3-21-2013

Aerial Networking for the Implementation of Cooperative Control on Small Unmanned Aerial Systems

Scott A . Songer

Recommended Citation

Songer, Scott A ., "Aerial Networking for the Implementation of Cooperative Control on Small Unmanned Aerial Systems" (2013). *Theses and Dissertations*. 1013.
https://scholar.afit.edu/etd/1013

AERIAL NETWORKING FOR THE IMPLEMENTATION OF COOPERATIVE CONTROL ON SMALL UNMANNED AERIAL SYSTEMS

THESIS

Scott A. Songer, Captain, USAF

AFIT-ENV-13-M-29

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENV-13-M-29

AERIAL NETWORKING FOR THE IMPLEMENTATION OF COOPERATIVE CONTROL ON SMALL UNMANNED AERIAL SYSTEMS

THESIS

Presented to the Faculty
Department of Systems and Engineering Management
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
In Partial Fulfillment of the Requirements for the
Degree of Master of Science in Systems Engineering

Scott A. Songer, BS
Captain, USAF

March 2013

AFIT-ENV-13-M-29

AERIAL NETWORKING FOR THE IMPLEMENTATION OF COOPERATIVE CONTROL ON SMALL UNMANNED AERIAL SYSTEMS

Scott A. Songer, BS

Captain, USAF

Approved:

Dr. John M. Colombi, PhD (Chairman)	Date
Dr. David R. Jacques, PhD (Member)	Date
Dr. Michael R. Grimaila, PhD (Member)	Date

AFIT-ENV-13-M-29

Abstract

The employment of Small Unmanned Aerial Systems (SUAS) for reconnaissance and surveillance missions is a vital capability of the United States military. Cooperative control algorithms for SUAS can enable tactical multi-vehicle configurations for communications extension, intelligent navigation, and a multitude of other applications. Past research at AFIT has designed and simulated a cooperative rover-relay algorithm for extended communications and has investigated its implementation through various modem configurations. This research explores aerial networking options for implementing cooperative control and applies them to an actual SUAS. Using Commercial Off-The-Shelf (COTS) hardware, a system was designed and flight tested to implement the rover-relay algorithm and provide a test bed system for future research in cooperative control.

Two different modem configurations were designed and tested. The first modem configuration was demonstrated through a series of ground and flight tests to successfully relay autopilot commands and telemetry between a ground station and a rover aircraft through a relay aircraft. This configuration effectively doubles the effective range of the rover system to 1.2 miles, together with an algorithm that autonomously navigates the relay aircraft to an optimal location. Secondly, a mesh network was configured and tested. This configuration successfully relayed aircraft telemetry to the ground station from each vehicle in the network. However, the network suffered from low throughput, which limited autopilot functionality, such as updating navigation waypoints to each aircraft. The results suggest the system be updated with more capable modems in a mesh configuration to broaden the possibilities for future research in cooperative applications.

Acknowledgments

Thanks to my advisors, Dr. John Colombi and Dr. David Jacques for all the guidance and support. Thanks to Rick Patton and Don Smith for helping out with hardware modifications and flight test support. Thanks to my fellow students Major Jonathan Welborn and Lieutenant Timothy Shuck for being great teammates. Lastly, thanks to my wife for giving me the emotional support to get through this. You were all extremely helpful and supportive throughout my time here at AFIT.

Scott A. Songer, Captain, USAF

Table of Contents

Page

Abstract iv

Acknowledgments v

List of Figures viii

List of Tables x

I. Introduction 1

1.1 Background 1
1.2 Problem Statement 3
1.3 Scope 4
1.4 Assumptions 5
1.5 Overview 7

II. Background 8

2.1 Chapter Overview 8
2.2 Air Force SUAS Doctrine and Need 8
2.3 The Overhead Watch and Loiter (OWL) SUAS 10
2.4 Alternative Autopilot Systems 15
2.5 Aerial Networking 16

III. Methodology 22

3.1 Chapter Overview 22
3.2 Design 22
3.3 Test 25

IV. Results 29

4.1 Chapter Overview 29
4.2 Design Stage 1 and Ground Testing 29
4.3 Flight Test #1 33
4.4 Design Stage 2 and Ground Testing 34
4.5 Flight Test #2 42
4.6 Design Stage 3 and Ground Testing 45
4.7 Summary of Flight and Ground Test Results 54

V. Conclusions and Recommendations 57

5.1 Chapter Overview 57
5.2 Retrospective and Challenges 57
5.3 Recommendations for Future Research 58
5.4 Summary 60

Appendix A. Flight Test Procedures 62

Flight Test #1 (24-25 September 2012) 62

Flight Test #2 (5-7 November 2012) 64

Bibliography 73

List of Figures

Page

Figure 1: Rover-Relay Cooperative Algorithm [1].. 2

Figure 2: OWL Testbed Aircraft... 10

Figure 3: OWL Avionics Bay [1] 11

Figure 4: OWL Operational Concept, DODAF OV-1 [1] 12

Figure 5: Gas-Powered Sig Rascal Aircraft.. 14

Figure 6: Relay-Assisted Point to Point Communication Diagram [1] 17

Figure 7: Mesh Network Transmission from Node A to Node D – Step 1 [1]................ 17

Figure 8: Mesh Network Transmission from Node A to Node D – Step 2 [1]................ 18

Figure 9: Mesh Network Transmission from Node A to Node D – Step 3 [1]................ 18

Figure 10: “Total Network Throughput Achieved as a Function of the Number of Competing Nodes. All Nodes are Within Each Others’ Radio Ranges, and all Nodes Send as Fast as 802.11 Allows” [12]. 20

Figure 11: “Throughput Achieved Along a Chain of Nodes, as Function of the Chain Length. The Nodes are 200 Meters Apart. The First Node Originates Packets as Fast as 802.11 Allows, to be Forwarded Along the Chain to the Last Node. The Throughputs for Chains of 20 and 50 Nodes are the Same as for 10 Nodes” [12]. 21

Figure 12: Organizational Flow of Chapter 4 29

Figure 13: Baseline Ardupilot OWL Vehicle Design.. 30

Figure 14: Turnigy 9x RC Controller with FrSky Radio Module 31

Figure 15: Ground Station Design with 3DR Modem.. 32

Figure 16: Programming a Repeating Circuit of Waypoints in Mission Planner............. 34

Figure 17: XBee Pro 900 Communications Relay Schematic.................................... 36

Figure 18: Sig Rascal Relay Vehicle Design.. 37

Figure 19: OWL Rover Vehicle Design 38

Figure 20: Reconfigured OWL with Ardupilot Mega 2.0, FrSky RC Receiver, and Video Modem 39

Page

Figure 21: Ground Station Design with XBee Modem 40

Figure 22: Flight Test #2 Ground Station Architecture 41

Figure 23: Gain Parameters for OWL Platform 43

Figure 24: Gain Parameters for Sig Rascal Platform 43

Figure 25: OWL Design with Redesigned Power Bus 46

Figure 26: RPSMA Antenna Placement on Sig Rascal Relay Vehicle 47

Figure 27: Sig Rascal Avionics and Relay Modems 48

Figure 28: Communications Extension Ground Test 50

Figure 29: Optimal DigiMesh Parameters for Rover-Relay in X-CTU application 52

Figure 30: Comparison of Digi Pro and DigiMesh Modem Frameworks when Transmitting Rover Waypoints and Relay and Rover Telemetry to Ground Station 53

Figure 31: Wave Relay Sector Antenna Array Router [19] 59

List of Tables

Page

Table 1: Flight Test Matrix 26
Table 2: Flight Test Results Matrix – Summary of Test Objectives and Results. 55
Table 3: Ground Test Results Matrix – Summary of Test Objectives and Results. 56

AERIAL NETWORKING FOR THE IMPLEMENTATION OF COOPERATIVE CONTROL ON SMALL UNMANNED AERIAL SYSTEMS

I. Introduction

1.1 Background

Small Unmanned Aerial Systems (SUAS) have become integral to surveillance and reconnaissance missions in DoD Overseas Contingency Operations. SUAS, compared with larger UAVs (Unmanned Aerial Vehicles) like the MQ-1 Predator and MQ-9 Raptor, are cheaper and more readily deployable within mobile ground units. SUAS such as the RQ-11 Raven are hand-launched and used for immediate short range surveillance. The current configuration for these hand-launched SUAS is for a single operator to control waypoints and monitor real-time video relayed from the aircraft at a ground station – usually with a device resembling a laptop computer.

Cooperative control algorithms could be applied in these systems to employ multiple aircraft to accomplish more extensive surveillance missions without multiplying the number of ground stations and operators required. In other words, a single operator could launch multiple aircraft and toggle between various cooperative settings such as flock, loiter, survey, relay, and so on. These cooperative configurations could extend range, accomplish broader or more complex surveillance in less time, or provide multiple sensing capabilities to a single target. The simplest of such cooperative control configurations is that of the rover-relay. A relay aircraft can be employed to extend the communication range of a rover vehicle while auto-navigating based on the location of the rover. Figure 1 displays the employment of a rover-relay system to extend communications range to surveil a distant target.

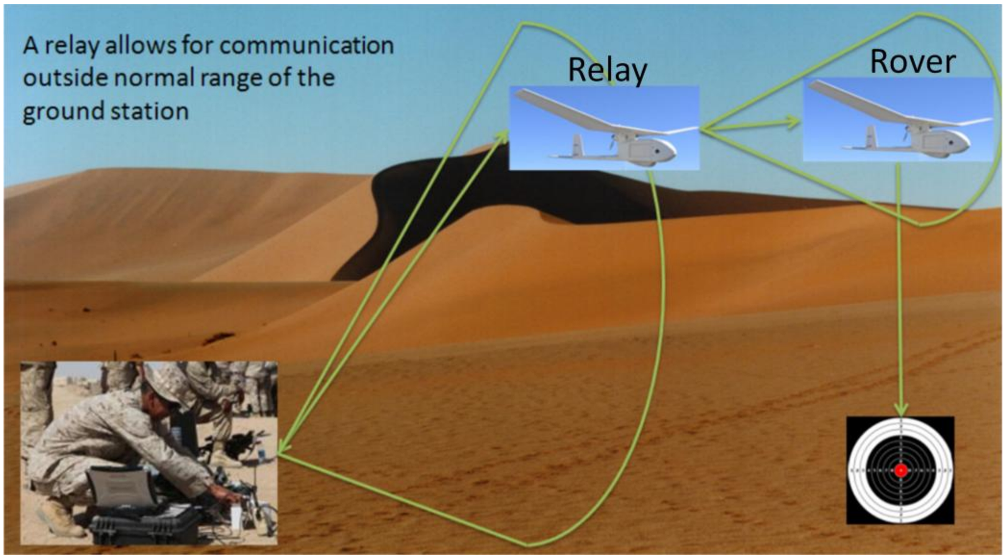


Figure 1: Rover-Relay Cooperative Algorithm [1]

Unfortunately, current SUAS platforms do not have the systems architecture to accommodate such capabilities; they lack cross-UAV aerial networking and onboard data processing functions. Meanwhile, the roles of SUAS and other UAVs are constantly expanding, and the potential benefits for extended range, enhanced communications capability, and cooperative control function are ever increasing.

This chapter will discuss the problem the research intends to solve, the scope of the research, the assumptions made in approaching the research, and lastly, an overview of the rest of the thesis' content.

1.2 Problem Statement

This research will investigate a SUAS architecture that supports the implementation of cooperative control algorithms. Past efforts at the Air Force Institute of Technology (AFIT) have already been accomplished toward exploring and designing cooperative control algorithms, as well as developing a communications relay using two UAVs. This past research has been conducted using modified RQ-11 Raven aircraft, known as OWLs (Overhead Watch and Loiter), as a platform for design and test. However, as previously mentioned, the current OWL SUAS does not support a wireless aerial network, nor does it support onboard data processing or storage capability. Currently, all algorithms except the basic autopilot functions must be stored and executed at the ground station, which communicates directly with a single UAV. This simple configuration limits the scope of what can accomplished with cooperative control, thus previous research has demonstrated cooperative control algorithms only through simulation.

The primary research question of this thesis is the following: what small unmanned airborne system communications architecture supports cooperative control through a COTS hardware and software configuration? The following questions support this larger question.

- What autopilot chipset facilitates more design choices?
- What autopilot and modem configuration supports a functional communications relay?

- What modem configuration provides an airborne mesh network with OWLs as nodes in the network?
- If a mesh network can be established, what cooperative control algorithms can be employed operationally on the SUAS?

It is likely that a wireless mesh networking architecture and onboard microprocessor would expand the current system to accommodate existing and unforeseen cooperative control capabilities, and that is precisely what this thesis is intended to explore.

1.3 Scope

The overall objective of this research is to demonstrate a system that supports a functioning communications relay between two aircraft and a ground station. The secondary objectives are to establish a wireless mesh network between UAVs, as well as an onboard microprocessing capability. This system would serve as an architecture to facilitate the future employment of cooperative control algorithms. Therefore, it is not within the scope of this research to design cooperative control algorithms, but to explore a configuration that supports them. The first step is to establish a functioning communication relay with two UAVs, in which a UAV closer to the ground station can relay commands to a more distant vehicle without interpreting the commands itself. This objective is defined as the relay of command and telemetry communications between the autopilot system and the ground station, not to include video. The second priority is to employ an existing rover-relay algorithm, which will result in an auto-positioning of the relay UAV based on the position of the rover, relay, and ground station, allowing the pilot to control only the rover UAV and benefit from an extended range of the relayed signal through the relay aircraft. The third priority is to transfer the rover-relay algorithm from the ground station onto a UAV-borne microprocessor. An on-aircraft microprocessing capability provides an architecture that supports expandability of different cooperative control algorithms. The final research objective is to establish a functioning mesh network of UAVs that relays commands to particular aircraft, automatically relaying between aircraft with available communications. Each of these objectives will be completed with either flight testing or ground testing to verify the design before progressing to the next objective. Also, the system design will be recorded and updated with each change to map the overarching system and its many interfaces and functions.

1.4 Assumptions

The research of this thesis will be mostly conducted on the AFIT campus with hardware and software owned by the AFIT/ENV, the Department of Systems Engineering and Management. A number of different autopilot chips, modems, airframes, and software applications have been used over the years by other students that remain at the disposal of the current OWL team. The details of this hardware will be discussed in later chapters. There also exists a limited budget for the purchase of new equipment. An assumption of this research is that a configuration will be achieved using existing Commercial off the Shelf (COTS) hardware (autopilots, modems, and airframes). The intent of this research is not to design a modem or an autopilot, but to configure a system capable of the functions previously mentioned. Microprocessor configuration, network design, and systems architecting are the skills to be utilized to achieve the research objectives.

Flight testing is a crucial part of the methodology for verifying the research objectives. All flight testing was accomplished at Camp Atterbury, a small Army Post in Indiana, with the support of CESI (Cooperative Engineering Services, Incorporated), a contractor hired by AFIT to provide certified UAV pilots and hardware support necessary for flight testing. Flight testing will be conducted professionally and methodically; flight plans will be written, approved, and followed to achieve test objectives and evaluate measures of effectiveness. A Test Review Board/Safety Review Board (TRB/SRB) meeting will be conducted before each flight test to gain approval of the test objectives as well as safety procedures. Each research objective will be verified with an operational flight test to validate the functionality of the design through quantitative mission objectives. Once flight testing has been concluded, the success of the research can be measured by which objectives were successfully flown, as will be discussed further in Chapter 3. However, flight testing is also a potential limitation of this research. Camp Atterbury is notorious for scheduling complications, and it is particularly hard to fly during the late fall and winter seasons due to temperature, wind, and precipitation. Therefore, all flight testing should be concluded prior to the end of November. This scheduling limitation condenses the initial research and design portion of this thesis. The main limitation of the research in the initial design phase is COTS hardware availability and the lead time associated with purchasing new equipment.

1.5 Overview

This thesis is comprised of background information, a systems approach to solve the problem, an evaluation of test data and quantitative findings, and finally, conclusions and recommendations for future research. Chapter 2 provides the background information of the OWL SUAS architecture and hardware as well as a review of relevant operational topics, networking, and cooperative control literature. Chapter 3 describes the analytical process, utilizing a systems architecture approach toward design and demonstration of the system. Chapter 4 reviews the findings of the research and flight testing presented in Chapter 3. The last section, Chapter 5, provides conclusions and possibilities for future research based on the overall results and accomplishments of the research.

II. Background

2.1 Chapter Overview

Chapter 2 examines the problem in context with Air Force doctrine, reviews past OWL research at AFIT, and investigates both current technology and relevant theory. Section 2.2 discusses the motivation for this thesis in context with Air Force operational doctrine and current UAS programs and capabilities. Section 2.3 discusses the objectives, accomplishments, and applied hardware configurations of previous OWL teams at AFIT, as well as the lessons learned and path forward for progressing OWL research in context with this thesis. Section 2.4 explores alternative autopilot configurations for the research, and lastly, Section 2.5 examines past research relevant to the application of aerial and mesh networking in SUAS applications.

