AALBORG UNIVERSITY

STUDENT REPORT

ED5-3-E16

XXX

Students:
Alexandra Dorina Török
Andrius Kulšinskas

Supervisors:
Christian Mai

November 22, 2016

AALBORG UNIVERSITY

STUDENT REPORT

School of Information and Communication Technology
Niels Bohrs Vej 8
DK-6700 Esbjerg
http://sict.aau.dk

Title:
xxx

Theme:
Scientific Theme

Project Period:
Autumn Semester 2016

Project Group:
ED5-3-E16

Participant(s):
Alexandra Dorina Török
Andrius Kulšinskas

Supervisor(s):
Christian Mai

Copies: x

Page Numbers: 31

Date of Completion:
November 22, 2016

Abstract:

xxx

Contents

Preface

The project entitled *xxx* was made by two students from the Electronics and Computer Engineering programme at Aalborg University Esbjerg, for the P5 project during the fifth semester.

From hereby on, every mention of 'we' refers to the two co-authors listed below.

Aalborg University, November 22, 2016.

Andrius Kulšinskas
<akulsi14@student.aau.dk>

Alexandra Dorina Török
<atarak14@student.aau.dk>

Chapter 1

Introduction

1.1 Introduction

The theme of this semester's project lies within *Automation*. Automation can be simply described as being the use of diverse control systems for fulfilling a certain task with little to no human interaction. As known from the previous semester, a control system is an instrument which has the role of adapting the behaviour of a system according to a desired state, also knows as steady-steady or reference. Any control system has three components: measurement, control and actuation. Without one of these, automation would not be possible. Essentially, the measure reflects the current state of the system, the controller is the brain that given the measurement decides which action will be performed and the actuator is the one executing the action.

Project ideas around the topic of automation are unlimited, since it is so widely spread. Having discussed a few of them that would meet the semester's requirements, we finally decided to work on the control of a quadcopter. Our decision was, for the most part, based on the fact that the university had the required equipment available, which enabled us to start working on the project right away.

UAVs have been attracting attention for many decades now. Powered UAVs were, at first, utilized by the military to execute reconnaissance missions. Nowadays, they have found other uses, such as aerial photography, search and rescue, delivery, geographic mapping and more. While there are different types of UAVs, our focus is on the multirotors. Multirotor can be defined as a rotorcraft with more than two motors. Based on this, the four most common types are tricopter, quadcopter, hexecopter and octocopter, each having 3, 4, 6 and 8 motors respectively. Each type of multirotor has its ups and downs - more motors mean higher liftforce and more reliable stability, but they also increase both price and damage caused in case of an error. Due to their price, size and ease of setup, quadcopters are the most popular type. They are popularly referred to as drones. Our goal for the project is to design a control system that makes it possible for the quadcopter to be stable - hovering mid-air - and to also act according to the user's input - maneuvering. Basically, our

input will be a certain height and the quadcopter will have to automatically adjust to that height and maintain its stability when no further inputs are given. A safety feature - obstacle avoidance - will also be implemented.

Chapter 2

Problem Description

Quadcopter control is a complex, yet interesting problem. One of the reasons why this control problem is challenging is the fact that a quadcopter has six degrees of freedom, but only four inputs which affects the linearity of the dynamics and makes the quadcopter underactuated. One other important thing to mention is that quadcopters, unlike ground vehicles, have very little friction that prevents their motion, so they have to provide their own damping in order to be able to stop or maintain stability.

This chapter will give a general overview on the pieces of technical equipment which we are using and show what the purpose of each one is. It will also present how our scopes - hovering and maneuvering - can be achieved by designing and implementing a control system. A solution in terms of the structure of our control system will be identified. We will also delve into the basic working principle of a quadcopter/flight dynamics.

2.1 Physical Setup

2.1.1 Motors

Controlling a quadcopter can be done efficiently by using high-quality motors with fast response, which will ensure more of a stable flight. The motors must also be powerful enough to be able to lift the quadcopter and perform the required aerial movements.

The motor that we are using is the Turnigy Multistar Brushless Motor seen in Figure 2.1.

Figure 2.1: Turnigy Multistar 2213-980 V2 Brushless Motor

2.1.2 Propellers

The propellers don't have such strict requirements as the motors. They are needed to be light and have a size and lift potential in order for the quadcopter to hover at less than 50% of the motor capacity. For our quadcopter, we are using plasting 10x4.5" propellers with light weight - 60g. They have a length of 254 mm and a pitch inclination of 114mm. They can be seen in Figure 2.2.

Figure 2.2: Hobbyking Slowfly Propeller 10x4.5

2.1.3 Electric Speed Controller

Electronic Speed Controller (ESC) is a widely used device in rotorcrafts. The purpose of an ESC is to vary the electric motor's speed. They also come with programmable features, such as braking or selecting appropriate type of battery. We need the ESC to have a fast response, for the same reasons mentioned for the motors in Section 2.1.1. The ESC that we are using is the TURNIGY Plush 30A which is shown in Figure 2.3.

Figure 2.3: TURNIGY Plush 30A Speed Controller

2.1.4 APM Flight Controller

ArduPilotMega (APM) is an open source unmanned aerial vehicle (UAV) platform which is able to control autonomous multicopters. It is illustrated in Figure 2.4. The system was improved uses Inertial Measurement Unit (IMU) - a combination of accelerometers, gyroscopes and magnetometers. The "Ardu" part of the project name shows that the programming can be done using Arduino open-source language.

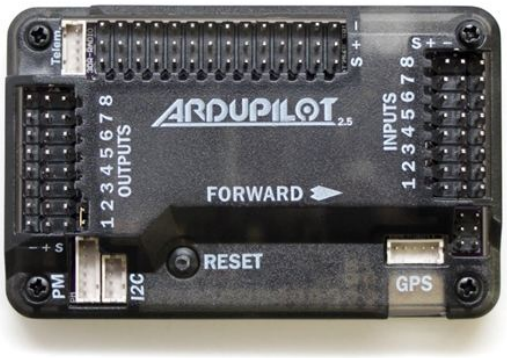

Figure 2.4: APM 2.5 board

2.1.5 Power Distribution Board

To reduce the number of connections straight to the battery, we used the a power distribution board made for a previous project. A board like this is an easy solution since it enables us to connect the four ESCs directly to the board and then connect the board to the battery.

