

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Maksim Gorkunov 142809IASB

MATLABIL PÕHINEV DROONI JUHTIMISE SÜSTEEM

Bakalaureuse töö

Juhendaja: Eduard Petlenkov, PhD
TTÜ, professor

Author's declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Maksim Gorkunov

20.05.2019

Abstract

In this thesis, a 4 rotor copter, or quadcopter, was designed based on a research of quadcopter's operation principles and how each separate component should work, for purpose of creating wireless interface with Matlab environment and using it to tune PID controllers of the drone to make it more stable.

Components which were used for the quadcopter throughout this work: jmt X4M360L frame, 4 *JMT F15844 920KV* motors, 4 30A SimonK ESCs, 2 clockwise 8.4*4.3 inches carbon fiber propellers, 2 counter-clockwise 8.4*4.3 inches carbon fiber propellers, *Turnigy 2200 mAh 3S 20C* and *HRB 2600 mAh 3S 30C* batteries, a power board, RadioLink R8EF radio transmitter and receiver, Radiolink T8FB remote controller, APM 2.8 microcontroller, M7N GPS module, APM ESP8266 wi-fi module and APM voltage sensor. Each of the components were chosen to be compatible with one another and to make the quadcopter suitable for aims of this work.

Mission planner software was used to modify PID controller values. Matlab software was used to wirelessly interface with the quadcopter to read and visualize data from sensors in real time.

Throughout this work, 2 separate tests were conducted. In the first part of testing all PID values were set to their maximum. In the second part of drone testing, results from first test were used to properly tune PID values in a series of experiments.

This thesis is written in English and is 47 pages long, including 8 chapters, 28 figures and 0 tables.

Annotatsioon

MATLABi põhinev drooni juhtimise süsteem

Antud töö raames kavandati neljarootoriline kopter ehk nelikopter, mis tugines üldiste kopterite ja iga eraldi komponendi tööpõhimõtete uuringutele. Eesmärgiks oli luua nelikopter, mis on võimeline MATLABi keskkonnaga juhtmevabalt ühenduma (*wireless connection*), et häälestada PID-regulaatoreid ning kontrollida ja muuta nelikopteri liikumist.

Kopteri kavandamisel kasutati järgnevaid komponente: JMT X4M360L raam, 4 JMT F15844 920KV mootorit, 4 30A SimonK ESC-d, 2 päripäeva 8,4 x 4,3 tolli süsinikkiust propellerit, 2 vastupäeva 8.4 x 4.3 tolli süsinikkiust propellerit, Turnigy 2200 mAh 3S 20C ja HRB 2600 mAh 3S 30C patareid, toiteplokk, RadioLink R8EF raadiosaatja ja vastuvõtja, Radiolink T8FB kaugjuhtimispult, APM 2.8 mikrokontroller, M7N GPS-moodul, APM ESP8266 wifi-moodul ja APM pingandur. Iga komponent valiti üksteisega ühilduma vastavalt töö raames püstitatud eesmärkide täitmiseks.

PID-regulaatori väärtuste muutmiseks kasutati lennujuhtimistarkvara (*mission planning software*). MATLABi tarkvara kasutati nelikopteriga juhtmevaba ühenduse kaudu andurite andmete reaajas lugemiseks ja visualiseerimiseks.

Selle töö käigus viidi läbi kaks testi. Esimeses testis määrati kõik PID väärtused maksimaalseks ning saadud tulemusi kasutati teise testi läbiviimiseks. Kopteri teises testis häälestati PID väärtused nii, et kopteri käitumine oleks võimalikult stabiilne.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 47 leheküljel, 8 peatükki, 28 joonist ja 0 tabelit.

List of abbreviations and terms

ESC	Electronic speed controller
IMU	Inertial measurement unit
RC	Radio control
FPV	First person view
GPS	Global positioning system
DC	Direct current
K_v	Motor velocity constant
RPM	Revolutions per minute
V	Volts
A	Ampere
g	Grams
Kg	Kilograms
BLDC	Brushless direct current motor
EMF	Electromotive force
LiPo	Lithium polymer
BEC	Battery elimination circuit
FHSS	Frequency-hopping spread spectrum
DSSS	Direct-sequence spread spectrum
GHz	Giga Hertz
PN	Pseudo Noise
PWM	Pulse width modulation
PID	Proportional Derivative Integral
OSD	On-Screen Display
USB	Universal serial bus
LED	Light emitting diode
I2C	Inter-integrated circuit
UART	Universal asynchronous receiver-transmitter
HAL	Hardware abstraction layer
EKF	Kalman Filtering Library
TCP	Transport control protocol
IP	Internet protocol
MAVLink	Micro Aerial Vehicle Link protocol

Table of Contents

Author's declaration of originality	3
Abstract	4
Annotatsioon MATLABil põhinev drooni juhtimise süsteem	5
List of abbreviations and terms	6
Table of Contents.....	7
List of figures.....	8
1 Introduction	9
2 What is a quadcopter.....	10
3 Components of quadcopter.....	13
3.1 Frame	14
3.2 Motors	15
3.3 ESC	18
3.4 Propellers.....	20
3.5 Power source	21
3.6 Power distribution system	22
3.7 RC (Radio control) module	23
3.8 Flight controller and IMU (Inertial Measurement Unit).....	25
3.9 Miscellaneous: GPS and Wi-Fi modules	27
4 PID controllers.....	28
5 Ardupilot control algorithm code analysis	30
5.1 Code architecture overview	30
5.2 Copter attitude control	31
6 Interfacing with Matlab.....	33
7 Quad-copter PID tuning and results via Matlab	34
8 Summary	38
Kokkuvõtte	39
References	40
Appendix 1	43
Appendix 2	45
Appendix 3	46

List of figures

Figure 1. A quadcopter with X shaped rotor setup [3].....	10
Figure 2. Lateral sideways movement. A-motors, B- movement physics [2].....	11
Figure 3. Achieving yaw motion of quadcopter [2].....	12
Figure 4. jmt X4M360L frame.[7].....	14
Figure 5. An electric DC motor consisting of conductive armature, permanent magnets, brushes and power source. [9]	15
Figure 6. Outrunner brushless DC motor [10].....	16
Figure 7. Inrunner brushless DC Motor [11].....	16
Figure 8. Propulsion of electromagnet activating signal which spins the rotor [12].	16
Figure 9. Rotor movement due to coil activation [15].....	18
Figure 10. BLDC states and rotation [15].....	19
Figure 11. Voltage drop on input A [15].....	20
Figure 12. Power distribution system of quadcopter [21].....	23
Figure 13. RadioLink T8FB receiver wiring diagram [26].....	25
Figure 14. APM 2.8 flight controller connections diagram.[30]	26
Figure 15. The scheme of the PID-controller as a sum of three components. [34].....	28
Figure 16. Static error [37].....	29
Figure 17. Basic structure of Ardupilot software architecture [38].....	30
Figure 18. Readings of Roll angle sensor.	34
Figure 19. Readings of Pitch angle sensor.	35
Figure 20. Readings of Yaw angle sensor.....	35
Figure 21. Readings of Roll angle sensor.	36
Figure 22. Readings of Pitch angle sensor.	36
Figure 23. Readings of Yaw angle sensor.....	37
Figure 24. C++ function which calculates PID controlling output signal and takes error and scaler parameters as input.	44
Figure 25. Simulink schematic of MAVLink messages deserializer.....	45
Figure 26. Mission Planner GUI [25].	46
Figure 27. All quadcopter PIDs' values are set to maximum [25].	46
Figure 28. New PID values, which are set to more optimal values based on Matlab graphs observation [25].....	47

1 Introduction

The goals of this thesis are to determine what a quadcopter is, examine principles of its work, do brief source code analysis, do Matlab integration with the quadcopter hardware and tune control algorithm. To achieve needed results a physical drone was a better solution. To construct drone that is suitable for aims of this work it was important to find information on each required components for the drone: what is it, why is it needed and what characteristics of the detail should be. Main way of getting this information was through online research.

Constructed quadcopter is used for code analysis of the flying control algorithm. Ardupilot [1], open-source autopilot software based on Arduino microcontroller, was chosen for this purpose. As result it was possible to interface drone's hardware with Matlab environment and get useful data of quadrotor behavior. This information would allow to observe the quadcopter more precisely and tune PID regulators parameters more conveniently and accurately.