2.2 Air Force SUAS Doctrine and Need

The Air Force Doctrine Document 1 serves as the fundamental statement of basic doctrine for the USAF, and is maintained under the direction of the USAF Chief of Staff [2]. This document defines the first principal of war as “unity of command,” which carries the primary objective of “directing military operations toward a defined and attainable objective that contributes to strategic, operational, and tactical aims” [2]. This high level doctrinal statement defines strategic and tactical objectives as key factors to unity of command. Surveillance and reconnaissance directly contribute to strategic and tactical objectives. The document goes on to define surveillance and reconnaissance as one of the seventeen “key operational functions.” It is also stated that doctrine is “about effects…not platforms" and that "airmen should be concerned with the best means of employing intelligence, surveillance, and reconnaissance (ISR) capabilities, not whether a particular ISR platform is airborne or in orbit" [2]. Simply stated, there are many means within the USAF of accomplishing ISR, but different situations warrant different platforms. ISR satellites and larger reconnaissance aircraft platforms are under constant demand and have limited availability in a time of war; therefore,

> "UAS have experienced explosive growth in recent history, providing one of the most in demand capabilities the USAF presents to the Joint Force. The attributes of persistence, efficiency, flexibility of mission, information collection … have repeatedly proven to be force multipliers across the spectrum of global Joint military operations" [3].

There are several SUAS employed by the United States armed forces to provide ISR, the most common platform being the RQ-11 Raven [4]. "Small UAS represent a profound technological advance in air warfare by providing situational awareness; the need for situational awareness and full-motion video dominates urgent requests from the field" [3].

The Office of Naval Research conducted a survey on accomplishments in UAV cooperative control, identifying aerial surveillance as the first of five major areas of active research. They found that "a major un-resolved issue for collaborative unmanned aircraft is wireless communication with other cooperating aircraft. The aircraft to ground problem generally involves out of line-of-sight, long range communications" [5]. This deficiency is precisely the topic of this research.

2.3 The Overhead Watch and Loiter (OWL) SUAS

The OWL was developed at AFIT as a test bed vehicle for research in surveillance and reconnaissance missions for SUAS. The OWL vehicle is a deconstructed RQ-11 Raven airframe retrofitted with COTS internal components including an autopilot chipset, modem, GPS (Global Positioning Satellite) receiver, a DC motor, video transmitter, and lithium polymer batteries [1]. The OWL vehicle is displayed in Figure 2 below.

Figure 2: OWL Testbed Aircraft

The nose of the aircraft is detachable, and there are several nose configurations containing different cameras and sensors. The aircraft has a wing span of 55 inches and a flight endurance of 20-25 minutes, powered by two 3-cell 2200mAh lithium polymer batteries. The internal avionics bay of the OWL is pictured in Figure 3. This past OWL configuration includes a Kestrel autopilot chipset and a Microhard modem.

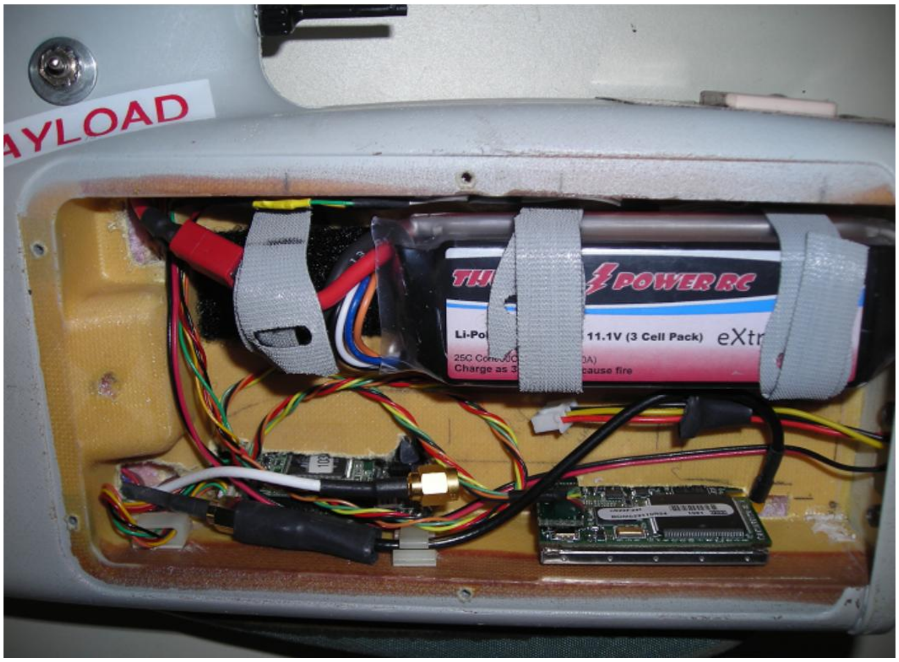


Figure 3: OWL Avionics Bay [1]

The airframe itself, including servo motors and control surfaces, is the only original Raven component remaining in the OWL.

Past research relevant to this thesis began with the development of a communications relay architecture by Lieutenant Matthew Seibert, which was intended to facilitate extended line of sight (LOS) communications [1]. In addition to the communications relay architecture, Seibert also examined theoretical network configurations for SUAS. Following Seibert's work, Captain Jeremy Boire developed a rover-relay algorithm to achieve autonomous routing of unmanned aerial vehicle relays to mimic optimal trajectories in real time. Boire's algorithm was validated in simulation, but never incorporated into a hardware implementation utilizing a communications relay configuration. Boire wrote, "the research seeks to provide a foundation for further study and implementation of automated relay routing in future systems" [6].

The operational concept for the rover-relay OWL system developed in past research is displayed in the Figure 4 below.

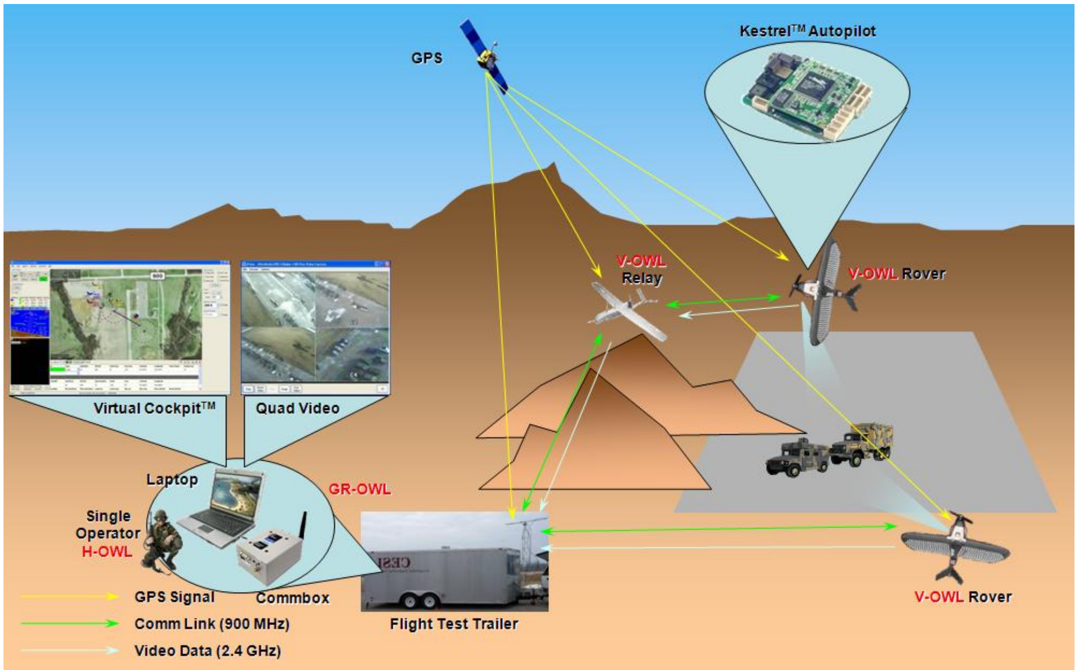


Figure 4: OWL Operational Concept, DODAF OV-1 [1]

This figure depicts a single operator flying a rover-relay pair as well as an additional independent OWL rover, which are controlled from the flight test trailer. The trailer is equipped with communications equipment as well as a ground station laptop computer. Prior OWL systems employ the Kestrel autopilot system and Virtual Cockpit software, which are products of Procerus Technologies [7]. The Kestrel autopilot requires a modem configuration in which the commbox (communications box, also referred to as a ground station) sends communications packets using a User Datagram Protocol (UDP) broadcast modem setting; the packets are then parsed by Kestrel autopilots within range to determine the packets' intended agent [7]. The UDP packets broadcasted by the Commbox are encrypted and use spread spectrum transmission (900MHz-920MHz). The Kestrel autopilot firmware includes a proprietary decryption process to interpret incoming communications packets. Therefore, it is not within the researcher's capacity to decipher or generate communications data at any point in the system between the Commbox and the autopilot [7].

In the Kestrel communications configuration, the modem behaves simply as a relay to transmit and receive the encrypted serial stream of data [7]. Previous OWL researchers utilized a Digi XTend 900 modem in the aircraft and ground station [8]. Seibert identified limitations of the Digi modem, and used Microhard modems during his research to design and test a communications relay configuration. Seibert wrote that Digi modems set in repeater mode are not capable of forwarding and interpreting data simultaneously, whereas Microhard modems in relay mode are, in fact, capable of sending and receiving data simultaneously [1]. Seibert was able to demonstrate extended range of communications by utilizing two Microhard modems with one set in relay mode. However, later OWL researchers have been unable to successfully employ the Microhard relay in conjunction with the Kestrel autopilot system due to the encrypted addressing design used between Virtual Cockpit and the Kestrel autopilot. The research recorded by Seibert and Boire in their theses, in theory, is directly relevant to this research, but the Kestrel-centered hardware configuration is not expandable into an operationally functioning rover-relay or mesh network due to the nature of Kestrel's encrypted communications. The next section of this chapter discusses alternative configurations that better accommodate a relay configuration.

In addition to the OWL platform, the researchers will be adding the Sig Rascal platform to the system. The Sig Rascal is a commercially available aircraft with a 110in wingspan. The aircraft can be flown with either a gas-powered or electric motor, and comes without servos, radio, or autopilot [9]. The Sig Rascal will be configured with identical autopilot and RC (Radio Controlled) hardware as the OWLs. The gas-powered Sig Rascal aircraft is displayed in Figure 5.

Figure 5: Gas-Powered Sig Rascal Aircraft

The advantage to the Sig Rascal is a much longer flight endurance with a gas engine and also increased payload space for equipment. The endurance of the gas-powered Sig Rascal with 5v nickel-metal hydride batteries powering the electronic systems is estimated at 45 minutes.

2.4 Alternative Autopilot Systems

The Ardupilot autopilot, manufactured by Sparkfun Electronics, is an open-source implementation of the Arduino microprocessor, fully user-programmable through the Arduino integrated development environment (IDE). Where the Kestrel autopilot is driven by the Virtual Cockpit software application, the Ardupilot can be controlled by different software applications such as Mission Planner, HappyKillmore GCS, and QGroundControl, which are all independent software developments that utilize Google Earth for geo-positioning and graphical presentation.

The advantage of the Ardupilot for this research is the open design, programmable onboard microprocessor, unencrypted communications, and low cost; the Ardupilot card costs less than one fifth the cost of a Kestrel autopilot, and also has a much more widespread hobbyist community supporting it. The Ardupilot has been proven to function with countless commercial and custom air vehicles as well as in harmony with different modems and RC equipment.

A traditional UAV command architecture fundamentally ties each aircraft to a ground station, with limited communication between aircraft, and with all computation (aside from autopilot functions) occurring at the ground station. For multi-UAS missions, this means each UAS has its own corresponding ground station and operator. For multi-UAS missions, a networking capability combined with an on-aircraft microprocessing capability would provide expanded capability.

QGroundControl provides the capability of controlling and monitoring multiple aircraft through a single software application. Also, QGroundControl has a fully open-source and build-from-source development package available for download on the internet. QGroundControl uses the Qt framework for its development, which allows full customization of widgets and commands to the existing build [10].

2.5 Aerial Networking

Ad hoc air-to-air mesh networks have been discussed in past research, primarily with MUAV (micro unmanned aerial vehicle) aerial sensor networks [11] [12] [13] [14]. In the article “Cognitive Agent Mobility for Aerial Sensor Networks,” Kai Daniel and others propose air-to-air links in an aerial network to “compensate connection losses of A2G (air to ground) links by means of relaying” [11]. These authors study the combination of sensor placement strategy with reliable communication networks, i.e. communication-aware mobility of an aerial network. Networks of this nature can be analyzed by three key performance measures: spatial 3D coverage, receive signal strength indicator (RSSI), and the rate of connectivity loss [11]. The notion of communication-aware aerial networks is a cross-discipline concept that merges control theory and network design.

For the purposes of this research, an ad-hoc aerial network is optimal to relay communications between the ground station and airborne vehicles. Specifically, a mesh network topology supports the autonomous relay of data through any node in a network to its destination node. For instance, in a general network a communication packet may originate at Node A with the destination of Node D as depicted in Figure 6.

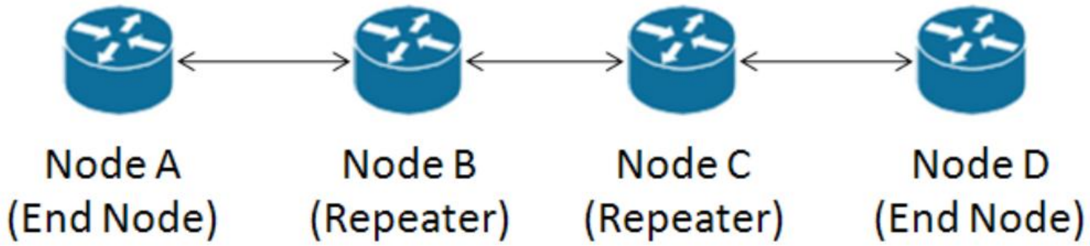


Figure 6: Relay-Assisted Point to Point Communication Diagram [1]

In an ad-hoc or "mesh network" each node is a peer, rather than some being designated as repeater or end nodes. When a communication packet is sent, it automatically relays across the network until it arrives at its destination node, as depicted in Figures 7-9 with a transmission originated at Node A and the destination at Node D.

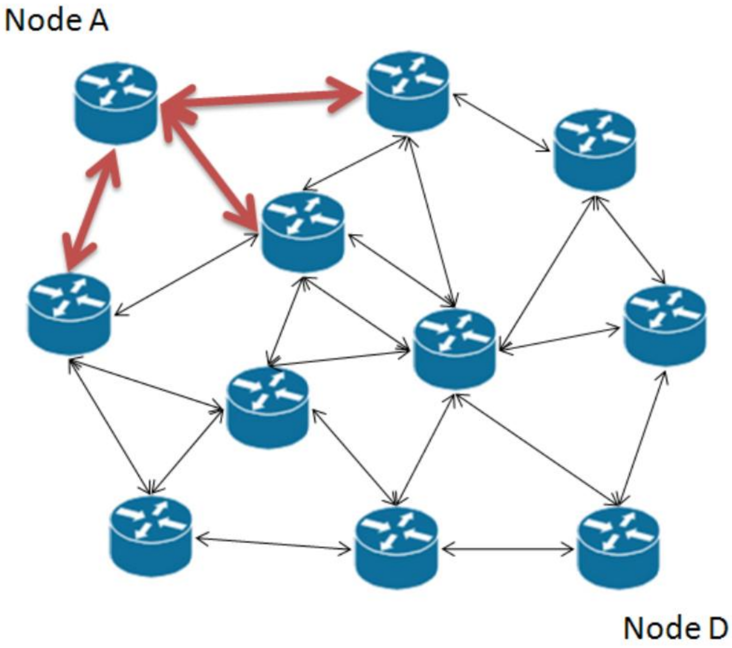


Figure 7: Mesh Network Transmission from Node A to Node D – Step 1 [1]

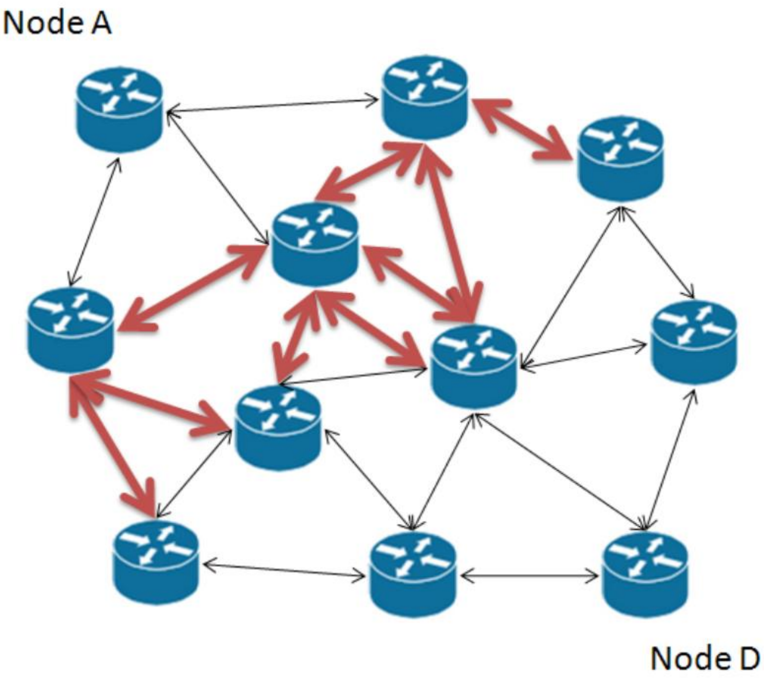


Figure 8: Mesh Network Transmission from Node A to Node D – Step 2 [1]

