2015

Conceptual Design Review

UAV PACKAGE DELIVERY SYSTEM

TEAM AVENGERS

- Tushar Agrawal (Software Lead and Project Manager)
- Sean Bryan (Mechanical and Communications Lead)
- Pratik Chatrath (Sensor Lead and Software Developer)
- Adam Yabroudi (Systems Engineer and Electrical Lead)

MASTER OF ROBTIC SYSTEMS DEVELOPMENT, FALL 2015 | FRIDAY, OCTOBER 2, 2015

Table of Contents

1. Project Overview

1.1 Objectives

1.1.1 Background

Currently, package delivery truck drivers hand-carry packages door to door. This model is used by Federal Express (FedEx), United Postal Service (UPS), United States Postal Service (USPS), and Deutsche Post DHL Group (DHL). We believe that drones have the potential to expedite this system.

Amazon is developing Prime Air with the same intent. However, we believe the most efficient system combines delivery trucks with Unmanned Aerial Vehicles (UAV's) which saves time, expense, and improves customer's satisfaction.

1.1.2 Problem Summary

Delivering packages to a house using UAVs.

1.1.3 Project Description

Given the coordinates of the house, a UAV with a package takes off from point A, autonomously reaches close to the house, scans the outside of the house for a visually marked drop point, lands, drops off the package, then takes off again to land on another platform at point B.

1.2 Use Case

Sam drives a package delivery truck for one of the largest parcel delivery companies. He arrives each morning to a preloaded truck and is handed his route for the day. Even though he has an assigned route, he sometimes is tasked with delivery packages to additional streets. These are often the packages that should have been delivered the day before. Thus it's critical that packages make it to the right house on time today.

Now that his company uses drones, Sam can cover more area in less time. He drives out to his first neighborhood for the day with two packages to deliver. He can quickly deliver the first package, which is heavier. The second package is lighter but a street over. After parking, he quickly attaches the second package to a drone and selects the address on the base station computer. The drone takes off and disappears over a rooftop as Sam unloads the first package.

Having delivered the first package, Sam gets back in the truck and starts driving. In the past, he would have driven to the next house and dropped off the package. Nowadays, Sam knows that the drone will deliver the package to the right house and catch up. This saves him a few minutes which adds up over the course of the day to real time savings. This makes Sam a little happy.

Meanwhile, the drone has moved within vicinity of the second house. It begins scanning around for the visual marker outside the house. The drone finds the marker and moves in for a landing. It's able to avoid people on the sidewalk and the large tree outside the house. The drone lands on the marker and does a quick confirmation, it checks the RFID code embedded in the marker. Confirming the correct house has been found, the drone releases the package and notifies the package delivery truck's base station. The base station then updates the drone on the delivery truck's position.

The drone catches up to Sam at a red light and they continue on their way. Sam's day continues this way.

On a major street, he has several packages to deliver in the area. He quickly loads up a few drones, selects the addresses, and watches to drones do all the work. Sam had to get a gym membership since he's no longer walking as much, but he's happy to be getting through neighborhoods substantially faster. Because the drones allow one driver to do more, the delivery company is able to offer package delivery at a more competitive rate with more margin. This makes customers happy in addition to getting their packages faster. In turn, they are more likely to use the delivery company, which makes the company pleased with their investment.

Late in the day, the base station on Sam's delivery truck notifies him that an adjacent route wasn't able to deliver a package. In the past, this would have meant that the package would be driven back to the warehouse to be resorted and delivered with tomorrow's load. This was a substantial waste of fuel and manpower. Today, routes can be dynamically updated. A drone will deliver the package to Sam's truck. Once he's in the area, the drone will deliver the package. The customer will never know there was a problem, and the delivery company saves money.

Sam arrives back at the warehouse, his truck empty. He's satisfied in the work he's accomplished, customers are happy that received their packages on time, and the delivery company is exceptionally happy with the improved efficiency and customer retention.

3. System-Level Requirements

Requirements

The critical requirements for this project are listed below under Mandatory Requirements. These are the 'needs' of the project. Additionally, the team identified several value-added requirements during brainstorming. These 'wants' are listed below under Desired Requirements.

3.1 Objective Tree

UAV package deliver has numerous requirements. A high-level breackdown is given visually through the Objective Tree shown below (Figure 1).

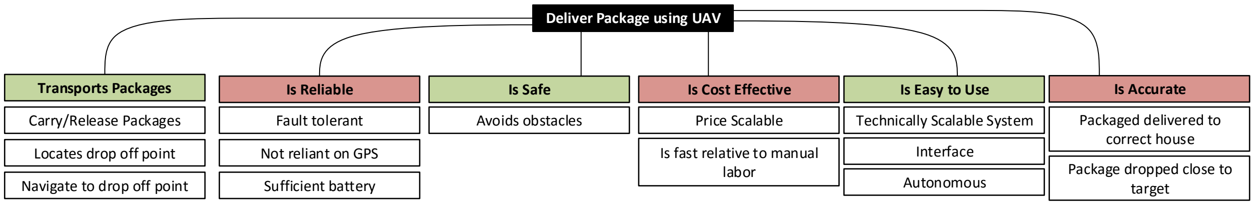

Figure 1: Objective Tree

3.2. Mandatory

3.2.1 Functional Requirements

M.F.1 Hold and carry packages.
M.F.2 Autonomously take off from a visually marked platform.
M.F.3 Navigate to a known position close to the house.
M.F.4 Detect and navigate to the drop point at the house.
M.F.5 Land at visually marked drop point.
M.F.6 Drop package.
M.F.7 Take off, fly back to and land at another visually marked platform.
M.F.8 Takes coordinates as input from the user.
M.F.9 Communicates with platform to receive GPS updates (intermittently).

