UAV Assisted Energy Delivery

PROJECT PLAN

Team: 43
Client/Advisers: Randall Geiger & Degang Chen
Alexandra Lowry
Brendan Rohlik
Connor Wehr
Garth Flaming
Kaitlin Maass
Kevin Angeliu

Team Email: sdmay19-43@iastate.edu
Team Website: http://sdmay19-43.sd.ece.iastate.edu

Revised: December 2, 2018

Table of Contents

List of Figures

List of Tables

List of Symbols

List of Definitions

UAV: Acronym for Unmanned Aerial Vehicle, such as a drone

FAA: Acronym for Federal Aviation Administration

LiPo: Acronym for Lithium Polymer, a type of battery

DroneKit: API (see definition of API) used to code drone

API: Acronym for Application Programming Interface

1 Introductory Material

1.1 Acknowledgement

With Gratitude and Enthusiasm, we thank Dr. Geiger, Dr. Chen, and the assistance from the ECpE department for the excellent advice that we were given over the project.

1.2 Problem Statement

At the current pace of technological advancement, more and more devices are requiring more energy to operate. Unfortunately, batteries and other energy supplying technology aren't advancing as rapidly. There's even the issue of a device might not even be connected to the power grid, which can be quite costly to connect. This is where our group comes in.

The goal of our team is to solve this energy problem and UAVs or drones are an excellent solution. The basic idea is for a network of drones to deliver energy to these remote devices. Our plan is to acquire a drone which we can customize and reprogram for automated flight. This drone will need to be able to fly from one location to another, assuming no obstacles, land safely, deliver lasting energy to an electronic device, and return home. This project will also require the design of a charging station for the drone, as well as a method for the drone to dock with a device in order to supply energy.

1.3 Operating Environment

The UAV will be expected to survive and operate weather conditions harsher than our problem statement assumes. Harsh conditions are such: high winds, rain, cold and hot temperatures, etc. However, that is beyond our scope for now. The goal of our group is more proof of concept, than actually implementing weather safety.

1.4 Intended Users and Intended Uses

There are several different users who will have their own specific uses for the UAV. The main users will be individuals in industries who need to deliver energy and data to multiple tethered and untethered nodes. They may want to have more options than a private individual user as to how the drones deliver their energy and data, such as specific ordering of which nodes are delivered first, specifying efficiency of delivery, and delivering to multiple nodes with multiple drones on a large scale.

Additionally, there will also be private, individual users who will be able to call for use of a drone at any time. A drone will then come directly to the user's location, and perform whatever action the user specified. Most often, the action will be to deliver energy to and charge some kind of electronic device.

1.5 Assumptions and Limitations

Assumptions:

- Drones will be able to navigate, dock, and deliver energy or data fully autonomously
- Drones will be able to deliver to multiple nodes before having to return to its base and charge back up
- Power stations for drone recharging can be assumed to be able to gather energy without needing to be recharged themselves, such as connected to the power grid
- System will be low-cost to implement

Limitations:

- Drones must follow regulations already in place, such as avoiding emergency response teams and avoiding no-fly zones
- Drones must avoid obstacles like power lines and buildings
- Due to regulations, we cannot fly the drone within the school zone

1.6 Expected End Product and Other Deliverables

The end product will be a drone that can fly, unassisted, to and from both tethered and untethered nodes. The drone will then attach to the node, transfer either data or energy, and either continue on its delivery path or return to a power station, which will recharge the drone. We will have a drone which can autonomously fly from one point to another at a distance of approximately 100 yards. We will also have a power transfer system, which will be built to allow the delivery of energy to the drone and electronic device. We hope to have completed the autonomous docking feature to allow the drone to shuttle data or energy between multiple tethered and untethered nodes.

Drone Automation: Drone automation is an ongoing process. Our first focus is very basic drone automation, e.g. Takeoff and Landing. The following focus will be directed flight from one location to another. Once the drone arrives at its location it will need to be able to accurately land on the target location. After a reliable connection is made, there needs to be a sleep time frame where the drone charges the device before leaving. When the device is done charging, the drone should take off, and return home.

- Automate takeoff and landing
- Sequential flight path to return home after delivery
- Landing accuracy
- Tests with docking station
- Final Presentation Preparations

- Improvements proposal
- Document Improvements

Docking Station: To allow the drone to land and establish a reliable connection for energy delivery the docking station is a necessary device. First, we will need to create prototypes for establishing a reliable connection. Once connection is capable of being made, it must also be able to break that connection so the drone can take off. Once achieved, the physical docking station can be designed.

