A SYSTEM ARCHITECTURE FOR PHASED DEVELOPMENT OF REMOTE SUAS OPERATION

A Thesis
presented to
the Faculty of California Polytechnic State University,
San Luis Obispo

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Aerospace Engineering

by
Eric Ashley
March 2020

COMMITTEE MEMBERSHIP

TITLE:	A System Architecture for Phased Development of Remote sUAS Operation
AUTHOR:	Eric Ashley
DATE SUBMITTED:	March 2020
COMMITTEE CHAIR:	Aaron Drake, Ph.D. Professor of Aerospace Engineering
COMMITTEE MEMBER:	Kira Abercromby, Ph.D. Professor of Aerospace Engineering
COMMITTEE MEMBER:	Arnold Deffo, Ph.D. Assistant Professor of Aerospace Engineering
COMMITTEE MEMBER:	Paulo Iscold, Ph.D. Associate Professor of Aerospace Engineering

ABSTRACT

A System Architecture for Phased Development of Remote sUAS Operation

Eric Ashley

Current airspace regulations require the remote pilot-in-command of an unmanned aircraft systems (UAS) to maintain visual line of sight with the vehicle for situational awareness. The future of UAS will not have these constraints as technology improves and regulations are changed. An operational model for the future of UAS is proposed where a remote operator will monitor remote vehicles with the capability to intervene if needed. One challenge facing this future operational concept is the ability for a flight data system to effectively communicate flight status to the remote operator. A system architecture has been developed to facilitate the implementation of such a flight data system. Utilizing the system architecture framework, a Phase I prototype was designed and built for two vehicles in the Autonomous Flight Laboratory (AFL) at Cal Poly. The project will continue to build on the success of Phase I, culminating in a fully functional command and control system for remote UAS operational testing.

ACKNOWLEDGMENTS

Although it required me to suffer through a couple more years of classes and write this thing, I would like to thank Cami for saying, "Let's go to graduate school" that one time. Without you, there would have been no master's degree for me, and I do not think I would have pursued the career I actually wanted. Also, thanks for doing this with me, because it would have been much more difficult on my own. I would like to thank my parents for always being supportive, for being great people, and for the genes that make me reasonably intelligent. Finally, I like to extend my gratitude to Dr. Drake for always telling me "if I just did it right, it would be correct." You always said the right words at the right times.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

NOMENCLATURE

AFL	Autonomous Flight Laboratory
AGL	Above ground level
C2 System	Command and control system
Cal Poly	California Polytechnic State University, San Luis Obispo
CG	Center of gravity
CPU	Central processing unit
CSV	Comma separated values
DAQ	Data acquisition unit
DOF	Degree of freedom
EFR	Educational Flight Range
EMI	Electromagnetic interference
FAA	Federal Aviation Administration
GCS	Ground control station
GNSS	Global Navigation Satellite System
GPIO	General purpose input/output
GPS	Global Positioning System
HAT	Hardware attached on top
I^2C	Inter-Integrated Circuit
IEEE	Institute of Electrical and Electronics Engineers
IMU	Inertial measurement unit
MCU	Microcontroller
MTOW	Maximum takeoff weight
NAS	National airspace system
SBC	Single board computer
sUAS	Small unmanned aircraft systems
UAS	Unmanned aircraft systems
USB	Universal serial bus

1. INTRODUCTION

1.1 Problem Statement

Many researchers utilize unmanned aircraft systems (UAS) as tools for collecting data and performing tasks. Specifically, small unmanned aircraft systems (sUAS) have become a significant contributor for wildlife monitoring, agriculture, and healthcare delivery. However, operational limitations imposed by the Federal Aviation Administration (FAA) inhibit these end users from harnessing the full potential of sUAS. In some cases, sUAS are not as practical or cost effective as a traditional aircraft to the end user (Christie, 241).

Researchers continue to utilize sUAS but cite three fundamental impediments to improved effectiveness. The FAA regulations do not allow for significant operational variation because of the restrictions requiring the vehicle to remain within line of sight of the operator and under 400 feet above the ground. Furthermore, these restrictions do not encourage the production of vehicles with improved capabilities because their utilization is not allowed in the current national airspace system (NAS). The regulations also impede the advancement of alternative mission types which may increase operational efficiency.

It is prudent to anticipate the future of sUAS in the NAS and address the needs of end users by assuming sUAS will perform beyond the limitations imposed by the current regulations. There is research which already discusses some of the aspects of a future sUAS operational concept, but no work has been found regarding the technical requirements needed for a future sUAS operational architecture. This thesis will provide a systems engineering approach to developing the capability which will allow for increased sUAS operational efficiency.

1.2 Background

The FAA defines sUAS as vehicles with a maximum takeoff weight (MTOW) under 55 pounds. The FAA's Part 107 Rules allow the operation of sUAS in the NAS, but restrict the operation to remain within line of sight of the operator, under 400 feet above ground level, and in designated airspace zones (Federal Aviation Administration, 1). Despite these rules, sUAS still find utility in certain applications as end users seek to find more effective methods for data collection and package delivery.

Increased effectiveness of sUAS across a range of wildlife and ecology monitoring tasks for sea lions, killer whales, and sea otters could change the way scientists collect ecological data. Data has been collected which directly compared the capability of these sUAS monitoring tasks with traditional methods. The conclusion of this research was that sUAS provide better image resolution and did not disturb the target animals as much as a traditional aircraft (Christie, 251). Furthermore, the ability of sUAS to successfully capture wildlife populations and count individuals rivals the capability of traditional methods (Linchant, 247).

Agricultural surveying is another space where research has found that sUAS could provide farmers with a faster solution to weed management strategies in comparison with traditional ground-based monitoring operations. Current ground-based system use real-time techniques to spray weeds as they are detected. This technique is improved by using sUAS to collect data quickly before the ground-based systems arrive allowing for a priori planning based on the sUAS data (Rasmussen, 243). A similar concept described the significant improvement in cost and weed mitigation over traditional "blanket" spraying techniques by using a networked multi-sUAS system to apply pesticides in a precision agriculture simulation (Stark, 301). The high-resolution imagery from sUAS flying at low altitudes could also improve the understanding agriculturalists have of crops needs and reactions to specific management techniques (Hunt, 2).

sUAS have shown utility delivering healthcare products in areas where rapid transport of critical medical supplies can be challenging as well. Zipline, a company headquartered in San Francisco, partnered with the country of Rwanda to address the problem of supplying hospitals with the blood when traditional infrastructure is impassable. Zipline has been successfully delivering critical healthcare products using their sUAS network since October 2018. The Zipline distribution center can deliver life-saving blood to one of 21 hospitals in a 75-kilometer range using their sUAS platform. An operator at the distribution hub monitors all missions and can send commands to the vehicle if needed (Ackerman, 34). Other studies have focused on a similar concept by comparing two models for sUAS delivery systems against traditional methods. The models concluded that under some circumstances it was advantageous to send critical medical supplies by sUAS than traditional methods. For example, the delivery company DHL's Parcel sUAS transferred medicine between Bavarian Alpine villages in 8 minutes when a DHL Parcel van would have taken 30 minutes. The models support sUAS capability for a more timely, efficient, and economical healthcare delivery system (Scott, 3297).

All of the examples cited exhibit the potential of sUAS, if not the outright capability. However, there are three impediments which these studies have cited are hindering progress of more extensive and effective sUAS utilization. These three impediments are the regulations on the operation of sUAS, the capability of current sUAS, and the operational methods for mission execution. The FAA regulations pose the greatest hurdle because with those regulations in place, the other two impediments will not be addressed by companies because those vehicles would have increased operational capability in the current NAS. However, there is widespread belief that the regulations will eventually change to allow for more capable vehicles employed with more sophisticated operational concepts such as a single operator in control of multiple vehicles and beyond line of sight control (Atkins, 11), (Trujillo, 936).

The future operational concept envisioned with the expansion of sUAS regulations in the NAS would allow for sUAS to fly autonomously beyond line of sight. This capability would significantly increase the utility of vehicles for some applications. As such, there has been considerable research focused on a future NAS where sUAS fly beyond line of sight of their operator. Safety measures are beginning to be analyzed at a modeling level which indicate there are effective collision avoidance strategies for sUAS flying in a congested NAS (Luxhøj, 933). Consideration has also been given to the implementation of an unmanned aircraft management system which would require sUAS to register flights to avoid airspace conflicts with other aircraft. Furthermore, the system could act as a form of air traffic monitoring resource for the remote operator and vehicle (Jiang, 124).

Investigations have also been conducted on the system and operator requirements associated with controlling multiple vehicles beyond line of sight. Data has been accumulated regarding the information an operator will require while controlling a vehicle, or a fleet of vehicles (Trujillo, 942), (Vincenzi, 925). Alternative analysis of a single operator controlling multiple vehicles have delved into the automation required for such as system (Cummings, 2). All the research concerning the human aspect of the operational concept conclude that more automation is required to reduce potential negative human impact on mission success. However, it is imperative that a human be ready to intervene during flight-critical decisions even if the complexity of that interaction is still unknown (Ruff, 6).

1.3 Project Definition

A plethora of research has been completed regarding the airspace systems and human control aspect of a sUAS, or multiple sUAS, flying beyond line of sight of an operator, but no literature was found for a system to test the operational concept. Due to this lack of research and need to understand the operational requirements in anticipation of sUAS integration with the NAS, the Autonomous Flight Laboratory (AFL) at California Polytechnic State University, San Luis Obispo (Cal Poly) has initiated a project for initial testing of such as system. The AFL is interested in validating the operational concept and identifying applications for which this operational concept is well-suited. The project for developing an initial system will be divided into three phases:

> **Phase I**: Develop a prototype system for collecting flight data from a sUAS and sending it to a remote operator at a ground station.
>
> **Phase II**: Complete Phase I and provide the remote operator with control of a payload onboard the sUAS.
>
> **Phase III**: Complete Phase I, Phase II, and provide the remote operator with control of the sUAS.

At the culmination of Phase III, the AFL will have the capability to test the operational efficiency of flying a single, or multiple, sUAS beyond line of sight. Phase III is dependent on the modification of the FAA's Part 107 Rules or authorization for beyond line of sight testing via a certificate of authorization. A flow chart for the three-phase development of the project is shown in **Figure 1**. It illustrates the major steps for development of each phase. Phase I requires background research and development of a system architecture for the full system followed by a prototype build. Phase II builds on the research and system architecture developed in Phase I to enhance the prototype. Phase III builds on the Phase II work and, with FAA approval, develops a beyond line of sight sUAS control system.

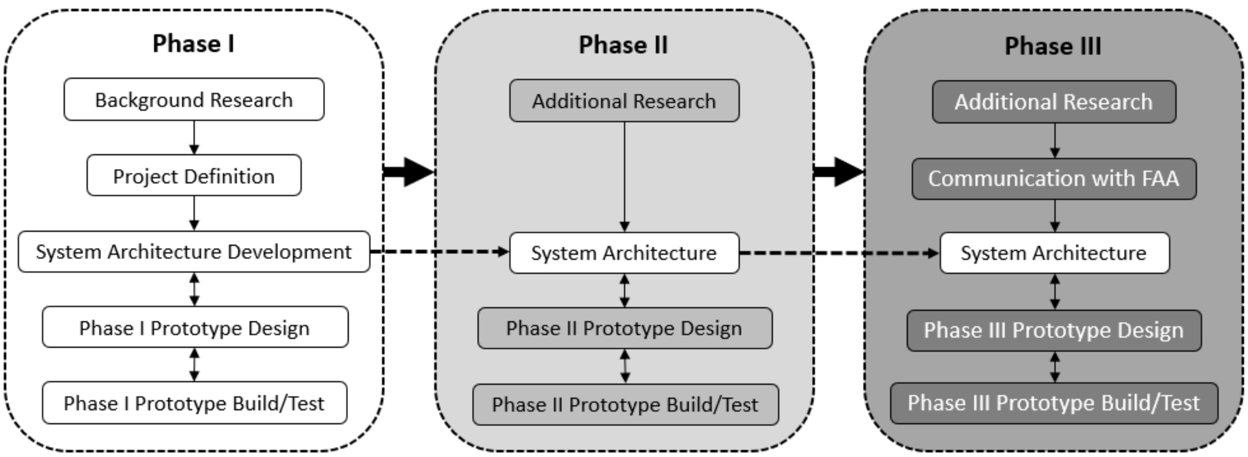

Figure 1: Full project development process.

This project for the AFL is meant as a first research test article for beyond line of sight operational functionality. All the development is intended to function with vehicles owned by the laboratory, and not as a template for building a mission-specific sUAS control system.

This thesis will complete Phase I of the project with these specific objectives:

1. Develop a complete system architecture for beyond line of sight command and control of a sUAS.

2. Design and build a prototype flight data collection system to work with the AFL's Vapor and Nova sUAS.

3. Test the prototype to validate the system architecture design and prototype development.

The deliverable for the thesis will be a validated system architecture. Validation for the system architecture will be provided by an initial Phase I prototype system which can collect and transmit flight data from a sUAS to a remote ground station but does not provide any control of the vehicle. The purpose of the initial Phase I prototype is to demonstrate the concept is sound rather than validate the prototype functionality. Once the system architecture is validated by the Phase I prototype, the project will be ready to continue on to Phases II and III.

2. METHODOLOGY

2.1 Introduction to System Architecture Development

The development of the system architecture was the most significant portion of the project because its scope included all three phases of the project. A successful system architecture should provide a framework which can be filled in using the specific requirements for a design. It should be able to accept design versions and iterations while ensuring the stakeholder needs.

To this end, the development of a system architecture can be approached many ways. There is no universal method for system architecture development. The result of the system architecture design is assessed by the success of the specific design built using the system architecture framework. Validating the development at the end inherently makes system architecture development an iterative process. As issues with the architecture are found during a specific design iteration, the architecture should be modified to accommodate the needs of the design and then reassessed. The notion which remained constant through the development of the system architecture was to focus on the elements fundamental to the system performing as desired by the system stakeholders.

Dennis Buede's *The Engineering Design of Systems* was used as a guide for outlining the general form of system architecture development (Buede, 51). Some of the specific development concepts were based on the needs of the stakeholder rather than blindly following Buede's recommendation. Moreover, a technical report from Shaun Hayes from the Naval Post-Graduate School in Monterrey was used as a reference (Hayes, 17) because Hayes successfully applied Buede's system.

Further guidance for development of the system requirements were provided by Lockheed Martin's report for an intelligent transportation system and IEEE's guide for developing system requirements (Martin, 12), (IEEE Computer Society, 11). These sources provided procedural, but no technical, insight for developing system requirements.

The system architecture development began by defining the operational concept of the system, using that operational concept to inform the design of the physical and functional architectures, and finally combining all the elements into a cohesive operational architecture. This general form was followed by Hayes and in this thesis, but the detailed processes for filling out the system framework were based on what fundamentally agreed with the development of this system architecture.

2.2 Operational Concept

The development of the operational concept is essentially envisioning the system in use for all scenarios. The scenarios are representative of the employment needs specified by the stakeholders. The scenarios should be exhaustive, including any foreseeable use case of the device and the non-operational, life scenarios where the system is utilized but not under normal use conditions. The rigorous characterization of scenarios should begin with the most simplistic scenarios, slowly becoming more complicated. As the system is imagined in increasingly more complicated scenarios, the needs of the stakeholders and systems become abundantly clear.

After the scenarios for the system are described, those scenarios are used to ensure all the necessary interactions of the system are identified. To illustrate these interactions visualizations showing the relationship between the inputs and outputs of the system, and its subsystems, are drawn out. The system will have some known inputs from the external context entities, perform some function using that information, and perform some output actions in response to the inputs. The precise function and method for perform the tasks are not described, but the connection between elements in the system with external elements provide insight into the interaction of the system with the world. These diagrams are called external system diagrams because they include the key interactions of the system with external systems and context entities.

The final consideration of the operational concept is to organize the objectives of the subsystems into a hierarchy. From the external system diagrams, the interactions between the system, subsystem, external systems, and context entities provide the basis for organizing the interactions. The order of the objective hierarchy will assist with specific design decisions because they list the objectives from the most to the least important. These objectives are not described fully, but rather a representation of the interactions and how they should be prioritized. Using the scenarios, external system diagrams, and objective hierarchy, the operational concept is a fully defined framework which will be a reference for defining elements of the functional and physical architectures.

2.3 Functional Architecture

The functional architecture is the objectives and subsequent functions required to satisfy those objectives. The functional architecture serves as a decomposition of the system's top-level functions. This decomposition contains the functions which the system is required to perform to so that the system will operate as desired.

The first step of the functional architecture is the organization of the system functions into a hierarchy. Through composition and decomposition methods, the system functions and sub-functions should be defined. The functions can be identified from scenarios defined the operational concept, external system diagrams, or needs from the physical architecture under concurrent development.

Using the definitions of the functions, the direct relationships between the inputs and outputs of the systems, subsystems, and context entities are highlighted. Essentially, which interactions require which functions. Furthermore, the added detail will allow for sequencing the functions for these interactions. The methods for modeling these relationships can take the form of flow block diagrams or data flow diagrams.

During these processes, decisions about the system will need to be made which will require input from the stakeholders. The key takeaway from the stakeholder at this point should be for any glaring absence of necessary features. As with all the processes in the system architecture development the completion of the functional architecture may require multiple iterations.

A set of specifications describing the each of the system elements should be traced from the stakeholder needs drafted in the operational concept section. This process should verify that all the stakeholder needs are satisfied by an appropriate function. This check will also correlate with resources defined in the physical architecture during the development of the operational architecture.

Finally, it is crucial to think about the potential failures and fault tolerances in the system. By adding in functions which can detect failures, the system can become more robust. This stage of the functional architecture is last because it can be difficult to consider failures so early on in development. Thus, it is often filled in more completely during the operational architecture phase. The consideration of potential failures and fault tolerances, additional input and output needs may be identified.