2 What is a quadcopter

A quadcopter, also called quadrotor helicopter or drone (Figure 1), is an aircraft which uses four rotors and propellers mounted on top of them to achieve lift. Unlike classical helicopter which uses a combination of pitch and yaw rotors, and propellers, to fly a quadcopter uses four pitch propellers to generate desired thrust. [2]

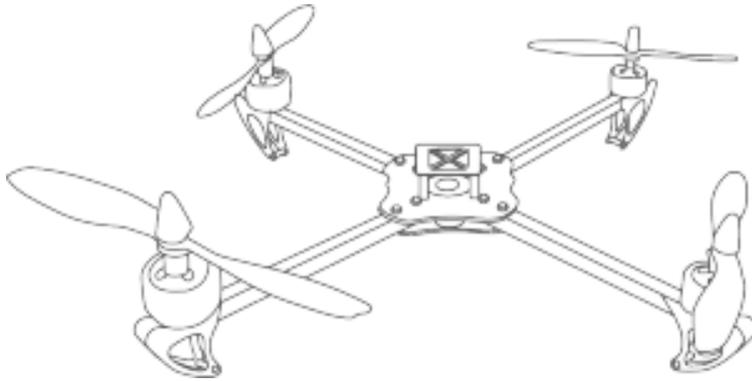


Figure 1. A quadcopter with X shaped rotor setup [3].

Propellers come in two pairs: one pair is rotated clockwise and the other one is rotated counter-clockwise. This setup allows achieving desired control by varying the speed of these propellers [2]. Clockwise and counter-clockwise rotations are needed to balance out the angular momentum of the drone. If all of propellers rotated in one direction, the quadcopter would rotate in same direction as well, which would make it impossible to control.

Laterally sideways movement (roll) of a quadcopter can be achieved by a tilt motion. To do this is needed to two side motors should be sped up and the other side rotors should be slowed down (Figure 2).

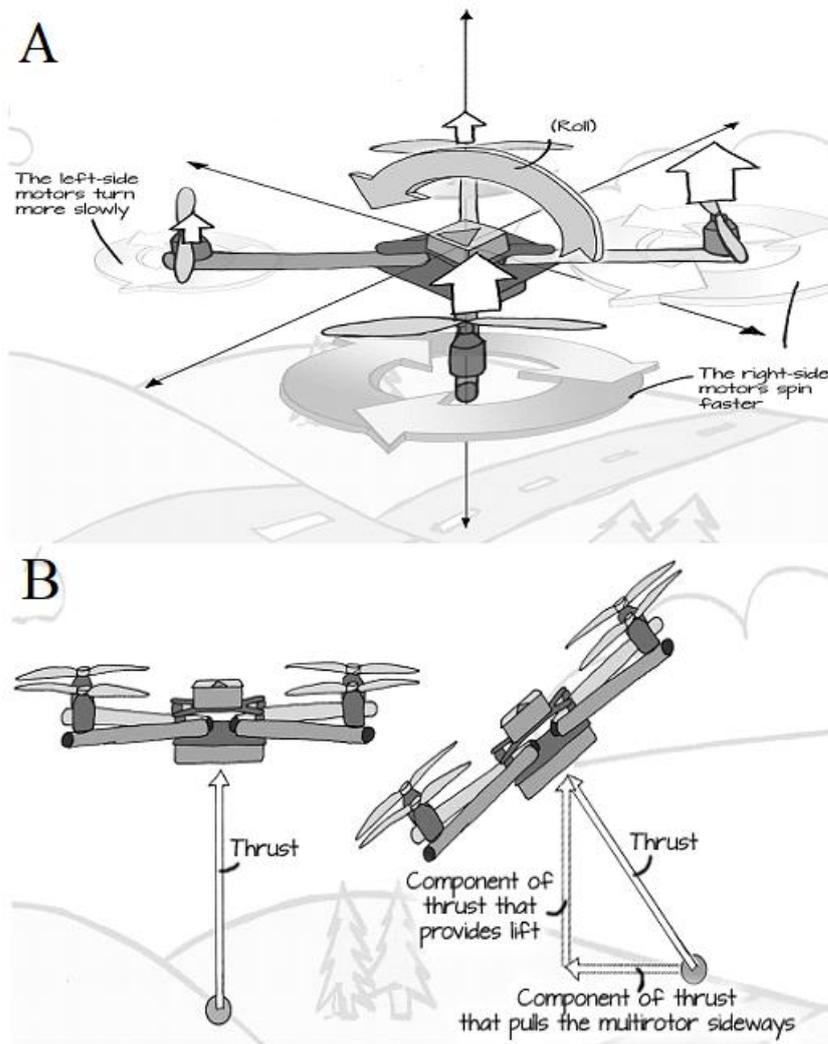


Figure 2. Lateral sideways movement. A-motors, B- movement physics [2].

To make quadcopter yaw it is necessary to unbalance the angular momentum of quadcopter. This is achieved by speeding up clockwise rotating motors and slowing down counter-clockwise rotating motors, as result, the body of the multirotor turns counter-clockwise (Figure 3).

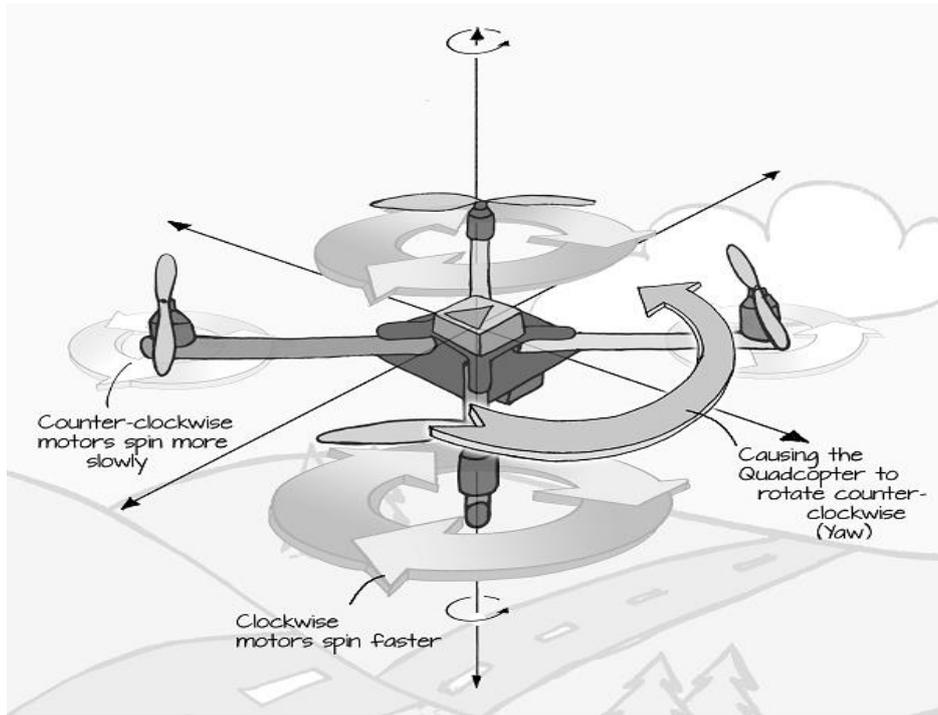


Figure 3. Achieving yaw motion of quadcopter [2].

3 Components of quadcopter

The most simple quadcopter generally consists of 9 components, which are: quadcopter frame, 4 brushless dc motors, 4 propellers, 4 electronic speed controllers (ESC), inertial measurement unit (IMU), radio control unit (RC), power unit (battery), power distribution board and flight controller. IMU sensors can be integrated in flight controller, which makes it considerably easier to setup a custom multirotor drone. [4]

Designing a custom quadcopter requires forward planning on what type of quadcopter it should be: a racing drone, an Aerial cinematography quadcopter, a professional quadcopter (such as package delivery) or a research unit. Depending on what kind of quadcopter is chosen, its physical and dynamic characteristics will differ. For example, a racing FPV (first-person view) drone is small, light and quite maneuverable, and an Aerial cinematography quadcopter is massive, heavy and steady to be able to carry different kind of cameras and take videos in stable manner. [5]

Planning a quadcopter starts with choosing a base for all electronics - a quadcopter frame. Once it is decided on what frame to get, the heaviest part of a drone is chosen next – the battery, a power source of a quadcopter. As soon as approximate weight of the drone is known, by adding up frame's, battery's and electronics' weights, choosing motors is the next step. Motors need to have enough thrust power to lift off the drone from the ground and steadily hover. Propellers are usually chosen depending on frame dimensions. For motors to work properly, electronic speed controllers are needed. Properly connecting all of the motors to a power source require a power distribution board. All of the remaining electronics: RC, IMU and flight controller – are usually chosen depending on personal preferences [6]. In the following subchapters all of the components are described in greater detail and specifications are presented for all of this project's used components, as well as the reasons behind them.

