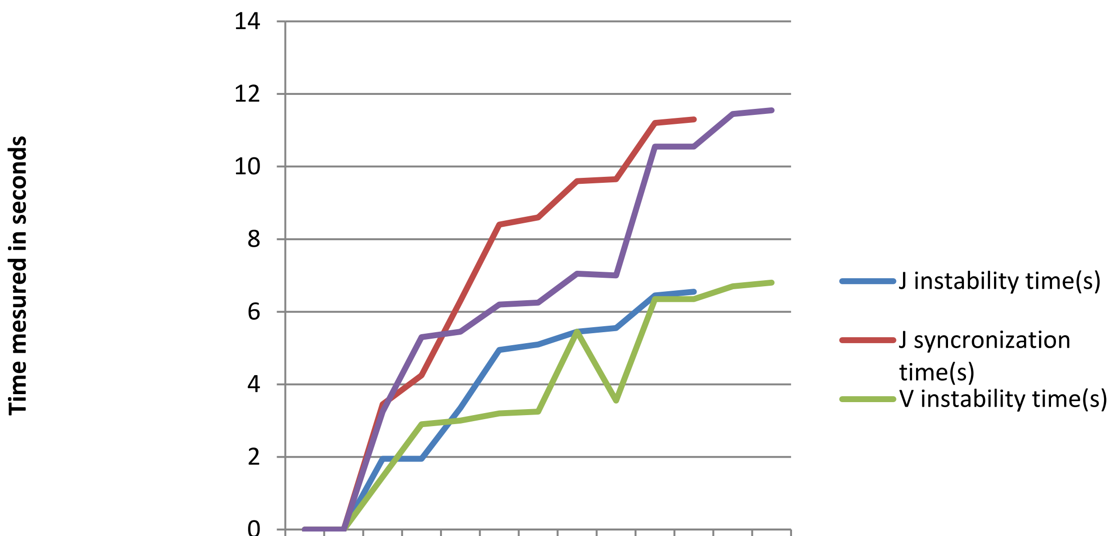


Figure 21: Wireless communication range = 1, Routing = single hop

Figure 21 above demonstrates that the communication error rate increases with number of UAVs participating in a swarm. Moreover, figure 21 demonstrate that J formation has high communication error rate than V formation. J Formation recorded a minimum of 0.8 % error in swarm of six UAVs and a maximum of 18.5% error rate in swarm size of thirteen UAVs. On the other hand, V formation recorded a minimum of 1.8 % in swarm size of seven UAVs and 18.8% in swarm of thirteen UAVs. Again the result demonstrates that communication error rate of 4.6 result distortion in UAV swarm synchronization using Drone Swarm Synchronization Protocol (UAVSSP).

5.2) Experimental results 2

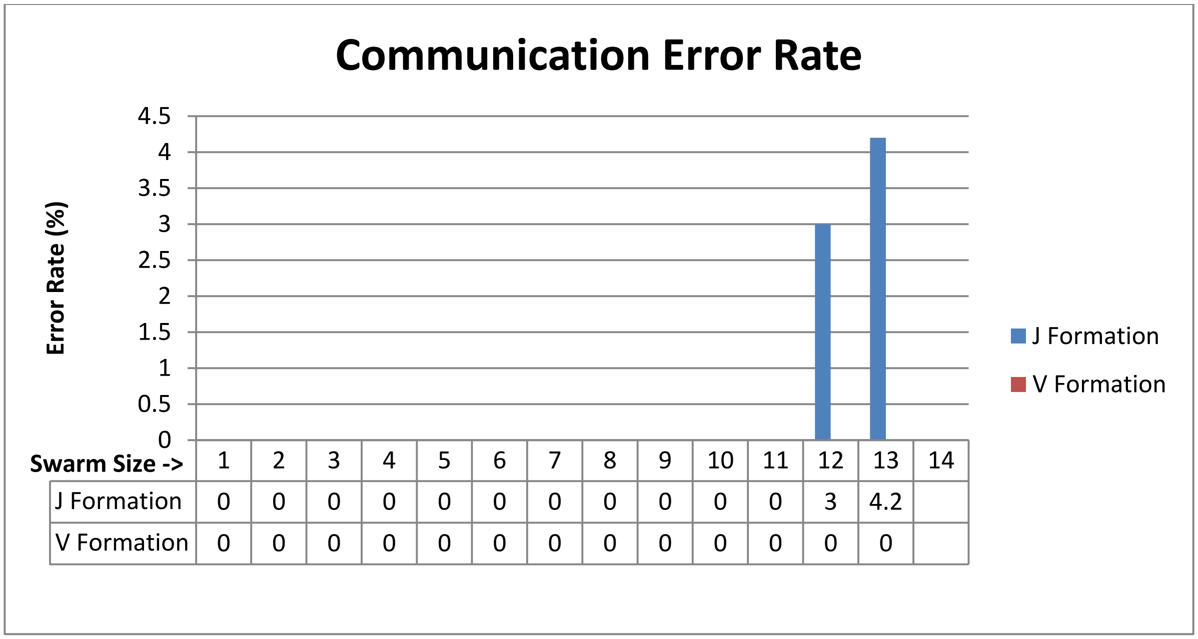
Figure 22 below shows the results of the second experiment using 3m wireless transmission range and single-hop MANET transmission.



Swarm size ->	1	2	3	4	5	6	7	8	9	10	11	12	13
J instability time(s)	0	0	2	2	3.4	5	5.1	5.5	5.6	6.5	6.6		
J syncronization time(s)	0	0	3.5	4.3	6.3	8.4	8.6	9.6	9.7	11	11		
V instability time(s)	0	0	1.5	2.9	3	3.2	3.3	5.5	3.6	6.4	6.4	6.7	6.8
V syncronization time(s)	0	0	3.3	5.3	5.5	6.2	6.3	7.1	7	11	11	11	12

Figure 22: Wireless communication range = 2, Routing = single hop

This second sets of experiments demonstrate that good wireless communication coverage improves swarm synchronization performance. This can be observed by 1) contrasting the number of UAVs synchronized in two formations and 2) contrasting synchronization time recorded in figure 20 and figure 22. In figure 20, it took maximum time of 7.5s to synchronize swarm of seven UAVs into J Formation and maximum time of 7.45 to synchronize swarm of eight UAVs into V formation as compared to the results gathered in these second sets of experiments. The results of this experiment presented in figure 22 shows that it took a maximum time of 11.3s for UAVSSP to synchronize swarm of eleven UAVs in J formation and maximum time of 11.6s for the protocol to synchronize swarm of thirteen UAVs.