3.2.2 Non-Functional Requirements

M.N.1 Operates in an outdoor environment.
M.N.2 Operates in a semi-known map. The GPS position of the house is known, but the exact location of the visual marker is unknown and is detected on the fly.
M.N.3 Avoids static obstacles.
M.N.4 Not reliant on GPS. Uses GPS to navigate close to the house. Does not rely on GPS to detect the visual marker at the drop point.
M.N.5 Sub-systems should be well documented and scalable.
M.N.6 UAV should be small enough to operate in residential environments.
M.N.7 Package should be light and small.

3.3. Desired

3.3.1 Functional Requirements

D.F.1 Pick up packages.
D.F.2 Simulation with multiple UAVs and ground vehicles.
D.F.3 Ground vehicle drives autonomously.
D.F.4 UAV and ground vehicle communicate continuously.
D.F.5 UAV confirms the identity of the house before dropping the package (RFID Tags).

3.3.2 Non-Functional Requirements

D.N.1 Operates in rains and snow.

D.N.2 Avoids dynamic obstacles
D.N.3 Operates without a GPS system.
D.N.4 Has multiple UAVs to demonstrate efficiency and scalability.
D.N.5 Compatible with higher weights of packages and greater variations in sizes.

3.4 Performance Requirements

P.1 UAV places the package within 2m of the target drop point.
P.2 UAV flies for at least 10 mins without replacing batteries.
P.3 UAV carries packages weighing at least 400g.
P.4 UAV carries packages that fit in a cube of 30cm x 30cm x 20cm.
P.5 One visual markers exists per house.
P.6 Visual markers between houses are at least 10m apart.
P.7. A landing column with 3m radius exists around the visual marker
P.8 Obstacles with a minimum cross section of 1.5m x 0.5m are detected and actively avoided.
P.9 An edifice with a minimum cross section of 8m x 5m is required to navigate through.

4. Functional Architecture

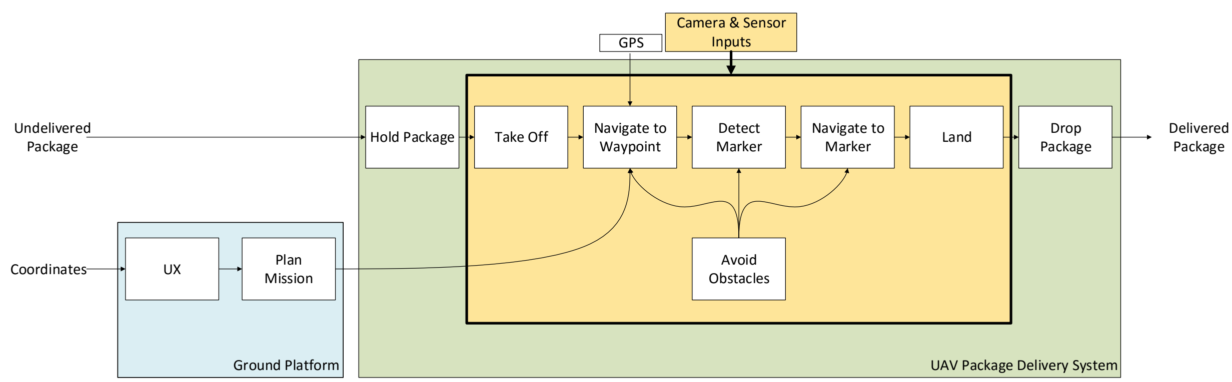

Figure 2: Functional Architecture Diagram

The functional block diagram of our system is as shown in Fig. 1. Viewing the whole system as a black-box there are 2 inputs – package to be delivered and GPS coordinates of customer location. The output of the system is the package successfully delivered at destination.

Looking in the black box now, the UAV initially loads the package. Coordinates of customer location are input to the User Interface. The developed Plan Mission software decides the navigation waypoints and plans path to destination. UAV then takes off and navigates using the waypoint to the vicinity of the destination using GPS input. It uses camera and sensor system to scan the vicinity of the customer destination and detect marker put up by the customer.

Once the marker is detected, UAV navigates to it and lands. During all the 3 functions – 'Navigate to Waypoint',' Detect Marker' &' Navigate to Marker' UAV continuously runs an obstacle avoidance algorithms on-board. UAV avoids the obstacles and keeps iterating its path. After landing UAV drops the package and flies back to the predetermined location.

5. Trade Studies

Our project contains three major trade studies. We had to choose the UAV platform, our visual processing board, and our sensor suite for obstacle detection.

5.1 UAV Trade Study

There are many factors that go into selecting a proper UAV platform such as cost, shipping time, payload capacity, and flight time. In the end, we used 10 different metrics to evaluate the best choice for our application. The top three highest weights were payload capacity, price, and flight time. Payload was given the highest priority since it was crucial to our project application. A UAV without the ability to carry a package was useless in our project. Price was second highest because it was a large constraint on our project. Flight time was given the third highest weight because of the necessity that the vehicle consistently reach the door of the house and return to the platform no matter how far the house was from the street.

The remaining factors were derived from our performance requirements and scaled appropriately based on their effect on the project timeline and ease of integration into the complete system and vision of our project. The final results can be seen below:

Metric	**weights**	**BEV FireFly6**	**3DR X8+**	**Erle Hexcopter**	**3DR Solo**	**3DR X8-M**
Payload	22	6.0	4.5	10.0	0.5	4.5
Price	20	5.0	6.5	7.5	10.0	0.5
Flight time	17	10.0	3.3	3.3	3.3	3.3
Size of vehicle	10	2.0	10.0	10.0	8.0	10.0
Shipping time	10	9.0	9.0	1.0	9.0	9.0
Flight controller capabilities	8	7.0	7.0	6.0	10.0	7.0
Firmware documentation	8	10.0	10.0	2.0	10.0	10.0
Ease of adding mechanical subsystems to frame	5	6.0	10.0	3.0	1.0	10.0
Total	**100**	**6.8**	**6.6**	**6.2**	**6.0**	**5.4**

Table 1: Comparison of UAV platforms

5.2 Vision Board Processing

As will be depicted in the cyberphysical architecture, our system has a separate board specifically for visual processing so that we can do processing in real time during flight. The second board also allows the system to meet safety requirements by allowing all safety critical functions to be run on the flight controller which won't get bogged down by computer vision algorithms.