2.4 Physical Architecture

The physical architecture is developed concurrently with the functional architecture. It represents the resources which comprise the system and correspond to each of the functions defined in the functional architecture. The first step is to create a generic physical architecture based on the functions from the functional architecture. From there, more detail is added until the physical architecture fully compliments the operational architecture during iterative development. Multiple versions of the physical architecture should be completed in concurrence with the functional architecture to act as options during the development of the operational architecture.

In addition to describing the physical resources needed, the physical architecture indirectly defines the procedures needed by the system. The physical elements in the architecture act in specific ways which inform the procedures and controls of the system. These ideas can help inform changes or additions to the functional architecture as well as provide information for the operational architecture to be developed next.

2.5 Operational Architecture

With the definition of the system in terms of the functions it will need to perform and the resources which will be performing the actions, the next step is to combine these parts to form the operational architecture. The major step required for defining the operational architecture in this thesis were to apply functions to their specific physical resources and define the inputs and controls required for a specific functional output.

The first step for developing the operational architecture is to apply functions from the functional architecture to resources in the physical architecture. Thus, diagrams of the relationship between the physical and functional architectures are defined as a framework which can be filled in during the specific design.

The second step is to understand the inputs, controls, and outputs for the various functions defined in the functional architecture. This occurs in a functional activation and control table such as the example in **Table 1.**

Table 1: An example of a functional activation and control table.

Function	Output	Required Input	Required Control
F.1 Function 1	**O.1** Output 1	**O.3** Output 3	**C.1** Control 1
F.2 Function 2	**O.2** Output 2	**I.1** Input 1	**F.1** Function 1 **C.2** Control 2
F.3 Function 3	**O.3** Output 3	**O.2** Output 2 **I.2** Input 2	**C.3** Control 3

A key part of the development of the operational architecture is the need to refine all parts of the architectural development simultaneously in order to reconcile any design issues. The operational architecture is required to be refined enough to provide a framework which is not overly broad so that it can be filled in by specific design requirements when employed. The performance and risk analysis of the system is also considered at the point of the development.

Once the analysis is complete the architecture is ready for use as a framework for specific design requirements.

2.6 Specific Design Methods

There are numerous techniques for systematically addressing preliminary and critical design. They use requirements as an input and apply creative solutions to reach a final output. However, for each design, a new design process must be started. To avoid the need to continuously redesign using the same requirements, a different approach was employed. The system engineering task was to develop the system architecture, then fill in the framework with design specifics. By utilizing this framework, much of the work for redesigning can be bypassed.

References for specific designs were scant, but a two were referenced for guiding the design decisions. One system utilized a microcontroller (MCU) (Brusov, 133) while the other was built around a single board computer (SBC) (Taha, 132). Both systems proved capable of collecting flight data on a sUAS.

2.7 Summary of System Architecture Development

This chapter described the methodology utilized to develop the system architecture in the following chapter. For developing a system architecture, an operational concept is developed, followed by the co-development of the physical and functional architectures, and culminating in the completion of the operational architecture. Upon the completion of the operational architecture, the system has been defined as a framework ready to be filled in with a stakeholder's specific requirements in place of a more traditional design procedure. By filling in the system framework with specific elements, many design options can be defined. Then, a selected design can move forward into an implementation phase.

3. SYSTEM ARCHITECTURE

3.1 System Architecture Design

The specific system architecture development for this thesis is described in the following section. The operational concept for the system was developed, then the functional and physical architectures were identified and refined, and finally the parts were combined into a cohesive framework called the operational architecture.

The operational architecture acts as the framework to be filled in during the prototype design phase. As design modifications are needed or requirements change, the architecture shall continue to serve as the framework which can support any updated designs. Thus, the process of defining the system architecture in the following section only needs to be completed once while allowing flexibility for the specific design and subsequent variants.

3.2 Operational Concept

During the development of the operational concept, the system was modeled as a black box. This meant that the functionality of the system was accounted for as it is intended to perform, but the mechanism controlling the functions remained unknown. This allowed for the needs of the stakeholders to be addressed as known inputs with desired outputs from the system while ignoring how inputs would become outputs. The stakeholders for this development are the members of the AFL. The details regarding the specific means with which the system converts those inputs into outputs will be addressed by the functional and physical architecture development.

3.2.1 Scenarios/System Use

The operational and life scenarios of the system were determined to support the description of the operational concept. In this case, the system is a sub-system for a larger sUAS mission, so its functions do not change much between mission types. The system will be used for employment, life, and validation tasks as shown in **Table 2.** The scenarios listed under the categories are a comprehensive list of envisioned scenarios but is not exhaustive. Employment scenarios include the missions sUAS vehicles fitted with the system are anticipated to fly. Life scenarios include use of the system for anything which is not flight data collection. Finally, validation and testing scenarios include the use of the system to validate the design satisfies the requirements.

Considering the various scenarios, a more detailed outline of the operational concept for the system was established. This detailed diagram of the operational concept the diagram highlighted how the system integrates with the normal mission tasks providing insight into the functionality of the system. The detailed version of the operational concept with the system-specific sections highlighted can be found in **Table 3.** The table flows from top to bottom from mission planning through post-flight tasks. Within each section, tasks flow from left to right. Underneath each header is a subset of tasks performed to complete the head task. The overall flow of the mission is consistent with the AFL's procedures. Within the AFL procedures, tasks which are specific to the remote data collection system are labeled with the development phase where the task will be required. Tasks which will be transferred to the remote data collection system from the standard control system have also been labeled with the corresponding development phase.

Table 2: Categories for employment, life, and validation and testing scenarios.

Employment Scenarios	Life Scenarios	Validation and Testing Scenarios
Proficiency • Pilot proficiency • Crew proficiency • Pilot training • Crew training *Testing* • Endurance testing • Flight control testing • Waypoint logic testing • Control logic testing • Payload testing • Modification testing *Wildlife Monitoring* • Visual tracking • Infrared tracking • Transceiver tracking *Rangeland Surveying* • Visual inspection • Multispectral inspection *Agricultural Data Collection* • Visual inspection • Multispectral inspection *Search and Rescue* • Visual searching • Infrared searching *Long-Range Delivery* • Single-package delivery • Multi-package delivery *Precision Data Collection* • Sample collection • Detailed visual • Detailed sensing	*Install/Uninstall* *Sensor Testing* *Data Offload* *Programming*	*Fit Checks* *Static Power-On* *Static Data Collection* *Static Communications Checks*

The key capabilities of the system were clear in the detailed operational concept because it was obvious when the system would need to be active and what functions it would need to perform. For each phase, the key capabilities of the system change the operational concept of the mission significantly. For Phase I the system only needed to collect and transmit flight data. Phase II required that the system collect and transmit data while adding the additional capability to control the payload. Phase III required that the system collect and transmit flight data, control the payload, and provide control of the vehicle. With each phase the capability of the system became more advanced, and mission success became more reliant on the system. A simplified version of the mission concept highlighting the different phases can be found in **Table 4.**

Table 3: Operational concept for the system.

1. Mission Planning			
1.1. Build mission plan →	**1.2. Prepare aircraft for mission** →	**1.3. Prepare ground support system** →	**1.4. Brief crew**
Consult with customer Survey flight area Verify safety requirements are met Manage all required approvals Interface with outside parties	Charge batteries / Prepare fuel Install payload Install remote data system [I / II / III] Perform pre-flight maintenance Verify mission plan Verify communication [I / II / III] Verify data collection [I / II / III]	Charge batteries Verify mission plan Verify communication [I / II / III] Verify data collection [I / II / III]	Cover mission plan Assign roles Cover crew resource management actions

↓

2. Pre-Flight		
2.1. Transport to mission location →	**2.2. Setup flight control system** →	**2.3. Prepare aircraft for launch**
Pack Load into transport Transport to location	Verify mission plan Verify environmental conditions are within limits Setup ground control station (GCS) Setup remote GCS [I / II / III] Verify data link [I / II / III] Verify command and control [III]	Install flight components Install and power on payload Power on aircraft at launch point Verify all communication links [I / II / III] Verify mission plan

↓

3. Flight		
3.1 Start mission →	**3.2. Fly mission plan** →	**3.3. End mission**
All crew remain clear of aircraft Initiate launch [III] Verify successful launch Enter mission plan [III]	Continually monitor airspace Ensure performance within limits [I / II / III] Monitor flight path [I / II / III] Continually check environmental conditions Make payload updates (if needed) [II / III] Make flight updates (if needed) [III] Address off-nominal aircraft performance [III]	Verify landing position is clear Initiate landing sequence [III] Verify successful landing Verify collected data integrity Perform repeat missions (as needed)

↓

4. Post-Flight			
4.1. Mission shutdown →	**4.2. Transport home** →	**4.3. Debrief** →	**4.4 Aircraft Maintenance**
Uninstall payload Deactivate remote C2 data link [I / II / III] Shutdown GCS Shutdown aircraft	Pack Load into transport Transport to home	Discuss mission outcome Address issues Assign roles for post-flight maintenance	Offload payload data Uninstall payload (if needed) Uninstall remote data system [I / II / III] Clean aircraft Perform routine maintenance Perform repairs (if needed)

The utility of the operational concept diagram is organizing all the steps required to perform a mission without the system. Adding the system into the workflow of mission planning through mission completion, the concept of operation using the system becomes more clear. Critically, the completion of the operational concept showed that the system does not have much of an impact on mission planning, pre-flight setup, or post-flight tasks. The system does however contribute to major changes in flight procedures depending on the phase. During Phase I the system will only be used to monitor flight status, whereas during Phases II and III the system will also be used by the remote pilot to send commands to either the payload or vehicle. The exact mechanism for these functions were not determined using the operational concept but need for that functionality was identified.

Table 4: High-level operational concept for the system.

Phase	Mission Planning	Pre-Flight	Flight	Post-Flight	Applications
I	- Install	- Setup remote GCS - Establish data link	- Monitor flight	- Disconnect data link - Uninstall system - Offload data	- Operational concept feasibility for flight control
II	- Install	- Setup remote GCS - Establish data link - Establish payload link	- Monitor flight - Payload commands	- Disconnect data link - Uninstall system - Offload data	- Remote payload operation
III	- Install	- Setup remote GCS - Establish data link - Establish payload link - Establish vehicle command	- Monitor flight - Payload commands - Vehicle commands	- Disconnect data link - Uninstall system - Offload data	- Remote flight control of vehicle for mission

The development of the mission operational concept was based on the structure for building missions by the AFL. This ensured all three phases of the system will integrate seamlessly with the procedures used by the AFL. However, for integration with another operational structure, some small tweaks may be required. The changes would likely be small, but worth investigating before attempting to use this operational concept as a model.

3.2.2 External System Diagrams

Utilizing the operational concept, the operational understanding deepened by developing external system diagrams. External system diagrams describe the system in terms of the surrounding systems and context entities. In this case, the system connected with external systems such as the vehicle, global positioning system (GPS) infrastructure, and data link for communication with the remote ground station. The context entities included the flight characteristics of the vehicle, local weather, and the remote pilot input, when applicable in Phase III.

The simple external system diagram is a tool for understanding what the system will need to interact with, both functionally and physically. However, it describes those components at a high level as an introduction to the system in the context of the operational concept. The simple external system diagram is shown in **Figure 2.**

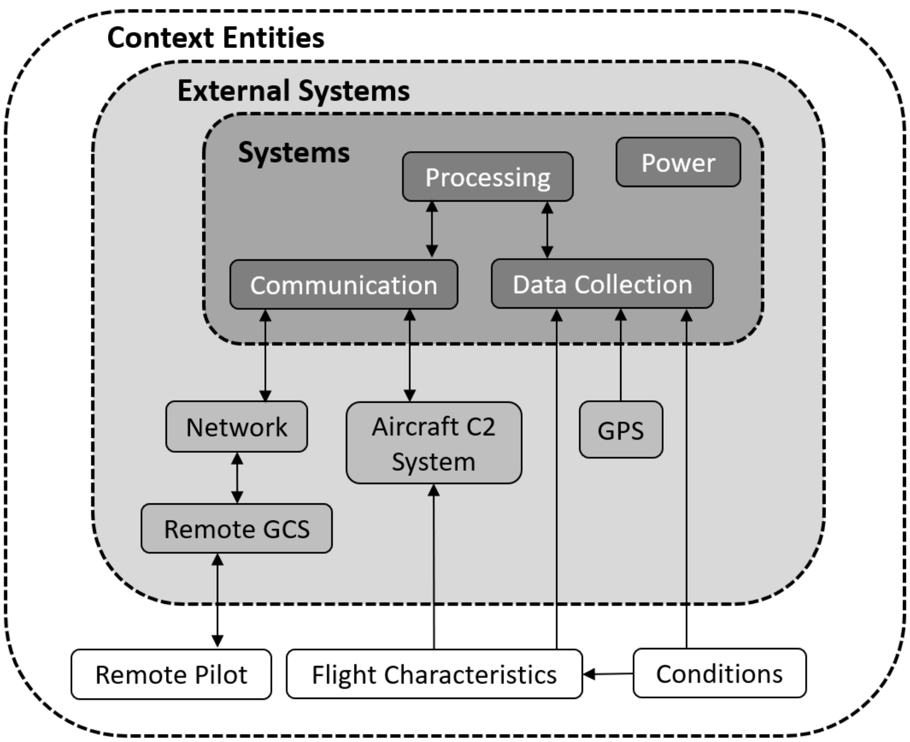

Figure 2: Simple external system diagram of the system.

The simple external system diagram shows that the system represented by the black oval consists of four main sub-systems which interact with various external systems and context entities. The communication sub-system connects to the external network which then connects with the remote ground control station. The communication system also connects with the vehicle command and control system to relay commands from the remote pilot during Phase III. The data collection sub-system connects with the GPS external system because it will collect GPS data. Furthermore, the data collection sub-system connects with the weather conditions and flight characteristics because the sensors in the system will collect data dependent on how the vehicle is flying. Internal to the system the processing sub-system links the data collection and communication, handling all the processes. Finally, the system is powered by an internal power source.

Using the relations developed by the simple external system diagram, a more detailed external system diagram evolved by considering the system interactions in a sequential manner. Essentially, the sequential nature of the external system diagram illustrates the system function as a loop of the data transmission. The detailed external system diagram is shown in **Figure 3**. This version of the diagram is the Phase III case, where the system is being used in place of the standard vehicle control system, thus the remote pilot is fully in command. The diagram reveals the remote pilot in the context of the operational concept of the system. After the data about the flight status has been collected, the remote pilot can see that data and decide to send a command to the vehicle. That command changes the flight characteristics which feed back into the data collection, thus completing the loop.

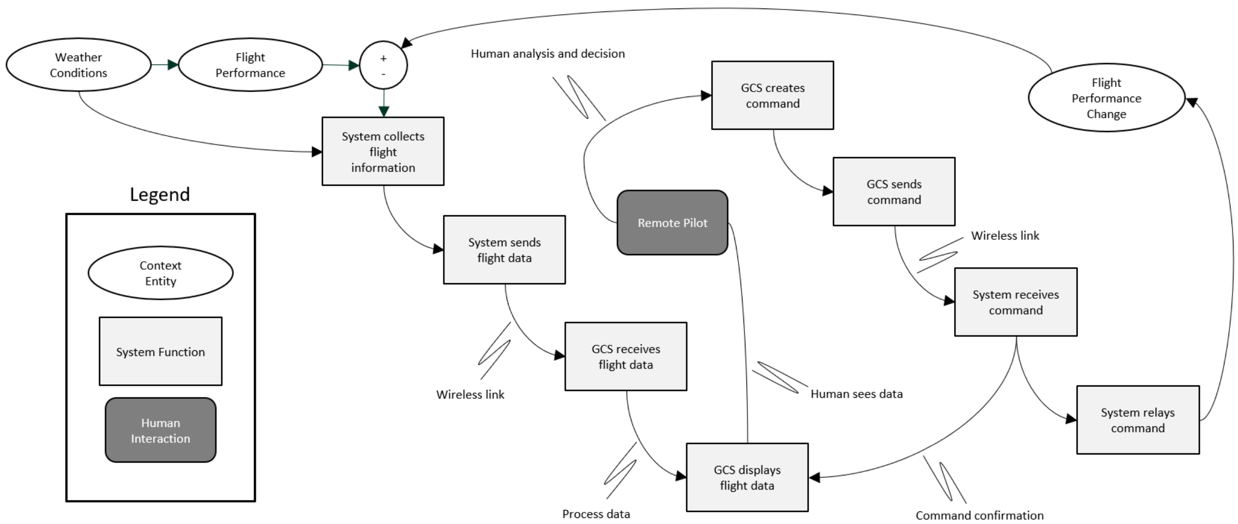

Figure 3: External system diagram of Phase III data control loop.

The Phase III case represents the most complicated operation of the system. Considering the operational concept is significantly different during Phase III than I and II, it is prudent to also illustrate the detailed external system diagram for those phases. The diagram showing the operational concept for Phases I and II is displayed in **Figure 4**, and it completes the comparison with the more complicated Phase III data loop. First, because the system is only being used for data collection and display, there is no interaction between the remote operator and the state the vehicle. This significantly reduces the complexity of the system because there is no feedback from the data collected. Instead, the vehicle is controlled using whatever standard control package is normally used while the remote data collection system acts as an open loop data collection system.

