Aerial Pathfinding Reconnaissance

Jason Gilman
Max Griffith
Mark Johnson
Dilanka Weerasinghe

CONCEPT OF OPERATIONS

REVISION 2 Draft
20 September 2020

CONCEPT OF OPERATIONS
FOR
Aerial Pathfinding Reconnaissance

TEAM 2

APPROVED BY:

____
Jason Gilman Date

____
Prof. S. Kalafatis Date

____
Souryendu Das Date

Change Record

Rev.	Date	Originator	Approvals	Description
1	9/6/2020	Jason Gilman		Draft Release
2	9/20/2020	Jason Gilman		Networking Subsystem Edits

Table of Contents

List of Figures

1. Executive Summary

Autonomous vehicles are becoming more popular for a litany of reasons, including safety, efficiency, and convenience. However, autonomous ground vehicles often suffer from a lack of situational awareness in attempting to navigate their environments. To combat this, product development in this field has tended towards increasing sensory capabilities – at a rapidly rising price point. Drones, while agile in their movement capabilities, suffer from poor cargo capacity. On the other hand, aerial drones offer a convenient platform for gathering data over large swathes of land. By combining the efficiency of autonomous data gathering with the capabilities of a ground vehicle, we can reduce sensory costs while maintaining its facilities.

This project's intent is to develop a prototype system which implements this relationship. To accomplish this, our proposed system will take a two-phase approach. First, the drone will survey a user-defined area and collect topographical data to be forwarded to a desktop application. The application will analyze this data to produce a map of obstacles in the area, and using the dimensions of the ground vehicle, determine a suitable path between the user's chosen endpoints. The application will then send navigation instructions to the ground vehicle to execute this path.

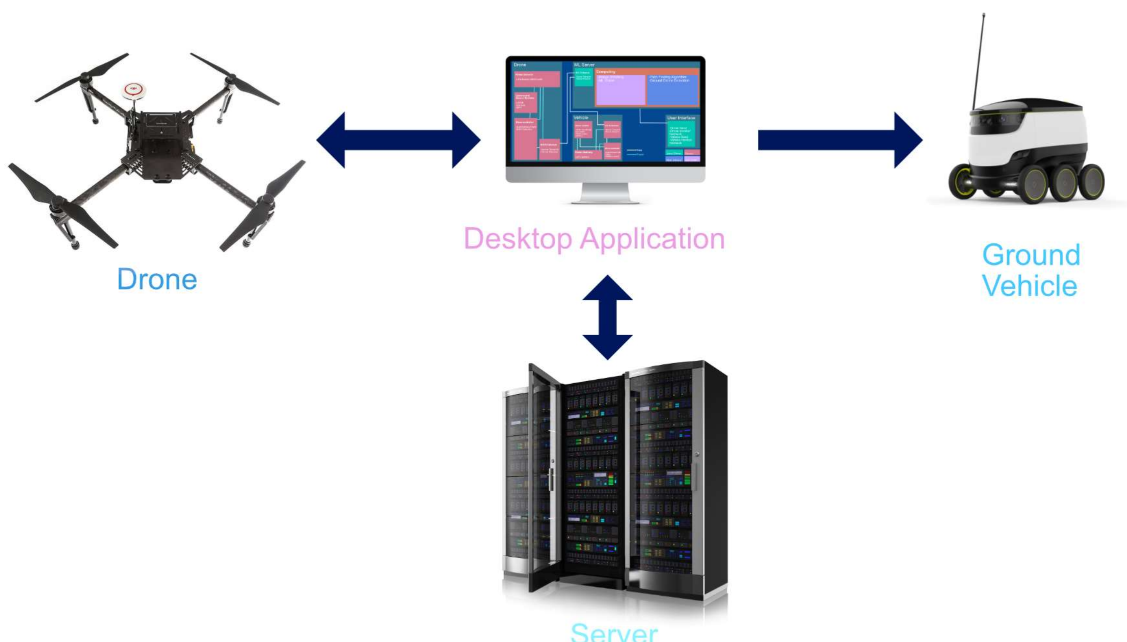

Figure 1. High-Level System Overview

2. Introduction

This document is an introduction to the Aerial Pathfinding Reconnaissance System (APRS), a system that will monitor a given area and provide a safe, reliable path for an otherwise-blind ground vehicle. This system applications are widespread, including navigation of natural disaster areas to military scouting. The APRS will serve as a protectant to the drone and the user as it eliminates human error in flight and provides an efficient and safe path to the land traveler.

2.1. Background

Aerial reconnaissance has special applications in military and industrial environments alike. Among these domains, drones equipped with LiDAR technology are sometimes used to supplement or replace traditional land surveys to provide an even more accurate and detailed physical map of the land. However, for some practical applications of aerial recon, the use of LiDAR may not be entirely appropriate, as it runs the risk of incurring unnecessary expense, complexity, and in extreme failure cases, even physical injury.

For example, consider land pathfinding. Pathfinding does not require the same degree of precision and accuracy as other industrial or military applications. In theory, the problem of pathfinding on land can, in general, be reduced to a simple distinction between traversable space and obstacles on a two-dimensional surface. For this purpose, the use of LiDAR is beyond excessive, as a reasonably accurate determination of this data can be made with much simpler hardware, such as an ordinary video camera.

The process of creating a map of an area using image data from a drone is referred to as *drone photogrammetry,* and it is a far cheaper process than an equivalent LiDAR solution, with industrial LiDAR drone mapping systems reaching prices between 2.5x and 10x the cost of a high-end photogrammetry system.[1] With a cost-efficiency motivation and a particular use-case in mind, this project focuses on the development of an even more hardware-flexible and economical approach than ordinary LiDAR-equipped solutions, trading a high degree of accuracy for practicality and mission-specific functionality.

