SWARM-BOTS

ECE4011 Senior Design Project

Section L2C, Carteblanche Team

Project Advisor: Dr. Thomas Michaels

Nakul Dureja
Henry Maccord IV
Mohammad Muhi
Matthew Pickett
Rohith Rajan
Yuanyuan Zhao

Submitted

April 21, 2014

Table of Contents

Executive Summary

The mechanical simplicity, easy maneuverability, versatility and low cost of quad-copters have made them popular with researchers and enthusiasts alike. One of the major limiting factors of a quad-copter is its payload carrying capability. A user needs to invest around $35,000 to obtain a quad-copter that has around 1 kg of payload capacity. The team proposes to design a novel approach to allow for similar payload capabilities at 1/10th the price.

The design involves implementing a tracking system that will enable an autonomous secondary quad-copter to track and follow the maneuvers of a user-controlled quad-copter. The user will control the primary quad-copter via an Android or iOS device, while the secondary quad-copter will send useful information to a base station (Laptop) that will be running a custom built GUI. The team aims to develop a modular design that could be expanded to include several quad-copters.

There are three main design components involved in the project: the optical tracking, the control mechanism and the user interface. The optical tracking mechanism would be responsible for getting the position information of the master drone, which would further be fed into the control algorithm as input. The control algorithm would calculate the necessary movement required by the slave drone to follow the master drone. These controls would then be transmitted to the slave bot. The user would be able to switch to the follow mode and back using a custom interface app. The expected outcome of the design is a fully functionally prototype that will cost around $3,500.

SWARM-BOTS

1. Introduction

The Carteblanche team intends to design a tracking system for the Parrot Drone 2.0 Quadcopter. The team is requesting $600.00 to fund the development of the prototype.

1.1 Objective

The team will design the necessary hardware and software solutions to allow optical based tracking of one Parrot Drone by another. The user will control a master bot using a mobile device or an RC controller via a custom GUI while the slave bot will autonomously follow the master.

1.2 Motivation

Recently, small scale unmanned aerial vehicles (UAVs), especially quad-copters, have become popular due to their agility and ability to perform quick and complex maneuvers [1]. Micro UAVs find applications in an array of surveillance, exploration, mapping, search and rescue fields. Because these micro UAVs are generally highly unstable non-linear systems, they require a robust control system in order to navigate and maintain their autonomy while performing tasks. The current technologies in this domain allow these UAVs to cooperate and navigate autonomously in structured as well as unstructured environments. The most widely used control systems utilize GPS data to locate the aerial unit in the 3D world. This poses theoretical challenges while navigating into areas with limited or no GPS availability or while performing tasks that require high precision location information. With the current level of advancement in embedded processors and computer vision systems, using a vision based sensor for navigation provides an alternative approach.

The motivation behind this project was to implement a cost effective method to distribute heavy payloads onto multiple aerial vehicles. The size and cost of a quadcopter increases with its payload carrying capacity. A novel approach to resolving this problem was to implement a swarm of affordable quadcopters, each with a small section of payload, which would optically track a single, user controlled quadcopter. The success of this project will provide a practical solution to a wide array of applications that range from search and rescue [1].

1.3 Background

Two key components of this project are the operation of quadcopters and the vision based navigation system. Extensive research is devoted to develop an autonomous tracking system.

1.3.1 Quadcopters

Quadcopters are mechanically unstable systems, stabilized electronically by hardware components and software. Quadcopters are mechanically unstable systems, stabilized electronically by hardware components and software. AR Drone 2.0, which is the quadcopter to be utilized for this project, is controlled by a 3-axis gyroscope, a 3-axis accelerometer, a 3-axis magnetometer, pressure sensor, 2 ultrasound sensors for ground altitude measurement, and a 60 fps QVGA vertically facing camera for ground speed measurement [2].

1.3.2 Vision Based Navigation System

The three basic building blocks of a vision based navigation system are the sensors, the processing unit and the control unit. The initial sensor setup to determine the relative XYZ coordinates is achieved either though monocular vision with some reference point in the environment or binocular vision with stereo cameras to accurately obtain the distance measurement [3]-[4]. The processing unit can either be onboard embedded system performing speeded up robust features (SURF) extraction [5] or on a ground control system hosting a multi core processor for achieving more complex image processing tasks implementing the simultaneous localization and mapping (SLAM) framework [6]. The SLAM algorithm along with the controller unit running in real time allows the micro UAV to perform complex maneuvers. The onboard controller unit handles the rotor movements following the instructions from the processing unit.

2. Project Description and Goals

The fundamental goal of the Swarm Bots design a communication control system that will allow multiple semi-autonomous aerial vehicles (slave bots) to track a single user controlled aerial vehicle (master bot). This will allow the user to have a 'swarm' of aerial vehicles in the air which he/she can easily control to perform complex tasks. The system will consist of a multiple quadcopters, LEDS, compass module, microcontrollers, transmitters, receivers and a power regulating circuit.

The slave drones should be able to:

- Establish a communication channel with the base station to
    - Send the video feed to the base station
    - Receive the control from the base station
- Follow the control instructions from the base stations and move accordingly.
- Shift to manual mode by an override operation.

The base station should be able to:

- Establish a communication channel with the slave drones to
    - Receive from the video feed from the slave drones

  - Send the processed control output to the slave drones.
- Process the video stream to track the master drone
- Execute control equations to produce movement vectors required to follow the master drone
- Convert the movement vectors to joystick controls and send it to transmitter

3. Technical Specifications

Table 1 lists the specifications for the stock Parrot AR Drone 2.0 and its Power edition. Table 2 will list the specification for the LED's that will be used in the project for the tracking aspect of the project. Table 3 will tabulate the specifications of the DX6i 6-Channel Full Range w/o Servos MD2 Transmitter for the easier use to control the Master drone from the base station. Table 4 lists the DSM2 AR6200 Receiver that will be attached with the Master and Slave drones for receiving instructions from the base station and or an another drone. Table 5 will provide us with the specification of the microcontroller that is used for power regulation between the LED's, the drone motors and flight time. And Table 6 will provide us the compass module, which will give us an accurate measurement of the rotation of the drones.

Table 1. Parrot AR Drone 2.0 and Drone 2.0 Power Edition Technical Specification

Technical Specifications	**Parrot AR Drone 2.0**	**Parrot AR Drone 2.0 Power Edition**
Front Camera Resolution	20p	720p
Front Camera Frame Rate	30fps	30fps
Wide Angle Lens	92 ° Diagonal	92 ° Diagonal
Processor	1GHz 32 bit ARM Cortex A8 Processor	1GHz 32 bit ARM Cortex A8 Processor
DSP	800 MHz video DSP TMS320DMC64x	800 MHz video DSP TMS320DMC64x
Software	Linux 2.6.32	Linux 2.6.32
Memory	1 Gbit DDR2 RAM at 200MHz	1 Gbit DDR2 RAM at 200MHz
Gyroscope Precision	2000 °/second	2000 °/second
Accelerometer Precision	+/- 50 mg	+/- 50 mg
Magnetometer Precision	6 °	6 °
Bottom Camera Resolution	QVGA	QVGA
Bottom Camera Frame Rate	60 fps	60 fps
Carbon Fiber Tube Weight	380g with outdoor hull, 420g with indoor hull (Three Total)	380g with outdoor hull, 420g with indoor hull (Three Total)
Brushless In Runner Motors Power Output	14.5 W	14.5 W
Hull Weight	380 g	380 g
Hull Dimensions	517 mm X 517 mm	517 mm X 517 mm
Pressure Sensor	+/- 10 Pa	+/- 10 Pa
Battery	1000 mAh	1500

Table 2. LED Strip Light Specifications

Product Specifications (LED Strips)	
LED Type	3528 SMD
LED Quantity	300 leds/roll, 5m/roll
Power Consumption	4.8 W/m, 16-20W/roll
Protection Grade	IP65 (Waterproff)
Emitter Color	Color Name
Viewing Angle	120 Degrees
Operating Voltage	12 V DC
Power Adapter for 5M	DC 12V 2A (24W)
Size (+/- 1mm)	L5000mm*W8mm
Storage Temperature	-20 °C
Life Span	50,000 Hours

Table 3. DX6i 6-Channel Full Range w/o Servos MD2 Specifications

Product Specification (Transmitter)	
Modulation	DSM2
Band	2.4 GHz
Receiver	AR6200
Programming Features	Helicopter & Planes
Model Memory	10
Modes	Mode 2
Transmitter (TX) Battery Type	AA NIMH 150 mAh Battery
Charger	4-Cell 150mAh wall charger

