First Semester Progress Report

Programmable Flight Controller

Brendon Camm	Lucas Doucette	Dylan Humber
B00649176	B00685962	B00695554

Submitted: December 6, 2016

Abstract

Contents

1 List of Acronyms

1. GUI - Graphical User Interface
2. PWM - Pulse Width Modulation
3. BLDC - Brushless Direct Current
4. USB - Universal Serial Bus
5. FOV - Field of View
6. RF - Radio Frequency

2 Objectives and Deliverables

2.1 Overall Objective

The main objective of the project is to develop a programmable flight controller that responds appropriately to control inputs and disturbances. The flight controller will recieve control inputs over WI-FI from a base station. The base station can be any device that is WI-FI enabled and has the appropriate software installed. The software on the base station will be a graphical user interface (GUI) that allows the user to send control inputs to the drone and view statistics of the drone during operation.

The objectives of the project have been broken down into what will be accomplished in the first semester (Short term objectives) and what will be accomplished in the second semester (Long term objectives). These objectives along with specific details of each can be viewed in sections 2.2 and 2.3 respectively.

2.2 Short Term Objectives (October 2016 - December 2016)

2.2.1 Simulation

The simulations will allow us to gain an understanding of how the controller will respond to specific inputs. The simulation can then be tuned until the output is within the constraints set by Dr. Rhinelander. We will be simulating both the flight dynamics and controller using MATLAB and Simulink exclusively.

2.2.2 Construction of the Drone

The drone parts will arrive separately and assembly will be required. The extent of the assembly will be to attach the 4 brushless DC motors and batteries to the base of the drone. On top of the assembly the preliminary layout of the required hardware will be decided on. The layout is subject to change as we begin the final assembly in the second semester.

2.2.3 Initial Design of the Controller

A preliminary design of the flight controller will be constructed in software.

2.2.4 Initial Testing

The initial design of the controller will be tested using the brushless direct current (BLDC) motors supplied by Dr. Rhinelander. The initial tests will allow us to gain insight on what changes to the flight controller must be main in order to meet the constraints. Along with the testing of the controller tests will be conducted to characterize the BLDC motor and electronic speed controller (ESC)

2.3 Long Term Objectives (December 2016 - April 2017)

2.3.1 Graphical User Interface Design

The graphical user interface (GUI) will be installed on any base station intended to be able to operate the drone. The key features of the GUI include: A means to access the controller, displays drones position (Coordinates and altitude) and the ability the load a new build onto the drone. Some minor features will include: displays the current software build on the drone, current flight time and total flight time.

2.3.2 Base Station Configuration

This will entail installing the GUI onto the base station and configuring the base station network adapter to be able to communicate with the raspberry-pi on the drone.

2.3.3 Network Tests

The intent of the network test is to gain an understanding of the network strength at various distances. A preliminary idea of how these tests will be conducted is to ping the raspberry-pi from the base station to see the time of response at these distances.

2.3.4 Final Controller Tests

The final controller tests will be identical to the tests run during the first semester of the project. The purpose of these tests will be to verify that the appropriate changes were made in order to meet the defined constraints so that flight tests of the drone may be completed.

2.3.5 Flight Tests

The flight tests will be run to validate that each constraint has been met to the best of our ability. The tests include, but are not limited to: looking at the drones response to disturbances, response to control inputs, response to loss of communication and verify that the GUI is reporting the expected data.

2.4 Deliverables

The final deliverables will be as follows:

1. Constructed drone

2. User manual

3. Graphical User Interface

2.4.1 Constructed drone

The constructed drone will consist of the base and BLDC motors supplied by Dr. Rhinelander. The microprocessor and microcontroller used for development and testing are property of Dylan Humber and Lucas Doucette respectively.

2.4.2 User Manual

The user manual will outline technical details of the controller design, instructions on how to use the GUI and characteristics of BLDC motors and ESC's.

2.4.3 GUI

The GUI will be loaded on to a Universal Serial Bus (USB) so that it may be installed on any device that Dr. Rhinelander desires to act as a base station. The GUI will be able to be installed on Linux, Windows and MAC devices.

3 Background and Significance

Machine learning is defined as the science of getting computers to act without being explicity asked. This is achieved by the development of computer programs that can teach themselves to grow and change when exposed to different sets of data. There are two traditional types of machine learning algorithms: Batch learning algorithms and on-line machine learning algorithms. Batch learning algorithms require a set of predefined training data that is shaped over the period of time to train the model that the algorithm is running on. On-line learning uses an initial guess model that forms covariates from that initial guess then passes them through the algorithm to form an evolved model a new set of covariates are formed from the evolved model and then fed back to make a new prediction. The loop runs continuously so that the evolved model is constantly growing and learning to adapt to certain situations. Dr. Rhinelander's research is concerned with on-line machine learning algorithms therefore the drone we are developing will configured to adapt with these algorithms.

Quadrotor drones have been on the rise in popularity in the last several years due to their simplistic mechanical design and many practical uses. The application of these drones vary from a hobbyist flying around their neighbourhood to military personnel carrying out high risk missions. A video recording device of some sort is generally attached to the drone and the video feed is relayed to a basesation for the operator to gain a field of view (FOV) of an area of interest. Having the capability to have a continuous video feed allows the drone to be used for many practical applications including but not limited to: Traffic condition monitoring and surveillance missions. While these drones are very sophisticated and advanced devices they are are missing one aspect that is very important to further Dr. Rhinelanders research: They are not totally configurable.