Our main criteria was processing speed and documentation. Documentation was critical because the board will have to integrate with the rest of the system and we will be designing this ourselves. As a result, being able to debug errors will ensure that the system functions as a whole. Ports/Interfaces were also critical because they affected how the board interacted with the cameras and the rest of the UAV. Our last criteria was price to ensure we met our project budget.

Rasperry Pi 2 and BeagleBoard-xM are currently tied for first place. The final decision will require hands-on evaluation and expert judgement.

Metric	**weights**	**Rasperry Pi 2**	**BeagleBoard-xM**	**Odroid-U4**	**BeagleBone Black**
Processing power	30	4	6	10	6
Documentation online	30	10	9	5	9
Ports/interfaces	25	7	10	8	6
Price	15	10	3	6	8
Total	**100**	**7.45**	**7.45**	**7.4**	**7.2**

Table 2: Comparison of Vision Boards

5.3 Sensor Suite

The sensor system will be critical for the obstacle avoidance functionality of our vehicle. Currently we are still conducting our trade studies because there are many various types of sensors that operate under different conditions. Ultrasonic sensors are affected by the UAV motors but have a wider field of view. Infrared can be affected by the presence of sunlight although some IR sensors are being built to be robust to sunlight. LIDAR provides more feedback than proximity sensors but requires it to be mounted to a spinning mechanism.

Our final solution might involve an integration of both LIDAR and IR sensors but that is still to be determined. One factor that is still in the air is the payload of the gripper that will be needed to build the package pickup system. If that is under allocated weight, we can reallocate that weight for our sensor system and buy more robust sensors which provide more accurate feedback.

6. Cyberphysical Architecture

The cyberphysical architecture can best be understood by Figure 3. On a high level, the system can be broken down into three major categories: mechanical components, electrical components, and software. The electrical components are the bridge between the software and the mechanical actuation.

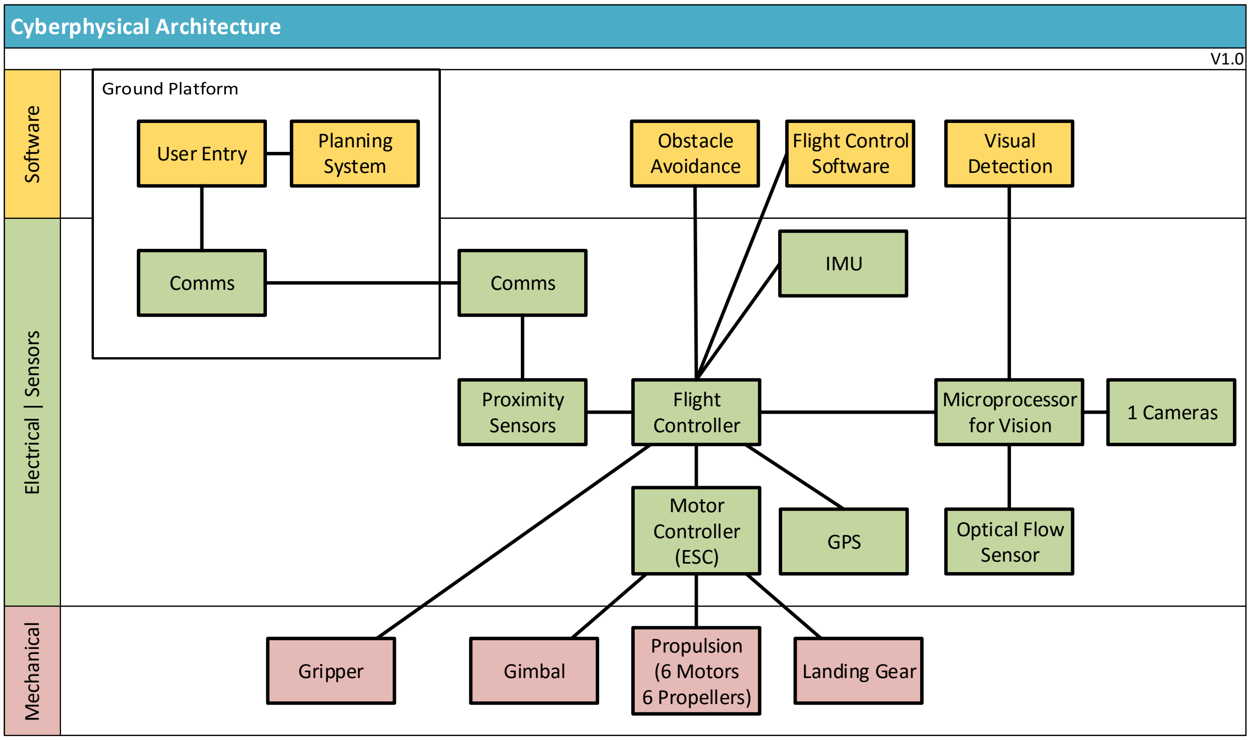

Figure 3: Cyberphysical Architecture Diagram

6.1 Mechanical System

The mechanical system of our project consists of the landing gear, the propulsion system, and the gripper. The landing gear and propulsion system are part of the drone kit that we purchase but must be controlled appropriately by our software. The gripper is a subsystem that we will design specifically for our project and application. This gripper will be the interface between the vehicle and the package and must allow the package to be dropped off upon arriving at the destination. It will be controlled by our flight control system which is the brain of the UAV.