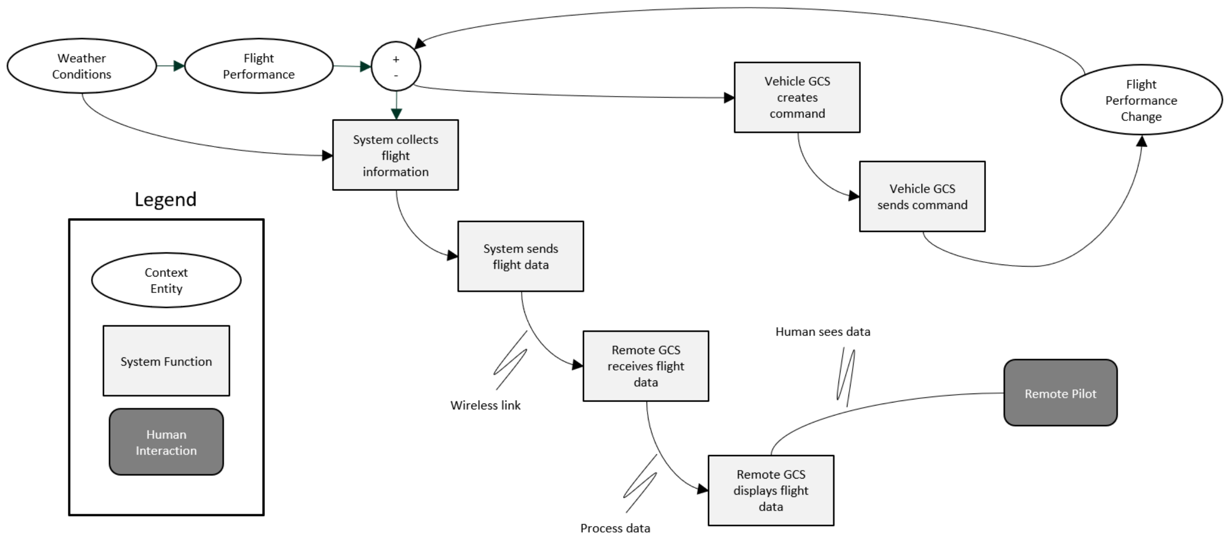

Figure 4: External system diagram of the Phase I and II data control loop.

The weather and flight performance are measured by the system. That information is processed and sent to the remote ground station. The remote ground station processes the data and displays it for the remote operator. For Phases I and II this is the end of the open loop and the data will continue flowing through that sequence. However, for Phase III, the operator will be able to make decisions based on the information displayed on screen. In the case the operator needs to intervene in the nominal operation of the flight, a command is created at the remote ground station by the operator. Then, the command is sent to the on-board system where it is processed. Finally, the on-board system sends the command to the remote vehicle where the action desired by the remote operator is completed. The subsequent data collected by the system should confirm the change requested by the operator.

3.2.3 Objective Hierarchy

An objective hierarchy was developed by examining the high-level tasks of each sub-system described in the external system diagrams. The hierarchy was intended to clarify which parts of each sub-system was most important. The components represented by the hierarchy should define values to the stakeholders. The operational objective is:

> Develop a robust communication system which provides the remote pilot in command the necessary data and command structure for mission success.

Each of the three sub-systems connect to this operational objective with their respective attributes. The following sub-sections detail the prioritized objective hierarchy for each of the sub-systems.

3.2.3.1 Data Collection

The data collection subsystem is comprised of the elements of the system which measure the flight characteristics of the vehicle during flight. The data collection subsystem is crucial for the system to perform correctly because the remote operator will have no direct visibility of the vehicle. Sensor data from the data collection subsystem is the only information with which the remote pilot will be able to make flight-critical decisions.

1. *Reliability:*

   The data collection subsystem shall be capable of collecting data without data dropouts or misrepresented data. This capability is pivotal to the functionality of the system because bad or missing data cannot appropriately inform the remote operator's decisions.

2. *Speed:*

   The data collection subsystem shall be capable of transmitting data at a reasonable rate for real-time decisions to be made. This aspect of the data collection is almost as important as the reliability, but correct data which is reported slowly is more useful than incorrect data which is reported quickly.

3. *Accuracy:*

   The data collection subsystem shall collect data which appropriately represents the state of the system. It does not need to perform highly precise measurements at the expense of reliability or speed. To avoid false indications of system issues, the system needs to provide accurate data, but highly precise data is unnecessary.

3.2.3.2 Data Processing

The data processing subsystem provides task management and data handling for the system in real-time. It is imperative that the data processing unit perform tasks without distortion of the data. Furthermore, the data processing subsystem is responsible for the timing of all the tasks for the system, so it is critical for this system to optimize functionality while reducing the amount of computation to reduce cycle times.

1. *Reliability:*

   The data processing subsystem shall collect, synthesize, and prepare data for transmission without misinterpreting or misrepresenting the data. All the data passing into and out of the system relies on the ability of the data processing unit to maintain and check the fidelity of that data.

2. *Prioritization:*

   The data processing subsystem shall control the order in which tasks occur if they cannot be run simultaneously. This task management will ensure that critical tasks are performed before tasks which are not as important for flight operations.

3. *Multitasking:*

   The data processing subsystem shall run tasks simultaneously if possible. The ability for the data processing subsystem to handle multiple tasks concurrently in combination with the task scheduling prioritization will allow for complete control of the data collection, processing, and communication systems.

4. *Speed:*

   The data processing subsystem shall function at a speed which does not inhibit the overall function of the system. If the data processing subsystem can prioritize and run processes simultaneously, the speed of the system should not be an issue, but a noted concern if processing tasks become too cumbersome for the processing unit.

3.2.3.3 Communication

The communication subsystem is the defining element of the system. Without the ability to communicate the collected data, the system does not have a purpose. Collecting and processing data on the vehicle is only useful in this operational concept if that data can be used in real-time to inform the remote operator of the state of the vehicle.

1. *Reliability:*

   The communication subsystem shall not lose data packets, garble data packets, or drop communication link during flight. There are methods for ensure the fidelity of the transmitted data but losing communication link could be devastating for the mission. Thus, a reliable system is imperative, with corrective actions that are taken if the data link is lost.

2. *Speed:*

   The speed of the communication subsystem shall be within the allowable real-time limits for the operator.

3. *Security:*

   The communication subsystem shall be secure to outside threat of hacking or other methods to illicitly gain control of the vehicle. For this thesis, no work will be put into the security of the system, but security will be necessary as the project progresses.

The limits of operation were vague in the section above. The details of the limits will be filled in during the system design section. For example, the speed at which data must be collected, processed, and transferred will depend on the needs of the system stakeholders and specific vehicle for which the system will be designed.

3.2.4 Stakeholder Needs

To complete the operational concept, a list of the stakeholder needs based on the information gathered from the detailed operational concept diagram, external system diagrams, and operational hierarchy was compiled. Following the guidelines in Lockheed Martin's Core System Requirements Specification the stakeholder needs were separated into functional, interface, non-functional, enabling needs (Martin, 12). Each need identified was linked to the justification for inclusion, derivation, and validation technique. An additional section was included to address constraints. The stakeholder needs represent the high-level elements of the operational concept which will be required to function as needed. As part of the description of each of the needs, their respective verification methods were also included. The verification methods take one of four forms:

- Demonstrate: The need is verified by the system without any external equipment.
- Test: The need is verified using an external piece of equipment.
- Analyze: The need is verified through logical conclusion or mathematical analysis.
- Inspect: The need is verified by visual inspection.

3.2.4.1 Functional Needs

The functional needs correspond with actions the system shall be capable of performing. At its core, this system serves as a relay between the remote operator and vehicle which means the functional needs of the system are related to the communication system. These parameters can be qualitative or quantitative depending on whether they are performance metrics or general functional needs.

1. *The system shall reliably connect with the remote ground control station (GCS) and vehicle command and control system:*

    It is crucial that the data connection between the vehicle and remote GCS be robust for safety. This need is a result of the connection between the system on board the vehicle and GCS in the external system diagram combined with the reliability ranked highest in the objective hierarchy. There is no metric associated with this need, but it necessitates a demonstration of the system's robust qualities for verification.

2. *The system shall support wireless communication with the remote ground station:*

   Wireless communication between the system and remote GCS is the central purpose of the system for Phases I, II, and III. Data must be transmitted from the sensors on the vehicle to the remote GCS to provide the remote operator with data that can inform any flight-critical decisions. This need was derived from the external system diagrams. It is verified by the system demonstrating the capability of the communication system.

3. *The system shall store data if collection rate is higher than transmission rate:*

   There are cases where the flight data collection rate will be higher than the rate at which the information can be transmitted to the remote GCS. In these cases, the system shall be able to store data at the highest rate while sending less data to the remote GCS. This need was suggested by the stakeholders to ensure high-fidelity data is recorded somewhere if the communication system does not have the bandwidth. It is verified by demonstration.

4. *The system shall contain logic for multitasking:*

   The variety of tasks the system will need to handle have different execution times and update needs. Thus, the system shall be capable of running tasks at different rates in parallel to ensure higher priority tasks are executed in a timely manner. This need is derived from the operational hierarchy and is verified by demonstration on the system.

5. *The system shall possess the capability to verify the integrity of data passing in and out:*

    A major issue with data transfer systems, especially high frequency data transfers, is packet garbling. This means that somewhere between the data collection and data display for the operator, the data changed. Thus, incorrect information was displayed for the remote operator which could cause an unnecessary alarm. This need was identified from the operational hierarchy. It is verified by demonstration of the ability to transmit data without misinterpreting data. Alternatively, it can be verified by intentionally testing the system with a garbled packet to see the response.

6. *The system shall have a power system which can sustain it for at least the duration of a flight:*

    The system will be powered by a source separate from the vehicle. This method reduces the complication of needing to power the system by different power sources. It also means the system has no effect on the system flight time. This need was identified in the external system diagram. It is verified by demonstration and analysis.

7. *The system shall control data collection rates:*

The system controls all the tasks for data collection and transmission. Based on the analysis of data collection and transmission times, the system shall regulate the rates at which these processes are performed to provide the remote operator with the most accurate flight information. This need was derived from the operational hierarchy. It is verified by the analysis of data collection and transmission rates.

8. *The system shall have the bandwidth to send/receive data at a rate consistent with the data collection:*

The bandwidth the communication system has for transmitting data shall be consistent with the amount of data which needs to be collected and sent to the remote GCS. This will depend on the specifics of the data collection and communication systems. It is derived from the external system diagrams and objective hierarchy and is verified through analysis coupled with demonstration.

9. *The system shall have sensors capable of detecting the current inertial state of the vehicle:*

The system needs to communicate to the remote operator what state the system is currently experiencing. In order to provide this information inertial sensors are required for acceleration, rotation rates, and heading. This is derived from the need to ensure flight performance in the full system operational concept. It is verified by inspection and demonstration.

10. The system shall have sensors for determining flight speed characteristics:

The system needs to communicate to the remote operator where the vehicle is in space and what speed it is traveling. This is derived from the need to ensure flight performance is within limits as described in the operational concept. It is verified by inspection and demonstration.

11. The system shall be capable of data transmission with the vehicle control system:

Data transmission to the vehicle is the key feature of the Phase III system. For Phase I and II the system does not need to communicate with the vehicle, but this connection provides the remote operator control of the vehicle. In addition to the physical communication link with the vehicle, the system must also be able to send the correct command signal such that the vehicle can execute the command. This need was derived from the external system diagrams and is verified by the demonstration of capabilities.

3.2.4.2 Interface Needs

Interface stakeholder needs define the interfaces with external systems. These interfaces include the physical size constraints and the communication interfaces.

1. The system shall fit in the payload compartment with a standard mission payload:

The sUAS is being used for a mission which requires a payload to collect data. The system is providing a method for performing the mission but does not supersede the main objective of the mission. Thus, the system must integrate with the vehicle and not interfere with the main payload. This need was derived from the full operational concept. It is verified by inspection and analysis.

2. *The system data storage shall be accessible via wired link, wireless link, or removable card:*

    The data on the system shall be accessible once the vehicle has completed its mission so that the full-fidelity data can be analyzed. This may occur by cable, card, or wireless communication. This need was identified through the operational concept post-flight tasks. The verification method is demonstration.

3. *The system software shall be accessible by wired or wireless computer connection:*

    The system will be programmed with scripts loaded into memory which will be run on the vehicle. That software must be accessible for modification through a wired or wireless connection. This was derived from the operational concept mission planning section. The verification method for this need is demonstration.

3.2.4.3 Non-Functional Needs

The non-functional needs of the system define characteristics such as reliability, maintainability, safety, and environmental requirements.

1. *The system shall not be a cause of electromagnetic interference (EMI) for the vehicle control or communication systems:*

    Electromagnetic interference can be difficult to predict and highly detrimental to a system. The system shall have minimal impact on the vehicle. Thus, the possibility of electromagnetic interference should be mitigated and tested to ensure no issues prior to flight. This need is verified through analysis and testing prior to flight.

2. *The system shall not move the CG beyond the limits of the vehicle:*

   The placement of the system shall also not influence the stability of the vehicle in a significant way. The CG of the vehicle shall not move beyond the limits specified by the manufacturer. This need was identified in the operational concept at the installation task. It is verified by inspection and analysis.

3. *The system shall not increase the weight of the vehicle beyond the max takeoff weight:*

   The system needs to remain as light as possible to impact the performance and payload carrying capability of the vehicle as little as possible. This was derived from the operational concept installation task. During installation the weight of the should be measured prior to installation to ensure the weight limit is not breached. The verification process for this need is through testing the weight with a scale.

4. *The system shall be robust to normal flight vibration:*

   The system shall be able to withstand the normal flight conditions of the vehicle which will include vibrations induced by the drive motors. This need was identified through the reliability section of the objective hierarchy. It is verified by demonstration or analysis.

5. *The system shall be robust to physical shock impact landings on the ground:*

   The system shall be able to withstand normal conditions of the vehicle, which may include harsh landings and failed takeoffs. In lieu of characterization of these events via some measurement, the system shall be over-engineered to handle shock events. This need was derived from stakeholder input. The verification of this need is possible through demonstration or analysis.

3.2.4.4 Enabling Needs

The enabling needs describe the production, development, testing, training, support, deployment, and disposal of the system.

1. *The system components shall be readily available:*

   To reduce the burden on the design process, any components selected for the system shall be readily available. This need is verified by inspection.

2. *The system components shall be technology which already exists:*

   The design process for this system is not the platform for developing new technologies. Thus, the components selected for the system shall already exist and be available. This need is verified by inspection.

3. *The development of the system shall stay within budget:*

   The development process will have a budget which needs to be considered. This is the cost requirement which is validated through analysis and inspection.

4. *The development of the system shall remain within the time schedule:*

   The development process will have a project timeline which needs to be considered. This time constraint is validated through inspection.

3.2.4.5 Constraints

The constraints pertain to how the system will be built and deployed. The listed items attempt to identify and solve the tradeoffs which may be found during development.

1. *The system reliability and fidelity shall take precedent over communication speeds:*

   It is more important that the correct information be transmitted to the remote operator than the information be transferred quickly. This need is derived from the operational hierarchy. It is validated by demonstration.

2. *The data collection speeds shall be verifiable using a software in-the-loop counter:*

   It will be most accurate to use the system for verifying the rates at which data is collected. The system is already controlling the rates, so reporting them should be trivial. This need is verified by demonstration.

3. *The data transfer rate between the vehicle and remote data system shall be validated by an on-board counter:*

   The data rate between the system and vehicle shall be monitored by the system. The system will be issuing commands, so it will have the send time, but some work will be required to receiving a receipt for command execution. However, if this data is to be collected and verified, it must be done by the system. Verification is through demonstration.

4. *The data transfer rate between the remote GCS and data system shall be validated by the GCS transfer protocol system:*

   The transfer rate between the system and the remote GCS can be calculated by the GCS. Given the much higher processing capability, any calculations which can be done on the GCS computer should be run there. This need is verified through demonstration.

3.2.5 Operational Concept Summary

The operational concept established the foundation for the development of a full system architecture. It identified the general vision for the system from the view of the stakeholders. The operational concept also served as a starting point for the concurrent development of the functional and physical architectures. By existing as a high-level description of the system functionality, the operational concept guides the description of how the inputs are converted to outputs in the functional and physical architectures. Due to the iterative nature of systems development, it was also modified during the continued development of the full system architecture. The information documented above was the final version of the operational concept after completion of the full system development.

3.3 Functional Architecture

The conclusion of the operational concept development was the list of stakeholder needs. The co-development of the functional and physical architectures begin to resolve how those stakeholder needs will be achieved in the final design of the system. The functional architecture sets out to describe the functions which will achieve the objectives posed by the operational concept. Concurrently, the physical architecture describes the elements of the system which can perform those functions.

The development of the functional architecture began by organizing the functions addressing objectives from the operational concept into a hierarchy. Then, the high-level functions which linked inputs and outputs of the system were identified. Third, the stakeholders were asked for input regarding the functional decomposition of the system. Finally, the inputs and outputs on the external system diagrams were linked to functions. In Buede's treatment of the functional architecture development, there is also a step for integrating fault tolerances and security functionality, but this step was ignored during this development due to time constraints. The continuation of this project should look into the safety and security of the system.

3.3.1 Functional Hierarchy

A hierarchy of the functions needed to satisfy the objectives presented by the operational concept contain details about the functions required to satisfy those objectives. This functional hierarchy includes functions which will be handled by the on-board system. There are corresponding functions which are employed by the remote ground station, but for simplicity only the on-board functions are represented in this decomposition.

3.3.1.1 Receive Commands from Remote GCS

The most important function for the on-board system is the ability to receive commands from the remote GCS. Although this function is not required for Phases I and II, intervention by the remote operator is the most important data the system will handle. If the remote operator needs to provide input to the vehicle, that data should be prioritized because it may be critical to safety. The most important part of the function is ensuring data integrity. Secondary to the integrity of the command is a receipt so the operator is sure the sent command was received and executed.