3.1 Frame

Frame is the component where all of electronics is mounted. Depending on specifications of required drone it will either be small, light-weighted and fragile, or big, medium-weighted and sturdy. Mostly all of frames are made of carbon fiber to balance out the durability-to-weight ratio. In this work's quadcopter should be highly customizable meaning that it can house such things as GPS, Wi-Fi or even a camera. Because of this, the size of frame should be bigger than average. A considerably good choice was jmt X4M360L frame (Figure 4). It is big enough to house all required electronics and has option to install stabilized dual axis camera mount.

Dimensions and specifications of frame:

- Width – 265mm
- Height – 200mm
- Length - 265mm
- Empty frame weight – 206g
- Recommended battery – 11.1v 3s 2200-5500mAh with instant discharge current capacity > 20c
- Load - about 1.5kg

Considering these specifications, it can be determined approximately what kind of power source and motors are needed.



Figure 4. jmt X4M360L frame [7].

3.2 Motors

An electric motor is an electromechanical machine that converts supplied electric energy to output mechanical energy. It is done on basis of Lorentz Law – when a wire carrying an electric current is placed in a magnetic field a Lorentz force is experienced. Electric motors use this principle to create a torque by placing a square-shaped coil of wire in magnetic field so that when there is current in the wire, the magnetic force would rotate the rotor armature thus creating motoring action (Figure 5). One type of electric motors is a DC motor. It uses a commutator, which consists of conductive and insulative segments, and brushes, through which current flows from power source to the armature. The coil armature and commutator ring are fixed to rotating shaft. As the armature rotates, due to the Lorentz force acting on it, the torque is created. When brushes reach non-conductive segment of the commutator, there is no more current in the coil, but it still continues rotating due to inertia force. When brushes reach the next segment, the polarity in coil wire is switched. This results in constant rotating motion. [8]

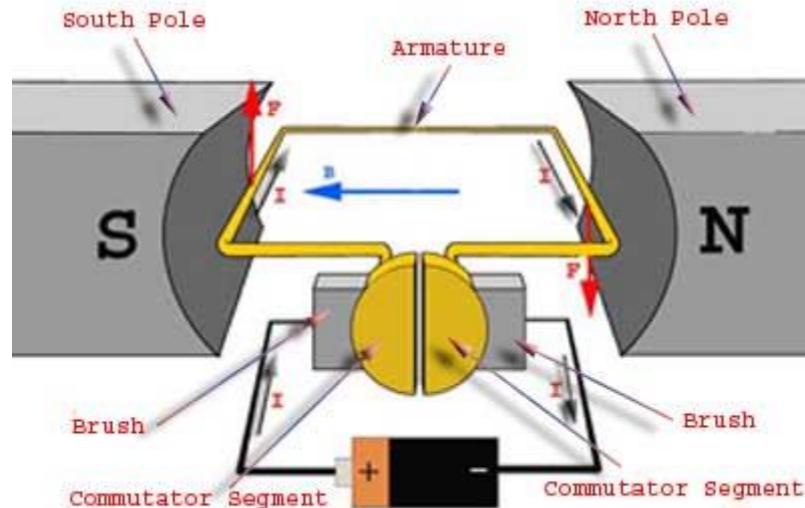


Figure 5. An electric DC motor consisting of conductive armature, permanent magnets, brushes and power source [9].

Type of motors used in quadcopters are different – they do not use brushes to break contact with armature and switch polarity to continue rotating motion. Instead they contain a series of electromagnets which are connected together in specific pairs. There are two different type of brushless motors: outrunner and inrunner (Figure 6 and Figure 7, respectively). They have same working principle, but different composition. The

difference between them is that outrunner motors have rotor part on the outside and inrunner motors vice versa.

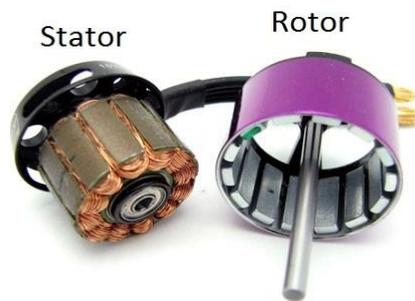


Figure 6. Outrunner brushless DC motor [10].

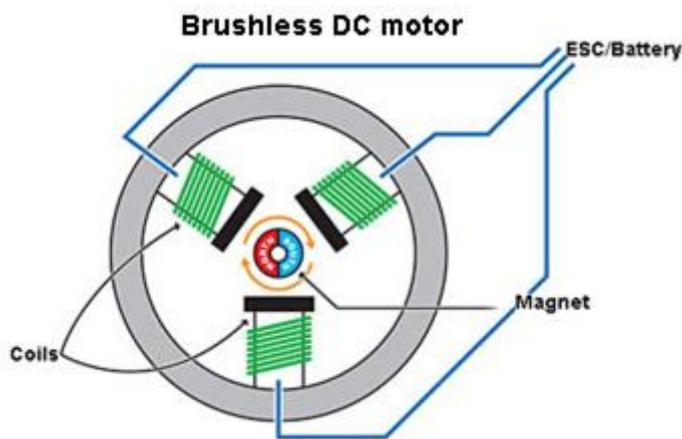


Figure 7. Inrunner brushless DC Motor [11].

Both types use an external electronic unit known as ESC to control the motor by activating and deactivating specific segments of electromagnetic coils in an extremely precise times to cause rotor spin due to magnetic force (Figure 8). Electromagnets are connected into 3 main sections thus brushless electro motors have 3 wires coming out of them.

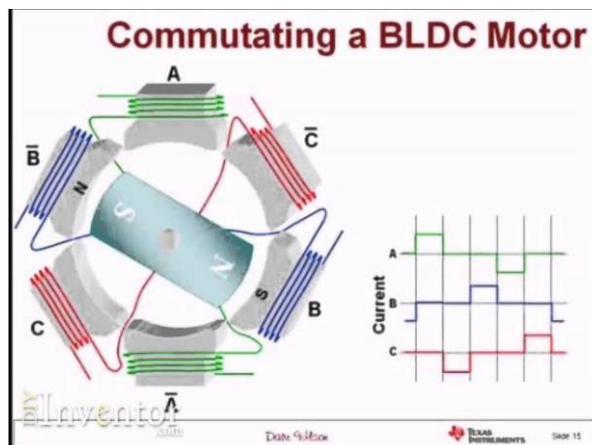


Figure 8. Propulsion of electromagnet activating signal which spins the rotor [12].

Quadcopters generally use outrunner type of brushless motors since they generate more torque. As all electric motors, quadcopter motors come in different sizes and power ratings. Their power ratings are measured in K_v , which is known as voltage constant, or dynamo constant, and is expressed in RPM per Volt. It defines how many RPM motor will have if one Volt of voltage will be supplied. For example, if 11.1V will be supplied to a 750 K_v motor, then $11.1 \times 750 = 8385$ RPM the motor will produce.

Best way to choose motors for a quadcopter is to check specifications of used frame and use electro motors with recommended K_v rating. General rule is that the lower K_v rating of the motor, the more torque it will produce. Hence low K_v motors are usually installed on larger, heavier drones and high K_v motors are used in more light-weight quadcopters.

The other way is to calculate approximate weight of the quadcopter and choose motors with K_v rating that will generate enough thrust to lift off and hover steadily. Generally thrust generated by motors should be at least 2 times more than quad copter's weight + 20% to be absolutely sure - 220% in total. To sum up, if we have a 1kg quad copter, we need a total thrust of about 2.2kg. Divided by 4, it's at least 550g of thrust per motor.

To calculate thrust serious simulation software is needed which take in account such things as propeller geometry, airspeed, RPM, electrical power and efficiency, and etc. It is much easier to calculate static thrust case, which uses following formula [13]:

$$T = [(\eta \cdot P)^2 \cdot 2\pi \cdot r^2 \cdot \rho]^{0.3333}, \text{ where}$$

- T = thrust (in Newtons)
- η = propeller hover efficiency = 0.7-0.8 is typical for installed low-pitch propellers
- P = shaft power = voltage * current * motor efficiency (in Watts)
- $\pi = 3.14159$
- r = propellers radius (in meters)
- ρ = air density = 1.22 kg/m³

But, best way is to use motor datasheet and find what thrust it will generate.

Based on datasheet states of chosen frame the maximum gross take-off weight should not be more than 1500g. Considering this, suitable motors - *JMT F15844 920KV* were chosen.