Table 4. Product Specification for DSM2 AR6200 Receiver

Product Specifications (Receiver)	
Type	Full Range Aircraft Receiver
Number of Channels	6
Modulation	DSM2
Band	2.4 GHz
Length	30.1 mm
Width	21.6mm
Height	12.3mm
Weight	10g
Voltage Range	3.5-9.6 V
Antenna Length	Main Rx: 30mm; Remote Rx: 30mm

Table 5. Product Specification of Arduino Mega Microcontroller

Product Specification (Microcontroller)	
Microcontroller	ATmega1280
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	128 KB of which 4 KB used by boot loader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Table 6. Product Specification of Compass Module

Product Specifications (Compass)	
Voltage Supply Range	2.7 to 5.2V
Update Rate	1 to 20 Hz
Heading Resolution	0.5 Degrees
Repeability	1 Degrees
Supply Current	1mA @ 3V

4. Design Approach and Details

4.1 Design Approach

The project aims to design a communication and control system that will allow a semi-autonomous aerial vehicle (slave) to track and follow a single user controlled aerial vehicle (master). There are three main design components involved in the project: the optical tracking, the control mechanism and the user interface. The optical tracking mechanism would be responsible for getting the position information of the master drone, which would further be fed into the control algorithm as input. The control algorithm would calculate the necessary movement required by the slave drone to follow the master drone. These controls would then be transmitted to the slave bot. The user would be able to switch to the follow mode and back using a custom interface app.

4.1.1 Optical Tracking Mechanism

Colored bot. markers and LED strips of red, blue, green and orange colors would be integrated with the hull of the master drone.

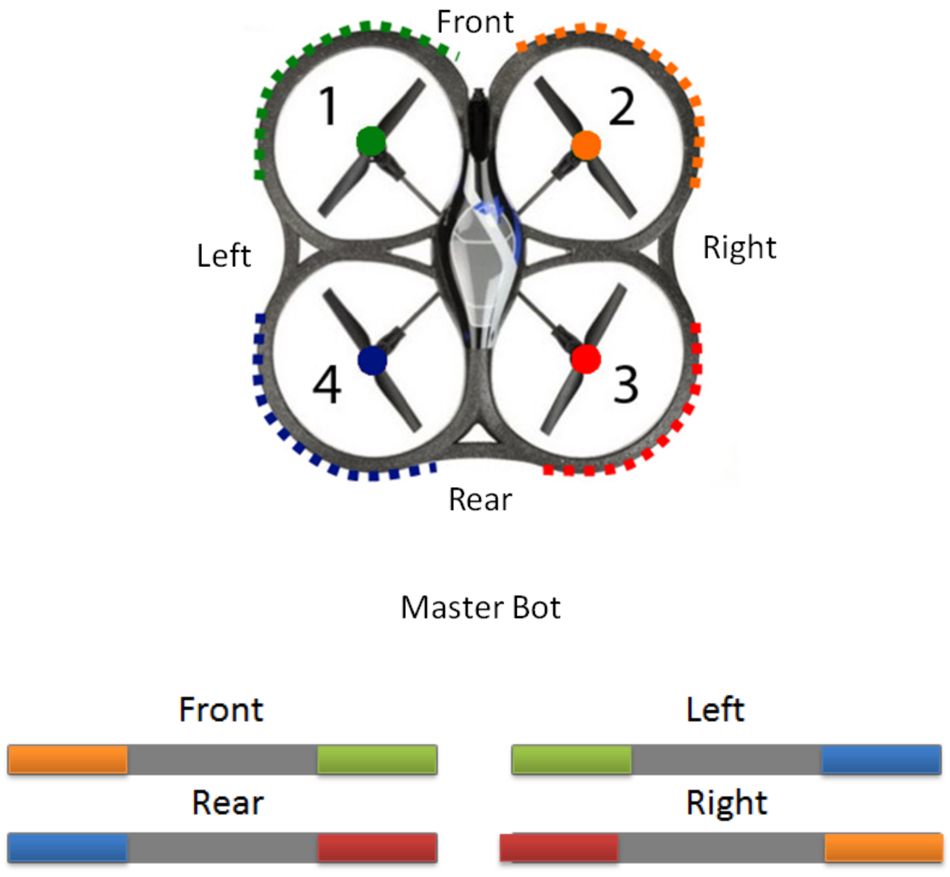


Figure 1. Top end side view of Master-bot with color coded LED strips.

Optical tracking methods would be applied to the video feed from the slave drone being sent to the base station. The color code and the onboard compass reading would determine the orientation of the master drone. Blob detection technique would be used to extract the X and Y pixel coordinates of the centroids of all the colored strips mentioned above in the frame. Trigonometric methods would be used to determine the distance from the master bot using the known distance between the colored strips on the master bot.

