A COHESIVE SIMULATION AND TESTING PLATFORM FOR CIVIL AUTONOMOUS AERIAL SENSING AND OPERATIONS

by

Stockton G. Slack

A thesis submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Electrical Engineering

Approved:

Calvin Coopmans, Ph.D. Major Professor	Jacob Gunther, Ph.D. Committee Member
Greg Droge, Ph.D. Committee Member	D. Richard Cutler, Ph.D. Vice Provost of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2023

ABSTRACT

A Cohesive Simulation and Testing Platform for Civil Autonomous Aerial Sensing and Operations

by

Stockton G. Slack, Master of Science
Utah State University, 2023

Major Professor: Calvin Coopmans, Ph.D.
Department: Electrical and Computer Engineering

Flying a small uncrewed aerial system (sUAS) impacts the safety of the airspace and there are many challenges and risks associated with collecting scientific data using an sUAS. This work demonstrates a cohesive simulation and testing platform, which allows for faster and safer testing and development of scientific data collection systems.

The cohesive simulation platform described in this work focuses on re-creating the aerial scientific data collection environment in a full software Gazebo-Classic simulation, and provides a means by which aerial scientific data collection systems can be tested on actual hardware. This testing is demonstrated using a smaller sized aircraft and a thermal sensing payload that is developed in this work.

This payload hardware testing provides a crucial link between the software simulation and testing on larger sUAS. Finally, test flights using the simulation and testing platform and the results from the test flights are presented.

(97 pages)

PUBLIC ABSTRACT

A Cohesive Simulation and Testing Platform for Civil Autonomous Aerial Sensing and Operations

Stockton G. Slack

Drones (also known as sUAS or small Uncrewed Aerial Systems) are often flown with cameras to take images of an area of land. These images can then be used to create a map by stitching these images together. This map can then be analyzed using scientific principles to learn things about the land and make decisions or take action based on the information.

The scientific application of drones is very advantageous, but flying a drone is inherently dangerous, impacting the safety of the airspace (particularly in the event of a crash), and drones are more dangerous the bigger they are. Smaller off-the-shelf drones are readily available to the public and are quite safe and easy to use. Larger near 55-lb fixed-wing mapping drones that can fly for 2.5 hours are quite costly and bring new risks into the equation. There are many barriers and risks to being able to successfully test equipment and to improving drone mapping technology.

This research focuses on creating a simulator that can simulate the entire process of creating these scientific maps. Simulating a drone, a camera payload, and a world for the drone to fly over. By having a simulator, researchers will be able to test out new technologies without having to risk flying a drone or without having to overcome the challenges mentioned above.

This research also focuses on creating a smaller simple camera payload that can be attached to a drone for performing test flights. This allows researchers to do scientific tests without risking flying larger systems.

This work enables the testing of sUAS payload systems many times in the simulation and then, when the system works as it should, the test flights with an actual drone can commence. This reduces the amount of time it takes to develop scientific drone systems and reduces the risk of flight.

To my supportive and patient wife, Cassie Dee Slack and our children.

ACKNOWLEDGMENTS

I'd like to thank Dr. Cal Coopmans, for supporting me and encouraging me throughout this research. He has taught me how to think outside of the box and, more importantly, to think about the box itself. From him I have learned how to look at the world from the perspective of a researcher. I'd also like to thank the other students working at AggieAir, as this project is not something that one individual could fully accomplish on their own, but has been the result of the combined efforts of many. I'll specifically note Richard Snider, Ezri Brimhall, Ethan Payne, Nathan Schwemmer, and Austin Washke, others not noted have contributed to the success of this research as well.

The past and current researchers that have been and are a part of the AggieAir lab deserve thanks for their contributions. Truly this research work would not be possible without those that have contributed past research and ideas, as well as those that are currently contributing to existing projects.

Funding for this research was provided by the USU Engineering Undergraduate Research Program, the National Institute of Standards and Technology (NIST), and the Utah Mineral Lease Funds.

I'd also like to thank my wife Cassie Dee Slack. I could not have accomplished this research without her patience and support. She believed in me many times when I didn't believe in myself and has been willing to support me through the stresses and anxieties, in addition to the joys and wonders, that have come with school and developing this research. I'd also like to thank my parents, Jeff and Jennifer Slack, for their example to me of diligence and hard work. They have always taught me that obtaining an education is one of the best investments one can make because it is an investment in oneself. I have found that to be very true.

Stockton G. Slack

CONTENTS

LIST OF FIGURES

Figure Page

ACRONYMS

CAASO	Civil Aerial Autonomous Sensing and Operations
sUAS	Small Uncrewed Aerial System
FAA	Federal Aviation Administration
GCS	Ground Control Station
GMS	Geospatial Mapping Simulation or Geospatial Mapping Simulator
SITL	Software in the Loop
HITL	Hardware in the Loop
STARDOS	Scientific Timely Actionable Robotic Data Operating System
SDF	Simulation Description Format
ROS 2	Robot Operating System version 2
UWRL	Utah Water Research Laboratory

CHAPTER 1
INTRODUCTION

The objective of this research is to implement a cohesive simulation and testing platform which allows for safer and more efficient development/research of civil small uncrewed aerial system(s) (sUAS or drone) data collection systems.

Data collection and flight with an sUAS are challenging [2] [3] [4] [5]. Because of the challenges associated with drone flight not all drone flights end with a safe and successful landing. Flying a drone in and of itself is inherently risky. In the event of a crash, drone flight could result in the drone being lost or damaged itself, the payload being damaged, the payload data being lost, or the harm of an operator or spectator. Berthe et al. noted that, "regulatory institutions at the global scale have performed a significant work to classify UAS regarding their lethal potential in a case of an impact with people on the ground" [6]. Flying an sUAS, while useful for collecting scientific quality data, inherently impacts the safety of the airspace.

Unfortunately, these risks will continue to be a part of drone flight, independent of what regulatory rules are put in place, there will always exist some amount of risk when flying a drone. It is important to note that when using an experimental drone or drone payload, the risks of drone flight can be greater due to unknowns associated with flying with a system that has not yet been proven in field tests.

This research seeks to improve the process by which the development of and research on civil sUAS data collection systems is carried out and to reduce the risk taken while performing tests on developmental systems. This research provides natural stepping stones that can be used to incrementally test and prove scientific data collection systems, with an emphasis on reducing the number of test flights that need to be flown and increasing the number of requirements that can be tested in simulation.

If the risks of scientific data collection using a drone are too high, then the valuable scientific quality data, that is gained when using an sUAS as the data collection vehicle, cannot be collected. Therefore, this work presents a cohesive simulation and testing platform for civil autonomous aerial sensing and operations (CAASO).

In this work: extensive background is given on AggieAir's (introduced in section 1.1) Scientific, Timely, Actionable, Robotic, Data Operating System (STARDOS) architecture in chapter 2; a literature review is presented on the works related to this research in chapter 3; the design and implementation of a novel cohesive simulation platform are presented in chapter 4, as well as the results from a simulated flight; and a novel cohesive testing platform are presented in chapter 5, as well as a description of the development of an sUAS thermal sensing payload and the results of an example data collection flight with this payload.

1.1 AggieAir

The AggieAir Lab at Utah State University has worked in small uncrewed aerial systems research for almost 15 years [7] [8] [9]. AggieAir has experience building, designing, and flying their own sUAS aircraft and payloads, having flown missions for well-known names like NASA and Northrop Grumman. AggieAir's research emphasis is on developing and implementing field ready drones and drone payloads for science [10].

While drones themselves are an area of research and development [11] [12] [13], AggieAir's main emphasis is on the type of data collection that can most easily be done through the use of flying robotic systems [14]. The data and information that can be collected by using a flying aircraft as the vehicle for a multi-spectral payload is unique, providing new insights and actionable data about the terrain that is flown over [15]. AggieAir is under the Utah Water Research Laboratory and focuses on collecting data consisting of "multi-spectral images collected for the purposes of scientific decision making in micro-agriculture, natural resources management, civil infrastructure inspection, etc." [16]

Using an sUAS to collect data and create geospatial maps (defined in 1.3) can provide new scientific insights. AggieAir has collected scientific quality data using sUAS and it has been used to estimate land surface temperature [17], to estimate chlorophyll for use in precision agriculture [18], and to perform an assessment of surface soil moisture [19]. Having this data can improve the ability of farmers and agriculturalists to make better decisions that will help their crop health and growth.

1.2 Objective and Novel Contributions

The objective of this research was in line with one of AggieAir's objectives, which is to improve the process of (or ability to) perform scientific data collection using sUAS. The STARDOS architecture focuses on advancing this objective. A explanation of the STARDOS architecture is provided in chapter 2. The STARDOS architecture strives to simplify and modularize the process of data collection for those that use sUAS to collect scientific quality data. It provides the ability to configure the payload, the ability to monitor the status of the payload, and the ability to process data in real-time.

While the research in this work is not explicitly focused on the development of STARDOS itself, the simulation and testing platform presented in this research is used extensively for the development and testing of STARDOS. This simulation platform is focused on providing the ability to test and improve scientific data collection systems. The simulation and testing platform presented in this work has been used as means of demonstrating the ability of STARDOS and as such is a critical part of the STARDOS architecture. More details and a demonstration of how the simulation platform is used as a part of STARDOS are described in section 4.3.3.

Additionally, throughout this work the AggieAir lab is referred to as contributing to the research work, implementing parts of the project, or making decisions. This is done because the credit for this research work cannot be fully given to a single individual. Many have contributed to the overall success of this work and research continues in many of these areas and on many of these projects at the AggieAir lab. When AggieAir is referred to in this work this is because others have contributed portions of work that have allowed the success of this research work.

The novel contributions of this work include: A cohesive simulation (see chapter 4) and testing (see chapter 5) platform for CAASO.

The cohesive simulation platform and cohesive testing platform provide natural stepping stones for scientific quality data collection development. These testing steps prepare the scientific data collection system for a fully successful flight using a larger platform, such as AggieAir's GreatBlue [16].

1.3 The need for a Cohesive Simulation and Testing Platform

Geospatial mapping is a visualization technique that shows how a map and its features fit into a specific geographic context. This context puts the map and its features into a known coordinate system. A typical example of this in sUAS mapping is when a drone is used to capture and collect imagery of a large area. The images collected during the flight are stitched together, using some form of image processing software, into a map. A map that is created from stitched imagery is often referred to as an orthomosaic and the software that does this stitching is often called photogrammetry software. Using GPS, and other sensor data collected during the sUAS flight, the orthomosaic is associated with a coordinate system creating a geospatial orthomosaic, or a geospatial map. The geospatial map can now be placed into other tools that utilize the coordinate system (e.g. ArcGIS, Google Maps) so that the orthomosaic map data can be used to make decisions, perform analysis, or enhance a larger map by providing higher quality data in a certain area.

Geospatial mapping is often performed using an sUAS. A wide variety of physical and technical challenges need to be overcome in order for a safe and successful data collection flight of a scientific sUAS. A few of these challenges include:

- Acquiring proper airspace access from the Federal Aviation Administration (FAA)
- Preparing the sUAS to ensure that it is flightworthy
- Managing the payload system to ensure that it will collect accurate data

These challenges are enhanced when using a larger and more expensive sUAS system. More care needs to be taken to ensure the aircraft is safe and flightworthy. Typically when using a larger sUAS the payload is more complex and requires more time to manage and prepare for flight. These and other challenges (such as preparing and training crew members, moving required equipment to the data collection site, and ensuring that the weather is appropriate for a successful data collection flight) need to be overcome in order to perform successful and safe test and operational field flights using a scientific sUAS.

When an sUAS simulator is used, the challenges of airspace access, aircraft flightworthiness, and payload readiness are reduced or completely eliminated. There is full airspace access, the aircraft is always flightworthy and ready to fly, and a simulated payload allows the user to isolate the specific variables that should be fixed for testing purposes.

Test flights, while more challenging, are still a necessary part of testing and improving a data collection system. If a smaller sUAS is used for testing and improving data collection systems the aircraft is simpler and takes less time and fewer preparations in order to be flightworthy. Likewise, a smaller simpler testing payload can be easier to prepare for flight. For these reasons this research focuses on implementing cohesive simulation and testing platforms. The following sections (section 1.3.1 and section 1.3.2) provide motivation for the research work and development of this platform.

1.3.1 A Cohesive sUAS Mapping Simulation Platform

While many sUAS simulators exist (see section 3.2) there are no sUAS simulators focused solely on simulating the data collection process required in CAASO. This work presents a drone simulator that can mimic the flight and data collection process. This kind of simulator enables safer and more time efficient research and development of systems associated with geospatial mapping. The restrictions of airspace access, aircraft flightworthiness, and payload readiness are removed when a simulator is used. This simulation capability is therefore known as a cohesive simulation platform for civil autonomous aerial sensing and operations.

As AggieAir has been developing the STARDOS architecture, with the objective of improving the process of scientific data collection, AggieAir has found the need for simulation and test platforms. The ability to test payload software on the ground, in simulation and in the lab on hardware payloads, is a crucial step of flight preparation. Simulating and ground testing provides greater confidence that the system will work as designed while in the air. In the development of the STARDOS architecture, if all of the system components needed to be tested on a flying aircraft the development cycle would be extremely slow. The development of a cohesive simulation platform has proved invaluable, providing AggieAir the flexibility to test the payload system extensively in the lab before taking it out for a real test flight.

The STARDOS architecture focuses on making uncrewed aerial system payloads simple and easy to use by pilots, field scientists, and other sUAS operators. A simulation environment focused on the scientific data collection not only aids in the development process of STARDOS itself. It provides a practice and training environment for the users of STARDOS.

Additionally, having an environment that more accurately simulates the geospatial mapping data collection experience allows for more efficient development of real-time processing algorithms, more test flights to be performed, more analysis of varying test flight patterns, and more safety. This increased safety comes because actual test flights can be done only when deemed necessary and when all other possible testing has been done in the simulation environment. A cohesive simulation platform not only encourages the development of systems like STARDOS but provides a platform where they can be tested and used.

1.3.2 A Cohesive Testing Platform

While simulation environments are extremely useful and allow for many problems in a system's software to be resolved before an actual sUAS flight takes place, there is no true replacement for running a system on actual hardware. Testing and debugging a system that has been implemented on hardware brings its own challenges to the table. While a simulation environment may strive to come as close as possible to re-creating the system that is to be used there are always differences. Additionally, just because something works in the simulation environment doesn't mean that when implemented in hardware that the system will react and respond the same.

We see this type of testing being applied by NASA in the Artemis program, as the first flight (Artemis I) "consist(ed) of an uncrewed flight around the Moon" [20]. Although an end goal of the Artemis program is to land more humans on the moon, they knew that it was necessary to first demonstrate the full functionality of their systems in hardware before taking greater and higher risks with human lives involved.

As mentioned previously, AggieAir's motivation to develop a testing platform has been enhanced by the development of the STARDOS architecture. The development of a cohesive testing platform has proven to be a crucial step to be able to take a system that has been tested in software and transitioned to a hardware setup. AggieAir's main sUAS platform is the 55-lb GreatBlue system introduced in "A 55-pound Vertical-Takeoff-and-Landing Fixed-Wing sUAS for Science: Systems, Payload, Safety Authorization, and High-Altitude Flight Performance" [16]. This paper describes in detail the advantages of having a large 55-lb system. While it can be advantageous for scientific quality data collection to fly a large aircraft, having to fly such a large system can be unfavorable for testing new developments in the data collection system; having to fly this large of an aircraft can be unwieldy.

There is a lot of logistical effort related to getting a larger aircraft ready to be fielded (e.g. the availability of trained pilots, the transportation of the aircraft and supporting systems, the amount of space required to fly a larger aircraft.) If these challenges need to be overcome in order to test on hardware, it is costly and time consuming to test out new scientific data collection software.

For this reason, in the same paper, AggieAir introduced the MiniBlue platform. The MiniBlue platform is a slightly smaller and scaled down version of AggieAir's GreatBlue. The MiniBlue platform "is a full-system trainer for GreatBlue. Wholly similar in some aspects (a transition VTOL aircraft with two power supplies and a combined payload computer/safety copilot)" [16]. Specifications for the MiniBlue platform can be found in the aforementioned publication. MiniBlue provides an opportunity for pilots to be trained and prepared to fly the larger 55-lb. GreatBlue aircraft, providing a similar enough flight experience. Fig. 5.3 shows an implementation of the MiniBlue platform.

Having a smaller system that takes less logistical effort to fly, that is focused on testing and training pilots provided a motivation to have a hardware testing payload platform. In this work a simple, affordable, thermal mapping payload solution is presented. This payload fills this need for the ability to test, on hardware, a scientific geospatial mapping system (such as AggieAir's STARDOS architecture). This payload, when combined with AggieAir's MiniBlue platform comprise the majority of the cohesive testing platform for civil autonomous aerial sensing and operations.

CHAPTER 2
STARDOS SUMMARY

With years of experience in collecting scientific quality data, AggieAir has identified three specific abilities that are desirable for an sUAS payload and ground station system that would aid in the process of collecting and analyzing scientific quality data.

1. The ability to configure and modify the settings and internal processes of a payload
2. The ability to monitor the status of the payload throughout an sUAS flight
3. The ability to process data in real-time

As the means of scientific data collection advance, there are a variety of different sensors used to collect scientific data [21] [22] [23]. Drones are used in a wide variety of situations and each mission may require a different payload setup. A modular payload provides the ability to interchange various cameras and sensors within a payload setup. For a payload to be fully modular, it also needs to provide software compatibility and configurability. The ability to configure, using software, the desired sensors and processing setup allows the user to tailor the data that will be collected to the mission itself.

When an sUAS carries a payload, it flies far enough away from the user that it is typically impossible to tell if the payload system is performing as intended without some form of communication link. If there is an issue (software or hardware related) with the data collection, the user will have no knowledge about the issue until the mission has been flown and they are able to physically access the payload again. At that point, it may be too late to address any issues and perform the mission flight again. Being able to monitor the current status of the payload allows the user to react immediately to any errors or issues that are detected. This is only possible if there is a reliable datalink connection that provides a payload status to the ground and a way for the user to view this payload status. In the event of an error, the user could make a remote adjustment while the aircraft remains in the air or bring the payload down to the ground to make a fix and then take off again quickly to complete the mission. Knowing the payload status throughout the duration of an sUAS flight saves users time and effort. Additionally, if the flight is flown and the data is not captured the mission is unnecessarily endangering the airspace.

Furthermore, there exists data that can be collected using a drone that may be time sensitive. As noted in section 3.3, first responders use thermal cameras for finding missing persons or identifying hot spots in disasters [24]. UAS with thermal cameras and data collected by them are used to study fire behavior and determine fire rate of spread [25] [26]. Non-visible hazardous materials, such as leaking gases, are detected using thermal cameras in the oil and gas sector [27]. These types of cases require that the data be processed and the correct information be delivered in a timely manner. If the data is delivered too late, then it becomes irrelevant or unactionable. These types of applications cannot rely on data or information being delivered at the end of an sUAS flight. This shows the need for the ability to process data in real time, as it is being collected on the drone.

With these requirements in mind, AggieAir has recently been developing and implementing the Scientific Timely Actionable Robotic Data Operating System software architecture, [28] (known as STARDOS). The goal of this architecture is to simplify and modularize the use of sUAS payloads by pilots, field scientists, and other sUAS operators. Providing the ability to configure the payload, monitor the status of the payload, and perform real-time processing onboard the payload.

The following sections present background information on the STARDOS user interface (StarCommand) in section 2.1, the STARDOS Datalink in section 2.2, and the STARDOS Backend in section 2.3.

2.1 StarCommand User Interface

sUAS pilots, field scientists, and other sUAS operators may not have a deep technical background into how a scientific sUAS payload functions. Their main objective is to collect the data, analyze them, and take or suggest some sort of action based on the results. It is crucial that sUAS operators be able to operate and monitor the sUAS payload functionality without knowing in depth how all parts of the payload work. STARDOS abstracts away the underlying technical layers providing an intuitive user interface called StarCommand.

The StarCommand user interface fills two main roles: it allows for payload configuration, and it allows the user to monitor the status of the payload before, during, and after the mission has been completed.

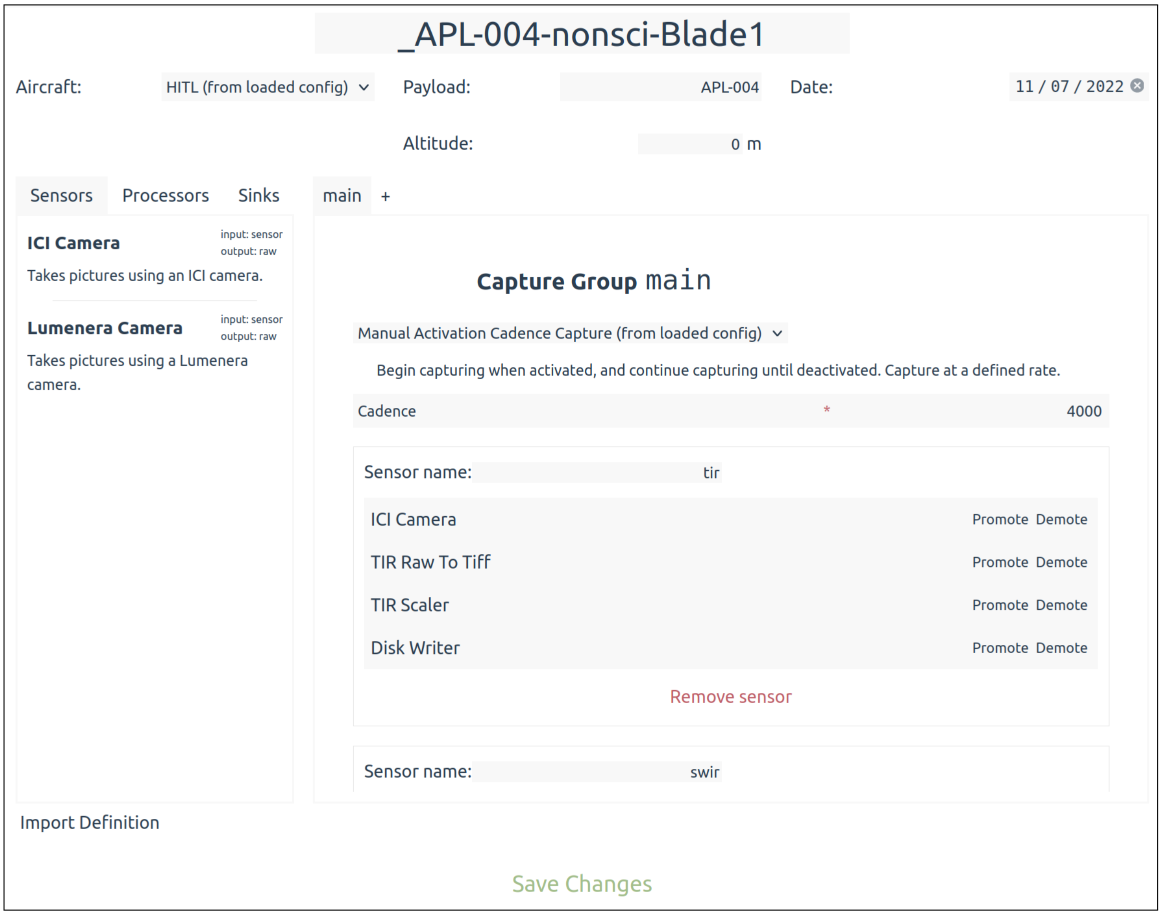
Payload configuration is done through StarCommand's configuration management tool. The configuration management tool is a web browser-based tool that allows the user to select which sensors will be used, the frequency that the sensors will capture data, and in what way the data will be processed or stored. A screenshot of the configuration management tool is shown in Fig. 2.1.

The other main functionality of StarCommand is payload status monitoring (and necessarily, this must be changed dynamically as the payload configurations change). It is crucial to be able to know the status of a payload in flight. If a mission is flown with a failure in the payload, this could result in corrupted data, incomplete data, and more than likely the necessity to re-fly the mission. Additionally, each flight lowers the safety of the airspace and increases the risk of not achieving mission success. Having the ability to monitor the payload status allows the ground crew to make a decision to continue the flight, fix the error in flight, or safe flight dictates they make an unplanned landing according to the information provided by the payload.

The StarCommand payload status monitoring tool displays information about the payload computer status and the status of individual nodes:

Table 2.1: The node and computer status information that is displayed in the StarCommand user interface

Payload Computer Status	**Node Statuses**
CPU Usage	Online
RAM Usage	Running
Swap Usage	Standby
Disk Usage	Initializing
	Warning
	Error
	Offline



_APL-004-nonsci-Blade1

Aircraft: HITL (from loaded config)

Payload: APL-004

Date: 11 / 07 / 2022

Altitude: 0 m

ICI Camera input: sensor output: raw

Takes pictures using an ICI camera.

Lumenera Camera input: sensor output: raw

Takes pictures using a Lumenera camera.

Capture Group main

Manual Activation Cadence Capture (from loaded config)

Begin capturing when activated, and continue capturing until deactivated. Capture at a defined rate.