2.2. Overview

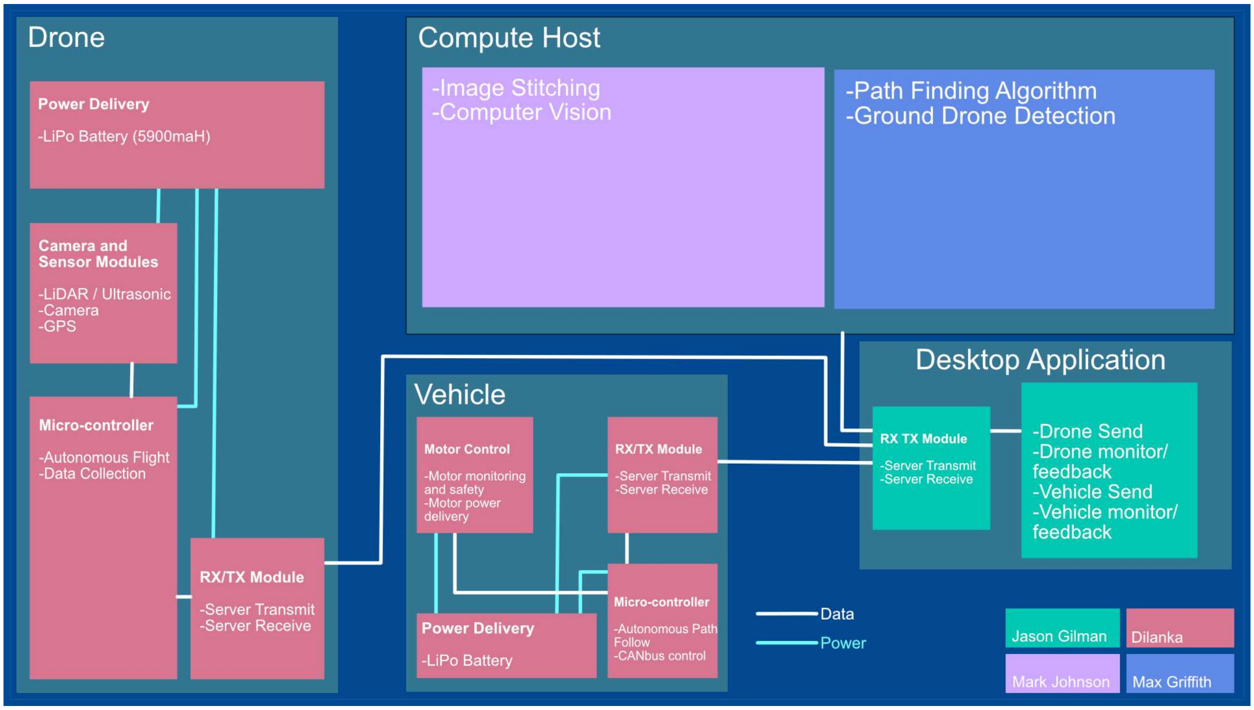

Figure 2. Functional Block Diagram

Our system uses video captured by a drone to provide a path that a user can blindly follow to its destination. We will first create a land map using video from a drone. Using an Aerial Pathfinding Reconnaissance System (APRS) application, the user will provide an area that the drone will survey. The drone will then navigate the chosen area at an altitude provided by the user.

The APRS system will utilize an array of LiDAR and ultrasonic sensors for object avoidance in the air and the drone will exclude any areas it is incapable of navigating. If it detects an area it is not capable of maneuvering it will utilize object avoidance techniques to continue on its path. Upon returning from flight, data is sent to a locally run desktop application that will stitch the land-map together and begin to detect obstacles. Using computer vision algorithms, it will be programmed to detect stationary obstacles and moving obstacles on the map of the area.

Once the full map is completed and the obstacles are pinpointed on the map the program will path-find an efficient and safe path for the land vehicle to follow. At this point the path is relayed to a small land vehicle and the vehicle is programmed to follow the path blindly until it reaches its given destination.

2.3. Referenced Documents and Standards

[1] "Drone Photogrammetry vs. LIDAR: What Sensor to Choose for a given Application" https://wingtra.com/drone-photogrammetry-vs-lidar/
[2] IEEE 802.11 https://standards.ieee.org/standard/802_11-2016.html
[3] FAA Part 107 https://www.faa.gov/uas/media/RIN_2120-AJ60_Clean_Signed.pdf

3. Operating Concept

3.1. Scope

Functions of the APRS:
The Aerial Pathfinding Reconnaissance System (APRS) will include two drones, an aerial drone and a land drone. The aerial drone will survey an area and transfer its data to an application which will plan out a specific path for a land drone to follow. This system will allow vehicles to safely navigate terrain without any human interaction or monitoring once the data is transported to the land drone.

Features of the APRS:
The drone will be using camera that will locate and distinguish fixated objects from objects that can be removed (e.g. humans or other animals) and will have a distance sensor to accurately calculate the size of objects and the correct travel distances for the land drone to follow.

3.2. Operational Description and Constraints

The Aerial Pathfinding Reconnaissance System will be operated to identify a safe ground path through terrain that can be surveyed by a flight drone within the timeframe of 20 minutes. Once the aerial drone has surveyed the area, the data collected from surveillance will be manipulated with machine learning to create a map that then will be deployed to a land drone that will be able to blindly follow the designated path.

A constraint that will occur will be limitations with the video capturing device. In low light, using pure surveillance could potentially obstruct views. Any usage that would be as a means for stealth will be impacted. Furthermore, the operating range of the drone is constrained by battery capacity and the computational power available to the user.

3.3. System Description

Data Collection and Drone Control: The hardware for this project will handle the data collection to be used by the Computer Vision and Mapping program. On the drone there will be sensors to handle aerial obstacle avoidance and recording the ground. The ground vehicle will lack sensors as it will follow the provided path blindly. It will have its necessary controllers to perform its required movement. The following are the smaller subsystems related to hardware on the drone and the ground vehicle:

- **Power Distribution (Aerial and Ground):** The drone will be using Lithium Ion Batteries for high efficiency, reliability under extreme weather and weight constraints. This system will power the drone cameras, sensors and transmission to the application. The drone is capable of holding two 5900mAh LiPo batteries and this will allow for 40 minutes of flight time. A LiPo battery will also be used to power the ground vehicle. A series of buck/boost converters will translate the LiPo voltage that will power the microcontroller, motor controller and motors on the system.

- **Camera (Aerial):** The camera will take video of the ground at certain points in the user provided area. These data taken points will be relayed back the application and will define the main components of the map.

- **LiDAR (Aerial):** LiDAR stands for Light Detection and Ranging. It operates by shooting a condensed laser or light beam at an area and waiting for the frequency of light to return to its sensor. The distance is calculated by using the speed of light and the time it takes for the light to return to the sensor. This system will not be using LiDAR to detect obstacles on the ground or to create a 3-dimensional map of the surrounding area. The drone will only be using its integrated LiDAR to handle its own obstacle avoidance in the sky.

- **GPS (Aerial and Ground):** GPS connects to orbiting satellites to triangulate the location of a sensor on the earth. GPS will be used to triangulate the precise location of the drone so that the computer vision can create an accurate map of the area. Without the GPS the camera data would not be connected to a given location. A GPS will also be held in the ground vehicle to calculate its location as it is following the path.

- **DJI Drone Controller (Aerial):** All system sensors are monitored through the DJI controller. This controller also handles the flight path generated by the user preferences. This controller is a proprietary DJI product. Using DJI’s SDK it can be modified to accommodate for any flight path and sensor detection needed. This drone controller will also have the wireless data transmission capabilities that will relay information back to the application.

- **Ground Controller:** The control of the ground vehicle is monitored and controlled by an STM32. Connected to the control unit is a Sabretooth motor driver that will act as the bridge from command to motor.

Computer Vision and Mapping: Once video data has been received by the desktop application, a set of custom software will analyze the data and produce an overhead view of the entire traversable ground area. First, the video data will be consumed in pieces to produce stereographic ground views, with moving obstacles eliminated from the data. These stereographic views of each area will then be analyzed further to produce a depth estimation map. With some lightweight mathematics performed over this field, a final map can be produced which delineates areas in which too great of a height difference is encountered at once, forming an obstacle.

Pathfinding and Drone Detection: The video data will also be analyzed by another system which locates the land vehicle’s starting and finishing positions within the environment. After determining these values, it will use the dimensions of the vehicle and a two-dimensional map of obstacles to produce a path, converting these directions and distances into commands for the land vehicle to follow.