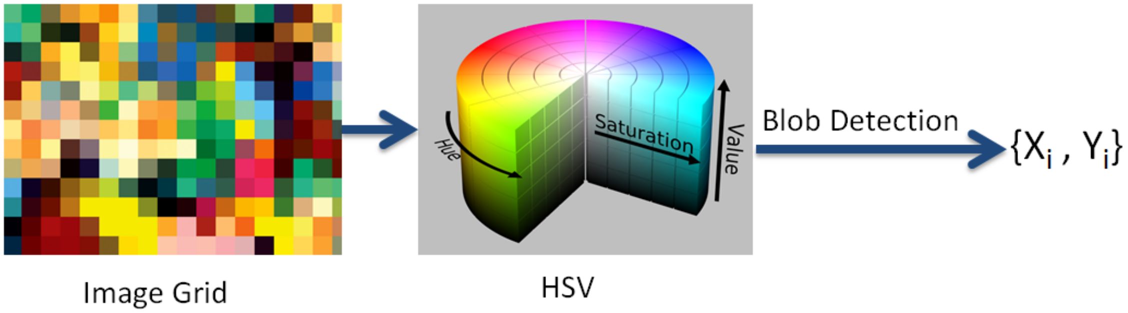


Figure 2. Blob detection mechanism

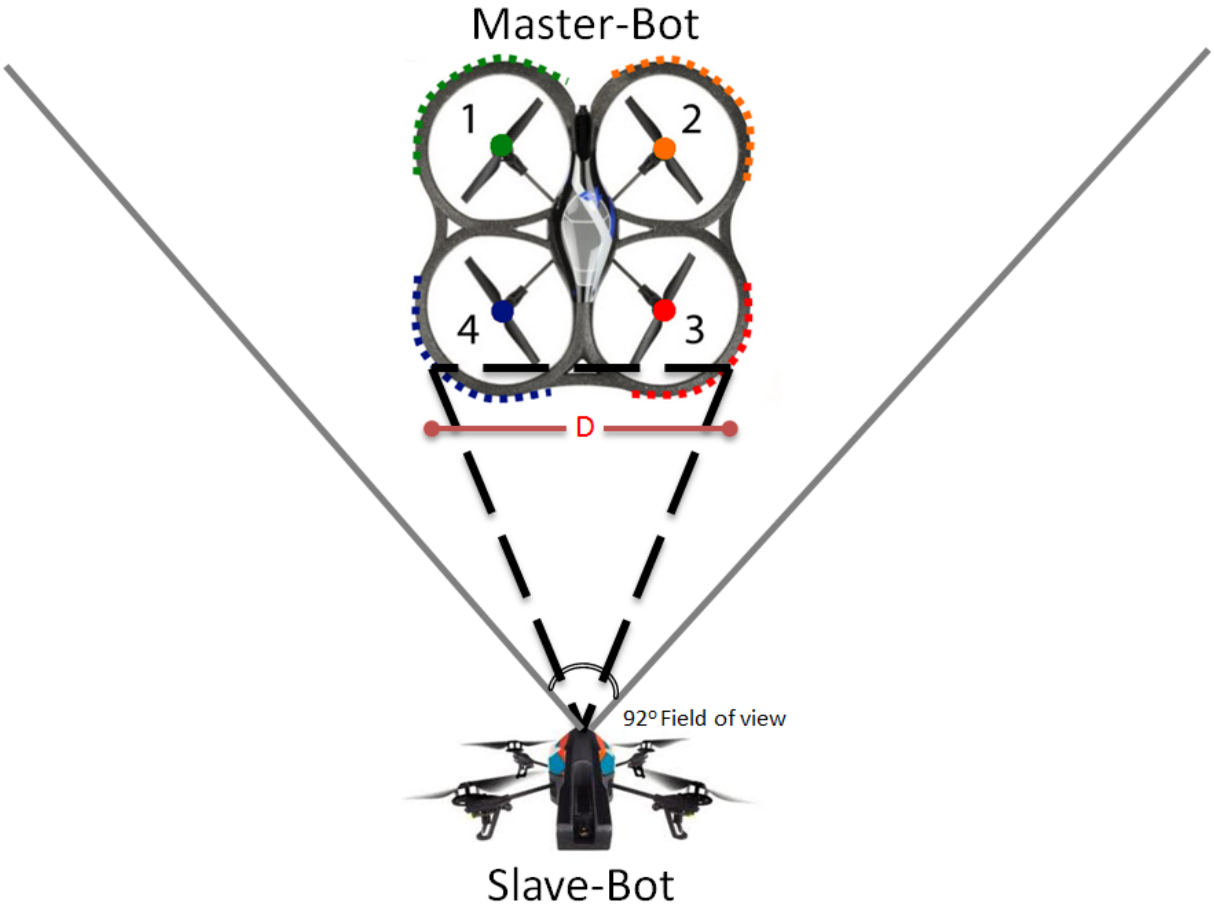


Figure 3. Master – Slave system

4.1.2 Control Mechanism

The user would control the master drone and the base station would control the slave drone to make it follow the master drone. The output data from the tracking mechanism would provide the position and orientation information about the master drone which would serve as inputs to the control mechanism. The control algorithm would calculate the movement vector required to move the slave drone in order to follow the master drone. The control movement vector would be sent to the slave drone via transmitter.

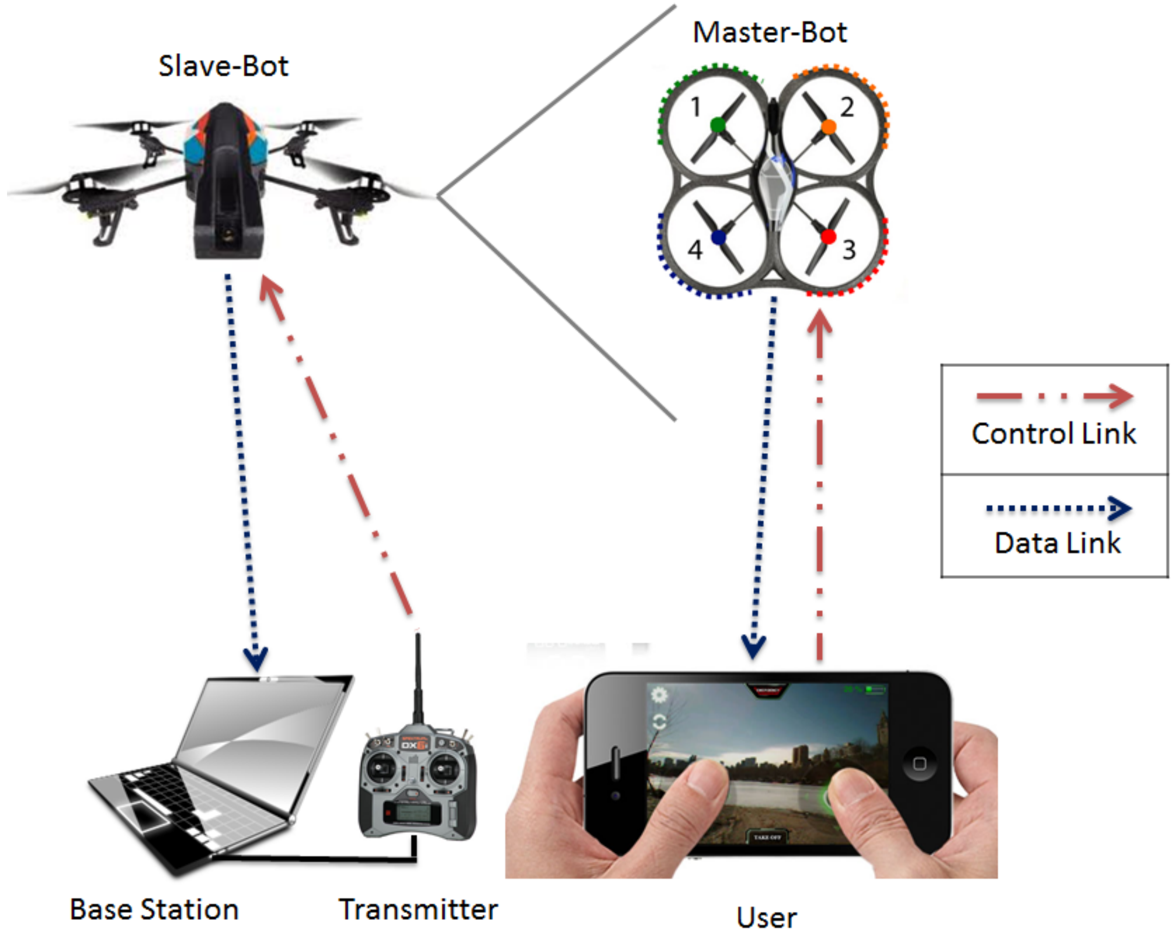


Figure 4. Data and Communication System design overview.

Figure 5 shows the control flow of the process. After the user turn on the slave drone, it would take off and hover. If the manual mode is turned off, the slave bot would look for the master bot. If master bot found, the slave drone would follow it at a preset distance and formation. If the slave drone does not find the master bot or at any time in the process loses the track of the master bot, it would do three 360° rotations to try to reestablish the visual connection. If the master bot is not found after the three rotations, the slave drone would land on the ground.