Cadence * 4000

Sensor name: tir

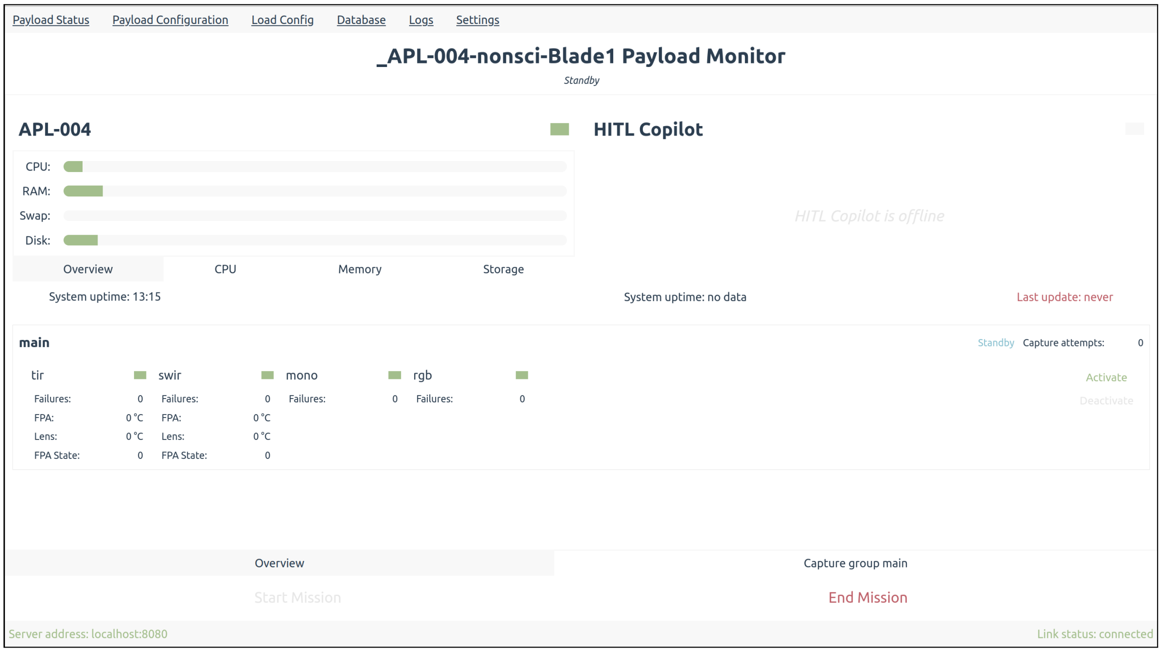
- ICI Camera
- TIR Raw To Tiff
- TIR Scaler
- Disk Writer

Sensor name: swir

Fig. 2.1: An example of the StarCommand Configuration Management Tool. Information about the aircraft and payload is configured in the upper portion, and information about the Capture groups, Sensor nodes, Processing nodes, and sink nodes is configured in the main center pane (these nodes are described in more detail in section 2.3). Users may select which nodes they want to use in the pane on the left.

The same information available about the payload computer is also available for the copilot (or companion) computer. Previous work describes the importance and use of a copilot computer [29] [30], additionally the PX4 Autopilot documentation describes it's use and setup [31]. An example of the StarCommand payload status monitor is shown in Fig. 2.2a and in Fig. 2.2b.

The StarCommand user interface can be accessed from any computer running a StarCommand instance. The data from and connection to the aircraft are provided by the STARDOS Datalink, as described in section 2.2, or this information can be passed through



Payload Status   Payload Configuration   Load Config   Database   Logs   Settings

_APL-004-nonsci-Blade1 Payload Monitor

Standby

APL-004 🟩

CPU:
RAM:
Swap:
Disk:

Overview   CPU   Memory   Storage

System uptime: 13:15

HITL Copilot

HITL Copilot is offline

System uptime: no data

Last update: never

main

Standby   Capture attempts: 0

tir 🟩		swir 🟩		mono 🟩		rgb 🟩	
Failures:	0	Failures:	0	Failures:	0	Failures:	0
FPA:	0 °C	FPA:	0 °C				
Lens:	0 °C	Lens:	0 °C				
FPA State:	0	FPA State:	0				

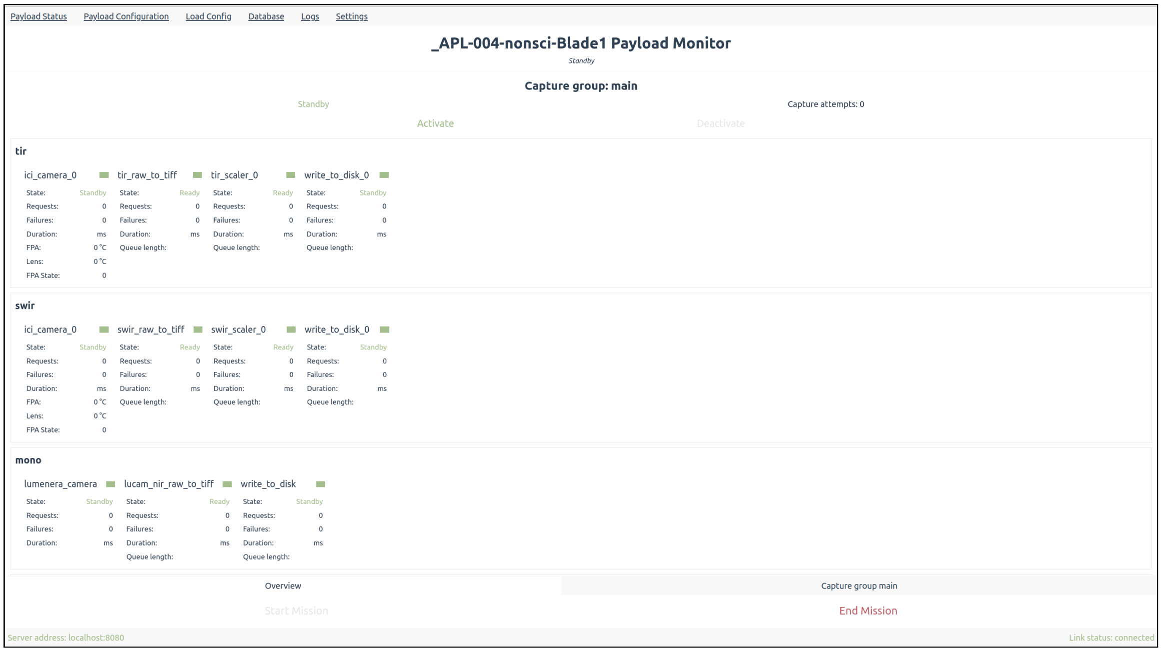
Activate
Deactivate

Overview   Capture group main

Start Mission   End Mission

Server address: localhost:8080   Link status: connected

(a) The payload computer status is shown in the top-left section. The copilot computer status (when one is in use) is shown in the top right section. During a mission, a node status summary is shown in the middle section (above the word Overview).



Payload Status   Payload Configuration   Load Config   Database   Logs   Settings

_APL-004-nonsci-Blade1 Payload Monitor

Standby

Capture group: main

Standby   Capture attempts: 0

Activate   Deactivate

tir

ici_camera_0 🟩		tir_raw_to_tiff 🟩		tir_scaler_0 🟩		write_to_disk_0 🟩	
State:	Standby	State:	Ready	State:	Ready	State:	Standby
Requests:	0	Requests:	0	Requests:	0	Requests:	0
Failures:	0	Failures:	0	Failures:	0	Failures:	0
Duration:	ms	Duration:	ms	Duration:	ms	Duration:	ms
FPA:	0 °C	Queue length:		Queue length:		Queue length:	
Lens:	0 °C						
FPA State:	0						

swir

ici_camera_0 🟩		swir_raw_to_tiff 🟩		swir_scaler_0 🟩		write_to_disk_0 🟩	
State:	Standby	State:	Ready	State:	Ready	State:	Standby
Requests:	0	Requests:	0	Requests:	0	Requests:	0
Failures:	0	Failures:	0	Failures:	0	Failures:	0
Duration:	ms	Duration:	ms	Duration:	ms	Duration:	ms
FPA:	0 °C	Queue length:		Queue length:		Queue length:	
Lens:	0 °C						
FPA State:	0						

mono

lumenera_camera 🟩		lucam_nir_raw_to_tiff 🟩		write_to_disk 🟩	
State:	Standby	State:	Ready	State:	Standby
Requests:	0	Requests:	0	Requests:	0
Failures:	0	Failures:	0	Failures:	0
Duration:	ms	Duration:	ms	Duration:	ms
		Queue length:		Queue length:	

Overview   Capture group main

Start Mission   End Mission

Server address: localhost:8080   Link status: connected

(b) On the Capture group status page the status of each individual node is shown in more detail.

Fig. 2.2: An Example of the StarCommand Status Monitoring

via *ssh* port forwarding to a remote location. As an aside, having the ability to port forward the STARDOS Datalink information means that if the main ground station computer has internet connectivity, a user can configure and help monitor the status of a payload from a distant location that also has internet connectivity. This would allow for in field debugging assistance, or potentially provide the ability for a supervisor to monitor the status of a mission if they are unable to be on site.

Having this information available to the crew allows them to make a decision to continue the flight, pause the flight, or end the flight. Giving the users the ability to monitor the payload status allows for safer and more effective flights as sensor and payload system functionality can be monitored throughout the entire mission.

2.2 STARDOS Datalink

The STARDOS datalink is a key part of the STARDOS architecture. It is what provides the connection from the aircraft to the ground. Some details exist on the STARDOS datalink in "A 55-pound Vertical-Takeoff-and-Landing Fixed-Wing sUAS for Science: Systems, Payload, Safety Authorization, and High-Altitude Flight Performance" [16] and in the follow on journal paper that is currently under review [32]. More detail about the STARDOS datalink will be provided in future works published by AggieAir.

As mentioned previously, while executing a mission the aircraft typically flies far enough away that the user is unable to access the payload system and/or data to determine whether or not the payload is functioning properly. It is out of reach of the user physically and typically out of reach of any sort of WiFi wireless or cellular data network connection. If the user wants to know about the status of the payload system or data being collected they must either use the existing 900 MHz aircraft to ground datalink connection, or they must add an additional radio datalink with the sole purpose of providing payload information. There are advantages to each:

Advantages of use of existing 900 MHz datalink:

- Only a single datalink is used, producing a simplified system architecture

- Less hardware needs to be brought out to the site of the mission and deployed
- Decreased weight (i.e. increased flight time)
- Does not consume extra power, because it is on the aircraft regardless of its use for the payload data.
- The system is more affordable, as additional datalink hardware does not need to be purchased

Advantages of additional payload datalink:

- Higher bandwidth, as the separate datalink could be solely dedicated to transmitting payload data.
- The data does not need to comply with the MAVLink [33] protocol that is required to be used for carrying aircraft/autopilot information on the 900 MHz datalink.
- If mis-used or if this datalink is frozen, it does not compromise the utility of the datalink connection to the aircraft, thus control of the aircraft is sustained.

With the objective in mind of innovating and improving the the process of or ability to perform scientific data collection using sUAS, AggieAir elected to take on the challenge of utilizing the existing 900 MHz datalink radio. The concept of taking advantage of the existing 900 MHz datalink has not yet been explored in literature.

Here a very brief explanation is provided. The existing 900 MHz datalink radio utilizes the MAVLink [33] protocol to transmit messages and information between two systems. As described in the MAVLink developer guide “MAVLink is a very lightweight messaging protocol for communicating with drones (and between onboard drone components)” [33]. MAVlink follows a publish-subscribe framework and provides a point to point messaging system. This allows for messages and information about the drone (e.g. attitude, position, velocity, etc.) to be relayed from the drone to other systems. As mentioned in the MAVlink developer guide, these messages can be sent to onboard computers, although they are often sent from the drone down to the ground station. The autopilot typically is the one to use the MAVLink protocol to send messages about the current state of the drone. If one wants to use the dedicated 900 MHz datalink radio the information sent to the ground needs to be sent within this protocol.

Therefore, AggieAir decided to fit their own data within one of the MAVLink messages that are relayed down to the ground. There are a wide variety of messages that can be sent using MAVLink, and many of them are unused by the autopilot. Most of these messages are small and may only be designed to carry a few 8 bit integers or characters. There are a few messages, however, that are larger than others; these are messages that are designed to send an array of floats down to the ground. After examining the messages that are available, it was decided to use the *LOGGING_DATA* MAVLink message which allows for 249 bytes of arbitrary data to be sent to the ground station through the MAVLink protocol. The *LOGGING_DATA* messages are originally intended to be used for MAVLink logging purposes, but AggieAir decided to 'high-jack' the use of these messages and use it to send STARDOS data.

With the motivation to notify the user about the current status of the payload, STARDOS uses the *LOGGING_DATA* message to send all of the payload computer status and node status information that is displayed in StarCommand. This information is described in section 2.1. This datalink connection provides an invaluable link to the aircraft payload.

An additional feature of MAVLink is the ability to transfer files from one system to another using the MAVLink File Transfer Protocol (MAVLink FTP [34]). This gives STARDOS the ability to upload configuration files from StarCommand to the payload computer. The user is then able to select which configuration they want to upload in the StarCommand user interface and then upload it to the payload computer. This ability makes the STARDOS system fully independent of wired connections and/or a reliance on a WiFi network, as all interaction with the payload is managed through the 900 MHz datalink. This is invaluable as it enables the user to fly missions in remote areas, such as farm land or reservoirs, that do not have internet access.

A simplified system overview of the STARDOS interconnections (including those provided by the Stardos Datalink) are shown in Fig. 2.3

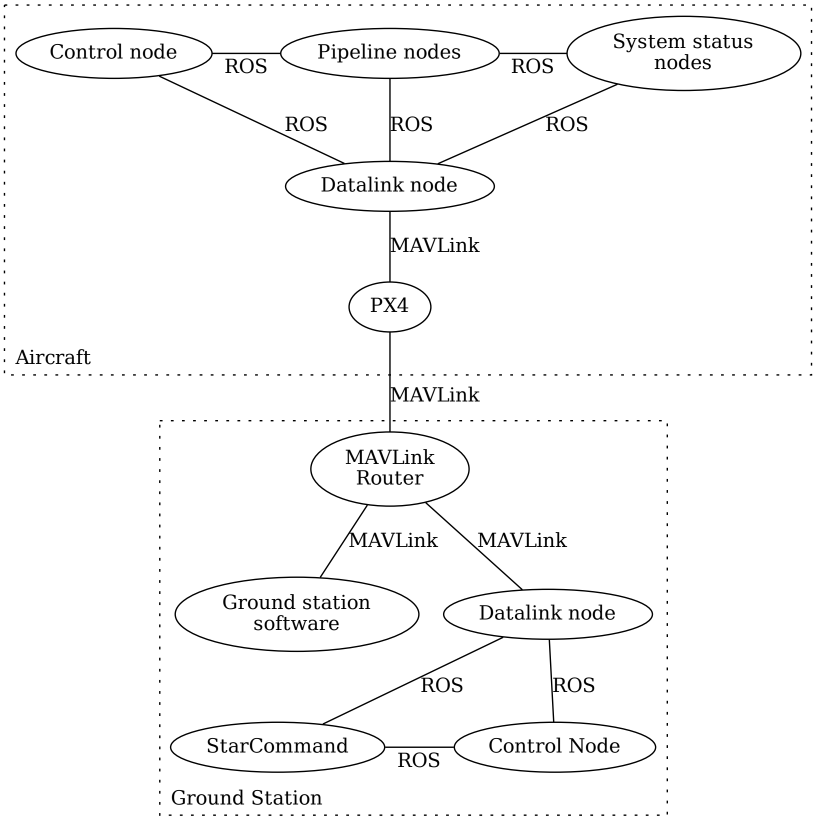


Fig. 2.3: A simplified system diagram showing the interconnections of the various components of STARDOS. Only MAVLink and ROS connections are shown. It is important to note that other protocols are used in smaller ways throughout the system. Also, because ROS is a publish-subscribe framework, anything on the same network can subscribe to topics published by other nodes.

2.3 STARDOS Backend

This section describes the STARDOS Backend, which consists of the software run on the aircraft payload.

As described in [28], STARDOS enables modularity by utilizing Robot Operating System version 2 (ROS 2) a newer version of the original Robot Operating system (ROS) [35]. In the Backend of STARDOS, ROS 2 enables simpler and more modular robotic systems. ROS 2 is a publish/subscribe framework. ROS 2 allows individual components of an overall software architecture to be broken down into software blocks called nodes. Nodes subscribe to topics in order to get messages/data from other nodes, and Nodes publish to topics in order to pass messages/data out to other nodes. Nodes can be interchanged or modified without affecting the system as a whole. ROS 2 Nodes that are used in the STARDOS architecture are referred to as STARDOS Nodes.

In the STARDOS architecture there are STARDOS nodes for individual sensors, Capture groups, data processing, and data storage. These are often referred to as sensor nodes, capture nodes, processing nodes, or sink nodes respectively. Sensor nodes contain software designed for interfacing with a sensor. Data collection in the sensor nodes is triggered by messages sent by the capture nodes. Sensor nodes are grouped together into Capture groups. A Capture group is a grouping of sensor nodes that are told to capture data simultaneously. Capture groups are independent STARDOS nodes that publish a capture message to a particular topic, indicating to the sensors subscribed to that topic when to capture data. Having these STARDOS nodes as a part of the STARDOS architecture allows for a modular system as users can interchange the different versions of Capture groups, sensor nodes, processing nodes, or data storage nodes that will best suit the needs of a specific use-case.

Since the publishing of the previous work [28], a few of the STARDOS architecture concepts have evolved. The concept of Capture groups has changed significantly. In the original STARDOS architecture, Capture groups were designed to be solely cadence-based. This meant that they would trigger a data capture at a set rate for the entirety of a mission. It was determined that there are various other sources which could be useful to use for triggering data captures. Thus, the concept of developing several varieties of Capture groups was implemented.

By developing several different Capture group nodes, the user is able to better select how and when they want their data to be captured. An example of an alternative Capture group variety includes an autopilot-triggered Capture group. Many flight planning software packages, including Universal Ground Control System by SPH Engineering (UgCS) [36] used by AggieAir, can predetermine the locations in which sensor data should be captured. These software packages can optimize the data capture so that a desired overlap can be achieved, a specific area of interest can be more carefully captured, and/or an excess of data is not obtained. The difficulty is that the autopilot is traditionally only able to connect to and send a capture message to a single individual sensor. By creating a Capture group around that functionality, the Capture group can be designed to trigger data capture in multiple sensors upon receiving the capture signal from the autopilot.

Data Processing nodes are nodes that consume either the raw data provided by the sensor nodes or data that has been partially processed by another processing node. These processing nodes can be as simple as converting the raw sensor data to a more human friendly data format, or processing nodes can perform complex tasks such as putting the data through some form of neural network, performing real-time image stitching, or any other form of data processing that could be of use. These nodes are the most relevant because they allow users to run real-time data processing during flight. Data storage nodes take the processed data and save the desired information to the disk for later evaluation/processing.

All nodes are managed by a STARDOS node called a Control node, which manages the startup and shutdown of the node processes. The control node acts similarly to the `init` process in a Unix system, starting up and ending ROS 2 node processes by taking advantage of SystemD process control features.

Data collection alone is quite difficult, as it requires correctly interfacing with sensors, triggering their data collection with appropriate timing, in addition to the storing, validation, and processing of their data. Collecting data onboard a flying robotic aircraft is more difficult as the operators do not have direct access to the sensors or computing hardware onboard the aircraft and thus cannot immediately analyze the sensors output for data validity. A real-time data status and data configuration through StarCommand are two features that allow STARDOS to fulfill its purpose: to simplify and modularize the use of sUAS payloads by pilots, field scientists, and other sUAS operators.

2.4 Summary

In this chapter a summary of AggieAir's STARDOS architecture has been presented. STARDOS is a scientific data collection system that facilitates payload management and monitoring through an intuitive user interface, StarCommand, as described in section 2.1. Additionally, it is necessary to have a communication link with the payload and AggieAir's implementation of the STARDOS datalink radio setup, taking advantage of the existing 900 MHz radio, was introduced in section 2.2. Lastly, the software onboard the payload itself for interfacing with hardware data sensors and processing and storing data was explained in section 2.3. STARDOS provides a new innovative way for users to be able to interact with and manage scientific data collection payloads.

CHAPTER 3
RELATED WORK

This section addresses preceding works that are related to the research presented. There is some initial motivation for simulator use on the risks of sUAS flight in section section 3.1. There is an exploration of work related to sUAS simulators utilized by PX4 and their current capabilities and applications in section 3.2. Furthermore, sUAS simulators that utilize Gazebo are described in section 3.2.1. Previous research is analyzed demonstrating the utility of low-cost thermal sensors and a need for a simple, affordable, thermal mapping sUAS payload in section 3.3.

3.1 Risks of sUAS Flight

Drone crashes are a fact of flight. Whether caused by user error [37] [38] or technological malfunction in the aviation equipment [39], drones have crashed and will continue to crash. This is an issue when doing research on drones and drone systems, not to mention the issue of safety. If a crash occurs and an aircraft is carrying a scientific or experimental payload then the payload and the scientific progress of that payload will be lost. The FAA has put in place rules and regulations to help keep people safe [40], but even when following these regulations, crashes still occur.

3.2 Simulation

Knowing that crashes occur, many have leaned on drone simulators in order to develop, test, and enhance the field of sUAS (for examples see [41] [42] [43]). Simulation allows for aspects of sUAS to be tested and proven, without the risks associated with an actual flight. The literature shows that simulations have been developed for testing a large variety of sUAS applications.

AggieAir has chosen to use PX4 [44] as their main autopilot software. Developers of

PX4, the BSD-licensed open-source autopilot software, have taken advantage of existing drone simulators to test out their autopilot code. They provide a variety of simulation options to those in the community for testing purposes that utilize existing simulation environments:

1. Gazebo Classic - Gazebo Classic is a 3D robotics focused simulator. It is widely used throughout the robotics industry in teaching [45] [46] and research [47] [48] scenarios.

2. Gazebo - "Gazebo supersedes Gazebo Classic, featuring more advanced rendering, physics and sensor models." [49] Use of the newly updated Gazebo simulation environment has only recently been added to the list of PX4's simulation capabilities.

3. FlightGear - FlightGear is visually and physically realistic, providing a relatively accurate environment for use in training sUAS operators how to manipulate the aircraft. It also simulates various types of weather and environmental conditions. FlightGear has also been used by Wang and Li for improvements in quad tilt rotor control [50], as well as Pan LongFei et al. to create a Flight Vision Simulation System Based on data collected using FlightGear [51].

4. JSBSim - "A simulator that provides advanced flight dynamics models. This can be used to model realistic flight dynamics based on wind tunnel data." [49]. JSBSim has also been used to improve control system designs [52].

5. jMAVSim - jMAVSim is an easy to use simulation setup for testing copter type vehicles.

6. AirSim - AirSim was developed by Microsoft and it uses Unreal Engine. It was designed as a platform for "AI research and experimentation." [53]

PX4 has built off of these simulation environments and provided a way to test out the PX4 autopilot code using these simulators. In addition to these software simulation environments, listed above, PX4 also provides a Hardware in the Loop (HITL) simulation environment as well as what they call Software in Hardware (SIH). For more info on the PX4 simulation options see the PX4 documentation [49].

3.2.1 Gazebo for a Scientific Data Collection Simulation Environment

With the objective in mind of simulating AggieAir's scientific quality data collection process, this research followed the constraints listed in section 4.1. This includes using the simulators provided by PX4 and that the simulation must use a quad-plane airframe style. An example of a quad-plane aircraft is shown in Fig. 3.1. With these constraints in mind, it was noted that the only simulation environment that currently supports the quad-plane airframe is the Gazebo-Classic simulator. Ultimately Gazebo-Classic was chosen as the simulator of choice for this research.

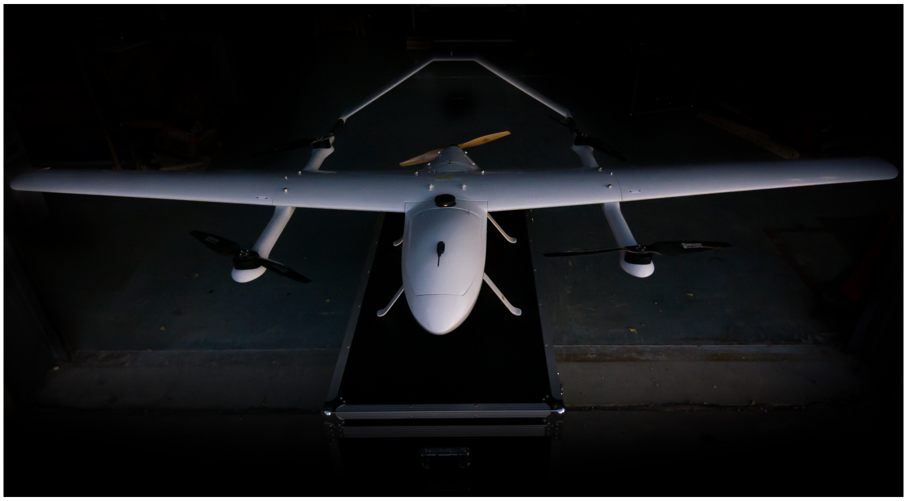


Fig. 3.1: AggieAir's GreatBlue quad-plane VTOL Transition Aircraft. The Quad term refers to the 4 motors and propellers used for vertical take-off and landing. The plane term refers to the fact that there is a separate pusher motor along with the wings to make horizontal flight possible.