3.3 ESC

Electronic speed controller is additional electronic unit that is needed for 3 phased brushless DC motors to work properly. It activates each of electro magnets pairs in motors at a precise time to create a magnetic force which gives a motor its rotation motion. [14]

Rotor motion depends on what magnetic coils are activated (Figure 9). Changing activation of coils will be referred to as switching the states. Generally, there are 6 states. In first state coils A and C are activated. Input A is activated as ground and input C is given positive voltage. Input B is left neutral. Coil C now has north polarity and coil A has south polarity, the rotor will spin accordingly – rotor north pole will be attracted to coil A south pole and pushed away from north pole of coil C. To continue spinning motion of rotor coil B will need to have south polarity, which achieved in state 2. This cycle repeats by switching states 1 to 6 until reach starting position of rotor and starts again. [15]

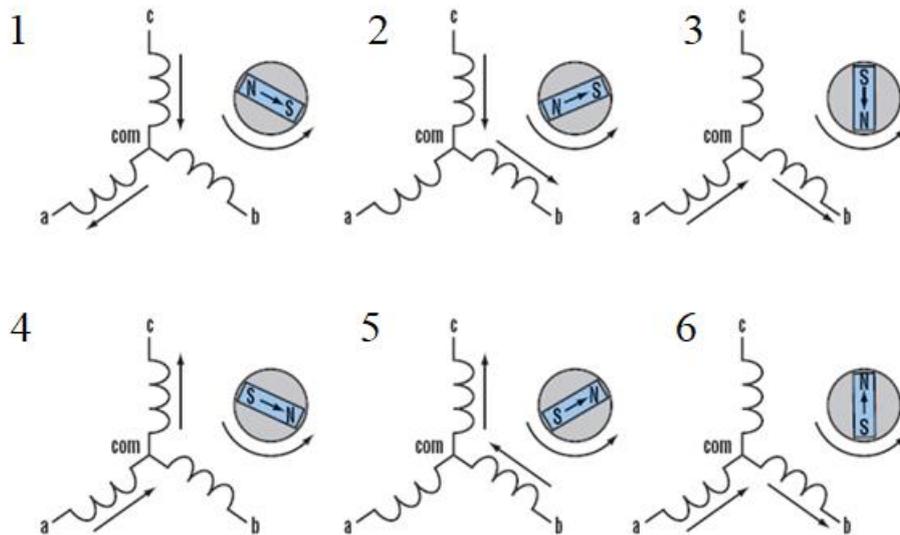


Figure 9. Rotor movement due to coil activation [15].

On Figure 10 is illustrated a brushless motor with 6 electromagnetic coils which form 3 pairs: A, B and C, which are implemented via transistor inputs. As with coils, there are 6 states which are switched in a specific order for a rotor to spin. In 1st state input A is left neutral, B is High and input C is Low. When input is High, it means that the corresponding coils are given positive voltage; when input is Low, it means that those coils are ground;

and when input is Off state, it means that coils are given high impedance input. In the following case, coils B are given High which means that they will have north polarity and coils C have Low input, which means that they will have south polarity. Due to repelling and attractive forces rotor will spin a few degrees in counter-clockwise direction. The next states will ensure that the motor will keep spinning. [15]

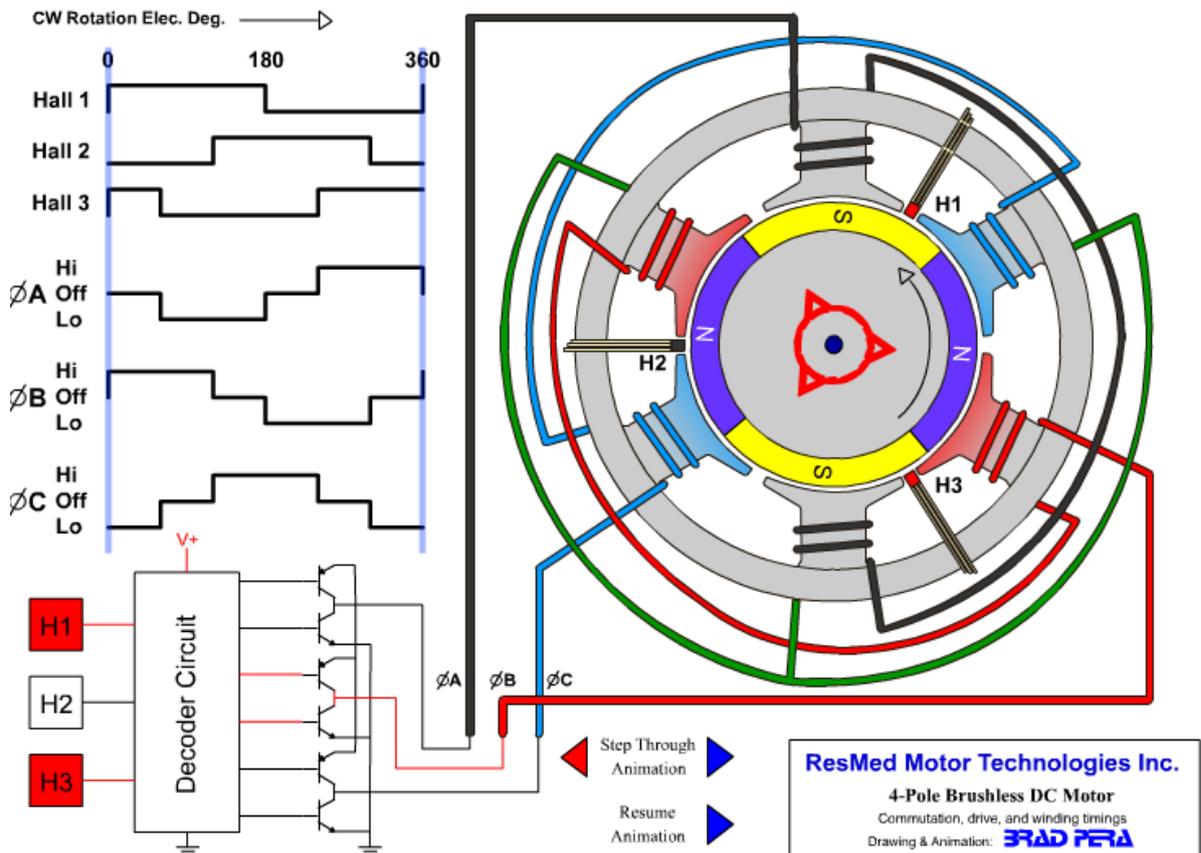


Figure 10. BLDC states and rotation [15].

Switching states at right time is an issue that can be solved by using the 3rd input, which is neither positive voltage, nor ground. In case of first state, it is coil A. According to Faraday's Law, when magnetic field passes through a conductor it induces EMF in it [16]. When a pole will pass through a coil at its peak position, there will be significant voltage drop at that input because of back EMF (Figure 11). That voltage drop can be registered and when it happens the next state can be switched. This will ensure that states will be switched at best time.

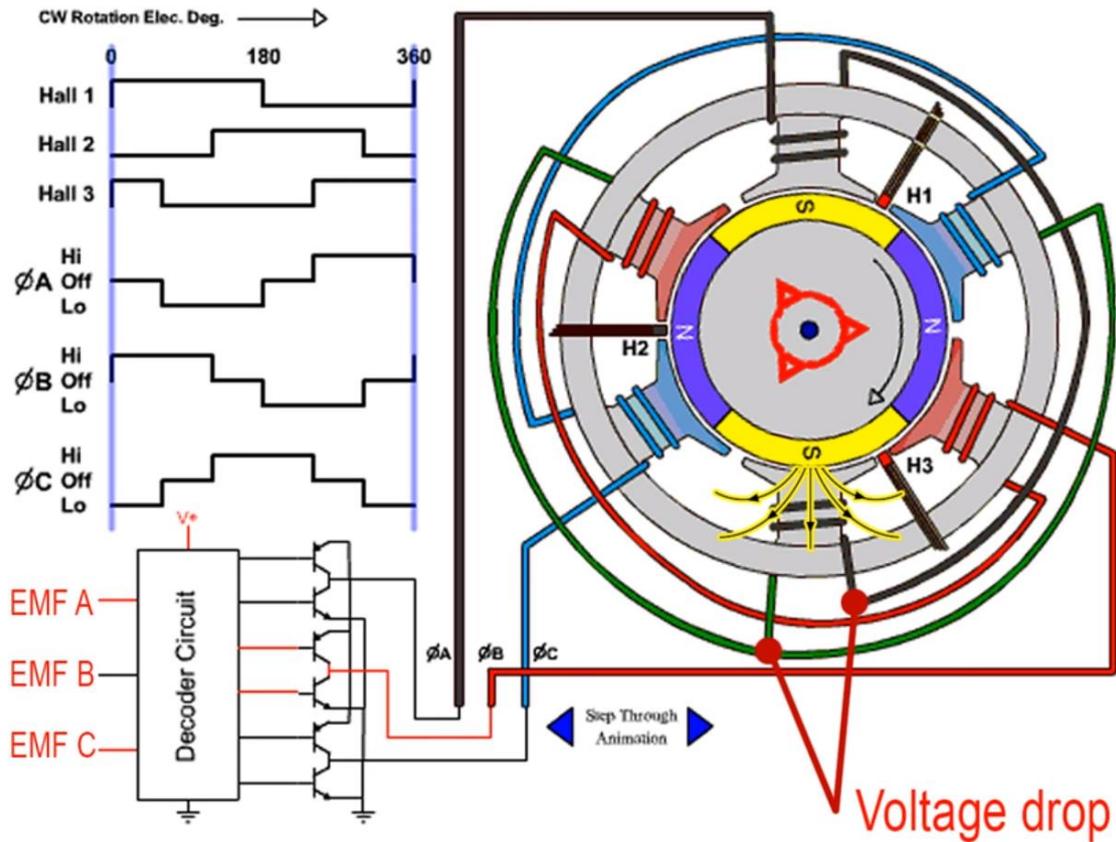


Figure 11. Voltage drop on input A [15].

