Autonomous Air Vehicle Flight Formation Coordination Using Onboard Real-Time Control

by

James Sease

B.Eng. University of Victoria, 2019

A Thesis Submitted in Partial Fulfillment of the

Requirements for the Degree of

MASTER OF APPLIED SCIENCE

in the Department of Mechanical Engineering

We acknowledge and respect the lək̓ʷəŋən peoples on whose traditional territory the university stands and the Songhees, Esquimalt and W̱SÁNEĆ peoples whose historical relationships with the land continue to this day

Autonomous Air Vehicle Flight Formation Coordination Using Onboard Real-Time Control

by

James Sease

B.Eng. University of Victoria, 2019

Supervisory Committee

Dr. Afzal Suleman, Supervisor

Department of Mechanical Engineering

Dr. Yang Shi, Departmental Member

Department of Mechanical Engineering

Abstract

The continuing development of Unmanned Aerial Vehicles (UAVs) and other low cost aircraft increases the use of the skies for defense, commercial, and recreational use. Aircraft can gain benefits through cooperation. They must also interact to operate safely. Aircraft are being developed for Urban Air Mobility, package delivery, and mapping, among other uses. Flying in a formation has a number of potential benefits. It can allow aircraft to combine sensor information, fly relatively close together taking less room in an airspace, and it can allow for improved fuel efficiency.

The primary focus of this research is to coordinate a group of UAVs in a controlled formation, and to allow the aircraft to respond to each other without an external central command center directing their trajectory. The flight formation control system runs a coordination program onboard one of the aircraft, this program creates a virtual leader which all other aircraft in the formation follow. Each aircraft has a preset offset from the leader based on the formation, which provides their target point in the formation. A limited form of flocking, simulating the behavior of a flock of birds, was explored to allow the aircraft to interact directly, ensuring safe spacing around the aircraft.

A development platform was also created consisting of a simulation environment and a set of three aircraft. The simulation environment closely models the performance of the aircraft. This allows development in the simulator to be transitioned to the flight testing phase seamlessly.

The system was able to function in both the simulation environment and in the real flight testing. It was able to reliably achieve and maintain flight formation. During straight and level flight, only minor errors where observed. During coordinated turns, the observed errors were higher. These observations were consistent in simulation and during flight testing.

The results of the flight testing showed that the proposed system is a functional way to coordinate a group of aircraft to fly in formation. The experimentation did show some weaknesses in the implementation of the system, and remediation measures are discussed and proposed for future work.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank my supervisor Dr. Afzal Suleman for the opportunity to engage with this material for my master's thesis. This work has allowed me to improve my engineering skills in many ways, including research, project management, optimization, troubleshooting, and project development.

This project was significantly assisted by help from the University of Victoria Center for Aerospace Research, UVic CfAR. CfAR provided me with a place to work and the equipment needed to perform my research, from computers to the aircraft used in this study. In addition, many of the team at CfAR were helpful in developing this project. Stephen Warwick, Grant Howard, Jack Sheng, and Sean Bazzocchi were all involved in providing feedback, comments, and advice on the work from the beginning. Sean Bazzocchi provided a significant amount of help in the configuration of the simulation environment, and in the commissioning testing of the UAVs. A larger group of people form CfAR assisted with the flight testing including Stephen Warwick, Sean Bazzocchi, Jack Sheng, Grant Howard, Jay Matlock, Jacob Verhelst, Binyan Xu, Mariana Santos, and Mariana Tavares. Thanks to everyone's help the system was able to be safely and effectively flight tested.

Finally I would like to thank my family and friends for their support during my continuing education. My girlfriend Katrina who listens to me talk about drones all day long, and discusses the implications of developing technology with me. My parents who have supported my path towards engineering since before I can remember, and who have taught me so much about both science and how to write technical work.

In memory of my grandparents,

John and Mary Sease, who always said

“The best thing you can invest in is your education.”

Chapter 1: Introduction

Unmanned Aerial Vehicles (UAVs), or drones, have begun to be used in groups to improve system functionality. The UVic Center for Aerospace Research (CfAR) has begun investigating the autonomous coordination of UAVs, and its associated benefits.

The increasing effectiveness and decreasing cost of equipment capable of operating UAVs has increased their availability. The increase in technology has also enabled the development of aircraft that use new modes of flight. An example is how most multi-rotor aircraft exclusively use differential thrust provided by fixed rotors for flight. This development has been allowed by the increasing availability of low cost sensors and controllers able to collect and use high quality information in real time control systems to fly the aircraft.

As the technology for commercial UAVs has become more reliable and less expensive, the roles UAVs fill have been expanding, creating new markets that were not previously practical. This includes aerial surveying and aerial photography within a small scope. The cost of a manned aircraft to perform such tasks is outside the budget of many groups who find value in such imaging, such as realtors or farmers, but UAVs can be well within their available budgets.

Another example of a new niche that UAVs are creating is aerial parcel delivery. Companies such as Amazon and DHL are working to provide package delivery directly to customers with UAVs. Others are working to make systems capable of delivering supplies to remote communities, without the need for ground infrastructure, which may not be usable at all times of year. InDro Robotics, for example, have developed Beyond Visual Line of Sight capabilities enabling them to provide air deliveries of medication to remote and island communities in British Columbia, Canada [1].

Because UAVs are smaller than traditional crewed aircraft they cost less to own and operate. This is because for any specific task the UAV can reduce size and weight due to the absence of an onboard pilot. Many small UAVs use electric power, with few moving parts. This significantly reduces maintenance costs.

UAVs are normally controlled by a human pilot located near the aircraft's takeoff location. Some operations such as surveying or searching may be performed at significant distance from the human pilot. For some operations the UAV may not be able to maintain a continuous link to the pilot, requiring it to function autonomously.

Electric Vertical Takeoff and Landing (EVTOL) passenger aircraft is another technology in development. Using UAV technology an EVTOL could fly and carry passengers without a pilot. Such aircraft are being developed with the intention of being used for the developing field of Urban Air Mobility, were they would be used for relatively short flights, into, out of, or between city centers. EVTOL operation would likely be far less expensive than other aircraft filling that role, such as helicopters. The comparatively lower cost is due to lower upfront and maintenance costs of electric motors compared to internal combustion engines, and the lower cost of electric power compared to jet fuel.

These UAVs, such as the Urban Air Mobility taxis, will need to be able to avoid collisions and share airspace. Many groups are looking into potential solutions and ways of avoiding UAV interference such as Iris Automation [2]. Enabling UAVs to optimize their flight paths around each other would allow them to reach their destinations efficiently, with the least possible risk.

In some cases UAVs may be in the location with the specific goal of supporting each other to complete a single objective. For this the UAVs would actively coordinate within a very limited airspace to accomplish their joint flight objectives. These aircraft could be acting together temporarily or for the duration of their flight. In either case, they would be acting as a Multi-Agent System. This could be done in order to provide live triangulation of sensor packages carried on multiple cooperating aircraft, or to support midair refueling.

Some groups are working on creating these multi-agent systems with a wide variety of purposes and goals. For example, Intel has developed a drone light show system that has been used at multiple high-profile events, including the Olympic Games opening ceremonies in 2018 and multiple Super Bowl half-time shows [3]. Another example is Boeing's Airpower Teaming System [4], which is intended to support a piloted aircraft with a group of autonomous companion aircraft.

This project seeks to develop an autonomous multi-agent autonomous aerial system. The system is intended to coordinate multiple UAVs to fly in formation without any ground station or human interaction.

1.1 Motivation

Motivation for this work is investigating an onboard system for coordinating multiple aircraft to fly in formation while using relatively inexpensive equipment. Onboard coordination of flight formations allows development and testing of increasingly complex UAV systems. These complex systems can greatly expand the functionality and efficiency of the aircraft and their operations.

Such systems include detect-and-avoid, in which an aircraft must use its onboard sensors to detect other aircraft so that it can alter its flight trajectory to avoid conflict. Another system is flying in formation to gain aerodynamic efficiency [5]. For example, geese fly in V-formation to reduce drag.

A group of aircraft can improve effectiveness through coordination [6]. The group of aircraft can divide tasks with each aircraft performing a different role. Dividing a survey area allows each aircraft to survey a smaller portion. The aircraft could also fly together, allowing them to combine their efforts. For example, data from an aircraft with a camera could be a combined with LIDAR data from another aircraft. Another example of combining data is using the visual information from two different aircraft to triangulate the position of a point of interest.

An onboard coordination system could be used as a traffic interaction system. Such a system could allow UAVs with different manufacturers and operators to interact [7]. Aircraft from the same operator could fly in formation to reduce the amount of airspace they require as a group, while navigating around the other aircraft.

1.2 UVic CfAR

This project is completed as part of a research project at the University of Victoria Center for Aerospace Research (UVic CfAR, or CfAR). CfAR is an aerospace research and development laboratory at the University of Victoria, British Columbia, Canada, focusing on un-manned aircraft with wingspans up to 5m and mass up to 200kg. CfAR also works with VTOL aircraft, including multirotors. CfAR often partners with corporations and agencies such as Boeing, Bombardier, Embraer, Defense Research and Development Canada (DRDC), and the Canadian Department of National Defense (DND). Projects with these groups include development and flight testing of scale technology demonstration aircraft and proof-of-concept aircraft for novel

aerodynamic testing. CfAR also works to improve the technology available for Unmanned Aerial Systems (UAS) intended for small-scale applications, such as power line surveying.

1.3 Objectives

The expanding use of Unmanned Aerial Systems (UAS) makes coordination of UAS an important area of research. This is especially true for multi-agent flight coordination of multiple UAVs. Several different approaches to operate multi-UAVs in flight are currently in development. It is important to test the methods for coordinating the UAS with the added complexity of the real-world environment. Flight testing provides final validation that multi-agent UAS systems can perform safely and responsibly in airspace shared with other UAVs and with manned aircraft. To this end, the following objectives are outlined:

a) Develop an Onboard Formation-Control System for a Multi-Agent UAS that will run onboard of a UAS which controls the movements of multiple aircraft as a single coordinated group. The program should allow for different formations to be used. The term onboard refers to the fact that this must run onboard of the aircraft, not on a remote computer such as a ground station.

b) Formulate a Flight Safety System to Enable Individual Aircraft to Avoid Dangerous Inter-Aircraft Situations to allow each individual aircraft to adjust its flight trajectory to maintain its flight safety. For this work we assume that the positions of all agents are known. Systems to detect and track non-cooperating agents are considered a separate project.

c) Create an Agent-Independent Coordination System so that any agent can be removed or added into the system without affecting performance. A fully distributed system with no centralized controller would meet this objective. A system using a centralized coordinator, which can replace the coordinator if it leaves the system, would also be considered agent independent for this purpose.

d) Develop a platform consisting of a flight simulation environment with matching aircraft to evaluate multi-agent systems without risk prior to flight testing, and making transition to flight testing smooth. The simulation environment must accurately model the aircraft so that experimental flight tests can be performed in a safe and predictable manner.

1.4 Thesis Outline

Chapter 1 introduces the background information on the growth of the UAV sector, and research motivations for exploring a distributed control flight formation of UAVs. The problem is described and the objectives are outlined.

Chapter 2 presents the current state of the art in multi-agent UAV systems, including details on multiple different methods currently being explored by other research groups for providing coordination.

Chapter 3 describes the design of fleet formation coordination and control system. This section proposes a method for an onboard fleet coordination system, discussing how centralized and decentralized controls will be implemented. It then describes how each part of the trajectory planning system functions, how the virtual leader operates within the system, and how each aircraft uses the information provided by the virtual leader to plan its trajectory to reach and maintain its position in the formation. This section ends by discussing the decentralized safety system that runs on all aircraft, allowing aircraft to gently modify their trajectories to maintain safe spacing.

Chapter 4 focuses on the simulations with a basic flight dynamics model and details a preliminary simulation of the proposed method. This simulation is performed using a low fidelity flight dynamics model of an aircraft. This simulation was used exclusively to provide a simple environment to help determine the details of the systems that would be needed for a more comprehensive version. This section explores a two-dimensional simulation of the environment, followed by a three-dimensional simulation.

Chapter 5 presents the design and validation of the simulation and real-world testing platform. The section starts by exploring the early design decisions, the selection of autopilots and onboard computers, and how they inform decision making for the rest of the research platform design. It then continues with a discussion of Robot Operating System (ROS). A series of different

program components are then described, with details on how each of them contributes to creating the multi-agent coordination system. The communication system is then explored, describing how each piece of hardware communicates with the other components. The formula to calculate the distance and direction between aircraft when using latitude and longitude is explained. Lastly the multirotors used in this work are discussed, followed by component validation, and aircraft commissioning.

Chapter 6 explains the simulations which were performed with the simulator configured in section 4. First the simulation goals are explained in detail, next the simulation environment and configuration and explored. The simulation test set is explained along with the data collected during the tests. This section ends with a brief discussion of the data, and its meanings.

Chapter 7 discusses the flight testing. It starts by discussing some assumptions that were made before the flight test, mostly relating to things anticipated to not significantly affect system performance. The details of the flight test operation are discussed. Then the data collected during the flight tests is presented and explored. The section ends with a discussion of results, and comparison with the results of the simulation performed in section 7.

Finally, Chapter 8 reports on the conclusions with a discussion of how well the results met the project objectives. It then discusses future work, including additional testing with the existing system to collect more data, and improvements to the system which could improve system performance. It also contains a discussion of other work which could use the formation control system created here to support their experimentation.

Chapter 2: State of the Art: Multi-Agent Systems

Multi-agent systems include multiple discrete systems, such as several aircraft flying in formation. Flying coordinated agents as a group can have advantages over a single, larger agent [8]. For example, military tactics coordinate flights of multiple aircraft to concentrate firepower. During a surveying, or a search and rescue operation, multiple aircraft could divide a search area into, smaller more manageable areas, increasing the speed of the operations. Sensor packages on different aircraft could be used for live triangulation, allowing for greater precision in data collection. Alternately, multiple aircraft could be equipped with different payloads that complement each other. For example, a thermal camera combined with an optical camera could be used to locate and identify subjects during a search and rescue operation.

Communication among aircraft is the foundation of multi-agent operations [9]. Communication can allow the aircraft to share instructions and other information about the situation or mission. There are a number of ways that aircraft can share information for coordinating their movements. Radio is the primary tool used for communication with UAVs.

Previous work has investigated the theoretical methods for control of multi-agent aircraft systems. This includes performing simulations of various coordination methods. Simulation has demonstrated significant potential for the methods being evaluated. Most of these control systems, however, have not been tested on actual aircraft or in real world environments. In particular, these theoretical control systems did not include limitations on aircraft computation power or radio communications.

Several methods for the multi-agent coordination were evaluated for this project. The first of these systems is Flocking [10], originally created to enable animal-like behavior in computer simulations used for video games and movies. Flocking steers each agent based on the positions and behavior of other nearby agents. The second system is Artificial Potential Field [11]. Artificial Potential Field ascribes an attractive or repulsive force to every agent, objective, and obstacle in the area of action. The combined forces are analyzed to steer the aircraft. The third

system is Leader-Follower, a relatively standard form of coordination where each agent follows a leader. A variation on leader-follower is Virtual Leader [12]. Virtual Leader replaces the leader from a leader-follower system with a simulated leader.

2.1 Flocking

Flocking is a method of directing movement of each agent by using the surrounding agents. Originally proposed by Reynolds [10] in 1987, it was initially used for movie and video game effects. Flocking has been used in several multi-agent UAV projects [12, 13, 14]. Flocking directs individual agents to move in ways similar to a flock of birds or a school of fish. This is accomplished with a variety of behavioral algorithms that respond to surrounding agents, as well as global algorithms that respond to the environment. Flocking algorithms must balance with each other, or the agents will scatter, lock their collective headings, or crowd together.

Each agent within a flocking system acts on information that it can perceive within its local area. The agents perceive information within a detection area, which is controlled by a maximum distance and a maximum angle from its forward direction as shown in Figure 1. This field of view is intended to represent the individuals' ability to actively perceive and react to the other agents around them.

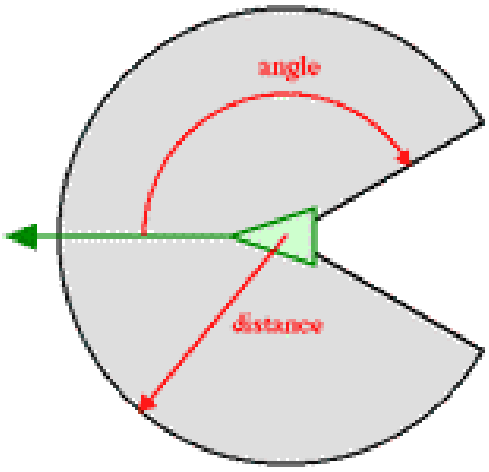

Figure 1: Flocking Agent Detection Area. Showing the angle and distance of the detection area. The agent is shown (green triangle) with its heading (green vector).
Source: Adapted from [15].

Decision calculations are made using three main algorithms: separation, alignment, and cohesion. Separation works to keep agents from getting to close to each other, preventing collisions. Alignment keeps agents moving in the same direction. Cohesion keeps agents in a group. A proper balance between separation and cohesion is required to maintain safe distances while maintaining a group. Other algorithms can be added, such as global goal-seeking. Global goal-seeking would guide all agents in the direction of an objective.

The results of each of the different algorithms are combined to create a single steering result. Each of the algorithms can be scaled when they are combined to alter behavior. The different flocking algorithms can be used in different combinations, changing group behavior.

Each of the main three algorithms collects and uses information from other agents nearby, using the detection area to control which other agents they interact with.

The separation algorithm is the basic anti-collision rule used by flocking. A vector is computed from every agent within the detection area, to the agent. The vectors are inversely scaled with distance, so that nearer agents produce larger vectors than more distant agents. The vectors are combined into one avoidance vector that points in the direction that will best allow the agent to increase distance between it and nearby agents. The combined vector is used to generate a steering directions for the system. Figure 2 shows the avoidance calculation with three aircraft within the detection area.