3.3.1.2 Transmit Data to Remote GCS

The second most important function for the system is to transmit the collected data to the remote ground control station. This function is critical to the purpose of the system. If the data cannot be transmitted to the remote operator, the system is not functional. For the transmission function, data integrity is more important than speed and scheduling because it is crucial that the correct data be sent to the remote operator than fast data which is wrong. However, managing the speed and schedule is still important to the proper function of the system.

3.3.1.3 Collect Flight Data

The collection of flight data is the least important function for the system because, although it is important to the functionality of the system, it is not as time dependent as commands or essential as data transmission. Data collection without transmission would be useless and transmitting good flight data instead of receiving time-sensitive commands could be detrimental to mission success. The sub task of managing the data collection schedule is most important because that prioritizes which data is the most important to collect and when those sensors should be polled. Following the scheduling, processing the data on board is less important because if the need arises, raw data can be sent to the remote ground station where the data can be processed.

3.3.2 Input and Output Relationships

The functions of the system were provided more detail by resolving the input/output relationships more closely. The main reference for this was the simple external system diagram. The input/output relationships depend on data interactions due to the nature of the system. The relationships are displayed in **Table 5**.

Table 5: Functional input/output relationships.

Input		Output
Flight characteristics	→	Raw flight data
GPS data	→	Raw flight data
Raw flight data	→	Processed flight data
Payload command from remote operator	→	Signal to payload
Flight command from remote operator	→	Signal to vehicle control system

The physical characteristics of the flight are recorded by the data collection system through a suite of sensors. Thus, the input into the system is the flight characteristics of the vehicle the system is on, and the output is raw flight data. The same is true for the GPS data collected by the system. It polls the GPS for position and records that data on board as raw flight data. Then, that raw flight data is processed by the on-board processing unit before being transmitted to the remote ground control station. This constitutes the input/output relationship of data from the vehicle dynamics to useful information for the remote operator.

The other direction of data flow is the remote operator issuing commands for the system. Phases II and III will have this capability. For Phase II the commands will go to the payload, not the vehicle while Phase III should be capable of handling either case. The input into the system is a command sent by the remote operator. This command is received, and the corresponding output is a signal to either the payload or vehicle depending on the command.

3.3.3 Stakeholder Input

Throughout the development process, the stakeholders were asked for input regarding the overall process and specific development details. Their input was critical for identifying many of the functional needs of the system. At the end, the system stakeholder was consulted to look for significant issues with the outcome of the development. Again, the stakeholders were able to provide insight and point development in a direction beneficial to the project without finding any major deficiencies in the functional development.

3.4 Physical Architecture

Along with the development of the functional architecture, the physical architecture is developed concurrently to ensure the functional needs can be met within manageable physical elements. Initially, these elements are vague, but as more focus is placed on functionality the specifics of the elements become clear. These elements can also be left vague to allow for more configuration during the specific design.

Thus, for the development of this physical architecture, the first step was to describe the elements needed to satisfy objectives set by the operational concept and functions set by the functional architecture. Then, that design was refined as needed during subsequent iterations. A diagram describing the physical architecture is shown in **Figure 5.**

The physical architecture consists of three main parts under the overall system. First, the on-board subsystem called the flight system in **Figure 5**. Within that system is the main structure which will contain the processing, data acquisition, and communication units. Additionally, there is another element for sensors which may need to be mounted to the vehicle separate from the main structure.

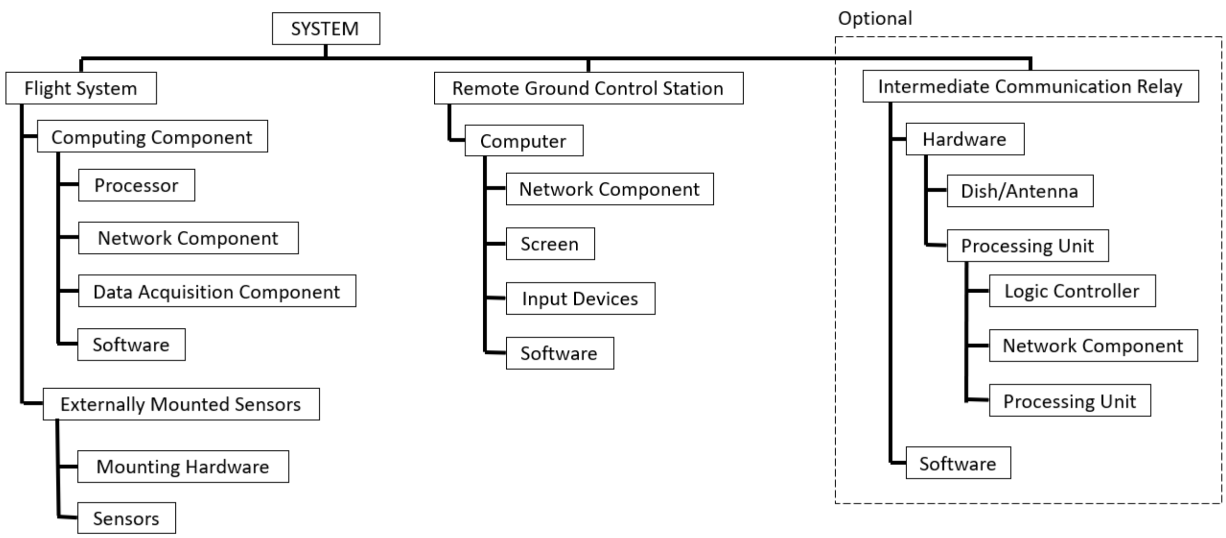

Figure 5: The physical architecture of the system.

Second, the remote ground control station describes the elements needed for the remote operator to access the data collected by the flight system. All that is required of this system is a computer connected to the same network the flight system is on whether that be by direct radio, network, or internet.

Finally, the intermediate communication relay can be part of the system if there is a need. The communication link between the flight system and remote ground control station was intended to exist using only components within those systems, but communication with the stakeholders revealed a possibility of using the system with an intermediate relay, so it has been included in the physical architecture. Functionally, its purpose would be to receive data from the flight system or remote ground control station and sending that data to the other. Although the intermediate communication relay exists in the architecture, it is not a required component of the system.

3.5 Operational Architecture

The operational architecture could be addressed with all the compiled information from the development of the system architecture. The operational concept describes the system using a composition of the results from the operational concept, functional architecture, and physical architecture. This operational architecture was used as the framework for applying design-specific requirements during the subsequent steps for this thesis.

3.5.1 Physical Component Functions

The functional and physical architectures developed in tandem are combined to reveal which parts of the physical architecture perform which functions. This provides more insight for the needs of the components which will eventually fill in the framework of the system architecture. The result of this composition is found in **Figure 6.**

The flight system computing component, in tandem with the sensors, are responsible for performing functions to control the data rates, any logical processing, and send/receive data. Although sensors are generally passive, the externally mounted sensors perform the function of collecting data.

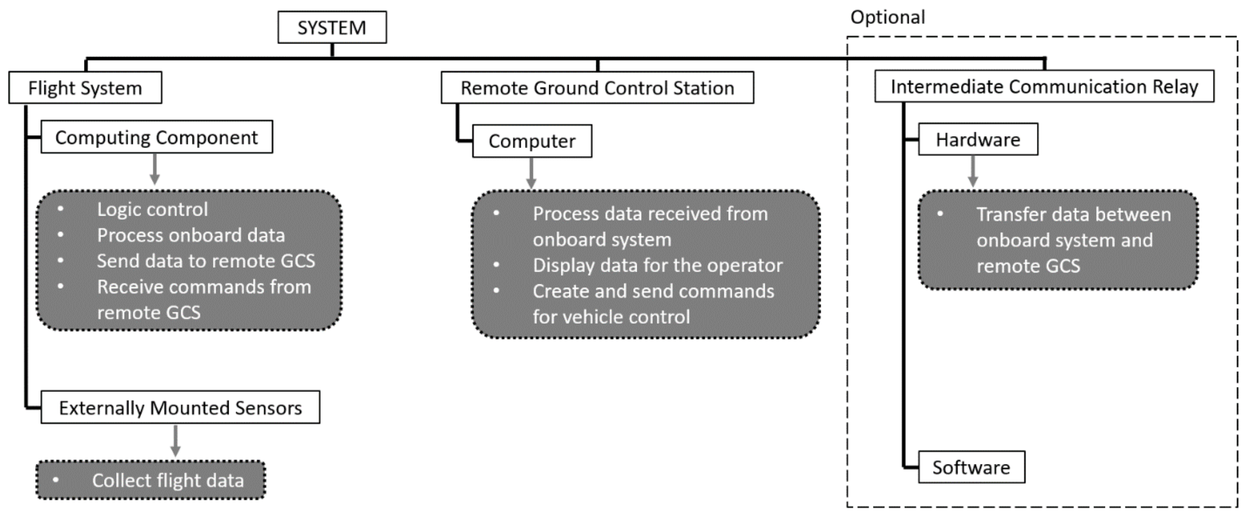

Figure 6: Functions performed by physical elements of the system.

The remote ground control station processes and displays data received from the flight system for the remote operator. It also performs the function of creating the various commands to control the payload or vehicle during Phase II and III of development.

The intermediate communication relay only functions as a data transfer unit. It receives data from either the flight system or remote ground control station and send that data to the other system. Again, this system is not required, and should only be used if necessary. It has been in included in the architecture for completeness.

3.5.2 Functional Flow, Activation, and Control

The characterization of the functional flow, activation, and control structures for the system provided an overview of the flow of information through the system. Furthermore, it provided a vision for some of the control software which would be required. This adds more detail to how he functions described in the functional architecture are motivated and connected. Each function is linked with an output, required inputs, and required controls to perform as expected. The functional flow, activation, and control table is shown in **Table 6.**

The *scheduling control* function is the feature of the on-board logical processing which prioritizes functions and tasks for the system. It is controlled by the code written to direct the task scheduling and requires a time input to properly assign the next task to run. For this reason, the scheduling control function is a required control for many of the subsequent tasks in the table because it is the function which controls the rest of the system.

Table 6: The functional flow, activation, and control for the system.

Function	Output	Required Input	Required Control
F.1 Scheduling control	**O.1** Next function to run	**I.1** Unified time	**C.1** Scheduling script
F.2 Collect flight data	**O.2** Raw instrument data	**I.2** Sensors outputs	**F.1** Scheduling control **C.2** Sensor script
F.3 Process flight data	**O.3** Processed data	**O.2** Raw instrument data	**F.1** Scheduling control **C.3** Processing script
F.4 Send flight data	**O.4** Transmitted data	**O.3** Processed data	**F.1** Scheduling control **C.4** Communication script
F.5 Display data	**O.5** Human-readable data	**O.4** Transmitted data	
F.6 GCS commands	**O.6** Control commands	**I.3** Human input	
F.7 Send command data	**O.7** Transmitted command	**O.6** Control commands	**C.5** Communication script

The *collect flight data*, *process on board data*, and *send data* functions are all processes which occur on the flight system as directed by the scheduling control function. These functions work together to collect flight data, process that data, and send the data down to the remote ground control station as described in the functional architecture. Furthermore, in the functional flow diagram, the connection between the functions is clear as well as the connection with some physical features of the system such as the sensors. The makes the functional flow, activation, and control diagram powerful for describing the system architecture.

In addition to the on-board functions, the *process data at GCS*, *display data*, and *create commands at GCS* functions are all occurring on the remote ground control station computer. These functions flow to receive data and display data to the operator. Then, with the operators decision-making ability in Phases II and III, a command can be issued to the system for controlling the payload or vehicle if needed.

The increased complexity of functions described by the functional flow, activation, and control table highlight the deepened understanding of the needs of the system. The table also reveals the interconnection between many of the elements of the system. With this heightened understanding of the system as a whole, the system architecture begins to take the form of a framework which could support specific design requirements.

3.6 Summary of System Architecture Development

This chapter described the development of the operational concept, functional architecture, physical architecture, and operational architecture. These development processes worked to take high-level ideas about the function of a proposed system and turn them into an operational framework which can be utilized for a specific design. The information presented in this chapter represented the end of development for the system architecture with no mention of the initial iterations. However, there was much iterative tuning to produce the system architecture framework. Finally, the validation of this system architecture is accomplished through the design phase and will be discussed during the **Prototype Development** and **Results** sections.

4. PHASE I PROTOTYPE DEVELOPMENT

The development of the system architecture describing a framework for supporting ground station monitoring and control of a remote sUAS was the first segment of the full development process for this thesis. With the framework in place, an initial prototype of the Phase I design is desired by the AFL as a test platform for the operational concept. As a test platform, the first prototype will serve as the basis for all future work utilizing the system architecture and will serve the purpose of validating the system architecture. Thus, the stages for completing the prototype development are to fill in the system architecture with specific requirements, make design decisions based on trade studies, and finalize a detailed version of the prototype design. Following the completion of these steps, the prototype should be prepared for validation and testing.

Prototype development, combined with the earlier development of the system architecture, is an iterative process where the deficiencies in the design process can be addressed by reverting back to the issues and making logical changes to influence the design output. These modifications can be made in the design space or to parts of the system architecture. For this treatment of the prototype development, changes which were made to the system architecture will not be addressed, as there is only space for the final forms of the developed systems.

To start the prototype design requirements needed to be defined. These requirements were developed by providing specific values to the user needs from the system architecture. By filling in the general user needs section with values specific to the AFL and vehicles, these needs became requirements. Some further requirements were established from stakeholder input.

With the requirements in place, designs with specific components could be compiled. Trade studies of single components and combined design concepts were evaluated in various trades studies to determine which, if any, design options satisfy the requirements. In the end, many design options satisfied the requirements, which forced the stakeholders to add additional constraints and make design decisions based on the available options. Some of the additional considerations made were, user friendliness, long-term support, and familiarity with components of the designs. This design is the Phase I prototype which will be used to validate the system architecture and AFL-specific design. Furthermore, it will serve as the base for the continuation of the project into Phases II and III.

4.1 Requirements

The requirements used for the prototype development were derived from a combination of the two platforms the AFL intended to use the system on. These two vehicles are the fixed-wing Altavian Nova and the rotorcraft Aerovironment Vapor 55. The reason these two vehicles were chosen as the basis of the prototype development was that they offer different operational capabilities for testing the system. The differences in flight characteristics between the two vehicles vary in speed, acceleration, duration, and mission capabilities. The two systems encompass the range of vehicles the system is expected to work with.

The method for defining requirements was to insert vehicle specific information into the user needs outlined in the system architecture. In doing so, the system architecture provided the framework defining the system which worked with the Nova and Vapor 55 vehicles. In addition to the requirements developed using the system architecture, the stakeholders were invited to add requirements and constraints as they saw fit. Furthermore, this process illuminated extra requirements which were not identified as user needs during the system architecture development. If the new requirement was an oversight, they were added to the user needs section of the system architecture, otherwise, the new requirement was added as a stand-alone part of the requirement section. At the end, the ability for the requirements defined using the system architecture to support design options confirms that the system architecture was successfully defined for this application.

4.1.1 Functional Requirements

The functional requirements specify actionable behaviors of the system. The functional requirements follow the user need from the system architecture which they were derived from. The justification for the requirement follows.

1. *The system shall reliably connect with the remote ground station and vehicle command and control system:*

    The signal between the remote GCS and the system shall not be lost for more than 8 seconds.

    The Nova is capable of flying at 50 knots meaning it can cover about 1/8 mile in 8 seconds. Traveling long distances without data updates should be avoided for safety. The communication time limit of 8 seconds ensures that the remote operator shall have updated flight data in intervals shorter than 1/8 mile. The Vapor 55 much slower than the Nova, so this requirement is based only on the Nova.

2. *The system shall support wireless communication with the remote ground station:*

The system shall communicate via wireless internet connection, local network, or radio.

The long-term vision for the system requires that the data be transmitted from the vehicle to the remote GCS by a wireless internet connection. Thus, the capability to send data files over wireless communication shall be tested. For testing it is unnecessary to spend the money on a wireless internet contract because communication through a direct radio link or wireless network can act in place of the wireless internet. The local network would also demonstrate the ability to send data files, whereas the radio communication would show a serial data communication capability.

3. *The system shall store data if collection rate is higher than transmission rate:*

A minimum 8 GB data storage device such as an SD card or flash memory chip shall be present to store data on board.

This requirement is based on the expected data package size of about 60 bytes being collected twice each second for two hours. This yields an 800-megabyte file meaning that the system needs to be capable of storing at a minimum 800 megabytes per flight. Ten times that amount of storage was chosen as the minimum to allow for overrun, extra data collection capabilities, and storing data for multiple flights.

4. *The system shall contain logic for multitasking:*

The system processor shall be capable of multitasking or being programmed as a task scheduler to simulate multitasking.

There are two implementations of processing units. A central processing unit (CPU) is capable of executing many tasks at once. CPUs are the chips which perform the logical processing for most computers. A microprocessor is a single chip version of a CPU which is only capable of executing a single task at once. Microprocessors are the chips at the heart of microcontrollers (MCU). Therefore, the integration of either of these two options will require a different software implementation to enable the execution of simultaneous tasks. With a CPU the processing unit has the capability by default, whereas with the microprocessor, a strategically programmed task scheduler will be required to meet this requirement.

5. *The system shall possess the capability to verify the integrity of data passing in and out:*

The system shall employ a check-sum algorithm for data transfer integrity verification.

Data integrity is important for the system because reporting incorrect data increases the likelihood of an incorrect action taken by the remote operator. There are a number of industry-standard methods for ensuring data integrity including SH-2 and SH-3 checksums. Some form of checksum shall be employed for detecting errors caused by the data transfer functions of the system.

6. *The system shall have a power system which can sustain it for at least the duration of a flight:*

The system shall have a battery life of at least three hours.