2.1.6 Battery

To power up our quadcopter, we will use a TURNIGY nano-tech Lipoly battery, which can be seen in Figure 2.5. Higher voltage under load, straighter discharge curves and excellent performance are the factors that make it suitable for our project.

Figure 2.5: Turnigy nano-tech 6000mah 3S 25 50C Lipo Pack

2.2 Flight Dynamics

In order to put a working quadcopter together, it is important to understand the basic flight dynamics - the forces affecting the vehicle as well as the motors' speed effect. There are 4 main forces affecting quadcopter - or any airborne vehicle, for that matter - thrust, lift, draw and drop. See figure 2.6.

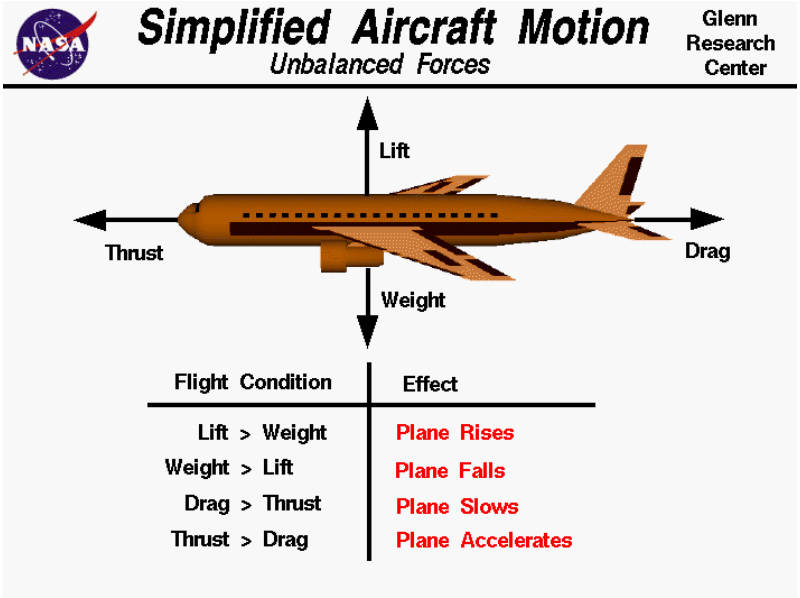

Figure 2.6: Forces affecting an airborne vehicle [1]

The drop - or the gravitational force - affects the vehicle at all times. As any object on Earth, its mass is driven towards the centre of the planet. This force, F_g, is always equal to: $F_g = mg$, where m is the mass of the drone and g is the gravitational constant.

The thrust is the force generated by the motors that is allowing the vehicle to move towards its heading. In case of a quadcopter, this force only exists when the force generated by the motors is uneven.

The lift force, following Bernoulli's principle, is created through a difference in the air pressure above and below the motor. Quadcopter's motors constantly generate lift force, which must be higher than the drop force in order for the vehicle to take flight.

Draw force is the resistance created by the air as the vehicle moves through it. It opposes the thrust force and therefore must be lower than the thrust force in order for the quadcopter to move on. In cases when is no wind, such as indoors area, this force can be disregarded due to lack of wind.

A powered off motors is only affected by the drop force and therefore stays on the ground. In order to lift it up, we need to understand the relationship between quadcopter and the thrust to weight ratio - or TWR for short. This ratio can be

determined by equation F_t/F_g and describes the vehicle's ability to move up. With TWR expressed as a number, assuming that each motor generates equal amount of thrust, three cases can be identified:

1. TWR < 1: The gravitational force is higher than the lift force and therefore the quadcopter is drawn towards the ground.
2. TWR $= 1$: The forces are equal, causing quadcopter's altitude to stay constant.
3. TWR > 1: The thrust is higher than drop force, allowing vehicle to move upwards.

Therefore, in order to get a quadcopter up in the air, it is necessary to generate enough thrust for TWR ration to be higher than one. In order to land it, the TWR must be smaller than 1, allowing the quadcopter to move downwards.

Moving on, it is necessary to understand the effects that motor speeds have on the vehicle. While equal amount of force generated by every motor will result in a movement on vertical axis, differences in motor speeds will create a change in quadcopter's pitch, roll or yaw. These three motions can be seen in figure 2.7.

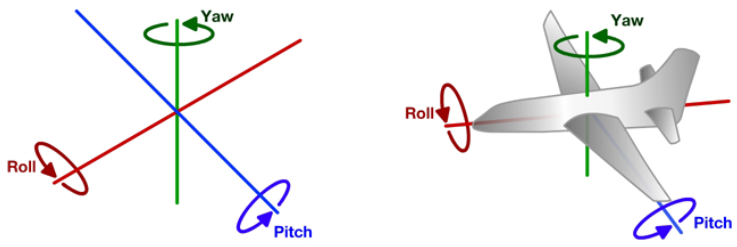

Figure 2.7: Pitch, roll and yaw motions in a 3-dimensional plane [2]

To examine the consequences of different motor speeds, let's assume there's a simple quadcopter with all 4 motors placed at equal distances from the centre. An example can be seen in figure 2.8.