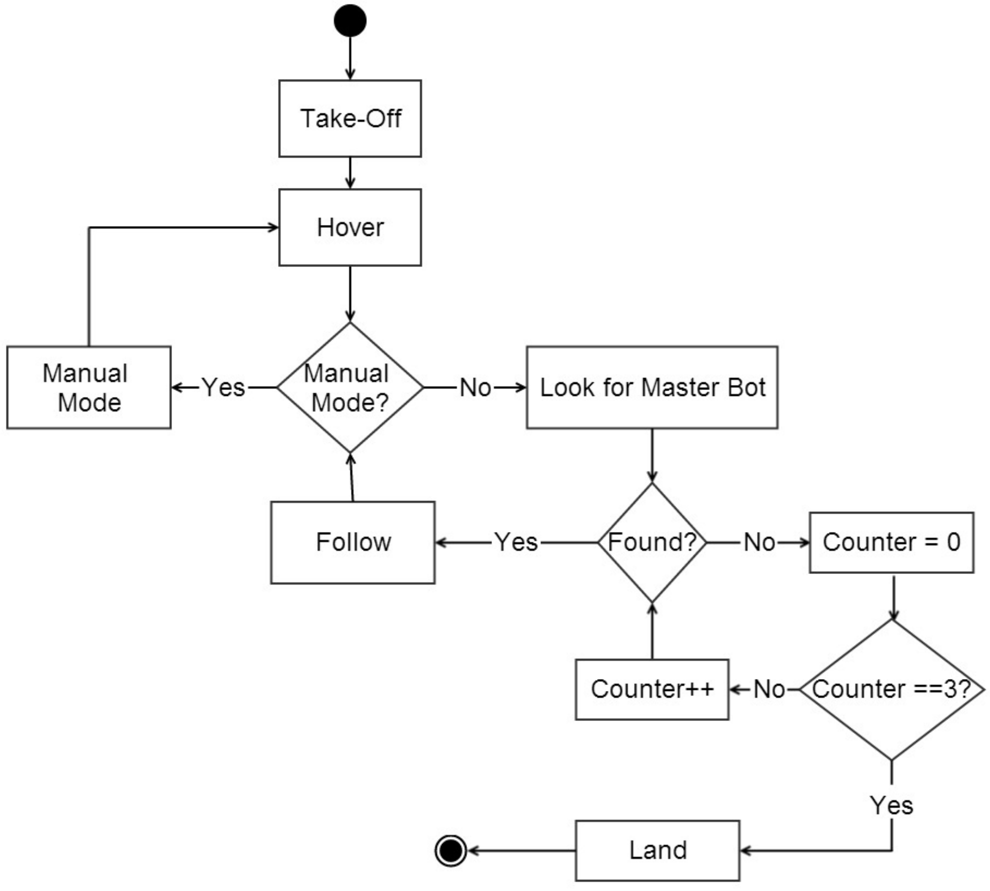
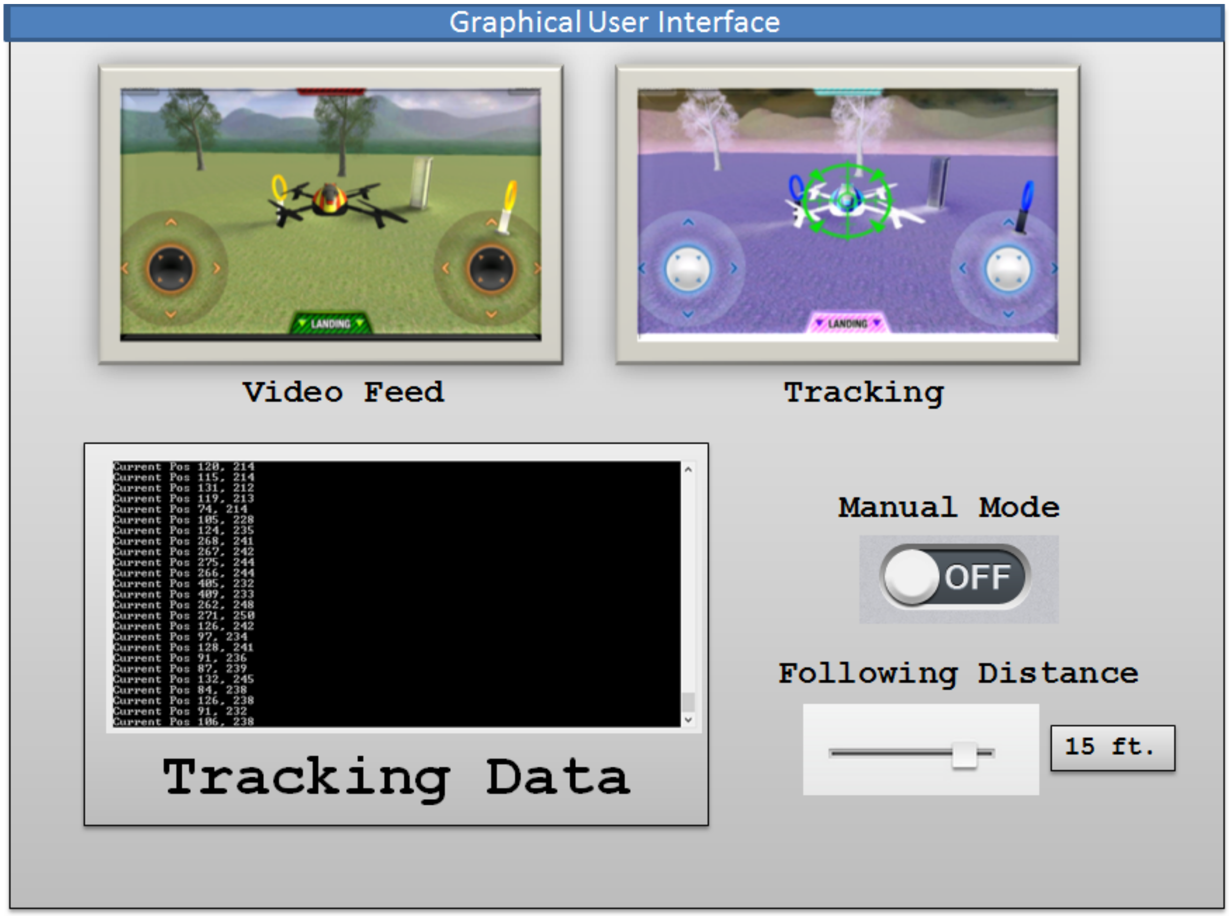


Figure 5. Process flow flowchart

4.1.3 User Interface

The user would be able to control the slave bot's movement and reactions using a custom user interface application. The interface shows the live video feed from the slave drone and its processed version detecting the master bot. The interface also allows the user to set the distance from the master drone to maintain. For safety concerns and regulations, at any time during the flight, the control of the slave drone can be shifted to the user by entering the manual mode. Entering the manual mode overrides all the autonomous control mechanism of the base station and gives the user full control.



```
Current Pos 120, 214
Current Pos 115, 214
Current Pos 131, 212
Current Pos 119, 213
Current Pos 74, 214
Current Pos 105, 228
Current Pos 124, 235
Current Pos 268, 241
Current Pos 267, 242
Current Pos 275, 244
Current Pos 266, 244
Current Pos 405, 232
Current Pos 409, 233
Current Pos 262, 248
Current Pos 271, 250
Current Pos 126, 242
Current Pos 97, 234
Current Pos 128, 241
Current Pos 91, 236
Current Pos 87, 239
Current Pos 132, 245
Current Pos 84, 238
Current Pos 126, 238
Current Pos 91, 232
Current Pos 106, 238
```

Figure 6. Proposed Graphical User Interface layout.

4.2 Codes and Standards

1. IEEE 802.11 is the method of communication that shall be implemented for connectivity between the ground station (computer), the Master Bot and the Slave bots [7]. IEEE 802.11 itself is already used by mobile devices, computers and the drones itself.
2. Protocols for communications and control are through the User Datagram Protocol (UDP) and the Transmission Control Protocol (TCP). The UDP is the protocol that exchanges information between the application layer and the internet layer with no "handshake" system [8].
    a. This makes the UDP an unreliable system to use for some particular cases that requires transmission of sensitive information. But UDP is good for transmitting information quickly from layers.
    b. TCP is very reliable with its error checked delivery system [9]. With its error checked system, the TCP is noted for its lag and redundancy. The UDP is defined in Request For Comment (RFC) 764 and the TCP is defined in RFC 793. These are established guidelines by Internet Engineering Task Force and the Internet Society to define how the protocols are standardized and utilized.
3. Universal Serial Bus (USB) 3.0 will be used to connect the receiver and the transmitter to the base station of the drones for setting up the establishment of the communication system. USB 3.0 is used for fast and reliable transfer of data that are physically connected to each other [10]. USB is also used to establish the interface between the drone control housing and an onboard/computer processor.

4.3 Constraints, Alternatives, and Tradeoffs

1. **Control System Design vs. Payload:** One implementation of the system is where the quad-copter sees only the central processing station such as a laptop (hence referred to as a 1-to-1

system) while the second implementation involves a quad-copter with on board processing providing it with autonomy. From a payload perspective a 1-to-1 system offers lesser payload than an on-board processing solution. Also to ensure optimal software performance additional hardware may need to be added which consequently increases the payload. The current proposed solution is to opt for a 1-to-1 system to obtain a successful working model and eventually build it to allow for autonomy.

2. **Communication System Design vs. Delays:** While a single-channel approach ensures simplicity and real time data at the cost of processing delays, a multi-channel approach introduces complexity, real time delay but ensures faster processing capabilities. The current proposed solution is to opt for a single channel approach in order to obtain more accurate data that can be later changed as per the final system design.
3. **Accuracy vs. Modularity:** The project aims to strike a balance between the number of quad-copters in the air and the resolution of the data stream. The modularity will depend on how generic the communication and control system are and how easily they can be cloned to create clusters. The accuracy of the data will depend on the data stream resolution. A 1-to-many system is less modular and less accurate as it splits the data channel while a 1-to-1 system would be more modular and accurate as it maintains the data channel. The current proposed solution is to opt for the 1-to-1 system to ensure data accuracy.

5. Schedule, Tasks, and Milestones

Individual tasks in this project are listed with members assigned, risk level, and difficulty level in Appendix A. The project will be designed, implemented, and tested in a span of six months, with three months of summer vacation, according to the Gannt chart in Appendix B. Software development and hardware development will occur simultaneously and both need to be completed before Oct 3rd in order to start the system test stage. The critical path shown in the network diagram in Appendix C will mainly go through hardware development and test stage, and the project will be completed by Oct 24th. The tasks with highest risk will happen in the test stage. Depending on test results, one or more weeks will be needed for review and debugging.