The three-hour battery life requirement is directly attributed to the flight time of the Nova. The Nova is listed with an 80-minute endurance. It was decided at least a factor of two was needed for the system in the case of a longer flight or other unexpected circumstance. The exact size of the battery is dependent on the power draw of the system. The Vapor endurance is around 30 minutes giving the system significant overhead for Vapor flights.

7. *The system shall control data collection rates:*

The logic system shall control the rate of data collection and communication.

Tasks should be scheduled using a prioritized list. Some tasks are more important than others and should be given processing time over other tasks. The processing unit which will be programmed with a script handling all the task scheduling shall be capable of executing tasks in a manner which provides the most control and understanding of the vehicle at all times.

8. *The system shall have the bandwidth to send/receive data at a rate consistent with the data collection:*

   The system communication bandwidth must be greater than 240 bytes/second.

   The expected acceleration, rotation, heading, altitude, position, and power data is expected to yield a packet of about 50 bytes for the Vapor 55 and 60 bytes for the Nova. To support sending two packets of data each second, the communication system shall be capable of sending 240 bytes each second. A more capable communication would give overhead for extra data or higher communication rates which are desired for a more robust system.

9. *The system shall have sensors capable of detecting the current inertial state of the vehicle:*

   The system shall interface with accelerometers, gyroscopes, and magnetometers for making inertial measurement.

   Nine degree-of-freedom inertial measurement units are capable of providing information about the state of the vehicle. A combination of these sensors shall provide the remote operator with a pointing vector for the vehicle.

10. The system shall have sensors for determining flight speed characteristics:

The system shall be interface with GPS and pitot-static air measurement systems to determine flight characteristics.

The two methods for determining speed are using GPS and a differential pressure measurement. GPS can only provide ground speed whereas the pitot-static pressure measurement can provide airspeed. A combination of the two sensors shall be used to provide the operator with information about the speed and position of the vehicle. The Vapor 55 might only be able to support GPS speed because the pitot-static airspeed measurement requires free stream air. The down wash of the main rotor on the Vapor 55 will likely interfere with the forward airspeed measurement.

11. The system shall be capable of data transmission with the vehicle control system:

The system shall communicate through an RS-232 port and 16-pin proprietary Altavian connector.

For Phase III the system needs to be capable of relaying commands to the vehicle control system. The Vapor 55 has an RS-232 port for interacting with the control system and the Nova has a proprietary 16-pin connector for control system communication. It is also worth noting that the system will need to be capable of providing the right type of signal for either of these vehicles. Currently, the signal type for commanding action is unknown.

4.1.2 Interface Requirements

The interface requirements define the connections the system will have to external systems and components.

1. *The system shall fit in the payload compartment with a standard mission payload:*

   The system shall be smaller than 7.5 x 4.5 x 4 inches.

   The system shall be kept as small as possible to minimize the impact it has on the vehicles. The Vapor 55 has a large payload volume of 20 x 6.5 x 5 inches. This volume is partially filled by 12 x 5 x 4 inches of batteries but can be configured to handle a significant array of weight distributions. Furthermore, the Vapor has a mount for a camera not included in that volume, meaning that the mission payload does not interfere with the space for the system. The Nova has a smaller payload volume at 7.5 x 4.5 x 4 inches. Currently, it is unknown what size the mission payload will be, so a maximum limit has been placed on the system at the size of the Nova payload bay with the intent of keeping it much smaller.

2. *The system data storage shall be accessible via wired link, wireless link, or removable card:*

   The system data storage shall be accessed through a USB, SD card, or wireless transmission.

   The data collected during flight shall be accessible by the crew once the vehicle is back on the ground. Depending on the setup of the on-board storage, that data will be accessed by a direct link via wire or wireless communication. Alternatively, it can be accessed by pulling a data storage card out of the system and accessing that through another computer.

3. *The system software shall be accessible by wired or wireless computer connection:*

The system shall be programmed or accessed through a wired, network, or Bluetooth connection.

The system will have scripts for running task management, data collection, and communication functions. This code shall be accessible for modifications through a wired, network, or Bluetooth connection. Either the code can be replaced with new code, or it can be modified on board if possible. This will allow for long term flexibility to add or change components.

4.1.3 Non-Functional Requirements

The non-functional requirements define overall operational characteristics of the system such as reliability, environmental, and safety.

1. *The system shall not be a cause of electromagnetic interference (EMI) for the vehicle control or communication systems:*

The system communication shall not interfere with the vehicle communication channels on 900 MHz, 902-928 MHz, and 2400-2485.3 GHz.

Interference between the system and the vehicle communication frequencies could cause incorrect data collection by the system and control issues with the vehicle. Thus, it is in the interest of the design to avoid crosstalk between the two systems. However, the frequencies for the Nova and Vapor 55 are the communication bands that many off-the-shelf components communicate on. A solution may be to allow the systems to communicate on narrow bands within the bands the vehicles work in. Alternatively, many radios perform frequency hopping in their ranges to limit interference. This technique could be used cautiously with significant ground testing to ensure no interference before flight.

2. *The system shall not move the CG beyond the limits of the vehicle:*

The system shall not impact the vehicle stability by moving the center of gravity (CG) beyond the allowable limit.

When the system is installed in the vehicle, its weight and placement shall not move the CG beyond the listed limits of the vehicle. For the Vapor 55, the CG limits are 1 cm forward of the main rotor and 0 cm forward of the main rotor. For the Nova, the CG limit is 14 ± 0.25 inches measured from the motor mount bulkhead.

3. *The system shall not increase the weight of the vehicle beyond the max takeoff weight:*

The system shall weigh less than 2.5 pounds.

A weight analysis of both vehicles was completed for both vehicles. The Vapor 55 had 6.5 pounds available with the standard camera payload installed and the Nova had 4.5 pounds available for the system and payload combined. The standard payload for the Vapor 55 weighs 2 pounds. Assuming the same payload weight for the Nova, that leaves 2.5 pounds for the system.

4. *The system shall be robust to normal flight vibration:*

The system shall survive nominal vibration loads from the Nova and Vapor.

The structure of the system will be rigid to keep components from rattling around due to vibrations, but the effect a specific vibration mode has on components will not be understood unless further analysis is performed.

5. *The system shall be robust to physical shock impact landings on the ground:*

The system shall be robust to shock impacts with the ground up to 50 g.

This value is triple the expected impact velocity (13 feet/s) of the Nova powerless landing from 400 feet based on the glide ratio. From 400 feet, the Vapor is expected to impact the ground at 160 ft/s which is too high to design around. The system will not be over engineered to the point where it would survive catastrophic failure of one of the vehicles.

4.1.4 Enabling Requirements

The enabling requirements represent factors which influence the system's ability to be built, supported, and deployed.

1. *The system components shall be readily available:*

   Components selected for the prototype design trade studies shall be available.

   Components under consideration for the system shall be available for purchase to keep the development time down and remove time wasted considering components which cannot be bought. Furthermore, well-documented components shall be prioritized because these components will be easier to integrate with the system.

2. *The system components shall be technology which already exists:*

   The prototype design shall be built using existing technology.

   Although novel data collection methods and new technologies can provide better data, the design process is not for new technology development in this case. Thus, components considered for the design must already work using existing technology.

3. *The development of the system shall stay within budget:*

The design portion of the project shall be within the budget set by the stakeholders.

A development budget was not set for the project, but all purchases need to be approved by the principle investigator of the AFL.

4. *The development of the system shall remain within the time schedule:*

The prototype design step shall be completed in the time frame allotted by the thesis project.

Thesis projects span one year for most students. The prototype design portion of the project must fit within that year schedule for the full thesis.

5. *Additional Stakeholder Requirements:*

The system shall utilize Viasat products if possible.

California Polytechnic State University had the opportunity to interface with some members of Viasat Inc. (a communications satellite company) and thought this project could benefit from their technology. If there is a possibility of using a Viasat product, there might be an avenue for procurement.

The system shall utilize open source software and hardware.

There are many proprietary systems which make building data collection systems easy such as the National Instruments LabVIEW software, but those system are expensive and often provide the end user with less flexibility over time. Thus, the project shall employ only open source hardware and software solutions to allow for maximum flexibility and control as the project moves forward.

The system shall be modular to allow for ease of modification.

Similar to the proprietary systems, there are sensor packages and auxiliary systems which can be integrated with a data collection system to provide detailed data. In this case, those systems will not be utilized because the ability to add or remove capabilities from the system is highly desirable. Furthermore, this requirement means that all sensors and components should be easy to add and remove from the physical structure of the system.

4.1.5 Constraints

The constraints requirements provide a structure for making decisions between various requirements. Essentially, when two requirements are at odds, these requirements provide a structure for deciding which requirement takes precedent.

1. *The system reliability and fidelity shall take precedent over communication speeds:*

   The system's communication reliability and fidelity shall be preferred to increased communication speed with worse data integrity.

   The software controlling the communication system and task management shall be slowed down if the integrity of the data is being compromised. It is ideal to have both speed and data transfer without losses, but those two factors are usually at odds for systems such as this.

2. *The data collection speeds shall be verifiable using a software in-the-loop counter:*

   The system hardware and software used shall report the data collection speeds.

   The system shall keep a record of the data collection speed even if that data is not transmitted to the remote GCS during flight.

3. *The data transfer rate between the remote GCS and data system shall be validated by the GCS transfer protocol system:*

The remote GCS shall report the data transfer rate.

The communication delay is an important aspect of the system which the remote operator should be able to see. Action should be taken if the delay grows too large and there should be safeguards in place to keep the vehicle safe if the delay between data collection and viewing at the remote GCS becomes too large.

4. *The data transfer rate between the vehicle and remote data system shall be validated by an on-board counter:*

The system shall measure the data transfer rate with the vehicle.

This requirement will be demonstrated during testing. The user should know the system time from command sent by the remote GCS to execution by the vehicle, but a live readout of the data rate is unnecessary because the number of commands sent will be low.

4.2 Component Selection

With the requirements defined, hardware components which could combine to produce a prototype which satisfied all the requirements could be identified. Component research was performed and compiled to provide the most complete analysis of potential design options. This portion of the prototype development was not critical to the validation of the system architecture. The system architecture would be validated by the performance of the system as a whole, not the individual components.

During the initial search for components, it became clear that the most important part of the system was the processing unit. The data acquisition units, sensors, communication systems, and operating system were all dependent on the selection of a processing unit. The interfaces and power distribution capabilities of the processing units change which components could be selected and which were not suitable. Thus, the decision to choose a processor first was made. Then, the other components could be selected to work with the selected processor.

4.2.1 Processor

The component selection process revealed that the most important variable regarding the design of the system was the processing unit. There were two options which could satisfy the requirements. These processing unit options were an MCU or a single board computer (SBC). An SBC allows for a full operating system to be installed and therefore enhances both the versatility and performance of the system. However, an MCU provides the system lower power usage and a more direct connection with the hardware. One specific version of the MCU was packaged an off-the-shelf flight controller called Pixhawk. This specific MCU is already programmed to fly sUAS and therefore seemed like it could be a good option for the system. A list of the processing units considered is available in **Table 7.**

Do note, the list of processing units was comprehensive, showing a range of options across types and capabilities, but was not an exhaustive list. Comparing all the potential options would have taken an exorbitant amount of time inconsistent with the duration of this thesis project. The units selected for comparison were the more well-known, well-documented options on the market making them exceptional candidates for this project which will be worked on by many people.

Table 7: The processing unit options for the system.

Name	Storage	Processor Cores	Processor Speed	RAM	GPIO	Data Ports	Input Power (W)	Cost
Single Board Computers (SBCs)								
Raspberry Pi Zero 1.3	SD Card	1	1 GHz	512 MB	40	0	12.5	$25
Raspberry Pi 4	SD Card	4	1.5 GHz	1 GB	40	5	12.5	$35
UDOO Neo	SD Card	1	1 GHz	512 MB	36	2	15	$50
BeagleBone Black	4 GB	1	1 GHz	512 MB	92	2	12.5	$65
BeagleBone X-15	4 GB	2	1.5 GHz	2 GB	240	7	60	$250
Xtreme/104 SBC	32 GB	1	1.6 GHz	2 GB	0	10	50	$700
Advantech MIC-1810	SSD	2	1.6 GHz	4 GB	24	8	45	$1400
NI sbRIO-9637	SD Card	2	667 MHz	1 GB	96	3	24	$1800
Microprocessors								
Arduino Mega	none	1	16 MHz	256 KB	54	0	18	$39
Arduino Due	none	1	84 MHz	512 KB	54	0	18	$39
Particle Boron	none	1	64 MHz	256 KB	20	0	12.5	$50
Pixhawk 4 Mini	SD Card	1	216 MHz	512 KB	25	1	11	$180

The wide range of cost highlights the vast difference in capabilities for many of these devices. Some options were ruled out because they did not satisfy the design requirements. For example, the proprietary boards, such as the National Instruments and Xtreme/104, were outside the requirements of this project. They had great connectivity potential and were the most stable, supported systems, but they were expensive and did not meet the open source requirement. The Pixhawk flight controller did meet the open source requirement but was expensive and provided built-in sensor capabilities which did not match the modular requirement. Furthermore, as a pre-built controller the lack of data ports would be prohibitive to potential sensor integration.

The Arduino MCUs provided all the necessary capabilities for this system and were the preferred option when interfacing directly with hardware – like the sensors intended in this project. The argument against MCUs was that the SBCs which are comparable in price offer far more flexibility with performance capability going forward. The Arduino MCU would have been capable of the tasks presented by all three phases of this project but would reduce the flexibility. Furthermore, the capabilities of the flight data recorders from research were considered. The MCU-based system was limited compared with the SBC, especially when considering it was not an off-the-shelf system (Brusov, 136), (Taha, 143). Thus, the decision to utilize an SBC was made because they offered the best performance, flexibility, support, and cost. The SBC chosen should be capable of supporting development through Phases I, II, and III as well as future developments yet unseen.

The three SBCs under consideration were the Raspberry Pi 4, Beaglebone Black, and UDOO Neo. These boards were all comparable as far as performance, size, connectivity, power, and price, but there was a best option. The Raspberry Pi 4 was the cheapest, had a stronger processor, and provided the most serial connectors. It did lose out to the Beaglebone Black in general purpose input/output pins (GPIO), but the 40 pins on the Raspberry Pi 4 would be enough for the project. Furthermore, the Raspberry Pi 4 has the longest heritage on the market, thus provides the most support and longevity.

The Raspberry Pi 4 was chosen for building the system around because it had the best infrastructure for long-term development without sacrificing any performance. It would serve as the processing unit for scheduling tasks, collecting analog and digital data, and transmitting that data to the remote GCS. This board would show the prototype capabilities with Phase I and easily be rolled into Phases II and III.

A positive sign for the capability of the system architecture was that there were a host of options which would satisfy user needs and system requirements set for the prototype design. This shows the versatility of the system architecture for accommodating elements of design. Further analysis of the system architecture will be conducted during the subsequent stages of prototype development.

4.2.2 Data Acquisition

The options for the data acquisition unit (DAQ) were split between two types. The first type was a hub which had headers for sensor wires. This hub connected with the Raspberry Pi using a universal serial bus (USB). There was not much documentation which verified these systems could interact seamlessly with the Raspberry Pi. The second type was of the hardware attached on top (HAT) variety. These DAQs were built specifically for the Raspberry Pi and plugged directly into the pin header on the board without interfering with the capabilities of those GPIO pins. The data acquisition options are available in **Table 8**.

The features listed for each of the DAQ units show the digital input/output channels, analog output channels, and analog input channels. These metrics do allow for the comparison of the different options, but the most important element was the analog inputs. More analog inputs meant more analog sensors could be attached for data collection. The Raspberry Pi already has GPIO pins, so the digital pin count was not important. Furthermore, there would be no need for the Raspberry Pi to output an analog signal.

The USB DAQ options provided more flexibility to swap units because they easily plug into the Raspberry Pi's USB ports. However, the unknown communication between the USB DAQ and Raspberry Pi meant that these options could cause delays and issues during prototyping. Furthermore, the documentation to pass on to others working on Phases II and III would need to be extensive to ensure no lag in development. One additional reason to not choose the USB DAQ was the fact that it would make the system larger because it was a separate unit from the Raspberry Pi. This did not mean the USB DAQ would exceed the size requirement for the system, but the system benefits from a small footprint.

Table 8: The data acquisition options for the system.

Name	Digital I/O	Analog In	Analog Out	Cost
Hardware Attached on Top (HAT)				
12-Ch 12-Bit ADC HAT for Raspberry Pi v3.0	0	8	0	$20
PI-16ADC Development Board	0	16	0	$36
Pi-Plates HAT	8	8	4	$45
Measurement Computing 118	0	8	0	$99
Measurement Computing 152	8	2	0	$99
USB Devices				
EECI ADC	0	12	0	$100
PruDAQ by GetLab	96	8	0	$110
LabJack U3	20	16	2	$120
Dataq Model DI-1110	7	8	0	$150
USB Bitscope 10	8	2	0	$245
openDAQ	6	8	0	$250
Omega OM-USB-1608G-Series	8	16	2	$450

Thus, it was in the interest of the project to select a HAT as the DAQ for the system. Most of the HATs had a similar number of analog input channels as the USB DAQs which meant not missing out on any functionality. Furthermore, the HATs would integrate with the Raspberry Pi which would not change the footprint. After considering the various options, the data acquisition unit selected was the Measurement Computing 118 HAT. This board provides the system with 8 analog inputs, can be expanded by adding more HATs on top to allow for up to 64 input channels.

4.2.3 Sensors

The sensors needed to satisfy the data collection requirements of the system were a three-axis accelerometer, a three-axis gyroscope, a three-axis magnetometer, an altimeter, a GPS unit, and pressure differential sensors. These sensors exist on the market in analog forms which would integrate with the Measurement Computing DAQ. However, for ease of integration for the Phase I prototype development, digital sensors were preferred. The digital sensors would provide the simpler programming option and would provide acceptable data for a first prototype.