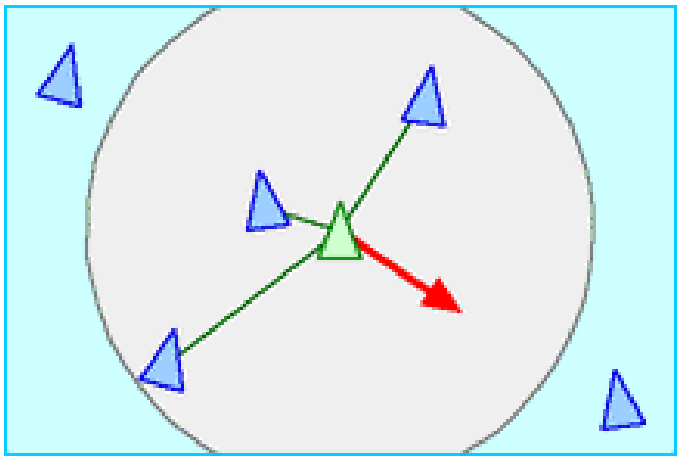

Figure 2: Boid Separation. Showing the resulting evasion vector based on three agents within the detection area. Active agent (green triangle), senses all other agents (blue triangles) within its Detection Area (grey circle), red vector shows the calculated evasion.
Source: Adapted from [15].

Alignment is used to keep agents heading in the same general direction. The heading of every agent within the detection area is averaged, this is used as the steering direction. This steers each agent to try to match its heading with the group. Figure 3 illustrates the algorithm, showing the averaged heading on the green agent, a red vector representing the required turn.

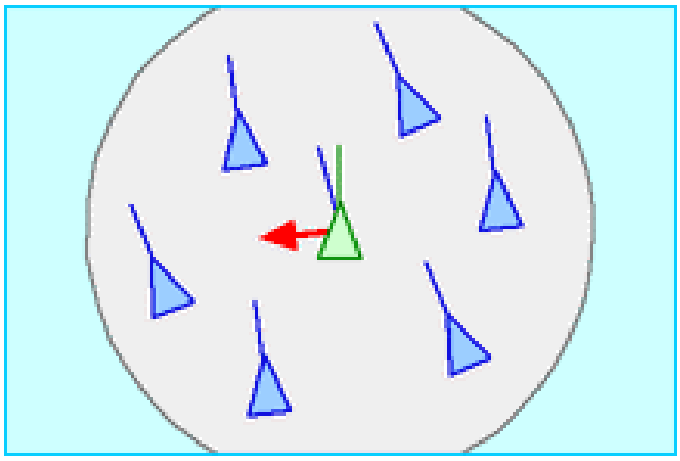

Figure 3: Alignment Diagram. Showing the alignment vector based on six surrounding aircraft within the detection area. Green triangle represents the active agent. Blue triangles represent other agents, with blue lines showing their headings. Green line shows the active agent's current heading, the blue line on active agent shows mean of detected agent headings. Red vector shows the alignment vector for the active agent. Source: Adapted from [15].

Cohesion is used to keep the group from scattering. The average position of all surrounding agents (not including the active agent) is calculated. The steering vector is calculated from the active agent to that point as shown in Figure 4.

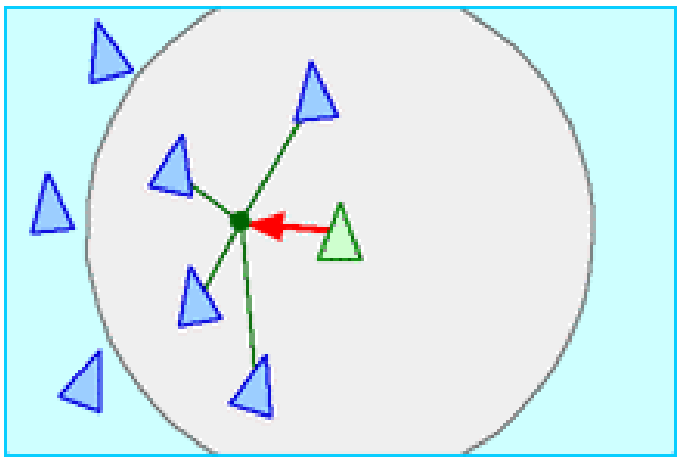

Figure 4: Cohesion Diagram. Showing the cohesion vector based on four other agents within the detection area. Green triangle shows active agent. Blue triangles show other agents. Green dot shows mean position of other detected agents. Red vector shows cohesion vector. Source: Adapted from [15].

Other algorithms can also produce steering calculations. As an example, if there is a global destination each aircraft could calculate a steering direction that would steer it towards that global goal.

The steering values created by each of the different rules are combined using a set of scaler values. The scalers can be adjusted to guide the agents to form into a loose group, like a flock of birds, or into a very close group like a tightly packed school of fish. Improper balancing of separation and cohesion will cause the agents to scatter or crash into each other. Cohesion is needed to keep the system traveling in a consistent direction. But, if it is too heavily weighted it could interfere with the system's ability to turn.

A Model Predictive Control (MPC) based implementation of Flocking was tested using a group of multi-rotor aircraft [16]. That test used one aircraft as a reference that the other aircraft would observe, allowing the researchers to guide the system's position.

2.2 Artificial Potential Systems

Artificial potential field assigns an attractive or repulsive force to every agent, obstacle, or point of intersect within the area [11, 17, 18, 19]. These forces are combined into a resulting force on each agent. The resultant force is sent to the control system in the aircraft to alter the agent's trajectory as if acted on by that force. Artificial potential has previously been used in conjunction with Virtual Leader in simulation to coordinate a group of fixed wing aircraft [18].

Each agent in the system is given a repulsive value. The value of the repelling force can be varied to coincide with differing constraints, or they can be the same for all agents within the system. Because the agents all have a repulsive value they will inherently push each other away, keeping agents from getting too close together.

Obstacles are also given repulsive values. Values given to the obstacles can be similar to those of the agents, or different. This creates an effect where the agents will steer around obstacles similar to how they steer around each other.

Local objectives or points of interest are given attractive values. This will cause the agents to prioritize traveling to those locations as opposed to wandering the area. Depending on the nature

of a mission or objective, the goals can be turned off once an agent has reached it, this allows for the system to adjust priorities during a mission.

In addition a global potential field can be added to the system. This could cause the aircraft to inherently move in a particular direction, e.g. east. Or it could steer to a particular location.

At any given time the resulting forces of the system can be used to calculate a multi-dimensional surface to visualize the potential energy across the area, as seen in Figure 5. The agents will follow the path of quickest descent within the system, similar to the path shown in Figure 5.

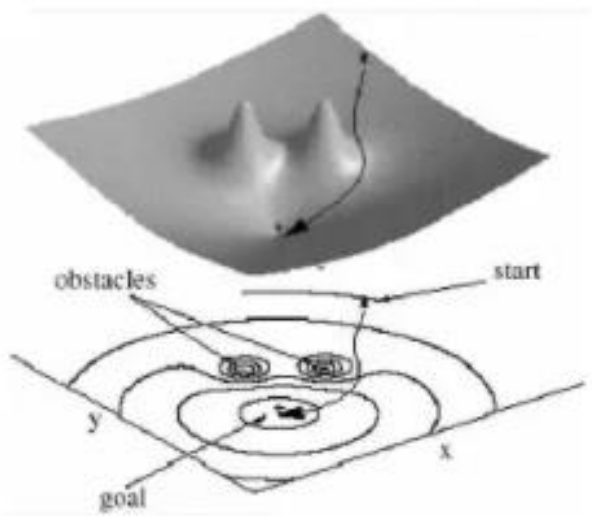

Figure 5: Artificial Potential Field. Shown in both two-directional (bottom) and three-dimensional representations (top). The arrow shows the path of steepest decent and agent would follow when affected by the represented potential field. This is analogous to water flowing over mountainous terrain.
Source: Adapted from [20].

Due to the potentially complicated nature of the resulting potential field, it is possible that there will be multiple local minima that could cause problems for the agents, as shown in Figure 6. If an agent does not have a path planning system that allows it to go backwards it may become stuck in a local minimum. Various solutions have been developed solving the local minima problem [11, 17, 19].

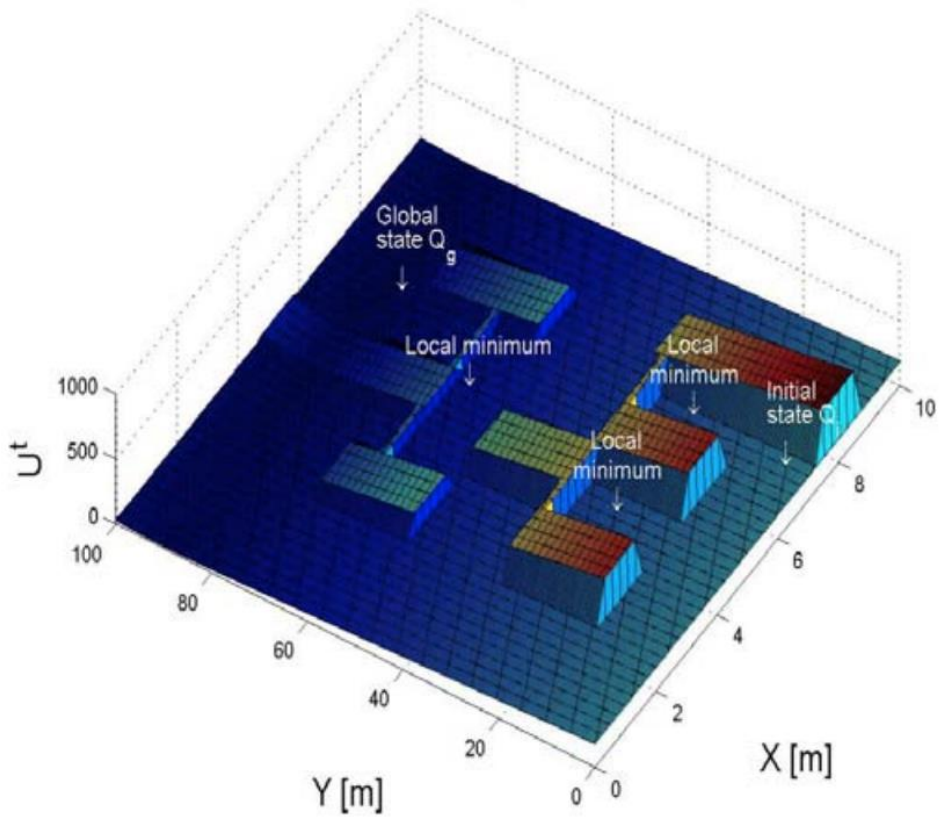

Figure 6: Artificial Potential Field with local minima. Height and color to show potential field, with red being high potential ranging to blue as low potential. Local minima are created within the field where they are surrounded by higher potential. An agent starting at the indicated initial state position could be trapped within one of the local minima.
Source: Adapted from [19].

Although the risk of local minima is relatively low in open airspace, when there is traffic near static obstacles, the risk of local minima increases. This is particularly worth consideration in the context of urban air mobility.

2.3 Leader-Follower Systems

Leader-follower is a simple way to coordinate a group of agents [21, 22]. One agent is the designated leader of the formation. Each follower in the system moves to a specific location with respect to the leader of the formation. The leader-follower system relies on the follower agents knowing the location of the leader, but does not require the leader to know the positions of the followers.

Leader-follower can be organized in one of two ways. There can be a global leader. In this method every agent looks only to the global leader to determine where they should be. This

method does not necessarily have communication between any aircraft other than the location of the leader to the followers. This lack of communication means there is no inherent feedback to prevent two followers from colliding, or to allow the leader to act if a follower is getting to close.

The second method divides the leader responsibility among aircraft. In this method each agent relies on its own individual leader, for example in a Delta formation each agent may look to the aircraft immediately to its left or right.

One notable difference between the two methods is shown if an agent is out of position. In the global leader method one agent could be missing, potentially from any position other than the leading position. In the more distributed method, a missing agent could take a large part of the formation with it, because it could be the local leader of a long chain of other agents, leading them out of position.

2.3.1 Virtual Leader

The Virtual Leader method is a modification of a standard leader follower method. With a simulated leader the rest of the system can be configured to use the simulated leader as either a global leader, or the first leader in a distributed leader system. This method has been explored in multiple applications [12], including simulated flight with fixed wing aircraft [18]. As an example, if a square formation were desired each agent could be given a location around a virtual agent that would be at the center of the formation. Although collisions are not specifically addressed, the virtual leader cannot be hit by other aircraft.

Because the virtual leader does not interact with the physical world, any effects that will change how the followers interact with the world will not affect the leader. This means that if the fleet is flying into a headwind the Virtual Leader would need to be informed so that it could alter its ground speed, otherwise it would lead the fleet at an inefficient airspeed or even leave the followers behind.

The virtual leader needs to be simulated from an existing computer within or connected to the system. The simulation could be performed onboard of one of the agents, or can be done on a connected base station.

2.4 Centralized and Distributed Systems

A multi-agent system can be designed to run as a distributed system, or as a centrally controlled system. A centralized system passes all important information to a single point, where decisions are made for the whole system. In a decentralized system, each element within the system is responsible for making the decisions for its own actions.

Each method has advantages and disadvantages. Because a centrally controlled system has one component making all decisions with the relevant data, it has the ability to coordinate more effectively making a more efficient system. But a failure in a centrally controlled system could cause a large system failure. Faulty communication can degrade the quality of information used for decision making, and can prevent sub-agents from receiving timely instructions. In a distributed system the system response to inputs may not be optimal, because the individual agents do not have all relevant information, but the distribution of responsibility makes the system more resistant to system failures, or to other issues which interfere with the system, such as communication speed.

Most multi-agent UAV systems rely on a central system for coordination [23]. In many cases this is a ground station computer which is capable of planning and optimizing the trajectory of every single aircraft in the system.

Some applications require more redundancy than a centralized system provides. A decentralized control structure is far more suited to applications where not every component can be expected to maintain a strong continual link to a single central system. An example of this is search and rescue, where rough terrain may prevent communication between most aircraft within a system, potentially dividing them in to multiple small groups. Another example is military or police operations where an opposing force may target the systems and render one or more aircraft inoperable during the flight mission. In both of these examples a decentralized system would prevent a full system failure due to a loss of some agents or communications.

Chapter 3: Design of Fleet Formation Coordination and Control System

3.1 Multi-agent Control and Decision Making

The primary concerns that must be addressed with the coordination scheme are formation control and collision avoidance. Both will be performed from onboard the aircraft. This will satisfy Objectives A and B.

Multiple methods, including Flocking, Artificial Potential, and Leader Follower, have been explored and used to allow multiple agents to coordinate their movements. This project used a combination of Flocking and Virtual Leader Follower to implement the Self-Controlled Fleet-Formation Coordination.

By combining these two methods the advantages of each will compensate for the disadvantages of the other. The primary coordination of the formation will use Virtual Leader Follower architecture, while flight safety and inter-agent interaction will be performed using flocking.

As mentioned previously, all aircraft will share their positions will all other aircraft. Detection of non-cooperating aircraft is not part of this work.

3.1.1 Centralized and Decentralized Control

The combination of Virtual Leader and Flocking allows for a combined Centralized/Decentralized Control structure. The Virtual Leader uses a centralized system to coordinate the formation. Flocking uses a decentralized system for aircraft interaction and safety.

The formation is controlled from onboard of a single aircraft. This aircraft acts as the center of the system for coordination of formation and movement. All aircraft run their own flocking algorithms, directly sharing all relevant information to maintain a decentralized safety system.

The Virtual Leader system was developed so that it could run from any aircraft.

3.2 Multi-Agent Unmanned Aerial System Coordination Plan

A Multi-Agent Unmanned Aerial System (MAUAS) was designed. The system is divided into multiple components: Aircraft Guidance (called Boid Brain), the Coordination and Virtual Leader (called Big Sister), and Inter-Agent Communication (called Collective Consciousness).

The virtual leader of the formation is simulated by Big Sister. Big Sister also determines additional coordination details, such as which formation to use. The Big Sister runs the Virtual Leader based on a preset flight path, which can require specific flight speeds or turning rates. It can also command flocking configurations used by the Boid Brains.

Each agent's Boid Brain calculates its own trajectory. Each agent's Boid Brain uses an individual preset offset from the Virtual Leader to determine its position, depending on the formation. Boid Brain also uses flocking algorithms to augment its steering controls based on the selection commanded by the Big Sister. Flocking is used primarily to provide non-emergency collision avoidance. The Boid Brain contains an emergency evasion mode, which will take evasive maneuvers if another aircraft gets too close.

3.2.1 Virtual Leader Simulation: Big Sister

The virtual center will follow the flight path the system is assigned. The flight path may contain additional commands, such as formation changes. Big Sister controls how the fleet will follow the Virtual Leader. Big Sister can change the leader follower configuration to use one of the agents as a conventional leader. If the fleet is following a conventional leader the Virtual Leader will use that position and velocity as its state when it switches back into Virtual Leader mode.

The Virtual Leader uses Equation (3.1) to update its position. The Virtual Leader uses Equation (3.2) to update its heading.

$$P_{New} = P_{Old} + \vec{V} \times \Delta t_{time\ step} \tag{3.1}$$

$$\theta_{New} = \theta_{Old} + \dot{\theta} \times \Delta t_{time\ step} \tag{3.2}$$

Equations (3.1), and (3.2), where P is position, θ is the global heading of the virtual center, V is linear velocity of the virtual center, and Δt is the time interval.

3.2.2 Individual Aircraft Guidance: Boid Brain

Formation Control

The Boid Brain calculates its target position using Equation (3.3), using the aircraft's position in formation and the formation's heading to calculate the individual aircraft's target position. The target velocity is calculated using equation (3.4), using the velocity of the Virtual Leader, modified with the formation's rotational speed and the aircraft's position in formation. The aircraft's target velocity is calculated using the PD loop shown in Equation (3.5), using the aircraft's target velocity ($\dot{P}_{Goal}$) as a feed forward value, determining the velocity the aircraft must fly to reach and maintain its position in formation.

$$P_{Goal} = P_{Fleet} + \theta_{Fleet}\Delta_{Formation} \tag{3.3}$$

$$\dot{P}_{Goal} = \dot{P}_{Fleet} + \dot{\theta}_{Fleet}\Delta_{Formation} \tag{3.4}$$

$$\vec{V}_{Action} = \dot{P}_{Goal} + P(P_{Goal} - P_{Actual}) + D\left(\dot{P}_{Goal} - \dot{P}_{Actual}\right) \tag{3.5}$$

Equations (3.3), (3.4), and (3.5), where $\Delta_{Formation}$ is the local offset of an aircraft's positon in formation (this uses the fleet's reference frame), *P* and *D* are the p and d gains of the PD control loop.

Flocking

Flocking is used for collision avoidance. This was expected to be most important when entering, leaving, or changing formation. The system can ignore the Virtual Leader and use only Flocking algorithms, where the aircraft sets its velocity to have a constant speed while maintaining its current heading. Flight heading and speed are then modified using the Flocking algorithms.