Device Networking and UI: Utilizing a locally run desktop application, and microcontrollers onboard the drone and autonomous land vehicle, data will be routed through a wireless network in our system. A UI will keep the user informed of the system's progress, as well as act as a controller for the system. The system can be launched, paused, and stopped through use of the UI. Optionally, a cloud server may be utilized for increased computational power when running software analysis.

3.4. Modes of Operations

The Aerial Pathfinding Reconnaissance System will have a single mode of operation. Initially, data is collected from sensors onboard an aerial drone. Upon returning from flight, the data will be sent through a wireless network to a local machine that is running the systems application. The application will analyze the collected data and generate navigation instructions. Finally, the navigation instructions will be sent through a wireless network to an autonomous vehicle. A user interface on the application will be utilized to launch, track, and update the system.

3.5. Users

The installation for the Aerial Pathfinding Reconnaissance System will be simple. The data and means of operating this system will be on a user interface contained in a desktop application, which will come with the drone. Any of this information will be in a user manual.

The controlling of the aerial done will require a coordinate system so that the drone will only survey a specified area with height requirements that the drone will follow. The parameters of this system will have to be determined by the user.

3.6. Support

Support for the Aerial Pathfinding Reconnaissance System will be provided in the form of user manuals outlining setup, UI navigation, system maintenance, system usage. The user manual will describe how to read incoming feedback from the drone and from the programmed vehicle. This user manual would be locating in the help section of the application and provided to the user in a printed format. A setup pamphlet will be provided detailing network setup as well as assembly of the drone and land vehicle.

4. Scenario(s)

4.1. Aiding Natural Disaster Relief

The Aerial Pathfinding Reconnaissance System could provide exceptional navigation assistance in dangerous environments. Natural disasters often create problems with physically getting relief to disaster sites. Utilizing an aerial drone to capture video data of disaster areas, and optimizing the object detection to avoid floods, blocked roads, or fires could expedite the arrival of aid to areas impacted by disaster.

4.2. Expanding Navigation Apps

Navigation apps such as Google Maps and Apple Maps utilize databases to find the fastest route for the user, although these databases often take a long time to map newly built roads. The Aerial Pathfinding Reconnaissance System could be deployed to newly developed areas to quickly detect roads and obstacles, map the area, and update the apps' databases.

4.3. Military Scouting

The Aerial Pathfinding Reconnaissance System can have a profound impact in a military application. By mapping dangerous environments, detecting obstructing or dangerous objects, and creating navigation instructions, military transport vehicles would be able to shift towards autonomy. This has the potential to save lives and expedite the transportation of soldiers, equipment, or aid.

5. Analysis

5.1. Summary of Proposed Improvements

The primary benefit of using advanced image processing over LiDAR technology is the potential for extreme cost reduction and maintainability in applications. As LiDAR sensors are expensive and sensitive devices, those wishing to use aerial reconnaissance for pathfinding are limited to options with potentially excessive upkeep.

Aside from being notoriously expensive, three-dimensional LiDAR is known to require extreme amounts of bandwidth. Potential alternatives, such as using multiple two-dimensional LiDAR sensors, requires the extra work of joining these very large datasets accurately on top of the initial cost. By using video, however, we have more control over the amount of data, and therefore bandwidth, required to function.

Leveraging advanced computer vision algorithms may potentially result in performance improvements as well, as processing hardware will be able to deal with the sensory data in discrete chunks, as opposed to arbitrarily large point clouds.

Furthermore, the use of video data as a primary source of sensory input can result in more hardware adaptability. For example, a LiDAR-based system may have the need to be finely tuned for a specific sensor to function effectively. Important details such as a sensor's operating range and environment may require special consideration, while in a purely video-based solution, any sufficiently detailed imaging will suffice.

Beyond practical advantages, video mapping also has the advantage that it may be safer to use in inhabited areas. Many LiDAR sensors, especially more powerful models, may use powerful lasers that can cause eye damage upon malfunction. Meanwhile, video recordings by themselves are never capable of directly causing physical harm.

5.2. Disadvantages and Limitations

The strongest limitation of a video-based approach, when compared to LiDAR, is its ability to function in low-light settings. For LiDAR sensors, which emit their own lasers or infrared light already, this is of no concern. However, an ordinary camera sensor will not be able to function in low light without its own lighting, or the use of a more expensive infrared camera.

One potential concern with using video to survey an area may be related to privacy concerns. LiDAR data is not sufficient to reasonably identify anyone without extreme precision and facial recognition. On the other hand, video data includes far more information about those observed by it. However, assuming that the system is only ever used in a public setting (i.e. in which no individual has any reasonable expectation of privacy), potential legal issues related to privacy are avoided altogether.

5.3. Alternatives

- LiDAR
    - More accurate depth estimation
    - Sensors alone are very expensive
    - Higher bandwidth requirements
    - More dangerous on failure, in some cases
- Machine learning
    - More direct detection of static and dynamic obstacles
    - High setup cost, requires long training sessions
    - High runtime cost, requires powerful hardware to run
    - Difficult to configure “black box” design
- Live Pathfinding
    - More stringent synchronization requirements
    - May require real-time computing onboard
    - Sensitive to network failure

Aerial Pathfinding Reconnaissance

Jason Gilman
Max Griffith
Mark Johnson
Dilanka Weerasinghe

FUNCTIONAL SYSTEM REQUIREMENTS

REVISION 1 – Draft
20 September 2020

FUNCTIONAL SYSTEM REQUIREMENTS
FOR
Aerial Pathfinding Reconnaissance

PREPARED BY:

Jason Gilman Date

Max Griffith Date

Mark Johnson Date

Dilanka Weerasinghe Date

APPROVED BY:

Jason Gilman Date

Prof. S. Kalafatis Date

T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
1	9/15/2020	Jason Gilman		Draft Release

Table of Contents

List of Figures

1. Introduction

1.1. Purpose and Scope

The Aerial Pathfinding Reconnaissance System (APRS) is intended to provide an effective approach to generating navigation instructions for autonomous vehicles. The APRS is able to efficiently analyze a user defined area such that an autonomous vehicle can traverse from endpoints that are input by the user. The APRS is designed so that the functionality of the system is abstracted from the end-user. A front facing UI will allow any person with basic knowledge in setting up drones to launch, update, and suspend the system. Figure 1 below shows a high-level overview of the APRS.