6. Project Demonstration

Test/Feature	Description	Verification
Slave Recognition of Master Bot	The slave copter is able to distinguish the master bot from the surroundings and determine positional data via video.	Slave sends back a verification signal alongside the video stream to the base station.
Slave Auto-Lock	The slave copter is able to find the master bot if it loses the master from its field of view. The slave will proceed to rotate while hovering to reestablish a visual connection with the master.	The slave will be manually pointed away from the master and then autonomous mode will be engaged, allowing the slave to find the master on its own.
Slave Safe Recover	If the slave is not able to recover vision of the master bot, it will safely land itself. This will only occur after predetermined attempts to reestablish sight of the master. Manual mode can still be engaged at this time.	The master will be completely removed from sight of the slave to force it into this mode. The slave will then land and send a message back to the base station.
Slave Manual Mode	At any point, a human user will be able to override any autonomy and take manual control of the slave copter(s).	A manual control system will be set up with a priority switch that will give the user total control of the slave.

Slave Track Master Rotation	If the master bot rotates while hovering, or takes a curved path, the slave bots will adapt their flight vector to remain in the same position in space relative to the flight vector of the master bot.	Master bot will hover in place and then rotate slowly. The slave bots should then move in a diagonal/circular path in the same rotational direction as the master.
Slave Out of Range	If the slave loses connection with the base station, it will automatically cease translational movement and land.	Connection between the slave and the base station will be purposefully terminated to force the slave to engage this precaution.
Slave Terrain Capture	Slave hovers in place and takes a capture from its vantage point. After this point, the slave will continue autonomous activity by reconnecting visually to the master bot. This feature will only be available while the master bot is in hover mode.	Slave bots will be able to temporarily halt motion and take a still image of the terrain with the camera. This image will be sent back to the base station, but for alternative processing determined by the application.
Slave Follow	The slave bot will use a video feed that is processed at a base station that determines position relative to a master bot based on visual references. This will be done with a single camera referencing multiple markers on the master bot, providing slaves with distance and angular information. Depending on the number of slave bots, different formations will be necessary to guarantee the safety of slave bots.	Formations will be configurable from the base station, and the interface will display data relevant to the position of the slave relative to the master bot, including raw and/or processed video feeds. The slave bots will remain within 20% of the set distance from the master, as set in interface.
Low Battery Notification	Data regarding the remaining capacity of the battery on each craft will be monitored. When the energy remaining is below a given threshold, the craft lands.	The base station will display energy information in the interface. When the battery drops below 25%, a warning message will be displayed. At 15% remaining capacity, a message will be displayed and the link with the base station will be disconnected.

7. Marketing and Cost Analysis

7.1 Marketing Analysis

There are currently no commercially available quadcopters that have a completely autonomous optical tracking capability. The success of this project will help in developing a novel approach to a distributed payload system by which the user can achieve a higher payload to system ratio while lowering costs. This could find applications in research, search and rescue operations, road accident mitigation and even novel applications for RC enthusiasts. Current research and design done in autonomous optical tracking involve pre-programming the flight path onto one or more quadcopters prior to takeoff. The MIT drone uses laser and rangefinders to localize itself in a preloaded 3D map for obstacle avoidance [11]. The team aims to add a new dimension by offering a modular design that will track the complex maneuvers of a quadcopter using only real time data.

7.2 Cost Analysis

7.2.1 Bill of Materials

Table 5 lists the equipment that needs to be procured at this stage of the project to complete a working prototype. In order to meet the proposed budget of $600.00, the advisor will sponsor the two Parrot Drones, replacement batteries, DX6i and the DSM2 receiver. Every customer purchasing this product would be required to have an Android or an iOS device to control the Master-bot, and as a result the cost of such a device has not been included in the table.

Table 7. Bill of Materials

Product Description	**Quantity**	**Unit Price ($)**	**Total Price ($)**
Parrot Drone 2.0	2	299.99	599.98
LED Strips (R/G/B/O)	4	7.68	30.72
HMC6352 – Compass Module	2	34.95	69.90
Arduino MEGA 32	1	64.99	64.99
Parrot Drone 2.0 - Battery	2	59.95	119.90
Android or iOS Device	1	0.00	0.00
DX6i 6-Channel Full Range w/o Servos MD2	1	199.99	199.99
DSM2 AR 6200 6-Channel Receiver Ultra Lite	1	79.99	79.99
Total Cost ($)			**1165.47**

7.2.2 Development Costs

The development costs for the product are created with a $30 per hour rate per team member which amounts the average engineer salary of $65,000 per year. There is an equal split between the hardware and software development costs.

Table 8. Development Costs

Project Task	**Labor (Hours)**	**Labor Costs ($)**	**Part Costs ($)**	**Total Component Costs ($)**
Software Development				
Distance Detection	40	1200.00	30.72	1230.72
Rotation Detection	80	2400.00	69.90	2469.90
Co-ordinates to Vector Conversion	40	1200.00	299.99	1499.99
Vector to Joystick Conversion	40	1200.00	279.98	1479.98
AR Drone API Hack	40	1200.00		1200.00
Software Interface Design	80	2400.00		2400.00
Hardware Development				
Hardware Component Research	40	1200.00		1200.00
Hardware Component Simulation	40	1200.00		1200.00
Parts Procurement	80	2400.00		2400.00
Layout and Fabrication	80	2400.00		2400.00
Prototype Assembly	40	1200.00	484.88	1684.88
Application Specific Add-Ons	80	2400.00		2400.00
Total Costs ($)				**21,565.47**

Assuming a 35% fringe benefits and 120% overhead, the development costs of the drone totals $ 60,952.03, as shown below in Table 7.

Table 9. Total Developmental Cost

Parts	**$ 1,165.47**
Labor	**$ 20,400.00**
Fringe Benefits, % of Labor	$ 6,140.00
Subtotal	$ 27,705.47
Overhead, % of Matl, Labor & Fringe	$ 33,246.56
Total	$ 60,952.03

The production run will consist of 5,000 units sold over a 5 year period at a price of $ per unit. The assembly of the LED system and the DX6i transmitter and the DSM2 receiver will be completed in 2 hours by a group of technicians employed at hourly rate of $ 15. Additional testing of the completed product will take additional 2 hours. The fringe benefits and overhead costs remain the same as before. Sales expense in terms in the form of advertising will up 10% of selling price, which be $350. At $3,500 per unit, the expected revenue is $17,500,000.00, yielding a profit of $ 500 per unit. The production costs, profit, and selling price of the product are indicated in Table 8.

Table 10. Selling Price and Profit per Unit (based on 5,000 unit production)

Parts Cost	**$ 1166**
Assembly Labor	**$ 30**
Testing Labor	**$ 30**
Total Labor	$ 60
Fringe Benefits, % of Labor	$ 10
Subtotal	$ 1,236
Overhead, % of Matl, Labor & Fringe	$ 1,483
Subtotal, Input Costs	$ 2,719
Sales Expense	$ 350
Amortized Development Cost	$ 31
Subtotal, All Costs	$ 3,100
Profit	$ 400
Selling Price	$ 3,500

8. Current Status

The team has a working version of software that filters an image based on a predetermined color and is able to track the position of the object as it moves along the screen. This software can distinguish multiple colors and objects simultaneously and independently while ignoring noise from colors in the surroundings. The next item to be completed is to adapt the current software version to recognize two objects of the same color as a single unit and track the centroid of that unit. It will also calculate and recording the distance between the centroids of the individual objects. Given a known separation distance and a perpendicular viewing plane, this will lead to the algorithm necessary to determine how far away the object is from the camera. Also, the method for powering and controlling the four arrays of LED's that will be on the master craft is in development.

9. References

[1] S.Weiss, D. Scaramuzza, and R.Siegwart. (2011, Dec.). Monocular-SLAM–based navigation for autonomous micro helicopters in GPS-denied environments. Journal of Field Robotics [Online]. 28(1), pp 854–874. Available: http://onlinelibrary.wiley.com/doi/10.1002/rob.20412/full

[2] Parrot, Technical Specifications, April 21, 2014. [Online]. Available: http://www.parrotshopping.com/uk/p_parrot_product.aspx?i=250846 [Accessed: Apr. 21, 2014]

[3] J. Courbon, Y. Mezouar, N. Guénard, P. Martinet. (2010, July). Vision-based navigation of unmanned aerial vehicles. Control Engineering Practice [Online]. 18(7), pp 789–799. Available: http://www.sciencedirect.com/science/article/pii/S0967066110000808

[4] T. Huntsberger, H. Aghazarian, A. Howard and D.C. Trptz. (2011, Dec.). Stereo Vision–Based Navigation for Autonomous Surface Vessels. Journal of Field Robotics [Online]. 28(1), pp 3–18. Available: http://onlinelibrary.wiley.com/doi/10.1002/rob.20380/full

[5] T. Krajnik, M. Nitsche, S. Pedre, L. Preucil, M. E. Mejail, “A simple visual navigation system for an UAV,” in 9th

[6] E.R. Davies. (2012, Mar. 19). Computer and Machine Vision: Theory, Algorithms, Practicalities (4thed.) [Restricted Electronic Resource]. Available: Georgia Tech Library Catalog. International Multi-conference on Systems, Signals and Devices, 2012.