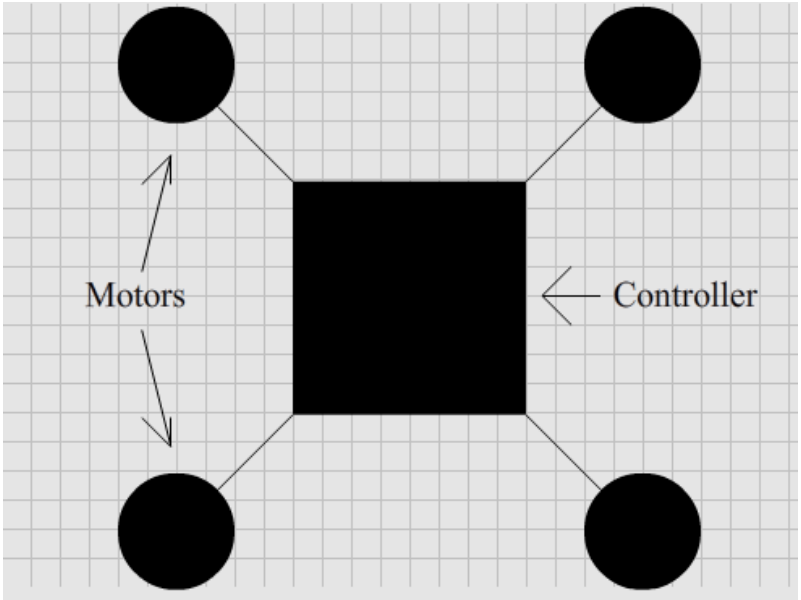

Figure 2.8: A quadcopter viewed from top

The direction of rotation and produced torque as well as the amount of force generated by the motors affect the quadcopter's movement. See figure 2.9 for an example in an idle - or hovering - position.

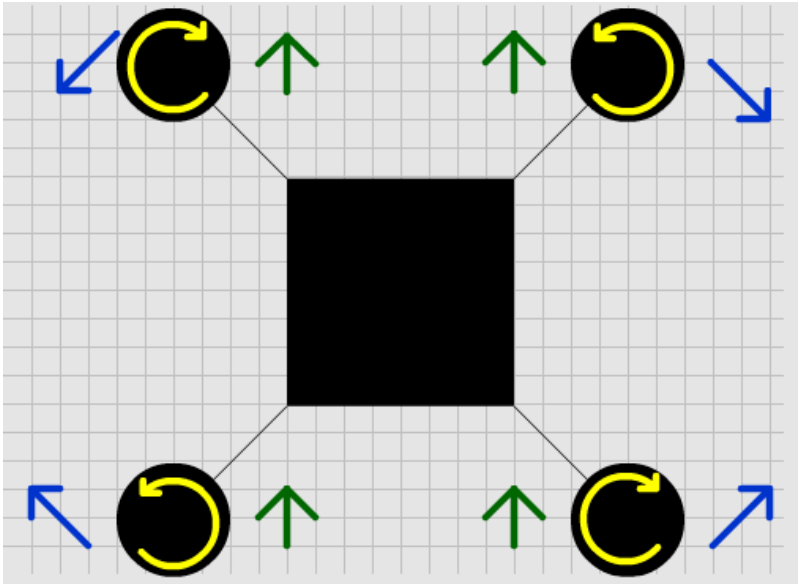

Figure 2.9: Motor direction overview

Here, the yellow arrows indicate the direction of rotation, the blue arrows show the direction of torque and the green arrows display the amount of force generated by the motors.

By controlling the speed at which the motors rotate, it is possible to change how quadcopter moves. Movement can be broken down in 3 separate sets of motor speeds:

1. If the two back motors rotate faster than the two frontal motors, the quadcopter will pitch forward. Switching the speeds will result in aft pitch. The required change in speeds for a pitch forward can be seen in figure 2.10.

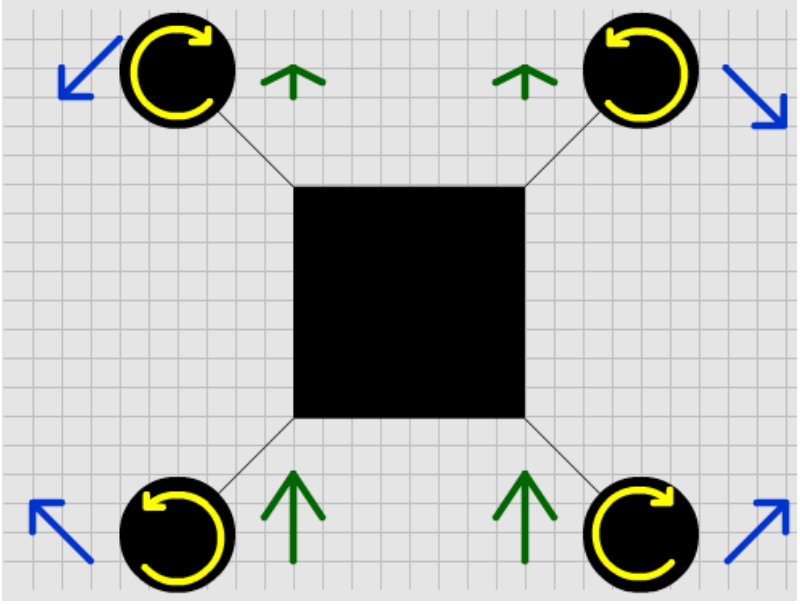

Figure 2.10: Motor RPM requirements for a pitch forward

2. If the two left motors have higher RPM than the two motors on the right side, the vehicle will roll to the right. Swapping the speeds differences will result in a roll to the left. See figure 2.11 for an example of a roll to the right.

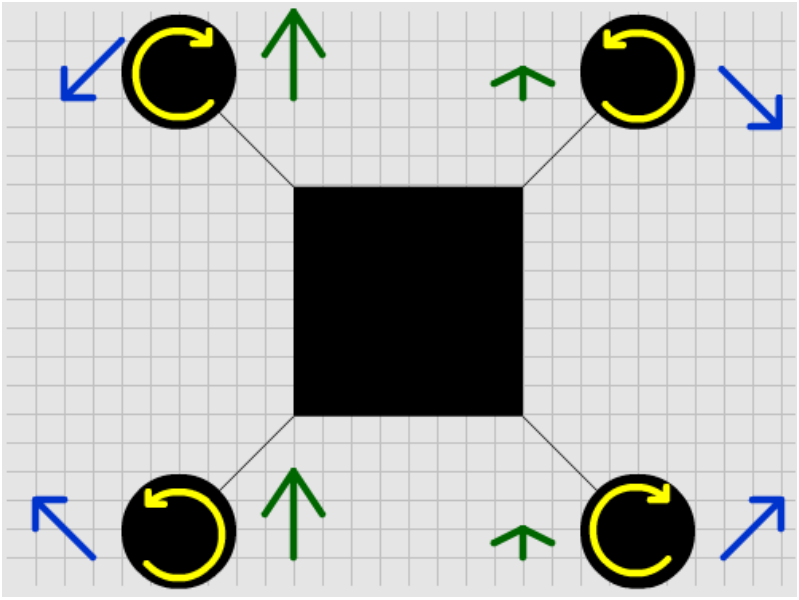

Figure 2.11: RPM requirements for a roll to the right

3. Increasing the speed of one of the diagonal pair of the motors will result in yaw to the direction of the torque of the motors. The copter will then spin

around its axis. Motor speed changes causing a yaw to the right can be seen in figure 2.12.