- Research Power Methods
- Connections Made/Broken
- Docking Station Designed
- Station built
- Drone Tests
- Final Presentation Preparations
- Improvements proposal
- Document Improvements

2 Proposed Approach and Statement of Work

2.1 Functional Requirements

Requirement 1: The drone will be able to take off and fly to a specified location autonomously.

Requirement 2: Drone will dock with landing pad for power transfer autonomously.

Requirement 3: Drone will be able to deliver power from the payload battery to the landing pad.

This project is made out of three different pieces that work together in order to accomplish the objective of having UAV power delivery. The first piece is to **fly** the drone to a remote location. In order to do this the drone will have to have the ability to fly autonomously to a given GPS coordinate. The drone will fly up to an altitude where it is expected to not have to run into any obstacles. We do not have object avoidance in the scope of this project and will not implement anything to avoid obstacles besides where we are testing and the previously mentioned flying up to a high altitude. This is the first step for the drone to complete its requirements and is needed for anything else to function. The drone will also receive the coordinates of the node before it takes off and will map a route to it. The drone will be able to receive commands as necessary.

The second part is to **attach** the drone to the remote node. In order to do this, there needs to be a secondary flying mode that will allow the drone to precisely land. To complement this there is also image processing to be able to identify where the landing pad is and how close the drone is to it. With the image processing the drone will be able to calculate how centered it is on the landing pad or how far off it is. Once it has calculated how far it is off it will adjust and recalculate to see how close it is. It will be able to repeat this process until it has landed on the node and as it lands makes a connection to the node. Here it will attach to make a valid connection to the node.

The third part is to **deliver** energy to the node. During this process the power in the drone will transfer over to the node. The drone will keep track of how much charge it has so that it knows when it is empty so that I can leave and return to base. During this process the drone will efficiently and quickly transfer power to the node. It is important that this is a quick process because if the drone's battery gets to low it will go back to base before it has transferred all of the payload's energy. The drone will also remember the place it left from in the first step so it can return there in this step.

2.2 Constraints Considerations

For our constraints, the biggest one is that we are restricted on where we can fly the drone. A point to consider for our project is flight legality. According to the FAA (Federal Aviation Administration), it is deemed unethical and illegal to fly drones or other similar UAVs with 5 miles of airports. Without a license we are legally unable to fly our drone on campus, we will need a location that's legal to fly drones and etc. Our group will need to follow the standard protocols to provide safety for us and the people around. Therefore, we need to test the drone in an environment that will qualify for space needed to fly, distance from the public as to prevent collisions with people or their property. [Ref. 1, 2, 3]

Another big constraint is that there is not a lot of time to fly the drone in good weather, as it is really hard to fly the drone when it is cold and snowy out, which will be most of the year. This means that we need to get as much done as we can before all of that sets in so we can see the big problems we need to work on over the winter. As well as in the spring when it rains it will be unfitting for the drone to fly. There is also a point at which the wind speeds could be too high for flying to be feasible. Overall, with flying the weather will always be important.

Another constraint that we will have to work with for the duration of our time working on the project is the drone's battery life. It will give us our range for as far away we can deliver energy. It also effects how large our payload can be because a heavier payload will take more power to lift.

We also have to worry about safety. When we have a flying drone with four spinning blades moving at high speeds, it is possible that someone could get hurt. We have to keep in mind where we are flying it and when, and to always have a backup plan to turn the drone off or restrain it in some capacity. It is also important to keep the drone in sight at all times. It can be easy to forget about safety when working on other things but it always remains important.

We also need to worry about areas that do not have a strong signal or no signal at all. During our design we need to make sure that the drone can handle times when it gets in a dead zone. Alternatively, we could decide that every location we go to does not have a dead zone, thereby circumventing the possibility of the drone having no signal.

The drone is also prone to breaking, especially the propellers. When flying the drone, it is very possible, even probable, that the drone will get into a few bumps and accidents. During these it is easy for a propeller to snap and need to get replaced. We will keep this in mind and make sure that we have replacement parts when reasonable or order more pieces for any unexpected events.

The drone itself has limitations. The drone is only able to carry so much weight at a time. This demands that we keep our payload under a certain weight. The drone only has so much power in its rotors so it won't be able to handle certain winds. The drone also only has a certain amount of precision in its movements. These are all things that we will keep in mind and work around to complete this project.

We are also limited by our own inexperience. While we are interested in working on this project, we don't have the experience to jump straight into it. We will have to do research at each step and figure things out as we go. This is not a huge problem but we need to factor issues that we will have into our plans and schedules. Since we are so unfamiliar it is only natural that we will have some issues with getting things working when first adjusting to them.

2.3 Technology Considerations

The Intel Aero drone will give us more freedom in with our programming and battery attachment. Intel offers a wide variety of open source code that goes along with it. This will help us program the drone easier and let us manipulate more aspects of the drone rather than if we chose the Phantom drone.