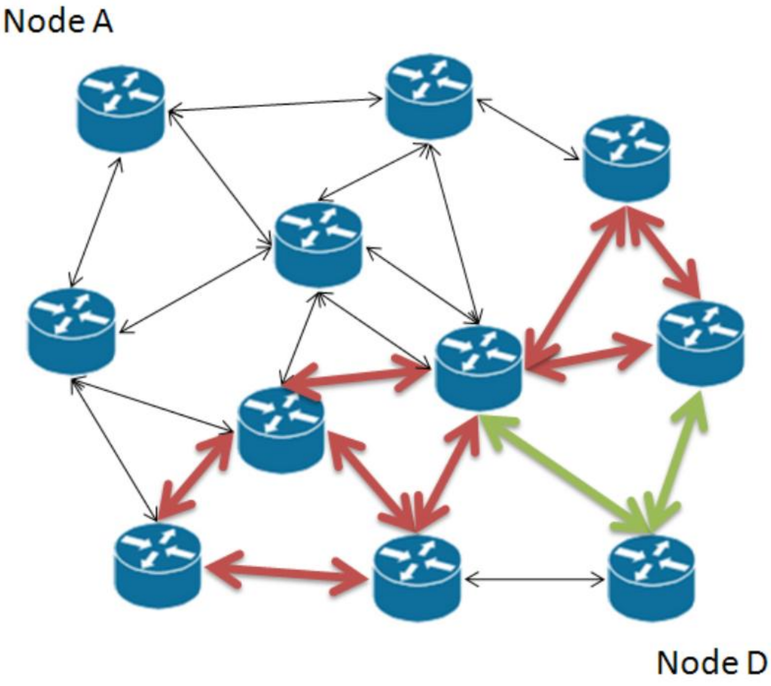


Figure 9: Mesh Network Transmission from Node A to Node D – Step 3 [1]

In the history of mesh networking, there have been countless protocols developed to implement the routing process of traversing data across the network. However, wireless mesh networking inherently creates limitations in data throughput and overall network capacity. In their seminal article “The Capacity of Wireless Networks,” P. Gupta and P. R. Kumar state that one of the main deficiencies of multi-hop wireless networks is a reduction in total network capacity due to the interference between multiple simultaneous transmissions [13]. As the number of hops increases across a mesh network, performance sharply degrades. When each node has an identical omnidirectional radio range, a two-hop transmission halves the throughput of a single-hop transmission simply because only one of the two hops can be active at a time to avoid wireless interference [14]. Gupta and Kumar demonstrate that in a mesh network of n identical nodes, each with a data rate of W bits per second, the throughput per node, $\lambda(n)$ is

$$\lambda(n)=\Theta\left(\frac{W}{\sqrt{n \log n}}\right)$$

bits per second. This function assumes random communication pattern and node placement [13]. Furthermore, M.I.T. researchers in the article “Capacity of Ad Hoc Wireless Networks” write that “a common observation in analyses of ad hoc routing protocols is that capacity is the limiting factor; that is, the symptom of failure under stress is congestion losses” [12]. Jinyang Li and others expand on Gupta’s formula by testing actual 802.11 networks to measure the degradation in throughput as the number of nodes in the mesh network is increased. Unfortunately, the network capacity decreases sharply for a few nodes; the degradation levels off for about 10 nodes and greater.

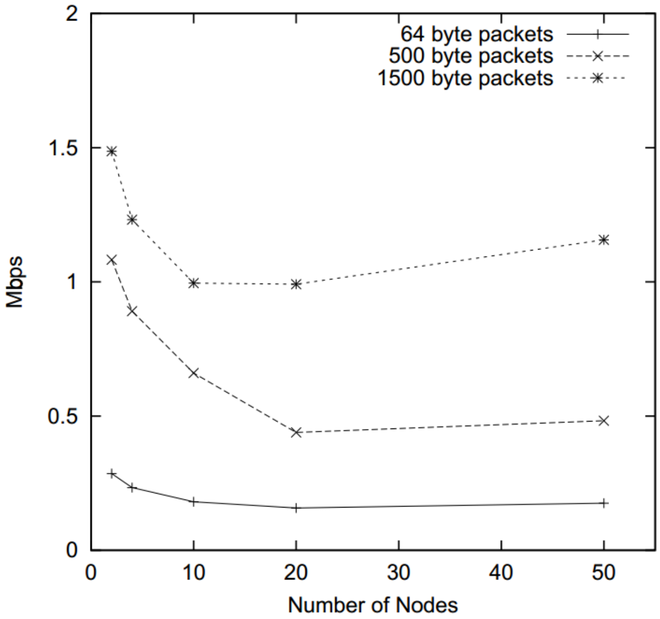


Figure 10: "Total Network Throughput Achieved as a Function of the Number of Competing Nodes. All Nodes are Within Each Others' Radio Ranges, and all Nodes Send as Fast as 802.11 Allows" [12].

Figure 10 exhibits that throughput degrades as the number of nodes is increased, with each node placed within range of each other. Figure 11 is more applicable to this research, as it shows the throughput degradation of a mesh as nodes are increased, but arranged in a chain.

Regarding the network configuration of the plot in Figure 11, Li and others note that the network approaches a utilization of 0.25 Mbps, which is 1/7 the maximum bandwidth, substantially less than the predicted 1/4 [12].

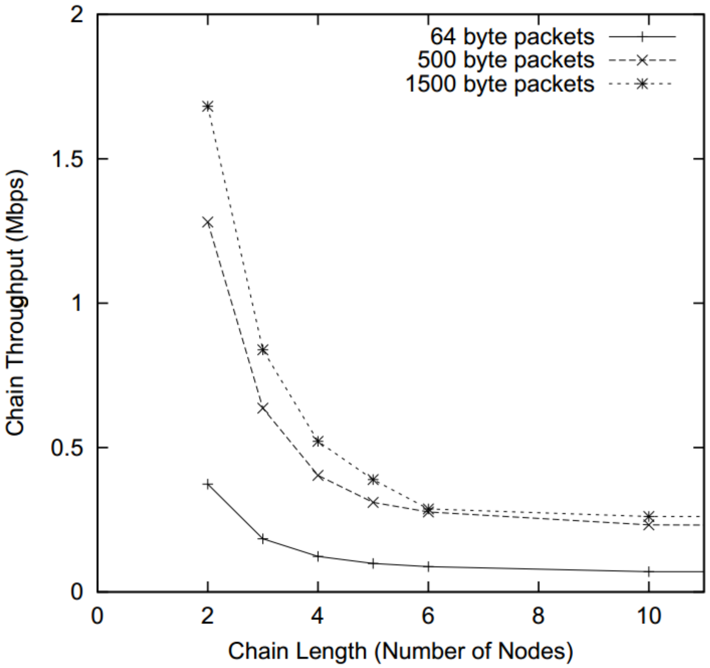


Figure 11: "Throughput Achieved Along a Chain of Nodes, as Function of the Chain Length. The Nodes are 200 Meters Apart. The First Node Originates Packets as Fast as 802.11 Allows, to be Forwarded Along the Chain to the Last Node. The Throughputs for Chains of 20 and 50 Nodes are the Same as for 10 Nodes" [12].

This supports the fact that the degradation of bandwidth in a mesh network is potentially greater in implementation than simulation. This effect is attributed to uneven bandwidth allocation between nodes by the chain scheduling logic of the network protocol and also to hopping interference between neighboring RTS (ready-to-send) and CTS (clear-to-send) packet transmissions [12].

With the low data rate (a maximum of 156 kbps) 900MHz radios used in this research, this problem of capacity loss in a mesh network configuration may prove problematic in routing aircraft telemetry and parameters between airborne vehicles and the ground station [15].

III. Methodology

3.1 Chapter Overview

The methodology of this research consists of an iterative sequence of design and test to accomplish the research objectives. Testing will directly validate each design objective, while analyzing the system performance. Chapter 3 is divided into two main sections accordingly: Design and Test. Section 3.2 describes the requirements of the system, the sequence of design, and how each phase is accomplished, to include a review of major design decisions. Section 3.3 describes the test plan for the research, a test matrix that incorporates test objectives and parameters toward design validation, and a methodology of data collection and analysis.

3.2 Design

The basic requirements of the system to be designed and tested are the following:

1. The system must be capable of navigating multiple aircraft through both autopilot commands and through radio control.
2. The system must execute and track planned waypoint flight paths and be able to change the course of the flight path through wireless autopilot command.
3. The system must be capable of relaying autopilot commands from one aircraft to another beyond direct LOS.
4. The system must be able to accommodate the implementation of a rover-relay command algorithm, which will autonomously navigate the relay aircraft.

The design sequence of the research directly correlates to the research objectives of this thesis and the requirements derived from the objectives. Specifically, the design sequence is as follows:

1. Equip OWLs and Sig Rascal with Ardupilot autopilot and modem.
2. Establish relay configuration between two modems.
3. Implement rover-relay algorithm from ground station.
4. Configure aircraft-to-aircraft mesh network between multiple systems (three or more).

3.2.1 Step 1

The first step of the design directly follows the decision to replace the Kestrel autopilot system of the OWL with the Ardupilot autopilot that was reviewed in the previous chapter. Inserting the Ardupilot into the OWL system consists of hardware reconfiguration, establishing new autopilot control gains for the OWL, installing an alternative RC control system, and employing a new ground station software application at the ground station. QGroundControl is the best suited ground station application for the research because it supports simultaneous control of multiple aircraft through a single ground station, where Mission Planner and HappyKillmore GCS do not. However, Mission Planner is the most mature and functionally stable software application for Ardupilot. At the time of this research, the 915MHz 3DR modem is the most popular option for hobbyists flying Ardupilot 2.0 configured vehicles through Mission Planner. Therefore, for the purposes of establishing a baseline Ardupilot-equipped system, the 3DR modem will be used.

3.2.2 Step 2

Step two of the design sequence involves selecting the optimal modem for the application of aerial networking and configuring multiple modems to provide a functional rover-relay relationship between two aircraft. The ground station must be capable of controlling and monitoring the rover vehicle beyond direct LOS communications range, with the relay vehicle automatically forwarding communications packets between the rover and ground station, thus increasing the range of the system.

3.2.3 Step 3

The implementation of Capt Boire's rover-relay algorithm at the ground station requires rewriting his code to receive telemetry data from two OWLs real-time, while calculating and transmitting new waypoints back to the relay OWL. The computational core of the algorithm will be preserved, thus retaining coherence with his research. This part of the research is to be accomplished in parallel with Lieutenant Timothy Shuck, a fellow AFIT/ENV Masters student. Lieutenant Shuck has a Controls focus in his studies at AFIT, and has the primary research objective of implementing Capt Boire's algorithm with the facilitation of the communications relay configured through the efforts of this thesis' research.

3.2.4 Step 4

Step four is the most challenging design step; it involves configuring a mesh network between aircraft using the modems selected in Step 2. The mesh network configuration must be capable of automatically relaying data between aircraft to an intended receiver. The addressee of relayed data can either be a particular aircraft or the ground station itself. In context to the rover-relay cooperative algorithm, the mesh network supports each aircraft as a potential relay or rover depending on their distance from the ground station. The network will echo communications through the network in order to ultimately transmit communication to the intended recipient, which may or may not be outside the direct range of the ground station itself. The routing protocol of the network may either be selected based on the optimal path for the small number of nodes and characteristics of this particular system, or it may be entirely determined by the modem selected.

3.3 Test

The test phase of the research validates and analyzes the design of the new OWL and Sig Rascal configurations. Tests are to be accomplished through ground testing as well as flight testing. The flight testing will occur at Camp Atterbury airfield with the support of CESI. Each flight test will be preceded by a TRB/SRB in order to gain approval of safety procedures as well as the test objectives to be accomplished. Ground testing will serve as validation of different design implementations to reduce risk of failure at flight test events.

3.3.1 Flight Test

The following test matrix summarizes the pass/fail objectives of each flight test mission.

Table 1: Flight Test Matrix

Flight Test	**Test Objectives**
Flight Test #1	1. Establish control gains for Ardupilot Mega in OWL and verify stable flight of vehicle 2. Validate functionality of Mission Planner for single and multi-vehicle operation
Flight Test #2	1. Establish control gains for Ardupilot Mega in Sig Rascal and verify stable flight of vehicle 2. Verify operability of QGroundControl 3. Verify operability of communications relay in flight 4. Determine direct communications range of autopilot system with current modems 5. Verify operability of rover-relay algorithm implementation at the ground station
Flight Test #3	1. Verify operability of rover-relay with mesh modem configuration

Flight test #1 has the primary objective of establishing control gains for the Ardupilot installed into the OWL airframe, and verifying stable flight with the programmed gains and parameters. These gains can only be fully mapped during a live test flight by optimizing existing generic gains of a similar airframe. Secondly, Flight test #1 has the objective of verifying the operation of Mission Planner in conjunction with the Ardupilot-configured OWLs. This objective is meant to familiarize the researchers with the system as well as to unveil any limitations that may impede progress with the research objectives leading into Flight Test #2. As such, this second objective also includes verifying the capability to fly two Ardupilot-equipped OWLs simultaneously using Mission Planner. The flight test will first establish and verify gains with a single aircraft, and then verify the functionality of Mission Planner with multiple aircraft in flight, using two ground stations.

Flight Test #2 has the first objective of establishing gains for the Ardupilot-equipped Sig Rascal vehicle. Secondly, QGroundControl will be tested with a single ground station and aircraft to verify the operability of the system using QGroundControl rather than Mission Planner. Thirdly, Flight Test #2 verifies the operation of the communications relay in flight. This will be verified by simply addressing commands for a specific OWL or Sig Rascal at the ground station and relaying them through a different relay vehicle, and then verifying that the rover vehicle responds to commands in flight without having direct communication with the ground station. This flight test objective also serves as a means of collecting telemetry on the performance of the modems on the OWLs and the ground station as an initial characterization of the aerial network capability. The fourth flight test objective is to determine the range of the system with a single ground station and a single vehicle in flight with the current modem configuration that is selected at this point in the design. Lastly, the rover-relay algorithm at the ground station will be tested by flying two aircraft with the communications relay configuration to verify the operation of the autonomous rover-relay implementation. This objective is successful if the relay vehicle auto-positions based on commands generated by the algorithm at the ground station, which determines the midpoint between the rover and the ground station.

Flight test #3 is the final flight test, which has the only objective of verifying the operation of the mesh network configuration, and testing its ability to accommodate the rover-relay ground station implementation. This objective will be flown exactly like the final objective of Flight Test #2; the mesh configuration should be a transparent replacement of the communications relay configuration.

3.3.2 Ground Test

Ground tests conducted between each flight test validate the design before each flight test and also serve as a means for data collection. The ground test objectives will be determined based on the course of progress in the design. In other words, the ground tests will verify the operability of the specific design features of the system before each flight test.

The ground tests' success is prerequisite to conducting the corresponding flight tests. Thus, the ground test objectives shall mirror the flight test objectives. The ground tests also serve as a means for additional data collection. A ground test matrix will be documented in Chapter 4 with objectives that address the specific design preceding and following each flight test.

The data collected in the ground tests shall consist of the communications quality of each aircraft and ground station modem, the communication loss rate of each modem pair, and the data throughput of the network. These data will be used in the next chapter to characterize the capabilities of the aerial communications relay as well as the mesh network in terms of range and bandwidth capabilities, and will also be used to analyze the performance capabilities of the rover-relay and future algorithms in the context of the network.

IV. Results

4.1 Chapter Overview

Chapter 4 of this research is divided chronologically following the sequence of design, ground testing, and flight testing that was accomplished during the course of the research. Each section will discuss the qualitative findings of the corresponding phase of the research as well as document quantitative design and test results. Figure 12 outlines the organization of Chapter 4.

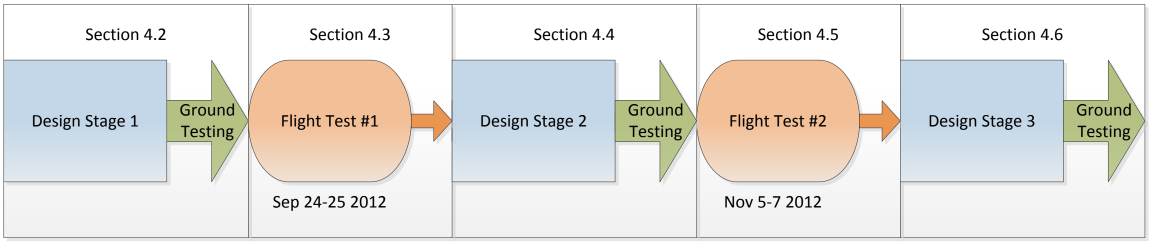


Figure 12: Organizational Flow of Chapter 4

Section 4.6 concludes the Design and Testing phase of the results and captures the final design schematic. Section 4.7 provides a summary of the overall testing results as discussed in the Methodology.

4.2 Design Stage 1 and Ground Testing

The baseline Ardupilot-equipped OWL design was accomplished after the initial design decisions were made to change from the Kestrel autopilot system to Ardupilot. This decision and design stage correspond with Step 1 of the Methodology. The researchers acquired COTS hardware to accomplish Step 1, including the Ardupilot Mega 2.0 autopilots, 2200mAh 3-cell lithium polymer batteries, 3DR modems, FrSky RC receivers and Turnigy 9x controllers. Two Sig Rascal aircraft, one electric and one gas-powered, which were already available, were equipped with Ardupilot autopilot boards, but were not fully configured for flight prior to Flight Test #1 due to parts and contractor personnel availability. The FrSky components make up the safety-pilot RC system, operating at 2.4 GHz.

4.2.1 Owl Design

The baseline Ardupilot-equipped OWL design is displayed in Figure 13 below.