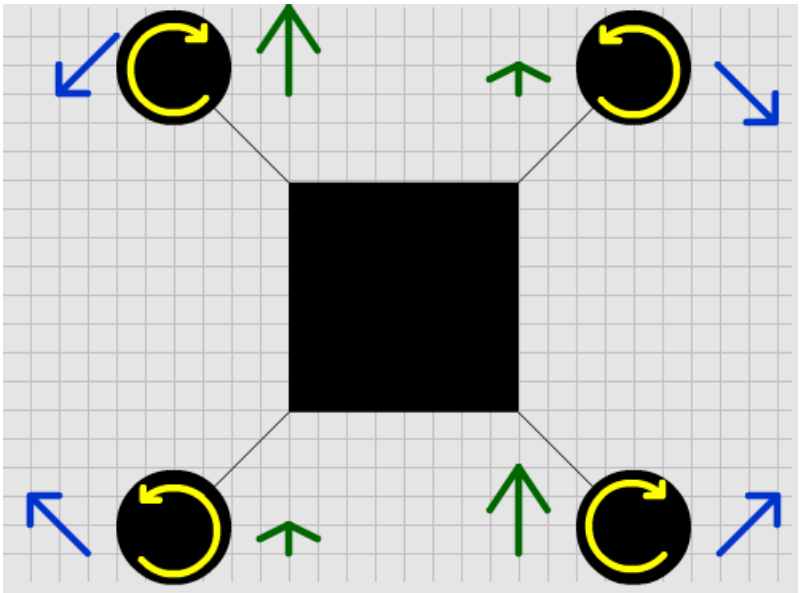

Figure 2.12: The two diagonal motors with increased RPM cause yaw to the right

Chapter 3

Mathematical Modelling

In this chapter we will take a look at mathematical models that describe our quadcopter system.

3.1 BLDC Motor

The mathematical model of a BLDC motor is is many ways similar to the one of a conventional DC motor. The main difference is represented by the added phases which affect the resistive and inductive components of the BLDC arrangement.

Therefore, we will start of by describing the mathematical modelling of a DC motor and then change it to fit the BLDC motor.

Figure illustrates a DC electromechanical system.

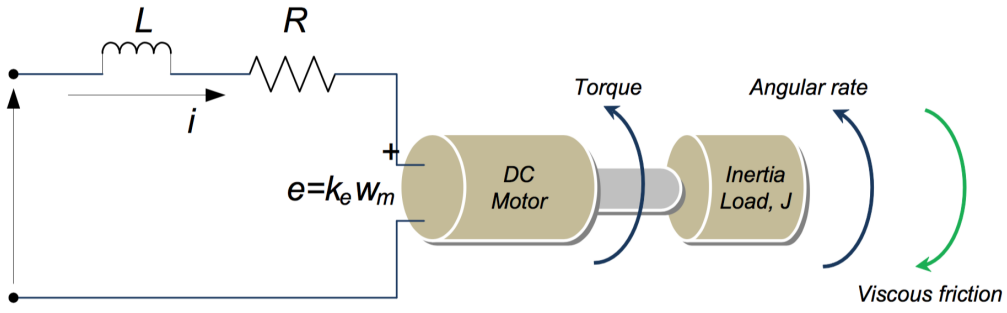

Figure 3.1: Typical DC electromechanical system

The components of the electrical circuit are the armature resistance - R, the armature inductance - L and the back EMF - e. By applying KVL, we obtain:

$$V_s = Ri + L\frac{di}{dt} + e \tag{3.1}$$

where V_s - DC source voltage and i - armature current.

Moving on to the mechanical part, further equations can be written using Newton's second law of motion:

$$J\frac{d\omega_m}{dt} = \Sigma T_i \tag{3.2}$$

$$T_e = k_f + J\frac{d\omega_m}{dt} + T_L \tag{3.3}$$

where T_e - electrical torque, k_f - friction constant, J - rotor inertia, ω_m - angular velocity and T_L - load torque.

The back EMF and the electrical torque can be described as:

$$T_e = k_t\omega_m \tag{3.4}$$

$$e = k_e\omega_m \tag{3.5}$$

where k_e - back EMF constant and k_t - torque constant.

Rewriting equations 3.1 and 3.2 gives:

$$\frac{di}{dt} = -i\frac{R}{L} - \frac{k_e}{L}\omega_m + \frac{1}{L}V_s \tag{3.6}$$

$$\frac{d\omega_m}{dt} = i\frac{k_t}{J} - \frac{k_f}{J}\omega_m + \frac{1}{J}T_L \tag{3.7}$$

Taking the Laplace transform of 3.6 and 3.7 yields:

$$si = i\frac{R}{L} - \frac{k_e}{L}\omega_m + \frac{1}{L}V_s \tag{3.8}$$

$$s\omega_m = i\frac{k_t}{J} - \frac{k_f}{J}\omega_m + \frac{1}{J}T_L \tag{3.9}$$

The next equation is obtained by substituting i from equation 3.9 into 3.8.

$$(\frac{s\omega_m + \frac{k_f}{J}\omega_m - \frac{1}{J}T_L}{\frac{k_t}{J}})(s + \frac{R}{L}) = -\frac{k_e}{L}\omega_m + \frac{1}{L}V_s \tag{3.10}$$

Assuming there is no load, equation 3.10 can be rewritten as:

$$\{(\frac{s^2J}{k_t} + \frac{sk_f}{k_t} + \frac{sRJ}{k_tL} + \frac{k_fR}{k_tL}) + \frac{k_e}{L}\}\omega_m = \frac{1}{L}V_s \tag{3.11}$$