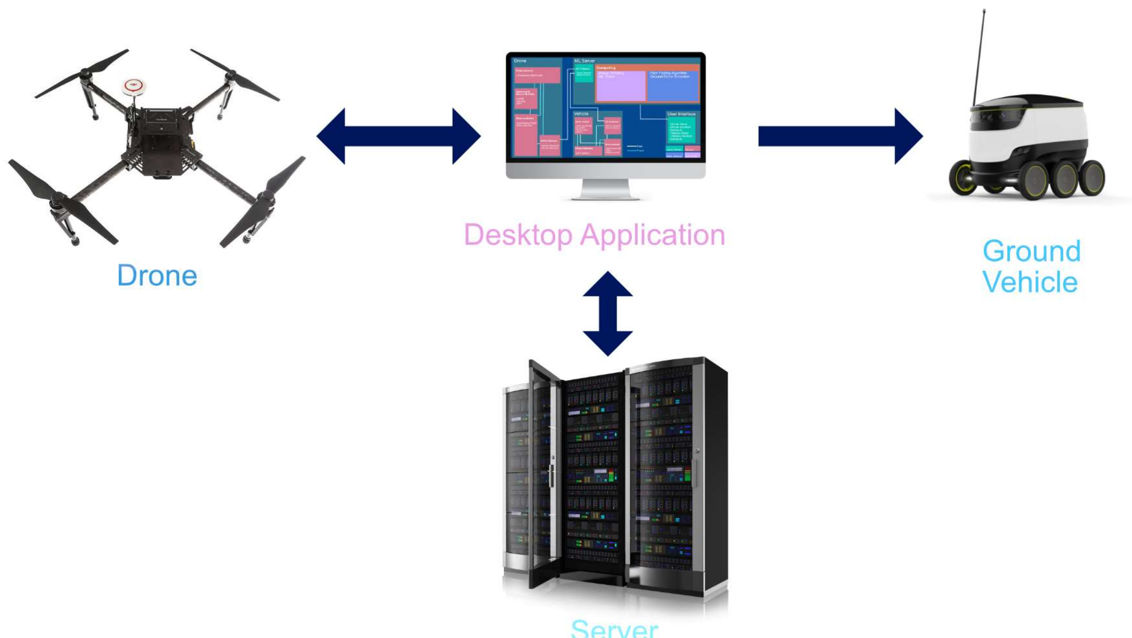

Figure 1. High-Level System Overview

The Aerial Pathfinding Reconnaissance System (APRS) will operate through interaction with a front facing UI. The UI will be run on a local machine that will serve as the main node in the APRS network, which connects all of the devices, vehicles, and software encapsulated in the system. Data collected by the drone will be sent to the local machine so that software can be utilized to generate navigation instructions. These instructions will then be routed to the ground vehicle where they will be used to traverse the user defined area.

The following definitions differentiate between requirements and other statements.

Shall:	This is the only verb used for the binding requirements.
Should/May:	These verbs are used for stating non-mandatory goals.
Will:	This verb is used for stating facts or declaration of purpose.

1.2. Responsibility and Change Authority

Jason Gilman, the team leader, has the responsibility to confirm that the APRS follows all of the defined specifications. Changes made to project documents or system specifications must be approved by Jason Gilman.

2. Applicable and Reference Documents

2.1. Applicable Documents

The following documents, of the exact issue and revision shown, form a part of this specification to the extent specified herein:

Document Number	Revision/Release Date	Document Title
IEEE 802.11	2016	IEEE Standard for Information technology—Telecommunications and information exchange between systems Local and metropolitan area networks—Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications
PEP 8	2001	Style Guide for Python Code
Part 107	2018	Fact Sheet – Small Unmanned Aircraft Regulations (Part 107)

2.2. Reference Documents

The following documents are reference documents utilized in the development of this specification. These documents do not form a part of this specification and are not controlled by their reference herein.

Document Number	Revision/Release Date	Document Title
	2019/04/11	Tarot 650 User Manual
	2018/08/10	DJI MATRICE 100 Product Release Notes
V1.6	2016/03	DJI MATRICE User Manual
V1.0	2015/09	Intelligent Flight Battery Safety Guidelines
MIL-STD-704F	1991/5/1	(D.O.D.) Aircraft Electrical Power Standard
SAE AS1212	2021/12/1	Commercial Electrical, Aircraft Characteristics

2.3. Order of Precedence

In the event of a conflict between the text of this specification and an applicable document cited herein, the text of this specification takes precedence without any exceptions.

All specifications, standards, exhibits, drawings or other documents that are invoked as “applicable” in this specification are incorporated as cited. All documents that are referred to within an applicable report are considered to be for guidance and information only, except ICDs that have their relevant documents considered to be incorporated as cited.

3. Requirements

In the following section, "Aerial Pathfinding Reconnaissance System" or "APRS" will refer exclusively to the entire system. This will include the power distribution, whether that be for the aerial drone or the land vehicle, the camera to capture ground footage and LiDAR to avoid aerial objects that may be in the drones' path, GPS systems to triangulate the precise location of the drone and to calculate the land vehicle's location as it is following the path, the DJI drone controller which will be used to accommodate for any needed flight path updates, a ground controller that will act as the bridge from command to the motor of the land vehicle, computer vision and mapping where video data will be consumed in piece to produce stereographic ground views without moving obstructions which will be analyzed to produce a depth estimation map, pathfinding and drone detection which will analyze the vehicles start and end positions in an environment and the map itself to produce a path and creating commands for the land vehicle to follow, and device networking and UI which will route the data through a wireless network and the UI will keep the user informed of the system's progress.

3.1. System Definition

The Aerial Pathfinding Reconnaissance System (APRS) is composed of four discrete entities:

- **Vehicle:** The ground vehicle will be used as a demonstration of the APRS. It contains no visual sensors for avoiding obstacles. Instead, it will receive directions from the locally run application and execute them accordingly.

- **Drone:** The aerial drone will be used to gather data on the operating environment of the ground vehicle. It will use an onboard transceiver to wirelessly relay video information for further processing on the local machine.

- **Laptop/Local Machine:** The local machine will run a desktop application that will use video information from the drone to produce directions for use by the land vehicle. To do this, it will stitch video frames together to use a map and to construct a stereographic view from which to infer depth. Then, a pathfinding algorithm will be used to generate the final directions. Optionally, these processes may be offloaded to a cloud server to increase computational power.

- **User Interface:** The user interface provided in a desktop application will form the frontend of the system, relaying information about progress and status to the user, and allowing the user to configure pre-flight parameters such as e.g. aerial path, altitude, etc.

Shared among these four elements are four subsystems. Some of these subsystems are spread across multiple entities, while some are limited to one location as shown in **Error! Reference source not found.**