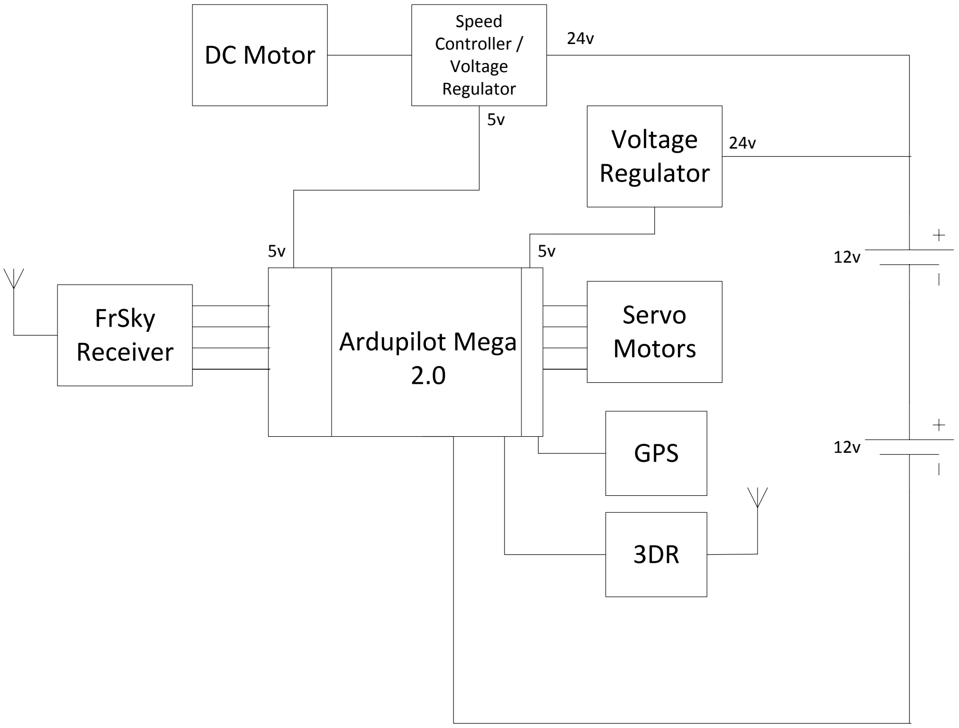


Figure 13: Baseline Ardupilot OWL Vehicle Design

Two 2200mAh 11.1v lithium polymer batteries wired in series provide power to the system. On a fresh charge, the batteries operate at over 12v and discharge down to 11v. The batteries provide 24v to the speed controller, which powers the DC motor. Two different 5v voltage regulators in parallel provide 5v to the autopilot board, which powers the FrSky receiver, 3DR modem, GPS receiver, and servo motors. The 3DR modem is used for command of the autopilot through the ground station laptop. The FrSky receiver provides a separate RC controlled safety pilot system, operating at 2.4 GHz. For the RC controller, the Turnigy 9x controller was modified to communicate through an FrSky 2.4 GHz module and was loaded with ER9x firmware. Figure 14 displays the customized safety pilot RC controller equipped with a 2.4GHz bi-directional antenna.

Figure 14: Turnigy 9x RC Controller with FrSky Radio Module

4.2.2 Ground Station Design

Lenovo Thinkpad laptops with Windows 7 64-bit operating system were used for the ground station using USB 3DR modems with 915MHz RPSMA antennas. Figure 15 displays the design of the ground station, including all the interfaces and software protocols involved.

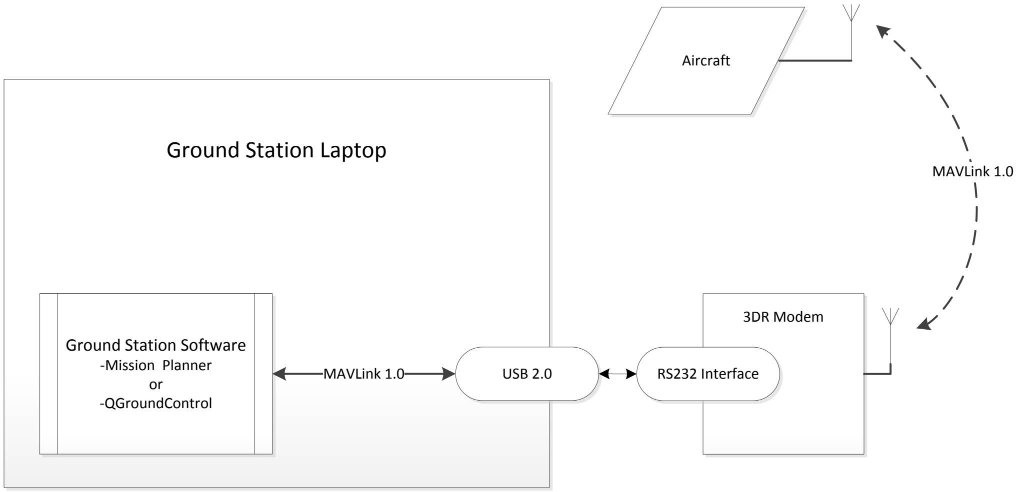


Figure 15: Ground Station Design with 3DR Modem

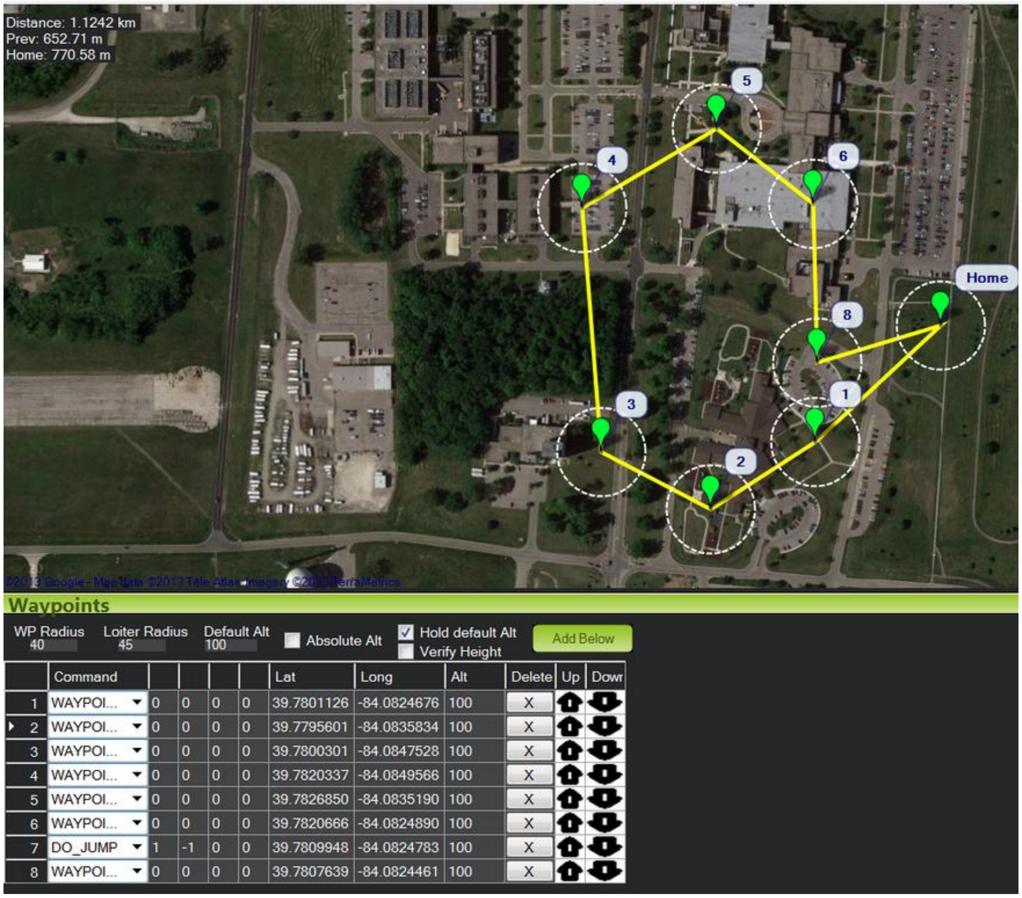
4.2.3 Ground Testing

The ground testing following Design Stage 1 consisted of firstly verifying the operability of the control surfaces of the OWLs after programming the RC control system to the appropriate servo motors of the aircraft. Then, waypoints were loaded to the Ardupilot boards using Mission Planner and the OWLs were carried along the flight circuit on the ground to verify that the waypoints were being followed by the autopilot system. The only way to verify this was to monitor the control surfaces of the aircraft to confirm that the direction of the rudder was steering towards the next waypoint after the current waypoint was reached. These ground tests were meant to simulate an actual flight as closely as possible without launching the vehicles and verify the operation of the RC and autopilot systems.

4.3 Flight Test #1

The first flight test at Camp Atterbury, Indiana was conducted on 24-25 September, 2012. The flight test objectives were to establish control gains for the Ardupilot equipped OWL, and to verify the feasibility of flying two OWLs simultaneously with two ground station computers running Mission Planner. The flight test procedures that were approved during the TRB/SRB preceding Flight Test #1 are in Appendix A.

Using the Ardupilot Mega suggested procedure for establishing and tuning gains, the gain parameters were established over the course of two days of flights. However, weather prevented the second objective from being accomplished. No safety-related incidents or vehicle crashes occurred. During the flight test, the researchers became familiarized with nuances of the Mission Planner ground station application. The most important lesson learned from the trip was how to correctly program a loop of repeating waypoints in Mission Planner.



Distance: 1.1242 km
Prev: 652.71 m
Home: 770.58 m

Waypoints

WP Radius 40 | Loiter Radius 45 | Default Alt 100 | ☐ Absolute Alt | ☑ Hold default Alt | ☐ Verify Height

	Command					Lat	Long	Alt	Delete	Up	Down
1	WAYPOI...	0	0	0	0	39.7801126	-84.0824676	100	X	⬆	⬇
2	WAYPOI...	0	0	0	0	39.7795601	-84.0835834	100	X	⬆	⬇
3	WAYPOI...	0	0	0	0	39.7800301	-84.0847528	100	X	⬆	⬇
4	WAYPOI...	0	0	0	0	39.7820337	-84.0849566	100	X	⬆	⬇
5	WAYPOI...	0	0	0	0	39.7826850	-84.0835190	100	X	⬆	⬇
6	WAYPOI...	0	0	0	0	39.7820666	-84.0824890	100	X	⬆	⬇
7	DO_JUMP	1	-1	0	0	39.7809948	-84.0824783	100	X	⬆	⬇
8	WAYPOI...	0	0	0	0	39.7807639	-84.0824461	100	X	⬆	⬇

Figure 16: Programming a Repeating Circuit of Waypoints in Mission Planner

Figure 16 displays the method for programming a repeating circuit of waypoints in Mission Planner. The intended flight path data is carried in waypoints 1-6. Waypoint 7 is a DO_JUMP waypoint with the values 1 and -1 in the first two value fields. Waypoint 8 is a dummy waypoint. With this set of waypoints, the aircraft will fly directly from waypoint 6 to waypoint 1.

4.4 Design Stage 2 and Ground Testing

Design Stage 2 corresponds with Step 2 and Step 3 of the methodology; therefore, the goal of Design Stage 2 was to establish a communications relay and to implement the rover-relay algorithm at the ground station. The 3DR modems selected in Design Stage 1 proved operational in the first flight test, but lack a repeater mode configurability to provide a communications relay. Digi XBee 900 modems were purchased to replace the 3DR modems to accomplish step two of the design phase. The XBee modems have multiple firmware packages they can accommodate, including Digi Pro 900 and DigiMesh 900 firmware. The Digi Pro firmware provides customization options in serial routing while the DigiMesh firmware provides the capability of configuring a mesh network. With the cheap unit cost of approximately $50 combined with the extent of firmware configurability, the Digi XBee modems were the best choice to accomplish the remaining design objectives.

Lieutenant Timothy Shuck developed a custom software deployment of QGroundControl using the Qt development environment to implement the rover-relay algorithm at the ground station. The code was designed to autonomously generate waypoints for the relay vehicle based on the location of the ground station and rover vehicle and upload them to the aircraft during flight [16].

4.4.1 Sig Rascal Relay Design

During this design stage, the gas-powered Sig Rascal vehicle was fully prepared for flight and was configured as the relay vehicle due to its long flight endurance of 45 minutes. The aircraft was configured with a 2-stroke CCRPRO GP26R 26.0cc two-stroke engine with a Walbro carburetor. An Ardupilot Mega 2.0 and FrSky RC receiver were already installed before Flight Test #1. A voltage sensor was added to monitor the battery voltage of the aircraft through the RC controller during flight. The sensor data is transmitted to the safety pilot's FrSky module mounted on the Turnigy ER9x controller.

Two XBee modems wired back-to-back on different hopping channels using the Digi Pro firmware provided the basis for the communications relay design. Figure 17 displays the design schematic that was installed on an Ardupilot-equipped Sig Rascal vehicle.

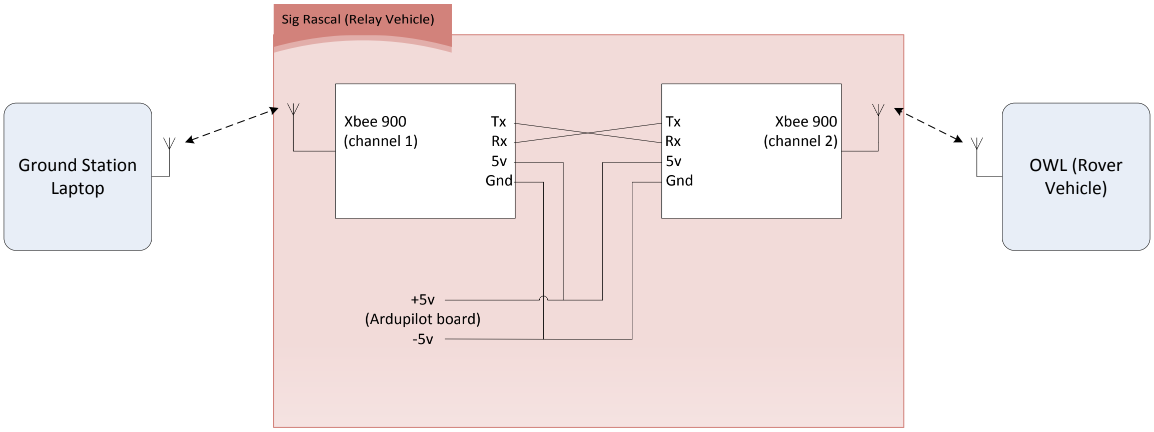


Figure 17: XBee Pro 900 Communications Relay Schematic

An XBee 900 modem programmed to channel 1 is wired to an identical modem programmed to channel 2, which communicates with the OWL rover vehicle. This wiring scheme allocates each modem as a repeater of the other so that each packet of data received by one is transmitted by the other, and vice versa. Every packet sent from the ground station must pass through the relay vehicle in order to transmit to the rover, and each packet sent from the rover must also pass through before being received at the ground station.

The gas-powered Sig Rascal relay vehicle design is depicted in Figure 18.

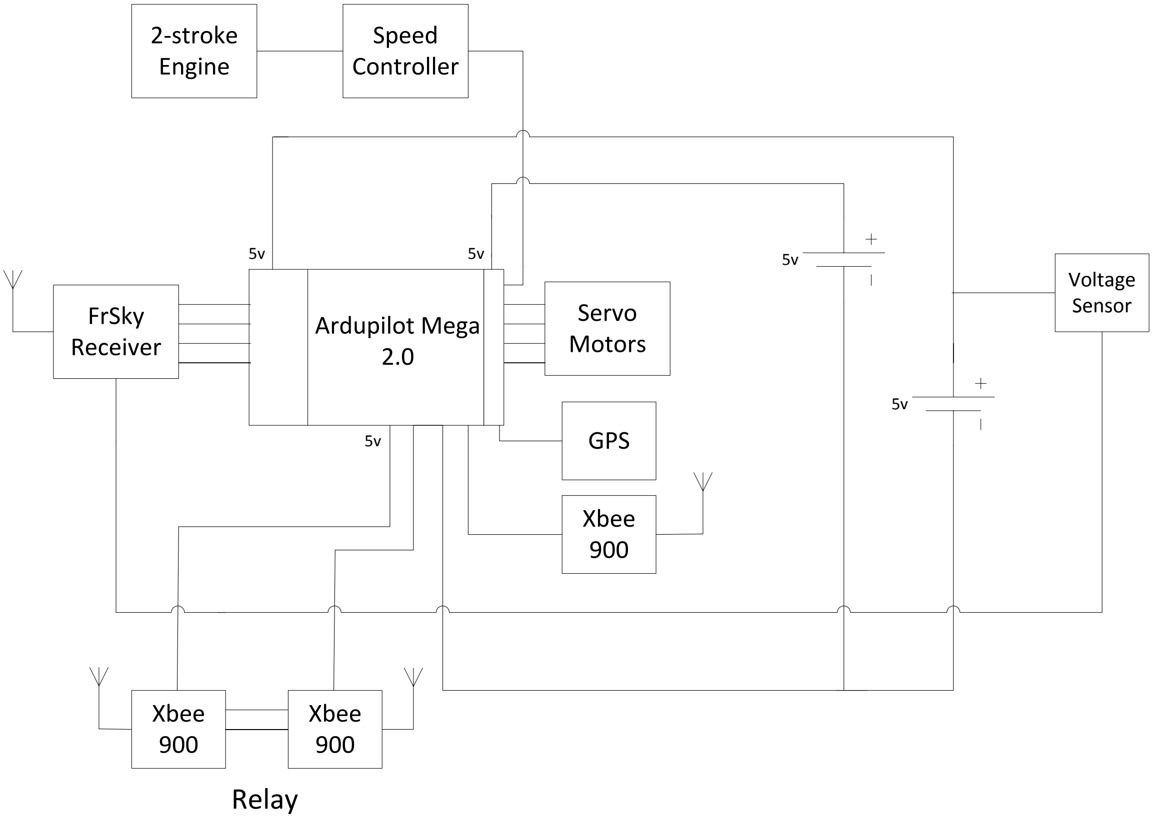


Figure 18: Sig Rascal Relay Vehicle Design

The Sig Rascal design includes three XBee modems: the two for the relay, and one for the command of the vehicle. The relay modems are programmed to channels 1 and 2, while the Ardupilot's modem is programmed to channel 1. Since channel 1 is used for the Sig Rascal's autopilot and half of the communications relay, a procedural sequence of establishing modem connections must be practiced to ensure the correct employment of the rover-relay system; this procedure is documented in Appendix A – Flight Test #2.8. Instead of using two 11.1v 2200mAh lithium polymer batteries in series, the Sig Rascal design employs two 5v nickel-metal hydride batteries, which power each side of the Ardupilot board separately.