Some literature was reviewed that pertains to Gazebo and Gazebo-Classic to better understand the utility of the Gazebo based simulators. Gazebo has often been paired with ROS or ROS 2. The works described in this section use both Gazebo-classic as well as the newer Gazebo, however this provides good insight into the varying ways that Gazebo based simulators can be used.

Gazebo and ROS are used widely throughout the industry of robotics and robotics research. Gazebo/ROS have been used to simulate path planning of quadrotors. For example, Hernandez et al. used the “resistive grid path planning methodology” to navigate and plan 3D trajectories for avoiding collisions [54]. The Gazebo/ROS combination is very versatile, allowing researchers to implement their own custom plugins. An example of this is when Alajami, Pous, and Moreno implented an RFID plugin that enables the simulation of RFID readers and antennas to be mounted on robots and used a ROS/Gazebo environment [55]. Gazebo/ROS is very modular, allowing researchers to test a wide variety of sensor placements and varying environments for robots. Safin, Lavrenov, and Martínez-García provide a Gazebo environment to test out simultaneous localization and mapping (SLAM) algorithms to navigate and map their environment. The simulation provides a way to test out varying terrains and makes it easy for the user to change and rearrange the sensor configuration and placement [56].

Gazebo and ROS have also been used by Miranda et al. as a simulation environment to provide numerical results in the design of a dual-core model predictive controller for a tiltrotor UAV [57]. Moon et al. developed a Gazebo/ROS based sUAS simulator focused on simulating realistic wireless communication between networked UAS [58].

Noting the wide use and the versatility of Gazebo-classic and Gazebo as sUAS simulators and seeing that Gazebo based simulation environments are being used to improve varying capabilities of sUAS. These capabilities include path-planning, controllers, and wireless communication. For these reasons AggieAir elected to move forward with Gazebo-Classic as the simulation of choice for this research work. More of the motivation for this selection is detailed in section 4.2.

3.3 The Utility of Low-Cost Thermal Sensors

First responders use thermal cameras for finding missing persons or identifying hot spots in disasters [24]. UAS with thermal cameras and data collected by them are used to study fire behavior and determine fire rate of spread [25] [26]. Non-visible hazardous materials, such as leaking gases, are detected using thermal cameras in the oil and gas sector [27]. Thermal UAS imaging is used to image swampy areas such as peatlands. [59] [60]. Close-range aerial imagery has been used for surveying hot springs [61]. Crop water stress indexes can be computed using thermal imagery collected using an sUAS [62]. Thermal cameras have been employed to collect data for a study on object detection for traffic management [63].

Recognizing the many use cases of thermal imagery, researchers have already started looking into low-cost thermal payload use-cases and implementing solutions. Dávila-Sacoto et al. did a study in which they "Detect(ed) Hot Spots in Photovoltaic Panels Using Low-Cost Thermal Cameras" [64]. Their study showed that low-cost thermal cameras were able to provide data with error rates below 10%. Al-Shammari et al. did an analysis of low-cost thermal sensors to see if they have the ability to detect oil spills, collecting data at both a petroleum company location as well as an indoor experiment [65]. Their results showed that the thermal sensor could perform well when detecting the difference between water and oil, but needed improvement if there were many different objects in the frame. Low-cost thermal sensors have also been put to the test in mining exploration by Pérez-Álvarez et al. [66]. In their research they utilize sUAS and low-cost sensors to distinguish between limestone and dolostone that contain varying iron content allowing them to detect places that could contain more minerals than others. In their research the low-cost sensors also proved valuable, differing by only 4.57% from traditional methods.

Low-cost thermal cameras have also been analyzed for feasibility for monitoring water stress in cherry trees [67]. Having this kind of information would allow farmers to collect data that prompts a decision (to water more or less). Perdana, Risnumawan, and Sulistijono created a convolutional neural network that can use low-cost thermal data to detect victims that may be stuck during a natural disaster [68]. There are many works that explore and exploit the practicality of low-cost thermal sensors.

While low-cost thermal cameras may not provide the same kind of scientific quality data when compared to more expensive options their data still proves useful. Low-cost thermal sensors can provide initial estimates or be used in cases where highly scientific data may not be required. There are many cases where just the presence of a temperature difference is data sufficient enough for the task.

These works, and many others similar to them, show both the great utility of thermal cameras as well as the great potential that low-cost thermal sensors have for fulfilling these roles. High-end thermal cameras can be expensive, costing thousands of dollars [69] [70]. The ability to collect thermal data in the field using a low-cost payload has proved valuable. AggieAir sees benefit in implementing a testing platform for CAASO that takes advantage of lower-cost thermal technology that has become available. The wide use and adaptation of thermal cameras in many fields, particularly with applications using sUAS, shows the need for a payload that provides thermal data more affordably.

3.4 Summary

These works relating to Gazebo as a simulation environment as well as low-cost thermal sensors provided insight into the need for the research presented in this work. Through the literature review understanding was gained that Gazebo-Classic is a plausible candidate that would provide a base for a simulation platform for CAASO. Moreover, low-cost thermal sensors have been developed to the point that they prove effective enough for many thermal imaging use-cases. Thus a low-cost thermal mapping payload seemed that it would prove valuable as the payload of a testing platform for CAASO.

CHAPTER 4
A COHESIVE SIMULATION PLATFORM FOR CIVIL AUTONOMOUS AERIAL SENSING AND OPERATIONS

As stated in section 1.1, “Knowing of the risks and dangers associated with sUAS flight, one of AggieAir’s objectives is to innovate and improve the process of or ability to perform scientific data collection using sUAS.” With this objective in mind, a simulation environment will provide an opportunity for researchers to test and improve the process of scientific quality data collection. As has been noted previously, if a new scientific data collection process may only be tested on an actual flying system, then it creates greater risk for the system and others. Because flying an sUAS is inherently high risk, some would say that the safest way to fly is to not fly at all. In this chapter, a simulation platform for CAASO is presented.

4.1 Constraints

While a simulator that can demonstrate every possible use-case of civil autonomous flight would be useful, it was important to set certain constraints on the simulation project to limit the scope of the research. These constraints, in particular focused on re-creating in simulation the current AggieAir scientific data collection system:

1. The simulation must have the ability to simulate a quad-plane airframe
2. The simulation must use the PX4 autopilot as the flight controller
3. The simulation must have the ability to capture data from a simulated camera
4. The simulation must be able to interface with the AggieAir STARDOS system to trigger data capture and for data processing throughout simulation

Having the research project focused on simulating the AggieAir scientific data collection system would allow AggieAir to test, in the same simulation environment, almost all of the systems that they use. These systems include the autopilot, the payload, and the ground station setup. Testing these systems in simulation this would reduce the number of actual sUAS flights and risk associated with performing a system test involving actual sUAS flight.

The simulation environment is focused on improving the scientific data collection process. In order to do this, it would be necessary for there to be data present in the simulator for the simulated camera to collect. An additional feature of the simulator is the ability to import existing data sets in the form of an orthomosaic (orthomosaic is defined in section 1.3) such that the simulated sUAS can 'fly' over them and simulate the collection of scientific data.

4.2 Simulation Environment Selection

As noted in section 3.2, many sUAS simulators already exist. For this research it was advantageous to use one of these existing simulation environments and modify it to be able to meet the desired simulator requirements. The simulation environments described in section 3.2 were evaluated to see which of these environments would be the best fit to meet the requirements described in section 4.1.

Three criterion were of particular interest: The airframes that the simulators supported, whether or not the simulators supported simulated sensors and/or cameras, and whether the simulation environment could be easily modified or customized. For the airframe, it was desireable for the simulator to have built-in support for the quad-plane VTOL airframe. Furthermore, it was important that the simulator support simulated sensors to be able to simulate the collection of scientific quality data. Lastly, the simulation environment itself needed to be customizable; this is key, as it allows the user to test their scientific data collection system in varying circumstances.

An evaluation of these criterion was performed and the results are shown in 4.1.

Table 4.1: Open-source drone simulators and features that they support.

Simulator	**Supported Airframes**	**Simulated Cameras**	**Customizable Environments**
Gazebo Classic	Quad, Hex, Quad-Plane VTOL, Tailsitter VTOL, Plane, Rover, Submarine	Yes	Yes
Gazebo	Quad, Plane, Quad-Plane VTOL [1]	Yes	Yes
FlightGear	Autogyro, Plane, Rover	No	Yes
JSBSim	Plane, Quad, Hex	No	No
jMAVSim	Quad	No	No
AirSim	Quad	Yes	Yes

For more information on the simulation environments either refer to section 3.2, or refer to the official PX4 documentation [49]. Additionally, for more information regarding the PX4 airframes (including images of each airframe configuration) see the PX4 documentation on airframes [71].

At the time of development, the only simulator that supporteds the quad-plane VTOL airframe is the Gazebo Classic Simulation environment (in the PX4 airframes the quad-plane VTOL style is referred to as a standard VTOL airframe). Because Gazebo Classic also supported both simulated cameras and customizable environments, Gazebo Classic was chosen as a suitable simulation environment for the simulation platform. At the time of writing, the newer Gazebo simulation environment has newly added support for the quad-plane VTOL. After investigation, however, the newer Gazebo does not yet have PX4 support for a MAVlink style camera, as is described in 4.3.1.

4.2.1 Advantages of the simulation environment

There are advantages to using a simulated environment. When using Gazebo Classic, one can speed up the simulation time. The PX4 documentation notes that "Powerful desktop machines can usually run the simulation at around 6-10x, for notebooks the achieved

[1] At the time of writing a quad-plane VTOL airframe is supported in Gazebo. At the time of development, however, the quad-plane VTOL airframe was not yet supported.

rates can be around 3-4x" [49]. Having a quick turn around from one simulated flight to the next is valuable. This feature has not yet been tested with the cohesive simulation platform and is left for future work. This allows more test flights to be performed in the same amount of time. Furthermore, the environment can be changed without having to physically get up and go to a new location. The user can simply swap out the environment that is being used in the simulator and the user can test to see if the same parameters that performed well in one environment will perform similarly in another. Although it was not thoroughly explored in this work, Gazebo Classic supports importing digital elevation models as well. This could enhance the capability of the simulator and provide even more insight and would be valuable future work.

4.3 Simulation Platform Development

In this section, information regarding the development and implementation of the quad-plane VTOL airframe with an attached camera in section 4.3.1 is presented. Furthermore, this section includes a description of how existing data sets (in the form of an orthomosaic) can be imported into the simulator in section 4.3.2. A description of the simulator interface to AggieAir's STARDOS system is described in 4.3.3. Lastly, an example simulation environment is presented in section 4.3.4 along with the system design for using this simulation environment with real data.

4.3.1 Quad-Plane VTOL with attached Camera Sensor

Gazebo Classic, a robotics focused simulator, natively provides support for a wide variety of simulated sensors. As described in the Gazebo Classic tutorials "A simulated sensor analyzes the environment and produces a data stream that closely matches the sensor's physical counterpart." [72]. Gazebo Classic provides the ability to model various types of sensors including cameras, contact sensors, as well as torque/force sensors. Additionally, one can even model sensor noise, which typically can aid the development of varying types of Kalman [73] or extended Kalman filters [74].

The PX4 simulation titled 'Gazebo Plane Cam' provides a MAVLink based camera.

The PX4 Gazebo Classic plane camera simulator was designed to give users the ability to simulate drone surveying using the ‘plane’ airframe [75]. While it could be useful to develop a custom Gazebo Classic camera sensor for use with the standard VTOL airframe, the objective of this research was not focused on enhancing or creating an individual gazebo sensor, but rather on creating the simulation platform as a whole. For this reason it was decided it would be best to combine the existing Quad-Plane VTOL airframe with the already existing Gazebo Classic camera that is used in the existing PX4 Gazebo Plane Cam simulation. These two elements combined create a new simulation that uses the Quad-Plane VTOL airframe/camera combination.

The existing camera sensor used in the PX4 Gazebo Classic plane camera simulation is called the “geotagged cam” [76]. It is called the geotagged cam because it performs what is known as geo-tagging on the images that it captures. geo-tagging is the process of attaching meta-data to the data captured which associates the data with a location (latitude, longitude, and altitude), the attitude of the aircraft, as well as other relevant information about how the data was captured. This may include data about the sensor itself, including the make and model, the field of view, the f-stop (or f-number), the orientation of the camera, etc.

The geo-tagged cam is managed by the PX4 gazebo camera manager plugin. This means that the Gazebo camera acts and behaves like a MAVLink camera. As described in the PX4 documentation [75], being a MAVLink camera allows the simulated camera to respond to MAVLink commands. These commands come from other MAVLink nodes which command the camera to capture data, to report its status, to set the camera mode, to set the camera zoom, and a few other options. Thus, MAVLink is the interface for interacting with the camera.

Gazebo Classic allows users to create and import their own models as described in their official documentation [77]. Gazebo Classic uses Simulation Description Format (SDF) files. In these files the user describes and defines how they want their model to be represented within the simulation world. The user is able to define a wide variety of parameters including shape and size, the dynamics of a specific part of the model, the amount of light that it should give off within the simulation world, and many other parameters. These parameters provide flexibility to the user so that they can tailor the simulation to try and recreate the environment in which the actual hardware will be tested and used. The user is able to configure and set up the models that they want to use as well as describe the environment that they are placed in.

To model the quad-plane VTOL airframe, the PX4 community created an SDF file to describe how the quad-plane model should look and act within the Gazebo Classic simulator. Additionally, the PX4 community created the MAVLink based survey camera, which also has it's own SDF file.

In the existing Gazebo Classic plane camera simulator, an SDF file describes the combination of the existing models of both the plane airframe and the geo-tagged camera. Similar to the plane camera simulator, a new SDF file was made for the Quad-Plane VTOL camera simulator. In this SDF file, descriptors are used appropriately to indicate that the existing quad-plane VTOL airframe and the geo-tagged camera should be combined into a single model.

After the SDF file is created it can be imported into Gazebo without running the PX4 simulation. This is done to show that the model is valid and that the airframe model and the camera model have both been properly combined. This model, however, is still unable to simulate flight, as that ability is provided by the PX4 autopilot software.

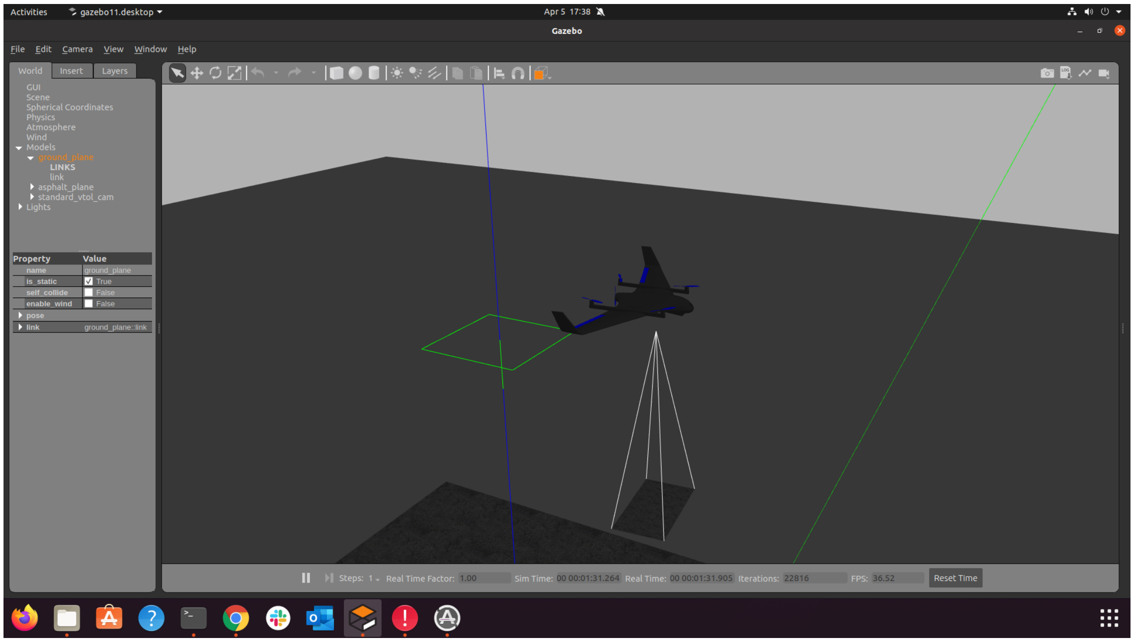
The files were either modified or created to allow the PX4 code to build the quad-plane VTOL camera aircraft and place it into the Gazebo Classic environment. By integrating the models with the PX4 build environment the aircraft is able to be flown by the PX4 autopilot code. The files that were modified and added on top of the PX4 repository are described in appendix A.

Similar to the other PX4 Gazebo Classic simulation environments the quad-plane VTOL camera aircraft can be built and run in a Linux environment using the command:

```
make px4_sitl gazebo-classic_standard_vtol_cam
```

As noted in the PX4 Documentation, proper setup of the environment needs to be done prior to running this command [49]. This command runs the PX4 software-in-the-loop (SITL) simulation using the quad-plane VTOL airframe with the geo-tagged camera attached.

In Fig. 4.1, an image of the Gazebo Classic quad-plane VTOL simulation running with the PX4 autopilot code is shown. In this figure we see the quad-plane airframe in flight. Below the aircraft are 4 diverging white lines that end with a black square at the bottom. These white lines represent the field of view of the Gazebo Camera. The black square is showing what the camera is currently viewing (because the ground in the simulation is black, the square is showing black).



Property	Value
name	ground_plane
is_static	☑ True
self_collide	□ False
enable_wind	□ False
pose	
link	ground_plane::link

Steps: 1 Real Time Factor: 1.00 Sim Time: 00 00:01:31.264 Real Time: 00 00:01:31.905 Iterations: 22816 FPS: 36.52

Fig. 4.1: The quad-plane VTOL camera aircraft running in the Gazebo Classic environment while being controlled by the PX4 autopilot code. The white diverging lines represent the field of view of the camera. The black box at the bottom of the white lines is what is currently in view of the camera.

4.3.2 Importing Existing Orthomosaic data sets

A key feature of the simulation platform for proving and testing aerial sensing and operations is the ability to simulate and test in a wide variety of environments. Gazebo has built in features that allow users to change the environment through aforementioned SDF files. These files are not only used to describe robots, aircrafts, or sensors but also the objects that make up the environment (e.g. buildings, trees, ground).

It is desireable to be able to use be able to import previously collected data sets to be a part of the simulation environment. This provides a simulation environment that is better for proving scientific data collection systems. This allows the user to interchange a variety of data sets into the simulation environment and prove and test their scientific data collection systems. By flying over previously collected data one can gain new insights of how to better collect scientific data in future sUAS missions, greater understanding can be gained relating to what types of patterns need to be flown or what sensors are to be used in a wide variety of environments.

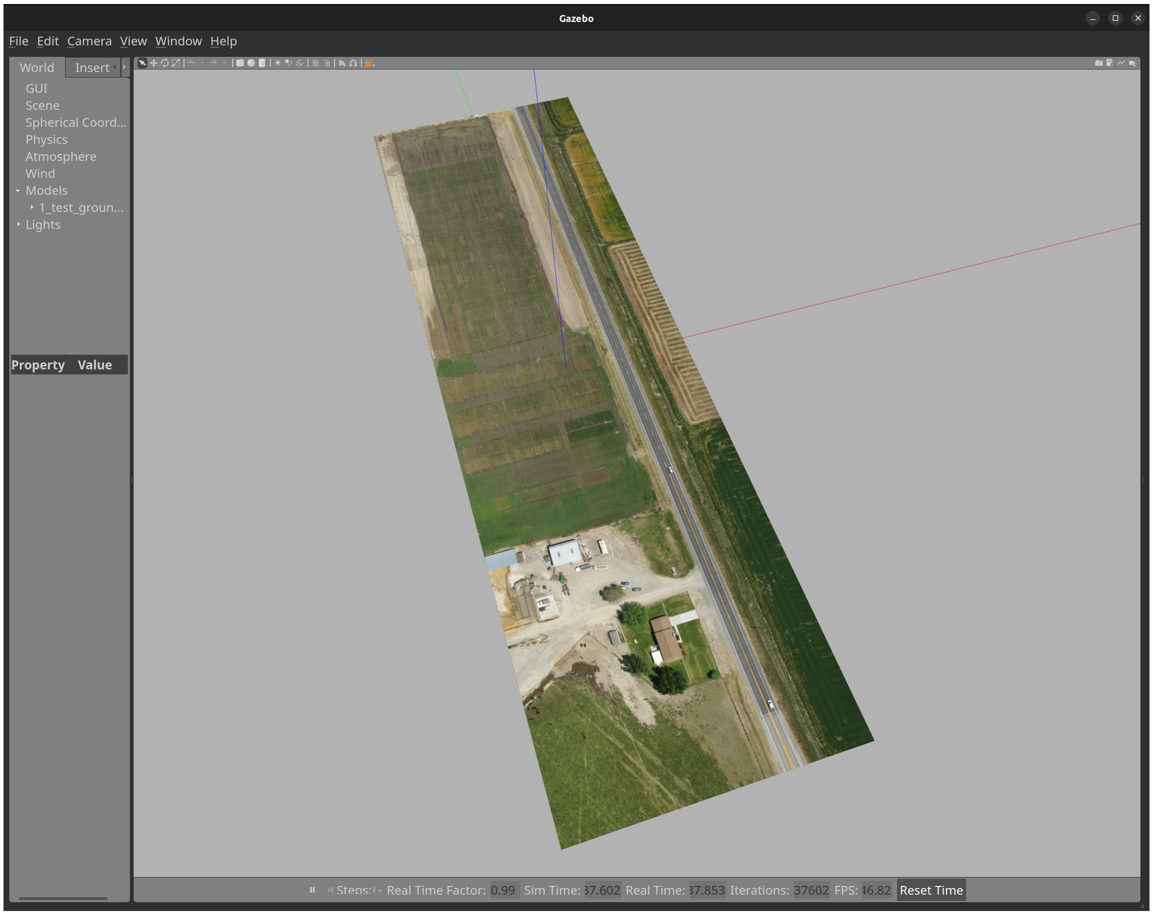
There exist free data sets in the form imagery that has been collected by satellites or using an sUAS available online. Additionally, organizations (such as AggieAir) have data that they have collected from previous scientific data collection flights using sUAS. These data are often compiled into orthomosaics. Orthomosaics are maps that are created by stitching together data that have been collected in the same area.

Generally the creation of an orthomosaic involves processing using key point extraction techniques. A key point is a point that is (often) visually distinct in an image. Key points are used in computer vision for analyzing and comparing data or identifying objects. The imagery that was collected can then be compared, rotated, aligned, and stitched together into a single image using these key points. Software that does this kind of image stitching also typically take into account image metadata. This metadata is provided through the previously described geo-tagging process.

The creation of orthomosaics is an area of of research in and of itself [78] [79] [80]. AggieAir's STARDOS architecture provides a solid framework in which this type of image processing could be done onboard the aircraft in real-time. In many instances, however, the data sets are collected and then processed using a post-processing software. An example of this is a software created by Agisoft called Metashape [81].

Gazebo Classic allows for images to be used to provide coloring (or texture) to the models that have been imported into Gazebo Classic, as is described in the Gazebo Classic documentation [82]. To represent a previously collected data set in Gazebo Classic, an SDF file is made describing a very large and flat surface. This large and flat surface represents the ground plane of an area of interest. The SDF file is modified to indicate that the orthomosaic should be the texture for this large surface, which is then imported into the Gazebo Classic simulation environment. This creates a large orthomosaic-based ground plane (an example of this is shown in Fig. 4.2).

In this example the size of the flat model is the size of the image in pixels divided by 10. This is only an approximate estimate of the relative scale of the orthomosaic. More analysis needs to be done in future work to determine what an appropriate scale factor would be in a general case to make the images accurately sized and scaled.



Steps: Real Time Factor: 0.99 Sim Time: 37.602 Real Time: 37.853 Iterations: 37602 FPS: 46.82

Fig. 4.2: An orthomosaic imported into Gazebo Classic. This is made from data collected using an sUAS and is used as the texture of a large flat surface object in Gazebo Classic. The orthomosaic is of an area that is approximately 30 acres. The location where the original orthomosaic data was collected can be found by looking at the latitude and longitude in decimal degrees: 41.574231 N, 112.131108 W.

4.3.3 STARDOS Simulation Platform

The simulation platform is brought together into one cohesive system for CAASO. A system diagram of this simulation platform is shown in Fig. 4.3.