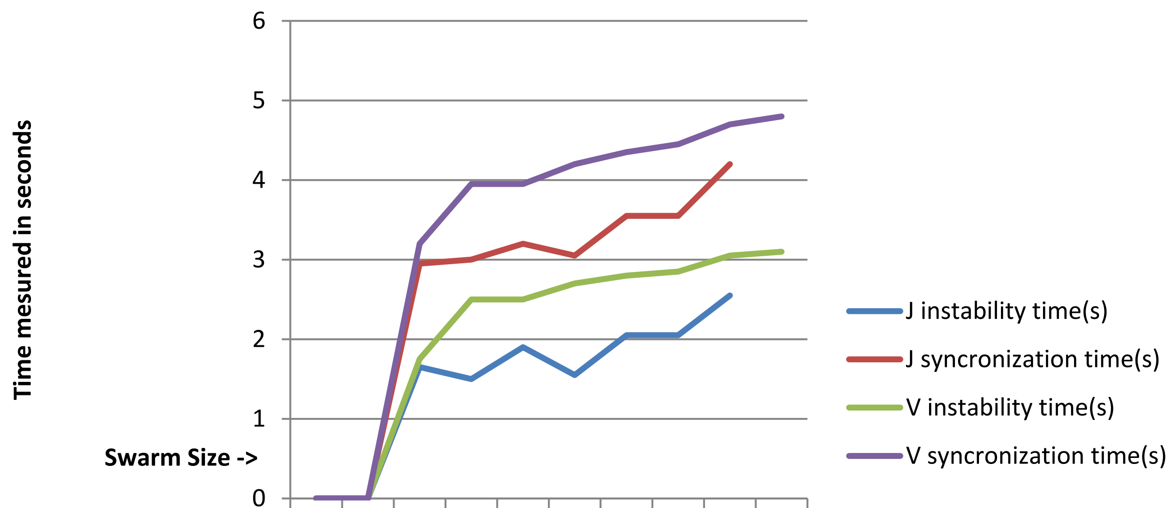
Swarm Size ->	1	2	3	4	5	6	7	8	9	10	11	12	13	14
J Formation	0	0	0	0	0	0	0	0	0	0	0	3	4.2	
V Formation	0	0	0	0	0	0	0	0	0	0	0	0	0	

Figure 23: Wireless communication range = 2, Routing = single hop

The plot in figure 23 above demonstrates less communication error rate is recorded in second sets of experiment than the first sets of experiment. This is as a result of good wireless transmission coverage. The communication error was only recorded in J Formation hence only eleven UAVs were synchronized. In addition, it shows that the instability of UAVs is not subject to wireless coverage. First sets of experiments (figure 21) was expected to have high instability rate due to less wireless coverage however, the experiment recorded less rate of UAVs instability than in second sets of experiments.

5.3) Experimental results 3

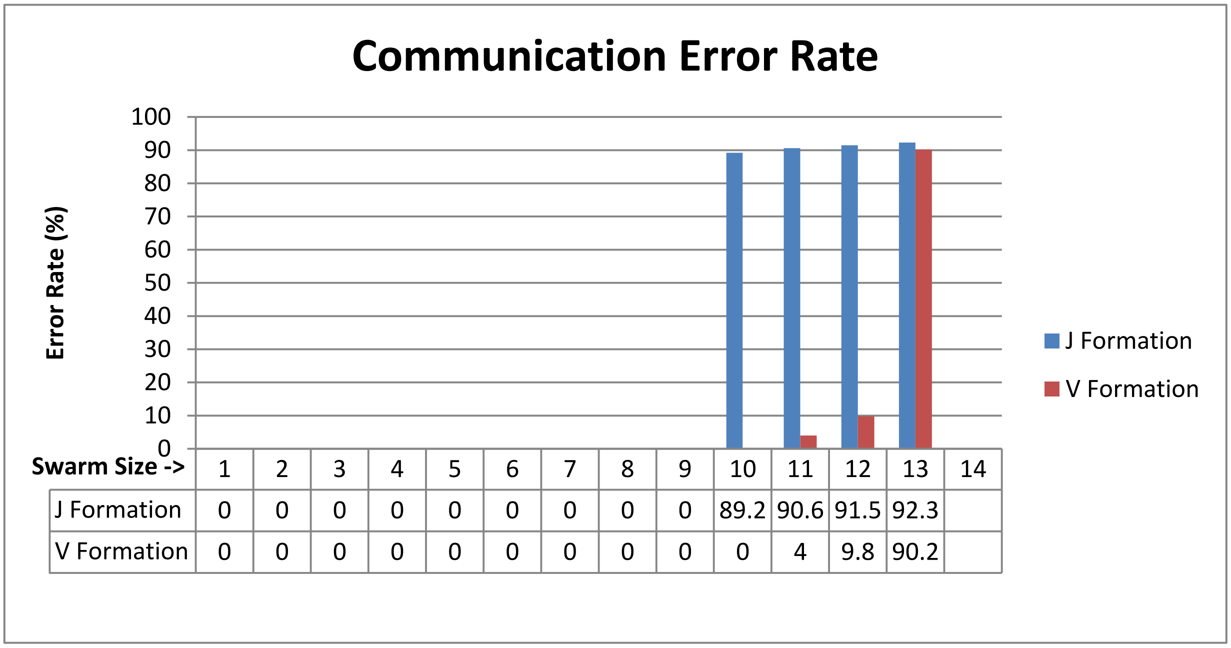
Figure 24 below shows the results of the third experiment using limited wireless transmission (transmission range = 1.5m) and Multi-hop MANET transmission. The focus of this set of experiment was to test the impact of routing in conjunction with wireless transmission range in the performance of the Drone Swarm Synchronization Protocol (UAVSSP).



	1	2	3	4	5	6	7	8	9	10
J instability time(s)	0	0	1.65	1.5	1.9	1.55	2.05	2.05	2.55	
J syncronization time(s)	0	0	2.95	3	3.2	3.05	3.55	3.55	4.2	
V instability time(s)	0	0	1.75	2.5	2.5	2.7	2.8	2.85	3.05	3.1
V syncronization time(s)	0	0	3.2	3.95	3.95	4.2	4.35	4.45	4.7	4.8

Figure 24: Wireless communication range = 1, Routing = multi -hop

By looking at results presented by Figure 20 in the first set of experiment and the results presented in Figure 24 above, it can be seen that routing has an impact in the performance of the UAVSSP. These two sets of results are compared since their experiments where carried using 1.5m wireless communication range. It can be seen that UAVSSP when using multi-hop is able to synchronize many UAVs for both formations (V and J) than when using single hop routing in wireless transmission of 1.5m. As observed in experiment 1 and experiment 2 the synchronization time for this experiment increases with number of UAVs participating in a swarm.



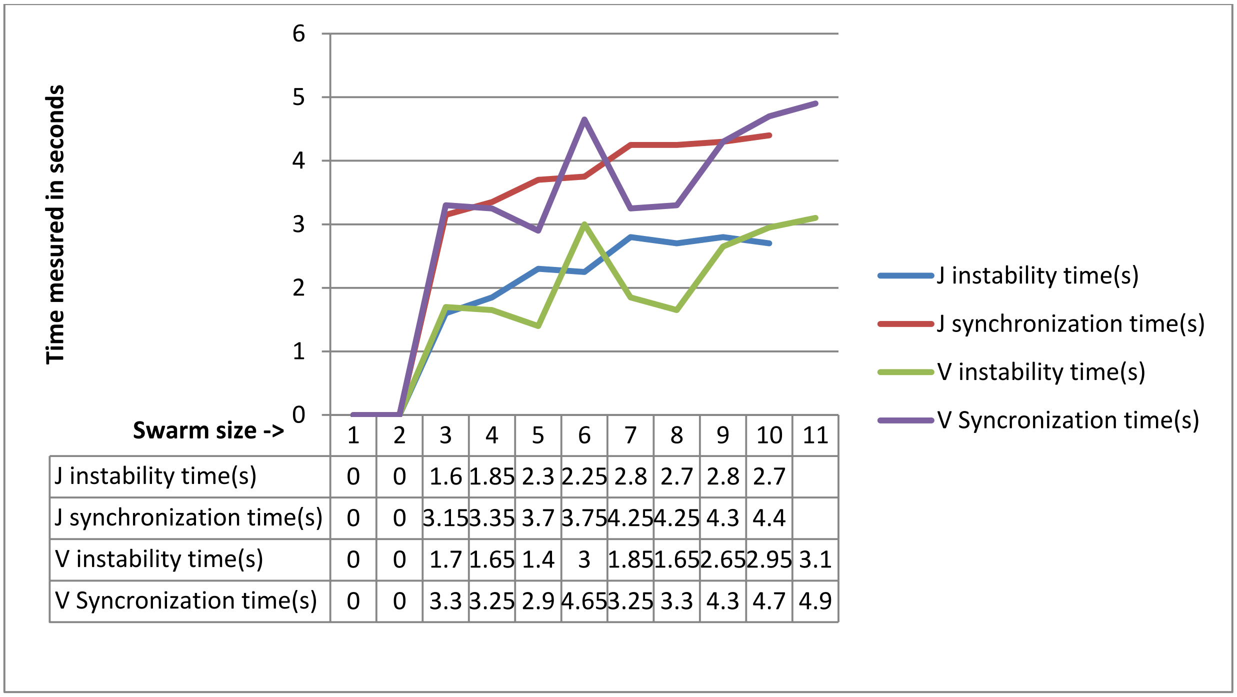
Swarm Size ->	1	2	3	4	5	6	7	8	9	10	11	12	13	14
J Formation	0	0	0	0	0	0	0	0	0	89.2	90.6	91.5	92.3	
V Formation	0	0	0	0	0	0	0	0	0	0	4	9.8	90.2	