Implementations of the flocking equations are shown in Equation (3.6) Avoidance, Equation (3.7) Alignment, and Equation (3.8) Cohesion. The flocking algorithms are combined in Equation (3.9). The scalers used for each of the flocking equations can be adjusted to change the flocking behavior. The Boid Brain will change these variables based on directions received from Big Sister.

$$\overrightarrow{Vav} = \frac{-1}{n}\sum_{i=1}^{n}\left(\vec{P}_i\frac{RT}{\left\|\vec{P}_i\right\|}\right) \tag{3.6}$$

$$\overrightarrow{Val} = \frac{1}{n}\sum_{i=1}^{n}(\overrightarrow{Vl_i}) \tag{3.7}$$

$$\overrightarrow{Vco} = \frac{1}{n * RT}\sum_{i=1}^{n}(\vec{P}_i) \tag{3.8}$$

$$\vec{V}_{flocking} = S_{av}\overrightarrow{Vav} + S_{al}\overrightarrow{Val} + S_{co}\overrightarrow{Vco} \tag{3.9}$$

Equations (3.6), (3.7), (3.8), and (3.9), where $\overrightarrow{\boldsymbol{Vav}}$, $\overrightarrow{\boldsymbol{Val}}$, $\overrightarrow{\boldsymbol{Vco}}$ are the velocity targets from Avoidance, Alignment, and Cohesion respectively. $\overrightarrow{\boldsymbol{P_i}}$ is the position of aircraft i with respect to the agent. RT is the detection range, default is 20 m. $\overrightarrow{\boldsymbol{Vl_i}}$ is the velocity of aircraft i. Sav, Sal, and Sco are the scaler values for Avoidance, Alignment, and Cohesion respectively.

Boid Brain combines the velocity targets generated by Formation Control from Equation (3.5) and Flocking from Equation (3.9), to generate the velocity command for the aircraft to use, as shown in Equation (3.10).

$$\vec{V}_{comand} = \vec{V}_{Action} + \vec{V}_{flocking} \tag{3.10}$$

Equation (3.10), where $\vec{\boldsymbol{V}}$ is velocity.

3.2.3 Inter-Agent Communication: Collective Consciousness

Big Sister sends information to the Boid Brains using Collective Consciousness, and the Boid Brains use Collective Consciousness to share their position and velocity information with each other for flocking.

The Collective Consciousness can also communicate other information between the aircraft if required.

Chapter 4: Flight Dynamics Model

The coordination system was modeled using MATLAB Simulink. The initial model was created using two dimensions of freedom, to allow for a relatively quick and low cost simulation to provide an initial proof of concept. This simulation allowed for easy introduction of new behaviors and controls into the system.

After validating the system in two dimensions, the simulation was increased to three dimensions. Three dimensional simulation allowed for improved testing.

4.1 Simulation Model

The initial Simulink simulation is shown in Figure 7. Custom functions for steering the aircraft and simulating the Virtual Leader were made using MATLAB functions. The simulation contained one sub-model for the Big Sister, and three instances of a sub-model representing a Boid Brain and its aircraft.

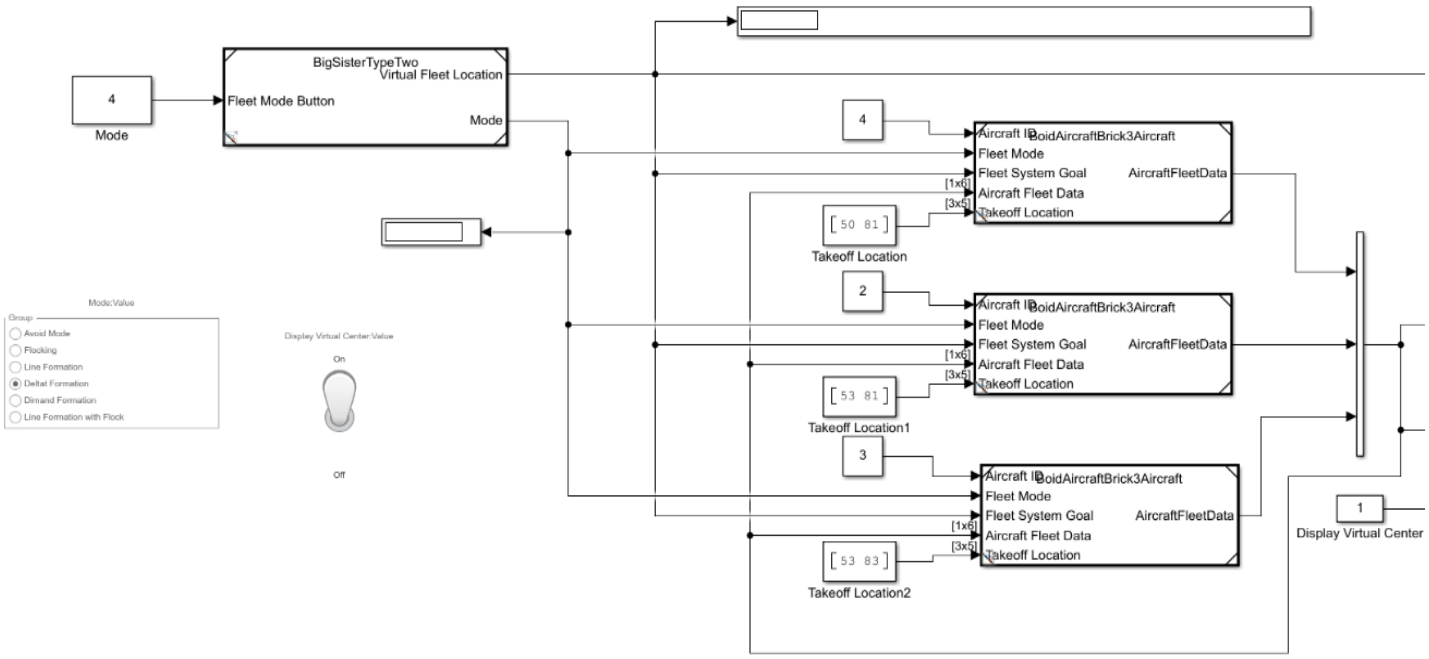

Figure 7: Simulink Fleet Model. Containing Big Sister (bold box on left) and three Aircraft Blocks (bold boxes on right).

The Big Sister model is shown in Figure 8. This shows the connections between the custom flight control algorithm and the Virtual Leader position simulation. The flight control algorithms use current state information to calculate the thrust vector and the rotational speed, for the Virtual Leader to stay on course. The rotational speed is used to update the heading of the aircraft. Thrust is modified by a basic form of air resistance, using the aircraft velocity, to produce the net force on the system. Net force is used to calculate the acceleration, and to update the velocity and position.

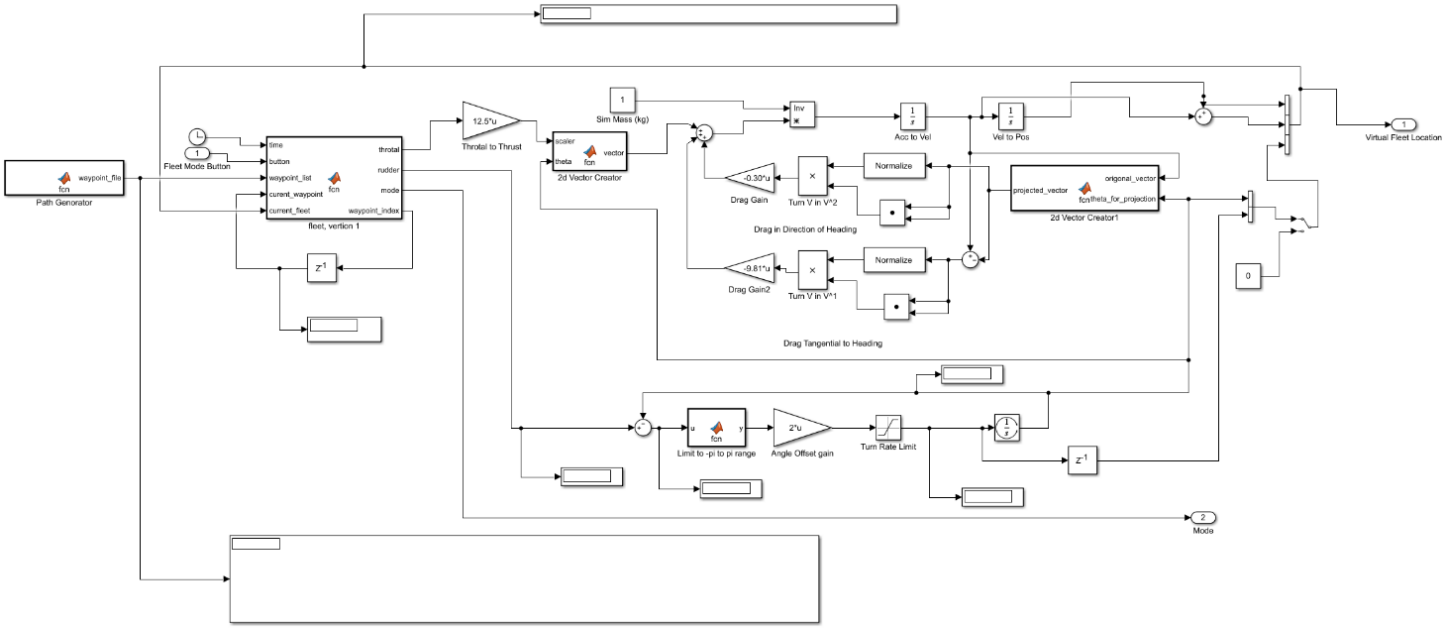

Figure 8: Simulink Big Sister block with Virtual Leader. Virtual steering calculation (bold box top left), position and velocity simulation (top middle to right), drag feedback (middle right), and heading (bottom right).

This simulation model results in a simulated Virtual Leader that acts in a relatively realistic way. This model is not a high fidelity representation of an aircraft, and moves slightly differently than an actual aircraft. This is sufficient to operate the Virtual Leader within this simulation.

The model of the aircraft is broken down into multiple layers. The top layer is shown in Figure 9. This model separates the custom steering of the Boid Brain from the aircraft sub-model that contains the autopilot and aerodynamics.

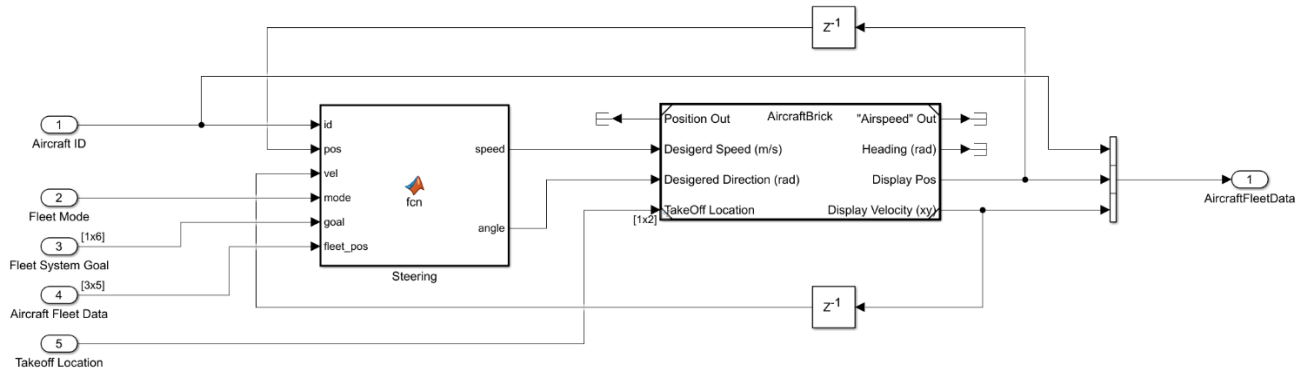

Figure 9: Simulink Top Level Aircraft Model. Custom steering (left) and aircraft model (right).

The aircraft model is divided into a simulated autopilot and a flight dynamics model (Figure 10).

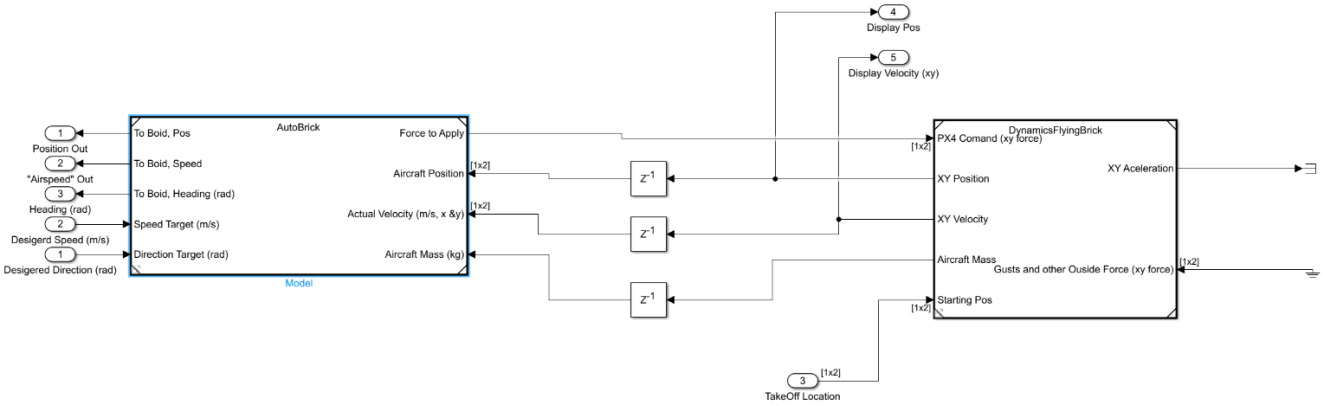

Figure 10: Simulink aircraft model. Aircraft Autopilot (left), and Flight Dynamics Model (right).

The autopilot model is shown in Figure 11. It is given the desired direction and speed, and calculates the force needed to change the current velocity to the target velocity. The output force includes a drag compensation calculation based on aircraft speed. Force output is limited to 30N. This limit combines with drag in the flight dynamics model, giving the system a maximum speed.

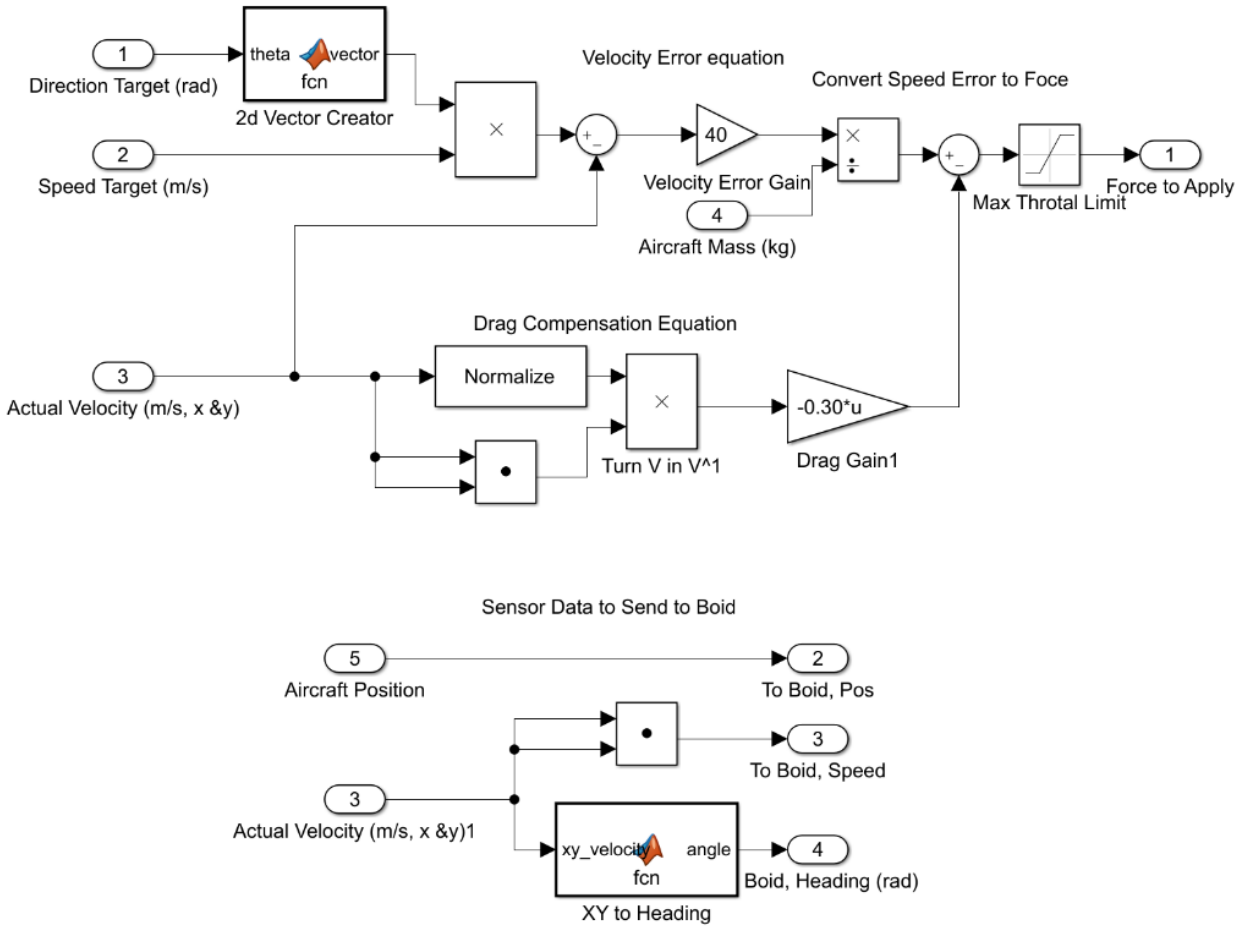

Figure 11: Simulink Autopilot. Converting direction to aircraft vector (top left), drag compensation (middle), and conversion of observed data to send to the steering system (bottom).

The flight dynamics model, shown in Figure 12, is not high fidelity. It applies forces on the x and y axes, and uses a point mass to simulate the aircraft. Drag is added to the force acting on the point mass. Due to drag compensation, drag only visibly affects movement when the autopilot reaches the thrust limit.