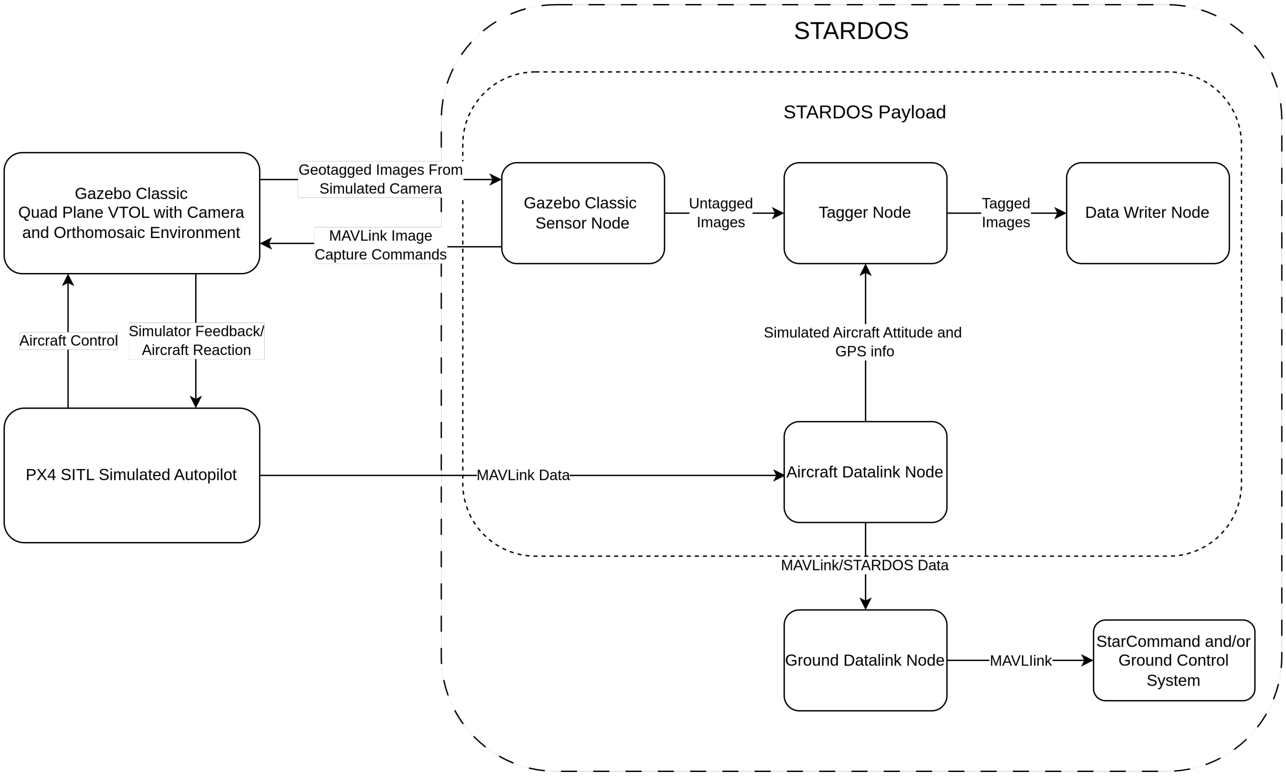

STARDOS
STARDOS Payload
Gazebo Classic Quad Plane VTOL with Camera and Orthomosaic Environment
Geotagged Images From Simulated Camera
MAVLink Image Capture Commands
Gazebo Classic Sensor Node
Untagged Images
Tagger Node
Tagged Images
Data Writer Node
Aircraft Control
Simulator Feedback/ Aircraft Reaction
Simulated Aircraft Attitude and GPS info
PX4 SITL Simulated Autopilot
MAVLink Data
Aircraft Datalink Node
MAVLink/STARDOS Data
Ground Datalink Node
MAVLlink
StarCommand and/or Ground Control System

Fig. 4.3: A system diagram showing how the STARDOS architecture is used in combination with the SITL simulation described here.

This system integration was done using AggieAir's STARDOS architecture and the software-in-the-loop simulation described previously. The resulting Gazebo Classic setup along side the ground control station software is shown in Fig. 4.4.

A STARDOS sensor node was developed to interface with the Gazebo Classic simulated camera. Having a STARDOS sensor node to interface with the Gazebo Classic simulated camera allows users to select the Gazebo camera as a sensor in the StarCommand user interface. The Gazebo Classic PX4 quad-plane VTOL camera simulation must first be started. The STARDOS system is then able to manage the camera sensor by sending *capture* commands and adjusting other camera parameters as needed through the STARDOS

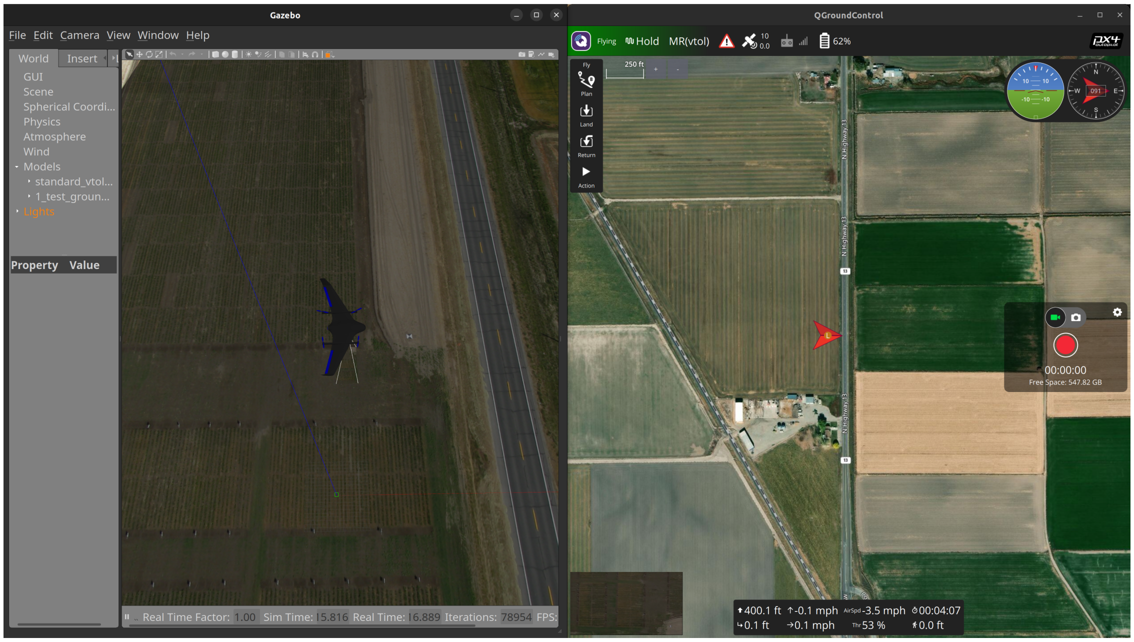


Fig. 4.4: The resulting modified Gazebo simulation environment: a VTOL Transition aircraft in a simulation environment with an orthomosaic environment (left), and the GCS showing the GPS location of the aircraft is aligned with the location of the orthomosaic

MAVLink interface.

From the perspective of the STARDOS system, it is unaware that the Gazebo Classic camera node is a simulated sensor and that it is being used in a simulation environment. This provides a realistic testing platform for STARDOS and can do the same for other scientific data collection systems.

Through the STARDOS datalink, the STARDOS architecture is able to interface with the PX4 SITL autopilot and get the information it needs to properly tag the images in the data set. This is done by the Tagger node. The default PX4 Gazebo Camera provides geo-tagged images. AggieAir decided to remove the geo-tags provided by the simulation itself in order to more fully test the STARDOS architecture pipeline, this is because on a real payload, the images need to be tagged by the STARDOS architecture, so before passing the images to the tagger node they are stripped of their metadata so that the STARDOS architecture can provide the geo-tag information.

4.3.4 Example Simulation Environment and Example Use-Case

A system diagram for a general use of the cohesive simulation platform for CAASO is shown in Fig. 4.5. This section discusses this diagram describing how one could use the simulation platform.

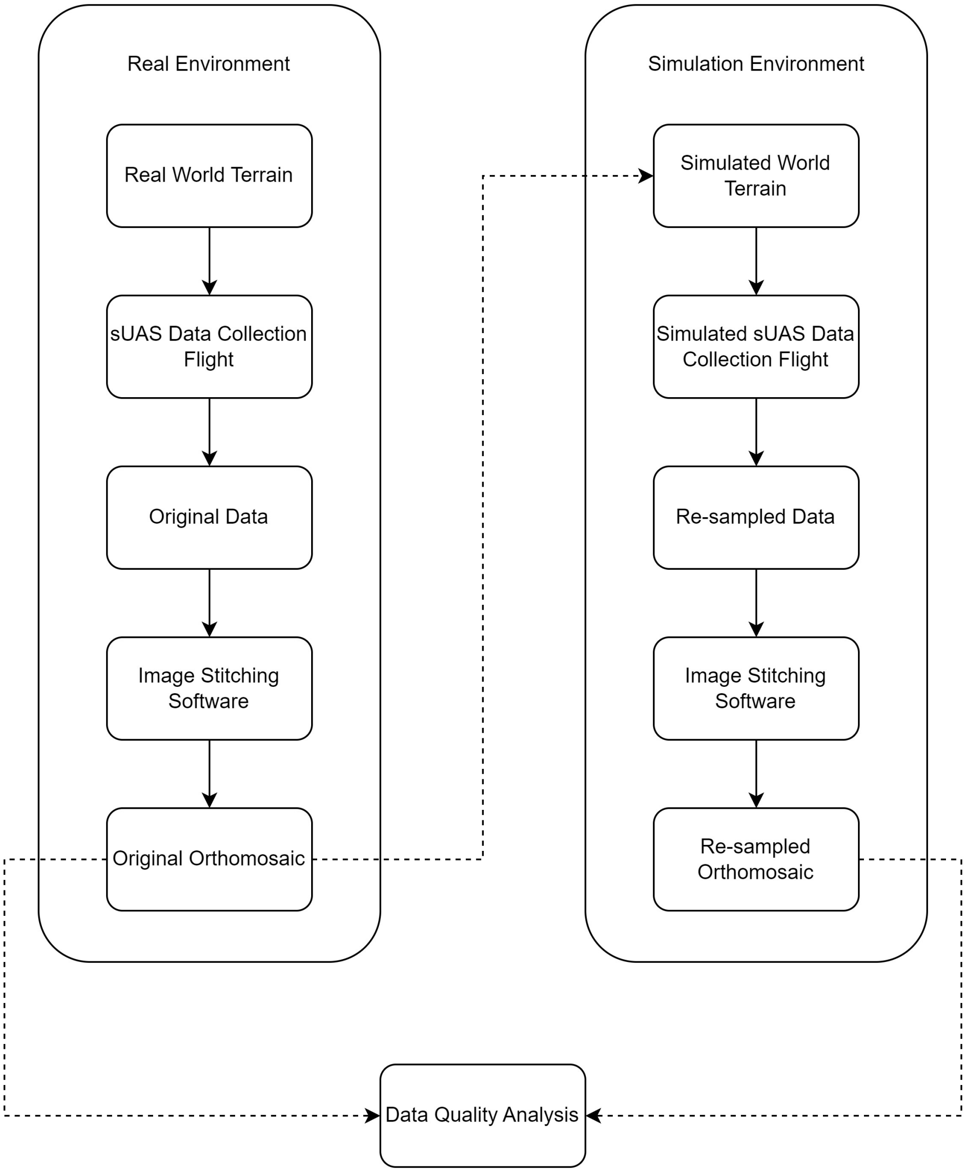


Fig. 4.5: A flowchart illustrating the concept of real data collection and re-sampled data collection.

There are two environments: A real environment, and a simulated environment. The real environment is any place in the world where one could collect scientific quality data of interest. The simulated environment is in the Gazebo Classic simulator that has been described in this chapter.

In the real environment, there exists real terrain. A real sUAS with a payload performs a data collection flight to collect what is referred to as the "original data". This original data is put through some form of data processing to create what is referred to as the original orthomosaic.

In the simulated environment, the original orthomosaic is placed into the simulation platform using the orthomosaic texturing technique described in section 4.3.2, creating the simulated world terrain. The quad-plane VTOL camera aircraft can then be used to fly a simulated sUAS data collection flight, taking images of the simulated terrain with the simulated camera. In this research, the process of imaging in the simulated environment the original orthomosaic is referred to as "re-sampling".

This produces a simulation-based data set. Hereafter referred to as the "re-sampled data set". The re-sampled data set can then be put into an image stitching software as any aerial data set. The image stitching software will produce a new orthomosaic based on the re-sampled data set. Orthomosaics generated in this fashion are referred to as "re-sampled orthomosaics".

Comparisons could then be made between the original orthomosaic and the re-sampled orthomosaic. However, this tool may prove more useful for making comparisons between various re-sampled orthomosaics that have been generated with slightly varying parameters. There may be varying quality output depending on the manner that the data is collected. In the simulation environment, the quality of the scientific data may be effected by parameters of the data collection flight. These parameters may include: image overlap, the amount of light, or flight pattern.

Using a data set that was previously collected by AggieAir, an area of approximately 30 acres was represented in simulation (see Fig. 4.2). The quad-plane VTOL airframe with the Gazebo Camera aircraft was used. Additionally, a STARDOS configuration was set up to use the Gazebo camera sensor node, AggieAir's metadata tagging node, and a node for writing the data to disk. These nodes were configured in series, such that the data was passed from one to the other consecutively. Data capture is triggered by a STARDOS capture group node. Fig. 4.6 Shows the node setup and data flow through the nodes for the cohesive simulation platform.

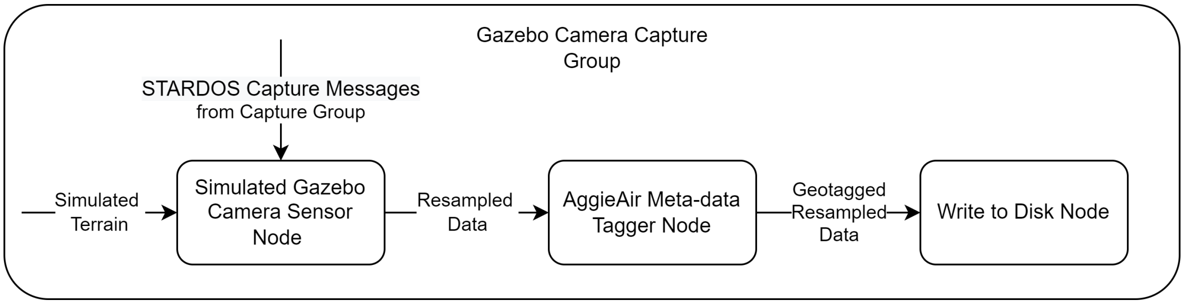

Gazebo Camera Capture Group
STARDOS Capture Messages from Capture Group
Simulated Terrain
Simulated Gazebo Camera Sensor Node
Resampled Data
AggieAir Meta-data Tagger Node
Geotagged Resampled Data
Write to Disk Node

Fig. 4.6: A diagram showing the data flow and node setup for the STARDOS example of the cohesive simulation platform for CAASO.

4.4 STARDOS Simulation Results

This section shows some results from the cohesive simulation environment with the example Gazebo Classic simulation and running a simulated STARDOS payload with the Gazebo camera sensor node. The cohesive simulation platform was set up as previously described using the example setup shown in Fig. 4.4. Additionally, STARDOS was configured with a configuration as described in section 4.3.4 and shown in Fig. 4.6.

Using PX4's mission planning software QGroundControl [83], a VTOL transition flight and mission plan was implemented. A screenshot of the resulting mission plan is shown below in 4.7. The flight plan was fully autonomous, including a VTOL takeoff and transition to horizontal flight, as well as a transition from horizontal flight to a VTOL landing. A flight plan consists of a number of way-points. Way-points are the ordered locations (typically Latitude, Longitude, and Altitude) that the aircraft will fly to. A flight consists of some number of way points, depending on the size of the flight plan. The simulated flight was setup with a total of 49 way-points (this includes the transition way-points for takeoff and landing) configured to fly at an altitude of 100 ft.

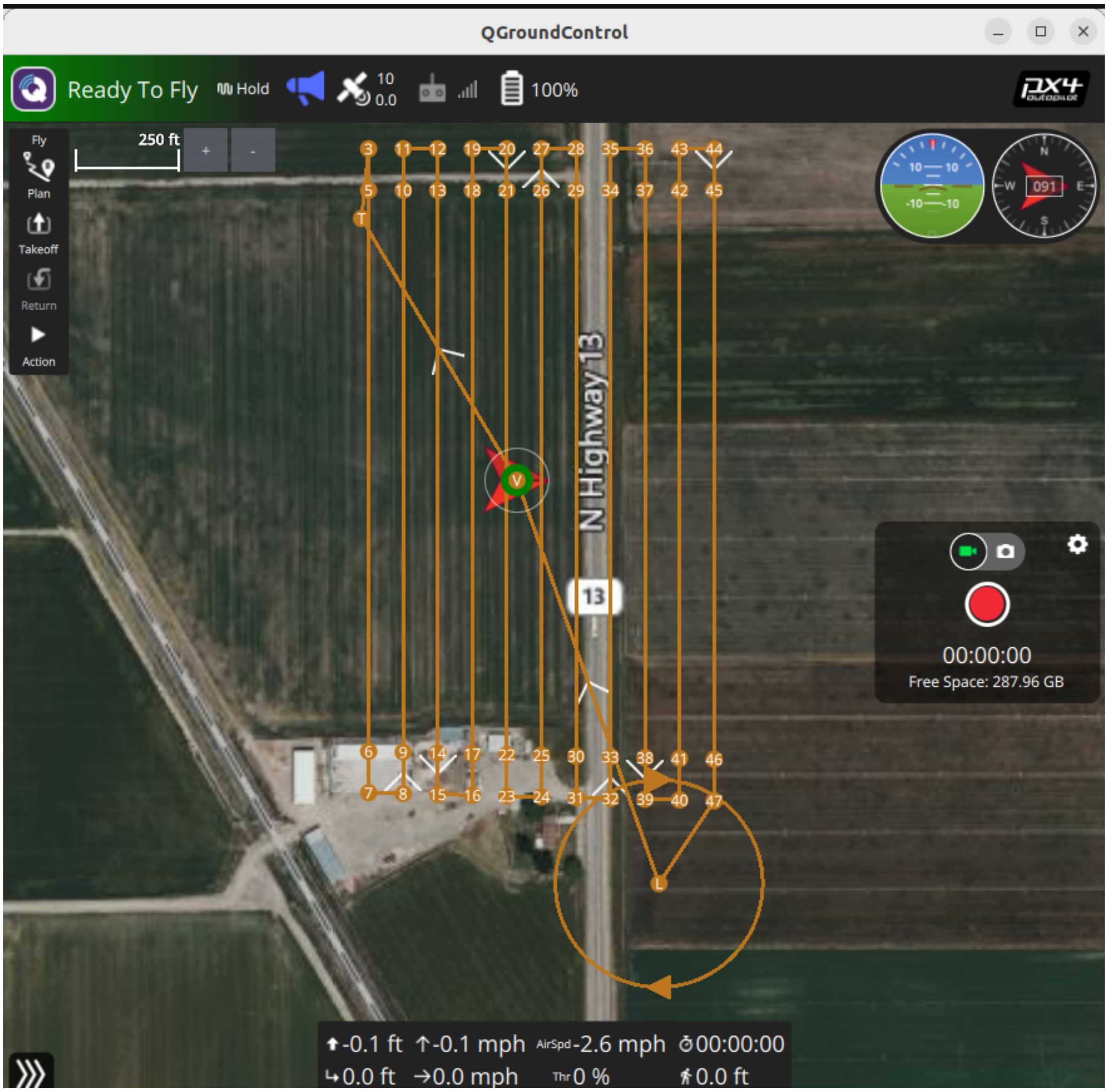
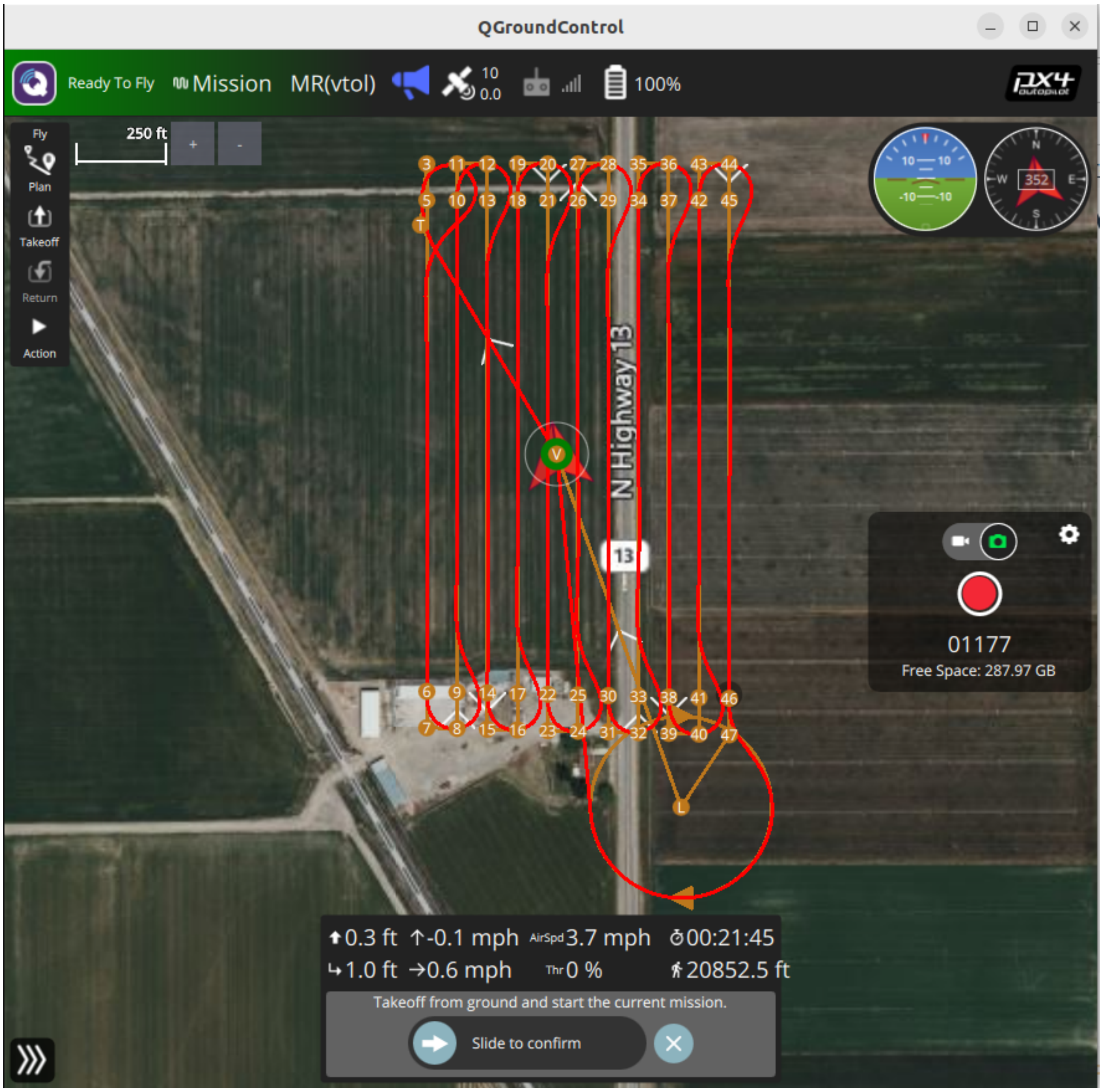


Fig. 4.7: The flight plan for the simulated flight in the Gazebo-Classic environment. The flight consists of: a quad rotor VTOL takeoff, a transition to horizontal flight, a transition back to quad rotor flight, and a VTOL landing.

The flight plan was successfully flown in the Gazebo-Classic simulation environment using the quad-plane camera airframe that was described in section 4.3.1 and shown in Fig.

4.1. Because the mission plan was done in horizontal flight the aircraft is unable to stop and turn right on each way-point, therefore, the aircraft has some overshoot due to it's limited turning radius. The actual flight lines are shown in red in Fig. 4.8.



0.3 ft -0.1 mph AirSpd 3.7 mph 00:21:45
1.0 ft 0.6 mph Thr 0 % 20852.5 ft

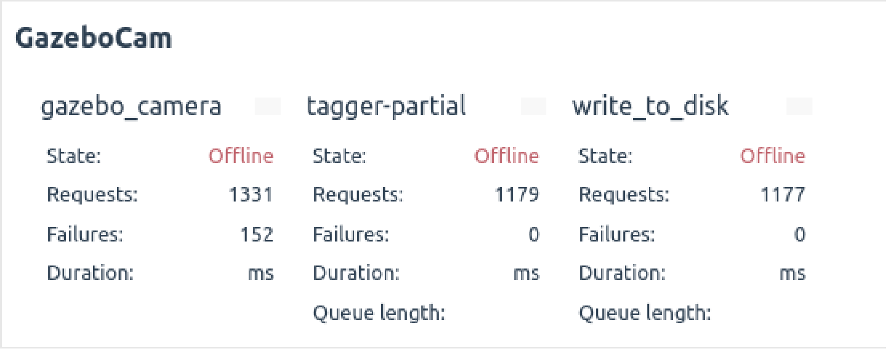
Takeoff from ground and start the current mission.

Slide to confirm

Fig. 4.8: The flight path actually flown by the simulated aircraft marked in red on top of the original flight plan marked in orange.