The digital options found spanned a wide range of prices, types, and configurations. The sensors are listed in **Table 9.** The key factors which differentiated the sensors were the accuracy, precision, and durability. All of those factors are desired for the system. However, because the Phase I system is a functional prototype, it was decided that expensive sensors would not be necessary to prove the functionality of the system. By purchasing expensive sensors, the project would be expected to rely on those options for a long time regardless of if they end up being the correct solution. Thus, the Phase I prototype would be built with cheap sensors capable of providing the information set in the requirements.

The specific sensors selected were the Adafruit 9-DOF Breakout, Adafruit MPL3115A2, and SparkFun GPS Breakout - XA1110. There is not much special about the specific sensors other than they were cheap and available. The 9 degree-of-freedom inertial measurement sensor includes the accelerometer, gyroscope, and magnetometer required for the vehicle state estimation. The altimeter measures the pressure and calculates an altitude based on the measurement, outputting both values. The GPS board was the one exception where a more capable sensor was selected. The XA1110 is capable of interfacing with the Global Navigation

Satellite System (GNSS). This extra functionality means the sensor will be useful, without needing to be replaced longer than some of the alternatives.

Table 9: Components selected to interface with the Raspberry Pi.

Name	Interface	Power	Cost
9 Axis Accelerometer/Gyroscope/Magnetometer			
Sparkfun MPU-9250 IMU	Digital I/O	3.3 V	$15
Adafruit 9-DOF Breakout	Digital I/O	3.3V	$20
Adafruit 9-DOF Orientation IMU	Digital I/O	3.3V	$35
Yost 3 Space Sensor	USB	3.3V	$65
TinkerForge IMU Brick 2.0	USB	USB	$75
Yost Labs 9-DOF	USB	USB	$145
Zenshin Tech LPMS-USBAL2	USB	USB	$250
Barometer			
Adafruit MPL3115A2	Digital I/O	3.3V	$10
Adafruit BMP388 - Precision Barometer	Digital I/O	3.3V	$10
Dracal USB Bar-20	USB	USB	$50
Vernier BAR-BTA	USB	USB	$70
GPS/GNSS			
USB/TTL Raspberry Pi GPS Tracker	USB	USB	$30
Adafruit Ultimate GPS with USB	USB	USB	$40
Adafruit Ultimate GPS Logger Shield	Digital I/O	3.3V	$45
Adafruit Ultimate GPS HAT for Raspberry Pi	Digital I/O	3.3V	$45
SparkFun GPS Breakout - XA1110	Digital I/O	3.3V	$50

Two additional sensors were added to support the air speed measurement requirement. These are the Adafruit MPRLS pressure sensors which measure 0-25 psi. A multiplexer was also purchased to allow these two sensors to communicate on the same I^2C lines. In tandem, these sensors can perform a pitot-static differential pressure measurement which can be used for airspeed. Upon further analysis, a differential pressure sensor would have been a more apt solution and is worth considering replacing these two sensors.

The sensor selection was not critical to the performance of the system. Most important was that they collected data required by the system requirements and that they could interface with the system. Then, cost was considered, selecting components which would not inhibit the AFL from changing them out if the need arose in the future.

4.2.4 Communications

The final component of the system selected was the communication system. For initial development, the communication system was not required because it was more important to connect the sensor inputs to the processor. However, as development proceeded and the device moved toward flight readiness, the communication system needed to be addressed. The Raspberry Pi was capable of data communication through a variety of protocols. The built-in protocols are Bluetooth and wireless network connection. Then, the Ethernet and USB ports can act as other communication avenues.

For initial testing, two data transfer protocols were chosen for the communication system. First, communicate via the Bluetooth as a surrogate for file transfer over the wireless internet connection envisioned for the Phase III system. Second, communicate over a USB-connected radio for simplicity and reliability during line of sight testing. The Bluetooth communication was already available with the Raspberry Pi and the components for radio communication were available in the AFL. Some XBee Pro S1 radios from a previous project were selected as the radio link. The data sheet showed that these radios could reach ranges of up to one mile in rural areas, but transmission distance was reduced to about 40 feet in urban settings.

The Phase III vision for this system calls wireless data transmission via internet satellites because the remote GCS will not be in proximity of the vehicle. However, it would likely be in the best interest of the system to maintain the ability to communicate by different protocols. Thus, the Phase I prototype being developed with the radio communication should be a facet of the system through the phases. It provides a cost-free way to perform line-of-sight testing and does not change anything about the data collection or processing parts of the system. In this way, the modular capability of the system show that one part of the system can change without disrupting the other parts.

4.2.5 Power

The Raspberry Pi 4 requires a 5-volt, 3 ampere power system. During development the Raspberry Pi can be plugged directly into a wall outlet using an adapter, but during flight it must be powered by a battery system. For the prototype system, an Anker 20000 mAh battery pack was chosen because it provided the 15 watts of power and had a large capacity.

4.3 Phase I Prototype Build

The Phase I prototype was assembled using the parts chosen during component selection. A housing was designed to be 3D printed based on the sizing of the components and the system size requirement. The elements of the system were connected electrically according to their data sheets. The wiring diagram is shown in **Figure 7** and the assembled system in **Figure 8**.

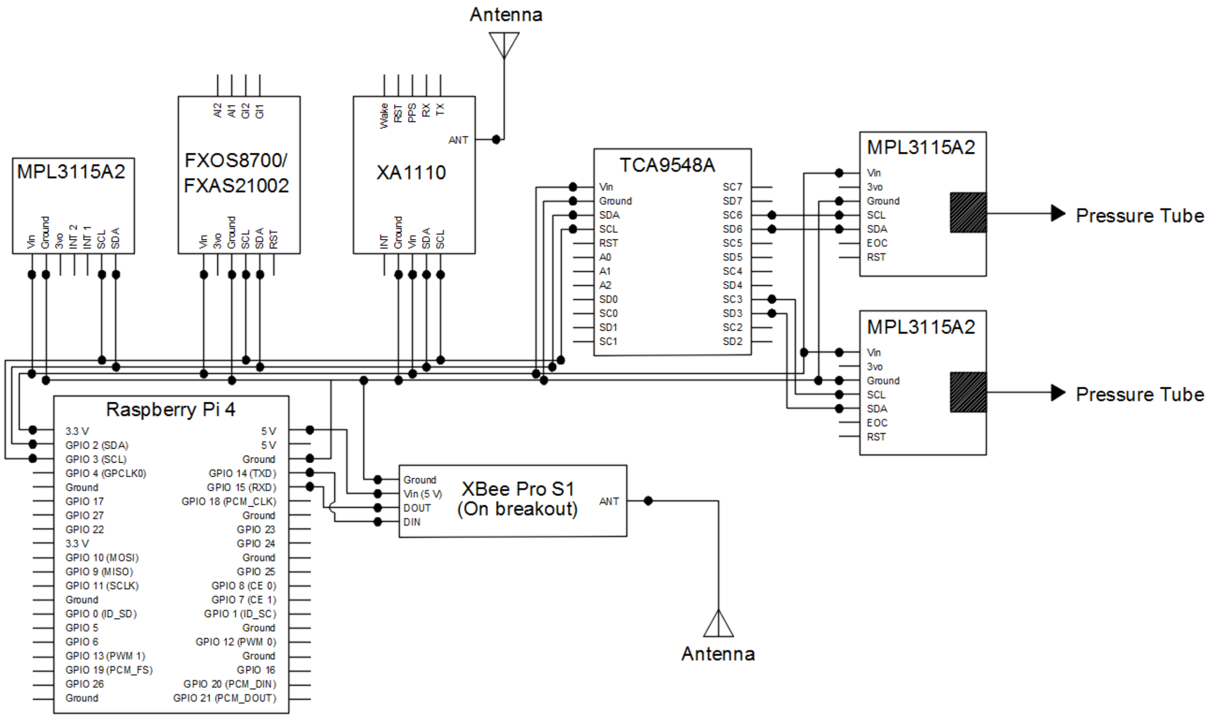

Figure 7: Wiring diagram of the system.

The system was programmed through the Raspberry Pi with Python and Bash scripts to poll the sensors for data and handle the system tasks. The combined functionality of these scripts allowed the system to collect data from the sensors, combine the data into a packet for transmission, and send the packet to the remote GCS. This functionality is the manifestation of the functional flow, activation, and control diagram.

Nova was the vehicle chosen for the first flight of the system because of the needs of the stakeholders. Thus, the first vehicle-specific integration occurred on the Nova. To account for the radio and GPS attenuation issues encountered with the Nova's fully carbon fiber body, a 3D printed plastic payload bay cover replaced the Nova's stock cover.

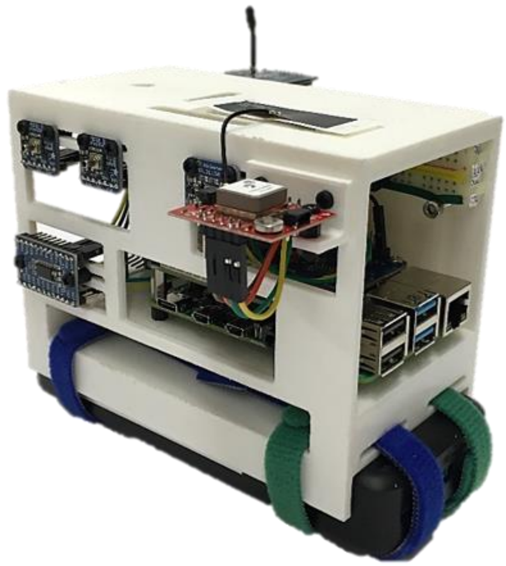

Figure 8: The packaged system for the AFL's Nova and Vapor 55 vehicles.

Another system modification required for the Nova was the addition of ballast weight to the system. One of the requirements for the system was to not move the center of gravity (CG) of the vehicle so that flight performance was not compromised by the inclusion of the system. A weighted clamping system was built to secure the additional weight onto the system in a way which would allow it to fit in the Nova and not move the CG position. The system in its Nova flight configuration is shown in **Figure 9**.

The packaged system with adjusted communication systems, added ballast weight, and functional software completed two test flights as the payload for the Nova sUAS on February 28, 2020. All systems were functional, and the flights proceeded as expected.

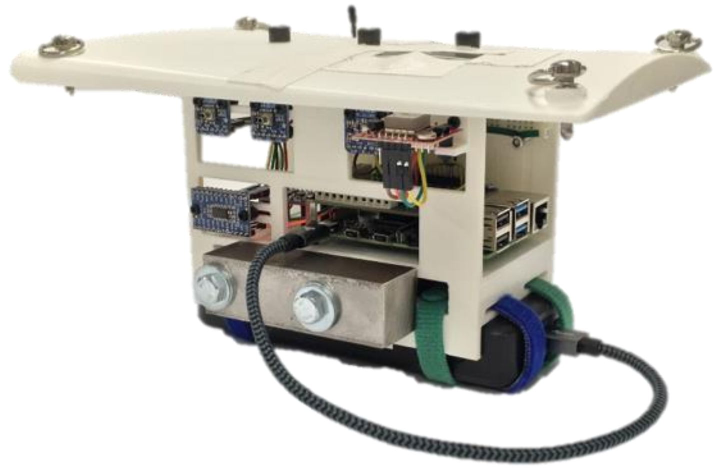

Figure 9: The packaged system modified to fly on for the AFL's Nova sUAS.

4.4 Summary of Phase I Prototype Development

The user needs from the system architecture were combined with specific information from the AFL Nova and Vapor 55 vehicles to develop design requirements. These requirements were logical and able to support the component selection for an initial Phase I prototype. The prototype proved the capability for the system architecture to act as a framework to design a system which meets requirements.

Further validation of the system architecture and Phase I prototype design were demonstrated with ground and flight tests. The performance of the system serves to validate the system architecture and demonstrate the capability of the Phase I prototype. However, the validation of specific components is unnecessary based on the scope of the thesis. Although it met many of the requirements, there is significant optimization from which the system could benefit during continued development. As these modifications are undertaken, the system should continue to be ground and flight tested and compared with the baseline results found in the **Results** section.

5. RESULTS

5.1 Testing

Various testing procedures were used to validate the design requirement. Components were tested as the prototype was assembled. The testing compounded as more components were added and the prototype became more capable. Once all the components had been assembled and tested individually, static testing as a full system was completed. Next, the system was ground tested to ensure robust functionality at distance. Finally, the prototype was loaded into the payload compartment of the Nova and tested in flight.

5.1.1 Component Testing

Component testing consisted of a simple functionality test for each element in the system. The component was connected with the Raspberry Pi as detailed in the data sheet and the script for utilizing the component was run to verify functionality. All the scripts were written separately to ensure the system remained modular. No results were gathered from this testing besides proving the functionality of each component prior to system integration.

5.1.2 Ground Testing

Once the system was fully integrated, it was tested as if it were in flight. These tests occurred in the lab or outside the lab on campus. The range of the XBee radios was limited on campus due to signal interference. Nonetheless, testing demonstrated the over capability of the system to perform in a flight-ready state while on the ground.

5.1.3 Flight Testing

The final test for the system was to install it on the Nova and fly at Cal Poly's Educational Flight Range (EFR). The test flight operation followed the operational concept from the system architecture development. While the vehicle was being prepared for flight, the system also underwent final communication checks. This process was repeated for two flight tests. Both tests followed the mission plan without any indication of an issue for either the vehicle or the system.

5.1.4 Testing Results

The condition of the data collected during testing was not critical to the completion of the Phase I prototype, besides acting to validate the system architecture and prototype design. However, a short treatment of initial results from the two flight tests will be presented to demonstrate the initial capabilities of the system.

For initial testing of the system, the most important data collected was the location of the vehicle in space. The GPS data was used to plot the location of the vehicle by latitude and longitude during the flight. An example of the data collected during one of the test flights is shown in **Figure 10.** The flight path indicates the data collected by the system is capable of showing a remote operator the location of the vehicle with spatial resolution.

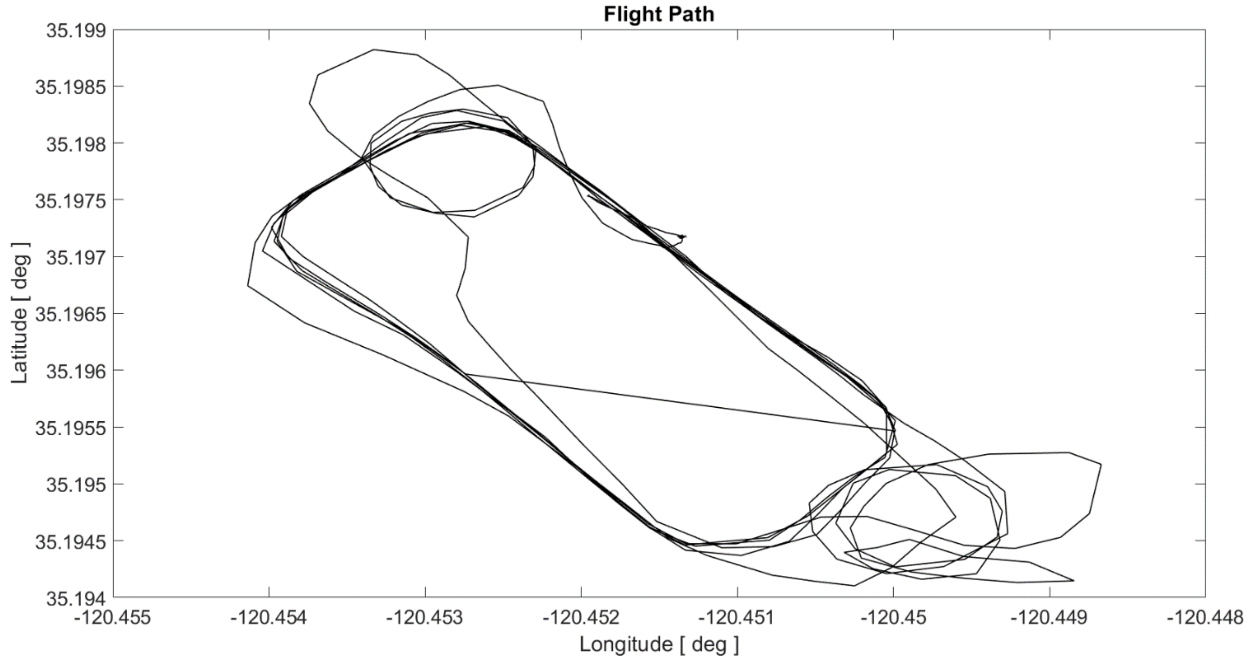

Figure 10: The path of the Nova vehicle during the second flight test.

An example of the acceleration plotted over time also shows the data as seen in **Figure 11** Even with the data rate which was not optimized for the flights, the accelerometer was able to characterize the launch and landing sequences at around 50 and 800 seconds. This proves that at below-optimal data rates of about 0.75 Hz acceleration events can be monitored using this system. With further improvement of the system, the characterization of these events can increase fidelity to become critical indicators of flight status for the remote operator.