Figure 25: Wireless communication range = 1, Routing = multi -hop

Unmanned Aerial Vehicle Swarm Synchronization Protocol (UAVSSP) could not synchronize swarm of ten UAVs in J Formation and swarm of eleven UAVs in V formation due to high communication error rate as shown in Figure 25 above. In J formation the communication error rate recorded when trying to synchronize swarm of ten UAVs is 89.2 % where-else in V formation the communication error when trying to synchronize eleven UAVs is 4 %. According to the observation made, communication error rate has to be below 3.5% for synchronization to take place therefore 4% is high enough to distort synchronization.

5.4) Experimental results 4

These set experiments were carried out to test the performance of Unmanned Aerial Vehicle Swarm synchronization Protocol (UAVSSP) using Multi-hop routing with wireless coverage of 3m. Figure 26 below shows the performance of the UAVSSP during the simulation.



Swarm size ->	1	2	3	4	5	6	7	8	9	10	11
J instability time(s)	0	0	1.6	1.85	2.3	2.25	2.8	2.7	2.8	2.7	
J synchronization time(s)	0	0	3.15	3.35	3.7	3.75	4.25	4.25	4.3	4.4	
V instability time(s)	0	0	1.7	1.65	1.4	3	1.85	1.65	2.65	2.95	3.1
V Syncronization time(s)	0	0	3.3	3.25	2.9	4.65	3.25	3.3	4.3	4.7	4.9

Figure 26: Wireless communication range = 2, Routing = multi -hop

As expected, the results show that the performance of UAVSSP improved after wireless transmission range was increased. Comparing this results with the results in experiment 3 (figure 23) and experiment 4(figure 26), it can be seen that more UAVs where synchronized in figure 26 than in figure 23. It shows that good wireless coverage range improve the speed at which UAVSSP synchronizes the swarm in this set of experiments. The time taken by the UAVSSP to synchronize UAVs in this experiment is less than the time taken in experiment 4.

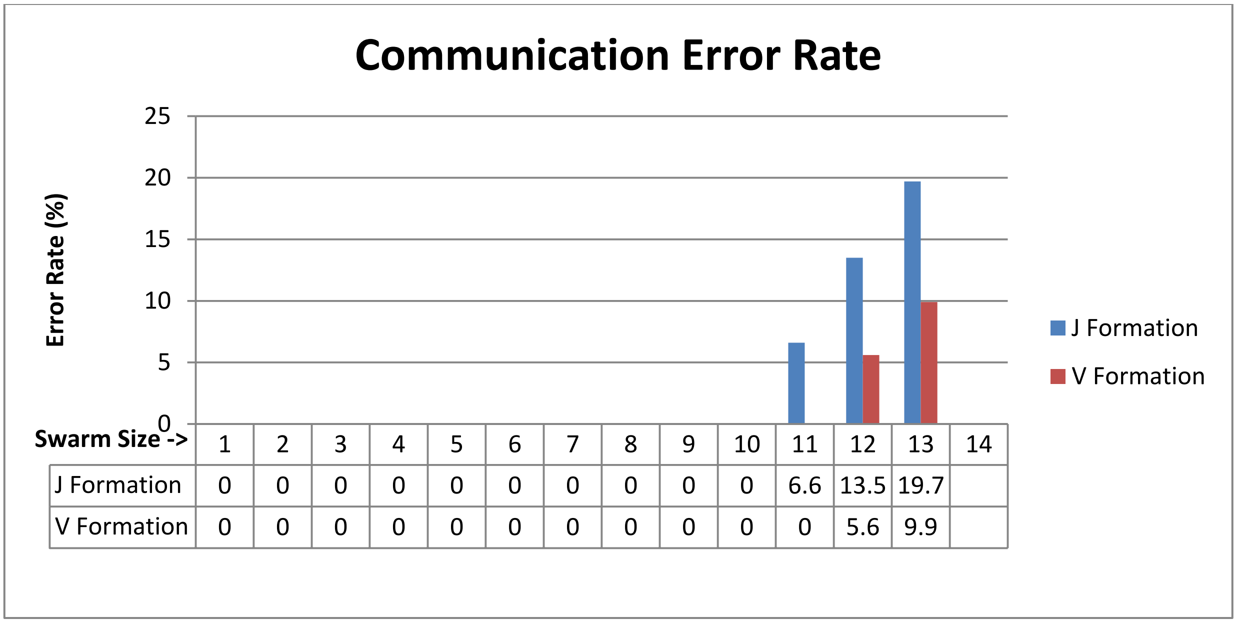
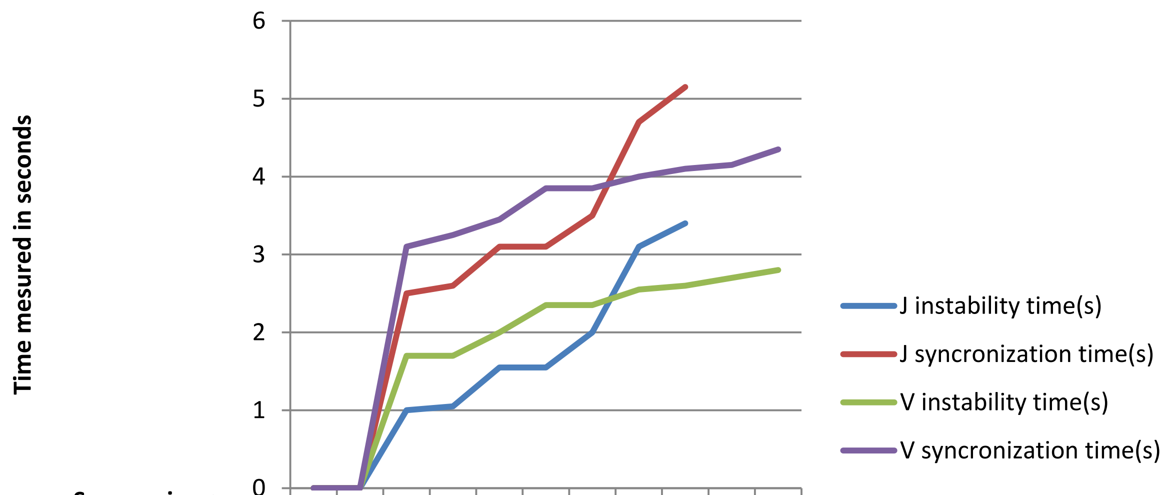


Figure 27: Wireless communication range = 2, Routing = multi -hop

Figure 27 shows that, the implementation of Unmanned Aerial Vehicle Swarm Synchronization Protocol in multi-hop MANET routing resulted in high communication error rate as compared to single-hop MANET routing in figure 23. This is because a packet that was multi-hoped by the time it reaches its destination point it will be out dated. In other words, a UAV that receives the packet will try to position itself based on delayed out dated coordinates due to retransmission and as a result it will sway out of its position in the swarm until it is out of wireless communication coverage range. This indicates that; 1) UAVSSP cannot synchronize large swarms using outdated coordinates.2) packets delay using multi-hop MANET grows exponentially with the number of UAVs in a swarm therefore increasing communication error rate.