Solving 3.11 gives:

$$V_s = \frac{s^2JL + sk_fL + sRJ + k_fR + k_ek_t}{k_t}\omega_m \tag{3.12}$$

The transfer function can be obtained as the ratio between the angular velocity ω_m and the source voltage V_s:

$$G_s = \frac{\omega_m}{V_s} = \frac{k_t}{s^2JL + (RJ + k_fL)s + k_fR + k_ek_t} \tag{3.13}$$

Considering that k_f tends to zero, the transsfer function can finally be written as:

$$G_s = \frac{\omega_m}{V_s} = \frac{k_t}{s^2JL + RJs + k_ek_t} \tag{3.14}$$

In order to obtain the mechanical and electrical time constants, we will need to manipulate equation 3.14, which gives:

$$G_s = \frac{\frac{1}{k_e}}{\frac{RJ}{k_ek_t}\frac{L}{R}s^2 + \frac{RJ}{k_ek_t}s + 1} \tag{3.15}$$

where the mechanical time constant is:

$$\tau_m = \frac{RJ}{k_ek_t} \tag{3.16}$$

and the electrical time constant is:

$$\tau_e = \frac{L}{R} \tag{3.17}$$

Therefore, substituting equations 3.16 and 3.17 into equation 3.15 yields:

$$G_s = \frac{\frac{1}{k_e}}{\tau_m\tau_es^2 + \tau_ms + 1} \tag{3.18}$$

Equations 3.15 - 3.17 will now be changed in order to fit a BLDC motor. Therefore, they will become:

$$\tau_m = \Sigma\frac{RJ}{k_ek_t} = \frac{J\Sigma R}{k_ek_t} \tag{3.19}$$

$$\tau_e = \Sigma \frac{L}{R} = \frac{L}{\Sigma R} \tag{3.20}$$

Having a symmetrical arrangement and a three phase motor, the constants will finally be:

$$\tau_m = \frac{J3R}{k_e k_t} \tag{3.21}$$

$$\tau_e = \frac{L}{3R} \tag{3.22}$$

Taking into account the phase effects, τ_m will be rewritten as:

$$\tau_m = \frac{J3R_\phi}{k_e k_t} \tag{3.23}$$

with k_e now being the phase value of the back EMF voltage time constant described by:

$$k_e = k_{e(L-L)}/\sqrt{3} \tag{3.24}$$

A relationship between k_e - electrical torque and k_t - torque constant is found in equation 3.25by using the electrical power and mechanical power formulas.

$$k_e = k_t \times 0.0605 \tag{3.25}$$

The next step in coming up with a final transfer function for our motor is to find τ_m and τ_e, which are functions of k_e, k_t, R_ϕ, J and L.

A numerical value for k_e can be obtained using equation 3.24:

$$k_e = \frac{11.1}{1.73} = 6.416 \tag{3.26}$$

Since k_t is a function of k_e, its numerical value can also be found using equation 3.25:

$$k_t = 6.416 \times 0.0605 = 0.388 \tag{3.27}$$

For R_ϕ, we have to measure the line-to-line resistance (between any two wires of the motor) and divide the result by 2 in order to get its value.

experiments on J and L / take them from other reports

3.2 Motor Dynamics

Starting from the top, a brushless motor produces torque given by equation 3.28.

$$\tau = K_t(I - I_0) \tag{3.28}$$

Here, τ is the torque, I is the input current, K_t is the constant describing torque proportionality and I_0 is the no-load current.

The voltage on the motor is the sum of a resistive loss and the Back-EMF, giving equation 3.29.

$$V = IR_m + K_v\omega \tag{3.29}$$

where V is the voltage across the motor, R_m is the motor resistance, ω is the angular velocity and K_v is a constant, describing the back-EMF generated per RPM. We can then combine equations 3.28 and 3.29 and put them into the power equation, to get a new equation 3.30.

$$P = IV = \frac{(\tau + K_tI_0)(K_tI_0R_m + \tau R_m + K_tK_v\omega)}{K_t^2} \tag{3.30}$$

To help simplify the model, we can assume that the motor resistance is very small and therefore can be disregarded. This gives us a new power equation 3.31.

$$P \approx \frac{(\tau + K_tI_0)K_v\omega}{K_t} \tag{3.31}$$

We can also, for the purposes of simplifying the model, assume that $K_tI_0 \ll \tau$. This is a reasonable assumption due to the fact that I_0 describes the current with no load on the motor and is usually a small number. Based on this, we can then obtain the final equation 3.32 for power.

$$P \approx \frac{K_v}{K_t}\tau\omega \tag{3.32}$$

From the conservation of energy, it is known that the energy that the motor expends during some period of time is equal to the force generated by the propeller multiplied by the distance the it displaces moves, given by the equation 3.33.

$$P \times dt = F \times dx \tag{3.33}$$

We can then say that the power is also equal to the thrust multiplied by the air velocity, as seen in equation 3.34.

$$P = F\frac{dt}{dx} = Tv_h \tag{3.34}$$

Here - T is the thrust and v_h is the air velocity. Since the model is describing a hovering quadcopter in a closed area, only v_h is an acting velocity (i.e. vehicle speed has no effect and free stream velocity is equal to zero).

Momentum theory gives equation 3.35, for the hover velocity v_h.

$$v_h = \sqrt{\frac{T}{2\rho A}} \tag{3.35}$$

where A is the area swept out by the blade and ρ is the air density. In the case of this model, τ is proportional to thrust T by a constant K_τ. Keeping this in mind, we can now use the simplified power equation 3.32 and derive a new equation 3.36.

$$P = \frac{K_v}{K_t}\tau\omega = \frac{K_v K_\tau}{K_t} T\omega = \frac{F^{\frac{3}{2}}}{\sqrt{2\rho A}} \tag{3.36}$$

Solving for thrust T, we can derive a new equation 3.37.

$$T = (\frac{K_v K_\tau \sqrt{2\rho A}}{K_t}\omega)^2 = k\omega^2 \tag{3.37}$$

where k is a constant, describing all other constants used in the equation.