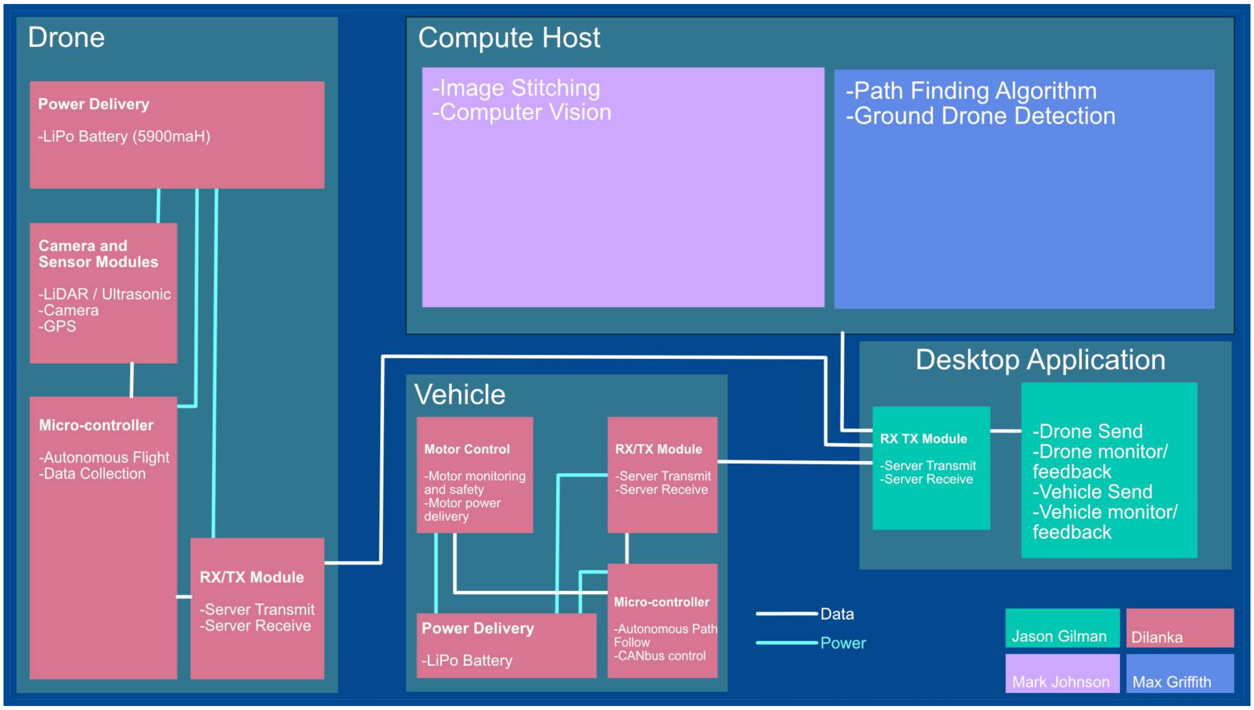

Figure 2. Functional Block Diagram

The color-coded sections of the diagram show which person will be managing their development, with each team member on one subsystem according to the following mapping:

- Jason Gilman: Device Network and User Interface
- Dilanka Weerasinghe: Data Collection and Drone Control
- Max Griffith: Pathfinding and Drone Detection
- Mark Johnson: Computer Vision and Mapping

The diagram also displays points of communication or otherwise shared resources. Where not specified, it is assumed that resources available to one entity are shared by all subsystems within it.

For example, consider how all of these components are used in our system. First, the drone uses its components (power, cameras, etc.) to collect imaging data of the environment. Then, through its connection to the local machine facilitated by their transceiver modules, this data is relayed for processing. Progress on this task is relayed to the user through the interface, which can also be used to set flight parameters and other options.

The processing of this data occurs in the following fashion: First, the imaging data (in the form of a video) is stitched together to form a complete bird's eye view of the environment. After depth estimation and obstacle detection is complete, the pathfinding algorithm and ground drone detection work together to find a path from its current position to the target. Finally, after a path is found and navigation instructions are created, they can be sent to the ground vehicle (which uses its components to carry out these instructions).

3.2. Characteristics

3.2.1. Functional / Performance Requirements

3.2.1.1. Flight Survey Area

The APRS shall have capacity to survey an area of ~1km in one flight.

> *Rationale: As required by the sponsor, the initial surveying flight must cover a defined plot.*

3.2.1.2. Payload Capability

The APRS shall have capacity to carry 1400 grams in addition to its weight.

> *Rationale: In accordance with the specifications of the drone it cannot exceed a total weight of 3680g.*

3.2.1.3. Movement Characteristics

The APRS aerial drone shall have a maximum yaw angle of 200 degrees and a maximum tilt angle of 35 degrees. This drone shall not fly faster than 15m/s.

> *Rationale: A characteristic of the Tarot 650 V2.1 drone details the above figures that the programs will not attempt to exceed.*

3.2.1.4. Flight Range

The APRS shall have capacity to travel 3.2 m away from the user starting location.

> *Rationale: The telemetry modules on commercial drones can only communicate at a functional maximum of this distance.*

3.2.1.5. Flight and Land Duration

The system aerial drone is operational for up to 25 minutes in the air. The land vehicle will be able to follow various paths for up to 30 minutes.

> *Rationale: This aerial drone contains 1 single 5000mah LiPo batteries. Base conditions through Tarot testing show that it can remain in the air stationary for 40 minutes. The Land drone will have a 4000mah battery that will allow it travel long distances.*

3.2.1.6. System Lifetime

The Drone will not be operational after a single flight path is followed. All components of the drone, the flight controller and the land drone will require a charge after operation.

> *Rationale: After each use of the APRS, components of the system require maintenance. The aerial and land drones' batteries must be charged before attempting to use again.*

3.2.1.7. System Networking

When transferring data between devices, the Aerial Pathfinding Reconnaissance System requires that the devices must be close enough such that the devices can be connected to a shared wireless local network.

Rationale: The APRS will utilize wireless local area networks to communicate between devices. In order to facilitate communication between devices on WLAN, the devices must be in relatively close proximity to each other.

3.2.1.8. Obstacle Detection

The APRS shall be able to accurately detect obstacles as tall as 5% of its altitude.

Rationale: As defined by the sensitivity of the sensors onboard the aerial drone, the APRS will be able to detect objects that meet a height requirement.

3.2.1.9. Optional Internet Availability

Optionally, processes may be offloaded to a GPU accelerated cloud server. To utilize this, internet access must be available to the local machine that is running the system's desktop application.

Rationale: In order to communicate with the hosted server, the local network node must have an internet connection.

3.2.2. Physical Characteristics

3.2.2.1. Drone Mass

The mass of the drone in the APRS shall be less than or equal to 3 kilograms.

Rationale: This is a requirement of the drone manufacturer to maximize flight time while maintaining all necessary functions of the drone. The drone body weighs 2180 grams and the all addition sensors weighing less than 800 grams.

3.2.2.2. Land Vehicle Mass

The mass of the land vehicle in the APRS shall be less than or equal to 5 kilograms.

Rationale: This is a requirement posed by our customer detailing that the system should be transferable by hand to any location.

3.2.3. Software Characteristics

3.2.3.1. Programming Style

Software subsystems written in Python shall conform to the PEP 8 Style Guide.

Rationale: The software components of the APRS should be written in a consistent programming style for maintainability purposes.

3.2.3.2. Source Commenting

Source code for software subsystems shall contain at least one comment for every 5 SLOC, and a documentation string at the beginning of each function definition.

Rationale: The software components of the APRS should contain sufficient internal documentation to understand and maintain the source code.

3.2.3.3. Error Handling

Upon any unrecoverable error conditions, software subsystems will use the standard exception handling capabilities of their source language.

Rationale: The software components of the APRS should utilize a consistent error handling mechanism to ensure ease of integration.

3.2.4. Electrical Characteristics

3.2.4.1. Inputs

a. The presence or absence of any combination of the input signals in accordance with ICD specifications applied in any sequence shall not place the user in a state of danger, place the APRS in a state of danger, reduce its life expectancy, or cause any malfunction, either when the unit is powered or when it is not.

b. No sequence of user command shall damage the APRS, reduce its life expectancy, or cause any malfunction.

c. In the event of fatal malfunction, the APRS shall be taken control by a user and may be stopped at any time to prevent such fatal error.

Rationale: By design, should limit the chance of damage or malfunction by user/technician error and by error of acts of god or unpredictable situations.