Using a magnetic connection for our power transfer will ensure that there is a secure connection between the battery pack on our drone and the landing station. The landing station will hold one end of the magnetic connection and the drone will hold the other. When the drone comes close to the landing station due to our image processing software it will magnetically connect to the landing station and transfer energy to it.

Our landing station will feature a place for the drone to land and also charge its internal battery. This will ensure the drone will have enough power to take off, transfer energy to the node, and come back on one single battery charge.

2.4 Technical Approach Considerations

The Intel Ready to fly drone will allow us to program the drone to whatever we need it to do. The drone uses an open source code so we can manipulate the landing procedure, how the drone will fly, making the drone autonomous, and where the drone will land. Being able to choose how the drone will land is very important because we need to be able to be precise within 1-2 centimeters when landing. The landing pad will have a big platform but the needs to be able to center itself over the landing pad so the connecting wire lines up with the magnetic connection on the landing pad. Making the drone autonomous is essential to the completion of this project.

The drone will need an image processing program to be able to detect the landing pad on the ground. The landing pad will need to have a color contrast to the ground so the drone's image processing program will be able to find the edges of the landing pad and land safely on top of the landing pad. Image processing will help the drone decide the difference between the square landing pad and the magnetic connection on the landing pad.

We need to consider the LiPo battery we are using and ensure the battery does not overheat in any way. We chose the LiPo battery to act against that fact but there is still a possibility if the drone crashes that the battery could overheat.

We are using a rechargeable battery to transfer the energy from the drone to the landing pad. The drone has a payload of 2 pounds, so we need to 3D print a carriage to hold the battery to the drone. We are 3D printing it because the plastic will be relatively light, so it will not go over the payload limit of the drone. The rechargeable battery takes a full charge at its initial station, and when given a signal from an untethered node it will autonomously fly to the landing pad and transfer the energy. Once the drone is finished, it will fly back to the initial station and recharge its flying LiPo battery and the rechargeable battery.

2.5 Safety Considerations

The biggest safety consideration is that we need to keep the away from other people, trees, and water. To achieve this, we need to be mindful of where and when we fly the drone, so we can avoid all of these hazards.

2.6 Task Approach

When given the project specifications, we needed to look into ways to implement this project. We had some of the previous groups work, so we needed to look over that and see if they were the best methods, or if there were better methods out there.

Our first thought was Amazon PrimeAir, where they used drones to deliver packages. They have a great idea for delivering packages and we want to do the same but with power. We knew that we needed to use their idea of GPS navigation. That is the only way to really automate the flight process to get to a certain area. They also have a weight limit which was something we needed to think about as well. You could also see in the video about AmazonAir that the drone landed on a pad with a type of QR code on it. We knew from the past group that they used image recognition and it worked very well. So, we looked into the idea of using a QR code for image recognition, but ultimately we will go with colors for image recognition.

While looking into AmazonAir, we found other companies that want to advance with drone deliveries. Places like Zipline use drones to deliver blood, medicine and other medical supplies. This is also very similar to AmazonAir using GPS to navigate to a target location and deliver a product. When looking into GPS navigation it seems the only way to get that information is to use radio signals to get information from satellites. We might also need to think of making a 3D map so the drone will consider obstacles and the height it needs to fly. We know that the drone needs to fly at a certain height per regulations, but if to be used in a real-world situation, it will also need GPS routing with obstacle avoidance. One way that we discovered in 3D mapping so it gets a layout of the land. So, we will either need to drone to use map data to avoid buildings, and obstacle avoidance for power lines and such, but it might also be nice to have a 3D map to make life easier. [Ref. 6]

Other considerations we will have is if we want the drone to go to multiple way points on one journey. This would be a stretch goal for the project, but we would want to think about this earlier in the project. [Ref. 7]

For technology we will use on the drone the first consideration is the software we want to use to code. We will choose between MavRos and DroneKit. DroneKit is easier high-level coding and is endorsed by this drone's manufacturer. MavRos is low level code that may provide more overwriting features, but is harder to learn and understand.

We will also need to make sure the landing pad is visible in grass, snow, cement, etc. so we need to think about high contrasting colors. Then we need to decide if the drone can land in any orientation or only one orientation on the landing pad for it to correctly transfer power.