Image capture was started using a manually activated capture group. The manually activated capture group uses *Activate* and *Deactivate* buttons in the StarCommand user interface to indicate when data capturing should begin. Data capture then occurs at a fixed cadence that is specified in milliseconds (this is configured in the STARDOS configuration file using the StarCommand configuration editor). For this flight a cadence of 750 milliseconds was used, so an attempted capture was triggered by the capture group every 750 milliseconds. The *Activate* button was clicked just before way-point 3, and the *Deactivate* button was clicked just after hitting way-point 47.

At the end of the simulated flight the StarCommand output (pictured in Fig. 4.9) showed that the capture group had requested the simulated Gazebo Camera to take 1131 images. Of these 1131 requests 152 of them failed [2]. It was also noted that in the Tagger node there were 1179 data processing requests (this is expected as that 1331 minus 152 gives 1179) with 0 documented failures. The Write to Disk node received 1177 data processing requests and also 0 documented failures. The undocumented 2 image deficit between the Tagger node and the Write to Disk node is concerning and needs to be investigated further as no data capture failures or data processing request failures should go undocumented.



GazeboCam

gazebo_camera		tagger-partial		write_to_disk	
State:	Offline	State:	Offline	State:	Offline
Requests:	1331	Requests:	1179	Requests:	1177
Failures:	152	Failures:	0	Failures:	0
Duration:	ms	Duration:	ms	Duration:	ms
		Queue length:		Queue length:	

Fig. 4.9: The StarCommand summary showing the number of requested data captures and data capture failures for each node individually.

The images were then examined and it was found that they included appropriate meta-data information. This meta-data can be examined using the Linux command-line utility exiftool [84]. The output of the exiftool utility is provided:

[2] These failures are more than likely caused by inconsistencies in how quickly the simulated camera is able to produce data. Occasionally a new capture will be triggered when the previous data capture had not yet finished, this results in a failed capture. The root cause of this error still needs to be investigated.

```
ExifTool Version Number         : 12.40
File Name                       : DSC00500.jpg
Directory                       : .
File Size                       : 789 KiB
File Modification Date/Time     : 2023:04:15 11:31:15-06:00
File Access Date/Time           : 2023:04:15 11:49:03-06:00
File Inode Change Date/Time     : 2023:04:15 11:31:15-06:00
File Permissions                : -rw-rw-r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Exif Byte Order                 : Big-endian (Motorola, MM)
Subfile Type                    : Full-resolution image
Make                            : makename
Camera Model Name               : modelname
Orientation                     : Horizontal (normal)
X Resolution                    : 1
Y Resolution                    : 1
Resolution Unit                 : None
Software                        : USU AggieAir STARDOS
Modify Date                     : 2023:04:15 11:31:15
Y Cb Cr Positioning             : Centered
Exif Version                    : 2.30
Date/Time Original              : 2023:04:15 17:31:14
Components Configuration        : Y, Cb, Cr, -
Flashpix Version                : 0100
Color Space                     : Uncalibrated
```

```
GPS Version ID                  : 2.2.0.0
GPS Latitude Ref                : North
GPS Longitude Ref               : West
GPS Altitude Ref                : Above Sea Level
GPS Satellites                  : 13
GPS Measure Mode                : 3-Dimensional Measurement
GPS Dilution Of Precision       : 0.8
XMP Toolkit                     : XMP Core 4.4.0-Exiv2
Roll                            : -0.36909249424934387
Pitch                           : 0.06491302698850632
Yaw                             : -0.36090824007987976
Image Width                     : 3840
Image Height                    : 2160
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:2:0 (2 2)
Image Size                      : 3840x2160
Megapixels                      : 8.3
GPS Altitude                    : 517.5 m Above Sea Level
GPS Latitude                    : 41 deg 34' 13.90" N
GPS Longitude                   : 112 deg 7' 47.81" W
GPS Position                    : 41 deg 34' 13.90" N, 112 deg 7' 47.81" W
```

Some of the key features to note are the GPS altitude, latitude, and longitude as well as the aircraft attitude (roll, pitch, and yaw). These fields would not be present in an image that has not been tagged as is shown in section 5.4. Because this meta-data information was attatched to the images by the STARDOS Tagger node they can be more easily processed by a photo stitching (or photogrammetry) software as described in section 1.3 to create an orthomosaic.

Two example images from the data set are shown below in Fig. 4.10a and Fig. 4.10b. These images represent the re-sampled data set. Fig. 4.10a was taken while the aircraft was in level flight, the majority of the images in the data set are taken like this. Images that are taken during level flight are preferred as the image can be projected almost straight down on to the ground plane. Fig. 4.10b was taken while the aircraft was turning in flight. Images like these can still be used by the photogrammetry software because the tagger node includes the attitude of the aircraft, this allows the images to be projected at an angle down to the ground and into their appropriate location.

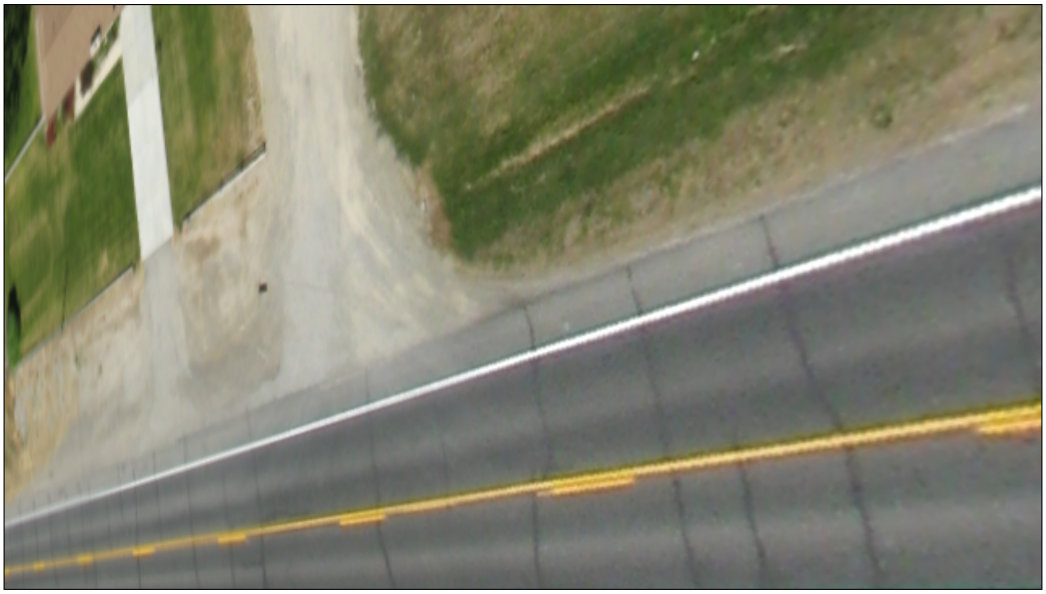
The images in the re-sampled data set were then put into the photogrammetry software Metashape [85]. The software initially showed the location that the images were taken, as indicated by the GPS data in the image meta-data. The output is shown in Fig. 4.11. It is important to see the similarity to the actual flight path of the aircraft, shown in Fig. 4.8. Each blue dot represents an image center, which indicates where the image was taken. This information is extracted from the GPS meta-data.

Using Metashape the re-sampled data set is turned into a re-sampled orthomosaic by aligning the images to make their features match up. This is shown in Fig. 4.12 where the images have been aligned. The black lines in the image are showing how each image was projected down onto the ground based on the attitude of the aircraft at that point, as well as how the features aligned in the photos.

A final output of the re-sampled Orthomosaic was generated using Metashape. In this there are a few places where the software was either unable to stitch the images, or the images were incorrectly placed. These artefacts mostly showed up around where the aircraft was turning. This indicates that there was insufficient overlap of images around those points. The final Orthomosaic generated by Metashape is shown in Fig. 4.13.

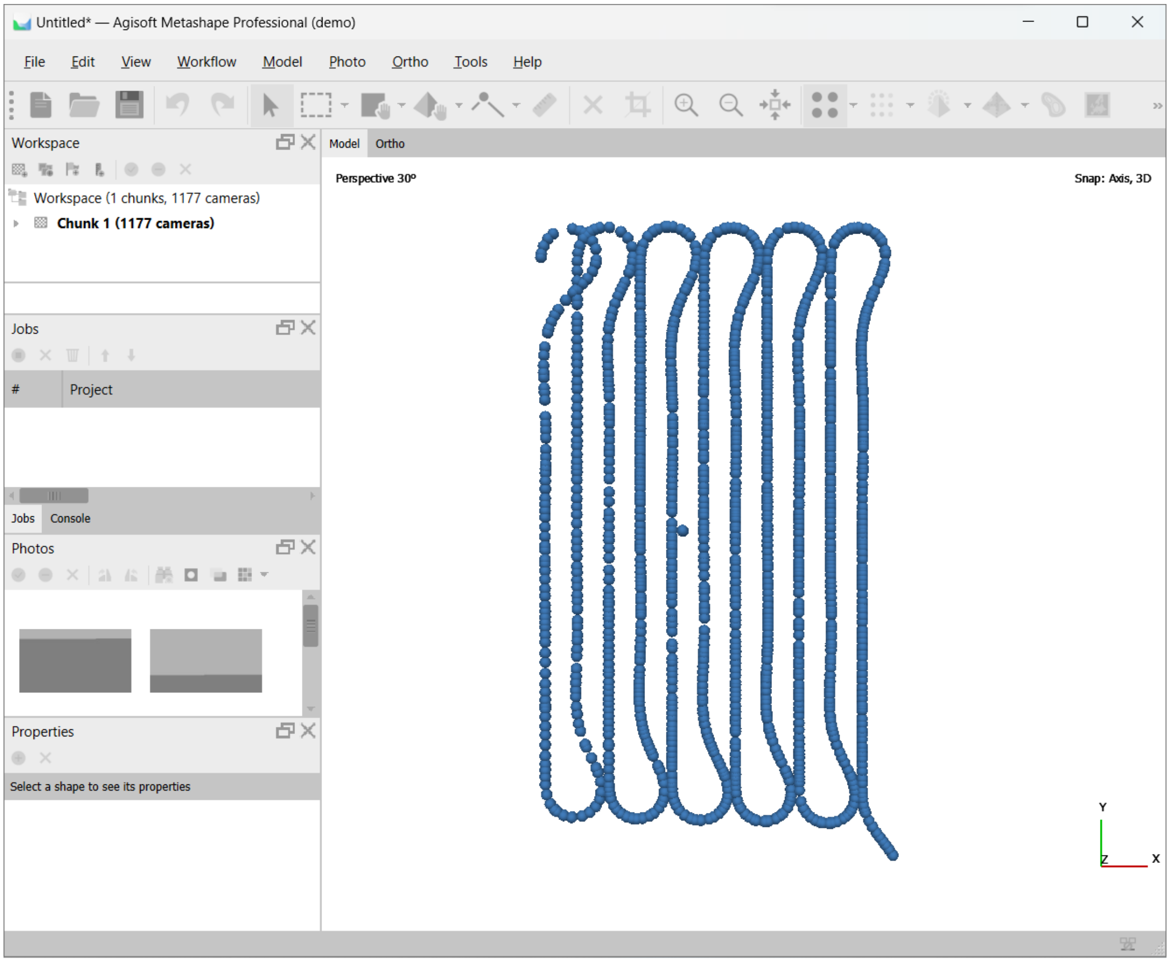
The Metashape software is able to generate a report that includes an analysis of the survey data. This report includes an image about the camera locations and error estimates as well as an image showing the image overlap. These images are shown alongside the final

(a) An example image that was taken using the simulated camera over the simulated environment while the aircraft was in level flight.



(b) An example image that was taken using the simulated camera over the simulated environment while the aircraft was turning in flight.

Fig. 4.10: Example images that are part of the re-sampled data set. These images were taken using the simulated camera over the simulated environment as shown in Fig. 4.2.



Workspace (1 chunks, 1177 cameras)

Chunk 1 (1177 cameras)

Select a shape to see its properties

Fig. 4.11: The photogrammetry software tool Metashape showing the locations of the image centers. This indicates where the images were taken with respect to one another.

orthomosaic in 4.14. The Metashape report shows that there were locations near some of the turns that were not imaged at all. This explains the difficulty of stitching and lack of image overlap in those areas.

The re-sampled orthomosaic can now be analyzed and compared against other data sets. Potentially more simulated test flights could be flown with varying flight patterns or by altering the camera slightly, then comparisons can be made to see how the data changes as individual variables are altered between flights. More importantly, real-time processing algorithms can be tested and proven in a simulation environment using STARDOS. In this example the images were collected and processed after the flight. STARDOS allows for

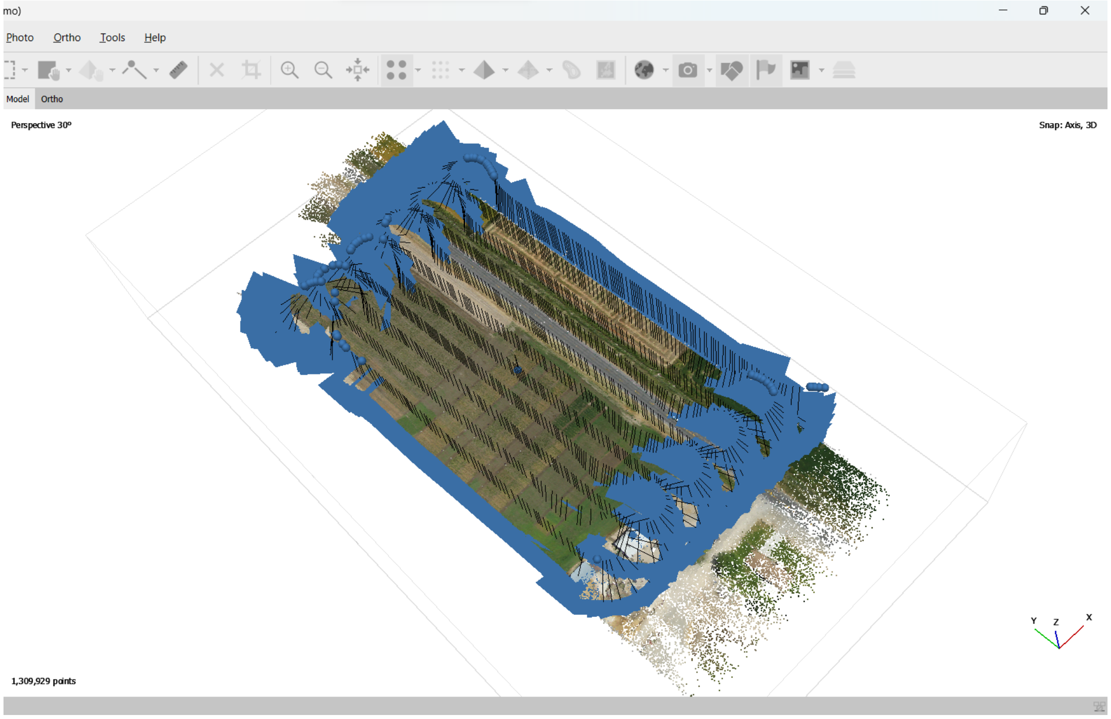


Fig. 4.12: The re-sampled orthomosaic output after the photos have been aligned using Metashape. The black lines and blue squares indicate the camera position and pose, based on the aircraft position, attitude, and the image alignment.The blue spheres represent the location that an image was taken of images that could not be aligned with the others.

varying processing nodes that process the data during the aircraft flight. This would allow users to perform a flight and have actionable data by the time that the aircraft lands. The cohesive simulation platform allows for this to be tested.

4.5 Summary

In section 4.1 the constraints of the project are introduced. In 4.2 the evaluation of various sUAS simulators were discussed and the decision to use the Gazebo Classic simulator was explained. In section 4.3 the development of the novel cohesive simulation platform for CAASO was presented. In section 4.4 the simulation platform for CAASO was demonstrated through a test flight using STARDOS and the image processing results were shown. This platform provides the ability to improve the process of collecting scientific quality data. Having the ability to simulate the entire data collection process is valuable.

This simulation platform allows users to go through that process and improve their scientific data collection without having to take on the risks that are associated with flying a real sUAS.

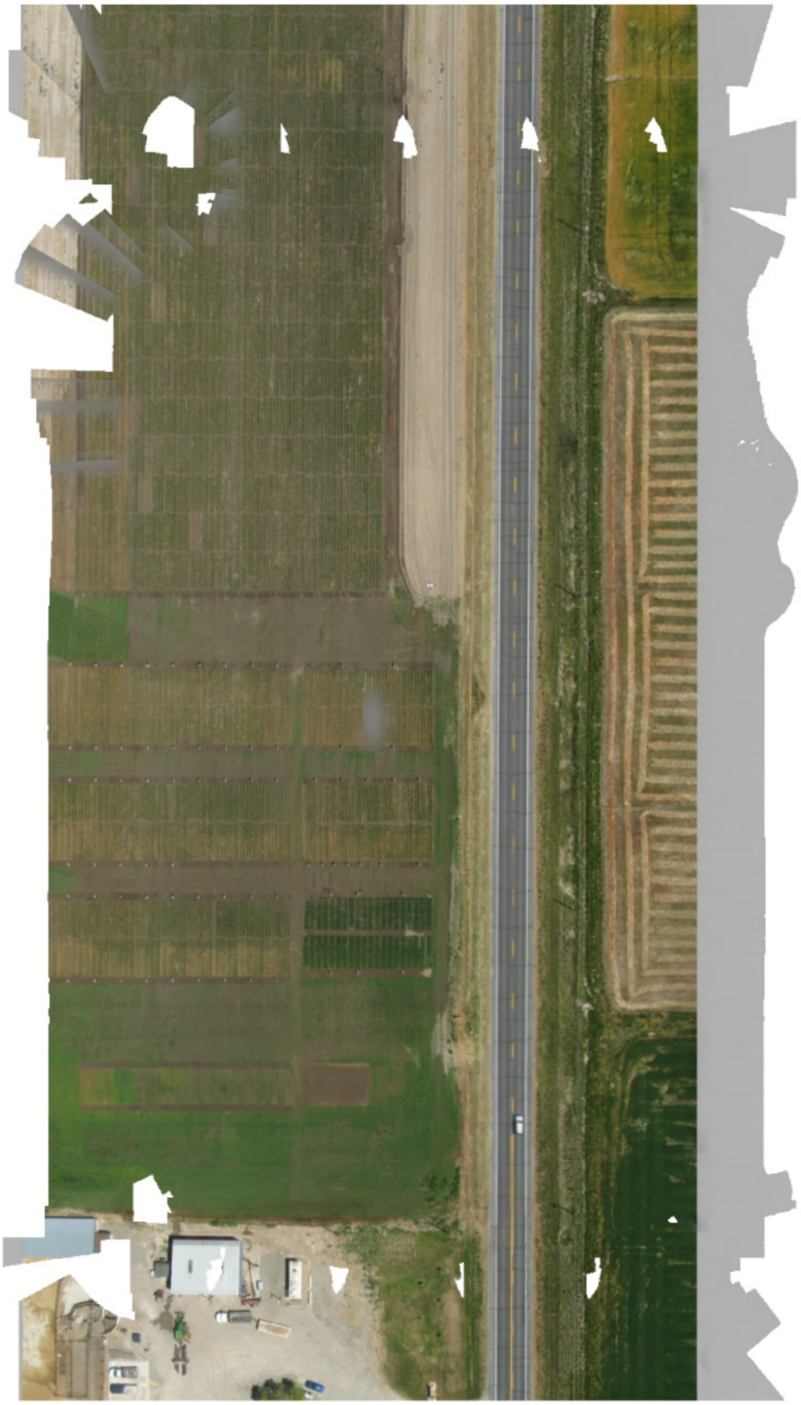


Fig. 4.13: The final orthomosaic output. This is the re-sampled orthomosaic, which was recreated completely from the images taken in the simulation environment by the simulated camera.

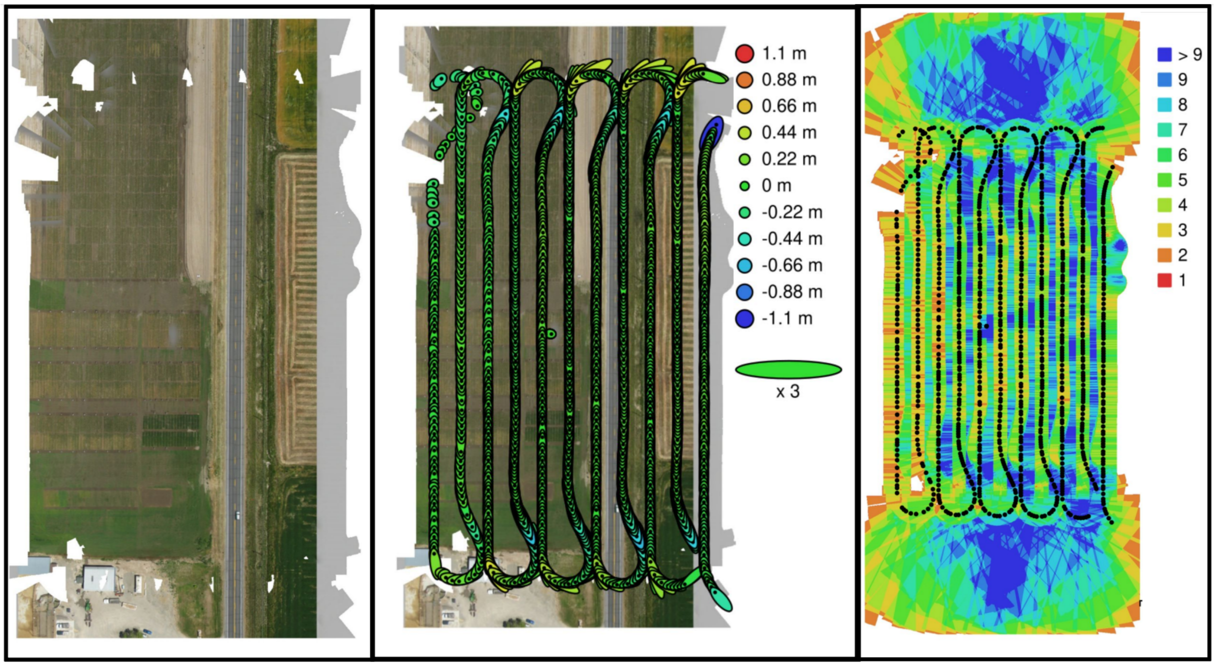


Fig. 4.14: The final orthomosaic (left) alongside the report generated analysis of the image location errors (middle) and the analysis of image overlap (right).

CHAPTER 5
A COHESIVE TESTING PLATFORM FOR CIVIL AUTONOMOUS AERIAL SENSING AND OPERATIONS

The simulation platform described in chapter 4 allows users to test out their sUAS data collection systems in a lab environment. This is done without having to take on any of the risks associated with a real sUAS flight (see section 3.1). While this simulation platform allows users to prove and test their systems, there truly exists no replacement for testing a system on an actual sUAS flight. There will always exist some difference between a simulation and the real world (especially when considering the actual hardware of an sUAS).

5.1 Testing Platform Motivation

This chapter presents a cohesive testing platform for CAASO. This platform provides a natural stepping stone for moving from testing in the simulation platform presented in chapter 4 to testing on larger sUAS systems. This stepping stone is the testing platform presented here.

With a desire to collect more scientific data, a larger sUAS platform needs to be used that has increased flight time and increased payload capacity. With greater flight time and payload capacity, however, often comes an increase in operational cost, an increase in operational overhead, and an increase in operational risk.

For example, AggieAir has recently published on the 55-lb sUAS system GreatBlue (shown in Fig. 3.1) that they have used for some of their data collection [16]. This system is very capable: nominal flight time of 2.5 hours, max payload weight of 11 pounds, an average cruising speed of 50 miles per hour, and an estimated max ground coverage (at an altitude of 2600 ft.) of 16000 acres.

The capability of the GreatBlue sUAS, however, requires that the system have a greater operational overhead cost associated with its use. The operation of the GreatBlue sUAS requires a minimum crew of two team members. The operational flight of a GreatBlue sUAS must be done by an FAA certified Part 107 [86] pilot that has received proper training on the GreatBlue sUAS system. The operational pre-mission and pre-flight preparation includes a more intense process to ensure that the aircraft is fully ready for flight.

In addition to these, and other, operational costs associated with the GreatBlue system, the payloads that are designed to be used in the operation of a GreatBlue sUAS are often more complex. One example of this is the AggieAir payload that is described by Coopmans et al. in “A 55-pound Vertical-Takeoff-and-Landing Fixed-Wing sUAS for Science: Systems, Payload, Safety Authorization, and High-Altitude Flight Performance” [16]. Another complex payload is described by Duncan et al. in “Integration and operation of a sUAS-based multi-modal imaging payload equipped with a novel communication system enabling autonomous platform control” [87]. Payloads such as these require their own preflight verification and have their own complexities. When attempting to test a new scientific data collection systems (such as AggieAir’s STARDOS) while also trying to integrate with these more complex payloads it takes more time and effort.