4.4.2 OWL Redesign

Besides the replacement of the 3DR modem with the XBee 900 modem, the OWL was fitted with several other design modifications. An FrSky sensor hub was installed in order to monitor the Lithium Polymer battery voltage in flight like on the Sig Rascal vehicle. The sensor hub is capable of utilizing many different sensors to provide feedback to the safety pilot through the FrSky module; these sensors include voltage sensors, accelerometers, thermometers, etc. [17]. A 5.8 GHz video modem was also installed in order to transmit video from the nose camera to a Yellow Jacket 5.8 GHz receiver on the ground, which was connected to a monitor and DVD recorder. The second iteration of the OWL vehicle configuration is captured in Figure 19.

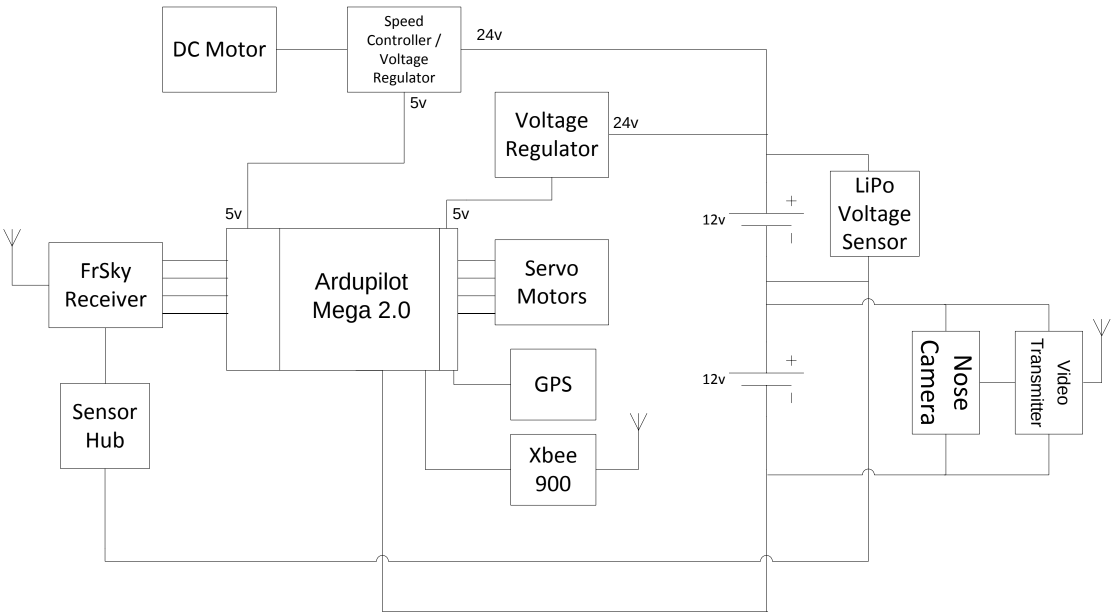


Figure 19: OWL Rover Vehicle Design

Figure 20 shows a photograph of the internal avionics bay of the OWL, which displays one of the 2200mAh Lithium Polymer batteries, the Ardupilot Mega 2.0 board, the video modem, and the FrSky receiver behind it.

Figure 20: Reconfigured OWL with Ardupilot Mega 2.0, FrSky RC Receiver, and Video Modem

4.4.3 Ground Station Design

For Design Stage 2, the same Lenovo laptops were employed for the ground station; however, the 3DR USB modems were replaced with XBee 900 modems attached to XBee USB explorer boards, which include a USB-RS232 chipset. 915MHz RPSMA antennas were attached to the XBee 900 modems. There were three different XBee ground station modems used: two modems programmed to channel 1 for communication with the rover OWL through the relay and with the Sig Rascal, and one programmed to channel 2 for direct communication with the OWL. Depending on the flight test mission, these modems were used with either one or both laptops. In the rover-relay configuration, both laptops were used, each with a single modem programmed to channel 1; one laptop was used to control the relay vehicle (the Sig Rascal), and the other was used to control the rover vehicle (the OWL). Also, Lieutenant Shuck's customized QGroundControl build was added to the ground station laptops. Figure 21 depicts the redesigned ground station.

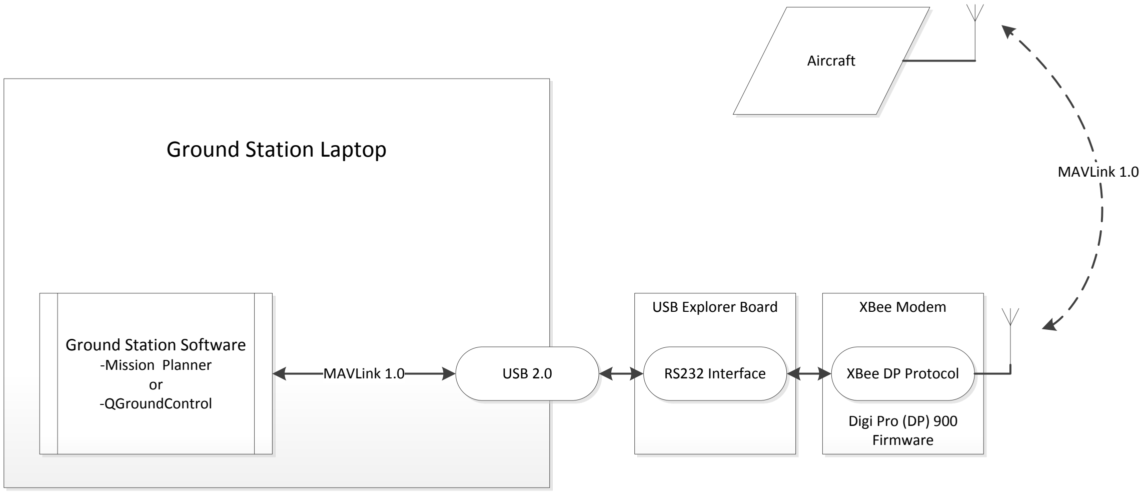


Figure 21: Ground Station Design with XBee Modem

4.4.4 Ground Testing

Prior to Flight Test #2, ground testing was achieved to validate the operation of the communication relay as well as the functionality of Lieutenant Shuck's rover-relay QGroundControl implementation.

The communications relay test involved the Sig Rascal, OWL, and ground station using Mission Planner. To verify the operation of the communications relay, waypoints and parameters were uploaded to the OWL using an XBee 900 modem at the ground station programmed to channel 1, forcing a bridged connection through the Sig Rascal's relay modems. The OWL was walked around outdoors to verify the telemetry was received at the ground station. The rover OWL was observed to be fully operational using the communications relay onboard the Sig Rascal. Also, the operation of the Sig Rascal was verified using an XBee 900 modem programmed to channel 1.

To validate the operation of the rover-relay QGroundControl implementation, two ground station laptops were employed as they would be configured during Flight Test #2. Instead of controlling both the relay and rover aircraft from a single laptop running QGroundControl, two separate laptops were used. The rover was controlled using a laptop running Mission Planner, and the relay was controlled using the custom build of QGroundControl, which employed the rover-relay algorithm to autonomously generate waypoints and upload them to the relay vehicle. Figure 22 illustrates the overall ground control station architecture for a rover-relay flight.

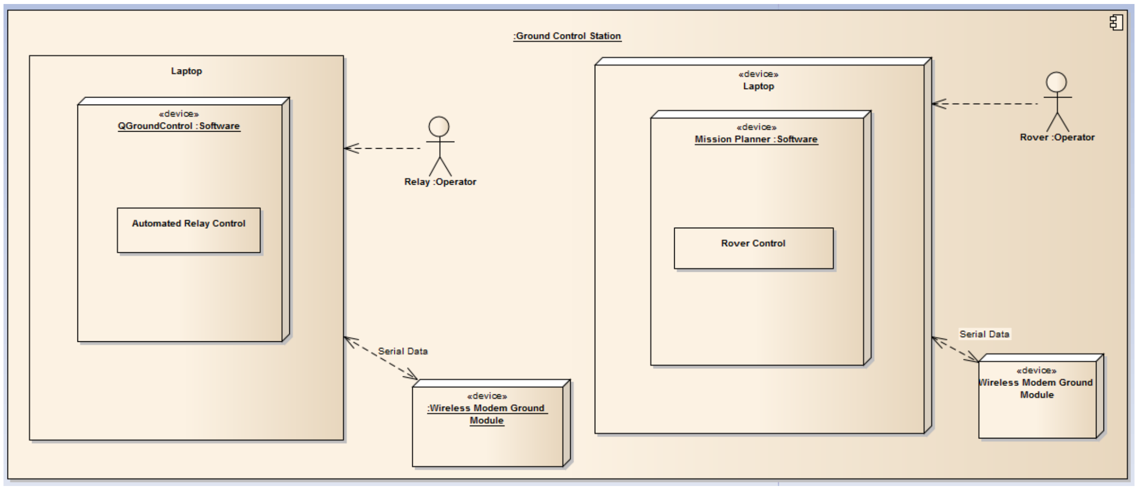


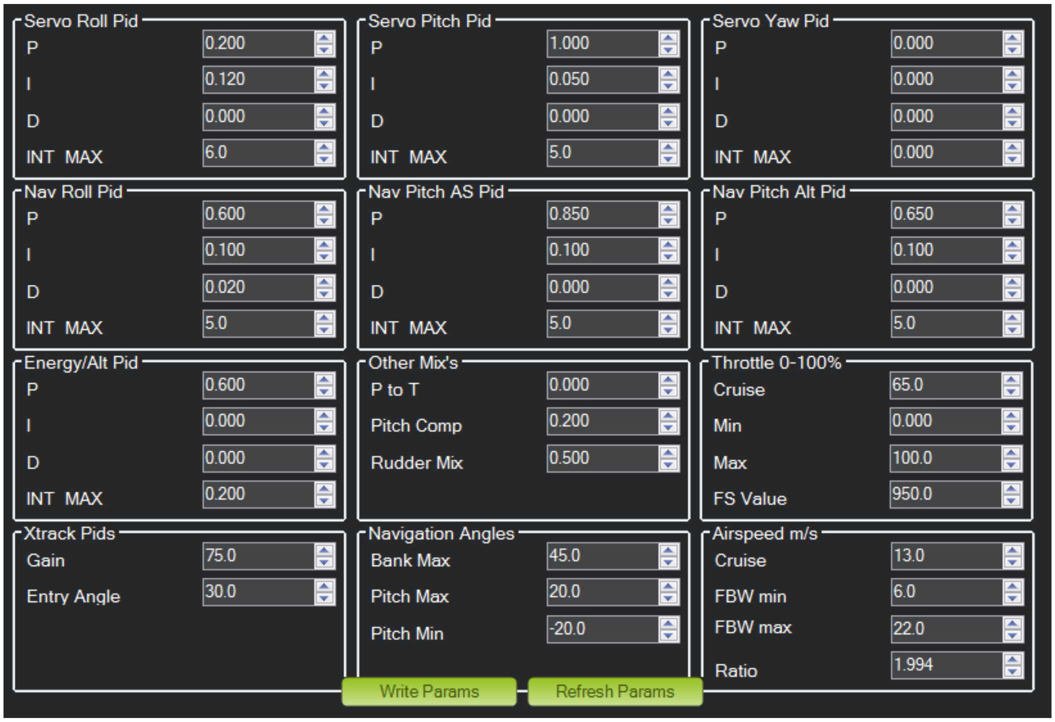
Figure 22: Flight Test #2 Ground Station Architecture

The Sig Rascal and OWL vehicles were walked around outdoors to simulate a flight pattern where the relay was positioned between the rover and ground station. It was validated that the custom QGroundControl application generated waypoints for the relay aircraft based on the midpoint between the rover aircraft and ground station, updating as the rover position changed in real-time.

4.5 Flight Test #2

The second flight test was accomplished at Camp Atterbury, Indiana on 5-7 November, 2012. The flight test objectives were to tune gain for the Ardupilot-equipped Sig Rascal, verify the operability of QGroundControl, characterize the range of the XBee modems, verify the operability of the communications relay, and to verify the operability of the autonomous waypoint navigation of the rover-relay custom QGroundControl application. The flight test procedures that were approved during the TRB/SRB preceding Flight Test #2 are in Appendix A.

The Sig Rascal was successfully tuned during the first day of flight testing. Also, the OWL control gains were modified to maintain a more consistent throttle setting in flight. The final gain parameters are captured in Figure 23 and 24.



Group	Parameter	Value
Servo Roll Pid	P	0.200
	I	0.120
	D	0.000
	INT MAX	6.0
Servo Pitch Pid	P	1.000
	I	0.050
	D	0.000
	INT MAX	5.0
Servo Yaw Pid	P	0.000
	I	0.000
	D	0.000
	INT MAX	0.000
Nav Roll Pid	P	0.600
	I	0.100
	D	0.020
	INT MAX	5.0
Nav Pitch AS Pid	P	0.850
	I	0.100
	D	0.000
	INT MAX	5.0
Nav Pitch Alt Pid	P	0.650
	I	0.100
	D	0.000
	INT MAX	5.0
Energy/Alt Pid	P	0.600
	I	0.000
	D	0.000
	INT MAX	0.200
Other Mix's	P to T	0.000
	Pitch Comp	0.200
	Rudder Mix	0.500
Throttle 0-100%	Cruise	65.0
	Min	0.000
	Max	100.0
	FS Value	950.0
Xtrack Pids	Gain	75.0
	Entry Angle	30.0
Navigation Angles	Bank Max	45.0
	Pitch Max	20.0
	Pitch Min	-20.0
Airspeed m/s	Cruise	13.0
	FBW min	6.0
	FBW max	22.0
	Ratio	1.994

Figure 23: Gain Parameters for OWL Platform

Group	Parameter	Value
Servo Roll Pid	P	1.000
	I	0.200
	D	0.000
	INT MAX	5.0
Servo Pitch Pid	P	1.100
	I	0.250
	D	0.000
	INT MAX	5.0
Servo Yaw Pid	P	0.500
	I	0.100
	D	0.001
	INT MAX	5.0
Nav Roll Pid	P	1.200
	I	0.100
	D	0.100
	INT MAX	5.0
Nav Pitch AS Pid	P	0.900
	I	0.300
	D	0.000
	INT MAX	5.0
Nav Pitch Alt Pid	P	0.650
	I	0.100
	D	0.000
	INT MAX	5.0
Energy/Alt Pid	P	0.750
	I	0.350
	D	0.000
	INT MAX	4.0
Other Mix's	P to T	0.100
	Pitch Comp	0.200
	Rudder Mix	0.500
Throttle 0-100%	Cruise	45.0
	Min	40.0
	Max	100.0
	FS Value	950.0
Xtrack Pids	Gain	30.0
	Entry Angle	30.0
Navigation Angles	Bank Max	45.0
	Pitch Max	20.0
	Pitch Min	-20.0
Airspeed m/s	Cruise	18.0
	FBW min	6.0
	FBW max	22.0
	Ratio	1.994

Figure 24: Gain Parameters for Sig Rascal Platform

The basic operability of QGroundControl was verified in a simple flight test using a single ground station and a single OWL. Waypoints could be uploaded to the aircraft using QGroundControl and the preloaded flight path was followed by the autopilot. Next a test to change the waypoints in flight using QGroundControl was attempted; however, instead of flying the in-flight updated waypoints the aircraft would “return to launch” and loiter. “Return to launch” means the UAV will fly to the preloaded waypoint 0. This waypoint is the default rally point for the Ardupilot to navigate toward if communication is lost, a failsafe condition is triggered, or if the return to launch flight mode is selected. It was discovered during day two of flight testing that the problem was due to a misunderstanding of the proper order of events necessary to confirm new waypoints on the Ardupilot. In order for Ardupilot to accept a new set of waypoints, the new waypoints must be written to the UAV, read from the UAV to confirm the waypoints were transmitted correctly, and written once more to activate the new waypoints. QGroundControl was designed to fly a path of pre-loaded waypoints without changing them mid-flight so the update waypoints process is cumbersome. Mission Planner was designed to more easily enable updating waypoints in flight. Mission Planner programmers developed a specialized command, not contained in standard MAVLink protocol that sets any user specified waypoint uploaded to the UAV to be the current navigation objective of the autopilot.