Another solution is to use motors with Hall Effect sensors. These motors will do the same thing but at more precisely. That is what electronic speed controller does – it switches the states of coils and ensures that the motor will rotate efficiently. Additionally, it converts high voltage of the battery to more appropriate one, since electronics only employ 5V of voltage [15]. To choose appropriate ESC it is needed to know how much current the motors will require. For example, this work’s chosen motors draw 30A of current, so ESCs should have output at least 30A. There are many ESCs that will fit the required characteristics. ESCs differ in the quality of used electronics and types of voltage converters. 30A SimonK ESCs is suitable and is used in this work’s quadcopter. It has high quality electronics to ensure best performance and will supply 40A at its peak.

3.4 Propellers

Propellers are essential part of any multirotor copter. Propellers of quadcopter have same working principle as of helicopter’s ones. But while helicopter generate lift force while employing only one propeller and adjusting its direction by changing the attack angle

[17], quadcopters, use 4 propellers to do that [18]. Choosing proper propellers is important because otherwise amount of thrust and stability will be lacking, and there even can be damage to motors as they will overheat.

Generally, flight efficiency is related to amount of air contact propellers are experiencing. Two characteristics are important for that: propellers' diameter and pitch. Diameter is distance between two farthest points of propeller, or its length. Pitch is a measurement of how much distance will propeller travel per one revolution. [19]

The best way to choose propellers is to use recommended specifications for the motors. If quadcopter is lightweight one and motors are over 1000K_v then smaller diameter and pitch propellers should be used. On the other hand, if it is a big quadcopter and motors are low K_v then larger propellers should be used to generate more thrust and torque. [19]

Additionally, a distance from center of drone to center of motor is measured and depending on it propellers are also chosen. This is done to ensure that motor will not overlap with each other and have enough space between each other to not affect one another because of turbulence. [19]

Considering the motor type *JMT F15844 920KV*, 12 inches in diameter and 6 inches in pitch propeller should be quite suitable, but since in future quadcopter might be using additional unit for aerial video capture, it is better to choose lower pitch propellers. The final choice was 8.4*4.3 propellers, where 8.4 is diameter in inches and 4.3 is pitch in inches.

3.5 Power source

Since all of the electronics need a power source to power it, a quadcopter would need one as well. When choosing power source, or a battery, for the drone several points need to be considered. Firstly, how much voltage do motors need to spin properly; secondly, what capacity of the battery should be; and thirdly, discharge rate of the battery, or current it can provide.

Generally, LiPo batteries are used to power drones. Its voltage depends on the cell count. Quadcopter batteries have small 3.7V batteries connected in series inside of them. Cell count is how many of that 3.7V batteries are there. For example, if it is specified that a battery is 3s, then there are 3 3.7V batteries connected in series and so it would be in total 11.1 volts of energy. Capacity refers to mAH. It means how much milliAmps of current can load draw in an hour time. For example, 1000 mAH battery can supply 1A for one hour before it depletes. And finally, discharge rate, or C rating, of battery represents the absolute maximum current that should be drawn from the battery. For instance, 2200 mAH battery with rating of 30C would be rated for $30 * (2200 \text{ mAH} / 1000) = 66\text{A}$. [20]

Quadcopters usually need 11.1V for motors to properly operate, so 3S, or 3 cell, would be needed. ESCs draw at least 30A of current and therefore a power source with 30C rating should suffice. The capacity of battery will affect the flying time. It is important to mention that the more capacity the battery has, the heavier it is, which in turn also affects the flying time.

Turnigy 2200 mAh 3S 20C and *HRB 2600 mAh 3S 30C* were chosen to supply power to the drone. Their weight are 188g and 218g respectively.

3.6 Power distribution system

There is not much to power distribution system in quadcopter as it is no more than a simple electronic circuit board which connects ESCs to a battery and powers them. Though, power distribution boards can be complex as well – they can embed voltage regulators and overvoltage protectors in them [21]. All in all, board can be simple enough just to connect all ESCs and RC component to the power source (Figure 12).

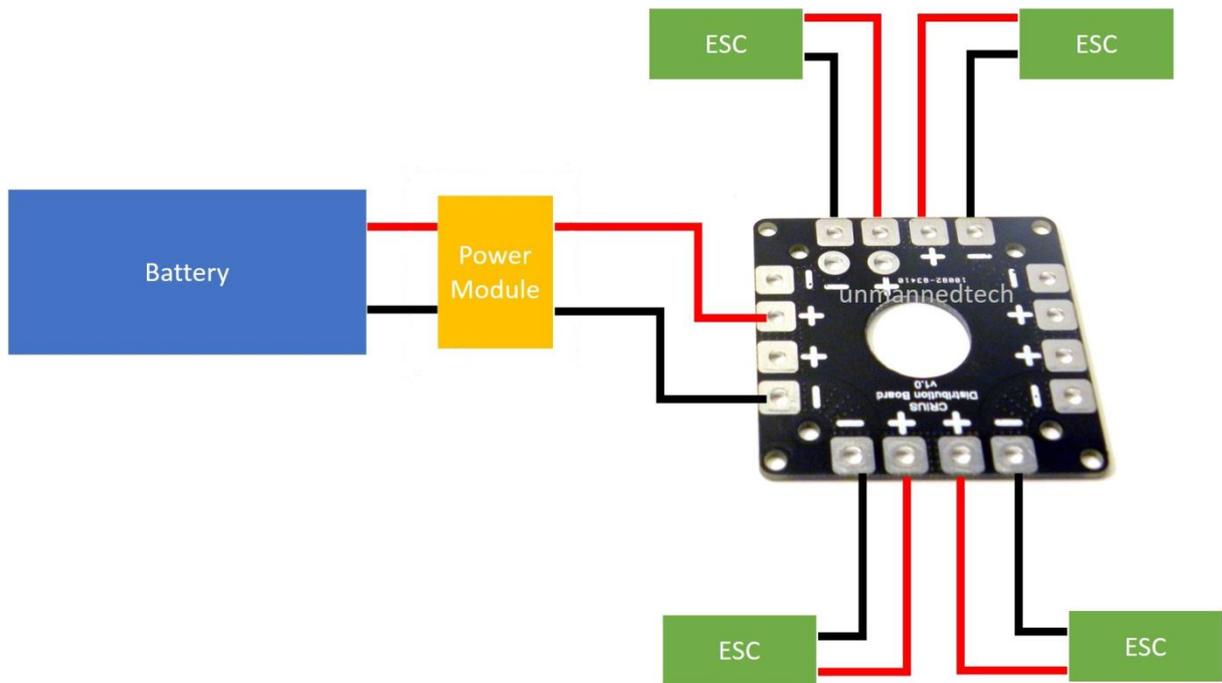


Figure 12. Power distribution system of quadcopter [21].

In this work used drone RC module is connected to the board, however, it is powered by BEC 5V outputs of ESCs. Chosen ESCs already have overvoltage protection and there is no need for additional one. Additionally, RC inputs are connected to board as well to give throttle signals to ESCs and motors respectively.