To overcome these challenges a cohesive testing platform for CAASO is presented in this chapter. This chapter describes how this testing platform was created and can be used. A simple, affordable, thermal mapping sUAS payload is described in section 5.2. The integration of this payload with AggieAir’s MiniBlue sUAS aircraft and the STARDOS system is described in section 5.3. Finally, the results of a test flight performed with the testing platform are presented in 5.4.

5.2 Simple, Affordable, Thermal mapping sUAS Payload

Thermal imagery collected from a drone has many use cases as described in section 3.3. However, there are a few challenges associated with using an sUAS to collect thermal imagery. Thermal cameras are more often used in complex sUAS payloads, similar to the ones mentioned in section 5.1. Additionally, high end thermal cameras can cost thousands of dollars [69] [70].

This section presents a payload for the testing platform for CAASO. The payload is made from simple commercial off the shelf components, so that it can be easily duplicated by others. Additionally, it is affordable (affordable in this research is defined as costing less than $500 USD.) making it more accessible for others to use and recreate. Furthermore, it has both a thermal camera and an RGB camera. For these reasons the payload has been titled the simple, affordable, thermal mapping sUAS payload (also referred to in this work as the payload or the thermal payload).

New thermal camera developments has made smaller and lower-cost thermal technology available. Both high quality higher cost and low quality lower cost thermal cameras have their own unique use cases. The Flir Lepton 3.5 thermal camera used in this research is a radiometric thermal imager with an accuracy (in low gain mode) of $\pm 10°C$ (see [88]) according to the data-sheet. In comparison, the higher quality thermal imagers used in the AggieAir Payload described in [16] and [17] (Infrared Cameras Incorporated Thermal Infrared 9640 P) have a radiometric accuracy of $\pm 0.2°C$ according to the data-sheet.

Higher quality cameras are valuable for the deep scientific insights that they can provide. They yield a higher resolution and more accurate data sets. This allows scientists to draw more accurate conclusions and therefore suggest more informed actions based on the data provided. Lower quality thermal cameras produce lower quality data, but they can be used in use cases where the quality is of less concern (in some cases low quality data is better than no data) or potentially where the risk of losing the payload is high. This new, more affordable thermal camera technology was taken advantage of in the development of the simple, affordable, thermal mapping sUAS payload.

To create the thermal mapping payload the FLIR Lepton 3.5 Thermal Camera ($164.00 USD) [89] paired with a micro-USB board ($109.00 USD) [90] is used as the affordable thermal sensor. This is paired with a Raspberry Pi 4 ($55.00 USD) [91] as the payload computer. Furthermore, a high quality Raspberry Pi Camera for the RGB data ($50.00 USD) [92] is paired with an appropriate camera lens ($50.00 USD) [93][1]. A summary of

[1] This specific camera lens was not used in this work, because a different lens was already available in AggieAir's lab, but using this lens is sufficient to make the payload.

these parts and their cost is shown in table 5.1.

On top of the cost of these individual components, there are small additional costs for mounting the payload hardware to the drone and for connecting cables to connect the various components for needed data and power. A system diagram showing the hardware components and their connections is shown in Fig. 5.1. Pictures of the fully assembled simple, affordable, thermal mapping payload are shown in Fig. 5.2a and Fig. 5.2b.

Table 5.1: A cost breakdown of the key components used to make the simple, affordable, thermal mapping payload.

Component	**Cost**
Flir Lepton 3.5	$164.00 USD
Flir Lepton Micro-USB Board	$109.00 USD
Raspberry Pi (4 GB version)	$55.00 USD
High Quality Raspberry Pi Camera	$50.00 USD
Lens for the Raspberry Pi Camera	$50.00 USD
Total Cost	**$429.00**

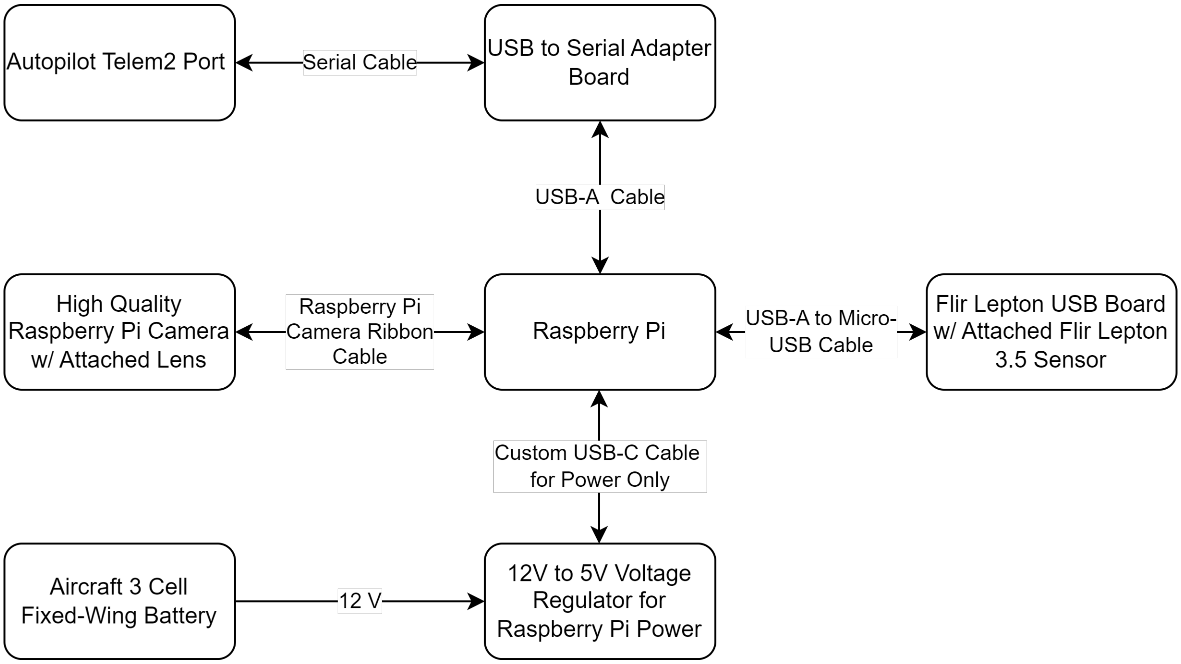

Autopilot Telem2 Port
Serial Cable
USB to Serial Adapter Board
USB-A Cable
High Quality Raspberry Pi Camera w/ Attached Lens
Raspberry Pi Camera Ribbon Cable
Raspberry Pi
USB-A to Micro-USB Cable
Flir Lepton USB Board w/ Attached Flir Lepton 3.5 Sensor
Custom USB-C Cable for Power Only
Aircraft 3 Cell Fixed-Wing Battery
12 V
12V to 5V Voltage Regulator for Raspberry Pi Power

Fig. 5.1: A system diagram of the simple, affordable, thermal mapping sUAS payload hardware setup.

When using this payload the user can fly an sUAS and test out various systems and algorithms without the fear of losing an expensive thermal camera or expensive payload system. This payload can be flown when low quality data is better than no data, or when testing with an experimental aircraft that may have a high risk of crashing. Additionally, if flying in an emergency situation or near hazardous materials the user can dispose of the aircraft if necessary and not incur a significant financial loss.

5.3 Testing Platform Development

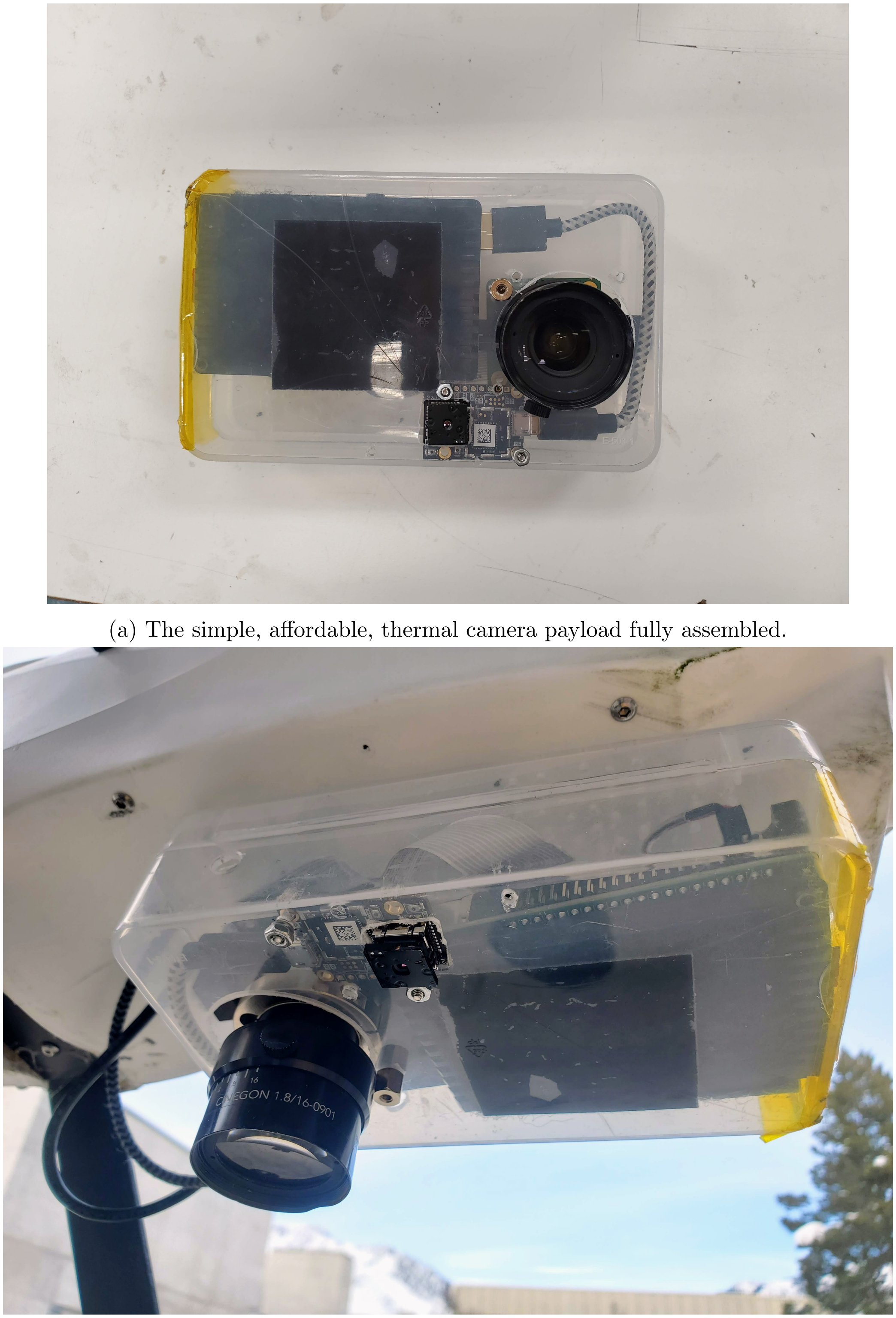
This section describes how the simple, affordable, thermal mapping sUAS payload was combined with AggieAir's MiniBlue [16] aircraft and the PX4 Hardware in the Loop simulation to create the cohesive testing platform for CAASO.

The objective of the testing platform is to provide an opportunity for testing out new sUAS payload systems (such as AggieAir's STARDOS), without having to risk larger and more expensive sUAS systems and payloads. This provides the benefits mentioned in the first paragraph of chapter 5 and in section 5.1.

AggieAir designed the GreatBlue platform, as mentioned in sec 1.3.2. In order to test out the system design the MiniBlue platform was created. AggieAir's MiniBlue (shown in Fig. 5.3) platform is a smaller quad-plane VTOL aircraft with the main material being styrofoam. MiniBlue has a lower payload capacity and reduced flight time in comparison to GreatBlue. These reductions, however, are an advantage as the aircraft is easier to use, safer (as it weights less), and when it is crashed it is easier to repair because of the materials used in its construction.

As described in [16] the MiniBlue platform "is a full-system trainer for GreatBlue. Wholly similar in some aspects (a transition VTOL aircraft with two power supplies and a combined payload computer/safety copilot), MiniBlue is only 6 pounds GTOW (Gross Takeoff Weight), and costs less than $1400" [16]. This simpler and more affordable sUAS aircraft is paired naturally with the simple, affordable, thermal sUAS payload.

Additionally, in preparation for an sUAS flight with the cohesive testing platform the simulation platform described in the previous chapter can be applied. Furthermore,



(a) The simple, affordable, thermal camera payload fully assembled.

(b) The simple, affordable, thermal camera payload mounted to AggieAir's MiniBlue Aircraft.

Fig. 5.2: Images of the simple, affordable, thermal mapping sUAS payload.

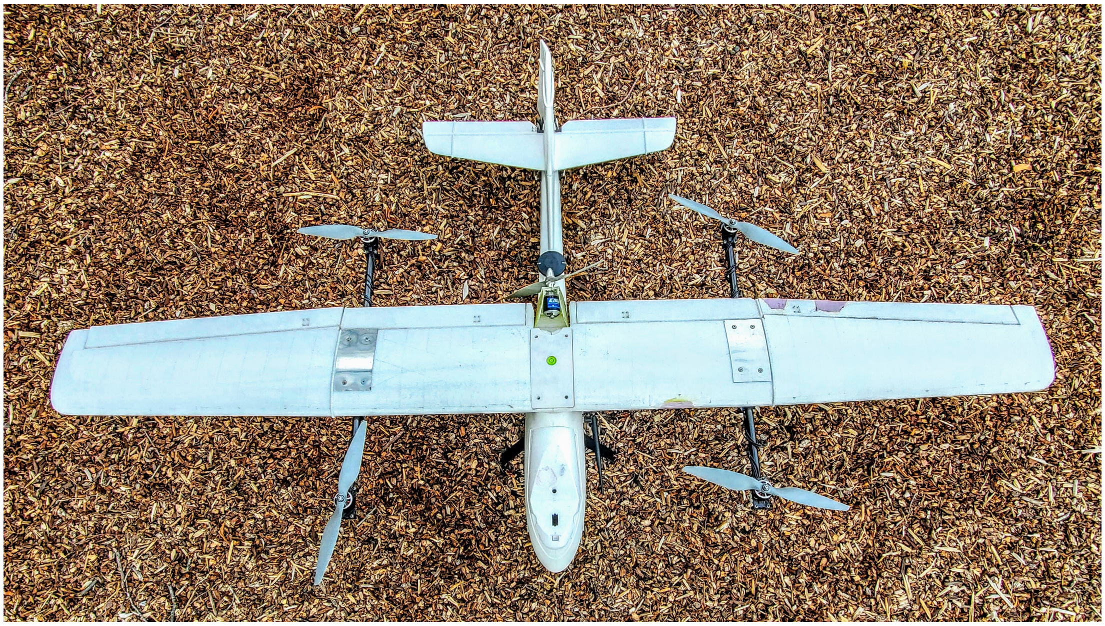


Fig. 5.3: AggieAir's MiniBlue platform, a quad-plane VTOL aircraft.

hardware simulation testing can also be done using PX4's Hardware in the Loop (HITL) setup as described in the PX4 documentation [94]. The HITL simulation can be run allowing the user to use a hardware autopilot and a hardware payload. This allows for simulation using actual hardware to occur before an actual sUAS data flight occurs.

The HITL setup is a key piece of the cohesive testing platform as it provides a bridge between the cohesive simulation platform (where everything is performed in simulation) and a real sUAS flight (where there is no simulation and actual autopilot and payload hardware are used). The HITL setup tests using both simulation, by simulating a flying aircraft, and hardware, by using a hardware autopilot and a hardware payload. An ideal simulation and testing workflow would follow the flow illustrated in Fig. 5.4.

The combination of the simple affordable thermal payload, AggieAir's MiniBlue platform, and the use of the PX4 HITL setup creates the cohesive testing platform for CAASO.

5.4 Testing Platform Flight and Results

In this section an example test flight using the combination of AggieAir's MiniBlue and the simple, affordable, thermal mapping sUAS payload is presented. This flight with the

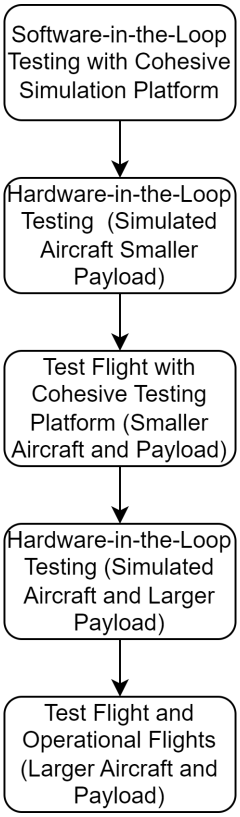

Software-in-the-Loop Testing with Cohesive Simulation Platform
Hardware-in-the-Loop Testing (Simulated Aircraft Smaller Payload)
Test Flight with Cohesive Testing Platform (Smaller Aircraft and Payload)
Hardware-in-the-Loop Testing (Simulated Aircraft and Larger Payload)
Test Flight and Operational Flights (Larger Aircraft and Payload)

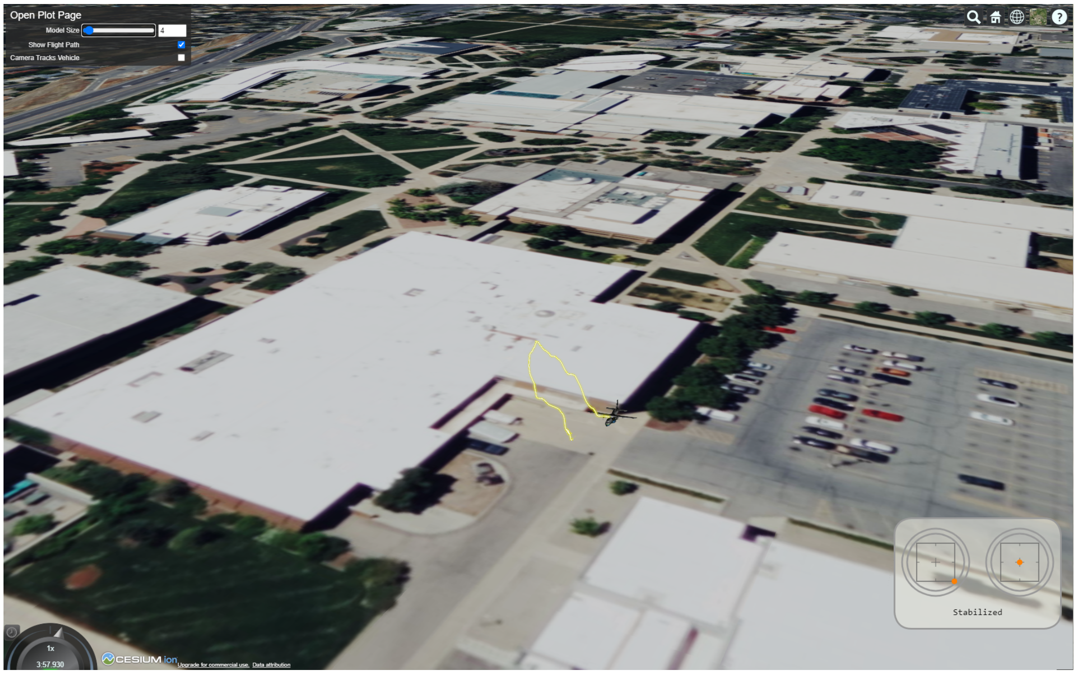
Fig. 5.4: A flowchart showing the stepping stones of simulation and testing that should be done when preparing for data collection flights. Various parts of testing and simulation can be repeated as necessary for debugging and improving the data collection system.

payload demonstrates the use of the cohesive testing platform. The results from the flight are analyzed and improvements are suggested based on the flight test results.

The test flight was performed on April 25th, 2023 at approximately 4:00pm. Only a VTOL flight was performed, due to weather, using the quad-rotor motor system. The horizontal flight capability of the MiniBlue platform was not tested during this flight, so this resulted in only a short data collection flight.

After the flight, the flight logs were recovered from the autopilot analyzed using PX4's log viewer web application [1]. The resulting flight path is shown in a screenshot of this log analysis tool in Fig. 5.5.

The flight location was on Utah State University campus in the parking lot next to the AggieAir lab (GPS coordinates of Flight 41.742904 N, 111.807116 W). A vertical takeoff



Open Plot Page

Model Size 4

Show Flight Path ☑

Camera Tracks Vehicle □

Fig. 5.5: The flight path of the test flight flown with the simple, affordable, thermal mapping sUAS payload. This information was pulled from the PX4 autopilot log and viewed in the PX4 flight log viewer [1].

was performed with the quad-rotor system, and a flight of approximately 2 minutes was flown manually by a safety pilot in the stabilized flight mode. The stabilized flight mode is described in the PX4 documentation: “The multicopter will level out and stop once the roll and pitch sticks are centered. The vehicle will then hover in place/maintain altitude - provided it is properly balanced, throttle is set appropriately (see below), and no external forces are applied (e.g. wind)” [95].

The simple, affordable, thermal mapping sUAS payload used a STARDOS configuration that consisted of 2 different manually activated capture groups. One capture group for triggering captures using the High Quality Raspberry Pi camera and one capture group for triggering captures at a different rate for the Flir Lepton sensor. Because it was desireable to have the lower quality Flir Lepton camera capture images at a different rate it needed to be in an independent capture group from the Raspberry Pi camera. This feature, of taking images at different rates, was not tested in the test flight shown, but it was tested in a previous test flight.

The Raspberry Pi camera capture group triggered image captures with 400 milliseconds between captures (giving approximately 2.5 images per second). The images produced by the Raspberry Pi camera were passed to a STARDOS tagger node. Once these images had been tagged with the appropriate meta-data they were passed to a disk writer node which would write them to the disk for permanent storage. The Flir Lepton capture group was set up similar to the Raspberry Pi camera capture group. Images were captured with 400 milliseconds between captures. These images were passed to a tagger node which then passed the tagged images to a disk writer node. A diagram depicting how the payload was configured is shown in Fig. 5.6.

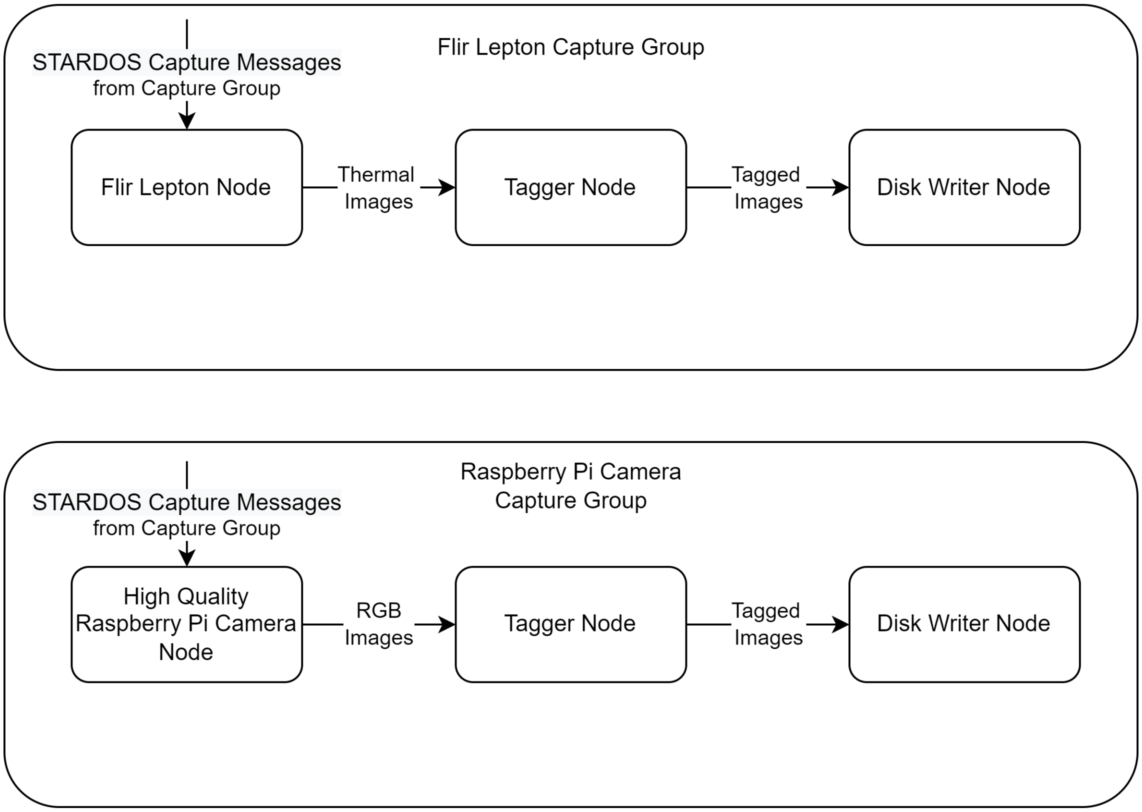


Fig. 5.6: A diagram depicting the capture group and node layout used in the test flight of the simple, affordable, thermal mapping sUAS payload on AggieAir's MiniBlue platform.

Before the flight numerous tests were done using the PX4 hardware in the loop simulation setup with the same payload and STARDOS configuration setup as would be used in actual flight.

In a previous test flight a few issues with the payload setup had been identified:

1. The capture rate for the thermal images (9 images per second) produced images faster than the STARDOS tagger node was able to process them. At the end of the flight the payload was stopped before all of these images could be tagged with the appropriate meta-data. This left about 347 images unprocessed.