The range of the XBee 900 modems was measured at 0.6 miles from ground station to aircraft with the OWL flying at a 300ft altitude. Intermittent communications link was demonstrated between 0.6 and 0.88 miles. Beyond 0.88 miles, communication was never recovered, and inside 0.6 miles, communication was never lost.

The operation of the communication relay was only partially validated. The communication link exhibited severe packet loss in flight, which restricted the operability of updating waypoints and parameters to the rover vehicle mid-flight. Telemetry data from the rover, however, was received at the ground station without interference. Between test flights, one of the relay modems on the Sig Rascal was changed from a ¼ wave wire antenna to an RPSMA antenna to increase gain. This configuration change resulted in better functionality of parameter uploads during flight, but still did not provide communication quality great enough to upload waypoints to the rover aircraft in flight.

The last flight test objective was to verify the operation of the custom QGroundControl rover-relay implementation, which was unsuccessful due to the aforementioned deficiency discovered in QGroundControl. The custom code was capable of generating relay waypoints based on the rover-relay algorithm, and was successful in uploading them to the relay aircraft, but the waypoints were not followed by the vehicle.

4.6 Design Stage 3 and Ground Testing

During the Flight Test #2, it became apparent that the OWL power bus wiring design was faulty and causing anomalies in the behavior of the autopilot. This anomalous behavior included sudden power cycles in the autopilot, causing the system to restart spontaneously during pre-flight preparations. In the field, the design was modified to remove a 5v switching regulator, which had been wired in parallel with the Castle speed controller's 5v regulator. Instead of powering either side of the Ardupilot Mega 2.0 board independently, a jumper was installed so that the power was shared on both sides from the regulated 5v of the speed controller. This final design of the OWL is displayed in Figure 25.

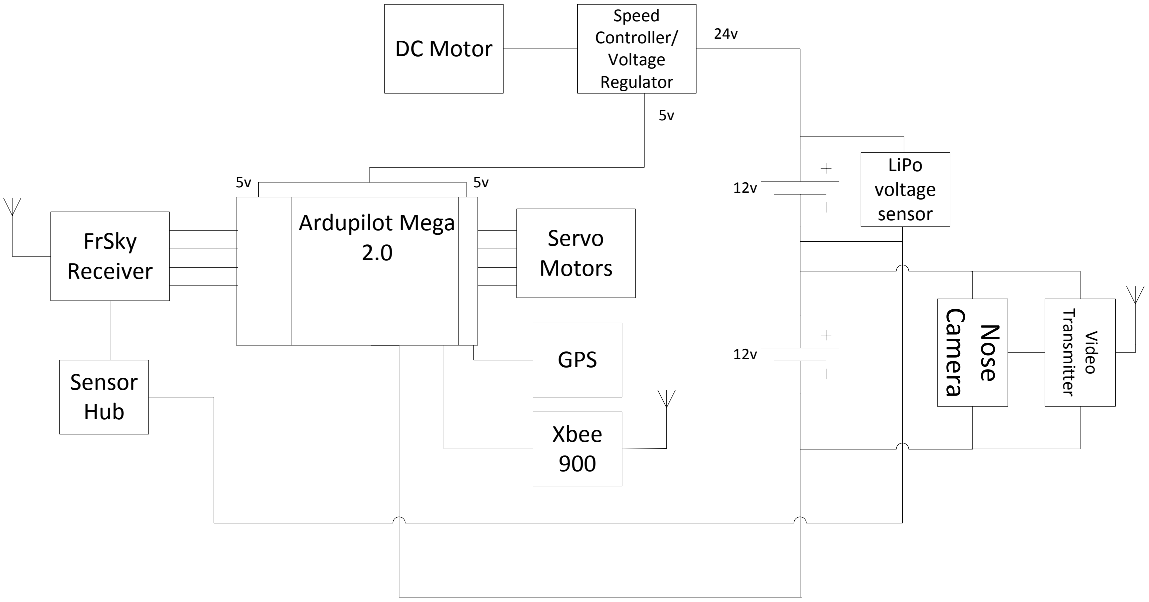


Figure 25: OWL Design with Redesigned Power Bus

This redesign of the OWL proved more robust than before. The single regulated 5v power source eliminated anomalous behavior in the OWL's autopilot system restarting spontaneously. This configuration was flown during the end of the second day of flight testing, and also during the third day without any observed power anomalies.

After the flight test, ground testing was accomplished to identify the cause of packet loss through the communications relay. The process of elimination was used by simply unplugging each electrical component of the Sig Rascal one by one, while observing the communications quality in Mission Planner to the OWL through the relay modems. It was revealed that it was not a single component generating RF noise, but the proximity of the three modems in the Sig Rascal that was causing packet loss. The original layout of the relay modems resulted in RF saturation in the 900MHz region and packet loss. The modem antennas were spaced out to the right, left, and bottom of the vehicle, replacing the ¼ wave wire antennas with RPSMA antennas.

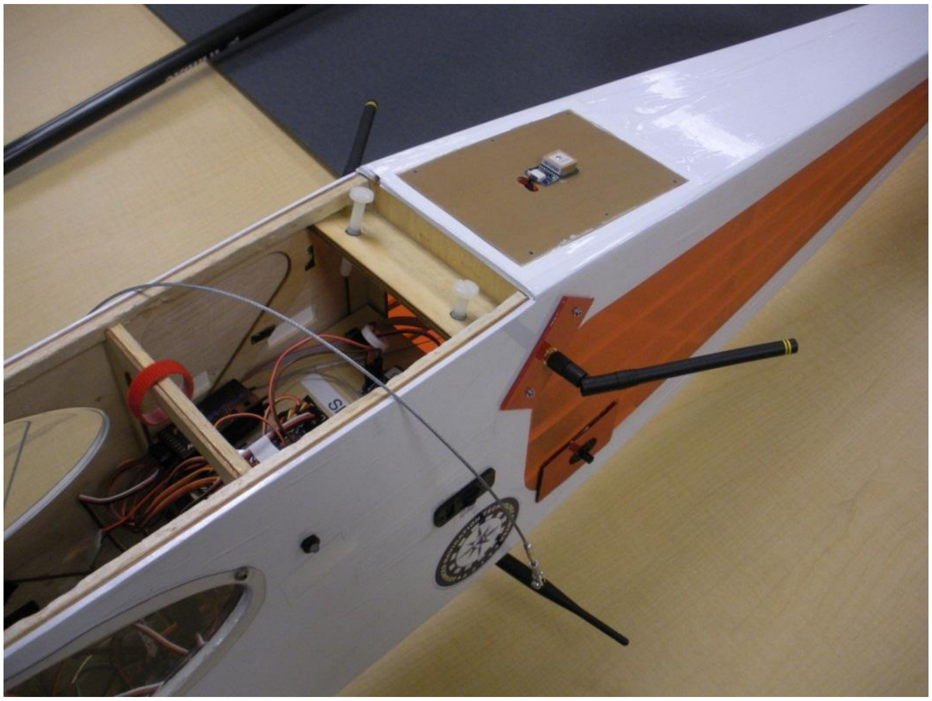


Figure 26: RPSMA Antenna Placement on Sig Rascal Relay Vehicle

Figure 26 shows the redesigned layout of the XBee 900 antennas on the Sig Rascal relay vehicle. To achieve this spacing, XBee 900 modems with an U.FL antenna connector were used. Figure 27 displays the internal Sig Rascal layout.

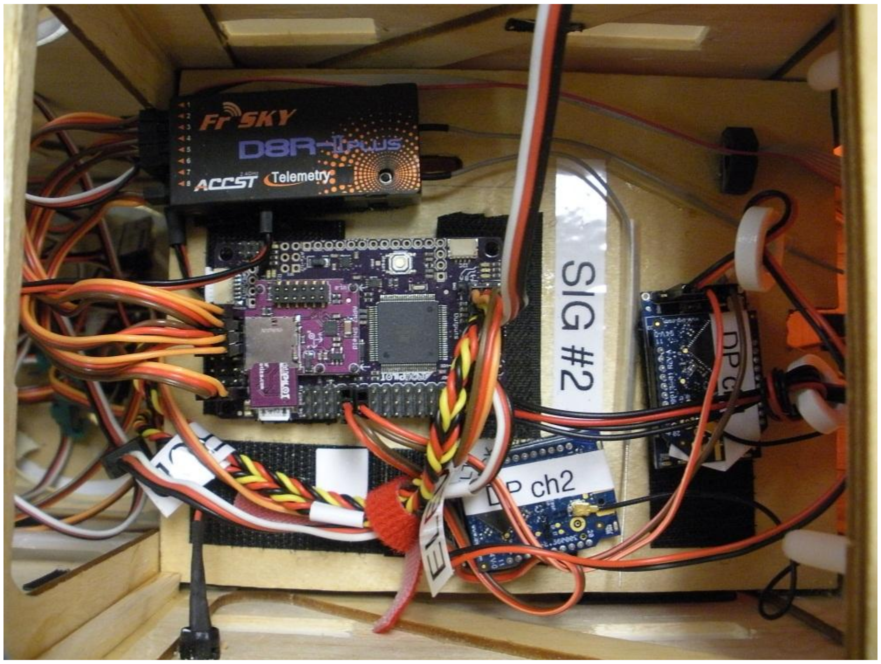


Figure 27: Sig Rascal Avionics and Relay Modems

The Ardupilot Mega 2.0 board, FrSky receiver, and XBee 900 relay modems can be seen. The XBee 900 modem programmed to channel 1 for autopilot control is stationed in the below inner fuselage area out of frame of the photograph. The U.FL coaxial cables are wired to the RPSMA mounts on the sides and bottom of the aircraft as displayed in Figure 26.

The next step of Design Stage 3 was to configure the XBee modems with the DigiMesh 900 firmware and test the functionality of a relayed link from the ground station to a rover aircraft, which corresponds with Step 4 of the methodology. This step was accomplished by simply removing the modems already installed in the OWLs and flashing them with DigiMesh 900 firmware, verifying the correct baud rate of 57,600 kbps. The results of the DigiMesh modem framework are further discussed in the next section.

4.6.1 Ground Testing

The ground testing objectives following Design Stage 3 were to verify the operation of the redesigned communications relay, measure the effective communications extension to the rover through the redesigned relay, and to verify the operation of relayed communications with the DigiMesh modems.

The signal quality of the relayed rover signal through the original communications relay was measured to vary from 60-80% with the ground station at a distance of 3 meters from the Sig Rascal and the rover OWL vehicle placed 2 meters from the Sig Rascal. The signal quality was increased to 90-100% after installing the U.FL modems with RPSMA antennas mounted to the outside of the Sig Rascal. This communications quality displayed in Mission Planner is simply an average of the ratio of successfully transmitted packets to dropped packets. The average is trailed for the duration of an established connection. A communications quality of 90-100% in the indoors environment where the test was held is as good as a direct modem-to-modem link at the same distance; therefore, the redesigned antenna layout of the communications relay was no longer degrading the link quality to the rover from the ground station.

The next test objective was to measure the effective extension of the redesigned communications relay on the ground. This test was accomplished by seating the ground station laptop outdoors on a table with the antenna approximately 1 meter off the ground, programmed to channel 1. The Sig Rascal was powered on near the ground station and a connection was established between the ground station and the Sig Rascal. Next, the Sig

Rascal was walked away from the ground station until the modem link was broken. Then, the Sig Rascal was walked back within range about 20 feet to allow the connection to recover. Next, the OWL was powered on near the Sig Rascal and a new link was established between the ground station and the OWL, using a modem programmed to channel 1 at the ground station. Then, the OWL was walked back towards the ground station while the link quality was monitored. The link to the OWL was broken exactly as the OWL crossed the ground station's position, meaning an exact 1:1, or 100% communications extension was achieved. Figure 28 offers a visual depiction of this ground test. With the demonstrated 100% communications extension, it can be projected that the operational communications extension is 0.6 miles from the relay to rover aircraft in flight, based on the measurements of Flight Test #2. This extension would provide a range of 1.2 miles from the ground station to the rover vehicle in a rover-relay flight.

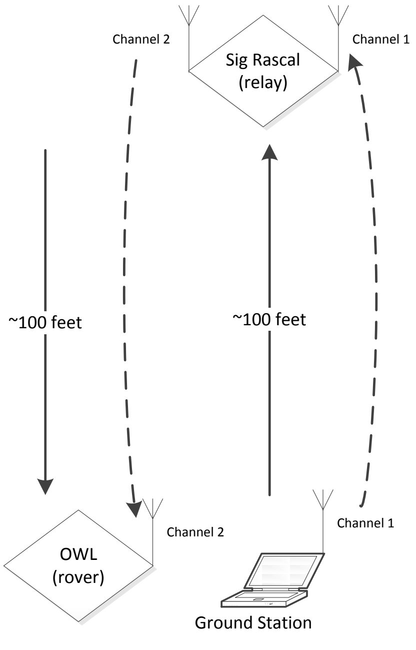
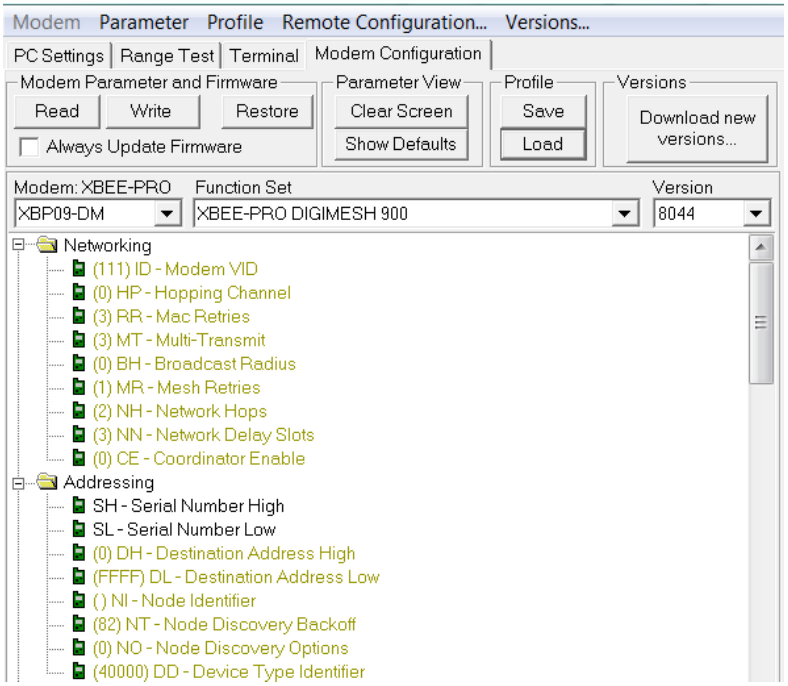


Figure 28: Communications Extension Ground Test

The last ground test objective for Design Stage 3 was to verify rover-relay communications through the DigiMesh modem framework. For these tests, QGroundControl was utilized due to its ability to control multiple aircraft simultaneously [10]. The first test verified the self-healing capability of the DigiMesh modem firmware. An OWL aircraft was walked out of range of the ground station, all modems programmed to the same hopping channel (channel 1). Then, an OWL was powered on between the ground station and the out-of-range rover OWL to act as a relay; both aircraft acquired a connection in QGroundControl, thus verifying the bridged connection across the relay node. Telemetry was visible for both aircraft, but waypoints could not be successfully passed to the rover OWL.

The next step was to find a configuration that would accommodate a complete connection between the ground station and rover OWL. Multiple adjustments were made to the DigiMesh firmware parameters to optimize the network configuration for the rover-relay employment with three chained modems (ground station-relay-rover). The actual routing protocol of the DigiMesh firmware is inaccessible to the researcher for redesigning or adjusting the algorithm [18]. Therefore, the configurability options are limited to adjusting the number of retransmissions allowed after a failed packet transmission, the number of network hops allowed, and similar parameters, which are visible in Figure 29. These options were adjusted to maximize the signal quality to the rover vehicle in the chained ground station-relay-rover network. The best signal quality accomplished was 60% between the ground station and the rover, which was accomplished with all three modems programmed with the parameters displayed in Figure 29.



Modem: XBEE-PRO XBP09-DM
Function Set XBEE-PRO DIGIMESH 900
Version 8044

Networking
- (111) ID - Modem VID
- (0) HP - Hopping Channel
- (3) RR - Mac Retries
- (3) MT - Multi-Transmit
- (0) BH - Broadcast Radius
- (1) MR - Mesh Retries
- (2) NH - Network Hops
- (3) NN - Network Delay Slots
- (0) CE - Coordinator Enable

Addressing
- SH - Serial Number High
- SL - Serial Number Low
- (0) DH - Destination Address High
- (FFFF) DL - Destination Address Low
- () NI - Node Identifier
- (82) NT - Node Discovery Backoff
- (0) NO - Node Discovery Options
- (40000) DD - Device Type Identifier

Figure 29: Optimal DigiMesh Parameters for Rover-Relay in X-CTU application

The nature of the Ardupilot system is that the aircraft telemetry is constantly passed back to the ground station. It became apparent to the researcher that the center node, the relay, was burdened by constantly transmitting the relay aircraft telemetry as well as the rover aircraft telemetry back to the ground station. When waypoints were attempted to be sent to the rover aircraft, QGroundControl would attempt to retransmit five times before timing out. To check if the waypoint transmission was successful, the rover waypoint data would be refreshed from the ground station, which synchronizes the waypoint list from the aircraft Ardupilot board. The process of updating waypoints from the ground station to the rover aircraft in a chained network is displayed in Figure 30. This figure compares this procedure with the DigiMesh framework to the previously established Digi Pro communications relay. With the previous Digi Pro design, each modem is less burdened at any given point. With DigiMesh, the relay aircraft's modem (the middle node) is constantly receiving and transmitting data from both end nodes.