5.5) Experimental results 5

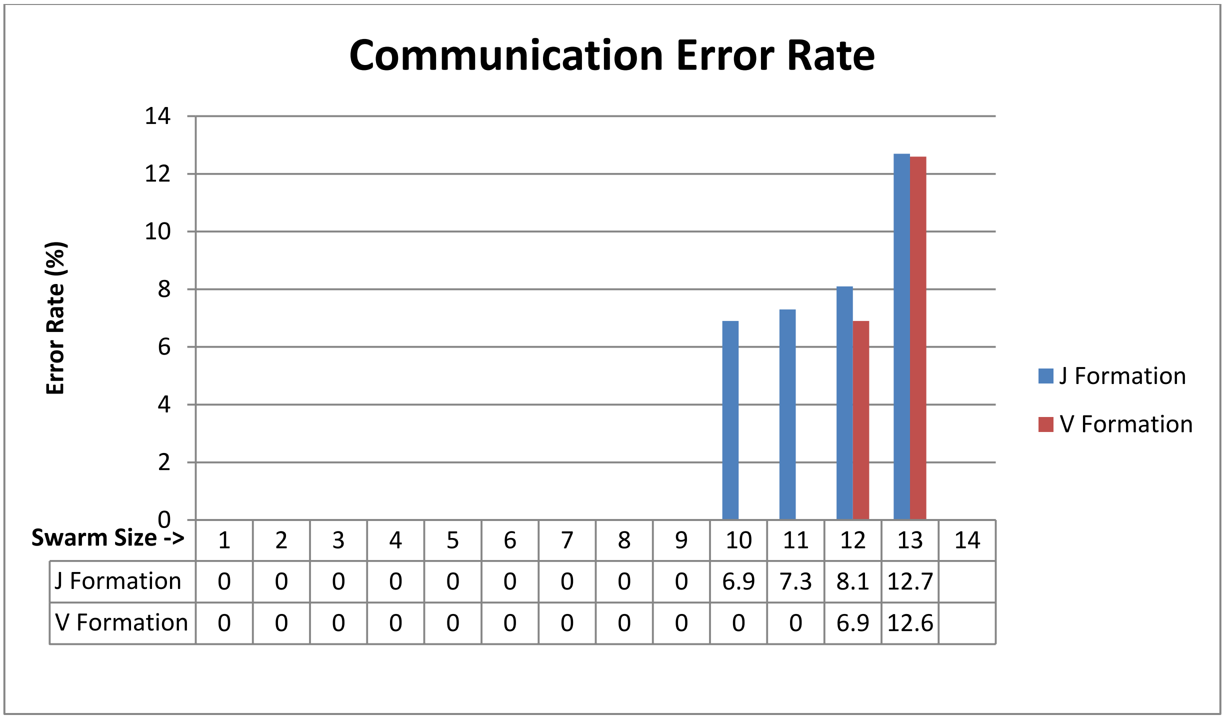
The scenario used in these sets of experiments is different from other experiments presented so far. The intention of executing these sets of experiments was to test whether the topology used in Unmanned Aerial Vehicle Swarm Synchronization Protocol (UAVSSP) does improve the protocol performance. The topology that was contrasted with UAVSSP topology in this case is star topology. The Figure 28 below shows the plot of the simulation results.



Swarm size ->	1	2	3	4	5	6	7	8	9	10	11
J instability time(s)	0	0	1	1.05	1.55	1.55	2	3.1	3.4		
J syncronization time(s)	0	0	2.5	2.6	3.1	3.1	3.5	4.7	5.15		
V instability time(s)	0	0	1.7	1.7	2	2.35	2.35	2.55	2.6	2.7	2.8
V syncronization time(s)	0	0	3.1	3.25	3.45	3.85	3.85	4	4.1	4.15	4.35

Figure 28: topology = star topology

The results presentation in the above figure 28 shows that J formation had low instability rate when synchronizing small swarm of three to six UAVs as compared to V formation. In a swarm made of more than six UAVs, the figure 28 shows that instability rate of J formation raised. This instability rates affected the overall swarm synchronization time. As plotted in figure 28, the V formation synchronization time improved as the number of UAVs in a swarm increases. This implies that the topology used in UAVSSP does contribute positively in the performance of the protocol.



Swarm Size ->	1	2	3	4	5	6	7	8	9	10	11	12	13	14
J Formation	0	0	0	0	0	0	0	0	0	6.9	7.3	8.1	12.7	
V Formation	0	0	0	0	0	0	0	0	0	0	0	6.9	12.6	

Figure 29: topology = star topology

Furthermore, the communication error rate in star topology is high as shown in figure 29 above. As a result, only nine UAVs in J formation and in eleven UAVs in V formation were synchronized. In star topology a leader has to communicate with all UAVs without hoping packets through other UAV because the topology does not support packet hoping. As a result, UAVs loitering not in proximity with a leading UAV experience *out of bounds wireless communication error*.

5.6) Results Presentation Summary

In summary, the simulation results for all experiments from figure 20 to figure 29 shows that;

- **Swarm synchronization cannot take place when communication error rate is above 3.5 %**: The highest error rate that didn't affect synchronization in developed protocol is 3.3 % in figure 21. Furthermore, the results in all experiments demonstrate that the rate of communication error determines the number of UAVs that can be synchronized in a swarm.
- **Synchronization time increases with the number of UAVs in a swarm.** By looking at all plotted results, it can be seen that graphs were rising whenever swarm increases. This is caused by high instability rate that grew when the swarm size was increased. The instability time and synchronization time are dependent. A change in instability time affects synchronization time and vice versa.
- **Good communication coverage improves the performance of the Unmanned Aerial Vehicle Swarm Synchronization Protocol (UAVSSP).** In the series of test performed, it shows that all experiments carried on 1.5m wireless transmission did not synchronize many UAVs as compared to experiments carried in 3m wireless transmission. Moreover, the speed at which the UAVSSP was synchronizing the UAV swarm in 3m wireless communication range surpassed the speed of synchronization when the wireless coverage was 1.5m.

Furthermore, experimental results has demonstrated that multi-hop MANET routing delays packet delivery therefore, single-hope MANET routing is preferred over multi-hop MANET in the protocol we developed.

6.0) Experimental Results Discussion

The performance of Unmanned Aerial Vehicle Swarm Synchronization Protocol (UAVSSP) was tested by considering three factors which are; 1) the cost of wireless transmission range on the synchronization time of UAVs in different shapes using UAVSSP, 2) the cost of networking topology on the synchronization time of UAVs using UAVSSP and 3) communication error rate using UAVSSP. The performance results of the protocol are plotted in sub-section 5.2 whereby maximum of thirteen (13) UAVs where used as the test bed of experiment.