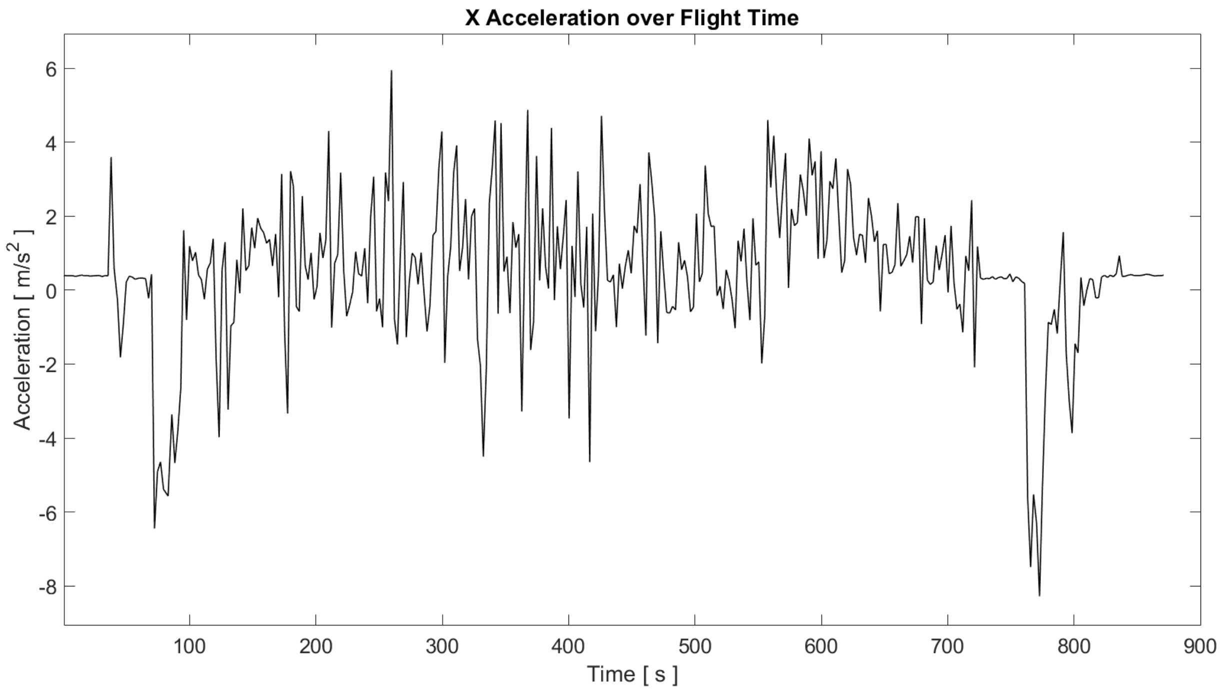

Figure 11: Acceleration in the thrust direction of the Nova during flight test.

Another important feature of the data collection system is its ability to measure the altitude of the vehicle. **Figure 12** shows the altitude data collected onboard the Nova over time. The targeted altitude for the flight was 360 feet above the takeoff location. The system averaged 375 feet during the target altitude phase of the flight indicating that this is a valid method for altitude measurement since the error was within five percent.

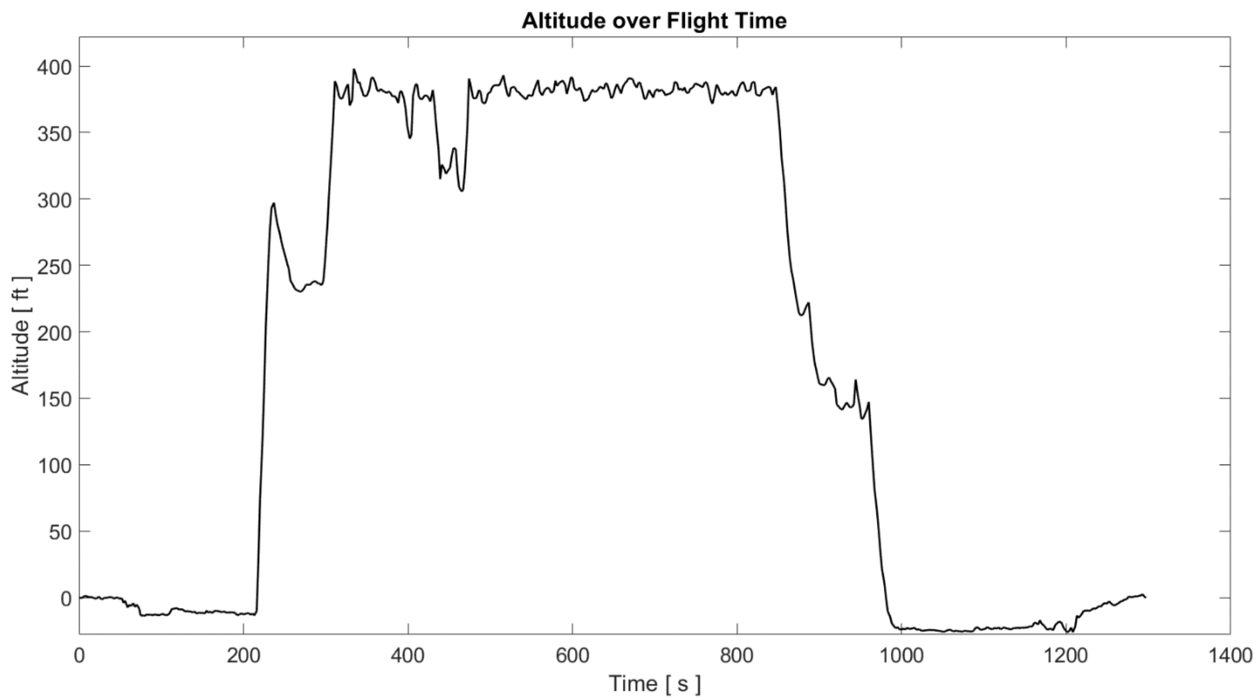

Figure 12: The altitude of the Nova vehicle during flight test.

The system demonstrated the ability to collect flight data which would be useful for a remote operator to understand the current state of the vehicle. However, the communication results of the flight test were not as strong. Only 53 percent of the flight data were received by the remote GCS over the two flight tests. The root cause of the communication issue is currently unknown.

The lost data packets were analyzed in terms of distance, altitude, and direction of travel as seen in **Figure 13**, **Figure 14**, and **Figure 15**. The three analysis methods did not identify a singular cause of failure for the communication system.

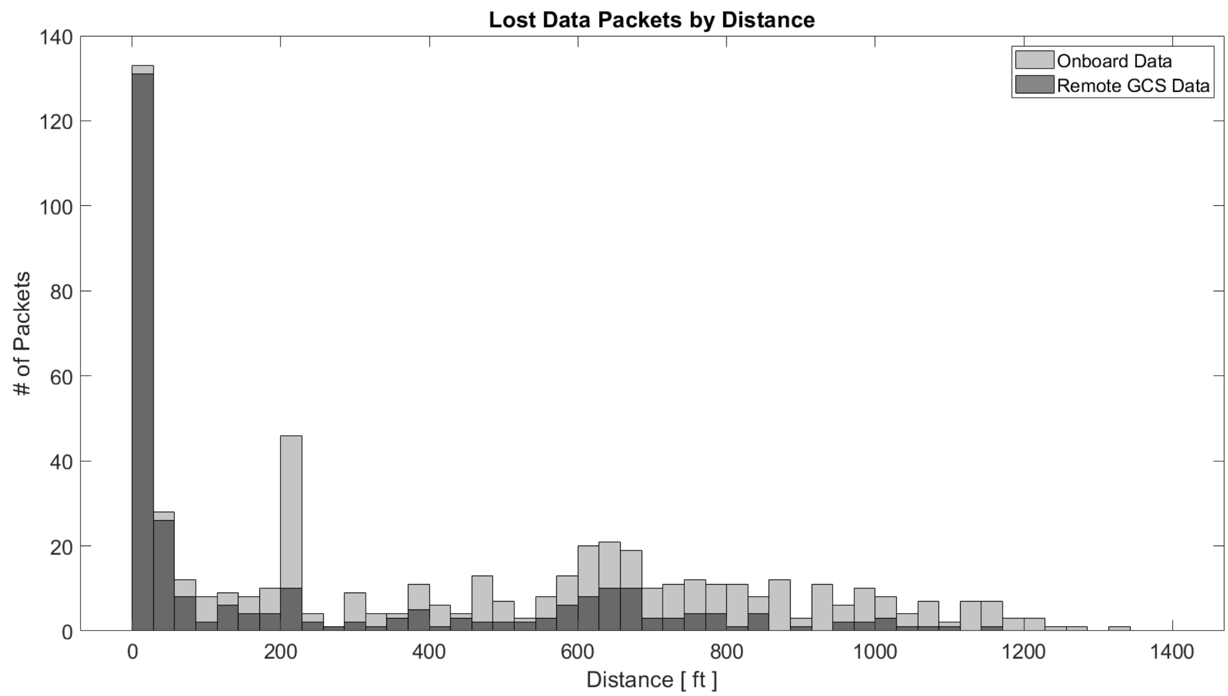

Figure 13: Data packets collected onboard and received at the GCS by distance.

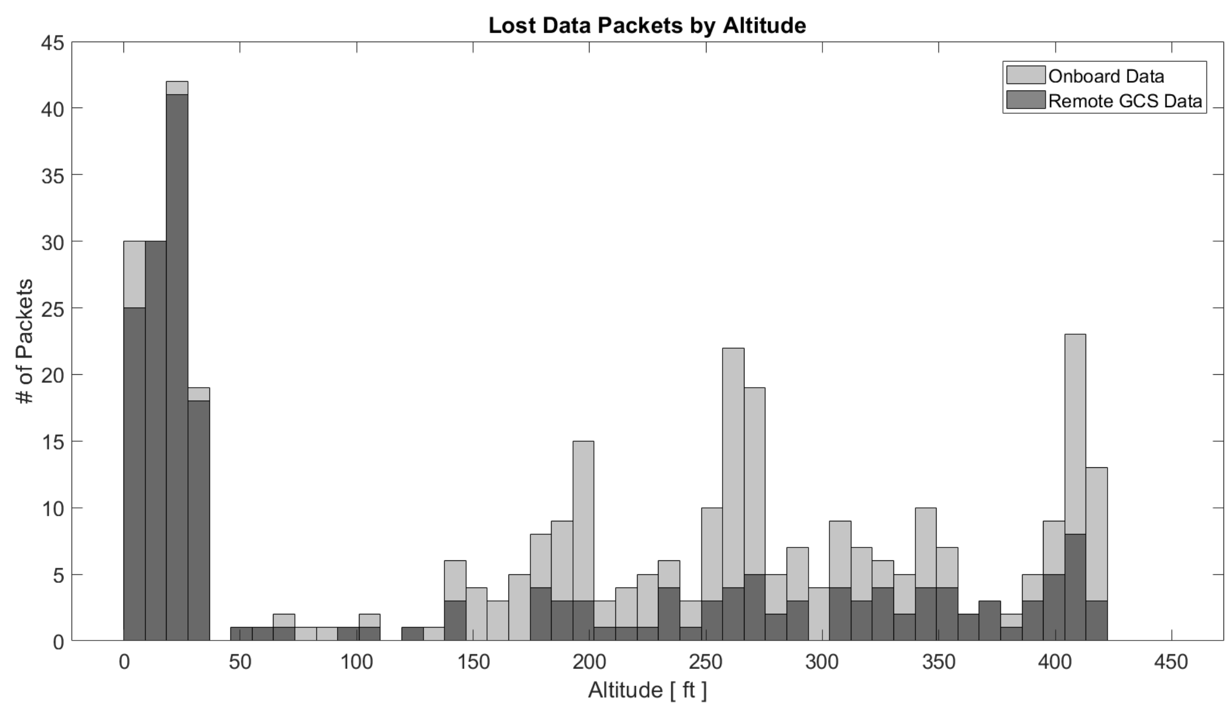

Figure 14: Data packets collected onboard and received at the GCS by altitude.

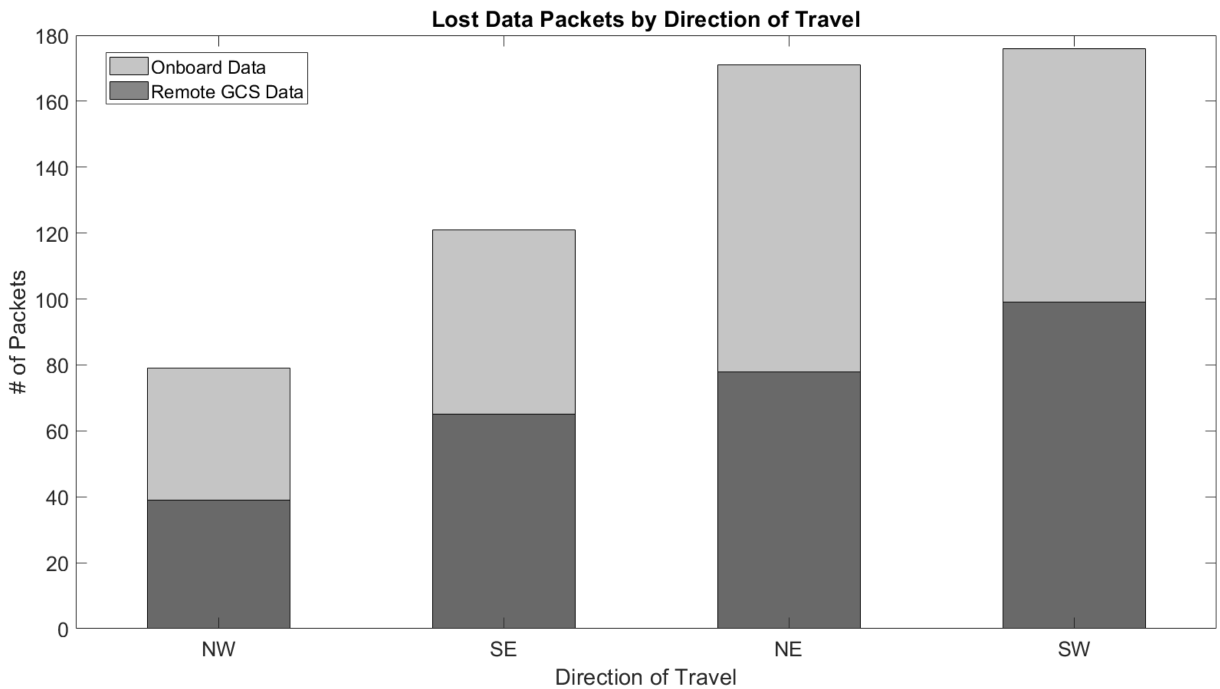

Figure 15: Data packets collected onboard and received at the GCS by distance.

The distance seemed like an obvious issue based on the poor range during ground testing. However, the histogram does not indicate that more data was lost at farther distances. Within 100 feet of the remote GCS, however, the system did receive 99 percent of the data packets collected. Beyond the 100 feet, the communication system consistently lost packets at all distances.

Similarly, the altitude and direction of travel data do not conclusively show that a certain height or direction cause the communication system issues, but rather any altitude above 50 feet and any direction of travel seemed to have data packets lost.

It is important to note that once the vehicle was flying all of the conditions were changing simultaneously. The distance, altitude, heading, and attitude of the vehicle are all inherently link on small time scales. Thus, the data does not identify a root cause for the communication issue.

The current understanding of the system and its interactions with external systems suggests that the issue is a combination of factors. Firstly, the Nova vehicle is carbon fiber which can cause issues with radio frequency communication systems. At distance, the signal strength may not have been strong enough to overcome the interaction with the Nova body. This signal blocking issue could have been compounded by the fact that the radio antenna was mounted at the top of the vehicle. When the vehicle was on the ground the two antennae were unobstructed, but once in the air that connection was obstructed by the Nova fuselage depending on its attitude. The direction of the antennae may have contributed to the data dropouts as well. When the Nova was flying directly overhead turning the remote GCS antenna on its side sometimes allowed for a connection to be reestablished. Further investigation of the system is required to determine the root cause of the communication issue.

A solution which may help identify a culprit are to relocate the antenna on the bottom of the vehicle for a direct line of communication with the remote GCS. Higher gain radio modules or larger antennae could also be used to improve the signal strength of the communication between the onboard system and remote GCS.

Alternatively, the data collected by the onboard system conclusively show the capability of the system. Since the final configuration of the Phase I prototype will not use radio communication for data transmission, it may be prudent to transition the communication system to a satellite-based system. In the system architecture framework, the transition to a satellite-based communication system should be easily implemented.

5.2 System Validation

The data from the flight test along with information gathered during component, static, and ground testing was used to validate the system architecture and prototype design. Validation of the system architecture was supported by both the ability for the system architecture framework to support feasible design concepts, and the ability of those design to be built and work as described. Validation of the prototype design relied solely on satisfying requirements with analysis, demonstration, inspection, and testing techniques using the prototype system.

The testing procedures for the Phase I prototype illustrated the system's ability to meet the requirements outlined during its design. Validation of the Phase I prototype provided confidence that the framework established by the system architecture will support the development of the Phase II and Phase III systems.

The subsequent sections of the Phase I prototype validation are broken into the same categories as the design requirements to indicate which requirements were met during testing and which need further attention. Each section consists of a table which matches the Phase I prototype validation, on the right, with the design requirement, on the left. If validation is complete, the technique describes how the requirement was satisfied. If the validation is not complete, the technique describes the how the requirement should be validated.

5.2.1 Functional Requirements

The functional requirements describe what the system shall do and are tabulated in **Table 10**. The qualitative and quantitative requirements are listed.

During static and ground testing, the system demonstrated the ability to sustain Bluetooth or radio data link for long periods of time. However, these tests were usually performed in close proximity. Ground testing was limited because radio interference significantly limits the capability of radios in urban settings. During flight testing it became clear that the radios from the AFL which were used because they were available could not reach the 3000-foot range required during the flight test. The software worked well, but the communication hardware needs improvement to satisfy the requirement.

The sensors onboard the Phase I prototype were not characterized or analyzed for accuracy. The data from these sensors seem to be correct, but the continuation of the project should look to characterize whatever sensors are install for long-term use. One system which was not used during the flight tests was the pitot-static airspeed measurement. Currently, two 0-25 psi total pressure sensors are installed as the sensors for making this differential measurement. However, it now understood that those sensors might work for the airspeed measurement but would likely have difficult resolving speeds because their full ranges reduce the small pressure resolution. This system can be tested for functionality. However, it seems likely that some variation of this system should be implemented.