2.7 Proposed Design

This project is made out of three different pieces that work together in order to accomplish the objective of having UAV power delivery. The first piece is to fly the drone to a remote location. In order to do this the drone will have to have the ability to fly autonomously to a given GPS coordinate. The drone will fly up to an altitude where it is expected to not have to run into any obstacles. To do this we have chosen the Intel Ready to fly Drone, as it has open source coding, and the greater freedoms of what to push to the drone compared to other drones. Furthermore, we are able to connect to the cameras easily with this drone and attach different sensors as we see fit easily with the open source code available. The drone will also receive the coordinates of the node before it takes off and will map a route to it. The drone will be able to receive other commands as necessary. To serve these commands to the rest of the drone, we will be using the DroneKit API, coded in Python. We have chosen this API due to the overall simplicity of the code as well as the high-level design and control it gives. We did not want to dig too deep into the functions of the drone due to what we are doing is not too complicated and having a higher-level code for commands will be good enough for us.

The second part is to attach the drone to the remote node. In order to do this, there needs to be a secondary flying mode that will allow the drone to precisely land. To complement this there is also image processing to be able to identify where the landing pad is and how close the drone is to it. The image processing can be done with an Intel RealSense camera made for and attached to the drone. This camera will then use the image processing code, such as edge detection, color mapping, and depth sensing to see where the desired landing is, and use these to make corrections to be over the target as the drone begins to land. Once it has calculated how far it is off it will adjust and recalculate to see how close it is. It will be able to repeat this process until it has landed on the node and as it lands makes a connection to the node. Here it will attach to make a valid connection to the node. The autonomous flying and the attachment procedure can be seen on the left side of the tree in Figure 1, our top-level design diagram.

Another part of the precision landing is the landing pad of the drone. The landing pad will be used to have the transfer point of the energy, and will be in the specifications of the image processing so the drone can sense that it is a landing point once it reaches there. Therefore, the landing pad will be painted a contrasting color to the surrounding environment of the landing pad, as well as have a unique shape cut into it for the drone to detect. The transfer point for the energy will be in the middle of this unique shape, as it will be easier for the drone to position itself when like this.

The third part is to deliver energy to the node. During this process we will have a separate battery source from what the drone is powered off of (this battery will be called the payload battery) have its energy transferred from the drone to a specific point on a landing pad. The payload battery will need to be small enough so it does not impar the landing of the drone, and will be placed so that the center of gravity on the drone is not changed with the battery. For this, we have chosen the battery to be a simple pocket battery similar to phone chargers, for their size and low weight. We will create a custom carriage to house the payload battery that will be attached to the drone. Like the payload battery, this will need to be small and lightweight to not impar the drone operation. The cord used to attach the battery to the landing pad will be a magnetic cord that can be easily attached and detached, similar to an Apple Mac charger. This is used to make sure the drone can land and take off from the landing pad and attach to the transfer point without human help. This can be seen on the right side of our tree in Figure 1.

During the entire process the drone will efficiently and quickly transfer power to the node. It is important that this is a quick process because if the drone's battery gets to low it will go back to base before it has transferred all of the payload's energy. The drone will also remember the place it left from during its first take-off so it can return there after the power transfer.