The final equation 3.37 now describes the thrust T generated by the motor with angular velocity ω as an input.

Chapter 4

Experiments

4.1 Initial Experiments

4.1.1 Calibrating and Programming ESCs

ESCs have 5 input pins, 3 of which come from the flight controller. These 3 wires supply the signal for the ESC to translate into the angle of the shaft for the motor. Before ESCs can be used, they need to be properly calibrated. Programming additional settings is optional, but in most cases necessary too. Normally, the calibration for UAVs is done by setting the throttle to full on the radio controller before powering the ESCs. However, since the board translates the throttle signal into a PWM signal, it is possible to calibrate and run the motors directly from the flight controller by sending a PWM signal using software.

The calibration is done by first sending the maximum length of signal the user wishes to use. The ESC emits sounds that are brand-specific and indicate whether the signal was successfully recognised or not. Once the maximum signal is accepted, the user the emits the minimum signal and waits for approval. Additional sound is then emitted, indicating that the calibration was successful.

Using Arduino's Servo library, a self-explanatory built-in function *servo.writeMicroseconds(int value)* is used to send the signal. By default, the library has the minimum signal set to $544\mu s$ and the maximum - to $2400\mu s$. For this project, we shortened the range down to $700\mu s$ and $2000\mu s$ for both signals respectively. A commented code used to calibrate the ESC connected to the first pin of the board can be seen in listing 4.1.

```
#define MAX_SIGNAL 2000 //define max and min signal length
#define MIN_SIGNAL 700
#define MOTOR_PIN1 12 //pin 1 on the board
Servo motor1;
void setup() {
  Serial.begin(9600); //begin serial communication at 9600 rate
  Serial.println("Program begin...");
  motor1.attach(MOTOR_PIN1); //attach the pin to the servo variable
```

```
9     Serial.println("Now writing maximum output.");
      Serial.println("Turn on power source, then wait 2 seconds and press
        any key.");
11    motor1.writeMicroseconds(MAX_SIGNAL); //this signal should only be
        sent when calibrating for the first time
      while (!Serial.available()); //wait for an input, so you have time to
         plug in the ESC
13    Serial.read();
      Serial.println("Sending minimum output");
15    motor1.writeMicroseconds(MIN_SIGNAL); //send the min signal, which is
         the only needed signal when already calibrated
      while (!Serial.available()); //wait for the beeps to indicate
        calibration is done
17    Serial.read();
      Serial.println("Calibrated");
19  }

21  void loop() {

23  }
```

Code Listing 4.1: Calibrating the ESC

Programming additional settings of the ESC is done in a similar manner - by sending minimum and maximum signals after the ESC emits particular sounds, indicating wanted selections. The programmable features and selections are brand-specific and can be found in the datasheets. For the purposes of this project, the ESCs have been programmed to include two features - brake mode off and selection of li-ion battery.

4.1.2 Expected and Real Motor Performance

The motors used in the prototype specify to be rated at K_v of 980. K_v is a constant describing the ration between RPM and the applied voltage and is expressed as $K_v = \frac{RPM}{V}$. Derived from this, voltage's effect on the RPM can be seen in figure 4.1.

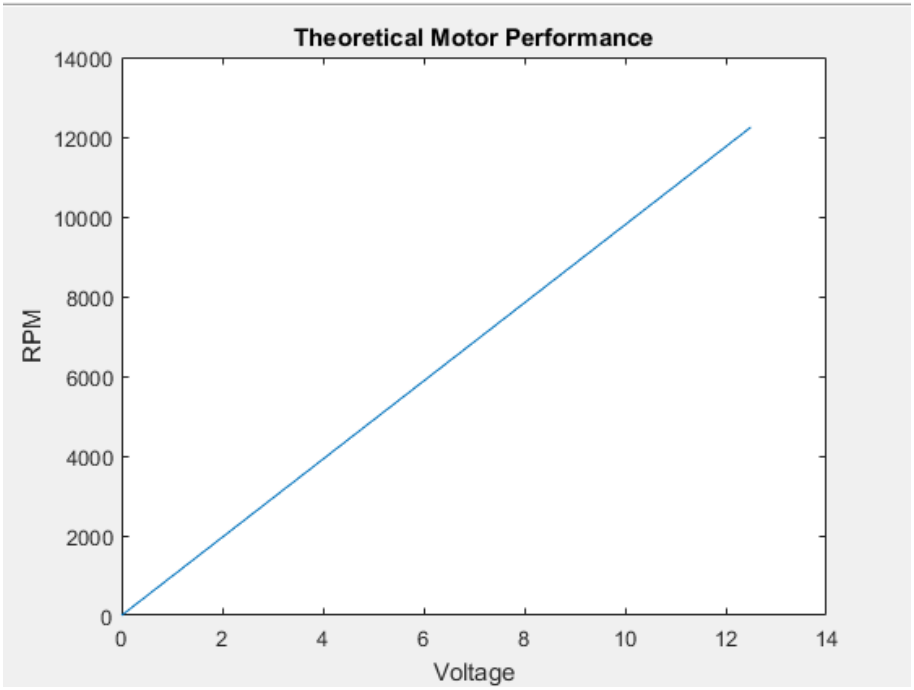

Figure 4.1: Expected motor performance

With a fully charged battery, the RPM is expected to be $980 \times 11.1V = 10878RPM$.

In order to confirm this, the actual RPM was measured using SHIMPO DT-205 digital tachometer. A piece of reflective paper was taped to one of the motors so that the tachometer would have something to lock onto, as seen in figure 4.2.

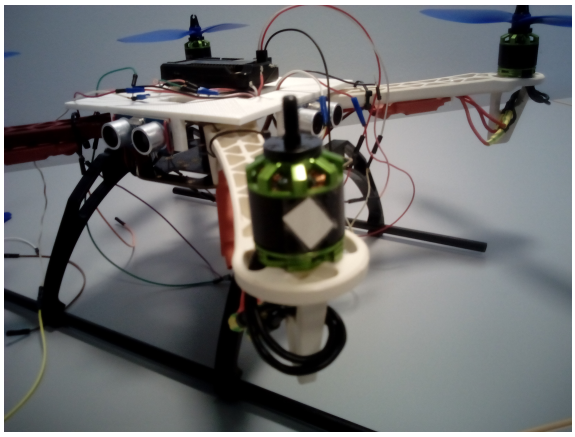

Figure 4.2: Reflective paper on the motor

By sending the maximum signal from the board, the RPM was measured to be 10812, with a small possible error. In conclusion, the motors' K_v coefficient found in the datasheet is in fact correct.