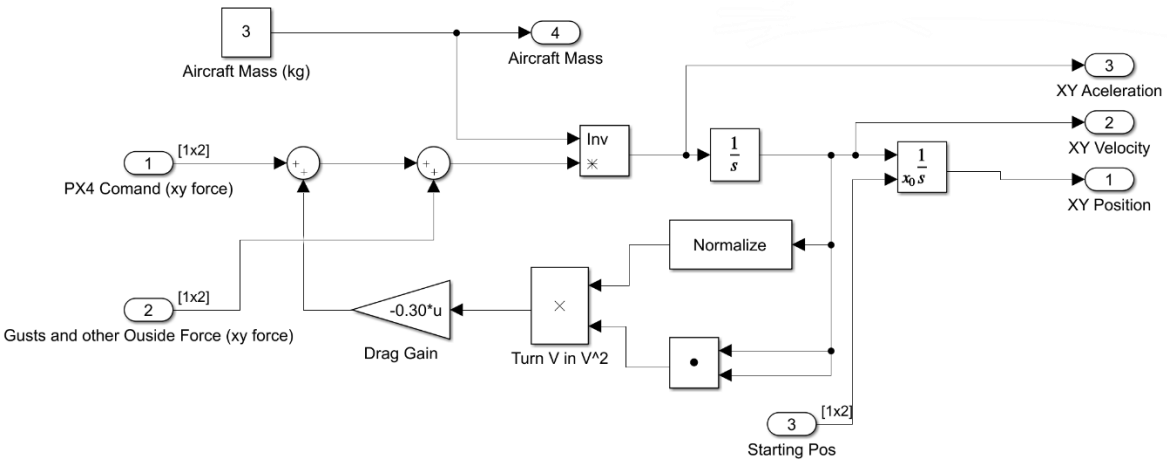

Figure 12: Simulink aircraft Flight Dynamics Model. Using force to calculate the aircraft acceleration, velocity, and position (top), including drag feedback (bottom).

The models presented above are two dimensional simulations. The three dimensional simulation added a z axis, but is otherwise the same. In both two and three dimensional simulations, sensors noise, filtering, and communication delays were not modelled.

4.2 Simulation Results

4.2.1 Two Dimensional Simulation results

Simulation of a line formation is shown in Figure 13. Each aircraft is represented by a circle. The velocity of each aircraft is shown by a blue line. Velocities are meters per second, while positions are meters. The Virtual Leader flies counterclockwise. The middle aircraft in the formation's position has no offset from the Virtual Leader.

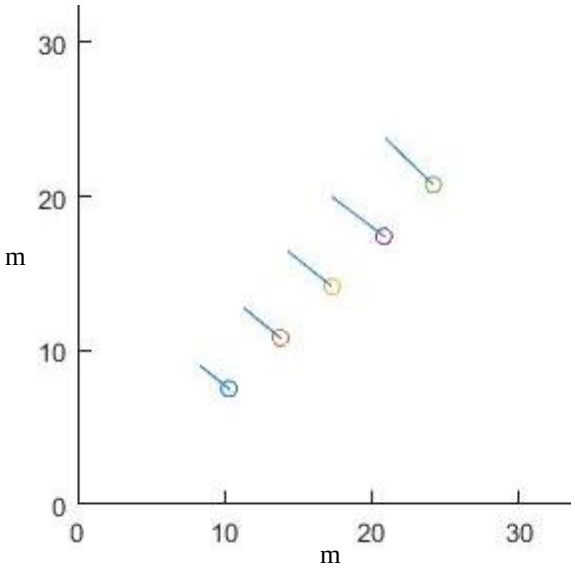

Figure 13: 2-D Simulink Line Formation, flying in a counter Clockwise Arc. Each circle represents an aircraft, and blue lines represent aircraft velocity. Magnitude of velocity is represented by length of line. Distance between hatch marks on axes represents 10m distance, or 10m/s velocity.

Figure 14 shows a delta formation simulation, flying counter clockwise. The Virtual Leader is located behind the lead aircraft, and between the aircraft of the second row. The Virtual Leader's arc has the same radius as above.

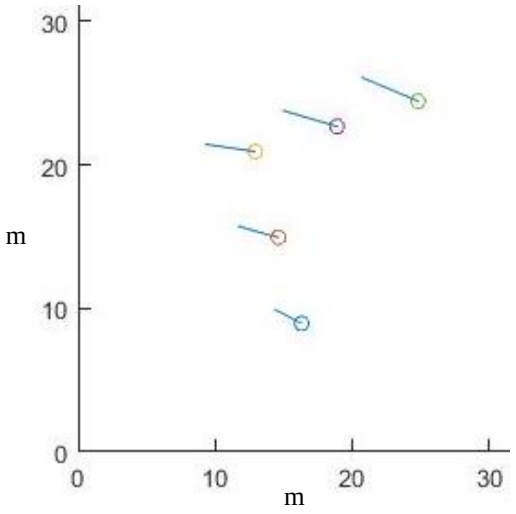

Figure 14: 2-D Simulink Delta Formation, flying in a counter clockwise ark. Each circle represents an aircraft, and blue lines represent aircraft velocity. Magnitude of velocity is represented by length of line. Distance between hatch marks on axes represents 10m distance, or 10m/s velocity.

Because the virtual fleet is flying in a circular flight path, the flight speeds of the aircraft differ based on their positions within the formation. The aircraft on the outside of the turn must fly faster, while the aircraft on the inside of the turn must fly slower than the Virtual Leader to maintain position.

It was observed that if a minimum flight speed and maximum turning rate were applied to the aircraft model it would cause emergent behavior for any aircraft unable to fly as slowly as the turn required to maintain their position. This resulted in the aircraft alternately turning left and right, increasing the size of the oscillation as it got further ahead of its target point. This looks like the aircraft is trying to lose speed by rapidly turning. Once the aircraft was too far ahead it would execute a sharp turn, passing around the inside edge of the formation and returning to its target point from behind.

4.2.2 Three Dimensional Simulation

The three dimensional simulation operated successfully without problems. Figure 15 shows the simulation. Vertical lines were added from each agent to the ground plane directly under the aircraft, to help with understanding the aircraft positions in space.

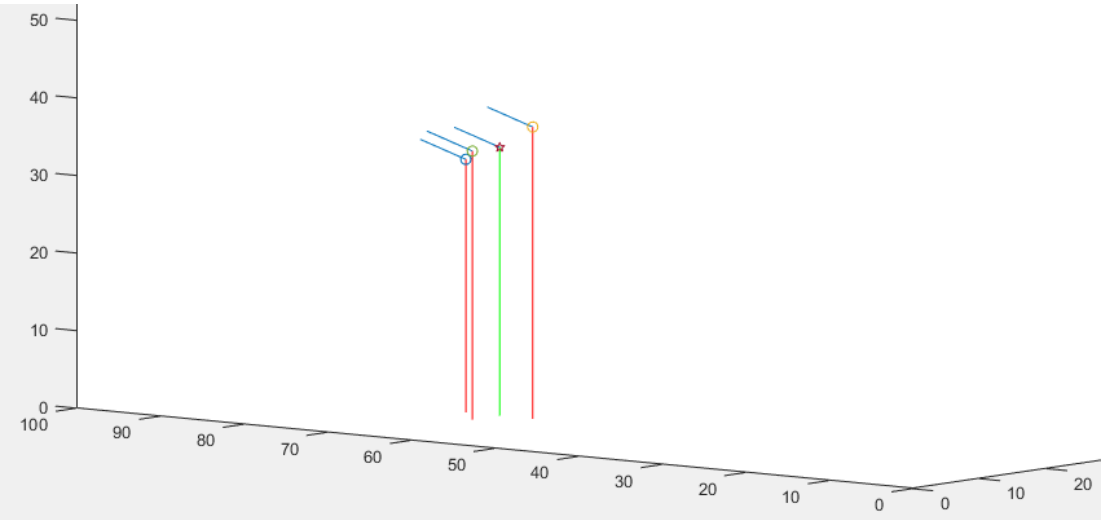

Figure 15: 3-D Simulink Simulation of three aircraft around the Virtual Leader. The star represents the virtual leader, circles represent follower aircraft. The blue lines represent aircraft velocity vectors. Magnitude of velocity is represented by length of line. Red and green lines travel straight down from aircraft to the ground plane, as an aide to identify relative positions. Distance between hatch marks on axes represents 10m distance, or 10m/s velocity.

The system was shown to operate effectively in the three dimensional environment. It responded effectively to the Virtual Leader, and Flocking functioned as intended.

Additional testing investigated the resilience of the system. These tests included adding manually controlled wind gusts which could be activated intermittently or continuously. The wind gusts had little effect, with the autopilot compensating accurately. The wind gusts would only affect the aircraft positions if the required airspeed were higher than the aircraft limits.

The Simulink simulations adequately demonstrated the proposed fleet formation control system, showing that the MAUAS could be taken to a more detailed simulation environment.

Chapter 5: Simulation Platform and Aircraft Design and Implementation

A research platform was developed to allow for both simulation and flight testing of the MAUAS. This system was developed as part of Objective 4, with the intention of being usable for other applications after this project is completed. Accordingly, the system must have the ability to be customized to specific needs as other projects require.

5.1 General Design

Each aircraft uses generic off-the-shelf components where possible. A small computer was mounted onboard each aircraft providing computational power for experimental systems. For this project the onboard computer ran the coordination software and sent instructions to its autopilot.

5.1.1 Autopilot

The open source PX4 UAV autopilot firmware was selected as the foundation for the research platform. Because PX4 is open source it can be modified and customized as needed. PX4 is already in use at CfAR for other systems, so there is institutional familiarity with the firmware, and additional experience during this project will contribute to other projects at CfAR. PX4 firmware can run on many types of autopilot boards, including custom-designed boards. Additionally, PX4 is compatible with 3rd party simulation environments and external software systems, such as ROS and the simulation environment Gazebo.

The PX4 has many different flight modes. One mode is Offboard. In Offboard mode the PX4 listens for commands coming to the autopilot from another computer. These commands can control position or velocity.

The PX4 also contains built-in safety systems, which can trigger pre-determined actions. These safety systems can be triggered by conditions such as low battery, loss of communication with the ground station or human pilot, or passing beyond a geo-fence, which defines an area where flight is permitted. The most typical response when triggered is for the aircraft to return to and land at its takeoff point.

The position of the aircraft is calculated by the PX4 firmware using an Extended Kalman Filter (EKF) which combines GPS, accelerometers, gyros, etc. With the EKF there is a small but noticeable level of drift that affects the system's position estimate.

5.1.2 Multi-Agent Software

The MAUAS is intended to run entirely on the onboard computer (OBC), which was mounted to the aircraft and connected to the autopilot. The MAUAS software uses Robot Operating System (ROS), because it allows for a modular design.

Robot Operating System

ROS uses a system of nodes and messages. ROS nodes are small independent subprograms, each performing a separate function. The nodes communicate by sending messages over dedicated communication channels used exclusively for that subject, called topics which use TCP/IP. Any node can be set to communicate with any selection of topics. This allows each node to get all the information it needs from other nodes directly, while ignoring information that is irrelevant. The MAUAS created here used several different nodes for each aircraft.

MAVROS

MAVROS is a software package for ROS which directly interfaces between a PX4 autopilot and the OBC where MAVROS is running. MAVROS runs as a ROS node that collects the information available from the PX4 and makes it available using ROS topics. MAVROS also sends messages it receives from other ROS nodes on the OBC and passes them to the PX4. In this way MAVROS acts as a software interface for the PX4.

Custom ROS Implementations

Three primary ROS nodes were created to implement the MAUAS, including Boid Brain, Big Sister, and Collective Consciousness.

Boid Brain

The Boid Brain node controlled the actions of the aircraft in flight. The PX4 autopilot could be switched into Offboard flight mode, allowing the Boid Brain's flight commands to be used. Boid Brain switches among several modes: Emergency Evade, Formation Control, and Flocking.

Emergency Evade

The Emergency Evade mode was based on a pre-determined safety threshold. If any other aircraft was within this distance threshold, the active aircraft immediately took evasive action. When evading, the aircraft calculated an evasion vector Equation (5.1) along which it flew. The evasive vector combines the vectors from all aircraft that are within the safety threshold, inversely scaled with distance. The aircraft treated a minimum altitude as an aircraft directly below itself. This prevented the aircraft from accidently flying too low or being pushed to the ground.

$$\overrightarrow{Vl}_{comand} = \sum_{i=1}^{n} \frac{-1}{\overrightarrow{V_l}} \tag{5.1}$$

Equation (5.1), where $\overrightarrow{V_l}$ is the vector from the active aircraft to aircraft i within its safety perimeter.

The inverse nature of Equation (5.1) means that the aircraft will consider closer aircraft as a more significant threat, prioritizing evading them above aircraft that are further away. It will continue to evade other aircraft within the distance threshold, reassessing the relative threats as distances change. If evasion is not necessary, Boid Brain functions in the Formation Control, Flocking, or a combined mode, based on the Fleet commands that it is receiving.

Formation Control

Formation Control is the core function of the Boid Brain system. Formation Control analyzes information about speed, position, and trajectory of the aircraft in relation to the Virtual Leader, and calculates the target velocity required to fly in formation. Equations used for these calculations are provided in Section 4. The PD control loop shown in Equation (3.5) was used to generate the velocity command for flying to and within formation.

Flocking Functions

Boid Brain calculated the velocity output generated by the flocking algorithms, as calculated by Equation (3.9). The scalers for the three different algorithms were selected based on a command from Big Sister.

Formation Boid Synthesis

The velocity vectors generated by the Formation Control and the Flocking systems are combined using Equation (3.10). This was sent to the PX4 through MAVROS, directing the aircraft.

Big Sister

Big Sister runs on only one aircraft at a time. Big Sister determines which formation the aircraft should use, where the formation should be centered, and the selection of flocking algorithms to use. Big Sister can run in one of two modes: Virtual Leader mode, or Follow the Leader Mode.

Virtual Leader Mode

In Virtual Leader mode Big Sister simulates a virtual aircraft. It sends the position and velocity of that virtual aircraft to all the aircraft in the system for the Boid Brains calculations.

The simulation used to create the virtual center was relatively simple, using a version of the model described in Chapter 5, modified to calculate position using latitude and longitude. Section 5.2 provides and discusses the equations used to incorporate latitude and longitude into the formation simulation calculations. Virtual Leader mode runs only when the PX4 autopilot of the aircraft is running in Offboard mode, otherwise the autopilot runs in Follow the Leader mode.

Follow the Leader Mode

Follow the Leader mode used the position and velocity data of the aircraft running Big Sister for the Virtual Leader. All other aircraft followed the Virtual Leader.

Operating in Follow the Leader mode did not change the settings of Boid Brain; but Big Sister functioned in Follow the Leader only when the PX4 autopilot of the Big Sister aircraft was not in Offboard mode. This was necessary, because if the aircraft was supposed to be in a position other than the exact center of the formation, its Boid Brain would as respond as though the aircraft were out of position. Running the system in Follow the Leader mode removed all of the

benefits of the Virtual Leader. The Follow the Leader mode was added to allow simpler control by human pilots for test setup and safety. This allowed a human pilot to guide the formation into its starting position, before passing control to the Big Sister using the Virtual Leader.

Collective Consciousness Node

The Collective Consciousness node is responsible for all inter-agent communication. The implementation was altered based on whether the system was simulated or in the real world. In both configurations of the system, the Collective Consciousness forwards information from Big Sister and Boid Brain to other aircraft in the system, using ROS topics or radios.

PX4 Interface

The PX4 Interface node was responsible for managing the interface between the MAVROS node and the other nodes in the multi-agent coordination system.

The PX4 Interface could change the flight mode of the PX4 autopilot. It could switch the aircraft into or out of Offboard mode. It could also initiate aircraft takeoff. This was used only in simulation due to safety concerns.

5.1.3 Communication Architecture

The communication architecture of the system is shown in Figure 16. System components interact with each other in the simulation. Everything within the OBC is a ROS node.

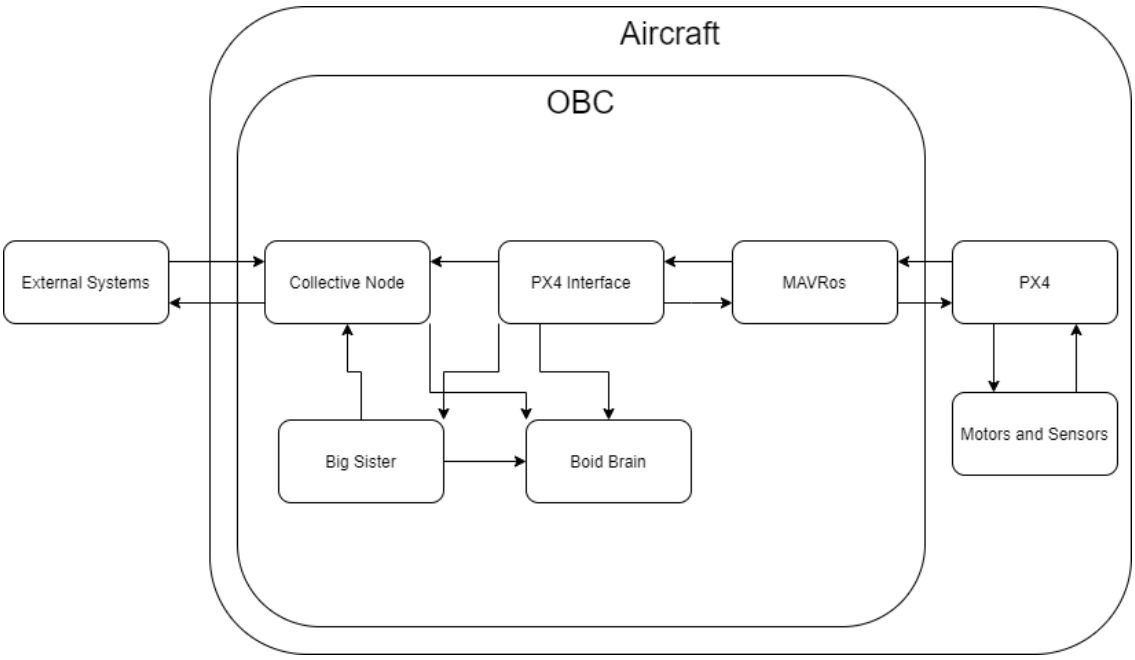

Figure 16: MAUAS Aircraft Architecture. Showing the four custom ROS nodes which communicate with the PX4 Autopilot through the MAVRos node. The custom ROS nodes also communicate with external systems on other aircraft or on the ground. Arrows show flow of information.

The flight test system differs only slightly from Figure 16. The aircraft were equipped with inter-agent radios. The radios were mounted onboard the aircraft and directly connected to the OBC, communicating directly with the Collective Consciousness node.

5.2 Navigation Model

The OBC received global position information from the PX4 autopilot. The PX4 used an Extended Kalman Filter (EKF) to determine aircraft position. The EKF used GPS, accelerometer, magnetometer, and gyros to estimate aircraft position, velocity, and orientation [24].