3.2.4.1.1 Power Consumption

a. The maximum peak power of the aerial system shall not exceed overcome the limits of the battery compartment on the drone. The voltage will not exceed 22.2V and in accordance with a maxima 4.5 A.

Rationale: A requirement drone systems to regulate power and maintain stable flight.

3.2.4.1.2 Input Voltage Level

The input voltage level for the APRS aerial drone shall be less than +29 VDC. At system power on the normal voltage limit may be exceeded at battery start.

Rationale: Aircraft bus specification compatibility, MIL-STD-704F

3.2.4.1.3 Charging and Capacity

The charge capacity of the battery will not exceed 99% of the 22.2 V battery operation. The maximum charging power shall not top 180W. The battery will display a low battery state when under 12.5% capacity.

Rationale: Limitations and safety features of LiPo batteries. By design to maintain battery and drone health.

3.2.4.2. Outputs

3.2.4.2.1 User Interface

The APRS will provide a UI encompassed in the system's desktop application that will update the user on the system's progression. When the system progresses from data collection to software analyzation, and ultimately to preparation of the land vehicle, the user will be visually updated within the UI.

> *Rationale: The APRS has many steps and allowing the user to interact through a simple UI will decrease the learning curve for the system.*

3.2.4.2.2 Low Battery Indication

The APRS will provide both an update to the UI and will enable a physical notification on the drone that it is incapable of traveling.

> *Rationale: To maintain the safety of the drone and the any nearby individuals.*

3.2.5. Environmental Requirements

The APRS shall be designed to withstand and operate in the environments and laboratory tests specified in the following section.

> *Rationale: This is a requirement specified by our customer due to constraints of their system in which the APRS is integrating.*

3.2.5.1. Pressure (Altitude)

The APRS can be operated within 0 to 120 meters above ground level, or 0 to 120 meters above a structure.

> *Rationale: This operating range is in compliance with FAA Small Unmanned Aircraft Regulations (Part 107).*

3.2.5.2. Thermal

The APRS shall be operated within a temperature range of -10° C to 40° C.

> *Rationale: As per the user manual, the battery that is included with the drone can operate within the temperature ranges of -10° C to 40° C.*

3.2.5.3. Inclement Weather Conditions

The APRS shall not be operated in any condition that involves inclement weather such as rain, snow, dust storms, high-speed winds, hail, or thunderstorms.

> *Rationale: The drone will not operate in any condition of inclement weather to ensure the safety of the aerial drone as it is not waterproof and cannot operate in extreme conditions.*

3.2.6. Failure Propagation

3.2.6.1. Aerial Drone Flight Failure

Upon error during flight time, the user should immediately suspend drone operation through use of the remote control provided.

> *Rationale: Per the requirements to the Texas A&M University ECEN Department, a user must be able to suspend the operation of the aerial drone using the provided remote control upon flight error.*

3.2.6.2. Low Battery

In the event of low battery, the drone will suspend operation and return either to the user or to a location where it can be recovered. The drone will not continue operation in a state of damage or irresponsible condition.

> *Rationale: To maintain the safety of the drone and the any nearby individuals.*

4. Support Requirements

4.1. Materials Provided

The following materials will be provided to the user.

4.1.1. Aerial Drone System

A fully encapsulated aerial drone system will be provided to the user. This will include sensors such as camera, LiDAR, and ultrasonic. A microcontroller onboard the drone will fulfill the interaction between hardware and software and provide functionality to receive and send data from the machine running the system's application. The aerial drone system will contain batteries and power distribution such that all electrical components are able to function properly.

4.1.2. Land Drone System

The land drone system will be provided to the user. This system will contain the vehicle's chassis, wheels, and motors. A microcontroller will be attached to the chassis so that software can be implemented to drive the motors and receive data from the local machine running the system's application. The system will contain batteries and power distribution such that all electrical components are able to function properly.

4.1.3. Desktop Application

Software will be provided to the user that will serve as the main interface to the user. The software will provide capabilities such as device networking, data transmission and analysis, and control over the system's processes.

4.2. Local Machine Requirements

The user must possess a local machine such as a laptop that is capable of running the system's desktop application. The recommended specifications are as below:

Processor: Intel Core I5 7th Generation or equivalent
RAM: 8GB DDR3
Operating System: Windows 10
Storage: 5GB

4.3. System Malfunction or Destruction

Upon destruction of the aerial or land drones, a professional will be required to service the affected system. If at some point a portion of the system malfunctions, the user will be updated within the UI, and will be able to make changes such as restarting or suspending the system.

Appendix A: Acronyms and Abbreviations

APRS	Aerial Pathfinding and Reconnaissance System
GPS	Global Positioning System
GUI	Graphical User Interface
ICD	Interface Control Document
SLOC	Source Lines of Code

Aerial Pathfinding Reconnaissance

Jason Gilman
Max Griffith
Mark Johnson
Dilanka Weerasinghe

INTERFACE CONTROL DOCUMENT

REVISION 1 – Draft
20 September 2020

INTERFACE CONTROL DOCUMENT
FOR
Aerial Pathfinding Reconnaissance

PREPARED BY:

Jason Gilman	Date
Max Griffith	Date
Mark Johnson	Date
Dilanka Weerasinghe	Date

APPROVED BY:

Jason Gilman	Date
Prof. S. Kalafatis	Date
T/A	Date

Change Record

Rev.	Date	Originator	Approvals	Description
1	9/20/2020	Jason Gilman		Draft Release

Table of Contents

List of Figures

1. Overview

This document will provide specifics into how the Aerial Pathfinding Reconnaissance System (APRS) will be implemented. It will also describe the APRS subsystems physically and electrically, providing information on how the system will function. The document will define the interaction between the subsystems, and the requirements to enable functionality.

2. Definitions

2.1. Definitions

REST	Representational State Transfer
APRS	Aerial Pathfinding Reconnaissance System
CV	Computer Vision

3. Physical Interface

3.1. Weight

3.1.1. Weight of Aerial Drone

The aerial drone system will weigh less than 3680g. This is to allow an average person to be able to transport and setup the drone. This weight includes the drone chassis, sensors, battery, power distribution components, and microcontroller.

3.1.2. Weight of Land Drone

The land drone system will weigh less than 6.1kg. The land drone system will need to be placed at the user's defined start point, so the land drone's weight was defined so that a user could transport and setup the drone, similar to the aerial drone. The weight includes all necessary parts for the land drone's functionality.

3.2. Dimensions

3.2.1. Dimension of Aerial Drone

The sensors, battery, power distribution components, and microcontroller onboard the drone will fit within the footprint of the drone's chassis. The drone has a diagonal wheelbase of 66.04cm with a vertical footprint of the drone is 38.1cm.

3.2.2. Dimension of Land Drone

The land drone's chassis will have motors, wheels, a motor controller, and a microcontroller mounted to it. The land drone can have a volume ranging from 0.03m^3 to 0.07m3.

3.3. System Setup Locations

The following section will define the requirements for where each system will need to be placed in order for the Aerial Pathfinding Reconnaissance System to function properly.

3.3.1. Placement of Aerial Drone

The aerial drone must be placed on a level surface with no obstructing objects within 0.5m. of the zone. The drone must be placed such that a wireless local area network connection hosted on the drone can be connected to by the local machine running the system's application. This placement will be used for takeoff and landing of the aerial drone, as well as transmission of data to the local machine.