4.1.3 Output Operating Range

Since there is no direct translation between the output signal from the board and the speed of the motor and the ESC allows more current than the motor can recognise, it is necessary to find the operating range for the board output that has the most noticeable effect on the RPM of the motor. Through trial and error and by measuring the RPM using SHIMPO DT-205 digital tachometer. The lowest operating point was found to be $762\mu s$, at which the motor finally begins to spin. The highest point was defined at $1200\mu s$. Higher values were found to have very small effect on the RPM. Therefore, despite the ESC recognising the values between 700 and $2000\mu s$, the most noticeable effect will be between 762 and $1200\mu s$ and will be primarily used in the project.

4.1.4 APM Frequency

Due to lack of proper documentation, it was necessary to do measure the frequency of the signals sent out by the flight controller. To do so, an oscilloscope was connected to one of the output pins of the board. Then, using the servo library, a signal was sent out. The interval between signals was found to be $20ms$, therefore, the frequency of the board is $50Hz$, as seen in figure 4.3.

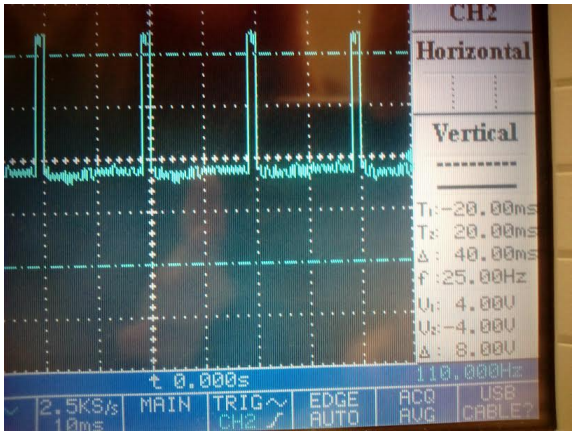

Figure 4.3: Oscilloscope measuring APM's frequency

The second experiment on the board was then made to determine how the flight controller handles the output signals during those $20ms$. First two outputs of the APM were connected to the oscilloscope, both utilizing the Servo library to send signals of length of $2000\mu s$. Results can be seen in figure 4.4.

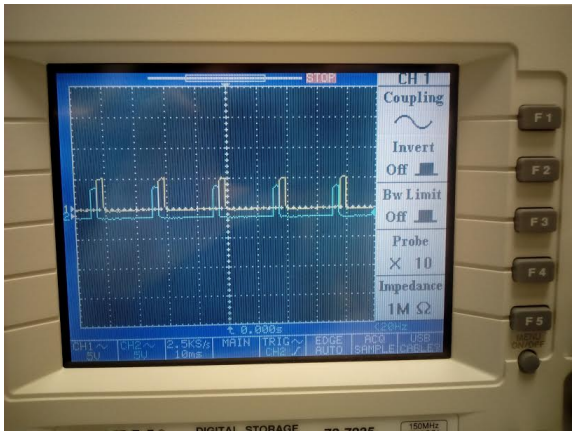

Figure 4.4: Readings of the two output signals

Then, for further testing purposes, both outputs were given different values in two scenarios, as seen in figure 4.5.

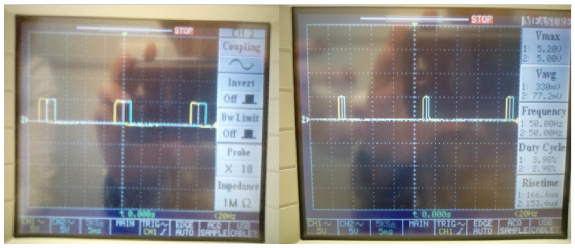

Figure 4.5: Left - signals running at $2000\mu s$; Right - $700\mu s$

From this, two conclusions can be made:

1. If the first signal is shorter than the second one, the second signal will still follow right after the first signal ends. In other words, the APM leaves no gaps between the outputs.
2. Since the board runs at the frequency of $50Hz$ and has a period of $20ms$, this leaves $\frac{20}{8} = 2.5ms$ maximum length for each output signal. The servo library is hard-capped at $2.4ms$ and thus is well within the limits of the board.

4.1.5 APM Output and ESC Output Relationship

In order to use the board's output signal as an input in a mathematical model, it is necessary to find an equation to translate it into an angular rate. First, an equation has been found that describes the relationship between motor's RPM, ESC's output signal frequency and number of motor poles [3]:

$$RPM = \frac{120 \times f}{n} \tag{4.1}$$

Here, f is the frequency and n is the number of poles (in the case of this project - 14).

Therefore, it is possible to measure the frequency at various output values, which can then be used to define an equation that uses the board's signal length as an input value and results in an RPM.

The frequency was measured using an (NAME) oscilloscope and by connecting the probe's ground clip to the ground of the battery and the probe tip to any one of the wires between motor and the ESC. Then, by providing different output signals from the flight controller, the frequency was measured on the oscilloscope screen. An on-board filter was used to filter out some noise, especially at lower output signals. The recorded frequency was later converted into RPM using equation 4.1, for later comparison. The output signal lengths and the corresponding frequencies converted into RPM can be seen in table 4.1.

$Output, \mu s$	RPM
762	2825
770	3189
780	3689
790	4211
800	4683
810	5161
820	5546
830	5874
840	6223
850	6532
860	6795
870	7008
880	7301
890	7534
900	7730
925	8233
950	8552
975	8786
1000	9129
1050	9566
1100	9814
1150	10071
1200	10260

Table 4.1: Frequency measured at ESC output, converted into RPM

While measuring, it was observed that the frequency would never reach a steady-state and therefore, a sizeable error is possible between measurements. Because of that, RPM was manually measured using the tachometer at same output values, as seen in table 4.2.