[7] IEEE Standard for Information technology--Telecommunications and information exchange between systems Local and metropolitan area networks--Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)* pp.1 - 2793, March 29 2012

[8] Request for Comment, 768, 28 August 1980

[9] Request for Comment, 793, September 1981

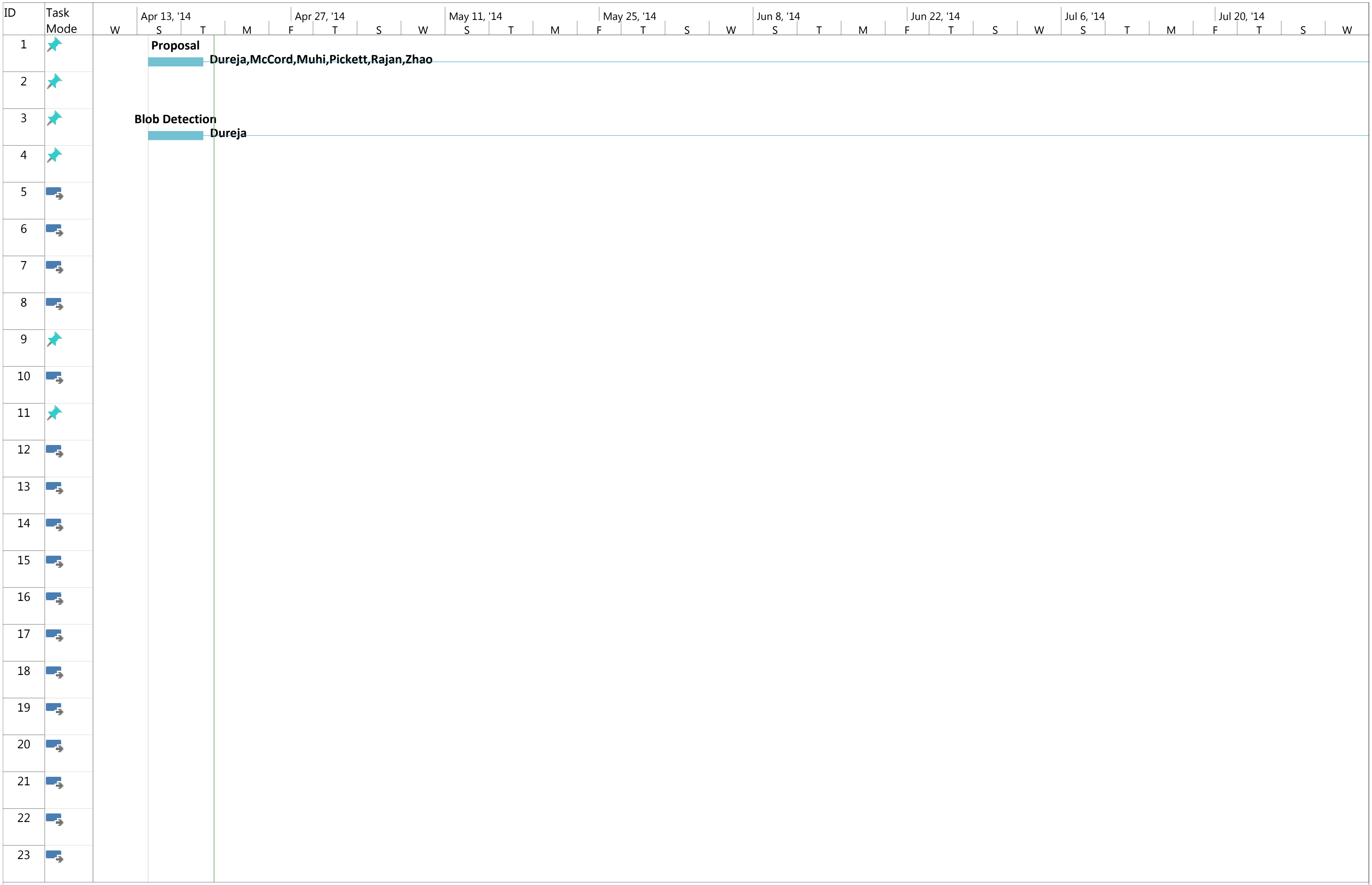
[10] *Universal Serial Bus 3.0 Specification*, Universal Serial Bus, 2008.

[11] E. Ackerman, “Laser-equipped MAV demonstrates aggressive autonomous flight,” *IEEE Spectrum*, May 16, 2012. [Online]. Available: http://spectrum.ieee.org/automaton/robotics/artificial-intelligence/laserequipped-mav-demonstrates-aggressive-autonomous-flight