3.7 RC (Radio control) module

Quadcopters need to be controlled from distance and a wireless system is required for that. A transmitter controller and receiver module are used for this purpose. A joystick remote control is widely used to direct movement of the drone. The receiver module gets signal from the remote controller and adjusts motion of quadcopter by varying speed of motors. [22]

A remote controller (transmitter) and receiver module always come in pair as different companies utilize different types of radio technologies to transmit signal and it is important to use matching equipment. [22]

Nowadays most RC equipment uses spread spectrum to transmit signal via radio, instead of narrow band broadcasting. Conventional wireless communication utilizing narrow band frequencies will not be suitable for quadcopter control because it is extremely prone

to interferences, which in turn will be dangerous and inefficient power wise. Quadcopter can easily lose control and not only damage itself, but also its surroundings, including people. To avoid such issues more modern, complex and improved technology of wireless communication is used known as Frequency-hopping spread spectrum (FHSS) and Direct-sequence spread spectrum (DSSS). Both of them use wide range of frequency spectrum to communicate. It means that instead of using only one frequency, spread spectrum technology utilizes several frequencies, 79 in total. For FHSS communication is done by hopping from one frequency to another one in a specific pattern. For example, for 15 microseconds signal is broadcasted on 2.401 GHz frequency, then it hops to 2.408 GHz frequency to send data there for another 15 microseconds and so on. This is done to minimize amount of interference between different broadcasting stations. DSSS has even better interference resistance because it uses much wider frequency spectrum. In essence, the message signal is used to modulate a bit sequence known as a Pseudo Noise (PN) code; this PN code consists of a radio pulse that is much shorter in duration (larger bandwidth) than the original message signal. In quadcopter RC DSSS is more commonly used as it is cheaper and has better data rate. [23][24]

RC remote controller must be able to transmit on at least 6 different channels to control all axis of motion: pitch, roll, throttle and yaw. Two remaining channels are used to control quadcopter modes and auxiliary functions such as emergency landing. [23]

The receiver module gets control signals from the remote control and it in turn propagates signals further to flight controller. If receiver module uses separate wires for different channels, PWM (pulse width modulation) is used to send signal to flight controller for processing. Otherwise PPM (Pulse position modulation) sum can be used, and it needs only one wire as it combines all of the channels' signals into one. [25] RadioLink T8FB RC remote controller and RadioLink R8EF receiver modules are used in this work quadcopter. It uses 2.4 GHz and 8 channels to control the quadcopter in PWM and PPM modes (Figure 13).

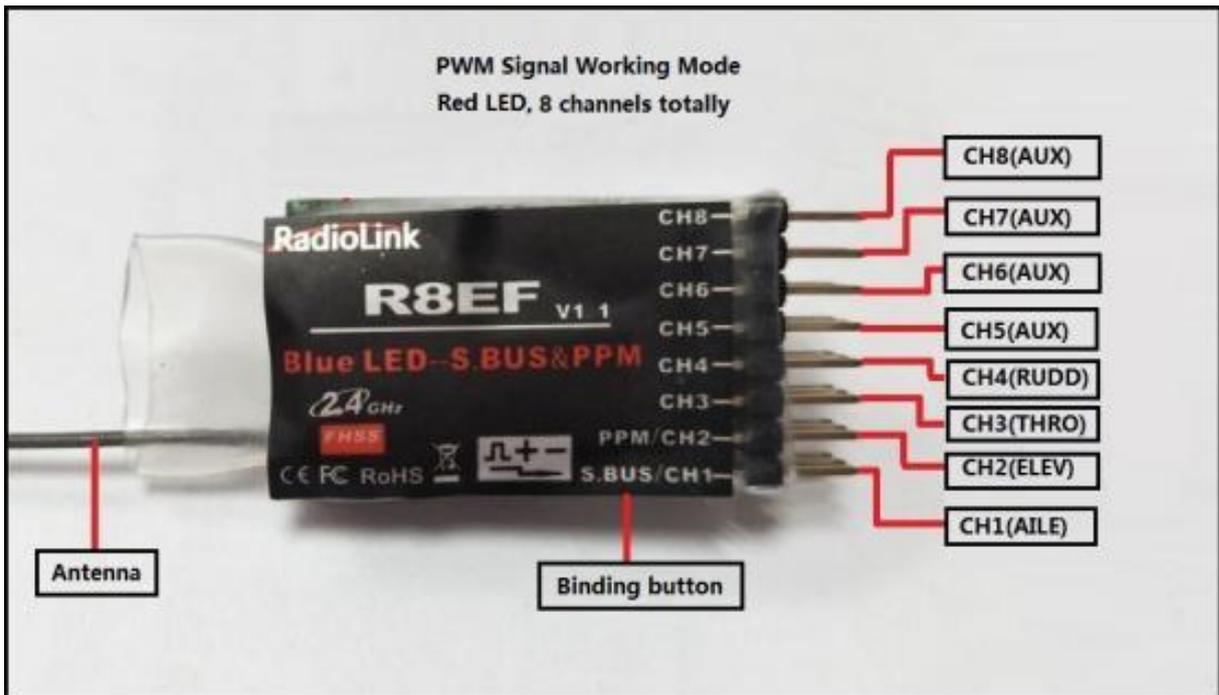


Figure 13. RadioLink T8FB receiver wiring diagram [26].

3.8 Flight controller and Inertial Measurement Unit

A flight controller is the core processing unit of quadcopter. Its function is to control RPM of each motor in response to input from RC. [27]

For a quadcopter it is impossible to know which direction it is moving on its own, what its height in reference to the ground or if it is hovering stably in vertical position. Most flight controllers employ additional sensing unit which helps flight controller to calculate optimum response. An IMU (Inertial Measurement Unit) is used for this purpose - a module that will measure velocity, orientation and gravitational forces of the quadcopter via 3-axis accelerometer, 3-axis gyroscope and additional 3-axis magnetometer [28]. This unit might also include barometer, a GPS module and other required sensors. Accelerometers are electromechanical devices that sense either static or dynamic forces of acceleration. Generally, accelerometers contain capacitive plates internally. Some of these are fixed, while others are attached to miniscule springs that move internally as acceleration forces act upon the sensor. As these plates move in relation to each other, the capacitance between them changes. From these changes in capacitance, the acceleration can be determined [29]. Gyroscopes, or gyros, are devices that measure or maintain rotational motion. When the gyro is rotated, a small resonating

mass is shifted as the angular velocity changes. This movement is converted into very low-current electrical signals that can be amplified and read by a host microcontroller. This way a gyroscope knows rotational acceleration of all axis of quadcopter. Magnetometers measure the magnetic field around them and, such as simple compass, help a quadcopter determine its orientation in space. [29] All this sensors play essential role in stabilizing and controlling quadcopters (Figure 14).

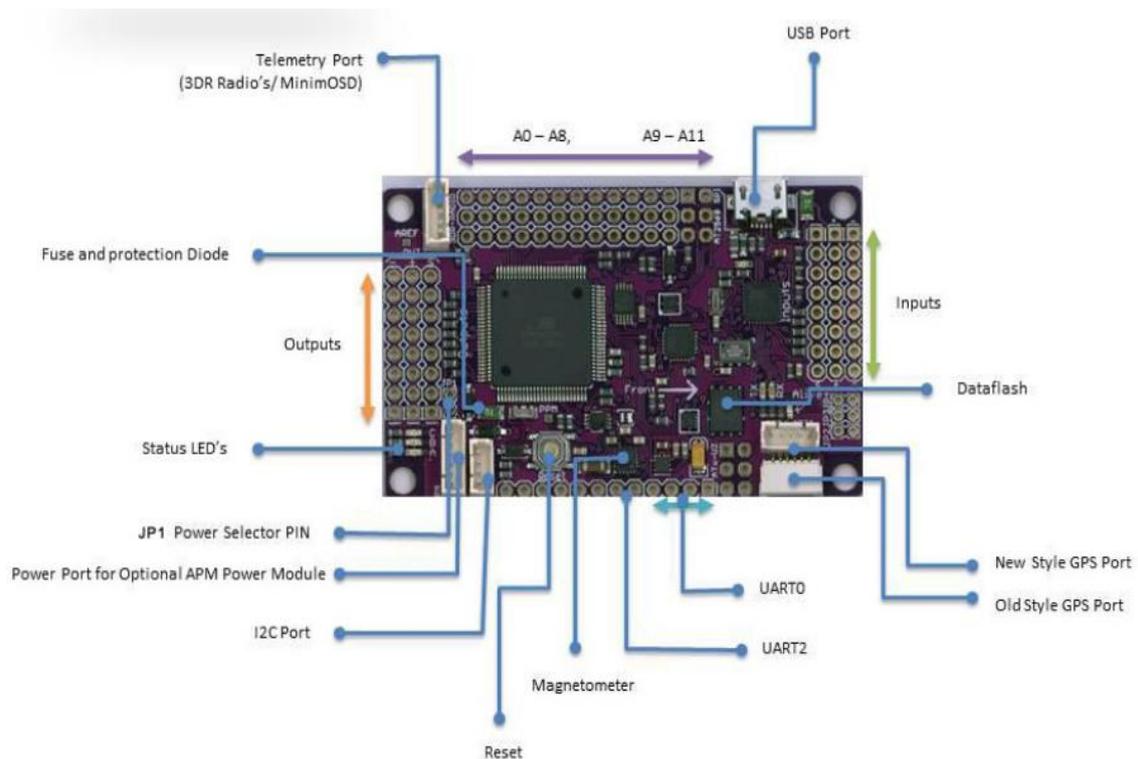


Figure 14. APM 2.8 flight controller connections diagram [30].