6.2 Electrical System

The electrical system is composed on a high level by flight control boards, the vision subsystem hardware, sensors, and communication hardware. The flight controller is the brain of the entire system and runs all critical flight control software. The flight controller interacts with all the sensors on the vehicle except for the cameras. These sensors include the IMU, GPS, and proximity sensors for obstacle detection. The flight takes inputs through communication to the base platform, from the obstacle detection algorithms, and also from the vision processing board. The output from the flight controller goes to the motor controller and is then converted into appropriate signals to control the propulsion system.

A microprocessor for running visual algorithms connects to the camera and optical flow sensors on board the UAV. The microprocessor computes all the necessary vision algorithms and outputs the result to the flight controller.

6.3 Software

The software of our system can be broken into two categories, software that is computed on the platform and software computed on board the UAV. The software on the platform performs two functions. The first is that it is a user interface for human input and control. The second function is path planning algorithms used to optimize the takeoff and return locations for the UAV. This information is then conveyed to the UAV by their communication channel.

The code that is occurring onboard the UAV can be broken down into three major functions. The first is flight control. This involves converting sensor and user inputs into outputs to the motor as well as safety critical functionality. The second major software function is obstacle avoidance. The UAV must avoid obstacles in flight and must sense its changing environment while in flight. The last set of algorithms revolves around computer vision and visual processing. This code converts visual inputs into meaningful outputs for the flight control.

Although, the code is broken down by functionality, it is worth noting that the code is actually very complex and has an architecture of its own. This is best modeled by Dronecode's software architecture shown in Figure 4.

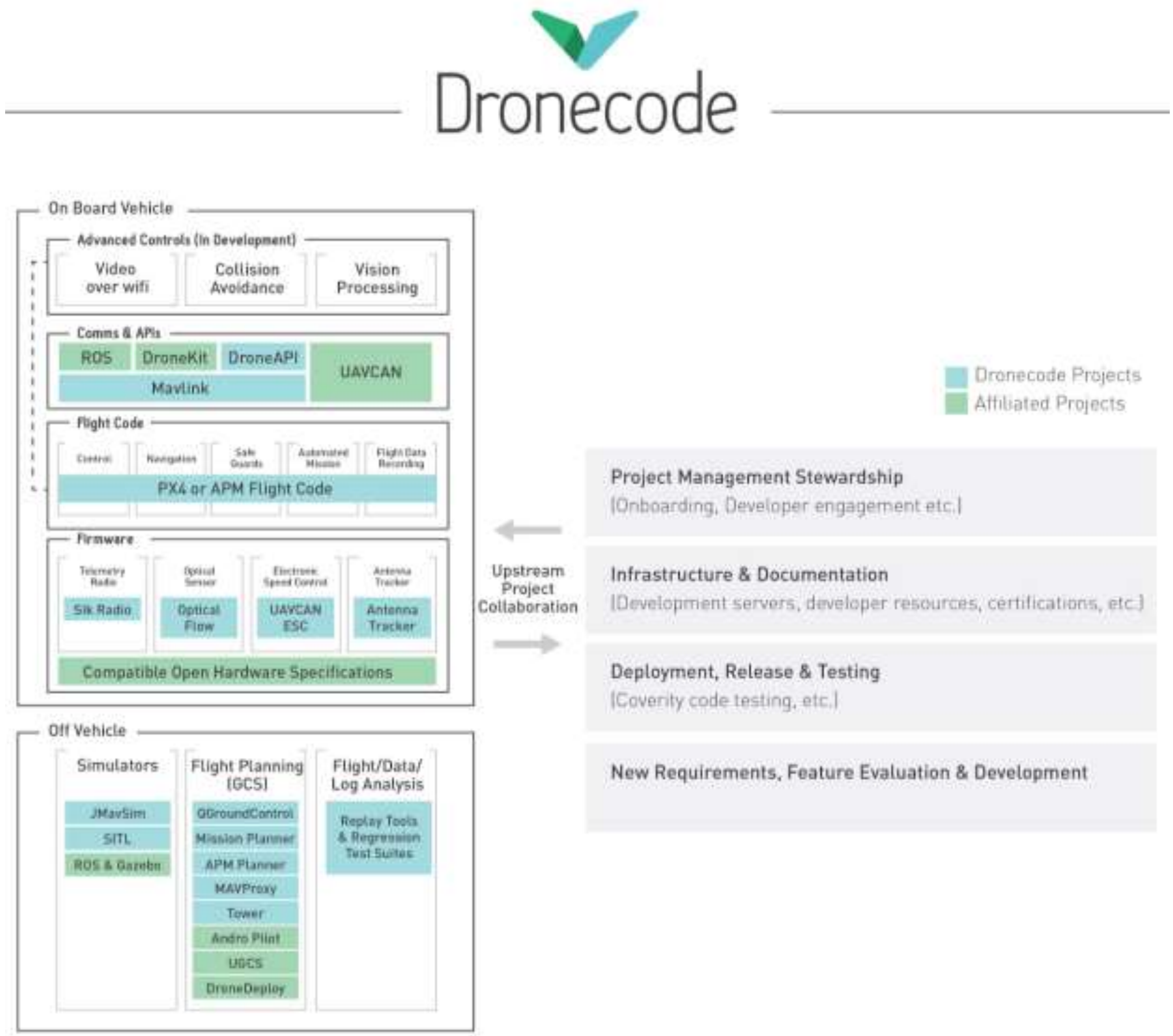

Figure 4: Dronecode Software Architecture

Reference: https://www.dronecode.org/dronecode-software-platform

As is shown by the diagram, the software architecture involves many subcomponents. This is abstracted away in the cyberphysical architecture because it is not code we develop and largely consists of firmware to allow the code to run on various software architectures. The software also has software to handle communication between the flight controller and the sensors as well as software to handle communication to platforms outside the vehicle.