2. After looking at the images generated during the flight it was determined that image capture stopped part way through the flight. This left a significant portion of the flight without captured data.

3. Many of the images seemed to be tagged with the same meta-data tags instead of with newer updated information. On certain groups of images, the GPS tags were not updated to a new location.

4. The Raspberry Pi Camera had too great of exposure, so much so that most of the images appeared to be white.

After the first flight and with more testing on the HITL setup, these issues were resolved. Several of them were related to capturing the thermal images too quickly, causing CPU overloading. For this reason, the second flight reduced the thermal image capture rate to the same rate as the raspberry pi camera capture group, instead of the 9 frames per second that had be captured previously. Additionally, adjustments were made to correct the over exposure on the raspberry pi camera images.

Because these issues were able to be resolved this shows the utility of the cohesive testing platform. An initial data collection flight using an sUAS was used to test out and push the limits of the STARDOS architecture. This lead to newly discovered issues that may not have been come up if it had only been tested in simulation. After this data collection flight the logs could be analyzed to determine root causes of the problems so that the STARDOS architecture could be improved and used in future tests and eventually for scientific data collection flights on larger platforms such as AggieAir's GreatBlue with larger and more complex payloads.

After resolving these issues, a second scientific data collection flight was flown. A data set was collected from the successful flight. During the flight approximately 200 thermal images were taken by the FLIR Lepton and just under over 200 RGB images were taken by the Raspberry Pi camera. An example of the meta-data that was attached by the STARDOS tagger node is shown here as output by the Linux command line utility *exiftool* [84]:

```
ExifTool Version Number         : 12.60
File Name                       : FlirLepton_000100.tif
Directory                       : .
File Size                       : 45 kB
File Modification Date/Time     : 2023:04:25 16:28:38-06:00
File Access Date/Time           : 2023:04:25 17:08:57-06:00
File Inode Change Date/Time     : 2023:04:25 17:08:57-06:00
File Permissions                : -rw-r--r--
File Type                       : TIFF
File Type Extension             : tif
MIME Type                       : image/tiff
Exif Byte Order                 : Little-endian (Intel, II)
Image Width                     : 160
Image Height                    : 120
Bits Per Sample                 : 8 8 8
Compression                     : LZW
Photometric Interpretation      : RGB
Make                            : makename
Camera Model Name               : modelname
Strip Offsets                   : (Binary data 45 bytes, use -b option to extract)
Orientation                     : Horizontal (normal)
```

```
Samples Per Pixel               : 3
Rows Per Strip                  : 17
Strip Byte Counts               : (Binary data 38 bytes, use -b option to extract)
Planar Configuration            : Chunky
Software                        : USU AggieAir STARDOS
Modify Date                     : 2023:04:25 16:28:38
Predictor                       : Horizontal differencing
Sample Format                   : Unsigned; Unsigned; Unsigned
XMP Toolkit                     : XMP Core 4.4.0-Exiv2
Roll                            : 0.007716231979429722
Pitch                           : -0.005419141612946987
Yaw                             : 1.7308346033096313
Exif Version                    : 2.30
GPS Version ID                  : 2.2.0.0
GPS Latitude Ref                : North
GPS Longitude Ref               : West
GPS Altitude Ref                : Above Sea Level
Image Size                      : 160x120
Megapixels                      : 0.019
GPS Altitude                    : 1477.4 m Above Sea Level
GPS Latitude                    : 41 deg 44’ 34.32" N
GPS Longitude                   : 111 deg 48’ 25.55" W
```

Some key parts of this meta-data include the GPS altitude, latitude, and longitude in addition to the attitude (roll, pitch and yaw) information of the aircraft. In contrast if we look at the default meta-data of an image, from a previous test flight, that did not get processed by the STARDOS tagger node we see the following:

```
ExifTool Version Number         : 12.50
File Name                       : FlirLepton_002000.tif
```

```
Directory                       : .
File Size                       : 47 kB
File Modification Date/Time     : 2023:04:07 10:37:44-06:00
File Access Date/Time           : 2023:04:07 10:37:44-06:00
File Inode Change Date/Time     : 2023:04:07 10:37:44-06:00
File Permissions                : -rw-r--r--
File Type                       : TIFF
File Type Extension             : tif
MIME Type                       : image/tiff
Exif Byte Order                 : Little-endian (Intel, II)
Image Width                     : 160
Image Height                    : 120
Bits Per Sample                 : 8 8 8
Compression                     : LZW
Photometric Interpretation      : RGB
Strip Offsets                   : (Binary data 42 bytes, use -b option to extract)
Samples Per Pixel               : 3
Rows Per Strip                  : 17
Strip Byte Counts               : (Binary data 38 bytes, use -b option to extract)
Planar Configuration            : Chunky
Predictor                       : Horizontal differencing
Sample Format                   : Unsigned; Unsigned; Unsigned
Image Size                      : 160x120
Megapixels                      : 0.019
```

The absence of the GPS and attitude information is noted as well as well as many other fields. This meta-data information needs to be added on to the images in order for them to be successfully processed by an image stitching software.

Examining the images we see the results of the test flight with the simple, affordable thermal mapping sUAS payload. Samples of these images are provided in Fig. 5.7, Fig. 5.8, Fig. 5.9a, Fig. 5.9b, Fig. 5.10a, and Fig. 5.10b.

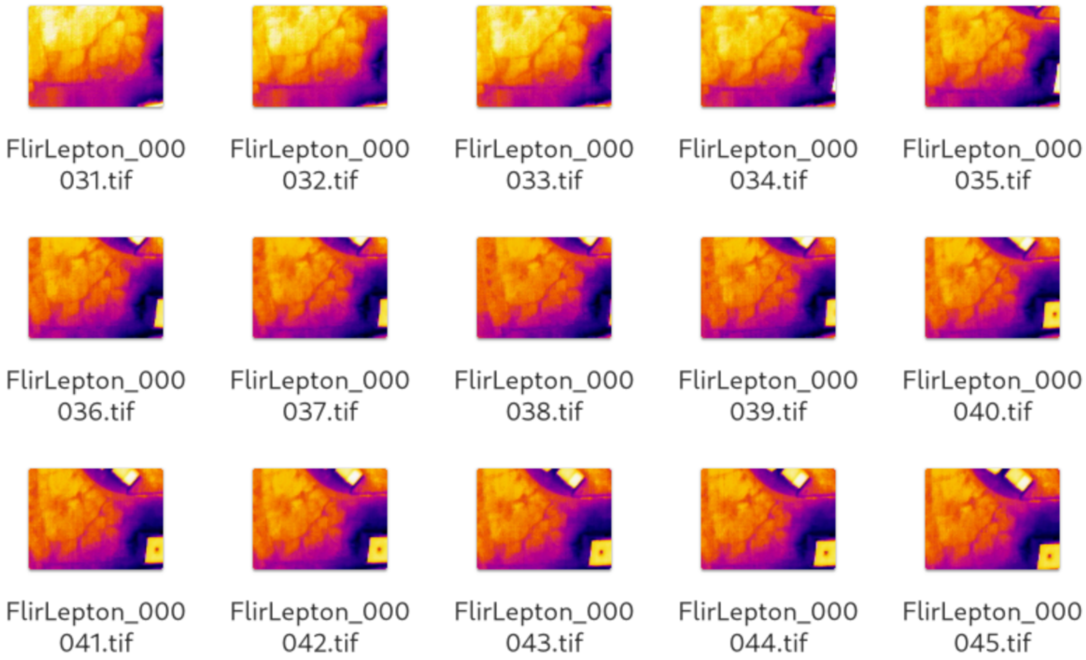


Fig. 5.7: A sample of the resulting thermal images from the test flight with the simple, affordable, thermal sUAS mapping payload.

This test flight, overall, proved valuable in testing and proving the design and implementation of the STARDOS architecture. Without an initial hardware flight test the issues identified may not have become readily apparent. The ability to fix the issues and easily perform a second test flight with the fixes implemented demonstrates the utility of having a testing platform as it allows for incremental testing and improvements in a scientific data collection system.

5.5 Summary

In this chapter a cohesive testing platform for CAASO is presented. In section 5.1 appropriate motivation is given demonstrating the need for a cohesive testing platform. In section 5.2 the simple, affordable, thermal mapping sUAS payload is presented, providing

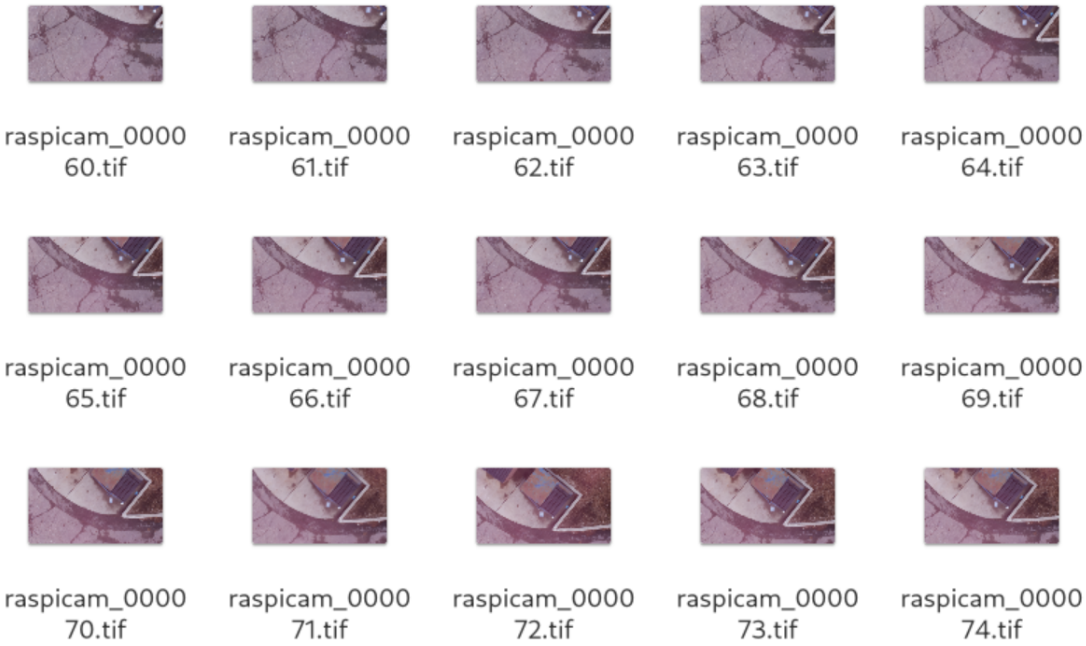
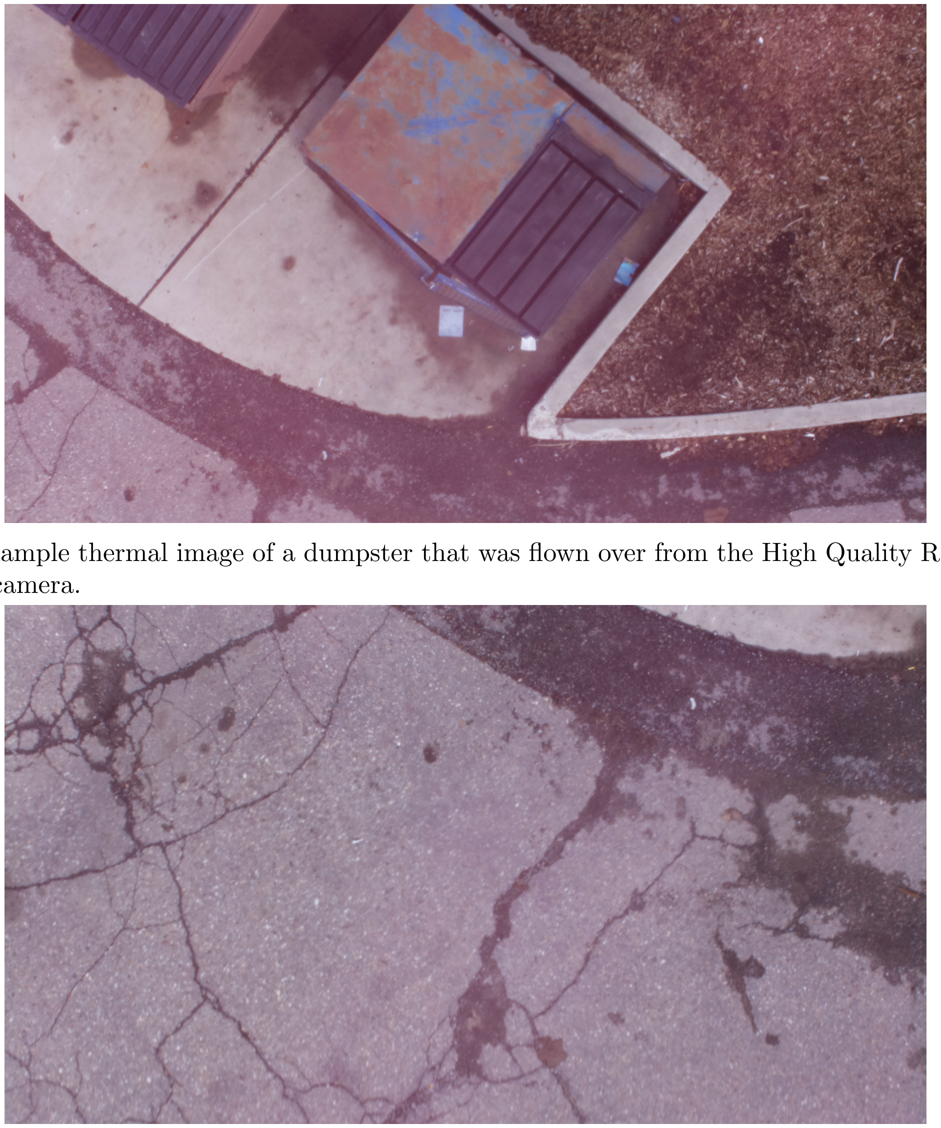


Fig. 5.8: A sample of the resulting RGB images from the test flight with the simple, affordable, thermal sUAS mapping payload.

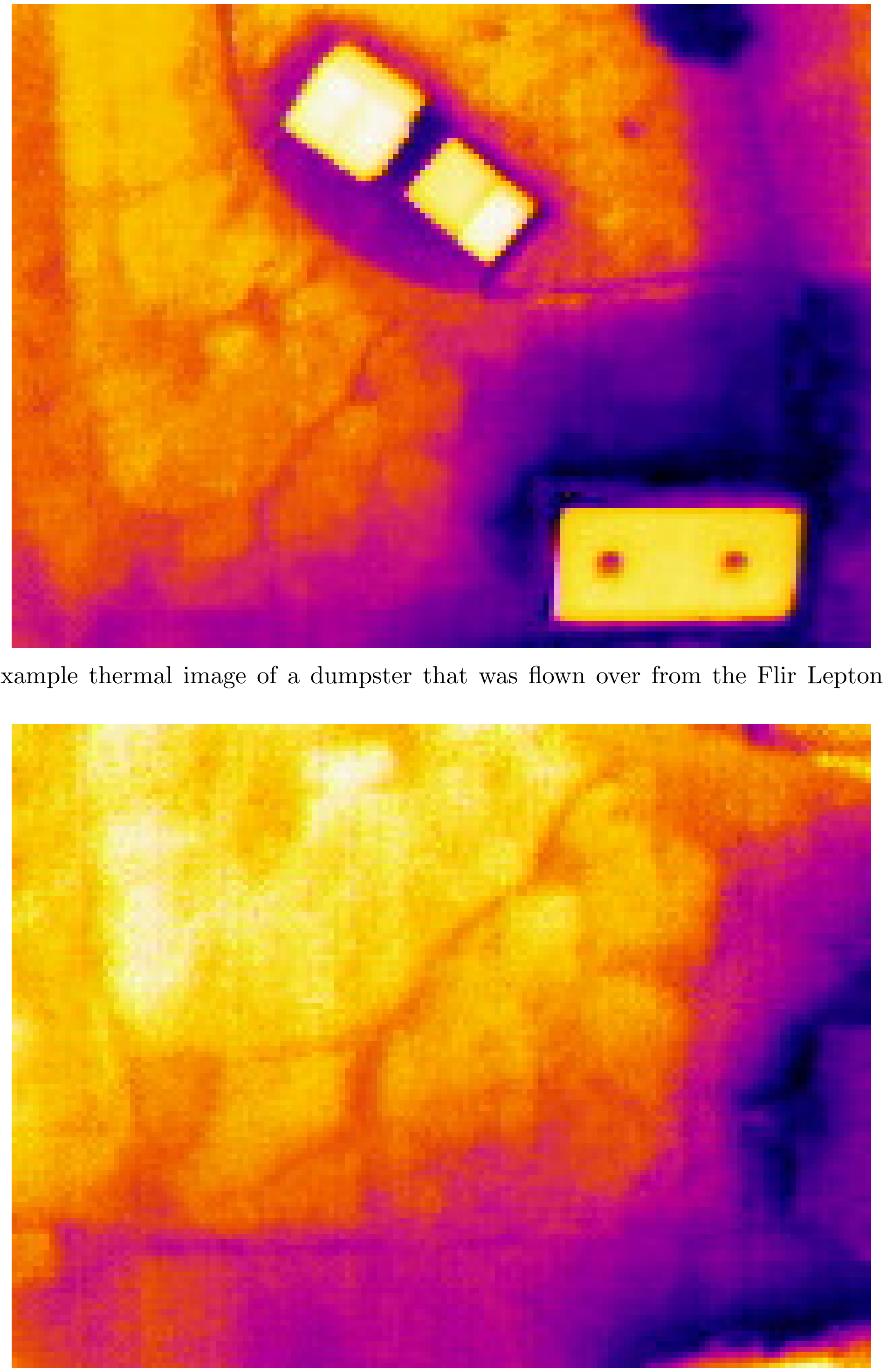
a solid payload foundation for the cohesive testing platform. Next, in section 5.3, the system integration is described of the simple, affordable, thermal mapping sUAS payload with AggieAir's STARDOS architecture. Finally, in section 5.4 the results from a successful data collection flight using the testing platform for CAASO are presented.



(a) An example thermal image of a dumpster that was flown over from the High Quality Raspberry Pi RGB camera.

(b) An example thermal image of some cracks in the pavement as captured by the High Quality Raspberry Pi RGB Camera.

Fig. 5.9: RGB images from the test flight.



(a) An example thermal image of a dumpster that was flown over from the Flir Lepton thermal camera.

(b) An example thermal image of some cracks in the pavement as captured by the Flir Lepton thermal camera.

Fig. 5.10: Thermal images from the test flight.

CHAPTER 6
CONCLUSION

This work presents a cohesive simulation and testing platform for civil autonomous aerial sensing and operations. In chapter 1 the motivation for this research work is presented along with detailed background information. In chapter 2 a summary of AggieAir's STARDOS architecture is presented. In chapter 3 related and relevant previous work is explored providing a foundation on which the research presented in this work could build upon. In chapter 4 a novel cohesive simulation platform is presented and information is given about how it can be used for the development and testing of sUAS data capturing systems. In chapter 5 a novel cohesive testing platform is presented and a test flight and the results from this test flight are presented.

AggieAir's STARDOS architecture was described in chapter 2. Although the development of AggieAir's STARDOS architecture was not the focus of this research, STARDOS and the cohesive simulation and testing platform have played motivating roles in the development of one another. Without the cohesive simulation and testing platform AggieAir's STARDOS would have taken much more time to develop. Having a simulation and testing platform allowed for many key parts of AggieAir's STARDOS to be tested thoroughly in the lab and on the testing platform as needed.

Additionally, the development of STARDOS motivated the development of the cohesive simulation and testing platform because of the key role that testing plays throughout the development of sUAS systems.

Furthermore, A cohesive simulation environment allows a large variety of testing to be done without the risks associated with an actual sUAS flight. Using the cohesive simulation platform avoids the challenges such as airspace access, sUAS flight worthiness, and payload management. The objective of this research was aligned with one of AggieAir's objectives: to improve the process by which scientific quality data can be collected. This was achieved by providing a cohesive simulation and testing platform for CAASO.

This platform creates a natural progression for moving from a development simulation environment, to a HITL setup for testing hardware and software systems, to actual test flights using the cohesive testing platform for testing almost all of the same factors (hardware and software) that will be present in a larger sUAS system, and finally to fully functional flight operations.

Future work includes:

- Determining which quality metrics are most valuable when making comparisons between re-sampled orthomosaic data
- Performing multiple simulation test flights with the simulation platform and comparing these results to improve real sUAS data collection
- Testing faster simulation speeds with the cohesive simulation platform and Gazebo Classic
- Scaling the Gazebo simulation environment such that the features shown are appropriately to scale
- Adding digital elevation models to the Gazebo simulation platform
- Implementing methods by which the cohesive simulation and testing platform could be used for crew training and mission preparation

The work described in chapters 4 and 5 illustrate the development of the cohesive simulation testing platform for civil autonomous aerial sensing and operations. This work focused on improving the process by which scientific data can be collected using an sUAS through the cohesive simulation and testing platform for CAASO by providing natural stepping stones for simulating and testing scientific data collection systems.

REFERENCES

[1] “Flight Review.” [Online]. Available: https://review.px4.io/

[2] M. Hassanalian, Mostafa Hassanalian, Mostafa Hassanalian, and A. Abdelkefi, “Classifications, applications, and design challenges of drones: A review,” *Progress in Aerospace Sciences*, vol. 91, pp. 99–131, May 2017, mAG ID: 2610728744 S2ID: 52a749aacfc4de71ba3646f99a97d990eb6c0faf.

[3] Mohd. Yamani Yahya, Wong Pao Shun, A. Md. Yassin, and R. Omar, “The Challenges of Drone Application in the Construction Industry,” *Journal of Technology Management and Business*, 2021, s2ID: 474320c5d5460ef2a79a58f764c534d5ab4f9713.

[4] D. Goldberg, M. Corcoran, and R. G. Picard, “Remotely piloted aircraft systems and journalism: opportunities and challenges of drones in news gathering,” Jan. 2013, mAG ID: 2899518757 S2ID: 8598af4989709cf120696d1a131951017bb0a3a4.

[5] M. Sibanda, O. Mutanga, V. Chimonyo, A. D. Clulow, C. Shoko, D. Mazvimavi, T. Dube, and T. Mabhaudhi, “Application of Drone Technologies in Surface Water Resources Monitoring and Assessment: A Systematic Review of Progress, Challenges, and Opportunities in the Global South,” *Drones*, vol. 5, no. 3, p. 84, Aug. 2021, mAG ID: 3198625367 S2ID: 12bd80c03e25843ac5cd6a3ef2ed5f3c7d3ac1c4.

[6] Julien Berthe, Julien Berthe, J. Berthe, F. Coussa, F. Bermond, and P. Beillas, “Drone impact on human beings : Experimental investigation with sUAS,” Jun. 2019, mAG ID: 2984185071 S2ID: 1c36a98c7de7cf1ba65b738ae2f6b8e6210cab52.

[7] H. Chao, A. M. Jensen, Y. Han, Y. Chen, and M. McKee, “AggieAir: Towards Low-cost Cooperative Multispectral Remote Sensing Using Small Unmanned Aircraft Systems,” pp. 463–490, Oct. 2009, mAG ID: 1595892362 S2ID: 82768a841d091599c548a6f9f9fe9fc9b0d5290b.

[8] A. M. Jensen, M. McKee, and Y. Chen, “Calibrating thermal imagery from an unmanned aerial system - AggieAir,” *IEEE International Geoscience and Remote Sensing Symposium*, pp. 542–545, Jul. 2013, mAG ID: 1998763242 S2ID: 467d70aaf8a3ea2b233168c418b4b98d9bf63bc9.

[9] M. Aboutalebi, A. F. Torres-Rua, N. Allen, and Niel Allen, “Multispectral Remote Sensing for Yield Estimation Using High-Resolution Imagery From an Unmanned Aerial Vehicle,” *Commercial + Scientific Sensing and Imaging*, vol. 10664, pp. 1–11, May 2018, mAG ID: 2804617345 S2ID: b90636845c86633142380cbb1fdec633733bd2cd.

[10] “AggieAir,” Mar. 2023. [Online]. Available: https://uwrl.usu.edu/aggieair/

[11] Anurag, K. R. Kota, and T. E. Lacy, “Development of a Frangible Design of Small Fixed-Wing Unmanned Aerial System,” *American Society for Composites 2021*, Jan. 2021, mAG ID: 3203938382 S2ID: a868089aa1f2f382f1c4098b465c500b76642fb3.

[12] A. E. Ahmed, A. A. A. Hafez, A. N. Ouda, A. N. Ouda, H. E. H. Ahmed, and H. M. Abd-Elkader, "Modeling of a Small Unmanned Aerial Vehicle," *World Academy of Science, Engineering and Technology, International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering*, vol. 9, no. 3, pp. 503–511, Mar. 2015, mAG ID: 1524353109 S2ID: ee96a99c51996dcb20eb8d364a2170a1c7d4912d.

[13] M. J. Sytsma, "Effects of Turbulence on Fixed Wing Small Unmanned Aerial Systems," Jan. 2013, mAG ID: 2289402773 S2ID: 286f446a21a59779e3612625071be264d1ad8b34.