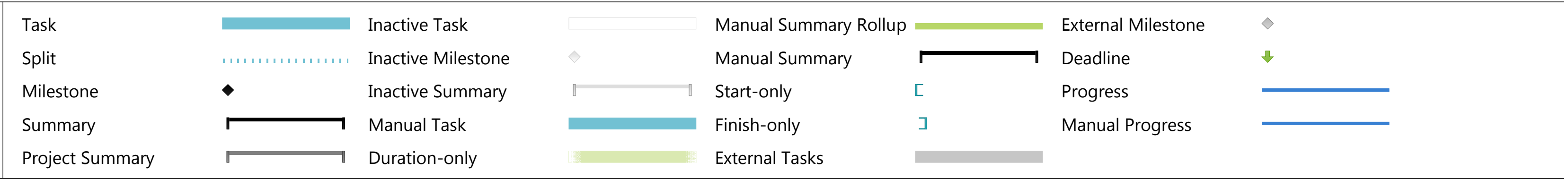
Appendix A. Task Sheet

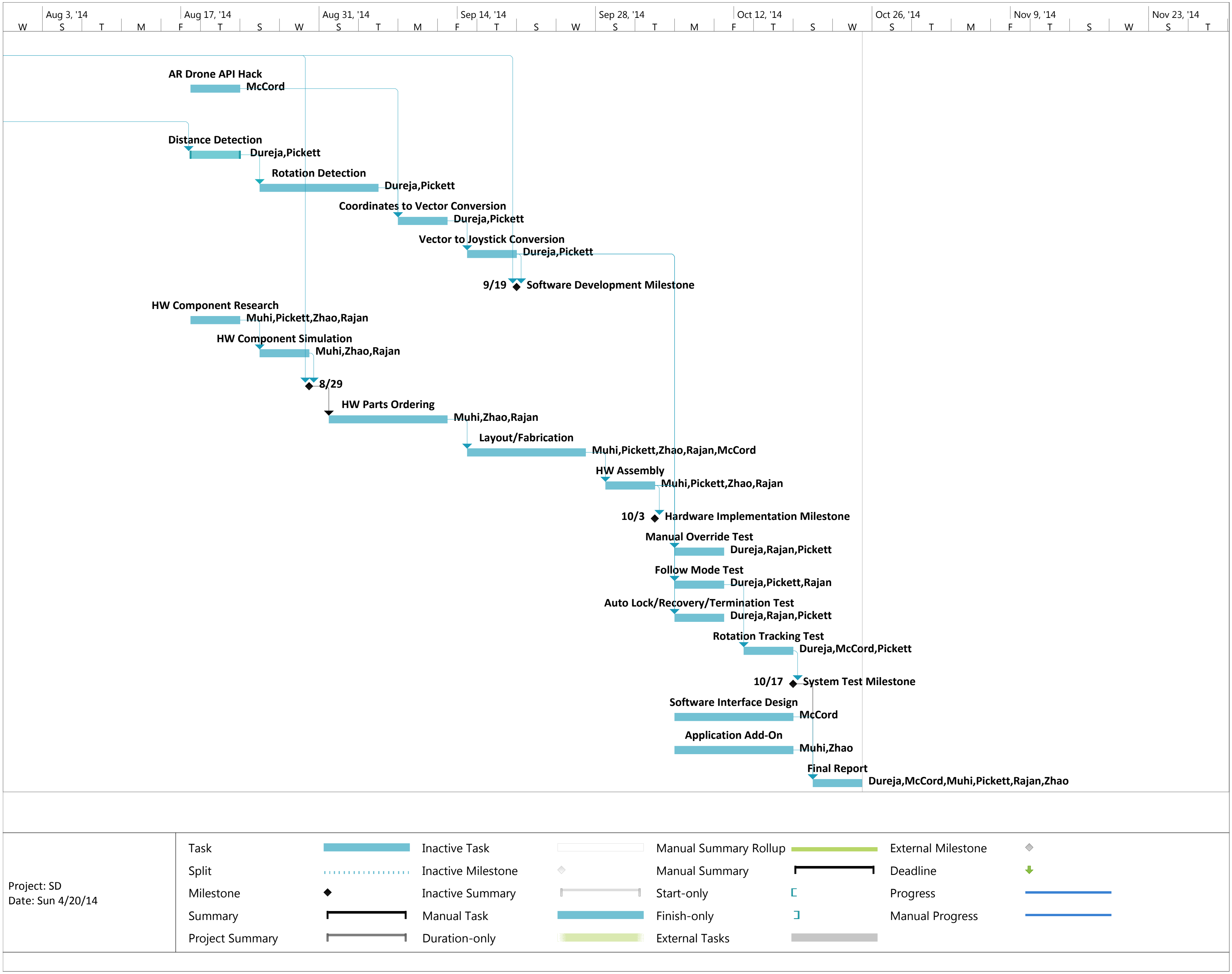
ID	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names	Milestone	Risk Level	Difficulty Level
1	📌	Proposal	1 wk	Mon 4/14/14	Fri 4/18/14		Dureja,McCord,Muhi,Pickett,Rajan,Zhao	No	low	low
2	📌	AR Drone API Hack	1 wk	Mon 8/18/14	Fri 8/22/14		McCord	No	medium	medium
3	📌	Blob Detection	1 wk	Mon 4/14/14	Fri 4/18/14		Dureja	No	medium	medium
4	📌	Distance Detection	1 wk	Mon 8/18/14	Fri 8/22/14	3	Dureja,Pickett	No	medium	medium
5	➜	Rotation Detection	2 wks	Mon 8/25/14	Fri 9/5/14	4	Dureja,Pickett	No	medium	high
6	➜	Coordinates to Vector Conversion	1 wk	Mon 9/8/14	Fri 9/12/14	5,2	Dureja,Pickett	No	medium	medium
7	➜	Vector to Joystick Conversion	1 wk	Mon 9/15/14	Fri 9/19/14	6	Dureja,Pickett	No	medium	medium
8	➜	Software Development Milestone	0 days	Fri 9/19/14	Fri 9/19/14	1,7		Yes	medium	medium
9	📌	HW Component Research	1 wk	Mon 8/18/14	Fri 8/22/14		Muhi,Pickett,Zhao,Rajan	No	low	low
10	➜	HW Component Simulation	1 wk	Mon 8/25/14	Fri 8/29/14	9	Muhi,Zhao,Rajan	No	medium	low
11	📌	Hardware Design Milestone	0 days	Fri 8/29/14	Fri 8/29/14	1,10		Yes	medium	low
12	➜	HW Parts Ordering	2 wks	Mon 9/1/14	Fri 9/12/14	11	Muhi,Zhao,Rajan	No	high	low
13	➜	Layout/Fabrication	2 wks	Mon 9/15/14	Fri 9/26/14	12	Muhi,Pickett,Zhao,Rajan,McCord	No	medium	medium
14	➜	HW Assembly	1 wk	Mon 9/29/14	Fri 10/3/14	13	Muhi,Pickett,Zhao,Rajan	No	medium	medium
15	➜	Hardware Implementation Milestone	0 days	Fri 10/3/14	Fri 10/3/14	14		Yes	medium	medium
16	➜	Manual Override Test	1 wk	Mon 10/6/14	Fri 10/10/14	7,14	Dureja,Rajan,Pickett	No	medium	medium
17	➜	Follow Mode Test	1 wk	Mon 10/6/14	Fri 10/10/14	7,14	Dureja,Pickett,Rajan	No	medium	medium
18	➜	Auto Lock/Recovery/Termination Test	1 wk	Mon 10/6/14	Fri 10/10/14	7,14	Dureja,Rajan,Pickett	No	high	medium
19	➜	Rotation Tracking Test	1 wk	Mon 10/13/14	Fri 10/17/14	17	Dureja,McCord,Pickett	No	high	medium
20	➜	System Test Milestone	0 days	Fri 10/17/14	Fri 10/17/14	19		Yes	high	medium
21	➜	Software Interface Design	2 wks	Mon 10/6/14	Fri 10/17/14		McCord	No	low	medium
22	➜	Application Add-On	2 wks	Mon 10/6/14	Fri 10/17/14		Muhi,Zhao	No	low	medium
23	➜	Final Report	1 wk	Mon 10/20/14	Fri 10/24/14	20,21,22	Dureja,McCord,Muhi,Pickett,Rajan,Zhao	No	low	low

Appendix B. Gantt Chart



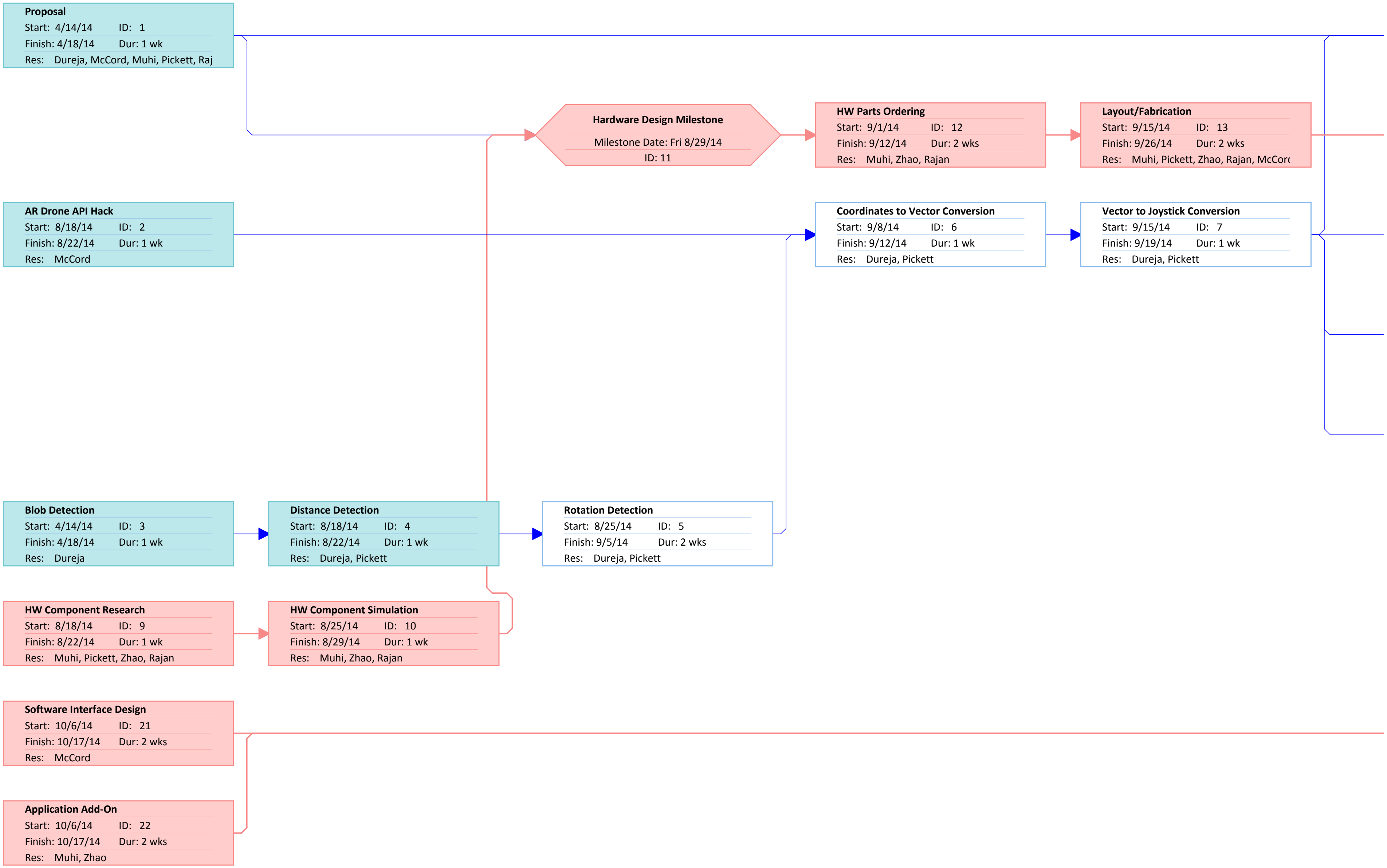
Project: SD
Date: Sun 4/20/14





Project: SD
Date: Sun 4/20/14

Appendix C. Network Diagram



Proposal
Start: 4/14/14 ID: 1
Finish: 4/18/14 Dur: 1 wk
Res: Dureja, McCord, Muhi, Pickett, Raj

