UAV Assisted Energy Delivery

DESIGN DOCUMENT

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

1 Introduction 3

1.1 Acknowledgement 3

1.2 Problem and Project Statement 3

1.3 Operational Environment 3

1.4 Intended Users and Uses 3

1.5 Assumptions and Limitations 3

1.6 Expected End Product and Deliverables 4

2 Specifications and Analysis 4

2.1 Proposed Design 4

2.2 Design Analysis 5

3 Testing and Implementation 6

3.1 Interface Specifications 7

3.2 Hardware and software 7

3.3 Functional Testing 8

3.4 Non-functional Testing 8

3.5 Process 10

3.6 Design Test Implementation 10

3.7 Results 11

3.8 Modeling and Simulation 11

4 Closing Material 12

4.1 Conclusion 12

4.2 References 12

4.3 Appendices 12

List of Figures

Figure 1: Top-Level Design Diagram

List of Tables

List of Symbols

List of Definitions

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

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 Introduction

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 and Project 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 is even the issues 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 Operational Environment

The UAV will be expected to survive and operate weather conditions more harsh 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 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 an 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 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. By the end of the first semester, we will have a prototype of the drone which can autonomously fly from one point to another at a distance of approximately 100 yards. We will also have a prototype of the power transfer system, and will have made visible progress on the autonomous docking system. By the end of the second semester, we will have completed the autonomous docking feature, and the drone will be able to shuttle data or energy between multiple tethered and untethered nodes.

First Semester:

- Autonomous flying prototype: December 3, 2018
- Power transfer prototype: December 3, 2018

Second Semester:

- Final UAV: May 1, 2019

2 Specifications and Analysis

2.1 Proposed Design

Our project will consist of programming a drone to autonomously fly to a node, land, and then transfer power. This will be done by having our group divide and conquer most of the time. One group will mainly be focused on coding, and the other group on design and power transfer.

The design will be broken into two big parts as explained above, with the autonomous flying being/programming being one section and then the power transfer being the other section. The flying will also be further broken down into two sections, one being for flying to the node and then landing on the node. This can be shown below in figure 1. The flying to the node code will be needed to be done first, as we need to make a part of the code “hand off” control of the drone once it reaches a certain distance from the landing node. The precision landing code will need to then interface with the same API the flying will use so that the transition will be seamless for when the drone goes back from the docking to the general flying.

The energy transfer group will need to develop a way that the drone can carry a payload battery that will not impede the flying of the drone, and that is light enough for the drone to carry it long distances. They will also need to develop a system that will attach and detach from a target node without the assistance of humans, so that the power transfer can also be autonomous. The group will also be in charge of making a shape on the target node that will be recognized by the precise landing code developed by the software team. They will need to quickly learn an AutoCAD software to design the carrier, as this is something, they have limited experience in. They also need to communicate frequently with the coding group to ensure that any changes the power transfer parts do interfere with the coding. Within the design group one person will work on building a carriage for the battery, another on creating a design for the target node.

The progress we have made in the first semester of this project is mainly research, but we do have a few physical things we can present. Starting with the coding side, we have done research and tried connecting to the drone. We have all created a GitHub and connected to it so we can share our work. Our current status is we can run simulations, but not connect to the drone with an SSH port. We have since looked into resetting the drone to defaults so we could set it up the way we would like it. They have also had accomplishments like being able to test code on a drone simulation program. This semester has mainly been research for general movement paths, precision movements and alignment, dropping the drone through the air smoothly, and hovering. These topics will all be implemented next semester. We will also get the drone reset and usable next semester.

The hardware side was able to accomplish building the frame of the dock this semester. They have also began researching the best colors and patterns for image processing. This is important so the drone can use the camera to see the landing pad and align itself properly. If it cannot recognize the landing pad it will lead to landing failure. Another item that has been accomplished by this team is creating a carriage for the battery to attach. This was done in Autodesk as a file that can be 3D printed or manufactured. Overall our group is making progress toward our goal and will be ready to test more in the spring semester and finishing our project.

2.2 Design Analysis

We started our project in August of 2018 and it will last until May of 2019. To accomplish our project, we will go through a series of tasks.