According to the observation made, synchronization of all thirteen UAVs using UAVSSP was achieved in figure 22. The set of experiment in figure 22 was implemented using single-hop MANET architecture and the wireless transmission range of 3m. The total time taken to synchronize swarm of thirteen UAVs in the experiment is 11.6s. In other experiments, synchronization of thirteen (13) UAVs was not possible because of high communication error rate. The swarm synchronization using the developed protocol (UAVSSP) is possible only when communication error rate is below 3.5%. These findings align with literature in [30] and [15].It shows that UAVSSP is taking too much time and in some instances fail to synchronize UAV swarm when UAV intercommunication is poor. It has to be noted that UAVSSP was only tested using simulation (V-Rep) therefore; findings gathered might not reflect the performance of UAVSSP in actual outdoor environments. As an example, when testing UAVSSP in actual outdoor environments using TCP/IP, the communication error rate is expected to diminish therefore increasing the performance of UAVSSP. TCP/IP can help on reducing communication error rate by retransmission of lost packets.

The UAVSSP when using single-hop can synchronize more than thirteen UAV in V formation. However, increasing the number of UAVs rises the time it takes for UAVSSP to synchronize a UAV swarm. This can be improved by using Multi-Layer UAV ad-hoc network architecture [37]. Multi-Layer UAV ad-hoc network architecture allows large swarm to be partitioned in small subgroups. This architecture was used in [29] to synchronize swarm of twenty UAVs.

The communication latency recorded in all set of experiments is 0.2s $\pm$ 0.1s for a radius of 1m – 6m using unicast transmission mode. Even though this recorded communication latency is higher as compared to 0.4s - 0.2s for a radius of 1m – 100m in [30], the results shows that UAVSSP was not highly affected by the latency.

7.0) Conclusions

This thesis presents a simple distributed outdoor flying UAV swarm synchronizing protocol (UAVSSP) developed based on Vicksec self-propelled particle (SPP) models. V-Rep simulator was used to simulate and test the behavior of UAVs under influence of the protocol we developed. The simulation results gathered after testing the performance of UAVSSP demonstrate that communication plays a vital role in UAV swarm synchronization. When UAV inter-communication is reliable, UAV swarm synchronization can be achieved using simple control protocol. Furthermore, the results demonstrate that the implementation of collision avoidance is not necessary when UAV intercommunication is reliable. The collision avoidance is not implemented in the UAVSSP yet no UAV collision was recorded when communication error rate was below 3.5% even though UAVs were floating in proximity of less than 3m. The effectiveness of UAV intercommunication is determined by networking architecture and cast (unicasting, multicasting and broadcasting) method used.

Adding on above mentioned observations, a developed protocol recorded high performance when using single-hop MANET routing in V formation. Alteration of casting (unicasting, multicasting and broadcasting) method does not affect UAVSSP performance. However, the implementation of MANET using broadcasting comes with disadvantage of flooding.

Another point learned concerns the implementation of *feed-forward ability* sometimes referred to as *feed-forward terms* in the control protocol and algorithms. *Feed-forward terms* refer to the ability of a control protocol to predict the position of a UAV in next time step. It shows that *feed-forward terms* can improve the speed at which UAV swarm synchronizes at. Moreover, *feed-forward terms* are needed in outdoor environments where localization system (GPS) does not have high precision rate that can be trusted in UAV state estimations. The incorporation of feed-forward terms in the developed protocol has helped when UAV swarm was experiencing high instability rate and in cases whereby UAV to UAV communication was not established.

8.0) Future work

According to [6], the development of algorithms and protocols for synchronizing swarm of UAVs for outdoor test benches is at infancy and far to be used in practical application. This is because most of developed control protocols and algorithms for outdoor environments were simulated rather than tested in actual outdoor environment. Simulators are very important when testing protocols because they allow testing of extreme cases without damaging actual UAV. However, some of variables like wind, temperature, and climate are difficult to be simulated. Therefore, to have protocols that can be trusted in real practical outdoor application, protocols needs to be tested in actual environments after simulation.

Future research must also focus on swarm synchronization protocols that promote cooperation amongst agents than obedience. In other word, agents (UAVs) within artificial swarm will only obey their leader if required to, otherwise agents (UAVs) within a swarm must have freedom to decide not obey a leader when a leader is jeopardizing a mission. As an example, a leading UAV can jeopardize a mission when it has fault with sensors and actuators. For agents (UAVs) to be successful on this regard, they must be able to study the movement patterns of their leader. These will help them to know when a leader is failing due to some defects.

Furthermore, future research should focus more in fault tolerant formation based outdoor UAV swarm synchronization protocol development. Fault-tolerant protocol will allow UAV swarm to complete a mission in desired shape even if fault has occurred. Fault can occur as a result of sensor and actuator malfunction or out of range communication error. In this case UAVs should make proper adjustments and continue with the mission.

One more important element of a realistic swarm synchronization protocol that can be used in the envisioned out door UAV swarm application is protocol simplicity. Simple protocols do not overstrain battery resources therefore increase UAV loitering time. This task can be archived easily by mimicking nature swarms when developing protocols for synchronizing swarm in outdoor environment. Protocols that were developed by mimicking nature swarm allow UAV to interact together using simple rules. This assertion was validated in the experimental results presented in this thesis. The results demonstrated that a simple developed protocol based on SPP models was able to synchronize UAVs in V and J formation even in presence of communication error below 3.5%.

Bibliography

[1] James Carroll. (2015, October) The future is here:Five applications of UAV technology. [Online]. http://www.vision-systems.com/articles/2013/12/the-future-is-here-five-applications-of-uav-technology.html

[2] Henri Eisenbeiss, "A mini unmanned aerial vehicle(UAV): system overview and image aguisition ," *International Archives of Photogrammetry.Remote Sensing and Spatial Information Science*, Nonemver 2004.

[3] Jessie YC Chen, Allen C Haas, and Michael J Barnes, "Human Perfomance Issues and User Interface Design for Teleoperated Robots," *IEEE Transaction on Systems, Man, and Cybernnetics, Part C(Application and Reviews)*, vol. 37, no. 6, pp. 1231-45, November 2007.

[4] Axel Burkle, Florian Segor, and Matthias Kollmann, "Towards Autonomous Micro UAV Swarms ," *Journal of intelligent & robotic systems*, vol. 61, no. 1-4, pp. 339 - 353, January 2011.

[5] Vicsek Tamás and Zafeiris Anna, "Collectivemotion," *Physics Reports*, vol. 517, no. 3, pp. 71-140, August 2012.

[6] Ying Tan and Zheng Zhong-yang, "Research advance in swam robotics," *Defence Technology*, vol. 1, no. 9, pp. 18-19, 2013.

[7] Hector Garcia-Molina, "Elections in a Distributed Computing System," *IEEE transactions on Computers*, vol. 100, no. 1, pp. 48-49, January 1982.

[8] Bijan Ranjbar-Sahraei, Gerhard Weiss, and Nakisaee Ali, "A multi-robot coverage approach based on stigmergic communication," in *German Conference on Multiagent System Technologies*, Springer Berlin Heildelberg, 2012, pp. 126-138.

[9] Andrew Zulu and Samuel John, "A review of control algorithms for autonomous quadrotors," *Open Journal of Applied Sciences*, February 2016.

[10] Levent Bayindir and Sahin Erol, "A review of studies in swarm robotics," *Turkish Journal of Electrical Engineering & Computer Science*, vol. 15, no. 2, pp. 115-147, 2007.

[11] Kristina Lerman, Alcherio Martinoli, and Aram Galstyan, "A review of probalistic macroscopic models for swarm robotics systems," *International Workshop on Swarm*

Robotics, pp. 143-152, July 2004.

[12] Yuhui Shi and C. Eberhart Russell, "Emphirical study of particle swarm optimization," in *In Evolutionary Computation*, vol. 3, 1999.