Hardware Design Milestone
Milestone Date: Fri 8/29/14
ID: 11

HW Parts Ordering
Start: 9/1/14 ID: 12
Finish: 9/12/14 Dur: 2 wks
Res: Muhi, Zhao, Rajan

Layout/Fabrication
Start: 9/15/14 ID: 13
Finish: 9/26/14 Dur: 2 wks
Res: Muhi, Pickett, Zhao, Rajan, McCor

AR Drone API Hack
Start: 8/18/14 ID: 2
Finish: 8/22/14 Dur: 1 wk
Res: McCord

Coordinates to Vector Conversion
Start: 9/8/14 ID: 6
Finish: 9/12/14 Dur: 1 wk
Res: Dureja, Pickett

Vector to Joystick Conversion
Start: 9/15/14 ID: 7
Finish: 9/19/14 Dur: 1 wk
Res: Dureja, Pickett

Blob Detection
Start: 4/14/14 ID: 3
Finish: 4/18/14 Dur: 1 wk
Res: Dureja

Distance Detection
Start: 8/18/14 ID: 4
Finish: 8/22/14 Dur: 1 wk
Res: Dureja, Pickett

Rotation Detection
Start: 8/25/14 ID: 5
Finish: 9/5/14 Dur: 2 wks
Res: Dureja, Pickett

HW Component Research
Start: 8/18/14 ID: 9
Finish: 8/22/14 Dur: 1 wk
Res: Muhi, Pickett, Zhao, Rajan

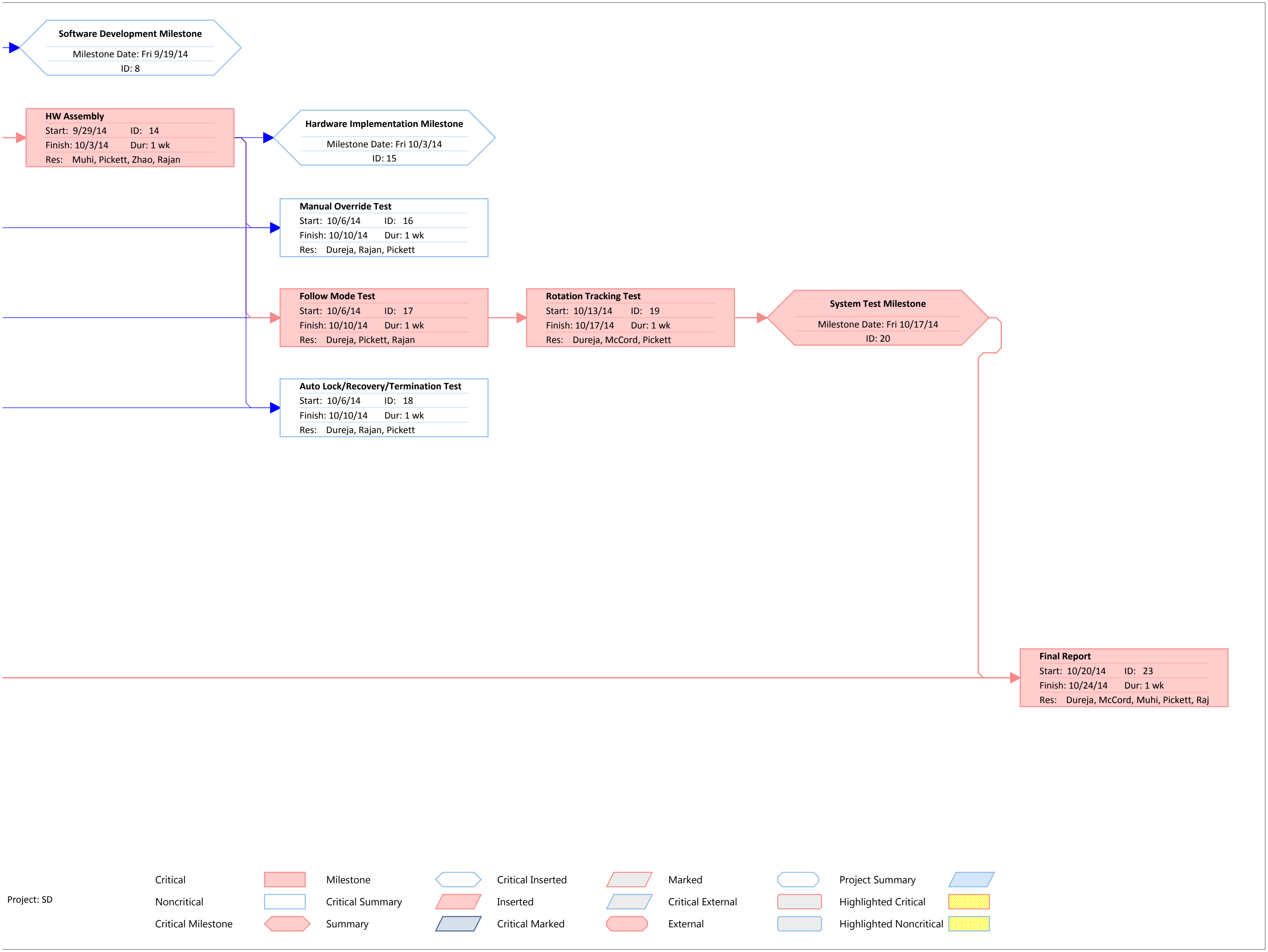
HW Component Simulation
Start: 8/25/14 ID: 10
Finish: 8/29/14 Dur: 1 wk
Res: Muhi, Zhao, Rajan

Software Interface Design
Start: 10/6/14 ID: 21
Finish: 10/17/14 Dur: 2 wks
Res: McCord

Application Add-On
Start: 10/6/14 ID: 22
Finish: 10/17/14 Dur: 2 wks
Res: Muhi, Zhao

Project: SD

Critical	🟥	Milestone	⬡	Critical Inserted	▱	Marked	⬭	Project Summary	▱		
Noncritical	☐	Critical Summary	▰	Inserted	▱	Critical External	⬭	Highlighted Critical	▨		
Critical Milestone	⬢	Summary	▰	Critical Marked	⬭	External	⬭	Highlighted Noncritical	▨		


Software Development Milestone
Milestone Date: Fri 9/19/14
ID: 8
HW Assembly
Start: 9/29/14
ID: 14
Finish: 10/3/14
Dur: 1 wk
Res: Muhi, Pickett, Zhao, Rajan
Hardware Implementation Milestone
Milestone Date: Fri 10/3/14
ID: 15
Manual Override Test
Start: 10/6/14
ID: 16
Finish: 10/10/14
Dur: 1 wk
Res: Dureja, Rajan, Pickett
Follow Mode Test
Start: 10/6/14
ID: 17
Finish: 10/10/14
Dur: 1 wk
Res: Dureja, Pickett, Rajan
Rotation Tracking Test
Start: 10/13/14
ID: 19
Finish: 10/17/14
Dur: 1 wk
Res: Dureja, McCord, Pickett
System Test Milestone
Milestone Date: Fri 10/17/14
ID: 20
Auto Lock/Recovery/Termination Test
Start: 10/6/14
ID: 18
Finish: 10/10/14
Dur: 1 wk
Res: Dureja, Rajan, Pickett
Final Report
Start: 10/20/14
ID: 23
Finish: 10/24/14
Dur: 1 wk
Res: Dureja, McCord, Muhi, Pickett, Raj
Project: SD
Critical
Noncritical
Critical Milestone
Milestone
Critical Summary
Summary
Critical Inserted
Inserted
Critical Marked
Marked
Critical External
External
Project Summary
Highlighted Critical
Highlighted Noncritical