Our first task we had was to get all the documentation for the project and generate some ideas. Then we went out to test the UAVs or drones so we could pick out which drone we wanted to work with for our project. After doing some research on what can be used to code each drone, we made a selection. We chose the Intel Aero Ready Drone. The Intel Aero Ready Drone was our first selection for the following reasons: the previous group used this drone, thus we would not have to repeat a lot of what they did; it had open source code, so it was an easy selection in terms of ease-of-use; and we already had a written image processing code for this specific drone. Then we needed to pick an API to code the drone with, and we needed to start designing the plug and landing pad. The landing pad was designed to contrast the color of grass (green), so the platform is painted orange to ensure the image processing code will pick up the edges of the platform. The platform also has a key hole design so when the drone is close to landing it will see the key hole and orient itself according. The next step would be to implement the plug onto the pad. As for the API, we have selected to work with DroneKit. We decided to use DroneKit because the previous group who did this project used it, so we can use some of their code and simulation code for the drone. We chose DroneKit because we researched the differences between DroneKit and mavROS, and we found DroneKit was easier to manipulate certain aspects of the drone we would not be able to with mavROS. Some of those aspects that were easier to manipulate in DroneKit included precision landing, precision movement, and the landing procedure as a whole.

As of our stance in the current semester, everything that follows will occur in the future. We will need to program the drone to have a take-off, navigate, and land phase or procedure. These will start being implemented around January 2019. The rest of this year, we would like to learn about how to control and code the motors on the drone. Find and restrictions for take-off and landing, and work with the GPS selection of the drone. We will need to take some time to research all these sections and then learn how to implement them. We also need to make sure everything is autonomous, but can be interrupted for manual control. As we continue throughout both semesters, we will need to do testing, and it will more than likely need to be done outside.

From our current standpoint, we know we will have many challenges and learning curves ahead. But we believe that if we stay on schedule, we will be able to conquer all the milestones we need. However, we know that coding will be a large important part we need to stay on top of. Overall, we expect the project to go well.

3 Testing and Implementation

1) Preliminary Testing
    a) Preparation for Takeoff
        i) Charging & Attachment
        ii) Stable and safely connected power supply
        iii) Ensure all systems are stable prior to flight
    b) Flight Testing
        i) Simple testing
        ii) Takeoff/Landing within a few feet
2) Interfacing between drone and device
    a) QGround Control
        i) Drone Connection
        ii) Familiarize with instruction delivery
        iii) Repeat flight testing
    b) Automation Testing
        i) Automated flight course through pre-programmed directions
        ii) Takeoff, reach destination and return
    c) Finish Preliminary and Communications testing
        i) Safe takeoff/landing sequences
        ii) Able to reach destination

3) Commence Sensory Testing
    a) Image-processing with on-board camera
    b) Gyroscopic Calibration
        i) Drone has self-balancing capabilities, we need to research into how to understand ways to communicate these changes to the drone
    c) Flight Height Limitations
        i) Height Detection
    d) Finish Sensory Testing
        i) Able to safely approach a destination with real-time input
        ii) Real-time data feedback; accurate and reliable
4) Battery Attachment
    a) Detach battery from drone
        i) Reliable connection
        ii) No performance hinderance
        iii) Adjust battery for payload weight requirements
    b) Battery Adjustments
        i) Battery needs to be adjusted in such a way that it can attach to a specified target once that drone arrives
        ii) Repeat testing with battery connectors
    c) Energy Deliverance
        i) Locate energy deliverance destination
        ii) Repeat drone sensory tests
        iii) Found accuracy of drone automated guidance
    d) Finish Battery connection and delivery testing
        i) Able to reach destination accurately with sensory input
        ii) Drone with adjusted battery can safely land and attach to a specified target
        iii) Drone can safely remove itself from charging target and return

3.1 Interface Specifications

Software specifications:

a. DroneKit - Open-source UAV communication tool. Allows us to connect to and provide flight and docking instructions for the drone.

b. MAVLink – Communication tool that allows us to use PX4 to connect to drone.

Hardware specifications:

a. Intel Drone Aero Board – Board drone uses to interpret code sent to it

b. Remote controller for drone – RF controller used to manually control drone

3.2 Hardware and software

Software:

a. Image Processing – A program, written in python, to find our landing station on the ground by looking for a specific color and shape that is our station

b. Python – Most instructional code is written in Python language

c. Ubuntu – Linux based OS, Open source software package allows us to flash our drone and construct the required software to achieve a particular task.

Hardware:

a. Battery Carriage – Holds the exterior battery onto the drone, and feeds the cord from the battery to the landing station.

b. Landing station – Square space elevated off of the ground for the drone to land on, has connector for the drone battery pack to plug in to.

c. Intel Compute Board - Flexible piece of hardware which handles DroneKit information and can receive additional inputs should we deem it necessary.

d. LiPo Battery - Relatively compact and powerful, can be extremely dangerous if not handled properly and should be used responsibly.

3.3 Functional Testing

Our drone will need to be able to take off from the home node and fly autonomously to the target node. If the drone gets within one meter of the target node, we will count that as success. This is the first step of the process and will be tested heavily before it is integrated.