[13] Aleksandar Jevtić and Andina de la Fuente Diego , "Swarm intelligence and its applications in swarm robotics," 2007.

[14] Khaldi Belkacem and Cherif Foudil , "An Overview of Swarm Robotics: Swarm Intelligence Applied to Multi-robotics," *International Journal of Computer Applications*, vol. 126, no. 2, pp. 0975 – 8887, September 2015.

[15] Sabine Hauert et al., "Reynolds flocking in reality with fixed-wing robots: communication range vs. maximum turning rate," in *In 2011 IEEE/RSJ International Conference on Intelligent Robot and Systems*, 2011, pp. 5015-5020.

[16] Craig Reynolds. (2016, September) Boids. [Online]. http://www.red3d.com/cwr/boids/

[17] Vicsek Tamas, Czirok Andras, Ben-Jacob Eshel, Cohen Inon, and Shochet Ofer, "A novel type of phase transition in a system of Self driven-particles," *Physical review letters*, vol. 76, no. 6, August 1995.

[18] B. Atkinson. (2015, October) Agricultural Drone Projects on the Horizon. [Online]. http://www.pmi.org/Learning/PM-Network/2014/agricultural-drone-projects-on-horizon.aspx

[19] C.Dillow. (2015, October) FAA's relaxed drone rules could mean big changes for industry. [Online]. http://fortune.com/2015/05/08/faa-drone-rules

[20] P.Hollins. (2014, August) FAA drone ruling said to be a setback for farmers, research. [Online]. http://southeastfarmpress.com/equipment/faa-drone-ruling-said-be-setback-farmers-research?page=2

[21] Sparkfun. (2016, September) What is an Arduino?. [Online]. https://learn.sparkfun.com/tutorials/what-is-an-arduino

[22] Arduino. (2016, September) Project hub. [Online]. https://create.arduino.cc/projecthub/team-omega-d2/ar-drone-and-1sheeld-discover-the-world-around-your-drone-de6a16?ref=search&ref_id=drones&offset=0

[23] Arduno. (2016, September) Project Hub. [Online]. https://create.arduino.cc/projecthub/gius_8/weather-station-for-drones-

1184cd?ref=search&ref_id=Weather Station for Drones &offset=0

[24] Arduno. (2016, September) Project Hub. [Online]. https://create.arduino.cc/projecthub/giando/control-a-parrot-ar-drone-with-linino-linino-1763e1?ref=search&ref_id=Control a Parrot AR Drone with Linino projects&offset=0

[25] The Linux Foundation. (2016, January) Dronecode Project Advances Unmanned Aerial Vehicles for Commercial Applications. [Online]. https://www.linuxfoundation.org/news-media/announcements/2016/01/dronecode-project-advances-unmanned-aerial-vehicles-commercial

[26] DroneCode. (2016, September) Linux collaborative projects. [Online]. https://www.dronecode.org/about

[27] de Lanerolle Trishan. (2015, July) Open source.com descover an open source world. [Online]. https://opensource.com/life/15/7/the-dronecode-foundation

[28] DroneCode. (2016, September) Supported Simulation Environments. [Online]. https://www.dronecode.org/software/supported-simulation-environments

[29] Alex Kushleyev, Mellinger Daniel , Powers Caitlin , and Kumar Vijay , "Towards a swarm of agile micro quadrotors," *Autonomous Robots*, vol. 35, no. 4, pp. 287-300, 2013.

[30] G Vasarhelyi et al., "Outdoor flocking and formation flight with automous Aerial robots," in *In 2014/RSJ International Conference on Interlligent Robots and Systems*, 2014, pp. 3866-3873.

[31] Deepika Rani Kampally, Annae Rashmi, and Kumar Santhosh, "Analysis of Algorithms to implement Swarm Bots for Effective Swarm Robotics,".

[32] Vásárhelyi Gábor et al. (2014) Outdoor flocking and formation flight with autonomous aerial robots. [Online]. https://www.google.co.bw/search?q=Outdoor+flocking+and+formation+flight+with+autonomous+aerial+robots*&biw=1024&bih=661&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjw5rCQ08PPAhUBI8AKHWaWDcQQ_AUIBygC&dpr=1#imgrc=WMvRyUez-1I04M%3A

[33] Ernst Jason. (2015, September) Robotics Trends. [Online]. http://www.roboticstrends.com/article/4_challenges_holding_back_swarm_robotics

[34] Zhigang Wang, MengChu Zhou, and Nirwan Ansari, "Ad-hoc Robot Wireless Communication," in *In Systems,Man and Cybernetics*, 2003, pp. 4045 - 4050.

[35] Eric W Frew and Timothy X Brown, "Airborne Communication Network for Small Unmanned System," *Proceedings of IEEE*, vol. 96, no. 12, pp. 2008 - 2027, December 2008.

[36] Brown X Timothy and Frew W Eric, "Networking Issues for Small Unmanned Aircraft Systems," *Journal of Intelligent and Robotic Systems*, vol. 54, no. 1-3, pp. 21 - 37, March 2009.

[37] Jun Li, Yifeng Zhou, and L Lamont, "Communication architectures and protocols for networking unmanned aerial vehicle," in *IEEE Globecom Workshops(GC Wkshps)*, 2013, pp. 1415-1420.

[38] Enrico Del Re, Fantacci Romano, and Giambene Giovanni , "Handover and dynamic channel allocation techniques in mobile cellular networks," *IEEE Transactions on Vehicular Technology*, vol. 44, no. 2, pp. 229-237, May 1995.

[39] Wzorek, Mariusz, Landen David, and Doherty Patrick, "GSM Technology as a Communication Media for an Automous Unmanned Aerial Vehicle," in *Proceedings of the 21st Bristol international conference on UAV systems*, 2006.

[40] adafruit. (2013, August) adafruit. [Online]. https://blog.adafruit.com/2013/08/23/sky-drone-fpv-uses-3g4g-cell-network-to-provide long range RC

[41] RCGroups.com. (2016, August) RCGroups.com. [Online]. http://www.rcgroups.com/forums/showthread.php?t=1979096

[42] A Rafay. (2012, July) How the Predator UAV Works. [Online]. http://defence.pk/threads/how-the-predator-uav-works.198822/

[43] Clamtac Imrich, Macro, and Liu J Jennifer, "Mobile ad hoc networking: imperatives and challenges," *Elsevier*, vol. 1, pp. 13-64, 2003.

[44] Bilstrup Katrin, "A Preliminary Study of Wireless Body Area Networks," Halmstad University, Halmstad, Sweden, Technical Report IDE0854, 2008.

[45] Gabriel M Hoffman, Haomiao Huang, Steven L Waslander, and Claire J Tomlin, "Precioin flight control for a multi-vehicle quadrotor helicopter testbed," *Control engineering practice*, vol. 19, no. 9, pp. 1023-1036, September 2011.

[46] Segor Florian, Kollmann Matthias, and Axel Bürkle, "Towards autonomous micro uav swarms," *Journal of intelligent & robotic systems 61*, vol. 1-4, pp. 339-353, 2011.

[47] Quintero A. P Steven, Collins E Gaemus, and Hespanha P Joao, "Flocking with Fixed-Wing UAVs for Distributed Sensing: A Stochastic Optimal Control Approach," in *In 2013 American Control Conference*, Jun 2013, pp. 2025-2031.

[48] Ali Emre Turgut, Fatih Gökçe, Hande Çelikkanat, and Erol Ahin, "Self-organized flocking in mobile robot swarms," *Swarm Intelligence*, vol. 2, no. 2, pp. 97–120, December 2008.