$Output, \mu s$	RPM
762	2754
770	3170
780	3664
790	4210
800	4715
810	5223
820	5617
830	5965
840	6285
850	6578
860	6846
870	7100
880	7368
890	7555
900	7790
925	8265
950	8614
975	8910
1000	9160
1050	9576
1100	9860
1150	10089
1200	10261

Table 4.2: RPM measured at different values

While measuring with tachometer, the error appeared to be less significant.

The two tables 4.2 and 4.1 were then plotted to see the difference, which is seen in figure 4.6.

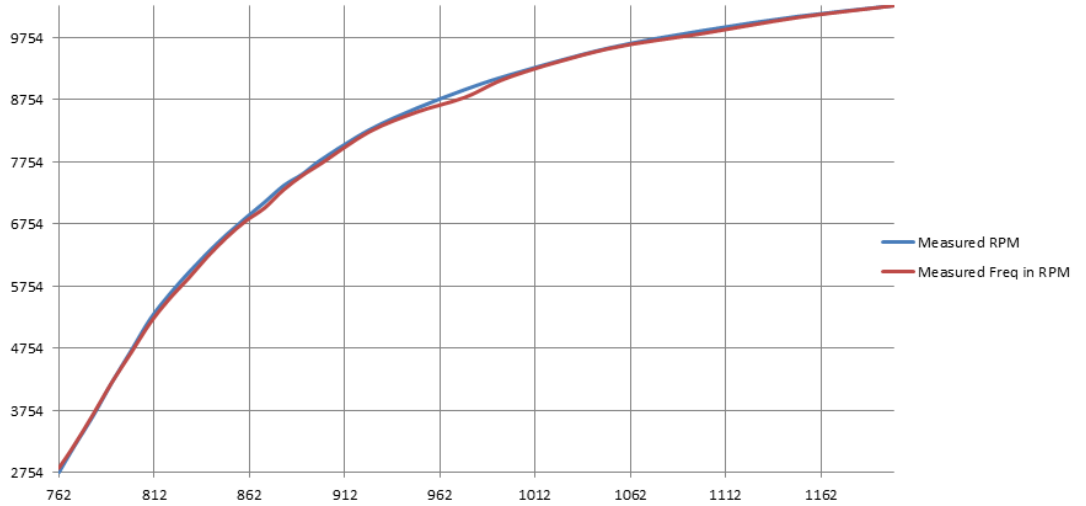

Figure 4.6: Plotted values of manually measured and frequency-based RPM

From the plotted graphs, it can be seen that the difference is very much noticeable, especially at certain points. Due to the fact that measurements with tachometer provided smaller errors than measurements with an oscilloscope, it was decided to use the values measured with the tachometer for later calculations.

In order to make use of these findings, the RPM values have to be converted into rad/s for use as angular rate. The conversion equation is:

$$\omega = \frac{2\pi \times RPM}{60} \tag{4.2}$$

The values measured with tachometer were then converted into rad/s and plotted on a Microsoft Excel 2010. Then, by utilising the "Format Trendline" function, a 4^{th} order polynomial equation fitting the original graph was obtained. The values and graphs can be seen in figure 4.7.

Input signal	rad/s
762	288.3982055995
770	331.9616237293
780	383.6931827584
790	440.8701690538
800	493.7536453892
810	546.9512809900
820	588.2108645071
830	624.6533392888
840	658.1636609271
850	688.8465491771
860	716.9114435492
870	743.5102613496
880	771.5751557217
890	791.1577499290
900	815.7668923822
925	865.5087760640
950	902.0559706007
975	933.0530181162
1000	959.2329568961
1050	1002.7963750259
1100	1032.5367854799
1150	1056.5176094023
1200	1074.5294072828

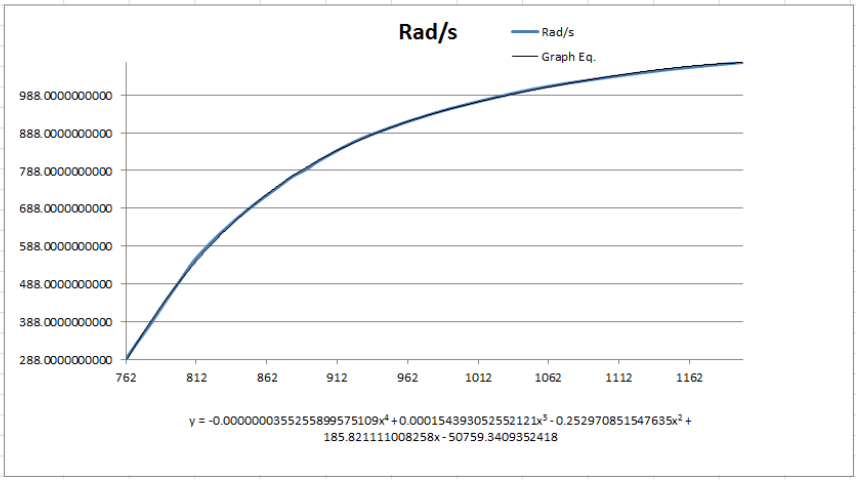

Figure 4.7: Converted and plotted RPM values and the trendline

This equation, $y = -0.000000035525589957510 9x^4 + 0.000154393052552121x^3 - 0.252970851547635x^2 + 185.821111008258x - 50759.3409352418$, can then be used to approximately convert the signal sent out by the flight controller in μs into angular rate.

Chapter 5

Discussion

Chapter 6

Conclusion

Bibliography

[1] Nancy Hall. Simplified airplane motion. https://www.grc.nasa.gov/www/k-12/airplane/smotion.html. (Accessed 06/11/16).

[2] Alex. The quadcopter : control the orientation. http://theboredengineers.com/2012/05/the-quadcopter-basics/. (Accessed 06/11/16).

[3] Vfds how do i calculate rpm for three phase induction motors? http://www.precision-elec.com/faq-vfd-how-do-i-calculate-rpm-for-three-phase-induction-motors/. (Accessed 19/11/16).

Chapter 7

Appendix

7.1 Appendix code