We will be building on top of this software architecture for many of our algorithms displayed in the cyberphysical architecture. We will have to dive deep into the code to perform customize functionality and tight control feedback loops but still that is irrelevant in the cyberphysical architecture that we displayed above.

7. Subsystems

There are five major subsystems in our project. These five subsystems are the electro-mechanical subsystem, the obstacle avoidance subsystem, the vision subsystem, the platform subsystem, and the flight control subsystem.

7.1 Electro-Mechanical Subsystem

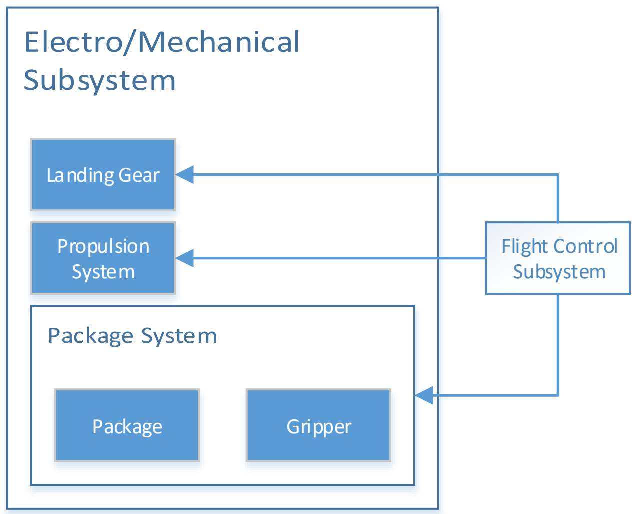

Figure 5: Diagram of the Electro-Mechanical Subsystem

This is the part of the vehicle that involves moving parts. Specifically this system is composed of the propulsion system, landing gear, and package system. The landing gear is specific to our vehicle choice and is composed of actuated legs controlled by a servo. The propulsion system consists of the motor controller, the electronic speed controllers (ESCs), the motors, and the propellers. The propulsion system is responsible for converting electrical signals into thrust and flight.

The last major component of the electro-mechanical subsystem is the package system. This package system consists of the package itself and the gripper which handles the package. Currently our gripper is composed of an electro permanent magnet board that can lift up to 5kg and weighs 35g. (http://nicadrone.com/index.php?id_product=59&controller=product) Should this fail, we have designs for a mechanical gripper that we would design ourselves but the electro permanent magnet far exceeds all other design options and we will work hard to ensure it integrates with our system.

7.2 Obstacle Avoidance Subsystem

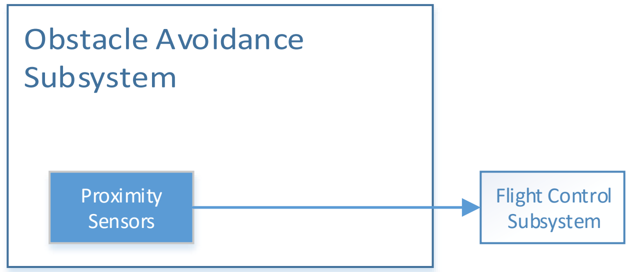

Figure 6: Diagram of the Obstacle Avoidance Subsystem

This subsystem is comprised of the sensors necessary to detect objects in the UAV's path. As described in the system trades section, selecting sensors is still under way. We have analyzed many different options such as LIDAR, IR, ultrasonic, and more, but some of the tradeoffs between weight and price are still undecided. We are aware of our constraints, but the relevant weights of the constraints is dependent on the success of our electro permanent magnet system as well as the weight and location of our cameras. This will be decided within a week of receiving and testing the gripper.

7.3 Vision Subsystem

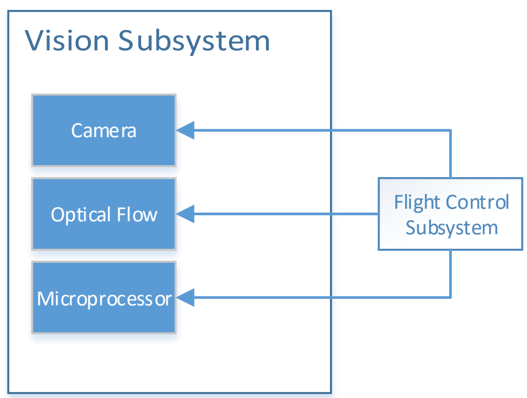

Figure 7: Diagram of the Vision Subsystem

The vision subsystem handles all the visual processing. It is the eyes of the UAV as well as a source of visual odometry. The vision subsystem contains a camera, an optical flow sensor, and a microprocessor. The microprocessor is the board that handles all the computer vision algorithms in real time. Based on trade studies, this board will be a BeagleBoard-xM or a Raspberry Pi 2. We plan to start with the BeagleBoard-xM because we have experience using them in previous projects and because it has higher processing power. Due to the fact that a Raspberry Pi 2 is cheap and well documented, though, we will also buy one of those as well as a backup.

7.4 Platform Subsystem

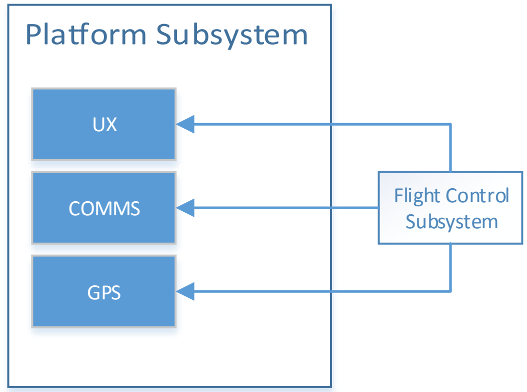

Figure 8: Diagram of the Platform Subsystem

The platform subsystem is a simple mechanism that models the UGV. It is an intelligent platform that contains a GPS, user interface, and communication channel to the UAV. Its sole purposes are to allow human input into the system and to provide a takeoff and landing spot for the UAV. This platform will be useful for many stages of testing from the out-of-the-box tests all the way to the final system validation tests.