The distance between two points on the earth, in latitude and longitude, was calculated using Equation (5.2) [25] . Further discussion of navigation around a sphere is provided in [26].

$$d = R \times 2 \times atan2\left(\frac{\sqrt{\sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos\varphi_{agent}\cos\varphi_{target}\sin^2\left(\frac{\Delta\lambda}{2}\right)}}{\sqrt{1 - \sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos\varphi_{agent}\cos\varphi_{target}\sin^2\left(\frac{\Delta\lambda}{2}\right)}}\right) \quad (5.2)$$

Equation (5.2), where d is the distance between the agent and target, R is the radius of the earth, φ is latitude in decimal degrees, $\Delta\boldsymbol{\varphi}$ is the difference in latitude between the agent and target in decimal degrees, and $\Delta\boldsymbol{\lambda}$ is the difference in longitude between the agent and target in decimal degrees [25].

Equation (5.2) assumes that the earth is a sphere, however earth is closer to an ellipsoid than a sphere [27]. There are more complex equations which produce more accurate results [28]. The errors caused by the spherical assumption, however, were not expected to be significant for the purposes of the multi-agent coordination system used in this project. Therefore the increased computational load to use a more accurate equation was not necessary.

Equation (5.3) [25] was used to calculate the bearing between two points in latitude and longitude. The bearing resulting from this equation is not constant over the direct path between the points, unless the bearing is directly north-south or along the equator. Consequently bearing must be recalculated if either point changes.

$$\theta = atan2\left(\frac{\sin\Delta\lambda\cos\varphi_{target}}{\cos\varphi_{agent}\sin\varphi_{target} - \sin\varphi_{agent}\cos\varphi_{target}\cos\Delta\lambda}\right) \quad (5.3)$$

Equation (5.3), where $\boldsymbol{\theta}$ is the bearing from the agent to the target, $\Delta\boldsymbol{\lambda}$ is the difference in longitude between the agent and target in decimal degrees, and φ is latitude in decimal degrees [25].

The Cartesian vector between these two points can be calculated using the distance and the bearing. The vectors between individual aircraft must be identified within a local frame. For this project, the North East Down (NED) frame was selected, as it is one of the most commonly used in aviation. The calculation of the NED vector between the aircraft is given in Equation (5.4).

$$\underrightarrow{Vc} = \begin{pmatrix} d \times \cos\theta \\ d \times \sin\theta \\ \Delta altitude \end{pmatrix} \quad (5.4)$$

Equation (5.4), where $\overrightarrow{\boldsymbol{Vc}}$ is the vector using the NED frame, with distance d, heading $\boldsymbol{\theta}$, and vertical distance from agent to target $\Delta\boldsymbol{altitude}$ (down is positive).

The calculation of a new point based on a starting point, distance, and bearing, is also changed when using latitude and longitude. Equations (5.5) and (5.6) replace equations (3.1) and 4.2 Equation (5.5) [25] is used to calculate the new latitude based on a starting point, distance, and bearing. Equation (5.6) [25] is used to calculate the new longitude using the newly-calculated latitude, starting point, distance, and bearing.

$$\varphi_2 = asin\left(\sin\varphi_1 \cos\frac{d}{R} + \cos\varphi_1 \sin\frac{d}{R}\cos\theta\right) \tag{5.5}$$

$$\lambda_2 = \lambda_1 + atan2\left(\frac{\sin\theta \sin\frac{d}{R}\cos\varphi_1}{\cos\frac{d}{R} - \sin\varphi_1 \sin\varphi_2}\right) \tag{5.6}$$

Equations (5.5) and (5.6), where φ is latitude in decimal degrees, λ is longitude in decimal degrees, d is the distance to the new point, $\boldsymbol{\theta}$ is the bearing to the new point, and R is the radius of the earth.

The distance and heading to the new position was provided by a NED vector. The NED vector to the new position was calculated using the velocity of the virtual leader, and the time since this calculation was last performed.

5.3 Simulator Environment

The PX4 system has tools to interface with Gazebo for simulations. MAVROS can also be configured to communicate directly with a Gazebo simulation of an aircraft.

The simulation environment was configured to match the flight test environment as closely as reasonably possible. For this project, the simulation was configured to duplicate the location where CfAR flight tests are performed. A Flight Dynamics Model (FDM) of the aircraft was created based on the Aircraft Platform, described in 5.4 below. The FDM includes mass, inertia, thrust, and torque of the aircraft. Custom firmware was created to operate the aircraft. The FDM and custom firmware also were used in the simulation.

The simulation ran the same ROS programs as the flight test system. Some settings in the Collective Consciousness and PX4 Interface nodes were changed when running the programs in the simulator, as described above in 5.1.2.

The ROS implementation of the MAUAS for simulation used distinct name spaces for each aircraft. Name spaces kept the ROS nodes of each aircraft from interfering with each other. Only collective nodes communicated with nodes in other name spaces.

5.4 Aircraft Platforms

For this project three aircraft platforms were built, based on an aircraft already in use at the CfAR lab. The existing aircraft was rebuilt, to improve its systems, and making it match with the two newly-made aircraft.

5.4.1 Aircraft Design and Construction

Flight Platform

The aircraft design was based on a commercial off-the-shelf quadcopter frame made by Tarot. The frame was 0.65m between diagonally opposite motor mounts. The aircraft used motors, propellers, and electronic speed controllers (ESCs) made by T-Motor. The combination of motors and propellers gave the aircraft approximately 73N of thrust, equivalent to 7.5kg. Each aircraft was powered by one 7,000 milliamp hour (mAh) 6-cell lithium battery. The batteries had a nominal voltage of 22.2 V. Each battery was able to support 15 minutes of flight, after which the battery was replaced.

Avionics

CUAV V5+ and X7 autopilots where selected as the primary candidates for use on the platform. These autopilots are similar, with only minor differences; the X7 has several advantages, a better processor, more RAM, runs at a higher frequency, and has higher quality IMU sensors. The original aircraft was already equipped with a V5+. CUAV autopilots were selected because they have powerful hardware and redundant sensors, and because these autopilots are already in us at CfAR in other systems, providing in-house institutional knowledge. Due to their relative similarities the V5+ autopilot on the original aircraft was used, with X7 autopilots on the new aircraft. Global position data were provided to the autopilots by a GPS, also made by CUAV.

Onboard Computer

Each aircraft was equipped with an onboard computer (OBC) to operate the algorithms for multi-agent coordination. A Raspberry Pi 4 (Pi4) was selected as the OBC. The Raspberry Pi has a quad core processor running at 1.5GHz, 8GB of Ram, Wi-Fi, Bluetooth, four USB ports, and 40 GPIO pins that can be configured based on need. The Pi4 OBC adds versatility to the system, enabling rapid modification and addition of code or hardware. The Pi4 is a relatively powerful computer given its size and mass. The pins can be configured to provide the board with 5 separate serial port interfaces, and Wi-Fi allows the Pi4 to be controlled remotely.

Inter-agent Radio

The aircraft were equipped with Microhard P900 frequency-hopping spread spectrum (FHSS) radios. These radios operate from 902-928MHz, support encryption, communicate at up to 230 kilobits per second, and have a range up to 60 km. These radios can be configured to work in a "mesh." When operating in mesh mode, each radio can send messages to a specific radio, or broadcast to all radios. Broadcast information is received by all radios within range of the sending radio. Messages to a specific radio are forwarded through the mesh to their destinations. The range allows aircraft to operate at large distances from each other and from support infrastructure while maintaining direct communication.

The P900 radio is connected directly to the OBC using a serial protocol. This allows the ROS system direct access to the inter-agent link. The OBC is configured to forward messages between the PX4 and the ground station.

Visual Identification Dome

Because the aircraft are intended to fly relatively close to each other (within 10 m), the aircraft were made visually distinct. Colored domes which covered the tops of the aircraft were made, each with a separate color. The domes also protect the electrical components of the aircraft from weather and debris.

5.4.2 System Validation

Navigation Position

Because the aircraft were operated near each other, the quality of their relative position data was examined. A Differential GPS (DGPS) is a system that provides positional corrections to GPS signals. A DGPS reference is placed at a fixed, known position, the reference tracks the errors present in the signals that it receives from the GPS satellites. The DGPS uses the errors to create a correction value, which is provided to a moving DGPS which uses the correction to improve its GPS accuracy to the centimeter level [29].

It was theorized that for DGPS to function effectively, all GPSs within the system must be affected similarly by sources of error. This was tested by measuring the drift in two systems operating near each other. The experimental setup attached two X7 autopilots to a rigid support structure, each with GPS receivers and a battery (Figure 17). Two different GPS receiver pairs were used in successive tests. Each GPS pair was mounted to the support structure with a 1.02m separation, each with its own autopilot. The test apparatus was placed so that GPSs were approximately East-West of each other. Autopilots logged the global position outputted by the EKF. This test was repeated with the second GPS pair. The duration of each test was two hours.

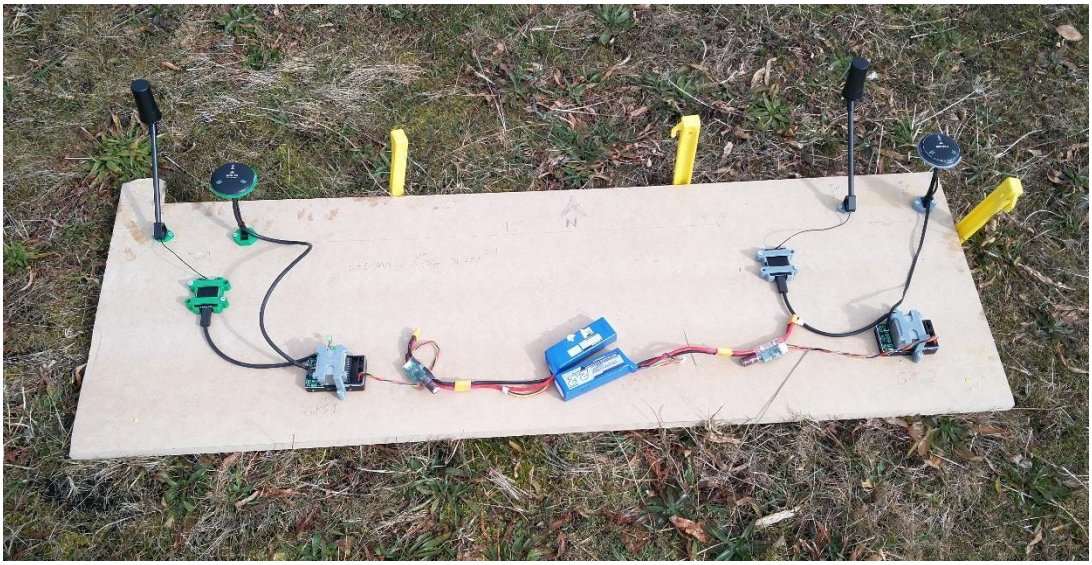

Figure 17: Experimental apparatus for measuring relative GPS drift. The two matched sets each include two GPS receivers (a self-contained disk and a two-piece rod antenna and receiver) connected to a CUAV X7 autopilots. Blue batteries are in the center.

Figure 18 shows the drift of the western system when using the two-piece GPS. This figure is centered on the mean position provided by the GPS. Data were collected at 10Hz. The standard deviation was 0.10 m.

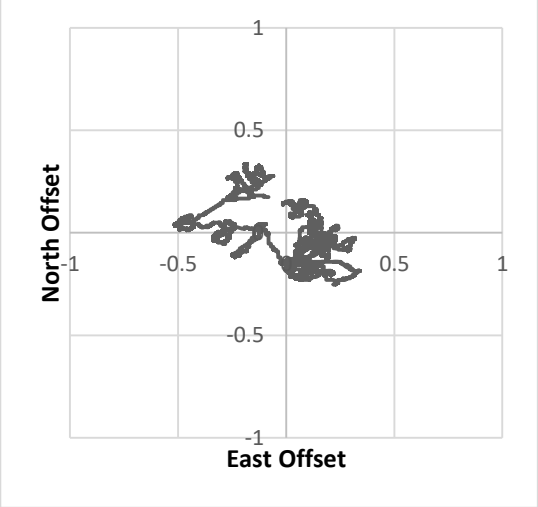

Figure 18: Western System Drift. Showing 2-dimensional drift (error) of individual position observations around the mean position. This represents the two-dimensional error with respect to its true location on earth. Scale units are meters.

Figure 19 shows the drift of the eastern system, with an error correction from the western system. This figure is centered on the position of the eastern GPS, as calculated by the western GPS. It was assumed that the eastern system error would match the western error at any given moment in time, as such the western system error was subtracted from the eastern system data as a correction factor. The mean offset between the two systems recorded by GPS was 1.06m, 0.044m higher than the measured distance. The standard deviation of the position drift of the eastern system using the correction was 0.07 m.

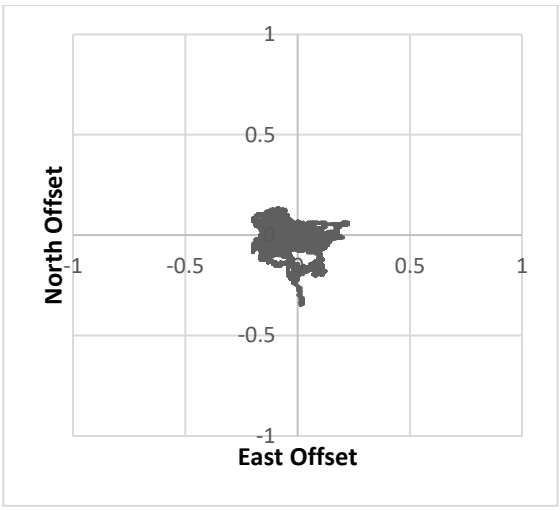

Figure 19: Time-synchronized. Relative error between western and eastern GPS systems. This represents the relative position error of the eastern GPS with respect to the western GPS. Units are in meters.

The standard deviation of the position drift of the eastern system with respect to the western system was 0.07m. This drift was smaller than the deviation of the western system by 0.03m, demonstrating that the drift of one GPS with respect to another GPS, using the same GPS receiver and antenna, is smaller than the drift of the GPS with respect to a global reference.

5.4.3 Commissioning

Calculating the mass, inertia, thrust, and torque of the aircraft is necessary to create a high-fidelity flight dynamics model (FDM) of the aircraft for use in simulation. The FDM allows for precise tuning of the PX4 autopilot of the aircraft. Thrust and torque of the motor propeller combination were already known from previous tests performed at CfAR.

Mass and Inertia

The inertia of one aircraft was found using a bi-fuller pendulum (BFP) Figure 20. The bi-fuller pendulum test suspended the aircraft from a pair of parallel lines. The lines were connected equally distant on either side of the aircraft's center of gravity. The aircraft was given an initial rotation about the axis parallel with the BFP lines. The rotational period of the aircraft was measured using an accelerometer. The separation of the support lines, the length of the lines, and the mass of the aircraft where used to calculate the moment of inertia of the aircraft about the rotational axis [30]. This was performed on all three axes of the aircraft: roll, pitch,

and yaw. A more detailed explanation of the theory and equations of bi-fuller pendulum testing is provided in [30].

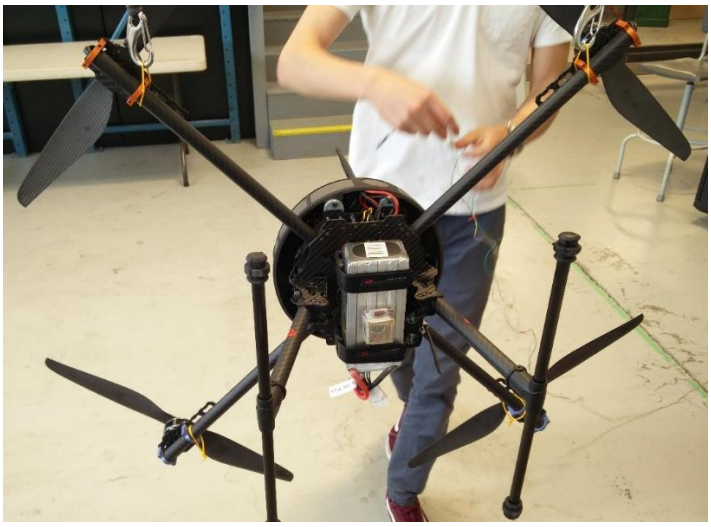

Figure 20: Bifilar Pendulum inertia test. Aircraft is supported from two of its motor mounts, using long cables, facing down, allowing it to rotate about the roll axis (axis pointing directly forward from the center of mass).

Aircraft masses are shown in Table 1, which includes the percent deviation of each aircraft from the mean. Because all three aircraft where built to be identical, and their masses diverged by less than 1.5%, it was determined that the mass and inertia of aircraft "Tarot 1" could be used for the FDM of all aircraft.

Table 1: Mass of the three quadcopter aircraft, mean mass is provided, as is each aircraft's percent deviation from the mean.

Aircraft	Mass [kg]	% deviation from Mean
Tarot 1	2.618	0.5%
Tarot 2	2.652	0.8%
Tarot 3	2.620	0.4%
Mean	2.630	

The moments of inertia for Tarot 1 are shown in Table 2. The simulator was configured to use these values. The PX4 autopilot tuning was also performed using these measurements.

Table 2: Inertia of Tarot 1, measured about all three primary axes of rotation. Inertias are used for FDM and the controllers of all aircraft.

Axis	Inertia [kgm^{2}]	Direction Reference
I_{xx}	0.42467	Right out of aircraft
I_{yy}	0.04337	Forward out of aircraft
I_{zz}	0.06327	Up out of aircraft

Vibration

Excessive vibration can cause a variety of problems for a UAV [31], including reduced flight time, increased wear on components, sensor errors, and state-estimation failures. State-estimation errors can lead to a flyaway, where an aircraft flies out of range of control signals.

A vibrational speed of less than 20 m/s is ideal for the V5 and X7. Vibration between 20m/s and 30m/s will result in decreased performance of the autopilot, and above 30 m/s will result in significant safety issues.

Vibration testing was performed on each aircraft. The aircraft were weighted down, Figure 21, and then all motors were throttled to full power for twenty seconds. The induced vibration was measured and logged on the autopilot.