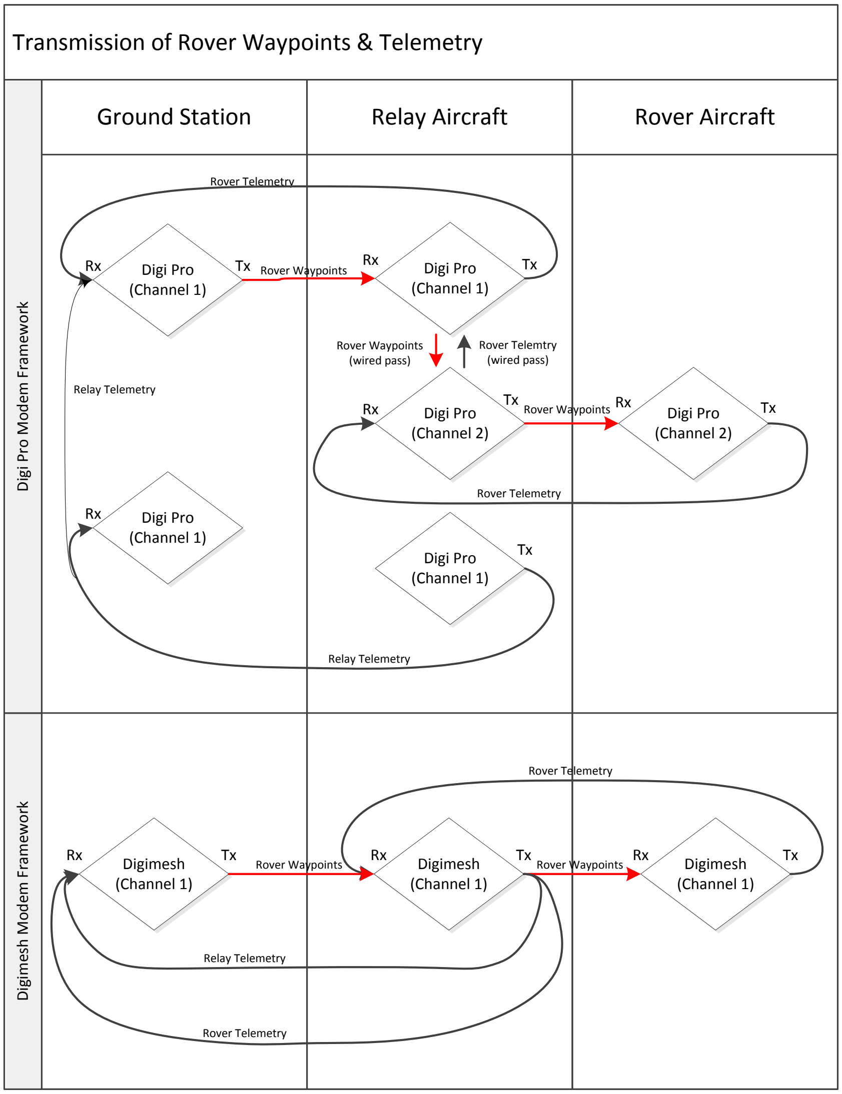


Figure 30: Comparison of Digi Pro and DigiMesh Modem Frameworks when Transmitting Rover Waypoints and Relay and Rover Telemetry to Ground Station

Even with the optimized DigiMesh parameters, the waypoints were never properly updated from the ground station waypoint list with the DigiMesh network. The telemetry feed was never suspended during ground testing, but updates to the aircraft from the ground station in the form of parameters or waypoints were unsuccessful. This could likely be due to the confusion of CTS and RTS packets being transmitted through the relay node from the two end nodes (rover and ground station), as described by Li and others in their research with 802.11 modems [12]. In any case, the root problem is limited network capacity for the Ardupilot application of chaining aircraft in a rover-relay implementation of the mesh network. Modems with higher network throughput in a mesh setting are required to completely implement an ad-hoc framework for Ardupilot.

4.7 Summary of Flight and Ground Test Results

The overall flight test results can be summarized in the Table 2, which corresponds with the flight test objectives established in the Methodology. Flight Test #3 was never accomplished due to the overall research schedule slipping into the winter months. The aircraft cannot be flown in sub-freezing temperatures. The mesh network modem configuration for the system was not successfully implemented in preparation for the flight test even if the schedule allowed for Flight Test #3.

Table 2: Flight Test Results Matrix – Summary of Test Objectives and Results.

Flight Test	**Test Objectives**	**Result**
Flight Test #1	Establish control gains for Ardupilot Mega in OWL and verify stable flight of vehicle	Success
	Validate functionality of Mission Planner for single and multi-vehicle operation	Success
Flight Test #2	Establish control gains for Ardupilot Mega in Sig Rascal and verify stable flight of vehicle	Success
	Verify operability of QGroundControl	Partial
	Verify operability of communications relay in flight	Success
	Determine direct communications range of autopilot system with current modems	Success (0.60-0.88 miles)
	Verify operability of rover-relay algorithm implementation at the ground station	Partial
Flight Test #3	Verify operability of rover-relay with mesh modem configuration	Incomplete

The ground test objectives were developed in preparation for each flight test as described in the Methodology. The flight test objectives were based on the design objectives of the Methodology, and the ground test objectives were based on proving the system capability to fulfill each flight test objective.

Table 3: Ground Test Results Matrix – Summary of Test Objectives and Results.

Ground Test	Test Objectives	Result
Ground Test #1	Verify operation of RC controller and autopilot system	Success
Ground Test #2	Verify operation of communications relay	Success
Ground Test #3	Verify operation of autonomous waypoint generation with custom QGroundControl code	Success
Ground Test #4	Verify operation of reconfigured communications relay with RPSMA antennas	Success
Ground Test #5	Determine operational distance of communications extension through relay	Success
Ground Test #6	Verify self-healing quality of DigiMesh modem configuration for rover-relay	Success
Ground Test #7	Verify operation of rover-relay communications with mesh network configuration to include ground station-rover link	Failure

Although the rover-relay QGroundControl implementation was never flight tested, Lieutenant Shuck verified its operation through hardware-in-the-loop (HIL) simulation after modifying the code after Flight Test #2 [16]. Therefore, all requirements described in the Methodology were satisfied by the design and demonstrated in flight and ground testing. However, the mesh network communications framework was not able to satisfy the design requirements.

V. Conclusions and Recommendations

5.1 Chapter Overview

Chapter 5 discusses the conclusions of the research accomplished as well as the researcher's recommendations for future work on the OWL system. Section 5.2 discusses the research accomplishments in context with the initial research questions, and also revisits the challenges faced in accomplishing the research. Section 5.3 provides recommendations for future research to further the OWL capability as a SUAS test bed at AFIT. Lastly, Section 5.4 summarizes the thesis and examines the highest level accomplishments framed by the problem statement discussed in Chapter 1.

5.2 Retrospective and Challenges

The primary research question presented in Chapter 1 was: what small unmanned airborne system communications architecture supports cooperative control through a COTS hardware and software configuration? With the constraints of the research, both in cost and availability of hardware, a system was designed using the RQ-11 Raven and Sig Rascal airframes combined with Ardupilot Mega 2.0 autopilots, FrSky RC receivers, and Digi XBee 900 modems that was demonstrated to be capable of operating in a rover-relay configuration with autonomous relay navigation.

The challenges and limitations discussed in Chapter 1 proved to be influential to the progress of the research. As with any experimental research, hardware purchasing and availability, airspace scheduling, and weather all constrained progress.

There were two unforeseen limitations that impacted the research: unimplemented MAVLink commands in QGroundControl and the proprietary routing protocol of the DigiMesh 900 mesh networking firmware. The first of these impacted Lt Shuck's portion of the research and resulted in the custom QGroundControl build failing to auto-navigate the relay aircraft during Flight Test #2. The code was later modified so that the aircraft would fly to the appropriate auto-generated waypoint, but was never proven in flight test [16]. The network throughput capability of the XBee 900 modems with DigiMesh firmware proved insufficient for the Ardupilot system. Within a mesh network of only two aircraft with a single ground station, the telemetry of a rover aircraft (beyond LOS from the ground station) was received at the ground station, but parameters and waypoints could not successfully be uploaded to the aircraft.

5.3 Recommendations for Future Research

Now that an Ardupilot-based system has been successfully designed with a functioning communications relay and a rover-relay algorithm implemented at the ground station, there are two immediate goals to be suggested for future research. The first is to establish a functioning mesh network onboard the existing aircraft. Purchasing more capable hardware would be the most risk-free and expedient method for accomplishing this. There are several 2.4 GHz networking solutions with much higher throughput and range capabilities than the XBee 900s. An example is the Persistent Systems Wave Relay™ system pictured below in Figure 31.

Figure 31: Wave Relay Sector Antenna Array Router [19]

The Wave Relay has already been utilized and flown by researchers at the Naval Postgraduate School who are also using a customized deployment of QGroundControl to implement their custom cooperative control algorithms. This system boasts a range of 10 miles, a throughput of 37 Mbps, and can be used with any 802.11 2.4 GHz receiver [19]. This system would be more than capable of sustaining a multi-aircraft mesh network, and also carrying video data within the same network as the autopilot data, which would limit the number of modems onboard the aircraft and vastly simplify the system.

The second suggested goal for future research is to delve into the Arduino development environment to utilize the onboard microprocessing capabilities of the Ardupilot Mega board to implement algorithms and/or custom commands directly from the aircraft. The accomplishment of baselining an Ardupilot-based SUAS for cooperative control provides the capability to utilize onboard microprocessing that the previous Kestrel system did not.

A third suggested research topic that has much potential with this system is the concept of communications-aware autonomous navigation. This idea is the topic of the article "Cognitive Agent Mobility for Aerial Sensor Networks" by Kai Daniel et al. Daniel, which was discussed in Chapter 2 [11]. In the rover-relay cooperative application, communication awareness would be very useful to incorporate into the control algorithm. If a single operator was to launch two aircraft in the rover-relay configuration to extend communication LOS, he or she would have to constantly monitor the range of the aircraft to avoid flying out of LOS. If communications awareness was included in the algorithm, the autopilot system would be capable of preventing the aircraft from leaving communications range and potentially realign each aircraft to maximize communications range beyond the simple rover-ground station midpoint. Furthermore, this layer of control in the system could be broadened into the implementation of many other cooperative control applications such as flocking.

5.4 Summary

This research concluded that a SUAS using the Ardupilot autopilot system is a capable test bed system for implementing cooperative control algorithms. The rover-relay is perhaps the simplest cooperative control implementation between multiple aircraft, but it responds to the problem statement that frames this research: the necessity for beyond LOS communications for small hand-launched UAS. Future research will expand on the system to implement other cooperative control algorithms that fill other capability gaps for SUAS in the United States Air Force.

Appendix A. Flight Test Procedures

Flight Test #1 (24-25 September 2012)

1. Preflight testing (completed at AFIT and in field)
    a. Communication check (initial)
    b. Control Surface check
    c. Trim Radio and save settings
    d. Communication check (distance)
2. In Flight Testing With Mission Planner
    a. OWL_A1 & OWL_A2
        i. Zero Sensors
        ii. Set Fail Safe Parameters
        iii. Trim Radio
        iv. Load Waypoints
        v. Launch OWL_A*
        vi. RC Pilot Flight
            1. Adjust Trim
        vii. Engage Autopilot
            1. Adjust Gains (as necessary)
        viii. RC Pilot Landing
        ix. Group Discussion Observations
    b. Sig Rascal_P1 (Petrol) & Sig Rascal_E1 (Electric)
        i. Zero Sensors
        ii. Set Fail Safe Parameters
        iii. Trim Radio
        iv. Load Waypoints
        v. Launch Rascal_*
        vi. RC Pilot Flight
            1. Adjust Trim
        vii. Engage Autopilot
            1. Adjust Gains (as necessary)
        viii. RC Pilot Landing
        ix. Group Discussion Observations
3. In Flight Testing With QGroundControl
    a. Communication check (initial)
    b. Control Surface check
    c. OWL_A1 Flight
        i. Zero Sensors
        ii. Set Fail Safe Parameters
        iii. Trim Radio
        iv. Load Waypoints

        v. Launch OWL_A1
        vi. RC Pilot Flight To Elevation
        vii. Engage Autopilot (observe QGroundControl)
            1. Try update of race track in flight
            2. Observe data logging capabilities
        viii. Land OWL_A1
        ix. Group Discussion Observations
    d. OWL_A2 Flight
        i. Zero Sensors
        ii. Set Fail Safe Parameters
        iii. Trim Radio
        iv. Load Waypoints
        v. Launch OWL_A2
        vi. RC Pilot Flight To Elevation
        vii. Engage Autopilot
        viii. Land OWL_A2

4. Multi-Aircraft Simultaneous Flight 1 With QGroundControl
    a. Replace batteries in OWL_A1 & OWL_A2
    b. Zero Sensors in OWL_A1 & OWL_A2
    c. Set Fail Safe Parameters in OWL_A1 & OWL_A2
    d. Load Waypoints for OWL_A1(elevation 350ft) & OWL_A2 (elevation 200ft)
    e. Launch OWL_A1
    f. RC Pilot Flight To Elevation
    g. Engage Autopilot Observe Lap
    h. Launch OWL_A2
    i. RC Pilot Flight To Elevation
    j. Engage Autopilot Observe Lap
    k. Update Waypoints OWL_A1
    l. Update Waypoints OWL_A2
    m. Land OWL_A1
    n. Land OWL_A2
    o. Group Discussion Observations
5. Multi-Aircraft Simultaneous Flight 1 With QGroundControl
    a. Replace batteries in OWL_A1 & Refill Petrol in Sig Rascal_P1
    b. Zero Sensors in OWL_A1 & Sig Rascal_P1
    c. Set Fail Safe Parameters in OWL_A1 & Sig Rascal_P1
    d. Load Waypoints for OWL_A1(elevation 250ft) & Sig Rascal_P1 (elevation 400ft)
    e. Launch Sig Rascal_P1
    f. RC Pilot Flight To Elevation
    g. Engage Autopilot Observe Lap
    h. Launch OWL_A1
    i. RC Pilot Flight To Elevation

j. Engage Autopilot Observe Lap
k. Update Waypoints Sig Rascal_P1
l. Update Waypoints OWL_A1
m. Land OWL_A1
n. Land Sig Rascal_P1
o. Group Discussion Observations

Flight Test #2 (5-7 November 2012)

1. Initial communications check out
    a. Video feed check (5.4 GHz)
        i. Initial Operation
            1. Is Video feed working?
    b. RC Safety Pilot check (2.4 GHz)
        i. Initial Operation
            1. Is RC Communications working?
        ii. Distance check
            1. On the ground place the FrSky transmitter in range check mode and walk the MAV down the flight line until communications are lost. Do conversion for approximated RC range. Record here _________________
    c. Auto Pilot check (914 MHz)
        i. Initial Operation
            1. Is RC Communications working?
        ii. Distance check
            1. Walk the MAV down the flight line until communications are lost. Record distance here _________________
    d. Record and Measure time spent fixing, recovering, launching, turning, flight time, wind speed, battery endurance
2. Verify MAVs are flying properly (In Flight Testing With Mission Planner)
    a. Power on RC controllers for OWL_A1 and OWL_A2
    b. For Each OWL_A1, OWL_A2 and Sig_AP
        i. Open Mission Planner
        ii. **Connec**t to MAV at baud rate of 57600
        iii. On the **Flight Data** tab select the **Actions** tab and click **Set Home Alt**
        iv. Verify that the altitude read out on the right of the flight data screen reads **0**
        v. **Repeat** iii-iv as necessary until successful

        vi. Trim Radio
        vii. Load Waypoints
        viii. Launch MAV
        ix. RC Pilot Flight
            1. Adjust Trim
        x. Engage Autopilot
            1. Adjust Gains (as necessary) **SEE APPENDIX**
        xi. RC Pilot Landing
    c. Group Discussion Observations
    d. Record and Measure time spent fixing, recovering, launching, turning, flight time, wind speed, battery endurance
3. Single MAV flight using QGroundControl (First test OWL_A2 , repeat procedure for Sig_AP)
    a. Power on RC controllers OWL_A2 and Sig_AP
    b. Zero Sensors
        i. Open Mission Planner
        ii. **Connec**t to MAV at baud rate of 57600
        iii. On the **Flight Data** tab select the **Actions** tab and click **Set Home Alt**
        iv. Verify that the altitude read out on the right of the flight data screen reads **0**
        v. **Repeat** as necessary until successful
        vi. **Close Mission Planner but do NOT power off MAV**
    c. Trim Radio
    d. **Open UNMODIFIED qgroundcontrol**
    e. **Connect** to MAV at baud rate of 57600
    f. **Wait for GPS to find location**
    g. **Load Waypoints** using waypoint widget
    h. **Verify Waypoints** by going to the onboard tab of the waypoint widget and clicking refresh
    i. Launch
    j. RC Pilot Flight To Elevation
    k. Engage Autopilot
        i. Try update of race track in flight
        ii. Observe data logging capabilities
    l. Land
    m. Group Discussion Observations
    n. Record and Measure time spent fixing, recovering, launching, turning, flight time, wind speed, battery endurance
4. Single MAV Distance Flight to Loss of Communications
    a. Power on RC controllers for OWL_A2
    b. Zero Sensors
        i. Open Mission Planner
        ii. **Connec**t to OWL_A2 at baud rate of 57600

        iii. On the **Flight Data** tab select the **Actions** tab and click **Set Home Alt**
        iv. Verify that the altitude read out on the right of the flight data screen reads **0**
        v. **Repeat** as necessary until successful
    c. Trim Radio
    d. **Wait for GPS to find location**
    e. **Load Waypoints** using waypoint widget
    f. **Verify Waypoints** by going to the onboard tab of the waypoint widget and clicking refresh
    g. Send Safety pilot and Observers to remote location (Must have range radio)
        i. Observer will have map of flight pattern
    h. **Verify both teams are ready and we are clear for launch**
    i. Launch
    j. RC Pilot Flight To Elevation
    k. RC Pilot flies OWL_A2 toward primary ground station
    l. Ground control operator is continually attempting to connect
    m. Monitor telemetry to observe when 914 MHz communications are established
    n. Ground control operator notes distance on map where communications were established
    o. Observe if after 30 seconds of flight OWL_A2 beings to navigate toward RTL
    p. Operator then notifies RC pilot to land OWL_A2
    q. Record and Measure time spent fixing, recovering, launching, turning, flight time, wind speed, battery endurance