7.5 Flight Control Subsystem

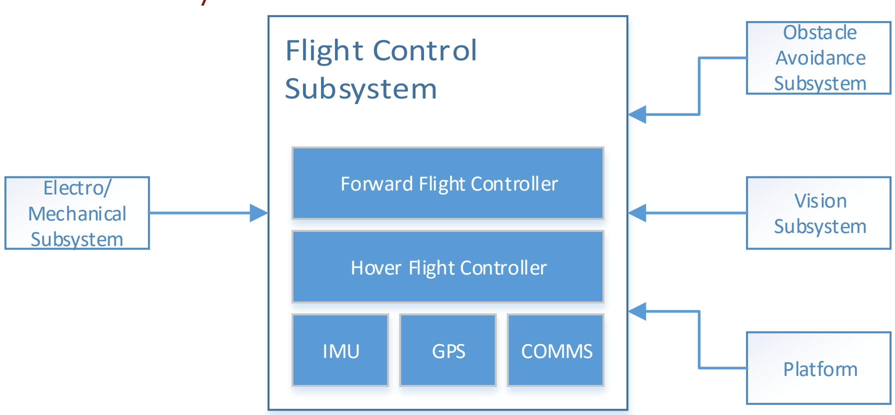

Figure 9: Diagram of the Flight Control Subsystem

The flight control subsystem is arguably the most critical subsystem of the entire vehicle. It is the only system that interacts with all other subsystems and it involves the most interconnected components out of any of the subsystems.

Due to the fact that our vehicle is a fixed-wing VTOL vehicle (FireFly6), it requires two flight control boards to fly. The two boards handle different modes of flight, namely forward flight and hover control. An APM flight controller will be used for forward flight and a Pixhawk will be used for hover control. Both boards will be running Arducopter software and connect to the Bridge, which is an intermediate motor controller on the FireFly6. For simplicity of further explanations, both flight control boards will be referred to simply as the flight controller.

The flight controller must interact with all sensors on the UAV. Such sensors include the GPS, IMU, and the proximity sensors indirectly by way of the obstacle avoidance subsystem. The flight controller also has a radio plugged into it to communicate with the platform subsystem. The flight controller takes all this incoming data from all the various subsystems and transforms it into control outputs for the propulsion system.

Although the flight control system is the most critical subsystem of the entire vehicle, it is also the most well tested part of the vehicle. Part of our analysis in the systems trade included looking at the flight controller capabilities and firmware documentation. Arducopter software has been super well documented and used by many people. Pixhawk and APM flight controllers have also been well tested over the years. As a result, although this system is so critical and thus produces a central point of failure for the vehicle, it also is one of the most robust because of conscious effort to make sure it was well tested by others.

8. Project Management

The following section outlines the high-level Work Breakdown Structure and schedule. For this project, we made a concerted effort to integrate existing technologies wherever appropriate. To ensure project success, great attention has been given to integration testing leading toward a full scenario test. The work for this project has been broken down at the highest level into Systems Engineering, Fabrication and Procurement, Systems Integration, and Testing.