3.3.2. Placement of Land Drone

The land drone must be placed such that a wireless local area network connection hosted on the land drone can be connected to by the local machine. The land drone should not be placed on a surface with a slope greater than 30 degrees or 58% grade. This placement will be used as the drone's start point.

3.3.3. Placement of the Local Machine

The user's local machine must be in proximity to the aerial and land drones initial locations such that the machine can connect to wireless local area networks hosted by each machine.

4. Electrical Interface

4.1. Primary Input Power

4.1.1. Aerial Drone

The standard power voltage provided to quadcopters and drones is 22.2V. The drone will be using a single 22.2V lithium ion battery with a capacity of 5000maH. A lithium ion battery was chosen due to its weight and efficiency in comparison to other battery types such as lead acid. A single one of these batteries will power the aerial drone in normal operation and for longer flights it may take 2 of these batteries in parallel. Below, in Figure 1, is a graph detailing the voltage drop off of a lithium battery versus a Lead-Acid.

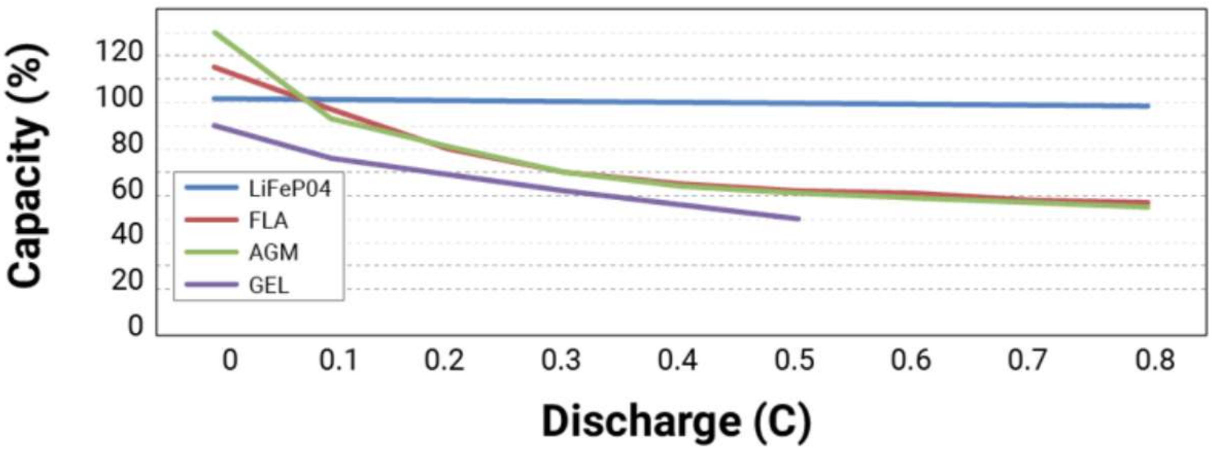

Figure 1. Lithium-Ion vs Common Lead Acid

The batteries will be attached to the drone with a custom fitting bracket that will securely hold the batteries during flight. The batteries must be fitted correctly to ensure no electrical malfunction occurs in the system.

4.1.2. Aerial Drone

The land drone will be using a high capacity lithium ion battery. To see why a lithium ion battery was chosen, see the above description on the benefits of lithium ion vs other battery types. Because of the varying power inputs on the digital controllers attached to this drone a series of boost and buck converters will be used in addition to the 4.2V battery. The batteries will be attached to the drone using a custom fitting bracket that will securely hold the batteries during land travel.

4.2. Signal Interfaces

4.2.1. Aerial Drone Connection

The onboard SDK for the Tarot 650 V2.1 utilizes serial ports on the PCB to communicate with the drone controller. The drone controller has high level methods to communicate with the telemetry module and flight controller. The microprocessor will issue commands to either the TTL UART port or the USB port. Several broadcasts can be received from the drone. Due to the large amount of data generated by the drone this application can record the data to any controller at the users need. The drone can offload tasks using the OSDK to the controller to perform our autonomous flight and data retrieval tasks.

4.2.2. Aerial Drone μC

In accordance with the requirements of the Tarot a handful of controllers can be utilized. A stm32 or Arduino can send the required information back and forth from the drone and to the server. The stm32 has a faster processor and thus can handle more tasks. Because of the low frequency of the broadcasts incoming from the drone either controller will fulfill the project goals. The microcontroller will run the Linux operating system to communicate with the drone. The drone can also be directed with a Mission Controller Program without any object avoidance application.

4.2.3. Land Drone Connection

This drone will be using a stm32 or Arduino to handle control of the motor controller and other system connections. The land drone will utilize a sabretooth (ii) motor controller to handle power distribution and control of the motors. The Land Drone will have a wireless connection to the user’s local machine using a WIFI module further outlined in Section 6.

4.3. Sensors

4.3.1. Aerial Drone Sensors

4.3.1.1. Camera

A camera will be used by the μC to record video of the ground. This camera will be directed straight down and will not have any connection to the flight controller on the drone. Only to the 3rd party controller (stm32/Arduino).

4.3.1.2. Ultrasonic / Lidar

Attached to the front of the drone will be a series of ultrasonic or lidar sensors to handle object detection and perform anti-collision movements. Because the drone will only be performing small movements in accordance with telemetry data gathered it will only travel forward and rotate. Therefore, we do not need 360 vision, instead the drone will see the immediate area in front and maneuver around obstacles slowly.

4.3.1.3. GPS

The flight controller will collect telemetry data such as GPS. The highly accurate GPS location data is what is used to send the drone to specific locations. The μC will send coordinates to the flight controller to follow.

4.3.2. Land Drone Sensors

4.3.2.1. GPS

The μC will collect telemetry data such as GPS. The drone will use the GPS to determine its location in accordance with the map provided and relay this to the user.

4.4. User Control Interface

4.4.1. Physical Interface

Both the land and aerial drones will indicate to the user the effective battery power and ready state. Both systems will include an indication light for powered on and system ready. The system ready indicator will show the user that it is prepared for remote instruction and operation.

5. Software Interface

5.1. Computer Vision and Mapping

5.1.1. Inputs

The Computer Vision and Mapping subsystem will take one or more videos, in MP4 format. For each field-of-view of the ground along the path, there are expected to be two clips which hold the following properties:

- The length of these videos should be long enough for a computer vision (CV) algorithm to have enough visual information to eliminate moving objects from detection.
    - For example, to ignore pedestrians, the clips' length should be at minimum the time it takes for a pedestrian to move a few feet from their original position.
- These clips should be taken sufficiently far apart (depending on the drone's altitude) to provide strong enough visual cues to infer depth with a second CV algorithm.
    - For example, a drone at 10m altitude will likely want to record their two clips at least 1m away from each other for improved depth estimation.
- Each clip's metadata should include a timestamp so that the correct sequence of video inputs can be determined.

Furthermore, the drone's flight altitude and land vehicle's wheel radius must be known to determine the scale at which obstacles are to be detected.

5.1.2. Error Conditions

The Computer Vision and Mapping subsystem will report an error through standard Python exceptions when a top-down map of the environment cannot be produced.