We will also have the drone land and dock autonomously on the target node. In order to do this, we must have precision control of the drone by at max one centimeter. We will use image processing and edge detection in order to find the target node. This target detection will have to read at least 90% off the target. The flying of the drone to the given location and the docking of the drone will both be tested independently several times. Once we are confident with both of them. (By them both passing their tests consistently) we will move on to integrating the two of them together. We will have the drone fly to the given GPS coordinate and then land on the target node with target detection and precision movement.

We will also need the payload battery on the drone to transfer power to the battery on the target node. In order to confirm this is functioning we will check to see if we can get a voltage reading from the target battery during and after the drone is connected to the target. This will be tested alone by just physically setting the drone down on the target node and connecting it by hand. If the connecting is solid and we have power getting transferred consistently we will integrate it first with the landing code by having the drone go up and then run its landing procedure as it finds the target. If we can get it to dock reliably while also making a connection that transfers energy to the target, we will integrate with the autonomous flying section. We will do integration testing in groups of two. We will not test them all together until the flying and docking work together and the docking and energy transfer work together.

3.4 Non-functional Testing

We have several different nonfunctional tests. The first of which is scaling our system to go miles to the target node. In the scope of our project currently we will just be testing up to 100 yards. The drone will have the potential to go farther but we will not be looking into testing beyond the length of our testing area.

We will also not currently be looking to expand our number of nodes or drones in the system. While this is something that we might look into if we finish things before we expect to. We currently only have one target node and one base. There is definitely potential to add more target nodes and to a lesser extent more home nodes. However, adding more drones to the system while possible and could eventually be added, would greatly complicate the project and is currently out of the scope of what we are doing.

There is also the possibility of having a larger payload battery for the drone to carry for more energy demanding systems. This will also necessitate a larger drone to be able to carry the larger payload. This is a cost area more than the others. Getting a new, larger drone with a large battery will be very expensive. There is also the question of if the code will be compatible with the other drone. If it is not compatible with the current drone system (Intel ready to fly) then systems would have to be rewritten in order to be compatible.

Another is the speed of the process. While we will be looking to have it be quick it will not be a focus of ours. We are going for more of a proof of concept. While we have no reason to be slow, we will not be altering our systems around making them as fast as possible. We are looking to make it happen not make it happen as fast as possible.

The power transfer is another thing we will not be looking into making as efficient as possible. Having power transfer efficiently is nontrivial and could be an entire senior design project on its own. We will look to be decently efficient and then keep it there for the sake of time.

We will also not be looking into security for the drone. We will have the default that the drone has and leave it at that. This project does not have any security aspect of it and we will be spending our time on getting the drone to do as expected instead. This would be an interesting project for someone else to look into at some point. I am sure people are looking into security of drones currently. With so many more drones being out and about these days there are likely more people than ever before that are interested in hacking into the drones.

3.5 Process

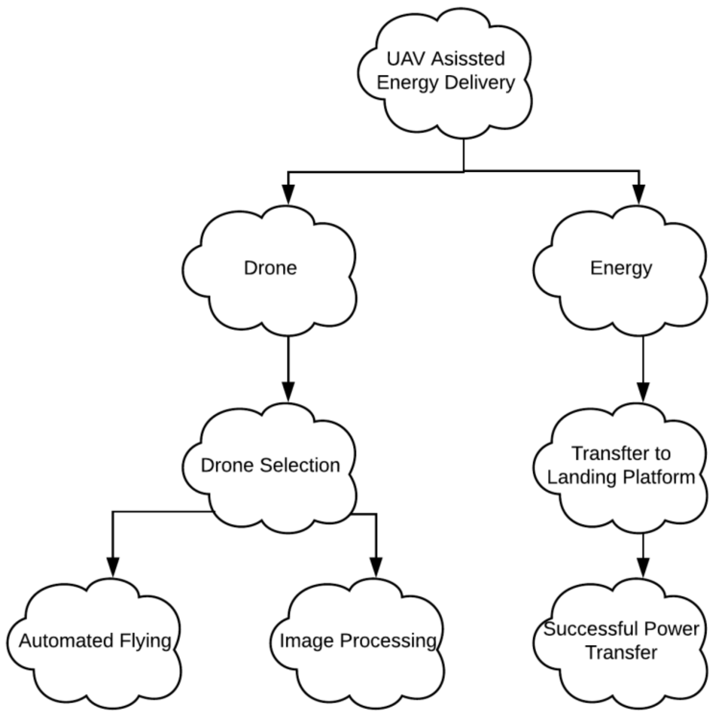

Figure 1

Above is the diagram of how we are breaking the work for the project up into different processes. This diagram represents our initial plan, but is subject to change as issue arise. For example, landing the drone is more complex than just telling the drone to turn the throttle off. We will need to use various sensors to ensure the drone lands at the proper location. During our time working with the drone we have also run into issues that needed to be handled. One of these issues was that we need to reinstall the operating system, Ubuntu, on the drone in order to connect to the drone through WiFi.