A flight controller takes readings from each of the sensors and uses that information in control algorithm – a Proportional Derivative Integral control loop. Its main task is to adjust output to motors with respect to error between desired position of quadcopter and actual position, for instance. Flight controllers are configurable and programmable, allowing for adjustments based on varying multi-rotor configurations. Gains or PIDs are used to tune the controller, yielding snappy, locked-in response. [27]

The choice for this drone is APM 2.8 (Figure 14) flight controller, which uses ArduPilot open-source firmware. It is based on Arduino Mega 2560 microcontroller and is programmed using C++ language. It supports various multirotor setups and hardware. [1]

3.9 Miscellaneous: GPS and Wi-Fi modules

Additional telemetry modules can be used in quadcopter to improve its flying capabilities. A GPS module can be used to monitor the position of the quadcopter on a map. By plotting a path on a map and using quadcopter's position, it is possible for a quadcopter to follow that path without being controlled by RC unit, or fly autonomously. This feature can be useful in such cases as low visibility or loss of signal with remote controller. [31]

APM 2.8 flight controller does not have an internal GPS module, but support I2C bus communication protocols which can be used by an external GPS module. M7N GPS module is made specifically for APM 2.8 flight controller and it does not require any additional drivers or soldering to implement. This GPS module also has internal magnetometer, or compass [30]. Magnetometer provides information on the direction at which quadcopter is pointing, and allows much more precise autonomous control [29].

One more additional telemetry unit which can be used in a quadcopter is a Wi-Fi module. It is mainly used for a quadcopter to communicate with ground station, such as a laptop or even smart phone, to provide pilot with information from all of electronic units. That can include data from all IMU sensors, battery voltage, position via GPS, input from RC, output to motors and so on [30]. Pilot can monitor what is happening inside of quadcopter while it is in motion and use that information to tune PID values in more efficient way. All gained information can be used in third-party programs to get monitor quadcopter on much different and efficient scale. One such program is MathWorks Matlab environment, which has much better tools of controlling complex systems.

4 PID controllers

PID controller is used in systems with feedback. PID stands for Proportional-Integral-Derivative. The essence of this controller is that, like all the other controllers with feedback, it constantly calculates the Error $e(t)$ as the difference between the required value $r(t)$ (Setpoint) and the process variable $y(t)$ and tries to correct this error by forming the control signal $u(t)$, which is the sum of the proportional, integral and derivative parts. [32] [33]

More detailed description of individual components of PID (Figure 15):

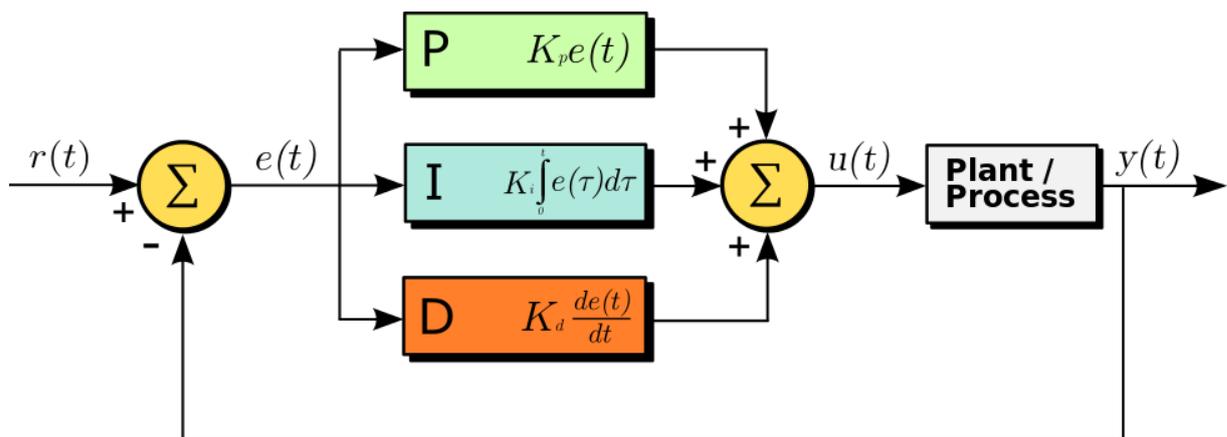


Figure 15. The scheme of the PID-controller as a sum of three components. [34]

- P - proportional component produces an output signal in proportion to the error. That is, the greater the error, the difference between the desired and available values, the more corrective impact there is [33]. In many systems, the use of only the P component is sufficient, but, specifically in quadcopter system, proportional control will not be enough, because due to the nature of the system, there will be many fluctuations in certain situations. These are called disturbances. In case of quadcopter, disturbances are mainly wind and self-caused turbulences. The higher P gain, the harder it tries to stabilize the copter. Generally speaking, high P gain means sharper control while low P gain means softer control. But if P is too high, the quadcopter becomes too sensitive and

over-correct itself. Eventually it will cause overshoots, and a quadcopter will have high frequency oscillations. [35]

- I is an integral component. In the mathematical sense, the integral part sums all the errors through the integral, and on the basis of this sum forms a control action. Most often, the integral component is used in conjunction with the proportional part. The essence of this union is that in many systems, the use of only the P part leads to the appearance of a static error or Steady-State error (Figure 16). Adding an integral component in the absence of interference should zero this error. In the quadcopter, the values of I part are not large, otherwise the error will be constantly accumulated and either there will appear overshoot or constant oscillation. Integral component work in an environment with high disturbances (e.g. wind). [36]

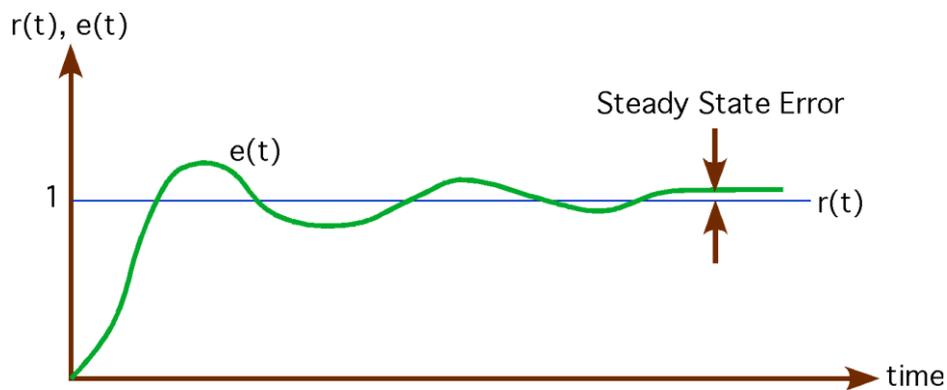


Figure 16. Static error [37].

- D is the derivative component. This part of the PID controller "anticipates" the future state of the system and, based on this, generates control actions. In the mathematical sense, at some point the derivative is taken, namely, the tangent to this point, and we look whether the derivative is positive or not, and how big it is. If the derivative is positive, the signal needs to be reduced and vice versa. [33]

5 Ardupilot control algorithm code analysis

5.1 Code architecture overview

The basic structure of Ardupilot architecture mainly consists of 5 parts (Figure 17):

- vehicle code
- shared libraries
- hardware abstraction layer (AP_HAL)
- tools directories
- external support code (i.e. mavlink, dronekit)