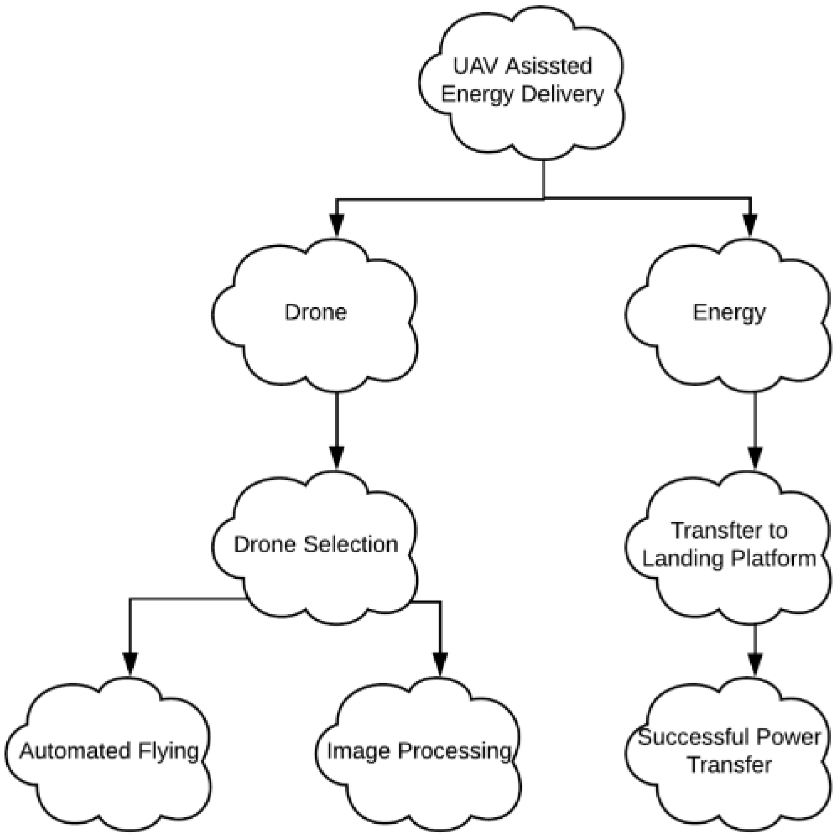

Figure 1

2.8 Previous Work and Literature

Currently, there is no official product that can deliver energy through a drone to a specific location as is required for this project. However, there have been previous groups who attempted this method of delivery. The last group to work on this project used the same Intel drone as the one we are using, and focused on the same idea we are working on: to autonomously deliver energy using our drone of choice. They were able to successfully implement image processing and landing station requirements, but were unable to meet all autonomous requirements. As such, we will utilize their project image processing, at least in some capacity, as well as some of their landing station ideas.

The group before them used a Dji drone. They were able to get the drone to autonomously land on the platform but had a very inefficient way to transfer energy. They had steel wool on the legs of the drone along with all over the landing pad and transferred electricity through the steel wool. This worked but was too inefficient with the small amount of energy that the drone could carry for it to be viable. Since they used the Dji drone we will not be able to repurpose their code.

Coding for drones is not particularly new. However, a lot of code that is written and out there will only work with very specific drones. While it is easy to get random code out there to get a drone to do this or that it is important to also keep in mind what drone you are using and if it is compatible.

Additionally, while there are no official solutions, there are APIs and libraries that already have solutions for specific, smaller parts of the project, such as how to arm and disarm the drone, fly the drone to a specific location, and cause the drone to land. The main API we are working with is DroneKit, which already has the necessary functions that we can implement into our overall code to solve our automation issues. Some of these functions include arming the motors, taking off, and landing. There is also a Git repository that has steps to follow and code for autonomous drone programming in Python, which will be a useful tool for the maneuvering of the drone while in flight. [Ref. 3]

2.9 Possible Risks and Risk Management

Title: Difficulty of drone operations
Risk Management: Mitigate risk
Information: The drone is very difficult to operate using the remote controller, so it will likely be even more difficult to operate autonomously using code. There is a possibility the drone can collide with objects, causing the drone to break. Due to the difficulty of operations, the drone is also inaccurate when flying, which is dangerous because during testing, people and surrounding objects could be physically damaged if the drone falls.
Mitigation Action: In order to mitigate this risk, we are going to fly in open fields where there are no objects for the drone to hit, not fly far off the ground so that the drone does not break if it falls due to some failure, and include some object avoidance in the code.

Title: Lack of knowledge about drones
Risk Management: Mitigate risk
Information: None of us have worked with drones or DroneKit before, so we are starting from scratch in regards to how we code. This could cause significant delays in our progress because the coding team not only has to learn the API we are using, but also learn how to interface with the drone, as well as learn how to code in Python.
Mitigation Action: In order to mitigate this risk, we have the entire four-person code team researching how to use DroneKit and any additional resources that DroneKit can be integrated with that we may utilize in our project.

2.10 Project Proposed Milestones and Evaluation Criteria

As discussed before, we will break down the project into three main priorities, which shall have their respective milestones. For the drone automation, we hope to have the drone flying autonomously short distances and being near a predetermined destination by the end of the first semester. We hope to have the payload battery, landing pad, payload carriage, and power transfer cord finished by the end of the first semester. The auto-docking will be the most intensive part of the project, as that will use image processing for it to align. The evaluation for all of these milestones will be done using the testing plan outlined in section 2.12.

2.11 Project Tracking Procedures

We are keeping track of where we are in the project by having weekly meetings with our advisors. During these meetings we individually show what we have contributed to the project. We then talk about what will happen in the next week, and how we plan to achieve each goal.

2.12 Test Plan

Requirement 1: The drone will be able to take off and fly to a specified location autonomously.

Test Case: For this requirement we want to test and see if the code we load onto the drone will allow it to take off and fly to a predetermined location without any manual control.

Test Steps:

1. Load code onto drone and activate it from ground control resource.
2. Observe how the drone flight for stability and any difficulties flying, such as swaying and wrong direction of flight.
3. Measure how far away the drone lands from the predetermined destination set in the code, measured in centimeters.

Expected Results: The drone will take off and fly to the specified area approximately. It does not have to be exact destination, within 1 meter of the target. The drone should also have a stable flight, no swaying or sudden drops when flying. There should be some telemetry data for the flight such as battery level, altitude, speed, etc.

Requirement 2: Drone will dock with landing pad for power transfer autonomously.