8.1 Work Breakdown Structure

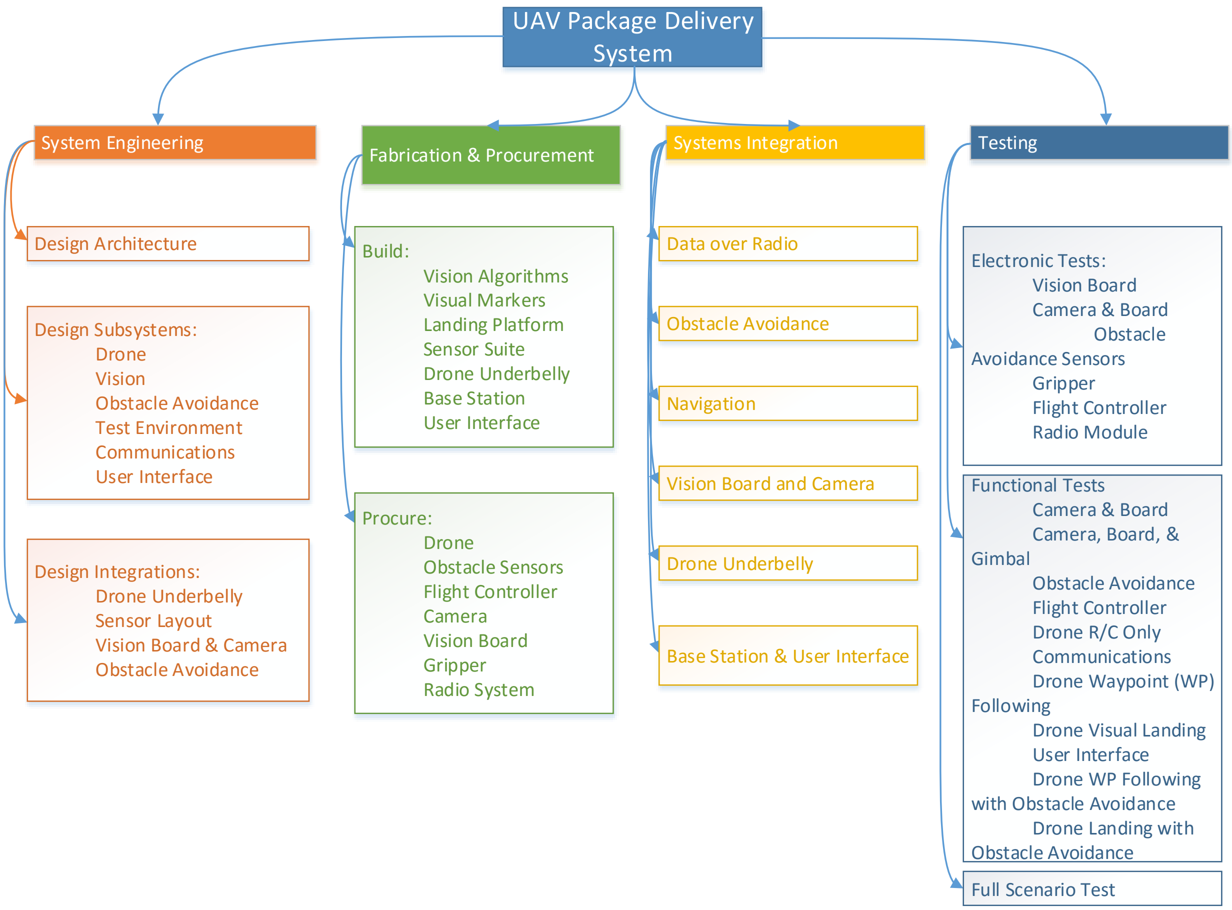

Figure 10: Work Breakdown Structure

8.2 Schedule

The following schedule was made using a Gantt chart and our best estimates of both development time and system dependencies.

Milestones	**Deliverables**	**Due Date**
Conceptual Design Review	CoDR Document: System Architecture, & Project Plan	Fri 10/2/15
Progress Review 1	Test Flight Controller	Thu 10/22/15
Progress Review 2	Assembled Drone Test Gripper Electronics Test Drone R/C-only Control	Thu 10/29/15
Preliminary Design Review	Test Camera and Board	Tue 11/3/15
Progress Review 3	Test Camera, Board, and Gimbal	Thu 11/12/15
Progress Review 4	Data Test of Radio Module Integrate Radio into Drone	Tue 11/24/15
Progress Review 5	Table Test Obstacle Avoidance Sensors and Algorithms Test User Interface	Sat 12/5/15
Critical Design Review	Integrate Base Station with User Interface and Comms Integrate Obstacle Avoidance Sensors with Drone	Tue 12/15/15
Final Design Review	Integrate Vision System Into Drone Integrate Gripper Into Drone Test Visual Landing of Drone Test Visual Landing with Obstacle Avoidance Full Scenario Test	Fri 5/13/2016

Table 3: Project Schedule of Deliverables

9. System Validation Experiments

9.1 Fall Validation

Test ID	A	
Test Name	Real time vision test	
Test Description	Validates a working embedded vision system which can detect and track visual markers in real time	
Equipment Required	Vision microprocessor and Camera system set up on a table, Visual Markers	
Step	**Step Description**	**Success condition**
A.1	Attach visual marker to a wall in view of the camera	
A.2	Initiate program	Marker should be detected and demarcated using computer vision
A.3	Move the visual marker either direction in X and Y while in camera view	Marker position should dynamically update

Table 4: Fall Validation, Test A

Test ID	B	
Test Name	Package carrying mechanism test	
Test Description	Validates the package carrying and dropping capabilities of the UAV	
Equipment Required	UAV fitted with the package carrying mechanism, Small package.	
Step	**Step Description**	**Success condition**
B.1	Place the UAV with the package attached to the carrying mechanism	
B.2	Initiate take off manually	
B.3	UAV lifts off and hovers 5m over the ground for 1 minute	The package remains securely attached to the UAV
B.4	UAV descends and lands	Package still attached
B.5	UAV drops package onto the ground	Package is released and lands on the ground

Table 5: Fall Validation, Test B

Test ID	C	
Test Name	UAV navigation test	
Test Description	Validates the flight control and navigation of the UAV (GPS allowed)	
Equipment Required	UAV, Laptop for waypoint control.	
Step	**Step Description**	**Success condition**
C.1	Place UAV on the ground. Assign waypoints for the UAV to cover	
C.2	UAV takes off and flies to the first waypoint	The UAV takes off smoothly and maneuvers to the waypoint
C.3	UAV goes from waypoint to waypoint as instructed	All waypoints are reached using a direct path and with a maximum error of 3m.
C.4	UAV returns to staring waypoint and lands	Final UAV position is within 5m of the starting UAV position

Table 6: Fall Validation, Test C

Test ID	D	
Test Name	Static obstacle detection test	
Test Description	Validates that static obstacles are detected at least in one direction accurately	
Equipment Required	Desk, Proximity sensing system communicating to the flight controller, two sizes of obstacles (can be boards with cross sections 1.5x0.5m and 2x2m)	
Step	**Step Description**	**Success condition**
D.1	Setup desk with proximity sensor and controller system	
D.2	Initiate system with configurations	The UAV takes off smoothly and maneuvers to the waypoint
D.3	Bring 2mx2m obstacle in front of the system	Obstacle is detected in a range of 50cm to 1.5m from the system and distance is identified with a maximum error of 20 cm.

D.4	Bring 1.5mx0.5m obstacle in front of the system	Obstacle is detected in a range of 50cm to 1.5m from the system and distance is identified with a maximum error of 20 cm.