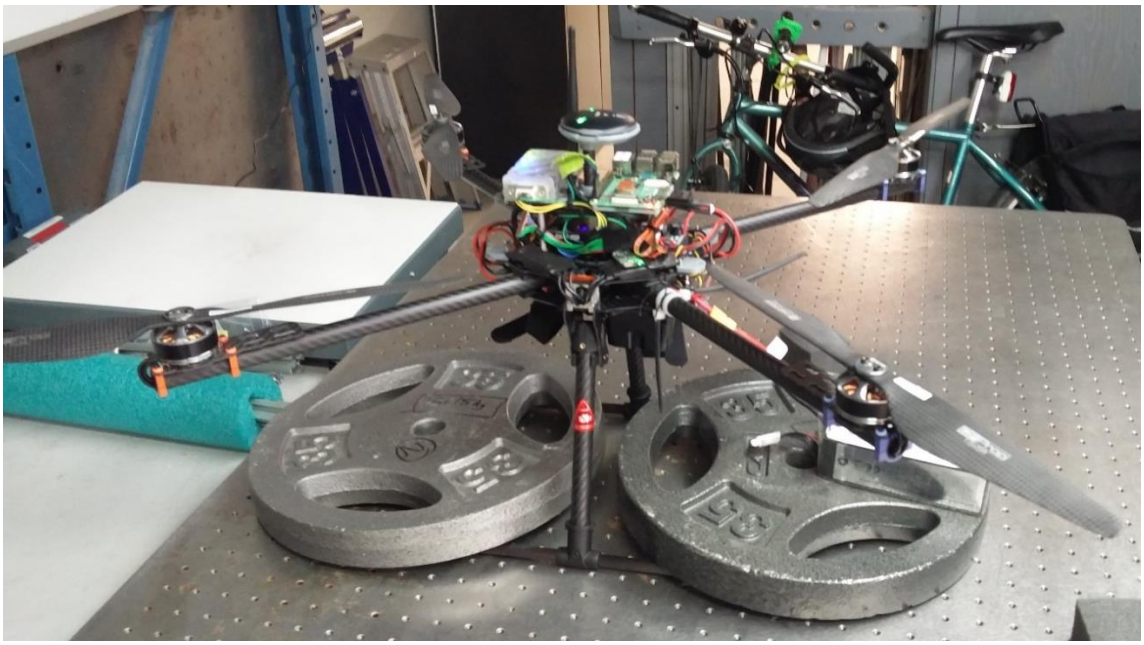

Figure 21: Vibration measurement test setup. All motors are set to full throttle while the autopilot records the systems vibration. Aircraft landing gear is weighted down with 70lbs (31kg) to prevent the aircraft from taking off during the test.

During the initial test only aircraft one passed the vibration test, with vibrations less than 0.02m/s, Figure 22, while the other aircraft both exceeded 0.03m/s. It was theorized that the vibration in the aircraft was due to imbalance in the propellers. The propellers were balanced by sanding small amounts of material from the propeller blades until the CG of the propeller was directly in line with the rotational axis. The vibration test was re-performed on the aircraft, and both aircraft passed.

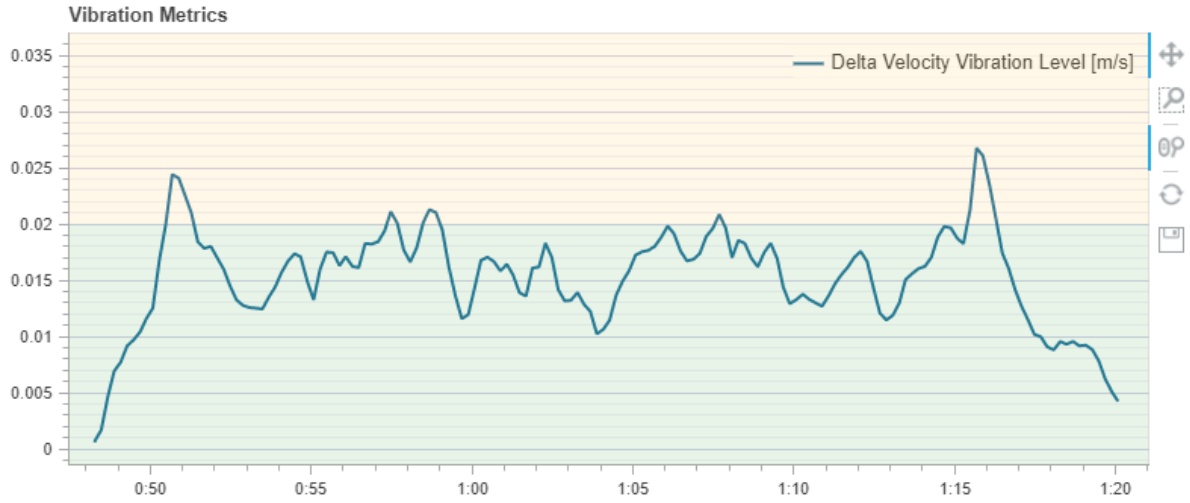

Figure 22: Autopilot Observed Vibration speed. This shows vibration velocity levels that will not cause state estimation issues for the PX4. This plot was created using [32], based on data logs collected during testing.

Radio

The time required to send and receive messages through the radio system will result in delays. Delay will affect the safety systems, flocking algorithms, and the ability for follower aircraft to follow the virtual leader. A test calculated the approximate delay between the systems.

Two systems were set up, using the Pi4 OBC and P900 radios. One system was configured as a sender, the other as a responder. The sender sent its current timestamp to the responder, which the responder returned. When the sender received the returned message it calculated the difference between the timestamp and its current time. The difference provided the round trip time for the message. The mean of the round trip time was 0.18s, therefore the one way delay was assumed to be 0.09s.

The test used messages of 16 bytes. Standard messages used by the coordination system are 168 bytes for the Big Sister messages and 49 bytes for the inter-agent messages. Given that the radio system operates at 28.8 kb/s it was assumed that the bandwidth would not cause additional delays.

5.5 CfAR Ground Station and Supporting Equipment

For testing proposes, each aircraft needed a ground station capable of controlling it, and a direct link to a human pilot. CfAR has a versatile mobile ground station capable of supporting many different aircraft and remote operations. Ground station equipment falls into three main categories: the CfAR van, which is equipped with computers, radios, and power supplies; several manual pilot consoles; and a variety of human-operated cameras for still and video photography.

5.5.1 CfAR Van

The CfAR van, Figure 23, contains the tools and equipment necessary for flight-testing. The most notable feature is a pair of built-in desktop computers. Each computer can act as a ground station, commanding one or more autopilot aircraft. These computers can also be used to access and control payload systems on one or more aircraft. In addition, they can be connected to other computers at the CfAR office, allowing testing data to be easily transferred.

Figure 23: CfAR Ground Station Van. In use during a flight test (top) and, showing interior with two desktop computers (bottom).

The van also contains multiple radios systems. These include a P900 FHSS mesh radio, like the radios installed on the flight test aircraft. The van also contains a radio intercom system, allowing the flight crew to use hands-free communication among members of the team.

The van is equipped with both battery and a gasoline-powered generator, providing onboard power to the computers and radios. It also allows aircraft batteries to be recharged on site.

The CfAR van is equipped with a Wi-Fi router with a cellular data link. This allows all electronic equipment present during flight tests to access the internet.

5.5.2 Pilot Consoles

For direct control of aircraft, CfAR uses hand-held UAV pilot consoles (Figure 24). These consoles have two joysticks which control the movement of the aircraft. They also have other controls (e.g. switches) that can be configured to control parts of the aircraft's configuration or actions, such as autopilot mode, flaps, or landing gear.

Figure 24: Jumper T16 UAV Pilot Console. Control sticks are used to control aircraft movement, while switches and knobs can be configured based on requirements.

Each hand-held console can control only one aircraft at a time, therefore three separate consoles were required to operate the three aircraft used for this research project.

5.5.3 Media Equipment

CfAR has digital photographic equipment used to record flight testing operations. The main cameras are a digital single-lens reflex (DSLR), and a 4K video camera. The DSLR camera is used to document flight testing operations. The 4K video camera is used to record the movements of the aircraft for after-flight review. A variety of other cameras are used for smaller miscellaneous tasks.

Chapter 6: Multi-Agent System Simulation and Validation

Simulations of the multi-agent formation system were performed using Gazebo simulation software. Almost all aspects of the system were tested in the simulation environment. The initial simulations to validate the interaction between the ROS systems and the PX4 used a generic aircraft model, a quadcopter called Iris. Iris is a default aircraft in the PX4 Gazebo simulation.

Simulation to validate the research platform was performed using the FDM that was created in parts 6.3 and 6.4. Simulated formations were initially configured to use 5m spacing. During the flight test the aircraft repeatedly breached the safety distance, so the spacing between aircraft was increased to 10m. The simulations were re-performed at this spacing. Further discussion of the spacing change is provided in section 6.3.1 Aircraft Spacing.

6.1 Simulation Objectives

Simulations were intended to represent the real world, presenting situations and requirements that would be expected during actual flight. Successful simulations of the multi-agent system would allow system development with minimal need for flight testing validation.

Objective 1: Validate the Multi-agent formation control system, verifying that it was able to maintain all aircraft in the selected formation while moving along a preset flight path.

Objective 2: Validate the Research Platform for flight testing, verifying that the PX4 configuration of aircraft was ready for flight operations and able to control aircraft without difficulty, while operating within the multi agent system.

6.2 Simulation Environment and Configuration

Simulated aircraft and actual aircraft differ primarily in how they communicate among themselves. Also, the FDM will not be completely accurate to all of the aircraft, although it should be close.

6.2.1 Simulated Communication

The mesh radio system used by the flight test aircraft was not included in the simulation. In simulation, aircraft communicate among themselves using ROS, while their link to the ground station runs through Gazebo. This would result in each follower aircraft receiving coordination information from Big Sister as though it were coming from within its own system.

6.2.2 Aircraft, and ROS Simulation Configuration

In simulation aircraft battery power was constant and the ROS system was configured to arm the aircraft, initiating takeoff. This configuration reduced the operator workload required to run simulations, as it allowed the aircraft to initiate a test flight as soon as the simulator was loaded, without needing to wait for an operator to command each aircraft in the fleet to takeoff.

6.3 Simulation Tests

6.3.1 Aircraft Spacing

The system was initially configured to maintain a distance of 5m between aircraft. During flight testing, however, it was necessary to operate the aircraft with a spacing of 10m. Simulations were repeated with the increased formation spacing.

When a formation of aircraft executes a coordinated turn, the target speed of each individual aircraft must change, based on its position in the formation and the rotational speed of the formation. The increased inter-aircraft spacing from 5m to 10m increased the relative difference of the aircraft speed with respect to the Virtual Leader. The aircraft on the outside of the turn was required to fly at a speed of 9m/s, however all aircraft were limited to a maximum speed of

4m/s for safety. Therefore the aircraft on the outside of the turn could not maintain its position in formation, returning to its position after the turn. This position error did not create a safety issues for the system.

6.3.2 Test Configurations

One simulation with four components was performed. The first two components ran the Big Sister in follow the leader mode, meaning that the Virtual Leader's position used the position of the lead aircraft. The lead aircraft used a PX4 autopilot mission to guide its movements. The last two components ran the Big Sister in Virtual Leader mode, where all aircraft were provided locations around the virtual leader. The Virtual Leader used the same mission waypoints as the PX4 autopilot for these tests.

In each leader configuration one component was straight and level flight, and the other was a coordinated turn. The overall mission was a square flight pattern with 90 degree turns to the right. Each leg of the flight path was approximately 200m in length.

Different leader configurations used different formations. When led by the PX4 autopilot, the follower aircraft took positions 10m on either side of the leader (Figure 25) forming a horizontal line. When led by the Virtual Leader, all three aircraft acted as followers, forming a delta formation around the Virtual Leader. The aircraft were all 10m from the Virtual Leader (Figure 26).

Figure 25: Simulated Line Formation using FDM of experimental aircraft in Gazebo. The center aircraft is leading the horizontal line formation towards image left.

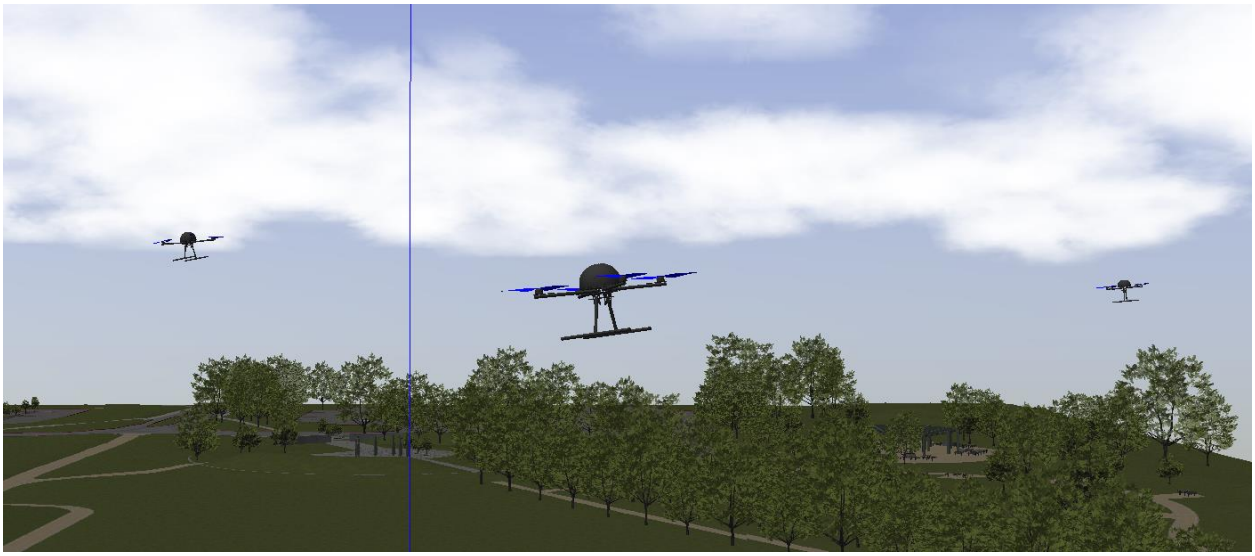

Figure 26: Simulated Delta Formation using FDM of experimental aircraft in Gazebo. The formation is using the virtual leader, the aircraft on the far left of the image is in the front, the near aircraft is on the left, and the aircraft in the back right of the image is on the right side of the formation. The virtual leader is located halfway between the two aircraft on the right of the image. Vertical blue line is an artifact of the simulation environment. The virtual leader is flying towards image left.

During flight simulations the ROS nodes on each aircraft logged several types of data. The ROS system evaluated the position error of the aircraft with the most up-to-date information available to that aircraft. The ROS system also calculated the distance between it and its nearest neighbor in the formation. The ROS nodes also logged the aircraft's position with respect to a global timestamp.

6.3.3 Data Analysis

MATLAB was used to analyze the data collected by the ROS nodes. Data were divided into two main categories: onboard and timestamp. Onboard data refers to data calculated by the ROS nodes during simulation, this included errors caused by delays in system communication. Timestamp data were calculated in MATLAB using the ROS timestamps to remove errors caused by internal communication delays.

Formation Position Errors

Position errors were the primary data of interest. The ROS system calculated onboard position errors using equation (5.2). This calculated the distance from each follower aircraft to

its target position with respect to the formation leader. The target position was calculated based on the most up-to-date information each aircraft had received from the leader. Onboard position error was calculated in real time during flight.

Timestamp position error was calculated after the simulation was completed using equation (5.2), similar to the onboard position error, however this target position was calculated using the leader's position with the same timestamp. Timestamp position error removes the communication delay. Both the onboard and timestamp position errors represent the accuracy of each aircraft with respect to its target point.

The position errors of the system when flying in straight and level flight while following the virtual leader are shown in Figure 27 and Figure 28. In both figures it can be seen that the aircraft performance is comparable across all aircraft. It can also be seen that the timestamp error is significantly higher than the onboard error.

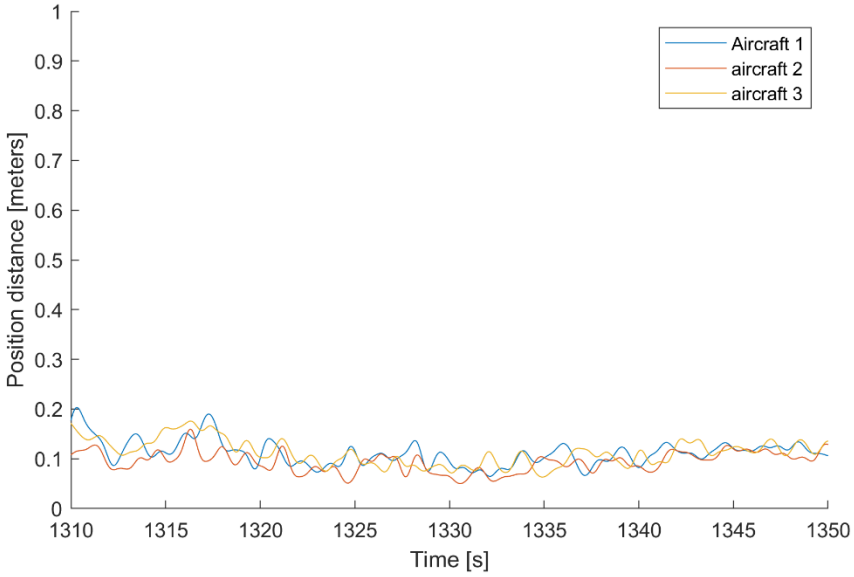

Figure 27: Straight and level formation flight, following Virtual Leader, showing onboard observed position error of all three aircraft.

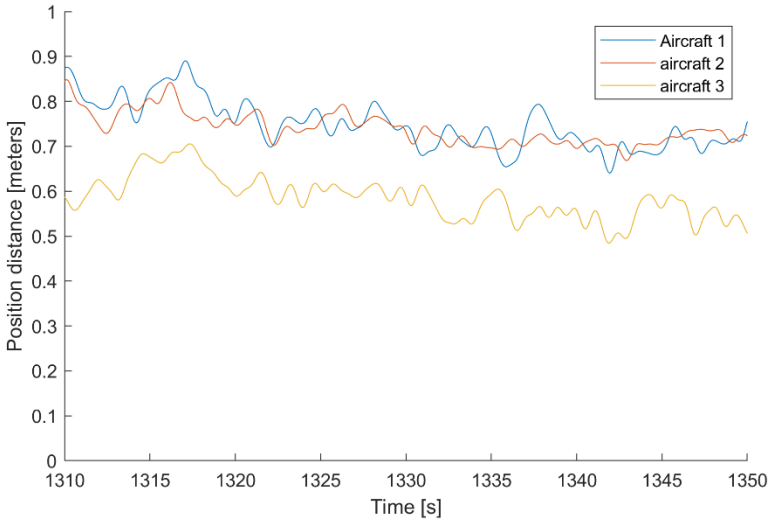

Figure 28: Straight and level flight, following virtual leader, showing timestamp correlated observed position error of all three aircraft.