5.1.3. Outputs

The Computer Vision and Mapping subsystem will output a two-dimensional Boolean matrix representing the locations of obstacles, with each element representing approximately one pixel of the final top-down image of the environment. In this case, the term "obstacle" refers to any region which presents too great of a change in depth, for which a path through that region is not possible for the land vehicle.

The Computer Vision and Mapping subsystem will also output a top-down color image of the surveyed environment intended for use in detecting the starting location of the ground vehicle.

5.2. Pathfinding and Ground Vehicle Detection

5.2.1. Inputs

The input of this system will be the two-dimensional Boolean matrix that represents the location of obstacles as outlined above.

5.2.2. Error Conditions

An error occurs if and only if no possible path is discovered from the Boolean map. The pathfinding and ground vehicle subsystem will report the error though standard Python exceptions and throw the exception to the user interface subsystem.

5.2.3. Outputs

There will be two outputs of this system: one pertaining to the map and the other pertaining to the drone.
The Boolean map will be changed throughout a greedy algorithm. The greedy algorithm will work as follows:

- With respect to a final path heading from South to North, the greedy algorithm will choose the straightest path that is valid. Once a northern path is deemed non-traversable, the drone will select a west path and repeat this process. If neither are deemed traversable, the drone will select an east path and repeat the process.

- If anytime during this calculation a dead end is spotted, the algorithm will mark the current index as false and travel back to the previous position and start the algorithm again. Once the end index is located, the algorithm will be complete, and the path will be sent to the drone.

- If there is no valid path, the Boolean map will be marked as entirely false by the algorithm itself.

The goal of this greedy algorithm is to find the shortest locally optimal path in a quick manner.

The second output will be the path of the drone. Once this new Boolean map is sent to the drone, the drone will move to the end position following the map that has been sent if there is a possible path. If no possible path is found, the error condition is met, and the drone will respond accordingly.

6. Communications / Device Interface Protocols

6.1. Wireless Communications

A combination of communication through wireless local area networks (WLANs) and the internet will be used. Communication between the user's local machine and the drones will be through the use of a WLANs. If the user meets the requirement of internet access on their local machine and wishes to offload processing to a GPU accelerated server, the internet will be used to send information from the user's machine to the server. All wireless communication will follow IEEE specification 802.11.

6.2. Host Device

A local machine, such as a laptop, provided by the user will serve as the host device for the system's desktop application. This device will allow for networking throughout the system. Optionally, the system's software analysis will occur locally or on a GPU accelerated server.

6.3. REST Architecture Constraints

The system will implement a RESTful API so that software can be utilized within the systems operation. GET and POST application state transitions will be used to send and receive data throughout the system.

Aerial Pathfinding Reconnaissance

Jason Gilman
Max Griffith
Mark Johnson
Dilanka Weerasinghe

EXECUTION AND VALIDATION PLANS

REVISION 1 – Draft
20 September 2020

Execution Plan

Milestone	Assigned To	Week 1 24-Aug-20	Week 2 31-Aug-20	Week 3 7-Sep-20	Week 4 14-Sep-20	Week 5 21-Sep-20	Week 6 28-Sep-20	Week 7 5-Oct-20	Week 8 12-Oct-20	Week 9 19-Oct-20	Week 10 26-Oct-20	Week 11 2-Nov-20	Week 12 9-Nov-20	Week 13 16-Nov-20	Week 14 23-Nov-20
Understand Project	All														
Initial planning/research	All														
Draft Conops Report	All														
Source Lab Materials (Drone/Sensors)	All														
Draft FSR Report	All														
Draft ICD Report	All														
Draft Exection Plan	All														
Draft Validation Plan	All														
Prepare Midterm Presentation	All														
Design Server-Side Architecture	Jason														
Remove Moving Obstacles from Data	Mark														
Increase familiarity with Python	Max														
Implement Sever-Side	Jason														
Map Environment from Data	Mark														
Create algorithm for valid accessible path	Max														
Research/Order Parts	Dilanka/Jason														
Create error handling for inaccessible path	Max														
Implement Client-Side/UI	Jason														
Estimate Depth on Path Sections	Mark														
Research Drone SDK / stm32 / Sabretooth	Dilanka														
Add flight mapping functionality	Jason														
Research software interaction with drone	Max														
Design Aerial/Land Electrical Schematics	Dilanka														
Program manual override of drone	Max														
Testing Electrical Off Drones	Dilanka														
Merge Depth Estimates with Map	Mark														
Build Air Drone	Dilanka														
Build Land Drone	Dilanka														
Prepare Status Update Presentation	All														
Implement Server on drone	Jason														
Test Flight of Air Drone	Dilanka														
Normalize Depth Field for Drone Altitude	Mark														
Program movement of drone	Max														
Test Travel of Land Drone	Dilanka														
Communicate with server on drone	Jason														
Derive Gradient Field From Elevation Map	Mark														
Program: Data Retreival of Air Drone	Dilanka														
Test Drone Simulation Programs	Dilanka														
Test Data Retreival From Drone	Dilanka														
Implement Obstacle Delineation	Mark														
Program drone path from map input	Max														
Package Software/Create Installer	Jason														
Prepare Final Presentaion	All														
Test and modify Air system	Dilanka														
Program: Autonomous Flight of Air Drone	Dilanka														
Draft Final Report	All														
Prepare Final Demo	All														

Legend
Completed
Deadline
Planned Work Time
In Progress
Late

Validation Plan

Legend	Complete	Incomplete

Subsystem	Test	Deadline	Status
Data Collection and Drone Control	Aerial Drone flies from Point A to Point B	8-Nov	
	Drone stops at midpoint locations on the map and records video and GPS location	8-Nov	
	Drone stops at user shutoff switch	15-Nov	
	Land Drone Moves	2-Nov	
	Drone detects obstacles in front view using Ultrasonic Sensors.	31-Oct	
	Electrical Systems Connected without Error	23-Oct	
	Drones Built and Power on	27-Oct	
Computer Vision and Mapping	Removes moving objects from videos to produce still frames of path sections	27-Sep	
	Produces a single image map of the environment from overhead views	4-Oct	
	Infers depth of a path section from an overhead stereographic perspective	11-Oct	
	Merges individual depth maps into a single depth map of the whole environment	25-Oct	
	Normalizes depth fields for altitude	1-Nov	
	Derives a gradient field from a height map	8-Nov	
	Delineates obstacle boundaries from an elevation gradient field	15-Nov	
Pathfinding and Drone Detection	Greedy algorithm for traversable path gives a valid path given a valid input	4-Oct-20	
	Create error handling for an inaccessible path	7-Oct-20	
	Manual override of drone works by controller by controller	23-Oct-20	
	Drone can move given basic commands basic commands	1-Nov-20	
	Movement of drone given from algorithnically designed path	13-Nov-20	
Device Networking and UI	Communicate with server using HTTP methods (Postman)	31-Sep-20	
	Communicate with server through UI	8-Oct-20	
	Complete intuitive UI design	8-Oct-20	
	Plot flight endpoints and return to user	15-Oct-20	
	Communicate with drone's server using HTTP methods through UI	5-Nov-20	
	Install software via encapsulated windows installer	16-Nov-20	