5. Multi-MAV Multi-Ground Station Familiarity Test (Direct LOS) Non-autonomous Relay Navigation
    a. Power on RC controllers for OWL_A1 and OWL_A2
    b. On two separate Laptops connect two Digi modems (one to each laptop)
    c. Open X-CTU and verify that each computer is talking to the attached modem successfully
        i. Select the test/query button. The computer is successfully connected if the type and model information is not garbled text
    d. On laptop one (L1) open Mission Planner
        i. **Power on** OWL_A1 while **holding the MAV level and steady**
        ii. **Connec**t to OWL_A1 at baud rate of 57600
        iii. On the **Flight Data** tab select the **Actions** tab and click **Set Home Alt**
        iv. Verify that the altitude read out on the right of the flight data screen reads **0**
        v. **Repeat** iii-iv as necessary until successful

        - vi. Trim Radio
        - vii. Load Waypoints
    - e. On laptop two (L2) open Mission Planner
        - i. Zero Sensors
            1. **Open Mission Planner**
            2. **Connec**t to OWL_A2 at baud rate of 57600
            3. On the **Flight Data** tab select the **Actions** tab and click **Set Home Alt**
            4. Verify that the altitude read out on the right of the flight data screen reads **0**
            5. **Repeat** as necessary until successful
            6. **Close Mission Planner but do NOT power off MAV**
        - ii. Trim Radio
        - iii. **Open UNMODIFIED qgroundcontrol**
        - iv. **Connect** to MAV at baud rate of 57600
        - v. **Wait for GPS to find location**
        - vi. **Load Waypoints** using waypoint widget
        - vii. **Verify Waypoints** by going to the onboard tab of the waypoint widget and clicking refresh
    - f. Launch OWL_A1
        - i. RC Pilot Flight To Elevation
        - ii. Engage Autopilot
        - iii. Verify Operation Status (if oddities are observed, land and trouble shoot) else
    - g. Launch OWL_A2
        - i. RC Pilot Flight To Elevation
        - ii. Engage Autopilot
        - iii. Verify Operation Status (if oddities are observed, land and trouble shoot) else
    - h. Maximize flight time of OWL_A1 to 15 minutes of flight without exceeding time limit
    - i. Record and Measure time spent fixing, recovering, launching, turning, flight time, wind speed, battery endurance

6. Multi-MAV Multi-Ground Station Familiarity Test (Direct LOS) Autonomous Relay Navigation
    - a. Power on RC controllers for OWL_A1 and OWL_A2
    - b. On two separate Laptops connect two Digi modems (one to each laptop)
    - c. Open X-CTU and verify that the computer is talking to the modem successfully
        - i. Select the test/query button. The computer is successfully connected if the type and model information is not garbled text
    - d. On laptop one (L1) open Mission Planner

        i. **Power on** OWL_A1 while **holding the MAV level and steady**
        ii. **Connec**t to OWL_A1 at baud rate of 57600
        iii. On the **Flight Data** tab select the **Actions** tab and click **Set Home Alt**
        iv. Verify that the altitude read out on the right of the flight data screen reads **0**
        v. **Repeat** iii-iv as necessary until successful
        vi. Trim Radio
        vii. Load Waypoints at altitude of 550 ft
    e. On laptop two (L2) open Mission Planner
        i. Zero Sensors
            1. **Open Mission Planner**
            2. **Connec**t to OWL_A2 at baud rate of 57600
            3. On the **Flight Data** tab select the **Actions** tab and click **Set Home Alt**
            4. Verify that the altitude read out on the right of the flight data screen reads **0**
            5. **Repeat** as necessary until successful
            6. **Close Mission Planner but do NOT power off OWL_A2**
        ii. Trim Radio
        iii. **Open MODIFIED qgroundcontrol**
        iv. **Connect** to both MAVs at baud rate of 57600 (do not enable multiplexing)
        v. **Wait for GPS to find location**
        vi. **Click** on map as close as possible to the location of the ground station as possible
    f. Launch OWL_A1
        i. RC Pilot Flight To Elevation
        ii. Engage Autopilot
        iii. Verify Operation Status (if oddities are observed, land and trouble shoot) else
    g. Launch OWL_A2
        i. RC Pilot Flight To Elevation
        ii. Engage Autopilot
        iii. Every 5 seconds click anywhere on the map
        iv. Verify Operation Status (if oddities are observed, land and trouble shoot) else
    h. Maximize flight time of first MAV to 15 minutes of flight without exceeding time limit
        i. Take manual control of MAV OWL_A2 and land it
        ii. Take manual control of MAV OWL_A1 and land it

i. Record and Measure time spent fixing, recovering, launching, turning, flight time, wind speed, battery endurance

7. Multi-MAV Multi-Ground Station Familiarity Test (Direct LOS) Autonomous Relay Navigation **with SIG_AP in place of OWL_A2**
    a. Power on RC controllers for OWL_A1 and OWL_A2
    b. Switch Sig_AP Aircraft ON (leave Autopilot switch OFF)
    c. **Power on** OWL_A1 while **holding the MAV level and steady**
    d. On laptop one (L1) open Mission Planner
        i. Plug in Ch1-Relay modem to laptop L1
        ii. **Connec**t to OWL_A1 at baud rate of 57600
        iii. On the **Flight Data** tab select the **Actions** tab and click **Set Home Alt**
        iv. Verify that the altitude read out on the right of the flight data screen reads **0**
        v. **Repeat** iii-iv as necessary until successful
        vi. Trim Radio
        vii. Load Waypoints
    e. Switch Sig_AP Autopilot ON
    f. On laptop two (L2) open Mission Planner
        i. Plug in Ch1-Sig modem to laptop L2
        ii. Zero Sensors
            1. **Open Mission Planner**
            2. **Connec**t to Sig_AP at baud rate of 57600
            3. On the **Flight Data** tab select the **Actions** tab and click **Set Home Alt**
            4. Verify that the altitude read out on the right of the flight data screen reads **0**
            5. **Repeat** as necessary until successful
            6. Hold Sig_AP level
            7. Under the configuration tab click on the calibrate level
            8. Verify on the flight data tab that the HUD is showing level flight
            9. **Close Mission Planner but do NOT power off MAV**
        iii. Trim Radio
        iv. **Open MODIFIED qgroundcontrol**
        v. **Connect** to Sig_AP at baud rate of 57600
        vi. **Wait for GPS to find location**
        vii. **Select** MAV001 (Sig) for control
        viii. **Load Waypoints** using waypoint widget
        ix. **Verify Waypoints** by going to the onboard tab of the waypoint widget and clicking refresh

    g. Launch OWL_A1
        i. RC Pilot Flight To Elevation
        ii. Engage Autopilot
        iii. Verify Operation Status (if oddities are observed, land and trouble shoot) else
    h. Launch Sig_AP
        i. RC Pilot Flight To Elevation
        ii. Engage Autopilot
        iii. Verify Operation Status (if oddities are observed, land and trouble shoot) else
    i. Maximize flight time of OWL_A1 to 15 minutes of flight without exceeding time limit
        i. Take manual control of MAV Sig_AP and land it
        ii. Take manual control of MAV OWL_A1 and land it
    j. Record and Measure time spent fixing, recovering, launching, turning, flight time, wind speed, battery endurance
8. Beyond Communications Line Of Sight (BCLOS) Flight Test
    a. Power on RC controllers for OWL_A1 and OWL_A2
    b. Switch Sig_AP Aircraft ON (leave Autopilot switch OFF)
    c. **Power on** OWL_A1 while **holding the MAV level and steady**
    d. On laptop one (L1) open Mission Planner
        i. Plug in Ch1-Relay modem to laptop L1
        ii. **Connec**t to OWL_A1 at baud rate of 57600
        iii. On the **Flight Data** tab select the **Actions** tab and click **Set Home Alt**
        iv. Verify that the altitude read out on the right of the flight data screen reads **0**
        v. **Repeat** iii-iv as necessary until successful
        vi. Trim Radio
        vii. Load Waypoints
    e. Switch Sig_AP Autopilot ON
    f. On laptop two (L2) open Mission Planner
        i. Plug in Ch1-Sig modem to laptop L2
        ii. Zero Sensors
            1. **Open Mission Planner**
            2. **Connec**t to Sig_AP at baud rate of 57600
            3. On the **Flight Data** tab select the **Actions** tab and click **Set Home Alt**
            4. Verify that the altitude read out on the right of the flight data screen reads **0**
            5. **Repeat** as necessary until successful
            6. Hold Sig_AP level

7. Under the configuration tab click on the calibrate level
8. Verify on the flight data tab that the HUD is showing level flight
9. **Close Mission Planner but do NOT power off MAV**

iii. Trim Radio
iv. **Open MODIFIED qgroundcontrol**
v. **Connect** to Sig_AP at baud rate of 57600
vi. **Wait for GPS to find location**
vii. **Select** MAV001 (Sig) for control
viii. **Load Waypoints** using waypoint widget
ix. **Verify Waypoints** by going to the onboard tab of the waypoint widget and clicking refresh

g. Send out RC pilot and distant area observer with map of flight path, cell phone and range radio
h. Launch SIG_AP
    i. RC Pilot Flight To Elevation and approximate relay position
i. Launch OWL_A1
    i. RC Pilot Flight To Elevation
    ii. Engage Autopilot
    iii. Verify Operation Status (if oddities are observed, land and trouble shoot) else
j. Ground Control Operator verifies that relay of communications is operational
    i. Is telemetry data displaying in the ground control software?
    ii. Can information be written to the rover MAV?
    iii. If yes proceed. If no fly OWL_A1 closer to Sig_AP.
k. **On Sig_AP**
    i. Engage Autopilot
    ii. Every 5 seconds click anywhere on the map
    iii. Verify Operation Status (if oddities are observed, land and trouble shoot)
l. Maximize flight time of OWL_A1 to 15 minutes of flight without exceeding time limit
m. On ground control operator's queue both RC pilots take control of their respective MAVs and land the MAVs
n. Record and Measure time spent fixing, recovering, launching, turning, flight time, wind speed, battery endurance

9. Stationary Target Flight Test
    a. Emplace stationary target
    b. Set waypoint pattern to loiter over target
    c. Launch OWL and monitor to ensure proper flight path

    d. Record and Measure loiter time and target observed time

10. Road Surveillance Flight Test
    a. Designate linear zone of observation
    b. Set waypoint pattern to observe linear zone of observation
    c. Launch OWL and monitor to ensure proper flight path
    d. Record and Measure loiter time and target observed time

Bibliography

[1] M. T. Seibert, A. J. Stryker, J. R. Ward and C. T. Wellbaum, System Analysis and Prototyping for Single Operator Management of Multiple Unmanned Aerial Vehicles Operating Beyond Line of Sight, Wright-Patterson AFB: Air Force Institute of Technology, 2010.

[2] United States Air Force, *Air Force Doctrine Document 1,* Washington D.C.: USAF Chief of Staff, 2011.

[3] United States Air Force, "United States Air Force Unmanned Aircraft Systems Flight Plan 2009-2047," Headquarters, United States Air Force, Washington D.C., 2009.

[4] United States Department of Defense, "FY 2009-2034 Unmanned Systems Integrated Roadmap," Department of Defense, United States of America, Washington, D.C., 2009.

[5] A. Ryan, M. Zennaro, A. Howell, R. Sengupta and J. K. Hedrick, "An Overview of Emerging Results in Cooperative UAV Control," in *43rd IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas, 2004.

[6] J. P. Boire, Autonomous Routing of Unmanned Aerial Vehicle (UAV) Relays to Mimic Optimal Trajectories in Real Time, Wright-Patterson AFB: Air Force Institute of Technology, 2011.

[7] Procerus Technologies, *Installation and Configuration Guide,* Vineyard: Procerus Technologies, Inc., 2008.

[8] Digi International Inc., *9XTend™ OEM RF Module,* Minnetonka, MN: Digi International Inc., 2008.

[9] Sig Mfg. Co., "Sig Rascal 110 - Almost Ready to Fly," Sig Mfg. Co., [Online]. Available: http://www.sigmfg.com/IndexText/SIGRC84ARFB.html. [Accessed January 2013].

[10] L. Meier, J. Camacho, B. Godbolt, J. Goppert, L. Heng, M. Lizarraga, L. Meier, K. Rudin, P. Tanskanen and A. Tridgell, "QGroundControl: Ground Control Station for Small Air-Land-Water Autonomous Unmanned Systems," 2010. [Online]. Available: http://qgroundcontrol.org/start. [Accessed 12 August 2012].

[11] K. Daniel, S. Rohde, N. Goddemeier and C. Wietfeld, "Cognitive Agent Mobility for Aerial Sensor Networks," *IEEE Sensors Journal,* vol. 11, no. 11, pp. 2671-2682, 2011.

[12] J. Li, C. Blake, D. S. J. De Couto, H. I. Lee and R. Morris, "Capacity of Ad Hoc Wireless Networks," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking,* New York, 2001.

[13] P. Gupta and P. R. Kumar, "The Capacity of Wireless Networks," *IEEE Transactions on Information Theory,* vol. 46, no. 2, pp. 388-404, 2000.

[14] K. Jain, J. Padhye and V. N. Padmanabhan, "Impact of Interference on Multi-Hop Wireless Network Perfomance," *Wireless Networks,* vol. 11, pp. 471-487, 2005.

[15] Digi International Inc., *XBee-PRO® 900 RF Modules,* Minnetonka, MN: Digi International Inc., 2011.

[16] T. J. Shuck, *Development of Autonomous Optimal Cooperative Control in Relay Rover Configured Small Unmanned Aerial Systems,* Wright-Patterson AFB: Air Force Institute for Technology, 2012.

[17] FrSky, "Products Show: Sensor Hub and Sensors," FrSky, 2010. [Online]. Available: http://www.frsky-rc.com/ShowProducts.asp?id=92. [Accessed 9 January 2013].

[18] Digi International Inc., *XBee-PRO® 900/DigiMesh™ 900 OEM RF Modules,* Minnetonka, MN: Digi International Inc., 2008.

[19] Persistent Systems, LLC, *Sector Antenna Array Router,* New York, NY: Persistent Systems, LLC, 2011.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 074-0188*

1. REPORT DATE *(DD-MM-YYYY)* 21-03-2013	**2. REPORT TYPE** Master's Thesis	**3. DATES COVERED** *(From – To)* March 2012 – March 2013

TITLE AND SUBTITLE Aerial Networking for the Implementation of Cooperative Control on Small Unmanned Aerial Systems	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Songer, Scott A., Captain, USAF	**5d. PROJECT NUMBER**
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENV-13-M-29

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL/RQQA
Mr. Derek Kingston email: derek.kingston@wpafb.af.mil
2210 8th St, Wright-Patterson AFB, OH phone: (937)255-6301

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL/RQQA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A: APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

13. SUPPLEMENTARY NOTES

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT

The employment of Small Unmanned Aerial Systems (SUAS) for reconnaissance and surveillance missions is a vital capability of the United States military. Cooperative control algorithms for SUAS can enable tactical multi-vehicle configurations for communications extension, intelligent navigation, and a multitude of other applications. Past research at AFIT has designed and simulated a cooperative rover-relay algorithm for extended communications and has investigated its implementation through various modem configurations. This research explores aerial networking options for implementing cooperative control and applies them to an actual SUAS. Using Commercial Off-The-Shelf (COTS) hardware, a system was designed and flight tested to implement the rover-relay algorithm and provide a testbed system for future research in cooperative control.

Two different modem configurations were designed and tested. The first modem configuration was demonstrated through a series of ground and flight tests to successfully relay autopilot commands and telemetry between a ground station and a rover aircraft through a relay aircraft. This configuration effectively doubles the effective range of the rover system to 1.2 miles, together with an algorithm that autonomously navigates the relay aircraft to an optimal location. Secondly, a mesh network was configured and tested. This configuration successfully relayed aircraft telemetry to the ground station from each vehicle in the network. However, the network suffered from low throughput, which limited autopilot functionality, such as updating navigation waypoints to each aircraft. The results suggest the system be updated with more capable modems in a mesh configuration to broaden the possibilities for future research in cooperative applications.

15. SUBJECT TERMS

SUAS, Cooperative Control, Aerial Networking, UAV

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			John Colombi, AFIT/ENV, ADVISOR
U	U	U	UU	86	**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-6565, ext 3347 (john.colombi@afit.edu)

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18