[14] A. F. Torres-Rua, M. A. Arab, L. Hassan-Esfahani, A. M. Jensen, and M. McKee, "Development of unmanned aerial systems for use in precision agriculture: The AggieAir experience," *IEEE Conference on Technologies for Sustainability*, pp. 77–82, Nov. 2015, mAG ID: 1937290727 S2ID: aafd3fab5059bee3c3ad213df52512763e5156e5.

[15] M. Kalua, A. M. Rallings, Anna M. Rallings, L. Booth, J. Medellín-Azuara, S. Carpin, and J. H. Viers, "sUAS Remote Sensing of Vineyard Evapotranspiration Quantifies Spatiotemporal Uncertainty in Satellite-Borne ET Estimates," *Remote Sensing*, vol. 12, no. 19, p. 3251, 2020, mAG ID: 3092595343 S2ID: 8d4cd1cd13f5ed8746e4cb73525f4f82fb3cbcba.

[16] Calvin Coopmans, Stockton Slack, Daniel J. Robinson, and Nathan Schwemmer, "A 55-pound Vertical-Takeoff-and-Landing Fixed-Wing sUAS for Science: Systems, Payload, Safety Authorization, and High-Altitude Flight Performance," *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, Jun. 2022, mAG ID: 4288047729 S2ID: 4ad32ae91a08b3d354862e9e4b2943a5287e5e57.

[17] A. F. Torres-Rua, "Vicarious Calibration of sUAS Microbolometer Temperature Imagery for Estimation of Radiometric Land Surface Temperature." *Sensors*, vol. 17, no. 7, p. 1499, Jun. 2017, mAG ID: 2645536640 S2ID: 39e2ecc5f36c293912b349aa44a2dfb78294ebcf.

[18] M. Elarab, A. M. Ticlavilca, A. F. Torres-Rua, Inga Maslova, I. Maslova, and M. McKee, "Estimating chlorophyll with thermal and broadband multispectral high resolution imagery from an unmanned aerial system using relevance vector machines for precision agriculture," *International Journal of Applied Earth Observation and Geoinformation*, vol. 43, pp. 32–42, Dec. 2015, mAG ID: 2086843634 S2ID: c79486948ddc60057e76c87e12ed1cad0f49bead.

[19] L. Hassan-Esfahani, A. F. Torres-Rua, A. M. Jensen, and M. McKee, "Assessment of Surface Soil Moisture Using High-Resolution Multi-Spectral Imagery and Artificial Neural Networks," *Remote Sensing*, vol. 7, no. 3, pp. 2627–2646, Mar. 2015, mAG ID: 1989700757 S2ID: c98cc6ca9129ce640b1b1d38a4ac7bbec6ebf9c0.

[20] M. Smith, D. A. Craig, N. Herrmann, E. Mahoney, J. Krezel, N. McIntyre, and K. Goodliff, "The Artemis Program: An Overview of NASA's Activities to Return Humans to the Moon," *IEEE Aerospace Conference*, pp. 1–10, Mar. 2020, mAG ID: 3080523071 S2ID: 4c3ddf7469fa4be7738fd501a957f20499a6f363.

[21] F. A. Almalki and Faris. A. Almalki, “Utilizing Drone for Food Quality and Safety Detection using Wireless Sensors,” *2020 IEEE 3rd International Conference on Information Communication and Signal Processing (ICICSP)*, 2020, mAG ID: 3094256658 S2ID: 8b252169ab4063641a3e61e6f0026374f3afcda1.

[22] I. Daugėla, Ignas Daugėla, J. S. Visockienė, J. S. Visockiene, Jurate Kumpiene, and J. Kumpiene, “DETECTION AND ANALYSIS OF METHANE EMISSIONS FROM A LANDFILL USING UNMANNED AERIAL DRONE SYSTEMS AND SEMICONDUCTOR SENSORS,” *Detritus*, vol. 10, no. 10, pp. 127–138, May 2020, mAG ID: 3024859416 S2ID: 5f463fb8cb9a6190a79044ec5d0841713d82f9ed.

[23] F. Svanstrom, C. Englund, and F. Alonso-Fernandez, “Real-Time Drone Detection and Tracking With Visible, Thermal and Acoustic Sensors,” *International Conference on Pattern Recognition*, Jul. 2020, aRXIV_ID: 2007.07396 MAG ID: 3042973333 S2ID: 0c12c5daa195c20cfbadb8f676fec0aa5b66b577.

[24] Z. Cook, L. Zhao, J. Lee, and W. Yim, “Unmanned aerial system for first responders,” *2015 12th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pp. 306–310, Dec. 2015, mAG ID: 2216847974 S2ID: c44c14cc05e7a8862cafaa9c6462f890a7b498d8.

[25] C. J. Moran, C. Seielstad, M. R. Cunningham, V. Hoff, R. A. Parsons, L. Queen, K. Sauerbrey, and T. Wallace, “Deriving Fire Behavior Metrics from UAS Imagery,” *Fire*, vol. 2, no. 2, p. 36, Jun. 2019, mAG ID: 2952085476 S2ID: d6630485077d9fb2fe538ca9fb27729f2c7a3c56.

[26] Saket Gowravaram, Haiyang Chao, Tiebiao Zhao, Sheena Parsons, Xiaolin Hu, Ming Xin, Harold Flanagan, and Pengzhi Tian, “Prescribed grass fire evolution mapping and rate of spread measurement using orthorectified thermal imagery from a fixed-wing UAS,” *International Journal of Remote Sensing*, pp. 1–20, Mar. 2022, mAG ID: 4221029153 S2ID: 6d00f250af327b900dad7183c4ad417a11758a5f.

[27] T. P. Singh, “Use of Optical Gas Imaging for Visualizing Gas Leakage and other Thermal Camera Applications in Oil & Gas Sector,” *Day 1 Mon, November 11, 2019*, Nov. 2019, mAG ID: 2989020114 S2ID: c4ab8717ebbf31c7f20b7c12d0e9a2f9f3e8bd67.

[28] C. Coopmans, S. Brimhall, R. Goodman, and S. Petruzza, “Scientific timely actionable robotic data operating system (STARDOS): architecture and progress,” *Defense + Commercial Sensing*, vol. 11747, pp. 63–73, Apr. 2021, mAG ID: 3155061949.

[29] C. Coopmans, L. Di, A. M. Jensen, A. A. Dennis, and Y. Chen, “Improved Architecture Designs for a Low Cost Personal Remote Sensing Platform: Flight Control and Safety,” vol. 3, pp. 937–943, Jan. 2011, mAG ID: 2017079583 S2ID: 29c07bcbcaae350cec5bed779a168304466b5d2a.

[30] C. Coopmans, B. Stark, and C. M. Coffin, “A payload verification and management framework for small UAV-based personal remote sensing systems,” *2012 5th International Symposium on Resilient Control Systems*, pp. 184–189, Sep. 2012, mAG ID: 2035001381 S2ID: 7a07088b166476cbefbd4e1181e976b6c00f2037.

[31] "PX4 - Companion Computer," Mar. 2023. [Online]. Available: https://docs.px4.io/main/en/companion_computer/pixhawk_companion.html

[32] C. Coopmans, S. Stockton, N. Schwemmer, C. Vance, A. J. Beckwith, and D. J. Robinson, "GreatBlue: a 55-pound Vertical-Takeoff-and-Landing Fixed-Wing sUAS for Science; Systems, Communication, Simulation, Data Processing, Payloads, Package Delivery, and Mission Flight Performance," *Journal of Intelligent & Robotic Systems (Under Review)*, 2023.

[33] "Mavlink," Mar. 2023. [Online]. Available: https://mavlink.io/en/

[34] "File Transfer Protocol (FTP) · MAVLink Developer Guide." [Online]. Available: https://mavlink.io/en/services/ftp.html

[35] M. Quigley, "ROS: an open-source Robot Operating System," Jan. 2009, mAG ID: 2901136733.

[36] "Ground Station Software | UgCS PC Mission Planning." [Online]. Available: https://www.ugcs.com/

[37] R. Pravin, T. Prabha, V. Durai, and N. Kumar, "Methods to Optimizing Human Error in Aviation Safety," 2013, s2ID: d6b8713071c0aa4b999d3603d5dcddd8ca4a6263.

[38] Daniela Doroftei, Geert De Cubber, and Hans De Smet, "Assessing Human Factors for Drone Operations in a Simulation Environment," *AHFE international*, Jan. 2022, mAG ID: 4285142675 S2ID: c85afbae0a1b48783efb9a05e1d32a8c120a5a9b.

[39] Jan B. Smith, "DC 6 Through 737 MAX – Identifying Unreliability with In-service Precursors to Avoid the First Crash," *Reliability and Maintainability Symposium*, Jan. 2022, mAG ID: 4296426173 S2ID: 7bf2494b0967de75742529a60d72ff01d18ccb0a.

[40] Jin-Keun Hong, "Safety Risk Management Policy of United States small unmanned aerial system," 2021, s2ID: fadb5d2defc13da1da9c35fbbfa99c4e717931f5.

[41] M. S. Lim, S.-S. Yeo, Sang Soo Yeo, Sang Soo Yeo, and Y. H. Kim, "Drone-Based Simulation for the Establishment of Integrated Safety System in Educational Space," *Journal of Digital Contents Society*, vol. 20, no. 8, pp. 1865–1871, Sep. 2019, mAG ID: 2978253784 S2ID: 3d1d0ae79bbb2fbfef06de86a9a0c0a35c18a41a.

[42] DongYeol Lee, SunHoo Park, and SangJoon Shin, "Risk Assessment of a Drone Under the Gust and its Precise Flight Simulation," , vol. 50, no. 3, pp. 173–180, Mar. 2022, mAG ID: 4214935453 S2ID: 7029d7cb5b288ad621c0afa6dce5fac1d32f9040.

[43] Zixian Zhu, Idris Jeelani, None Gheisari, and None Masoud, "Safety Risk Assessment of Drones on Construction Sites using 4D Simulation," *Proceedings of the ... ISARC*, Jul. 2022, mAG ID: 4289830326 S2ID: d68dc1fa927d9bc589df08008237206b660f4209.

[44] "PX4 Autopilot," Nov. 2022. [Online]. Available: https://px4.io/

[45] Y. Niu, H. Qazi, and Y. Liang, "Building a Flexible Mobile Robotics Teaching Toolkit by Extending MATLAB/Simulink with ROS and Gazebo," *International Conference on Mechatronics and Robotics Engineering*, pp. 10–16, 2021, mAG ID: 3151996935 S2ID: b764c1b340ab7a522676da4f6df969acec90a94d.

[46] J. Cañas, L. Martín, and J. Vega, "Innovating in robotics education with Gazebo simulator and JdeRobot framework," 2014, s2ID: beb900462ada537c04ec59cfb4d563bb6e7b90f4.

[47] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," *IEEE/RJS International Conference on Intelligent RObots and Systems*, vol. 3, pp. 2149–2154, Sep. 2004, mAG ID: 2167340365 S2ID: df0295715f8b43a8cbaa19f74a3dceb06e610eaf.

[48] Adam Wolniakowski and Maja Czarna, "A Framework for Model Sailing Simulation in Gazebo," *International Conference on Methods & Models in Automation & Robotics*, Aug. 2022, mAG ID: 4294975840 S2ID: f3fad3a9fb12e1e3e4c89321e4731add20cc5540.

[49] "PX4 Simulation," Mar. 2023. [Online]. Available: https://docs.px4.io/main/en/simulation/

[50] Z. Wang and J. Li, "3D Simulation of Flight Control System for Quad Tilt Rotor UAV Based on Flightgear," *IOP Conference Series: Earth and Environmental Science*, vol. 440, no. 5, p. 052056, Mar. 2020, mAG ID: 3011867358 S2ID: fb34b5e41b28252ccf94f47bfda3b2cc01cd30f1.

[51] Pan Longfei, Luo Jing, Shi Yu, Pang Lu, and Sheng Rong, "Design and Implementation of Flight Vision Simulation System Based on Flight Training Data and Flightgear," *2022 2nd International Conference on Big Data Engineering and Education (BDEE)*, 2022, s2ID: ec399ab4eab4a7736cefa10af9e8ced97a210935.

[52] T. Vogeltanz, "Control system designer for JSBSim with algorithm of automatic Ziegler-Nichols sustained-oscillation method," vol. 2293, no. 1, p. 080006, 2020, mAG ID: 3107097436 S2ID: b0ed36eb30e06b6caad4150a70e30245f3fc0617.

[53] Microsoft, "AirSim," Mar. 2023. [Online]. Available: https://microsoft.github.io/AirSim/

[54] Sergio Hernandez-Mendez, Carlos Hernandez-Mejia, Derek Benjamin Herrera Olea, Antonio Marin-Hernandez, and Homero Vladimir Rios-Figueroa, "Path Planning Simulation of a quadrotor in ROS/Gazebo using RGPPM," *2021 International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE)*, Nov. 2021, mAG ID: 4297099774 S2ID: 954cb75f5010dabc82eb52365395788f7e375bc3.

[55] Abdussalam A. Alajami, R. Pous, and Guillem Moreno, "Simulation of RFID Systems in ROS-Gazebo," *2022 IEEE 12th International Conference on RFID Technology and Applications (RFID-TA)*, 2022, s2ID: 1eb26faaefcb51dbf86bab6f25a45c0f458f950f.

[56] R. Safin, R. Lavrenov, and E. A. Martinez-Garcia, "Evaluation of Visual SLAM Methods in USAR Applications Using ROS/Gazebo Simulation," pp. 371–382, 2021, mAG ID: 3081806719 S2ID: 5ad607fd1990690ca8af0843b7080940c0148416.

[57] Gabriel M. Miranda, Leandro M. F. Machado, J. Arias-Garcia, V. Guilherme, and Raffo, "A MULTI-CORE SOFTWARE DESIGN OF A MODEL PREDICTIVE CONTROL FOR A TILTROTOR UAV," 2017, s2ID: 99689bdd5a7e5bb1408f8529782685c2f25e73ae.

[58] S. Moon, J. J. Bird, S. Borenstein, and E. W. Frew, "A Gazebo/ROS-based Communication-Realistic Simulator for Networked sUAS," *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2020, mAG ID: 3091819828 S2ID: 5ddb22ff9b97712b0f02af227d9124d153c1108c.

[59] L. Ikkala, H. Marttila, A.-K. Ronkanen, J. Ilmonen, S. Rehell, T. Kumpula, and B. Kløve, "Thermal UAS Imaging to Monitor Restored Peatlands," Mar. 2021, mAG ID: 3176789924 S2ID: 4deffc562db85b71260bd8ddd1899f96767e85b9.

[60] P. Alekseychik, G. G. Katul, I. Korpela, and S. Launiainen, "Eddies in motion: visualizing boundary-layer turbulence above an open boreal peatland using UAS thermal videos," *Atmospheric Measurement Techniques Discussions*, vol. 14, no. 5, pp. 3501–3521, 2020, mAG ID: 3096774711 S2ID: 49e54b3ef8ade41cd87a94405a1be480117a03c5.

[61] M. Präg, I. Becker, C. Hilgers, T. R. Walter, and M. Kühn, "Thermal UAS survey of reactivated hot spring activity in Waiwera, New Zealand," *Advances in Geosciences*, vol. 54, pp. 165–171, 2020, mAG ID: 3107285803 S2ID: 605b95299a37de73618c929242b126f0037b1489.

[62] A. M. de Oca, G. Flores, and G. Flores, "A UAS equipped with a thermal imaging system with temperature calibration for Crop Water Stress Index computation," *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 714–720, Jun. 2021, mAG ID: 3189636477 S2ID: 05137da960349c8e751a77d8a2ffdaef2e6813af.

[63] Hualong Tang, J. Post, A. Kourtellis, Brian Porter, and Yu Zhang, "Comparison of Object Detection Algorithms Using Video and Thermal Images Collected from a UAS Platform: An Application of Drones in Traffic Management," *ArXiv*, 2021, aRXIV_ID: 2109.13185 S2ID: 65bfd55d7d7386c00ba5188638d6d3041b8c7443.

[64] Miguel Dávila-Sacoto, M. Dávila-Sacoto, L. Hernández-Callejo, V. Alonso-Gómez, S. Gallardo-Saavedra, Luis Gonzalez, and L. Gonzalez, "Detecting Hot Spots in Photovoltaic Panels Using Low-Cost Thermal Cameras," *Ibero-American Congress of Smart Cities*, pp. 38–53, Oct. 2019, mAG ID: 2998640856 S2ID: 462fed5120dde19bd084b06f73a2ef4ba0712a74.

[65] A. Al-Shammari, E. Levin, R. Shults, R. Shults, Roman Shults, and Roman Shults, "Oil Spills Detection by Means of Uas and Low-Cost Airborne Thermal Sensors," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 45, pp. 293–301, Nov. 2018, mAG ID: 2900615399 S2ID: d079774e54bb7058d9c0f23a4d0e078eea10b31a.

[66] R. Pérez-Álvarez, J. Sedano-Cibrián, J. M. de Luis-Ruiz, G. Fernández-Maroto, and R. Pereda-García, "Mining Exploration with UAV, Low-Cost Thermal Cameras and

GIS Tools—Application to the Specific Case of the Complex Sulfides Hosted in Carbonates of Udías (Cantabria, Spain)," *Minerals*, vol. 12, no. 2, pp. 140–140, Jan. 2022, mAG ID: 4210420555 S2ID: c708cf4c3195a9db9470c1a734bb380c253805c6.

[67] P. J. Blaya-Ros, V. Blanco, V. Blanco, R. Domingo, F. Soto-Valles, and R. Torres-Sánchez, "Feasibility of Low-Cost Thermal Imaging for Monitoring Water Stress in Young and Mature Sweet Cherry Trees," *Applied Sciences*, vol. 10, no. 16, p. 5461, Aug. 2020, mAG ID: 3048260270 S2ID: c3697cc6c5e7f6bc6a4d8cdf67443606a7267400.

[68] M. I. Perdana, A. Risnumawan, and I. A. Sulistijono, "Automatic Aerial Victim Detection on Low-Cost Thermal Camera Using Convolutional Neural Network," *Conference on Computer and Communications Security*, pp. 1–5, 2020, mAG ID: 3093544860 S2ID: 9b896c2b0d92cfb17e62d6b71aa2e6ae91a6680e.

[69] "Fluke RSE600/C 60HZ - Fixed Mount Thermal Imager with Case, HS1 - 640 x 480 Resolution, 60 Hz | TEquipment." [Online]. Available: https://www.tequipment.net/Fluke/RSE600/C-60HZ/Scientific-Thermal-Imagers/?Source=googleshopping&gclid=Cj0KCQjwuLShBhC_ARIsAFod4fLTiW9DJBMgmaFc62C4bEKKP1EFn_9YwSXC4JM4oQut67A2HXgGiK8aAjUlEALw_wcB

[70] L. Technology, "Teledyne FLIR BFS-U3-244S8C-C." [Online]. Available: https://www.loretech.io/products/teledyne-flir-bfs-u3-244s8c-c

[71] "PX4 Airframes," Mar. 2023. [Online]. Available: https://docs.px4.io/main/en/airframes/airframe_reference.html

[72] "Gazebo Tutorials - Sensors," Apr. 2023. [Online]. Available: http://classic.gazebosim.org/tutorials?cat=sensors

[73] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, Mar. 1960. [Online]. Available: https://asmedigitalcollection.asme.org/fluidsengineering/article/82/1/35/397706/A-New-Approach-to-Linear-Filtering-and-Prediction

[74] S. Julier and J. Uhlmann, "New extension of the Kalman filter to nonlinear systems," *Defense, Security, and Sensing*, vol. 3068, pp. 182–193, Jul. 1997, mAG ID: 2020934227 S2ID: a8b141556f2dd7694e5f6343f8ce3650f8ca5b60.

[75] "PX4 Simulation - Survey Camera," Mar. 2023. [Online]. Available: https://docs.px4.io/v1.12/en/simulation/gazebo.html#simulated-survey-camera

[76] "PX4-SITL_gazebo-classic/gazebo_camera_manager_plugin.cpp at main · PX4/PX4-SITL_gazebo-classic." [Online]. Available: https://github.com/PX4/PX4-SITL_gazebo-classic

[77] "Gazebo Tutorials - Make a Model," Apr. 2023. [Online]. Available: https://classic.gazebosim.org/tutorials?tut=build_model

[78] Worasit Sangjan, Rebecca J. McGee, and Sindhuja Sankaran, "Optimization of UAV-Based Imaging and Image Processing Orthomosaic and Point Cloud Approaches for

Estimating Biomass in a Forage Crop," *Remote Sensing*, vol. 14, no. 10, pp. 2396–2396, May 2022, mAG ID: 4280611656 S2ID: 94fc59ac6bf9470b9738b7a80fe074dac5f9ebb5.

[79] Saneev Kumar Das, Pratik Bose, Kalpita Patra, and S. Chakravarty, "UAV Based Multispectral Image Processing Framework: A Band Combination Approach," *2023 International Conference on Machine Intelligence for GeoAnalytics and Remote Sensing (MIGARS)*, 2023, s2ID: e7fb1049fdf55de8fb74e5938bb94878464597fa.

[80] N. Liba and J. Berg-Jürgens, "Accuracy of Orthomosaic Generated by Different Methods in Example of UAV Platform MUST Q," vol. 96, no. 1, p. 012041, Nov. 2015, mAG ID: 2507232699 S2ID: 005635bd1354327535c71763864c5f14aa9ff69c.

[81] "Agisoft Metashape: Agisoft Metashape." [Online]. Available: https://www.agisoft.com/

[82] "Gazebo : Tutorial : Color And Texture Models." [Online]. Available: https://classic.gazebosim.org/tutorials?tut=color_model&cat=

[83] "QGC." [Online]. Available: http://qgroundcontrol.com/

[84] "ExifTool by Phil Harvey." [Online]. Available: https://exiftool.org/

[85] "Agisoft Metashape," Nov. 2022. [Online]. Available: https://www.agisoft.com/

[86] "14 CFR Part 107 – Small Unmanned Aircraft Systems." [Online]. Available: https://www.ecfr.gov/current/title-14/chapter-I/subchapter-F/part-107

[87] A. M. Duncan, Andrew M. Duncan, Bogdan Vacaliuc, B. Vacaliuc, Matthew D. Larson, M. D. Larson, C. Davis, D. Hughes, David Hughes, and K. A. Corcoran, "Integration and operation of a sUAS-based multi-modal imaging payload equipped with a novel communication system enabling autonomous platform control," *Defense + Commercial Sensing*, vol. 11021, p. 1102103, May 2019, mAG ID: 3012018645 S2ID: c12fc8788a4094ed7f06ff48993ae233ac5b29dd.

[88] "Lepton LWIR Micro Thermal Camera Module | Teledyne FLIR." [Online]. Available: https://www.flir.com/products/lepton?vertical=microcam&segment=oem

[89] "FLIR Lepton 3.5," Nov. 2022. [Online]. Available: https://www.digikey.com/en/products/detail/flir-lepton/500-0771-01/7606616

[90] "PureThermal-2," Nov. 2022. [Online]. Available: https://www.digikey.com/en/products/detail/groupgets-llc/PURETHERMAL-2/9866290

[91] "Raspberry Pi 4B/4GB," Nov. 2022. [Online]. Available: https://www.digikey.com/en/products/detail/raspberry-pi/RASPBERRY-PI-4B-4GB/10258781

[92] "Raspberry Pi HQ Camera," Nov. 2022. [Online]. Available: https://www.digikey.com/en/products/detail/raspberry-pi/SC0818/12339164

[93] A. Industries, "16mm 10MP Telephoto Lens for Raspberry Pi HQ Camera." [Online]. Available: https://www.adafruit.com/product/4562

[94] "Hardware in the Loop Simulation (HITL) | PX4 User Guide." [Online]. Available: https://docs.px4.io/main/en/simulation/hitl.html

[95] "Manual/Stabilized Mode (Multicopter) | PX4 User Guide." [Online]. Available: https://docs.px4.io/main/en/flight_modes/manual_stabilized_mc.html

APPENDICES

APPENDIX A

A.1 File Modifications made for creating and building the quad-plane VTOL camera aircraft

This is where modifications and additions to the PX4 repository [44] are described. All paths are relative the the home directory of the PX4 repository.

In the Tools/simulation/gazebo−classic/sitl_gazebo−classic/models/ directory the *standard_vtol_cam* folder was added. Inside of that folder two files were created:

- standard_vtol_cam.sdf - The sdf file provides the description needed to create the quad-plane vtol camera aircraft.

- model.config - The config file provides some basic description about the quad-plane vtol camera.

In the src/modules/simulation/simulator_mavlink/ directory:

- sitl_targets_gazebo −classic.cmake - this file was modified to include info about building the quad-plane VTOL airframe.

Lastly, in the ROMFS/px4fmu_cmmon/init.d−posix/airframes directory:

- 1045_gazebo−classic_standard_vtol_cam - this file is used to describe the parameters that should be modified and/or set when running the quad-plane vtol camera aircraft.

- CMakeLists.txt - this file was modified to include info about building the quad-plane VTOL airframe.