[49] Don Miner, "Swarm robotics algorithm: A survey," Univ. Maryland, College Park, Tech May 2007.

[50] Chlamtac Imrich, Conti Marco, and J-N Liu Jennifer, "Mobile ad hoc networking: imperatives and challenges," *Ad hoc networks*, vol. 1, no. 1, pp. 13-64, 2003.

[51] M T Hyland, B E Mullins, R O Baldwin, and M A Temple, "Simulation-based Performance Evaluation of Mobile Ad Hoc Routing Protocols in a Swarm of Unmanned Aerial Vehicle," *Advanced Information Networking and Applications Workshops*, vol. 2, pp. 249-256, May 2007.

[52] Stefano Michieletto, Davide Zanin, and Emanuele Menegatti, "Nao Robot Simulation for Service Robotic Purposes," *European Modelling Symposium(EMS)*, pp. 477-482, November 2013.

[53] Alba Fenucci, Marina Indri, and Fabrizio Romanelli, "A real time distributed approach to collision avoidance for industrial manipulators," in *IEEE Emerging Technology and Factory Automation(ETFA)*, 2014, pp. 1-8.

[54] Enrico Mingo Hoffman et al., "Yarp Based Plugins for Gazebo Simulator," *In International Workshop on Modelling and Simulation for Autonomous*, pp. 333-346, May 2014.

[55] Daniel Cook, Andrew Vardy, and Ron Lewis, "A Survey of UAV and Robot Simulators for Multi-Vehicle Operations," *IEEE/OES Autonomous Underwater Vehicle(AUV)*, pp. 1-8, October 2012.

[56] Joao O Barros, Vitor Santos, and Filipe Silva, "Bimanual Haptics for Humanoid Robot Teleoperation using ROS and V-REP," *In Autonomous Robot Systems and Competitions(ICARSC)*, pp. 174-179, Aprial 2015.

[57] Eric Rohmer, Surya PN Singh, and Marc Freese, "V-REP: a Versatile and Scalable Robot Simulation Framework," in *IEEE/RSJ international Conference on intelligent Robots and Systems(IROS)*, Tokyo, 2013, pp. 3-7.

[58] Erik Kuiper and Nadjm-Tehrani Simin, "Mobility Models for UAV Group Reconnaissance Application," *2006 International Conference on Wireless and Mobile Communication(ICWMC'06)*, pp. 33-33, July 2006.

[59] Martin Barksten and David Rydbeg, "Extending Reynolds' flocking model to a simulation of sheep in the precence of a predator," School of Computer Scince and Communication , 2013.

[60] Arduino. (2016, September) What is Arduino? [Online]. https://www.arduino.cc/en/Guide/Introduction

[61] Vásárhelyi et al., "Outdoor flocking and formation flight with autonomous aerial robots," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 3866-3873.

Appendices 1

```
--read its present position
targetPos=simGetObjectPosition(targetObj,-1)
position = simGetObjectPosition(d,-1)
me_X = position[1]
me_Y = position[2]
me_Z = position[3]

--initialize safe landing
final_x0 = me_X
final_y0 = me_Y
final_z0 = me_Z

simSetBoolParameter(sim_boolparam_force_show_wireless_emission,false)
simSendData(sim_handle_all,me_ID,"signal",me_ID,sim_handle_self,1,3.1415,6.283)
simSetBoolParameter(sim_boolparam_force_show_wireless_emission,false)

simSetBoolParameter(sim_boolparam_force_show_wireless_reception,false)
signal = simReceiveData("signal",sim_handle_self)
simSetBoolParameter(sim_boolparam_force_show_wireless_reception,false)

if not signal then
   rank = "leader"
```

```
   while now() < 5 do
        simSetBoolParameter(sim_boolparam_force_show_wireless_reception,false)
        signal = simReceiveData("ID",sim_handle_self)
        simSetBoolParameter(sim_boolparam_force_show_wireless_reception,false)

        if not leaderID then
            wing = ""
        else
            wing = signal
            simSetBoolParameter(sim_boolparam_force_show_wireless_emission,false)
simSendData(sim_handle_all,me_ID,"xCoordinate",me_X_position,sim_handle_self,1,3.1415,6.283)
simSendData(sim_handle_all,me_ID,"yCoordinate",me_Y_position,sim_handle_self,1,3.1415,6.283)
simSendData(sim_handle_all,me_ID,"zCoordinate",me_Z_position,sim_handle_self,1,3.1415,6.283)
simSendData(sim_handle_all,me_ID,"rotCorr",rotCorr,sim_handle_self,1,3.1415,6.283)
simSetBoolParameter(sim_boolparam_force_show_wireless_emission,false)
 end
 end
    else
        --Receive leader's ID ----
        simSetBoolParameter(sim_boolparam_force_show_wireless_reception,false)
        leaderID = simReceiveData("ID",sim_handle_self)
simSetBoolParameter(sim_boolparam_force_show_wireless_reception,false)
        --Send Acknowledgement --
simSetBoolParameter(sim_boolparam_force_show_wireless_emission,false)
```

```
simSendData(sim_handle_all,me_ID,"Acknowledgement",me_ID,sim_handle_self,1,3.1415,6.283)
simSetBoolParameter(sim_boolparam_force_show_wireless_emission,false)
--Transmit coordinates
simSetBoolParameter(sim_boolparam_force_show_wireless_reception,false)
leaderX = simReceiveData(leaderID ,"xCoordinate",sim_handle_self)
leaderY = simReceiveData(leaderID ,"yCoordinate",sim_handle_self)
leaderZ = simReceiveData(leaderID ,"zCoordinate",sim_handle_self)
rotCorr = simReceiveData(leaderID ,"rotCorr",sim_handle_self)
simSetBoolParameter(sim_boolparam_force_show_wireless_reception,false)

--Stores the coordinates of its leader
leader_X = leaderX
leader_Y = leaderY
leader_Z = leaderZ
fVelocity = rotCorr

-- Request for its follower
simSetBoolParameter(sim_boolparam_force_show_wireless_emission,false)
simSendData(sim_handle_all,me_ID,"signal",me_ID,sim_handle_self,1,3.1415,6.283)
simSetBoolParameter(sim_boolparam_force_show_wireless_emission,false)

simSetBoolParameter(sim_boolparam_force_show_wireless_reception,false)
signal = simReceiveData("signal",sim_handle_self)
simSetBoolParameter(sim_boolparam_force_show_wireless_reception,false)
```

```
        --If no response is given, the request is retransmited for 5 seconds then a
        --will assume wing rank otherwise a wingHybrid rank
        if not signal then
            while now() < 6 do
                simSetBoolParameter(sim_boolparam_force_show_wireless_reception,false)
                signal = simReceiveData("ID",sim_handle_self)
                simSetBoolParameter(sim_boolparam_force_show_wireless_reception,false)

                if not leaderID then
                    rank = "wing"
                else
                    wing = signal
                    rank = "wingHybrid"
                    simSetBoolParameter(sim_boolparam_force_show_wireless_emission,false)

simSendData(sim_handle_all,me_ID,"xCoordinate",me_X_position,sim_handle_self,1,3.1415,6.283)

simSendData(sim_handle_all,me_ID,"yCoordinate",me_Y_position,sim_handle_self,1,3.1415,6.283)

simSendData(sim_handle_all,me_ID,"zCoordinate",me_Z_position,sim_handle_self,1,3.1415,6.283)

simSendData(sim_handle_all,me_ID,"rotCorr",rotCorr,sim_handle_self,1,3.1415,6.283)
                    simSetBoolParameter(sim_boolparam_force_show_wireless_emission,false)
```

```
                end

            end
        else
            wing = signal
            rank = "wingHybrid"
            simSetBoolParameter(sim_boolparam_force_show_wireless_emission,false)

simSendData(sim_handle_all,me_ID,"xCoordinate",me_X_position,sim_handle_self,1,3.1415,6.283)

simSendData(sim_handle_all,me_ID,"yCoordinate",me_Y_position,sim_handle_self,1,3.1415,6.283)

simSendData(sim_handle_all,me_ID,"zCoordinate",me_Z_position,sim_handle_self,1,3.1415,6.283)
            simSendData(sim_handle_all,me_ID,"rotCorr",rotCorr,sim_handle_self,1,3.1415,6.283)
            simSetBoolParameter(sim_boolparam_force_show_wireless_emission,false)
    end
```