The last requirement in the table is for Phase III communication with the vehicle. It was included because it was an important factor from the system architecture. The Raspberry Pi has numerous data interfaces which are likely to be capable of connecting with the Nova and Vapor 55 vehicles, but that communication will need to be tested during the Phase III development.

Table 10: Functional Requirements Validation.

Requirement		Validated	Technique	
1.	The signal between the remote GCS and the system shall not be lost for more than 8 seconds.	No	D:	A maximum lost connection of 110 seconds was measured during testing flight testing. This is a result of using an underpowered radio communication system.
2.	The system shall communicate via wireless internet connection, local network, or radio.	Yes	D:	The system successfully transferred files via Bluetooth and serial data via a radio connection. Serial data transfer and file transfer are the two types of data transfer protocols required for the three communication methods.
3.	A minimum 8 GB data storage device such as an SD card or flash memory chip shall be present to store data on board.	Yes	I:	A 32 GB micro SD Card was installed. About 15 GB are utilized by the operating system, leaving 17 GB of storage space.
4.	The system processor shall be capable of multitasking or being programmed as a task scheduler to simulate multitasking.	Yes	D:	Multiple scripts were run simultaneously using multiple terminal events.
5.	The system shall employ a check-sum algorithm for data transfer integrity verification.	Yes	T:	A check which verifies the right amount of data arrived from the flight data collection system was implemented on the remote GCS.
6.	The system shall have a battery life of at least three hours.	Yes	A:	The standard power draw of the Raspberry Pi 4 under load is about 7.6 W. The 20000 mAh battery from the AFL provides 13 hours of battery life.
7.	The logic system shall control the rate of data collection and communication.	Yes	D:	Data collection and transmission frequencies are controlled by a task managing script.
8.	The system communication bandwidth must be greater than 240 bytes/second.	Yes	D:	Initial testing demonstrates a transfer rate of about 640 bytes/second for both Bluetooth and radio protocols.
9.	The system shall interface with accelerometers, gyroscopes, and magnetometers for making inertial measurement.	Yes	D:	An Adafruit Precision NXP 9-DOF board was integrated with the system and data collection was verified.
10.	The system shall be interface with GPS and pitot-static air measurement systems to determine flight characteristics.	Yes	D:	A SparkFun XA1110 GPS board and two Adafruit MPRLS pressure sensors were integrated with the system. Data collection verification is still needed.
11.	The system shall communicate through an RS-232 port or 16-pin connector with the vehicle.	No	T:	For Phase III, the system shall communicate with the vehicle flight controller through a physical connection.

Legend: **A:** Analyze │ **D:** Demonstrate │ **I:** Inspect │ **T:** Test

5.2.2 Interface Requirements

The interface requirements describe how the system will interact with external systems and are tabulated in **Table 11**. The external systems include the physical interaction the prototype will have with the vehicles as well as the ground systems which need to interact with it for programming and data extraction.

Table 11: Interface Requirements Validation.

	Requirement	Validated	Technique	
1.	The system shall be smaller than 7.5 x 4.5 x 4 inches.	Yes	I:	The design for the system enclosure is 7 x 4 x 3 inches.
2.	The system data storage shall be accessed through a USB, SD card, or wireless transmission.	Yes	D:	The data on the system was accessed directly through the system like a computer. Data can be transferred by removeable USB, Bluetooth, or internet connection.
3.	The system shall be programmed or accessed through a wired, network, or Bluetooth connection.	Yes	D:	The system was programmed through the operating system. Scripts can also be loaded onto the system through USB, Bluetooth, or internet connection.

Legend: **A:** Analyze | **D:** Demonstrate | **I:** Inspect | **T:** Test

The ability for the system to be accessed by an operator for programming and data transfer was trivial because the system is a computer. The system can also be accessed by remote desktop if it is not connected with a monitor. The remote desktop connection was demonstrated during the flight tests.

The housing for the system allows it to fit into both the Nova and Vapor 55 vehicles. It can be modified as needed to satisfy more specific fit issues, such as requiring ballast weight when flying on the Nova.

5.2.3 Non-Functional Requirements

The non-functional requirements describe the overall characteristics of the system such as reliability, safety, and maintainability. These requirements are tabulated in **Table 12**. The non-functional requirements were primarily defined to keep the vehicles safe. There are a number of parameters the system could influence which might compromise the stability or control of the vehicles. Thus, these requirements were implemented to ensure the system does not endanger the vehicles.

Table 12: Non-Functional Requirements Validation.

Requirement		Validated	Technique	
1.	The system communication shall not interfere with the vehicle communication channels on 900 MHz, 902-928 MHz, and 2400-2485.3 GHz.	Yes	T:	The system utilized a radio which communicated on the 900-925 MHz frequency. However, the radio was sophisticated enough to frequency hop within that range to avoid interference with the flight system.
2.	The system shall not impact the vehicle stability by moving the center of gravity (CG) beyond the allowable limit.	Yes	D:	The system weight and size allow its installation into the Vapor 55 and Nova without moving the CG beyond the listed limits.
3.	The system shall weigh less than 2.5 pounds.	Yes	T:	The assembled weight of the system is 1.6 pounds. This includes a very large battery which could be reduced.
4.	The system shall survive nominal vibration loads from the Nova and Vapor.	No	A:	A vibrational analysis of the system shall be performed to confirm the system is not affected by normal vibrations. Two nominal flight tests did not appear to show any system issue with Nova flight vibrations.
5.	The system shall be robust to shock impacts with the ground up to 40 feet/s.	No	A:	An analysis of the components in the system shall be performed to confirm their resilience to the shock impact. Two nominal landings showed no issues arising from the impact of the vehicle upon touchdown.

Legend: **A:** Analyze | **D:** Demonstrate | **I:** Inspect | **T:** Test

Analysis of the system's resilience to shock impacts and flight vibration effects are important for ensuring the system is robust. However, these analysis projects are beyond the scope of the thesis proposed here. The demonstration of the prototype's resilience to these effects will be demonstrated with flight tests. All the components performed nominally, and there was no damage to any part of the system over two normal test events. Thus, the system's ability to survive the nominal flight conditions of the Nova has been verified, with Vapor 55 results pending a flight test using that platform.

The system by weighs 1.6 pounds. This allows it to fly on either the Nova or Vapor 55. For both vehicles, the weight and placement of the prototype will be important to not move the CG too far from the nominal condition. In order to keep the CG position within limits on the Nova, extra ballast weight was added to the system, so it weighed the same as the standard ballast payload.

The radio communication requirement is critical to allowing the prototype to fly on either vehicle. During initial ground testing and flight testing the current system did not interfere with the Nova flight control system despite working on the same frequencies. The radios were sophisticated enough to ensure they hopped on frequencies which did not already have data transfer on it. For the continuation of this project, the communication will not operate on the same radio channels, so this will not be an issue.

5.2.4 Enabling Requirements

The enabling requirements describe parts of the development which allow the system to succeed. These requirements are tabulated in **Table 13**. The enabling requirements focus on the ability for the prototype to be built and provide some guidelines set by the stakeholders.

Table 13: Enabling Requirements Validation.

Requirement	Validated	Technique
1. Components selected for the prototype design trade studies shall be available.	Yes	I: All the components selected for the Phase I prototype were available and arrived in a timely manner.
2. The prototype design shall be built using existing technology.	Yes	I: All the components selected for the Phase I prototype were off-the-shelf with no development needed.
3. The design portion of the project shall be within the budget set by the stakeholders.	Yes	I: No budget was officially set for the project, but the stakeholders approved all purchases.
4. The prototype design step shall be completed in the time frame allotted by the thesis project.	Yes	I: The proposed tasks for this project were completed to the satisfaction of the stakeholders.
5. The system shall utilize Viasat products if possible.	No	A: No Viasat products fit the needs of the system because they standard data rate was far below the system need.
6. The system shall utilize open source software and hardware.	Yes	I: The Raspberry Pi and its operating system are open source. The controlling scripts were written in Python and Bash languages.
7. The system shall be modular to allow for ease of modification.	Yes	D: The Phase I prototype was built from components with singular tasks, except for the Raspberry Pi. Thus, components can easily be swapped for modifications. Furthermore, the housing is built to accept many sensor form factors.

Legend: **A:** Analyze | **D:** Demonstrate | **I:** Inspect | **T:** Test

The enabling requirements were all validated except for the stakeholder suggestion to use Viasat components. There were no systems in Viasat's catalog which fit into the system architecture based on physical size and data bandwidth. However, all the other enabling requirements were met by using cheap off-the-shelf components and open source software to build a modular system.

5.2.5 Constraints

The constraints pertain to the deployment of the system. They are in **Table 14** and describe a critical trade-off for the system and performance validation needs.

Table 14: Constraint Requirements Validation.

Requirement	Validated	Technique
1. The system's communication reliability and fidelity shall be preferred to increased communication speed with worse data integrity.	Yes	D: The data transmission rate was reduced during initial testing to keep the system from interrupting itself and losing data packets.
2. The system hardware and software used shall report the data collection speeds.	Yes	D: The system is not keeping a record of data collection rates, but it was able to time task times. Currently, when data collection is run in series it takes ~1 second.
3. The remote GCS shall report the data transfer rate.	Yes	D: The remote GCS shall be programmed to output a data rate for the operator. The GCS outputs the most recent data in the terminal with a timestamp.
4. The system shall measure the data transfer rate with the vehicle.	No	D: The time from an operator's command at the remote GCS to action taken by the vehicle shall be known for the Phase III system.

Legend: **A:** Analyze | **D:** Demonstrate | **I:** Inspect | **T:** Test

The constraint to prioritize good data over fast data was satisfied during early development of the prototype. The communication system was too fast for the data collection system and crashed the system. This was fixed by modulating the communication system frequency. Further optimization of these systems is required as development proceeds.

The data collection rates have been demonstrated during the various tests. A future task will be to report these rates in real time to the remote operator. The final requirement to validate the data transfer rate between the system and the vehicle control system is based on the need for the operator to know the latency for commanding the vehicle. However, that requirement will not be met until Phase III.

5.3 Results Summary

The Phase I prototype was tested at the component level, statically as a full system, dynamically as a full system, and in full flight configuration. These tests validated much of the system architecture and many of the design requirements. A few requirements were not validated during the test campaign.

The most important requirement which was not met was the functional requirement to reduce loss of data connection for under 8 seconds. During one of the flight tests the data connection was lost for 110 seconds. The remedy is to perform future testing with an optimized direct radio link, or to continue development toward the complete operational configuration where the data connection occurs through satellite communication. The other requirements not met which are pertinent to the Phase I prototype are the vibration and shock analysis. During the two flight tests, the robustness of the system was demonstrated by surviving two nominal flights. However, a more complete analysis could be completed in the future to fully characterize vibrational loads.

A few other requirements were included with the requirements that were not intended to be completed until Phase III but were important to keep in mind during the development of the

Phase I prototype. The Raspberry Pi should allow for connection and communication with the Nova and Vapor 55 through serial, pulse-width modulated, or general-purpose outputs.

The overall result of the system architecture and design for the Nova and Vapor 55 vehicles was a success. The system architecture provided a framework which was successfully filled in with requirements for the Nova and Vapor 55 vehicles. Those requirements described many versions of a design which could provide the characteristics required in the system architecture. The chosen design was built using off-the-shelf components and programmed using open source coding techniques. The end result was tested from component level to flight. The results of the initial Phase I prototype validated the system architecture's ability to describe a system which performs the functions required. The Phase I prototype requirements which were not met were not a reflection of the system design, but the individual components. Due to the nature of its development, validating individual components was not the purpose of the prototype and, therefore, not of consequence.

6. CONCLUSION

sUAS are a valuable tool for many researchers, agriculturalists, and healthcare professionals. However, all these industries cite issues with the utilization of sUAS. An increase in operational efficiency could make sUAS indispensable tools for some of these applications. Currently, the operation of sUAS in these field are limited by FAA regulations. In turn, the regulations stifle the development of more capable vehicles and more advanced operational concepts. The project underway through the AFL is focused on obtaining the capability to analyze a future operational concept in the anticipation of modified FAA regulations. The AFL's project aims to build a prototype system which allows an operator in the lab control a vehicle flying remotely.

Phase I of the three-phase project has been successfully completed. The Phase I prototype provided a platform for validating the full system architecture. It demonstrated the system architecture was able to support many design variations, one of which was successfully built into the Phase I prototype system. System testing also showed the capabilities largely met the requirements set using the system architecture framework. This further validated the system architecture and the Phase I prototype as a success.

Although the Phase I prototype was successful, there were aspects of it which were not optimized. Further work should be dedicated to improving the performance and capability of the prototype. The initial elements of the system which van be address are the data communication system, the simultaneous task management for executing tasks onboard the system, and the live data representation on the remote GCS.

Flight testing the system with the remote GCS in the AFL would be a major step toward the operational concept presented in the introduction of this paper. When the Phase I system is capable of the long-range data communication reliably and with high fidelity data, the project will be ready to move on the Phase II and Phase III development.

The current capability of the Phase I prototype illustrated that the system architecture is defined appropriately and will also be able to support the Phase II and Phase III development. The continuation of the project in Phases II and III will expand the capability of the Phase I prototype to allow for payload and vehicle control. When those features of the project are addressed, the full potential of the system to be utilized as a test platform for the validation of a remote operator sUAS mission concept of the future will be realized.

The Phase III system will support the research and development of operational concepts at the university level. The AFL will be able to investigate the sUAS beyond line of sight operational concept for supporting applications such as wildlife tracking, agricultural management, and time-critical package delivery. As those applications are tested, the system can be refined and adjusted to increase the value of sUAS as mission-specific tools which end users can utilize for collecting valuable data and performing critical tasks.

REFERENCES

Ackerman, E., & Strickland, E. (2018). Medical delivery drones take flight in East Africa. *IEEE Spectrum*, *55*(1), 34-35.

Atkins, E. (2014, May). Autonomy as an enabler of economically-viable, beyond-line-of-sight, low-altitude UAS applications with acceptable risk. In *AUVSI unmanned Systems*.

Brusov, V., Grzybowski, J., & Petruchik, V. (2011). Flight Data Acquisition System for Small Unmanned Aerial Vehicles.

Buede, D. M., & Miller, W. D. (2016). *The engineering design of systems: models and methods*. John Wiley & Sons.

Christie, K. S., Gilbert, S. L., Brown, C. L., Hatfield, M., & Hanson, L. (2016). Unmanned aircraft systems in wildlife research: current and future applications of a transformative technology. *Frontiers in Ecology and the Environment*, *14*(5), 241-251.

Cummings, M. L., Bruni, S., Mercier, S., & Mitchell, P. J. (2007). *Automation architecture for single operator, multiple UAV command and control*. Massachusetts Inst Of Tech Cambridge.

Federal Aviation Administration. (2016). Summary of Small Unmanned Aircraft Rule (Part 107). FAA News.

Hayes, S. P. (2009). *Systems Architectures for a Tactical Naval Command and Control System*. Naval Postgraduate School Monterey CA Dept of Systems Engineering.

Hunt, E. R., Daughtry, C. S., Mirsky, S. B., & Hively, W. D. (2013, August). Remote sensing with unmanned aircraft systems for precision agriculture applications. In *2013 Second International Conference on Agro-Geoinformatics (Agro-Geoinformatics)* (pp. 131-134). IEEE.

IEEE Computer Society. Software Engineering Technical Committee. (1998). *IEEE guide for developing system requirements specifications*. IEEE.

Jiang, T., Geller, J., Ni, D., & Collura, J. (2016). Unmanned aircraft system traffic management: Concept of operation and system architecture. *International journal of transportation science and technology*, *5*(3), 123-135.

Linchant, J., Lisein, J., Semeki, J., Lejeune, P., & Vermeulen, C. (2015). Are unmanned aircraft systems (UAS s) the future of wildlife monitoring? A review of accomplishments and challenges. *Mammal Review*, *45*(4), 239-252.

Luxhøj, J. T. (2015). A socio-technical model for analyzing safety risk of unmanned aircraft systems (UAS): An application to precision agriculture. *Procedia Manufacturing*, *3*, 928-935.

Martin, L. (2011). Core System Requirements Specification (SyRS). US Department of Transportation, Research and Innovative Technology Administration, 131.

Rasmussen, J., Nielsen, J., Garcia-Ruiz, F., Christensen, S., & Streibig, J. C. (2013). Potential uses of small unmanned aircraft systems (UAS) in weed research. *Weed research*, *53*(4), 242-248.

Ruff, H. A., Calhoun, G. L., Draper, M. H., Fontejon, J. V., & Guilfoos, B. J. (2004). *Exploring automation issues in supervisory control of multiple UAVs*. Sytronics Inc. Dayton OH.

Scott, J., & Scott, C. (2017, January). Drone delivery models for healthcare. In *Proceedings of the 50th Hawaii international conference on system sciences*.

Stark, B., Rider, S., & Chen, Y. (2013). Optimal pest management by networked unmanned cropdusters in precision agriculture: A cyber-physical system approach. *IFAC Proceedings Volumes*, *46*(30), 296-302.

Taha, Z., Tang, Y. R., & Yap, K. C. (2011). Development of an onboard system for flight data collection of a small-scale UAV helicopter. *Mechatronics*, *21*(1), 132-144.

Trujillo, A. C., Fan, H., Cross, C. D., Hempley, L. E., Cichella, V., Puig-Navarro, J., & Mehdi, S. B. (2015). Operator informational needs for multiple autonomous small vehicles. *Procedia Manufacturing*, *3*, 936-943.

Vincenzi, D. A., Terwilliger, B. A., & Ison, D. C. (2015). Unmanned aerial system (UAS) human-machine interfaces: new paradigms in command and control. *Procedia Manufacturing*, *3*, 920-927.