Test Case: We want to have the drone as it is landing align itself correctly over the landing pad and plug to ensure there will be a connection to deliver energy.

Test Steps:

1. Have drone already in the air over the target destination.
2. Observe how the drone aligns itself over the target as it is landing, observe any big changes over a meter or sudden drops to land.

3. Measure how far off the target the drone is in centimeters, and see if we can get footage of camera to see what used to align itself to make tweaks.

Expected Results: Drone orients itself to where it expects to land as well as the actual landing points to compare to where we set the landing point to be. Landing should take place within a couple of centimeters of the target location. The drone will have a smooth landing procedure, no sudden drops or movements.

Requirement 3: Drone will be able to deliver power from the payload battery to the landing pad.

Test Case: Have the cord used to connect the payload battery connected to the payload, and connect the plug together to get a voltage reading.

Test Steps:

1. Connect one end of the cord to the payload battery, the other end will attach to the landing pad plug.
2. Measure the voltage reading on the landing pad side to see if there is power transfer.

Expected Results: Voltage level reading on the landing pad side of the plug is within 1V when the payload battery is not attached to the landing pad plug.

Testing using the drone will be done with the payload battery and carriage attached to see how the extra weight can impact flying. We will also be testing each "leg" of the flying procedure separately to ensure all directions are functional separately.

Verifying tests: There are many parts to testing one piece of the drone. When it comes to the code, we will do simple software testing for bugs or syntax errors way before it gets on the drone. Once that is completed, we will run the code in a drone simulation to see if it functions as expected and the output looks reasonable. We do these checks so that when we test things out on the drone, we know that the code should be working or at least no what errors have already been ruled out.

When testing a physical piece of the drone, the carriage that holds the battery for example, we will do stress tests and do research on the materials used so that we will on good information expect it to not fault during normal performance. We will also do checkups after testing it in action to see if there is any notable damage.

The platform will be tested by primarily having the battery that is normally connected to the drone connected to the platform independently. We will verify that power can be transferred from the drone's payload battery to the platform long before we are testing with landing the drone on the platform. It will also have structural tests that if the drone lands on the platform, it will not be worse for wear.

Acceptance tests: When checking if a piece of our project is ready for the final version will mostly be a lot of checkups and tests after things are working as expected. We will keep checking in on physical parts of the project to see if they are holding up well and still at peak performance after several uses.

When checking up on software pieces of the project we will do more extensive testing on the code. We will try giving many different possible inputs for any given piece to see if it reacts within parameters. We will mainly be focusing on the edge cases during our tests to see if those cases are handled correctly. We will do extensive testing on our primary use and do our best to make sure we can confirm any data we put in so we know we are looking at the results of what we wanted.

Overall acceptance tests are seeing how parts do under stress as well as how they interact with each other. While one piece may work perfectly alone it also has to work well with several other systems. We plan to make exhaustive tests to make sure each thing works well alone as well as in a group.

3 Project Timeline, Estimated Resources, and Challenges

3.1 Table of Tasks

TASK	MAIN CONTRIBUTORS	SUBTASKS	TIME ALLOTTED
Power Transfer	Kevin, Brendan	Finding how to carry the extra power, how the power will transfer, Testing, creating the dock	Each subtask in this area needs to be perfected since this is the main goal of the project, we would like to spend the rest of fall semester doing research and testing the most efficient power transfer that way it will be ready to implement in the spring
Auto Nav.	Allie, Connor, Garth, Kaitlyn	Getting GPS working, follow coordinates/directions, testing, minimal obstacle	These tasks will be taking up most of both the semesters, since to make the power transfer work we will need to

		avoidance	have the drone be able to auto navigate to the docking pad and then land on it's own. We will most likely spend the fall semester giving it GPS coordinates to avoid and making sure it can navigate to a signal or GPS coordinates.
Plug Design	Kevin, Brendan	Making the CAD Files, Making sure the drone can land on the plug correctly, Make the plug, testing	These subtasks will mostly get done this semester. We would like to have the plug design testable/tested by December. That can then be implemented and printed in the spring.
Other Code	Allie, Connor, Garth, Kaitlyn	Testing, Motors spin at appropriate speeds, drone has emergency code (manual override), Learning how to use the API	These subtasks in this group are all important. However, most of these will go on throughout the entire year. These are things that will need to be improved as we progress on our project.

Table 1

There may be more tasks added to this chart, but the main tasks are listed above with the main goals of the project listed in the chart above

3.2 Feasibility Assessment

This project is feasible because we have the resources available to us given by the university. We have been given the proper funding, materials, and group members to finish the project. Some challenges we believe we will face are challenges with the precision landing, the automatic docking, and making the drone fly autonomous. The precision landing will be a challenge because it will be hard to bypass the landing procedures the drone has. The drone right now lands using meters, and we need it to be more precise (centimeters). We have found some code online to help us through that process however, so it should not be too much of a problem. Another challenge we will face is trying to get the drone to automatically dock. It will be hard to get the image processing program that precise to see a 1-2 cm area and land on it precisely.