Table 7: Fall Validation, Test D

9.2 Spring Validation

Test ID	E	
Test Name	Obstacle-less package delivery test	
Test Description	Validates that packages can be delivered to a house without any obstacles in the path	
Equipment Required	Platform, Open space (outdoor environment), Fully equipped system.	
Step	**Step Description**	**Success condition**
E.1	Place UAV with package at platform with visual marker	
E.2	Initiate system by entering GPS coordinates for the house and GPS coordinates of the return to point	Delivery begins
E.3	UAV takes off autonomously towards the house	
E.4	Reaches waypoint near the house using GPS	Autonomously and accurately navigates using GPS to reach near given GPS coordinates
E.5	Identifies and navigates to the visually marked drop off point (with no obstacles in the path)	Identifies the visual marker near the GPS destination and autonomously navigates to it
E.6	Lands and drops the package	The package should be upto 2m from the center of the visual marker
E.7	Autonomously takes off and navigates to the GPS of the return back point (communicated in the beginning)	Package should remain delivered. UAV departs
E.8	Detects, navigates and lands at the platform with the visual marker	Should land within 2m of the center of the visual marker

Table 8: Spring Validation, Test E

Test ID	F (uses E)	
Test Name	Package delivery test with obstacles	
Test Description	Validates that packages can be delivered to a house even with static obstacles in the path	
Equipment Required	Platform, Open space (outdoor environment), Fully equipped system, Obstacles (cross section: 1.5m x 0.5m, 2m x 2m).	
Step	**Step Description**	**Success condition**
E.1	Place UAV with package at platform with visual marker	
E.2	Initiate system by entering GPS coordinates for the house and GPS coordinates of the return to point	
E.3	UAV takes off autonomously towards the house	
E.4	Reaches waypoint near the house using GPS	
F.1	Identifies the marker and plans path to navigate to it	Identifies the visual marker near the GPS destination.
F.2	Place an obstacle (2mx2m) in the path of the UAV	Avoids the obstacle
F.3	Place an obstacle (2mx2m) on the side of the UAVs intended path	Avoids the obstacle
F.4	Repeat with 1.5m x 0.5m obstacles	Avoids the obstacle
F.5	Repeat with both obstacles, placed in the front and the sides	Avoids the obstacles
E.6	Lands and drops the package	
E.7	Autonomously takes off and navigates to the GPS of the return back point (communicated in the beginning)	
E.8	Detects, navigates and lands at the platform with the visual marker	

Table 9: Spring Validation, Test F

10. Team Roles & Responsibilities

10.1 Team Members and Roles

Member	**Role**
Adam Yabroudi	System Engineer & Electrical Lead
Pratik Chatrath	Sensor Lead & Software Developer
Sean Bryan	Mechanical and Communication Lead
Tushar Agrawal	Software Lead and Project Manager

Table 10: Team Roles

10.2 Team Responsibilities

Role	**Primary**	**Secondary**
Technical		
Communication System	Sean Bryan	Adam Yabroudi
Mechanical Design	Sean Bryan	
Obstacle Avoidance System	Pratik Chatrath	Tushar Agrawal
UAV Control System	Adam Yabroudi	
UAV Vision System	Tushar Agrawal	Pratik Chatrath
Project Management		
Timekeeping and team follow-up	Tushar Agrawal	Sean Bryan
Sponsor Liaison	Tushar Agrawal	
Budget & Ordering	Adam Yabroudi	
Documentation		
Website	Sean Bryan	
Document Design	Sean Bryan	

Table 11: Team Responsiblities

11. Budget

Item	Comes With	Add On	QTY	price/ unit	Price
Electro-Mechanical System					
FireFly6 Pro Combo					$1519
	Px4hawk set (gps, radio, and pixhawk)				
		Configured			$20
	APM set (gps, radio, and APM)				
		Configured			$20
	High efficiency propulsion bundle				
	10x4.5 CCW props				
	10x4.5 CW props				
	Adapter ring				
	1 6s BEC				
		Turnigy Multistar High Capacity 3S 5200mah	2	$50	$100
		FrSky Taranis, Spektrum DX9, Futaba 14SG radios			$230
		Radio configuration			$20
Nica Drone Electro Permanent Magnet					$45
Extra Pixhawk					$200
				Subtotal	**$2154**
Vision System					
FF6 Gimbal					$260
camera					$200
BeagleBoard-xM					$150
PIX4FLOW kit (optical flow)					$150
				Subtotal	**$760**

Obstacle Avoidance System					
RPLidar 360 scanner					$400
Lidar-Lite v2					$115
MaxBotix Ultrasonic Rangefinder					$25
IR sensors			5	30	$150
				Subtotal	**$690**
				Total:	**$3604**

Table 12: Project Budget

12. References

1. Meier, L.; Tanskanen, P.; Fraundorfer, F.; Pollefeys, M., "PIXHAWK: A system for autonomous flight using onboard computer vision," in Robotics and Automation (ICRA), 2011 IEEE International Conference on , vol., no., pp.2992-2997, 9-13 May 2011
2. Figure 4: Dronecode Software Architecture https://www.dronecode.org/dronecode-software-platform, accessed 10/2/15
3. Web link: "Electro Permanent Magnet" from Nica Drone, http://nicadrone.com/index.php?id_product=59&controller=product, accessed 10/2/15
4. Web link: "Using LIDAR-Lite to sweep an arc for sense-and-avoid", http://diydrones.com/profiles/blogs/using-lidar-lite-to-sweep-an-arc-for-sense-and-avoid?xg_source=activity, accessed 10/2/15
5. Web link: "Precision Land ArduCopter Demo", http://diydrones.com/profiles/blogs/precision-land-arducopter-demo?xg_source=activity, 10/2/2015
6. Web link: "Raspberry Pi, Beaglebone Black, Intel Edison – Benchmarked", http://www.davidhunt.ie/raspberry-pi-beaglebone-black-intel-edison-benchmarked/, accessed 10/2/15
7. Web Link: "3D Robotics Partners with Intel, Develops New Drone Power", http://3drobotics.com/3d-robotics-partners-intel-develops-new-drone-power/, accessed 10/2/15