As mentioned, Dr.Rhinelanders research is concerned with on-line machine learning algorithms and without a platform that is completely configurable his research would be limited. Before the machine learning algorithms are implemented onto the drone it must first be able to be controlled. This is where we come in, we have been tasked by Dr. Rhinelander to develop a flight controller that recieves control inputs over Wi-Fi. ***Having a completely open source flight controller will allow for the addition of the machine learning algorithms to the flight controller software so that the drone can learn to partially, and eventually fully fly on it's own and make intelligent decisions.***

4 Proposed Approach and Validation

5 Preliminary Results

5.1 SimuLink System Models

Using SimuLink, a model of the quadcopter system and control system was built. This was used to perform preliminary hovering tests. The SimuLink model was broken down into several blocks. Each of these blocks was built to handle key aspects of the system. The system can be seen in it's current state in the following figure. Each of the blocks shown are discussed in detail below.

system_arch.jpg

Figure 1: SimuLink Control System Architecture

5.1.1 Unput Model

5.1.2 Controller Model

5.1.3 Motor Characterization

5.1.4 Multibody Base

5.2 Proof of Concept and Testing Performed

To date, several tests and proof of concepts have been performed to determine methods of controlling hardware, limitations of the hardware or software and to determine feasibility of communication protocols. The test subjects include:

- PWM
- Electronic Speed Controller
- Motor Lift Characteristics
- Wi-Fi Range
- I2C Communication Channel
- Playstation Controller Integration

5.2.1 PWM

Using an Arduino micro controller, a potentiometer and a simple DC motor, a circuit was devised to test a PWM output of the Arduino to drive a motor circuit. The motor circuit was isolated using a mosfet as the switching operator and a diode to ensure there would be no damaging back emf in the system. A 9V battery powered the isolated circuit.

Through programming, the voltage across the potentiometer was taken into the Arduino as an analog input and mapped to a digital output as a PWM duty cycle. The script was successful in providing an input controlled PWM value to the motor. The code was modified to suit the needs of the electronic speed controllers by applying an input controlled pulse width as a function of time in replacement of a duty cycle as the ESC's require a pulse width range of 1060 μs and 1860 μs.

5.2.2 Electronic Speed Controller

The Afro ESC 12Amp BEC UltraLite Multirotor ESC V3 was tested at St. Mary's University with the assistance of Dr. Rhinelander. An Arduino running a script to map a potentiometer to a PWM duty cycle was used as an attempt to control the motor through the ESC. A 12V power supply fed the ESC while the Arduino controlled the duty cycle of the speed controller producing a voltage output to control the motor speed.

The ESC testing was successful, the testing proved the Arduino's capability to control the motor with a variable input. The testing had flaws as a PWM duty cycle was used instead of a timed pulse width input. The duty cycle had potential of operating correctly as the range of times could have been calibrated to a range in the duty cycle although this method proved to be difficult due to low values causing the ESC to enter calibration mode. The script used to operate the ESC was re-written as a timed pulse width to ensure complete compatibility and ease of future integration.The pulse width script was tested using the ESC and was successful. The provided motor was successfully driven under no load conditions for the full range of pulse width values.

5.2.3 Motor Lift Characteristics

Using a 12V power supply, the provided ESC and the Multi-Star Elite motor the characteristics of the motor's lifting capacity were tested. A weight was attached to a support system with the motor and blade seated on top. The apparatus was placed on a scale and the scale's reading was zeroed. As the rotor speed increased, the reduction in weight read by the scale was considered the lift capacity.

The test was performed beginning at a pulse width of 1127μs which was found through experimentation to be the cut in pulse width for motor operation. The test was performed at increments of 25μs. The resultant current draw values and lift values were documented. The lift values were used to create a simulink lookup table to characterize the motor's available force for simulation purposes.

During the load testing, it was noticed that the current draw from the individual motor was high. The power supply being used had a current limit of 3 A therefore the maximum current draw allowed during testing was 2.95 A to ensure no brown out due to lack of supply. The current limiting factor resulted in the test ending at a pulse width of 1525μs and a lift value of 137.5g.

5.2.4 Wi-Fi and Bluetooth Range

A simple proof of concept regarding the range of Wi-Fi communications was performed. The test incorporated a a Wi-Fi communicating camera tethered to a Wi-Fi output from a cell phone. A user walked down Spring Garden holding the cell phone and found the approximate distance at which the phone and the camera lost communication. It was found that the range was approximately 100 ft with line of sight available but no Wi-Fi boosting technology. Bluetooth communications were also tested using the Raspberry Pi 3 connected to a Playstation 4 controller although the communication channel held a strong connection for only approximately 10 ft, this distance was considered insufficient for the scope of the project.

5.2.5 I2C Communication Channel

The flight controller will consist of both an Arduino micro controller and a Raspberry Pi 3 Linux based computer system. The flight controller will require a communication channel between each component to transmit data between each system. To enable these communications, an I2C bus is planned to be utilized.

The I2C bus was enabled on the Raspberry Pi and a sample open source script was run on the Arduino to begin I2C communications. The test resulted in the Raspberry Pi recognizing the Arduino on I2C bus.

5.2.6 Playstation Controller Integration

Using an open source python script, a Playstation 4 controller was integrated with the built in Bluetooth communication channel of the Raspberry Pi 3. All inputs available from the Playstation controller were taken in as values to the Raspberry Pi, confirming the compatibility of systems. Based on the Playstation controller integration testing, the feasibility of the Bluetooth communication was put into question. Although successful, it was decided that based on range constraints, Bluetooth would not be used as the communication channel, a hardwired serial connection to a base station will be utilized.

6 Work Plan and Milestones

7 Distribution of Tasks

8 Budget