3.6 Design Test Implementation

After we create a solution, we need to be able to test our creation to make sure it fits our requirements and specifications. Our first main part to our project is auto navigation. Auto navigation will first be tested in an open field. The first check point is telling the drone to go a certain distance in a straight line. If it can do that then we will move on to testing if it can travel in a square that we set corners as points. If it can do that, it will be time to use GPS coordinates and image processing to make it find the landing pad. The flight times that we can measure for criteria is the length of time it takes to arrive. Since the drone only has a twenty minutes of flight time, we would like it to get to the specified node as a one-way trip in one to ten minutes. Then we can say the return trip at max is twenty minutes, then the drone will be near out of power.

If it can find the landing pad the next test will be if it can land properly aligned so the power transfer or receiving can take place. For landing will measure the time it takes to land. It needs to be less than one minute as not to consume too much battery. When aligning we also need it to be centimeter precision. If it cannot utilize centimeter precision, it may not align properly.

The second part of our project is power transfer/delivery. Tests that we will start with is if the battery can be attached to the drone via the carriage we are designing. We then need to make sure the drone can fly with the extra weight attached. We will then need to test the cord on the cord on the drone and the landing pad and make sure they can connect and disconnect without human interaction. We will also time the power transfer time. We want the power transfer to be under two minutes.

By considering a total time throughout the whole process we can make sure the drone will have enough power for the entire trip. It will also help us work on efficiency of the code and hardware involved. All these individual tests will help put our large project come together and perform per the specifications.

3.7 Results

Testing is minimal so far, but present results include a possible calibration issue with the placement of the drone battery that causes the drone to lilt to one side at lift off. This can cause flight failure. We need to conduct more testing with manual flying, as well as search for forum answers about a possible Intel-provided solution to the problem.

Implementation has been tough at some points this semester for a variety of reasons. One of them was when we tried to connect to the drone but it was a failure, as the previous group had altered ports and set passwords that we did not have. We contacted them, and were able to get some information but not all of it. This is still ongoing, but we hope that a reset of the operating system of the drone will have us push our code to the drone for testing. We also had a problem with the drone not being able to fly for about a month, as an accident when flying the drone caused most of the parts to break off and become unflyable. Replacement parts were quickly ordered, and once they arrived, they were installed and tested as soon as possible. The drone is now back to flying order and ready for testing.

3.8 Modeling and Simulation

The image processing code for the drone will be supplied by the previous group's iteration. They did a good job of ensuring the code was up to standard with the edge detection of the docking node, and it was able to find the shape and color of the docking node. Being able to find the docking node itself is good because it shows the code has edge detection and it can find the color of the docking node, but last it can find the shape we put our plug into. The plug is designed so it is a key hole for the drone to land on, if the image processing can detect this we are on the correct track. The platform will be colored to contrast green (the color of grass) so the image processing code can easily detect the platform. It will also have a key hole for the image processing to detect when it needs to orient itself so the magnetic cords attach correctly and there are no problems with the power transfer.

We need a rechargeable battery for the power deliver system. It will start with the drone taking a signal from an untethered node. The drone will then move to the node, process what it sees on the ground, ensure safe landing, and land. As the drone is landing the cord attached to the rechargeable battery will find the magnetic connection on the docking node. This will start the power delivery to the node from the drone. The drone will wait five minutes to deliver power to the node, and once finished it will leave the node and return to the home base platform.

4 Closing Material

4.1 Conclusion

Having the ability to transfer energy from one location to another with minimal to no manpower and with minimal set up is something that will change how energy issues are solved. Having the ability to send out a fleet of drones to deliver energy to a location at a moment's notice has not been done before. We have a drone that is able to autonomously fly to a remote node. Then transfer energy to that node followed by the drone returning to its base. All of this being done without any human input. This is composed of three pieces. The first being the drone is able to autonomously fly from one location to another. Next the drone is able to land on a remote node that asks for a charge. Once it has landed on the node it delivers energy to the node until it has expended its charge. Finally, it returns to its starting location so it will be able to recharge and start the process over again.

4.2 References

1. "DroneKit by 3D Robotics," *DroneKit*. [Online]. Available: http://dronekit.io/. [Accessed: 12-Oct-2018].

2. "Intel® Aero Ready to Fly Drone," *Intel*. [Online]. Available: https://www.intel.com/content/www/us/en/products/drones/aero-ready-to-fly.html. [Accessed: 12-Oct-2018].

4.3 Appendices