The difference between the position errors shown in Figure 27 and Figure 28 was caused by delays within the system. Every ROS message takes time to go from the sending node to the receiving node. Because of this, there is a difference between where the aircraft thinks the Virtual Leader is and where it actually is. Equation (6.1) was used to calculate the delay, using the difference in position errors of 0.63m and the cruising speed of 2.7m/s. The delay was 0.23s. This communication delay reflected the difference in the time it took for the virtual leader information to travel from Big Sister to the Boid Brain. This means that the aircraft was tracking a position 0.23 seconds behind where it should have been.

$$\Delta t = \frac{\Delta P_{error}}{|V|} \tag{6.1}$$

Equation (6.1), where $\boldsymbol{\Delta P_{error}}$ is the difference between the observed error and the timestamp error, V is the velocity the aircraft was traveling, and $\boldsymbol{\Delta t}$ is the time delay.

Aircraft Separation

Aircraft spacing was calculated both onboard during simulation, and after simulation using the timestamps. The onboard evaluation is shown in Figure 29, and the timestamp evaluation is shown in Figure 30. The aircraft were in a delta formation, with follower aircraft 10m forward of and 10m to each side of the virtual leader. Aircraft on opposite sides of the formation should have 20m spacing. The forward aircraft should be 14.1m from each of the other aircraft. The

spacing data used were drawn from a section of straight and level flight, followed by a coordinated turn at time 1230s, and returning to straight and level flight.

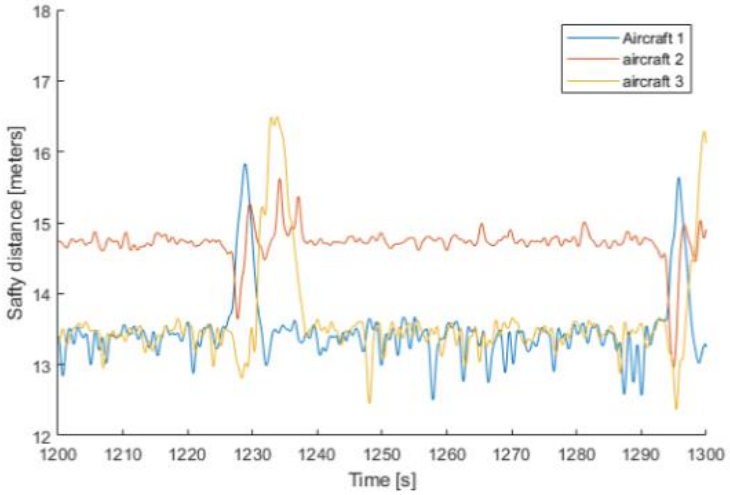

Figure 29: Straight and level flight with coordinated turn, showing the aircraft spacing as observed by the aircraft.

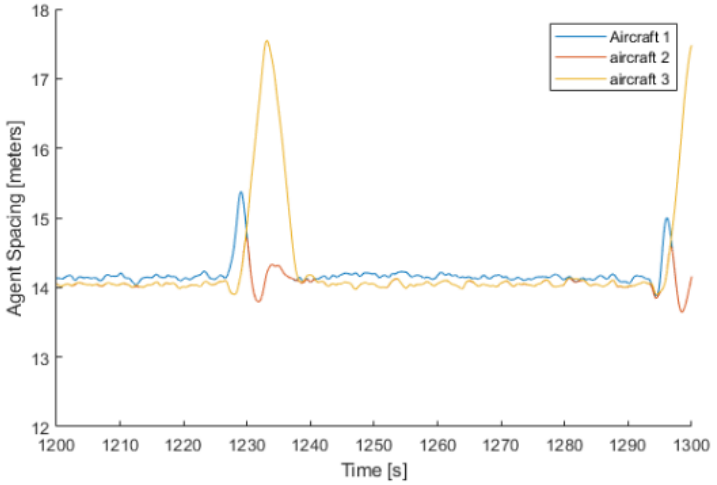

Figure 30: Straight and level flight with coordinated turn. Showing the aircraft spacing, using timestamp correlated data.

Because the aircraft were in a delta formation the aircraft on the sides of the formation both reported the distance from them to the aircraft in front, while the aircraft in front reported the distance to the closer of the other two. Distances measured by the front aircraft (aircraft 2) were larger than the distances measured by the other two. The onboard observation of the distance was 0.6m different from the timestamp calculation for all aircraft. This corresponded to the delay that was calculated previously. As noted above, each aircraft observes the position of each

other aircraft where it was 0.23s, or 0.6m, earlier. This means that the aircraft in front observed the other aircraft falling behind their positions and being further away than they were, and the other two aircraft observed the front aircraft falling behind its position bringing it closer to them than it was.

The timestamp data in Figure 30 shows that during the flight the three aircraft maintained the correct spacing among themselves when flying in straight and level flight, with minimal error, correctly meeting the desired 14.1m spacing. During the coordinated turn the aircraft in the front and the aircraft on the inside of the turn maintained their relative positions with only slight errors. The aircraft on the outside of the turn however, had markedly increased spacing during the turn.

6.3.4 Simulation Results

Simulation results focus primarily on position error. Table 3 shows onboard position error and timestamp error for each simulation configuration. For all data points the largest error observed by each aircraft is recorded.

Table 3: Simulated Flight Test position error data. Showing onboard and timestamp corrected errors. The Comparison is the difference between onboard and timestamp errors. Tarot 2 is runs the Big Sister. Tarot 1 is on the inside of the coordinated turn, and Tarot 3 is on the outside of the turn. All measurements are in meters.

System	Onboard Error [m]	Timestamp Error [m]	Comparison [m]
Follow PX4, Straight & Level Flight			
Tarot 1	0.1	0.8	0.7
Tarot 2	0.7	0.15	0.55
Tarot 3	0.1	0.7	0.6
Virtual Leader, Straight & Level Flight			
Tarot 1	0.1	0.8	0.7
Tarot 2	0.1	0.8	0.7
Tarot 3	0.1	0.6	0.5
Follow PX4, Coordinated Turn			
Tarot 1	9.5	10.5	1
Tarot 2	0.6	0.1	0.5
Tarot 3	12.4	13.3	0.9
Virtual Leader, Coordinated Turn			
Tarot 1	2.6	2.9	0.3
Tarot 2	4.5	5.5	1
Tarot 3	7.2	8.1	0.9

In Table 3, Tarot 2 is the leader of the PX4 formation. Tarots 1 and 3 are followers. When following the PX4 autopilot of Tarot 2, the three aircraft formed a straight line perpendicular to the direction of flight, with Tarot 1 on the right, Tarot 2 in the center, and Tarot 3 on the left. When following the virtual leader all three aircraft act as followers in a delta formation: Tarot 1 on the right, Tarot 2 in front, and Tarot 3 on the left of the virtual leader. Coordinated turns always turned to the right, Tarot 1 was on the inside of the turn and Tarot 3 was on the outside of the turn.

Straight and level flight

During straight and level flight, position errors of follower aircraft (Tarots 1 and 3 during PX4 formation, all three Tarots during Virtual Leader formation) were 0.1m across all tests. In comparison, the timestamp-correlated data shows that the followers are 0.8m from their targets. Table 3 also compares position errors for each of these sets of tests, showing that the difference in position errors of follower aircraft in the simulation were consistently 0.7m when operating in straight and level flight. This results from the time delay of 0.23s, as described above.

When an aircraft acts as the leader using the PX4 for navigation, its recorded errors are measurably different. The error calculated using the timestamp on the leader drops to 0.15m, which corresponds to a delay of 0.05s. This comes from a combination of internal delays and lack of synchronization between the Big Sister and Boid Brain nodes. The error calculated onboard increases to 0.7m. This change in which method records the higher error is because the internal time delays, the aircraft is always responding to where the leader was, but because this is now it, it thinks that it is ahead of the leader.

Coordinated turn

In a coordinated turn an aircraft faces the direction of its velocity throughout the turn. In this work, a coordinated turn means the formation faced the direction of the formation velocity throughout the turn. Most aircraft, including fixed wing aircraft, use coordinated turns to reduce drag.

During coordinated turn simulations, aircraft in formation performed less precisely than during straight and level flight. The PX4 leader was not significantly affected during turns. The followers had position errors of 10.5m and 13.3m for the inside and outside of the turn,

respectively. This was due to a combination of a rapid pivot made by the leader's PX4 autopilot, and the internal time delays. This rapid pivot required follower aircraft to fly in excess of the speed limit placed on them, meaning they could not hold their positions. This error was slightly mitigated on the inside of the turn because the forward speed of the formation and the rotational effect counter each other.

During the Virtual Leader test of a coordinated turn, position errors of the followers were noticeably smaller than during the PX4 test. Aircraft 2, running Big Sister, had larger position errors, because it was no longer the leader. The position error was greater for aircraft further from the point of rotation, with the inside of the turn performing best, while the outside of the turn was worst. This was caused by a combination of the formation rotating too quickly for the aircraft to keep up, and from the time delay.

The timestamp evaluation of the virtual leader system during the coordinated turn showed that the internal delays caused different levels of error in different aircraft. The flight speeds of Tarot 2 and Tarot 3 were above the Virtual Leaders speed of 2.7m/s, limited to the maximum of 4.0m/s. The flight speed of Tarot 1 was 2.5m/s. The new flight speeds and time errors induced by the internal time delay for each aircraft are shown in Table 4. The time error of Tarot 1 was calculated to be 0.12s. The time delay for Tarots 2 and 3 were approximately 0.25s, the same delay that was observed during previous tests. The time delay within the system was not affected by the turn. The reduced error found in Tarot 1 is because the movement of its position in formation was easier for the controller in the Boid Brain to follow.

Table 4: Target speed of each aircraft during a coordinated turn, the time error that would result from the listed speed and the difference in the aircraft's maximum position error are shown.

Aircraft	Target Speed [m/s]	Position Error [m]	Time Error [s]
Tarot 1	2.54	0.3	0.12
Tarot 2	4.0	1.0	0.25
Tarot 3	4.0	0.9	0.23

6.1 Discussion of Simulation

From the simulation it was determined that the aircraft have some noticeable time delays that are internal to the ROS system. Despite these delays, aircraft were able to perform with position

error of less than a meter during straight and level flight. When the aircraft formation was executing a coordinated turn, position errors were higher. The ROS system on the OBC of each aircraft observed the position errors and used the PD control loop to determine the guidance needed to fly in formation. The ROS systems inability to fix the observed error shows that the PD control loop was inadequate to correctly guide the aircraft.

The magnitude of separation between aircraft changed during the coordinated turn shown in Figure 30. These data show that position errors did not overly affect the relative spacing of the aircraft, shown by the aircraft on the inside position and the lead position maintaining correct spacing, with a little more than a 1m relative error. The aircraft on the outside position of the turn was forced out of relative position by just over 3m; part of this error was due to the speed limitation preventing it from flying the required speed.

Time delays within the system caused position measurement errors of less than 1m, which did not compromise the safety system. The safety parameter that the aircraft used was already configured to accommodate a delay greater than the internal delay of 0.23s found in the system.

The overall result of simulation testing demonstrated that there are improvements possible within the system. The movement of the leader directly affects the difficulty of the paths the followers try to use. The PD control system in the Boid Brain was a source of errors; it was unable to accurately follow its target position during turns. The internal delays with the ROS system created a small but noticeable offset between data the OBC receives and the true state at that time.

Chapter 7: Flight Test

7.1 Assumptions

Two important assumptions were made prior to flight testing. It was assumed that the PX4 and internal delays would not cause any position errors above those found in the simulations. It also was assumed that the radios would not be a significant source of delays above the internal delays. This assumption was based on the radio delay of 0.09s (Section 6). A minimum safe spacing was calculated for the aircraft, using worst case errors. Worst cases included long internal delays, several dropped radio messages in a row, and compounding position drift in the PX4s, bringing the aircraft closer together. Additional distance was added between aircraft so that they would correct spacing before human safety pilots would override control.

7.2 Flight Test Operation

Flight testing was performed at a farm field in Saanich BC, Figure 31. This field is used regularly for flight testing by CfAR. The red box shows the limits of the area used for flying aircraft during flight testing. The blue circle marks the location of the ground station.

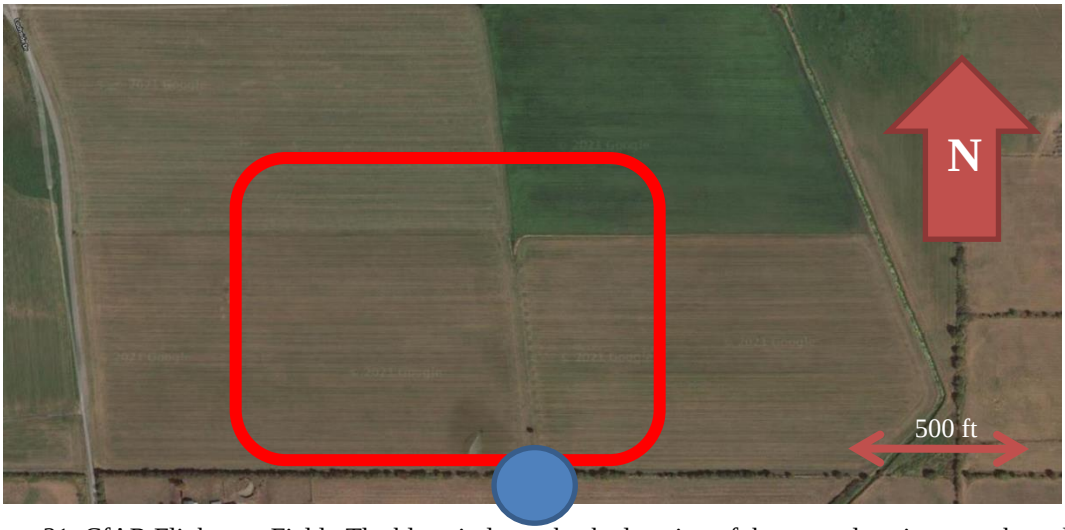

Figure 31: CfAR Flight-test Field. The blue circle marks the location of the ground station van, the red box shows the limits of the area used for flight testing.
Source: Adapted from [Google Maps]

The flight testing setup consisted of the CfAR van as the ground station and a crew of trained UAV operators. Three batteries were on hand for each aircraft, allowing the full system to be flown three times without needing to recharge batteries. The flight crew included a manual UAV safety pilot for each of the three aircraft, who for safety could override aircraft control. One UAV ground station operator monitored all the telemetry of all aircraft. Other ground crew included the flight test director, one range safety officer, and photographers and videographers.

7.3 System Validation Flights

7.3.1 Aircraft Shakedown and Multi-Agent Safety Validation

Each aircraft was flown individually in a shakedown flight. Shakedown flights served three primary purposes. The control system of the PX4 was validated on the aircraft, and some PID (proportional, integral, and derivative) tuning of the PX4 was performed. The manual safety pilots familiarized themselves with the aircraft and controls. The aircraft were found to be in full working order.

The safety distance and safety evasion were tested using two aircraft. One aircraft hovered using the Boid Brain, while another aircraft was manually flown towards it. When the manually-flown aircraft passed within the safety distance of the hovering aircraft, the hovering aircraft flew away as programed. When the manually-flown aircraft moved away, the hovering aircraft returned to its original position.

7.3.2 Formation Validation and Adjustment

The initial formation testing was performed with two aircraft. The first aircraft was set to hover using the PX4 to control its position. The second aircraft was manually flown to its approximate location in formation (5m from the first aircraft) and switched into Boid Mode. The second aircraft was unable to hold its position without a small amount of drift. When it drifted towards the first aircraft it breached the safety perimeter, triggering it to take evasive action. When returning to its target position the second aircraft would overshoot its target point by a small amount, and would re-trigger its evasion action. Because of this repeated evasion and

return the formation spacing was doubled, from the original 5m to 10m. The increased spacing corrected the problem.

The increased formation spacing caused changes in the performance of the system; these changes were the same as those described in section 6.3.1 of Simulation.

During this phase of flight testing, the OBC on Tarot 2 developed voltage problems during flight making it unreliable and risking a communication failure. This aircraft was grounded for the safety of the equipment and flight personal. This problem likely was caused by a bad connector, which could not withstand the vibration of the aircraft in flight. All other aircraft were checked for this problem, to ensure it would not interfere further with flight testing.

7.4 Testing of Formation Flight

7.4.1 Flight tests and testing procedure

The first flight test was a hover test of a single aircraft in Boid Mode, using a static point as the virtual leader. The aircraft was set to hold a constant position with respect to that point. A fixed point was used so that no internal or radio delays would affect the aircraft's estimation of the position of the virtual leader.

The second test used one aircraft in Boid mode following another aircraft which was flown manually in a box pattern, while maintaining a constant heading. The constant heading reduced complexity for the human pilots in case they needed to take full control of the system.

The third test used the Virtual Leader with two following aircraft in Boid Mode. The virtual leader followed the same path that was used during simulation, executing the same turns. This was done to allow the multi agent control system to be tested, and to provide a comparison between the simulator and the aircraft.

During testing, each aircraft recorded the same forms of data collected during the simulations. These data include the onboard evaluation of position error and spacing to the nearest aircraft. The positions were recorded with timestamps, allowing for analysis after flight.

7.4.2 Experimental Results

Data from flight tests were analyzed the same as data from simulations. Results are presented in Table 5.

Table 5: Flight testing position error data. Onboard and timestamp-correlated errors are shown, the difference between onboard and timestamp error is also shown. The Big Sister was run onboard Tarot 1 for all multi aircraft tests.

System	Onboard Error [m]	Timestamp Error [m]	Comparison [m]
Hover Test			
Tarot 1	0.1	0.1	0.0
Follow Human, Straight & Level Flight			
Tarot 1	0.0	0.0	0.0
Tarot 3	0.5	0.7	0.2
Follow Human, Corner			
Tarot 1	0.2	0.1	0.1
Tarot 3	2.1	3.0	0.9
Virtual Leader, Straight & Level Flight			
Tarot 1	0.2	0.6	0.4
Tarot 3	0.3	1.2	0.9
Virtual Leader, Coordinated Turn			
Tarot 1	4.5	4.4	0.1
Tarot 3	8.0	9.2	1.2

Hover Test

The hover test measured how well the Boid Brain maintained a static position. It maintained a position error of 0.1m. This was the same for both the observed error and the timestamp error.

Follow Human

The follow human test measured how well the Boid Brain aircraft (Tarot 3) followed a manually-piloted aircraft (Tarot 1). The manually-piloted aircraft flew a box pattern where it started from each corner at a hover, accelerated to the next corner, then returned to a hover. The box was flown twice during this test. The bearing of the manually-flown aircraft was constant, facing north. The Boid Brain, also with a north bearing, maintained a position 10m east of the manually-flown aircraft, but with noticeable position errors.