3.3 Personnel Effort Requirements

Below is our Agile method that has been working very well for our group thus far:

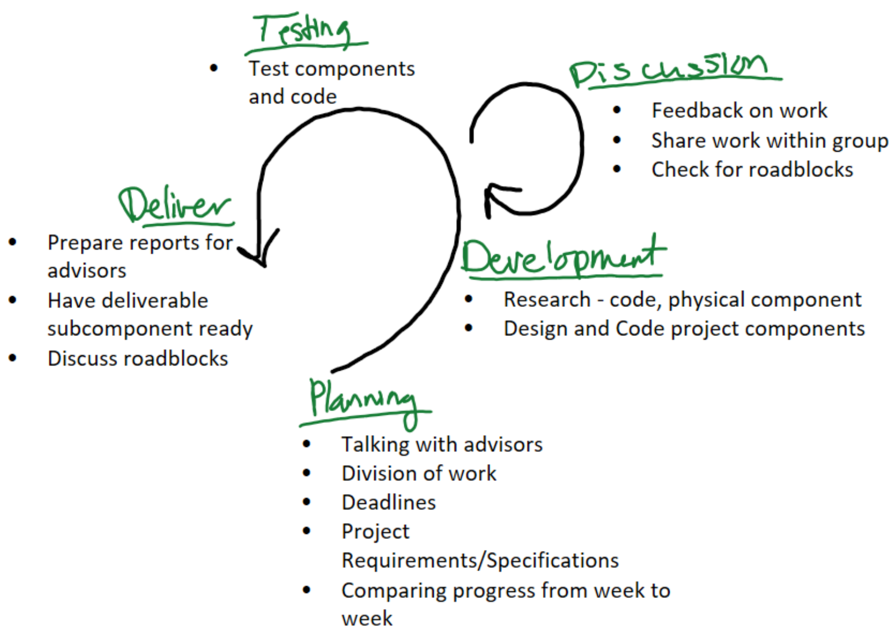

Figure 2

This Agile method requires us to plan our week and tasks. Then start working on the tasks during the beginning of the week. Then midweek, we have a discussion and make sure everyone is doing ok. Then at the end of the week, we test and talk about the results. This allows us to spend a good week on the task and get help from others if needed.

3.4 Resource and Financial Requirements

We will need parts for our drone, including:

(1) Propeller A - $19.99

(1) Propeller B - $19.99

(4) Carbon Fiber Landing Gear (Airframe) - $14.00

(1) Carbon Fiber Airframe Top Board + Middle Board + Silicon Rubber Protection for FPC (airframe) - $36.00

(1) Carbon Fiber Airframe Bottom Board (Airframe) - $14.00

(1) Aluminum Posts 4 M3 X10 Screw X 8 (Airframe) - $5.00

(1) Anker 6700 mAh Battery Pack - $23.99

(1) Intel Micro USB to USB for Drone - $16.00

(1) USB Hub - $9.89

(1) Mini HDMI to HDMI - $5.99

[Ref. 4]

We would also like to purchase a new drone so we have two drones for testing purposes.

Intel Aero Ready Drone - ~$1100

[Ref. 5]

Other materials include coding software (DroneKit), and possibly 3D printed parts for the dock or plug.

If all of these parts are used it will be ~$1342 for our project to be completed.

3.5 Project Timeline

ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Project Scope	1	1	1	1	100%
Fly Drones	1	1	1	1	100%
Select Drone	2	1	2	2	100%
Meet with sd18	2	2	3	2	100%
Research	2	3	3	3	100%
Schematics	4	6	4	6	100%
Skeleton Code	4	6	4	9	95%
CAD Files	8	2	8	2	100%
Board Ready	8	2	8	2	100%
Auto Navigation	8	5	10	5	30%
Plug Protoype	12	3	13	3	10%
Auto Nav. Proto.	12	6	12	6	10%
Dock Prototype	14	6	14	0	0%