Appendices 2

```
if( rank == "leader") then
        simSetBoolParameter(sim_boolparam_force_show_wireless_emission,false)

simSendData(sim_handle_all,me_ID,"xCoordinate",me_X_position,sim_handle_self,1,3.1415,6.283)

simSendData(sim_handle_all,me_ID,"yCoordinate",me_Y_position,sim_handle_self,1,3.1415,6.283)

simSendData(sim_handle_all,me_ID,"zCoordinate",me_Z_position,sim_handle_self,1,3.1415,6.283)
        simSendData(sim_handle_all,me_ID,"rotCorr",rotCorr,sim_handle_self,1,3.1415,6.283)
        simSetBoolParameter(sim_boolparam_force_show_wireless_emission,false)
    end

    if(rank == "wingHybrid")then
        --Getting its own position
        targetPos=simGetObjectPosition(targetObj,-1)
        position = simGetObjectPosition(d,-1)
        me_X_position = position[1]
        me_Y_position = position[2]
        me_Z_position = position[3]

        simSetBoolParameter(sim_boolparam_force_show_wireless_reception,false)
        leaderX = simReceiveData(leaderID,"xCoordinate",sim_handle_self)
```

```
    leaderY = simReceiveData(leaderID,"yCoordinate",sim_handle_self)
    leaderZ = simReceiveData(leaderID,"zCoordinate",sim_handle_self)
    velocity = simReceiveData(leaderID,"rotCorr",sim_handle_self)
    simSetBoolParameter(sim_boolparam_force_show_wireless_reception,false)

    if not leaderX then
        --Try to read coordinates from a leading UAV
        while now() < 5 do
            simSetBoolParameter(sim_boolparam_force_show_wireless_reception,false)
            leaderX = simReceiveData(leaderID,"xCoordinate",sim_handle_self)
            leaderY = simReceiveData(leaderID,"yCoordinate",sim_handle_self)
            leaderZ = simReceiveData(leaderID,"zCoordinate",sim_handle_self)
            velocity = simReceiveData(leaderID,"rotCorr",sim_handle_self)
            simSetBoolParameter(sim_boolparam_force_show_wireless_reception,false)

            --if coordinates have been received before 5 seconds brake outside while loop
            if (leaderX) then
                break
            end
        end

        if not leaderX then
            --Initializing targetObj1 table for safe landing
            targetObj1[1] = final_x0
```

```
            targetObj1[2] = final_x0
            targetPos[3]  = final_x0
        else
            leader_x2 = leaderX
            leader_y2 = leaderY
            leader_z2 = leaderZ
            fVelocity = rotCorr

            xC,yC = velocity(leader_x, leader_x2, leader_y, leader_y2, fVelocity)

            value = Cij(xC,yC,leader_x,leader_y)
            --Checks if the speed of a leading UAV is falling behind
            if (value < 0) then
                targetObj1[1] = leaderX
                targetObj1[2] = leaderY
                targetPos[3]  = leaderZ
                fVelocity     = velocity
                rotCorr       = fVelocity/2  -- reduces its velocity
            else
                -- Initializes targetObj table by immaginary coordinates
                xP,yP = position(xC,yC,me_x, me_y)
                targetObj1[1] = xP
                targetObj1[2] = yP
                targetPos[3]  = leaderZ
```

```
                    -- Transmits its coordinates to its wing UAV
                    simSetBoolParameter(sim_boolparam_force_show_wireless_emission,false)

simSendData(sim_handle_all,me_ID,"xCoordinate",me_X_position,sim_handle_self,1,3.1415,6.283)

simSendData(sim_handle_all,me_ID,"yCoordinate",me_Y_position,sim_handle_self,1,3.1415,6.283)

simSendData(sim_handle_all,me_ID,"zCoordinate",me_Z_position,sim_handle_self,1,3.1415,6.283)

simSendData(sim_handle_all,me_ID,"rotCorr",rotCorr,sim_handle_self,1,3.1415,6.283)
                    simSetBoolParameter(sim_boolparam_force_show_wireless_emission,false)
                    --re-initialize old coordinates with current coordinates.
                    leader_x = leader_x2
                    leader_y = leader_y2
                    leader_z = leader_z2

                end
            end


        end


    if( rank == "follower") then
```

```
--Getting its own position
targetPos=simGetObjectPosition(targetObj,-1)
position = simGetObjectPosition(d,-1)
me_X_position = position[1]
me_Y_position = position[2]
me_Z_position = position[3]
--receiving coordinates from a leader
simSetBoolParameter(sim_boolparam_force_show_wireless_reception,false)
leaderX = simReceiveData(leaderID,"xCoordinate",sim_handle_self)
leaderY = simReceiveData(leaderID,"yCoordinate",sim_handle_self)
leaderZ = simReceiveData(leaderID,"zCoordinate",sim_handle_self)
velocity = simReceiveData(leaderID,"rotCorr",sim_handle_self)
simSetBoolParameter(sim_boolparam_force_show_wireless_reception,false)
--Try to read coordinates from a leading UAV
if not leaderX then
    while now() < 5 do
        simSetBoolParameter(sim_boolparam_force_show_wireless_reception,false)
        leaderX = simReceiveData(leaderID,"xCoordinate",sim_handle_self)
        leaderY = simReceiveData(leaderID,"yCoordinate",sim_handle_self)
        leaderZ = simReceiveData(leaderID,"zCoordinate",sim_handle_self)
        velocity = simReceiveData(leaderID,"rotCorr",sim_handle_self)
        simSetBoolParameter(sim_boolparam_force_show_wireless_reception,false)
         --if coordinates have been received before 5 seconds brake outside while loop
        if (leaderX) then
```

```
                break
            end
        end

        if not leaderX then
            --Initializing targetObj1 table for safe landing
            targetObj1[1] = final_x0
            targetObj2[2] = final_x0
            targetPos[3]  = final_x0
        else
            -- Initializes targetObj table by immaginary coordinates
            leader_x2 = leaderX
            leader_y2 = leaderY
            leader_z2 = leaderZ
            fVelocity = rotCorr

            xC,yC = velocity(leader_x, leader_x2, leader_y, leader_y2, fVelocity)

            value = Cij(xC,yC,leader_x,leader_y)
            --Checks if the speed of a leading UAV is falling behind
            if (value < 0) then
                targetObj1[1] = leaderX
                targetObj1[2] = leaderY
                targetPos[3]  = leaderZ
```

```
                fVelocity      = velocity
                rotCorr        = fVelocity/2
            else
                xP,yP = position(xC,yC,me_x, me_y)
                targetObj1[1] = xP
                targetObj2[2] = yP
                targetPos[3]  = leaderZ
                 --re-initialize old coordinates with current coordinates.
                leader_x = leader_x2
                leader_y = leader_y2
                leader_z = leader_z2

            end
        end


    end
end
```