During straight and level flight the Boid Brain observed a position error of 0.5m, with a timestamp error of 0.7m. During the corners the observed a position error was 2.1m, with a timestamp error of 3m. The variability of the manual flight speed during this test made comparison with the other tests invalid.

Virtual Leader

Tarot 1 ran the Big Sister program, which simulated the Virtual Leader. The Virtual Leader flight test is comparable to the Virtual Leader simulation, because the simulations were re-done using the same formation spacing of 10m. The behavior of the Virtual Leader was identical in both tests.

The internal delay of the system was calculated using Tarot 1's position error. The observed error of Tarot 1 was 0.2m, and timestamp error was 0.6m. The difference between the observed and timestamp errors, 0.4m, was used with the flight speed of 2.7 m/s to calculate that the internal delay was approximately 0.15s Equation (6.1).

The radio delay in the system was calculated using the position errors of Tarot 3 and the internal time delay. Tarot 3 observed its position error as 0.3m, while its timestamp error was 0.9m. This increase in observed error shows that the control systems performance is degraded when the virtual leader information is passed through a radio. The position error difference of Tarot 3 is 0.6m, corresponding to a delay of 0.33s. This delay includes the internal time delay of the onboard computer, which when removed reveals a radio induced time delay of 0.18s.

The coordinated turn position errors of Tarot 3 agree with the time delay calculated above. The position error difference of Tarot 3 was 1.2m. Because Tarot 3 was on the outside of the turn its formation position moved faster than the 4.0m/s maximum speed allowed. The time delay was 0.3s, similar to the 0.33s time delay found during straight and level flight.

During the flight test neither aircraft deviated markedly from the intended spacing. The timestamp evaluated spacing between the aircraft is shown in Figure 32, showing the spacing during a large portion of the flight. Most of the flight was straight and level, with two coordinated turns at time 1415s and 1480s. The aircraft maintained the correct spacing of 20m during most of the operation, while spacing discrepancies during the turns were less than 1 meter.

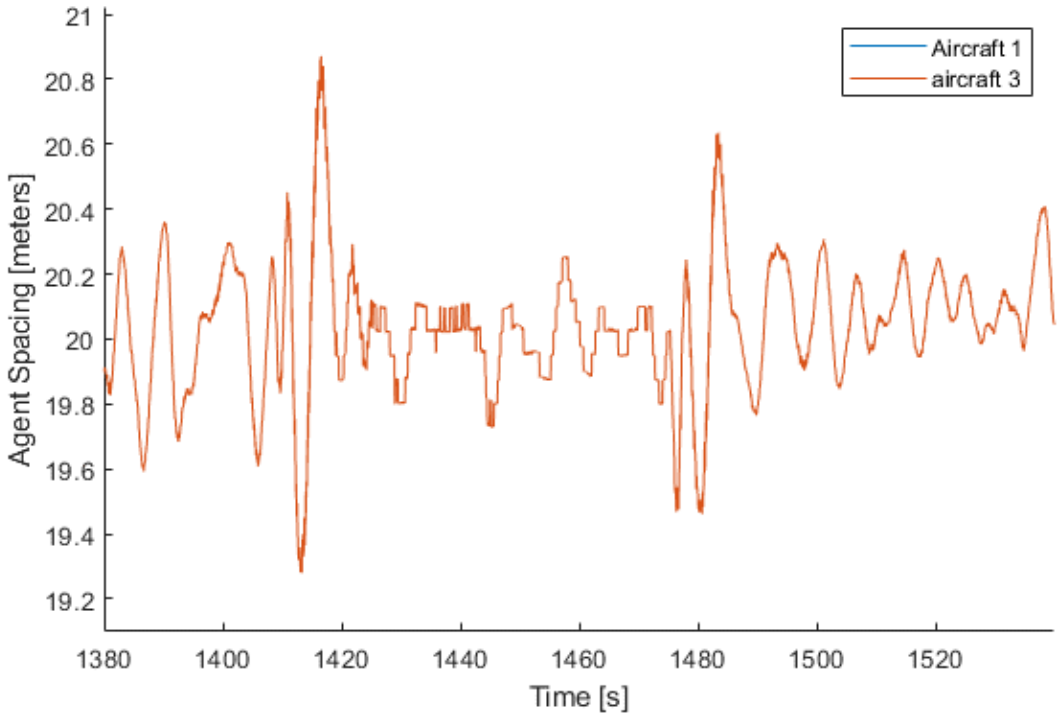

Figure 32: Aircraft spacing, calculated with timestamp data, during Virtual Leader flight test. The two lines are identical, subsequently the blue line is hidden.

7.5 Discussion of Flight Test Data

Flight testing results clearly show the delays present in the system. Sources of these delays were internal to the aircraft, as well as from inter-agent communication. Delays internal to each aircraft's systems were 0.15s, resulting in consistent position errors. This delay was created by the messaging time between the ROS nodes, and the time for MAVROS and the PX4 to communicate. The total delay limited the performance of each aircraft.

The radio induced delay during transmission of messages from the aircraft running the Big Sister to follower aircraft affected the follower's performance. This radio delay of 0.18s contributed to the followers' overall position errors. This delay also affected each aircraft's knowledge of other aircrafts' positions. The radio delay between aircraft increased the minimum safety distance required.

The observed position errors of each aircraft provided the information needed to evaluate the performance of the PD control loop. During straight and level flight, the observed errors were less than 0.3m. This showed that the PD control loop was able to effectively guide aircraft

during steady state flight. During the coordinated turn, however, the observed errors were 4.5m or greater. This showed that the PD control loop was not well suited to guide the aircraft in non-steady state conditions.

While the above position errors were with respect to the virtual leader, relative to each other the two aircraft were able to maintain the correct spacing, with an error of less than 0.9m.

7.5.1 note Comparison of Simulation vs Flight

The results of the flight test must be compared to the simulation to evaluate the performance of the multi-agent coordination system, and to evaluate how well the simulator models the flight test platform. Internal delays onboard each aircraft (e.g., ROS, PX4) were shorter on the flight test aircraft than in the simulation. The delay onboard the flight test aircraft was 0.15s, compared to 0.23s in the simulation. The radio communication added a 0.18s delay to all aircraft communications, which was not modeled in the simulation.

In straight and level flight, the simulator accurately modeled flight test aircraft performance. In coordinated turns, position errors were smaller in the simulator than with the flight test aircraft. The higher performance level of the simulator was not explained by the lack of radio delays.

In both the simulator and the flight test, the PD controller was shown to have reasonable performance during straight and level flight, whereas it was ineffective during coordinated turns. These problems were compounded on the flight test aircraft by the addition of radio delays.

In both the simulator and in the flight test aircraft, even when the aircraft had significant position errors with respect to the virtual leader, they were able to maintain the correct spacing among themselves, showing that both systems were able to maintain system safety, even during coordinated turns.

Chapter 8: Conclusions and Future Work

This work has demonstrated the viability of an onboard, multi-agent, formation-control, system based on a Virtual Leader. It also explored the strengths and weaknesses of the system. This work included the creation of a research platform that can be used for continuing this work, or for exploring other multi-agent research. Project conclusions are presented below with respect to the objectives presented in section 2.1.

The coordination system created for this project was able to run onboard the aircraft, without any ground station support. The aircraft coordination system was able to maintain aircrafts' positions with respect to the virtual leader with an error of less than 0.7m when traveling in straight and level flight. The position error with respect to the virtual leader was larger ($\leq$9.2m) during a coordinated turn; however the error between follower aircraft was less than 1m.

The system can be improved. The PD control system of the Boid Brain was shown to be ineffective in both simulation and flight testing. Although it performed well during straight and level flight, the PD loop did not accurately guide the aircraft during the coordinated turns. It was also shown that other aspects of the control, such as the safety speed limit, were detrimental to the system performance, preventing the aircraft from flying at the speeds that were required by the formation. Additionally, delays both internal to the aircraft system and from radio communication added to the position error.

The emergency evasion system created for this project was effective. The time delay between the aircraft affected the minimum safe spacing of the aircraft. Longer communication delays require larger minimum spacing. Creating a system to adjust the aircraft trajectories to improve inter-agent interaction was not completed. Although flocking was initially developed to address this, and some simulation was performed, final simulation and flight testing were not performed due to time constraints.

The Big Sister coordination system developed here is a centralized system which can be run onboard any aircraft in the system. However, Big Sister is not fully agent-independent because if Big Sister stops functioning, it does not automatically start on another aircraft, requiring a

human operator to manually restart it. The emergency evasion system created for this work was decentralized, therefore fully agent-independent.

A multi-agent research development platform was created. The platform included a simulation environment, and several flight test aircraft.

The experiments showed that the aircraft running the virtual leader performed very similarly in the simulation and in the flight test. There were slight performance differences, but they had no meaningful effects. The assumption that radio delay would not affect performance was incorrect. The radio contributed to delays in the system. Radio delay would need to be included in the simulation to provide a more accurate representation of the flight test system.

8.1 Future Work

This work can be continued in a number of different directions. The multi agent system developed here could be used to support other research work.

8.1.1 Additional Testing of the Current System

Not all of the objectives were fully completed; additional work would be worthwhile and could fully explore the original objectives of this work.

8.1.2 GPS testing

Further investigation of GPS drift would be helpful for close proximity flight. There are existing GPS systems that use a reference from a moving GPS to improve the relative position accuracy of nearby GPSs. Further investigation of this could allow for closer proximity aircraft operation based on GPS sensor information.

8.1.3 Improve System Code and Control Loop

The code that runs the coordination system, although capable of demonstrating the proof of concept to a reasonable level of success, could be improved to decrease delays within the system, improving performance.

The PD control loop should be replaced with a PID control loop to improve the aircrafts' performance. Tuning of the control loop is also required whether or not it is replaced.

A positon prediction system could be added to reduce the effect of radio delays. This would require including timestamps on all inter-agent messages, including from the virtual leader. This could be used to determine how far, and in what direction, the aircraft (or leader) had moved since the information was collected.

8.1.4 Integration of Non-Cooperative-Agent Detection System

Other research for tracking the positions of non-cooperating agents (UAVs, manned aircraft, and birds) could be added to this system. The avoidance system could integrate tracking information from a detection system, allowing the Boid Brain on each aircraft to adjust their trajectory to avoid conflicts with non-cooperating agents.

8.1.5 Redundant Fleet Coordinator

The Big Sister could be altered to automatically activate on a different aircraft if the currently running Big Sister fails. To do this a version of Big Sister should be run on all aircraft, with only one of them sending coordination information. If the Big Sister sending information stops sending, another Big Sister would take over running the coordination. This would prevent a complete system failure, instead risking only a momentary delay while the system waits for the new Big Sister to take control.

8.1.6 Fixed Wing Development

The formation control system could be moved onto fixed wing aircraft platforms, opening many possible areas of further research.

Energy Efficient Formations

Fixed-wing energy efficiency can be improved through the use of formation flying, with energy savings up to 18% [5]. By implementing the formation control system on fixed wing

aircraft, the multi-agent fixed-wing platform could be used for further research into the energy savings of flying in formation at various aircraft sizes.

Midair refueling

Fixed-wing aircraft in close formation could be used to explore autonomous midair refueling of gas or hybrid UAVs. Midair refueling could increase effective range.

8.1.7 Interaction among Multiple Formations

The interaction among multiple formations could be investigated. Having multiple formations of aircraft interact where each group acts as a single entity, could allow larger numbers of aircraft to safely share a limited airspace, such as in a city or around an airport.

8.1.8 Triangulation with Multi-Aircraft Sensors

The formation control system created here could support other systems using sensor data from multiple aircraft to triangulate the positions of non-cooperating agents. This system has the ability to position the aircraft, within a reasonable error, in predetermined locations. The system can also record the positions of each aircraft with respect to time, allowing sensor data to be correlated with the position of each aircraft at any given moment.

Bibliography

[1] Y. Brend, "Drone test flight successfully delivers prescription drugs in Canada for 1st time," CBC News, 29 Aug 2019. [Online]. Available: https://www.cbc.ca/news/canada/british-columbia/drones-used-deliver-drugs-saltspring-london-drugs-drone-1.5264178. [Accessed 9 September 2021].

[2] S. Simmie, "Iris Automation: A closer look at detect and avoid," DroneDJ, 28 Jun 2021. [Online]. Available: https://dronedj.com/2021/06/28/iris-automations-detect-and-avoid-a-closer-look/. [Accessed 20 Dec 2021].

[3] Intel, "Shine Bright with Intel Drone Light Shows," Intel, [Online]. Available: https://www.intel.ca/content/www/ca/en/technology-innovation/aerial-technology-light-show.html. [Accessed 9 Sept 2021].

[4] Boeing, "Boeing Airpower Teaming System," [Online]. Available: https://www.boeing.com/defense/airpower-teaming-system/index.page. [Accessed 12 August 2021].

[5] M. J. Vachon, R. J. Ray, K. R. Walsh and K. Ennix, "F/A-18 Performance Benefits Measured During the Autonomous Formationi Flight Project," 2003.

[6] J. Bolting, "An open benchmark for distributed formation flight contorl of Fixed-Wing Unmanned Aircraft Systmes," *IFAC-PapersOnLine,* vol. 50, no. 1, pp. 10256-10261, 2017.

[7] Airbus, "Airbus UTM: Deploying Unmanned Traffic Management solutions," Airbus, 2021. [Online]. Available: https://www.airbus.com/en/innovation/autonomous-connected/unmanned-traffic-management/airbus-utm. [Accessed 11 Nov 2021].

[8] A. Giagkos, E. Tuci, M. S. Wilson and P. B. Charlesworth, "UAV fligt coordination for communication networks: genetic algorithms versus game theory," *Soft Compute 25,* p. 9483–9503, 2021.

[9] R. O. Saber and R. M. Murray, "Flocking with Obstacle Avoidance: Cooperation with Limited Communication in Mobile Networks," in *IEEE Conference on Decision and Contorl*, 2003.

[10] C. W. Reynolds, "Flocks, Herds, and Schools: A Distributed Behavioral Model, in Computer Graphics," in *SIGGRAPH '87 Conference Proceedings*, 1987.

[11] Mohamed, Mohamed and M. Abbas, "Obstacles Avoidance for Mobile Robot Using Enhanced Artificial Potential Field," *Al-Khwarizmi Engineering Journal,* pp. 71-82, 2013.

[12] S. Housheng, W. Xiaofan and C. Guanrong, "A connectivity-preserving flocking algorithm for multi-agent systems," *Internatilan Journal of Contorl,* vol. 82, no. 7, pp. 1334-1343, 2009.

[13] R. Olfati-Saber, "Flocking for Multi-Agent Dynamic Systems: Algorithms and Theory,"

IEEE Transactions on Automatic Control, vol. 51, no. 3, 2006.

[14] M. Jafari and H. Xu, "A biologically-inspired distributed fault tolerant flocking control for multi-agent system in presence of uncertain dynamics and unknown disturbance," *Engineering applications of artificial intelligence,* vol. 79, 2019.

[15] C. Reynolds, "Boids: Background and Update," 6 September 2001. [Online]. Available: http://www.red3d.com/cwr/boids/. [Accessed 14 August 2020].

[16] Y. Quan, Z. Jingyuan and L. Xiang, "Outdoor flocking of quadcopter drones with decentralized model predictive control," Elsevier Ltd, 2017.

[17] P. Vadakkepat, K. Tan and W. Ming-Liang, "Evolutionary Artificial Potential Fields and their application in real time robot path planning," in *IEEE Conference on Evolutionary Computation*, 2000.

[18] J. Zhang, "Fixed-Wing UAV Formation Control Design with Collision Avoidance Based on an Improved Artificial Potential Field," *IEEE Access,* vol. 6, no. 2885003, pp. PP 1-1, 2018.

[19] F. Bounini, D. Gingras, H. Pollart and D. Gruyer, "Modified Artificial Potential Field Method for Online Path Planing Applications," *IEEE Intelligent Vehicles Symposium (IV),* pp. pp. 180-185, 2017.

[20] J. B. Mbede, H. Xinhan and W. Min, "Fuzzy motion planning among dynamic obstacles using artificialpotential fields for robot manipulators," *Robotics and Autonomous Systems,* pp. 61-72, 2000.

[21] J. Meng, M. Abubakr and E. Magnus, "Leader-Based Multi-Agent Coordination: Controllability and Optimal," in *Proceedings of the 2006 American Control Conference*, Minneaplois, 2006.

[22] J. Zhijian, W. Zidong, L. Hai and W. Zhen, "Interconnection topologies for multi-agent coordination under leader–follower framework," *Automatica,* vol. 45, no. 12, pp. 2857-2863, 2009.

[23] A. Glaser, "Intel invented a way for a single operator to fly hundreds of drones at once," Vox, 4 Nov 2016. [Online]. Available: https://www.vox.com/2016/11/4/13517550/intel-single-operator-fly-hundreds-drones-shooting-star. [Accessed 20 Dec 2021].

[24] PX4 Development Team, "Using the ECL EKF," PX4 Autopilot, 06 09 2021. [Online]. Available: https://docs.px4.io/master/en/advanced_config/tuning_the_ecl_ekf.html. [Accessed 17 September 2021].

[25] Movable Type Scripts, "Calculate distance, bearing and more between Latitude/Longitude points," Unlisted. [Online]. Available: https://www.movable-type.co.uk/scripts/latlong.html. [Accessed 30 August 2021].

[26] K. Gade, "A Non-singular Horizontal Position Representation," *The Journal of Navigation,* pp. 395-417, 2010.

[27] NOAA, "Is the Earth round?," NOAA, 26 02 2021. [Online]. Available:

https://oceanservice.noaa.gov/facts/earth-round.html. [Accessed 17 09 2021].

[28] C. Karney, "Algorithms for geodesics," *Journal of Geodesy*, vol. 87, pp. 43-55, 2013.

[29] W. A. Stone, "An Overview of Global Positioning System Continuously Operating Reference Statsions," NOAA, [Online]. Available: https://www.ngs.noaa.gov/PUBS_LIB/GPS_CORS.html. [Accessed 20 September 2021].

[30] C. Ma, Joint Unscented Kalman Filter for Dual Estimation, 2015.

[31] PX4 Development Team, "Log Analysis using Flight Review," PX4 Autopilot, 3 June 2021. [Online]. Available: https://docs.px4.io/master/en/log/flight_review.html#vibration. [Accessed 21 September 2021].

[32] Px4 Team, "Flight Review," PX4 Autopilot, 2021. [Online]. Available: https://review.px4.io/. [Accessed 11 nov 2021].