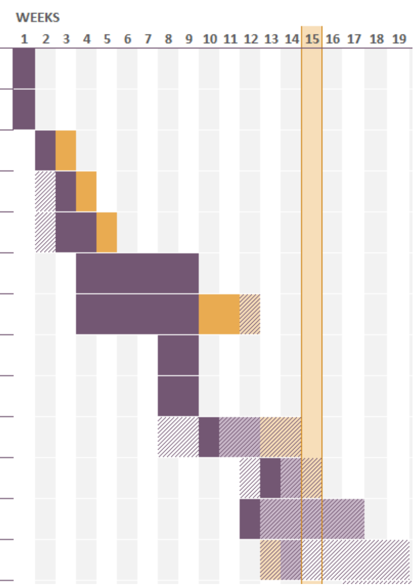

Figure 3

The above figure details the first semester. It begins breaking down the three main goals of autonomous navigation, autonomous docking, and power transfer into smaller goals. Research is necessary throughout the semester, but the initial research was meant to determine what API we would use and how to use that chosen API. Other goals, like completing schematics and skeleton code are not specific to any one main goal, but have to be completed for all goals. Auto navigation is to be completed mostly by the software team, and is broken into researching the process of auto navigation and prototyping the methods we will use. Power transfer is to be completed by the hardware team, and is broken down into creating CAD files, getting the board ready, and creating a plug prototype. Auto docking is also to be completed by the software team and starts with prototyping, but this goal will be completed more in the second semester.

ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Dock Prototype	14	6	14	0	0%
Auto Nav Arrival	17	5	0	0	0%
Auto Nav Return	19	5	0	0	0%
Target Detection	19	2	0	0	0%
Connector Constructed	22	2	0	0	0%
Auto Nav Testing	22	2	0	0	0%
Docking Testing	24	5	0	0	0%
Navigation and Docking Integration	26	2	0	0	0%
Power Transfer Testing	28	5	0	0	0%
Final Integration Testing	29	2	0	0	0%
Final Working Demo	31	2	0	0	0%

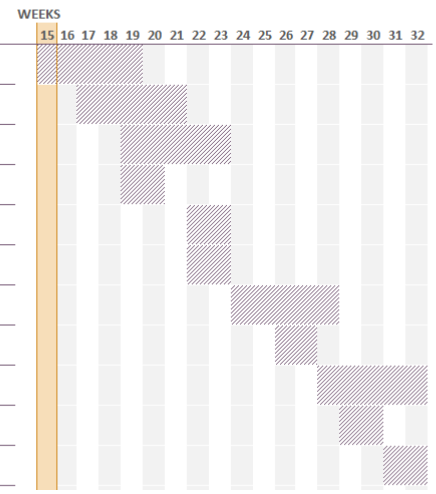

Figure 4

The above figure details the second semester. It builds upon the previous semester in that the auto navigation prototyping that was started in the first semester is completed, as well as having working demonstrations for the auto navigation and power transfer. The software and hardware teams will come together to finish the auto docking, and the larger goal is further broken down into having a working auto dock demo. All final testing will be done at this stage, both separate testing for each individual portion and integrated testing to ensure all components come together. The working demos will come together to create one final working product by the end of the semester.

4 Closure Materials

4.1 Closing Summary

There are many different situations where delivering power to locations is both crucial and difficult. An easy example is after a natural disaster it would be necessary for there to be power at first aid stations or hospitals or just living areas when the power has been knocked out. However, in such situations it can be difficult to transport batteries or generators quickly to places that would need the power. We will make an automated drone system that will autonomously send drones out to locations holding a payload in order to transfer power over to it. There will be multiple drones that can service multiple different nodes and different locations. This will all be able to be done with as little human input as possible. The drones will be able to fly over and hazardous conditions that would impede ground transport and will return to base to refill their payload to continue to service the different locations as quick and as efficiently as possible.

4.2 References

1. "No Drone Zone." FAA Seal, 11 July 2017, www.faa.gov/uas/where_to_fly/no_drone_zone/.
2. "Unmanned Aircraft Systems." FAA Seal, 15 Nov. 2018, www.faa.gov/uas/.
3. "Autonomous Drone Programming in Python." GitHub, github.com/intel-aero/meta-intel-aero/wiki/04-Autonomous-drone-programming-in-Python.
4. "Aero Drone Parts." Yuneec, us.yuneec.com/intelaerodroneparts.
5. "Intel® Aero Ready to Fly Drone." Intel, click.intel.com/intel-aero-ready-to-fly-drone-2404.html.
6. "Prime Air." Amazon, Amazon, www.amazon.com/Amazon-Prime-Air/b?ie=UTF8&node=8037720011.
7. Corrigan, Fintan. "Drone Waypoint GPS Navigation Technology And Uses Explained." DroneZon, DroneZon, 11 Sept. 2018, www.dronezon.com/learn-about-drones-quadcopters/drone-waypoint-gps-navigation-technology-explained/.

4.3 Appendices

Past group's work: http://sdmay18-40.sd.ece.iastate.edu/

DroneKit documentation: http://python.dronekit.io/