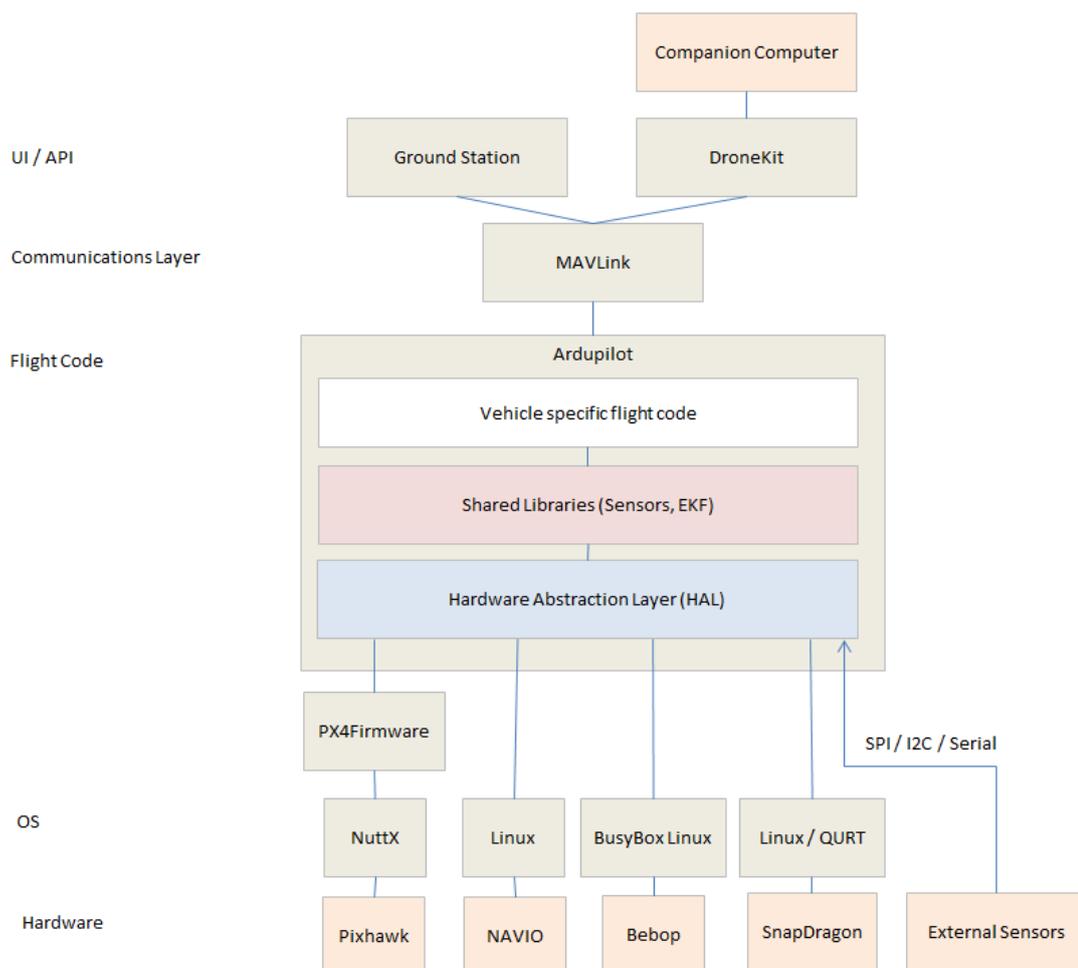


Figure 17. Basic structure of Ardupilot software architecture [38].

The vehicle directories are the top level directories that define the firmware for the specific vehicle. Since Ardupilot platform is not made only to support quadcopters, but vehicle such as planes and rovers, it consists of many different setups. They are needed to account for different dynamics these vehicles have. [38]

Shared libraries include sensor drivers, attitude and position estimation (aka EKF) and control code (i.e. PID controllers). [38]

The AP_HAL layer (Hardware Abstraction Layer) is intermediate layer between hardware drivers and flying algorithm software. Basically, it combines all of available hardware parameters and drivers in a single layer. When Ardupilot software, be it PID controller or position estimation algorithm, needs to communicate with specific hardware, like ESCs or any sensor, it makes a request to HAL layer and it, in turn, communicates with drivers. HAL layer is needed to avoid writing various different firmwares for specific hardware. [38]

The tools directories are miscellaneous support directories. For example, tools/autotest provides the autotest infrastructure. [38]

MAVLink consists of communication protocol libraries and is needed to connect quadcopter to ground station, be it a laptop or a smart phone. Quadcopter can either use a Wi-Fi module to connect wirelessly or USB cable. Via ground station quadcopter can be calibrated, firmware updated, tuned or tracked. [38]

5.2 Copter attitude control

Ardupilot implementation of PID control algorithm in C++ is quite straightforward. Without getting too much into C++ classes and interfaces, the whole algorithm can be summed up into one function, in which all control algorithm components are computed one by one. (Appendix 1, Figure 24)

Firstly, getting time derivative variable is needed. After that P component is computed by adding error to K_p , and then adding it to output variable. Next, D component is computed by resolving previous derivative value which is then filtered through low-pass filter and

multiplied by K_d and added to output. And lastly, integral component is calculated by multiplying K_i by integrator value, which is accumulated throughout quadcopter working time, and finally added to output which forms control signal of quadcopter.

6 Interfacing with Matlab

Matlab environment is a perfect tool to study how quadcopter behaves on different control algorithm setups. As a quadcopter generally has a plentiful amount of different sensors, a good part of them can be used in Matlab to visualize how different tuning setups affect output of PID controllers. This can be extremely useful when a multicopter needs to be set up perfectly.

For a quadcopter to communicate data with Matlab, providing a way for data to travel is needed. Since quadcopter is mobile, using wireless data transmission mediums is preferable as wired mediums will limit drones flying capabilities. Wi-Fi network is perfect for data transmissions over air and thus fits the task well. Ardupilot library provides a way to send internal sensor data over network via MAVLink messages. In a nutshell, these are header files, which are then serialized into a bitstream that is sent over network.

For Matlab to be able to understand these messages an additional module needs to be installed (aditya00j_simulink_mavlink). This module is written in Objective-C and is compiled to be used in Simulink schemas. Once source code is compiled, it can be used as sfunction in Simulink. Sfunction will take input from TCP/IP receiver, which in turn will be connected to the quadcopter via Wi-Fi. Outputs of sfunction are deserialized messages from MAVLink which contain attitude information such as roll, pitch and yaw values. These values can be plotted on a graph and can be studied to see how, for example, big are disturbances depending on different PID values. (Appendix 2, Figure 25)

7 Quad-copter PID tuning and results via Matlab

Proper quad-copter tuning is a complicated task as there are multiple PID and P controllers for different parameters. To more precise, there are PID tuning capabilities for Roll, Pitch and Yaw, as well as Throttle and Altitude hold (Appendix 3, Figure 26). Most important are Roll and Pitch as they are most crucial for proper quad-copter handling. To edit the PID values, one of the ways is to hardcode them into the quad-copter firmware or use Mission Planner software that writes PID values to the drone dynamically.

To see how different PID values affect dynamics of the drone two separate tests were conducted. During the first test the values were set to maximum and it is predictable that the whole system will be prone to disturbances and will overshoot (Appendix 3, Figure 27).

The following 3 graphs will represent Roll, Pitch and Yaw raw angle sensor readings:

- Roll angle sensor readings (Figure 18).

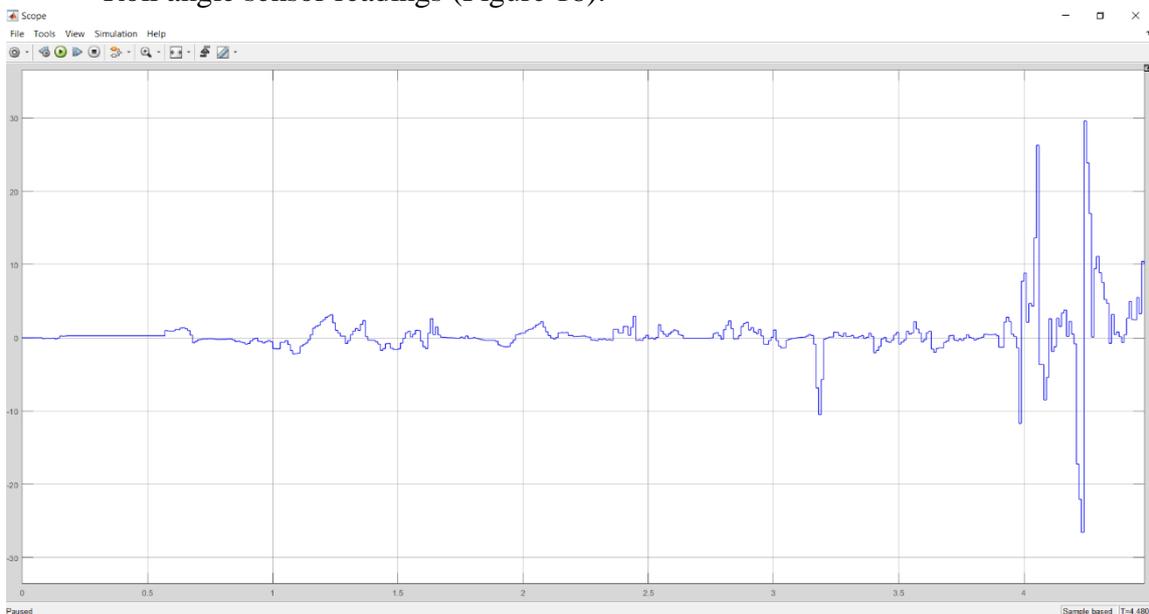


Figure 18. Readings of Roll angle sensor.

- Pitch angle sensor (Figure 19).

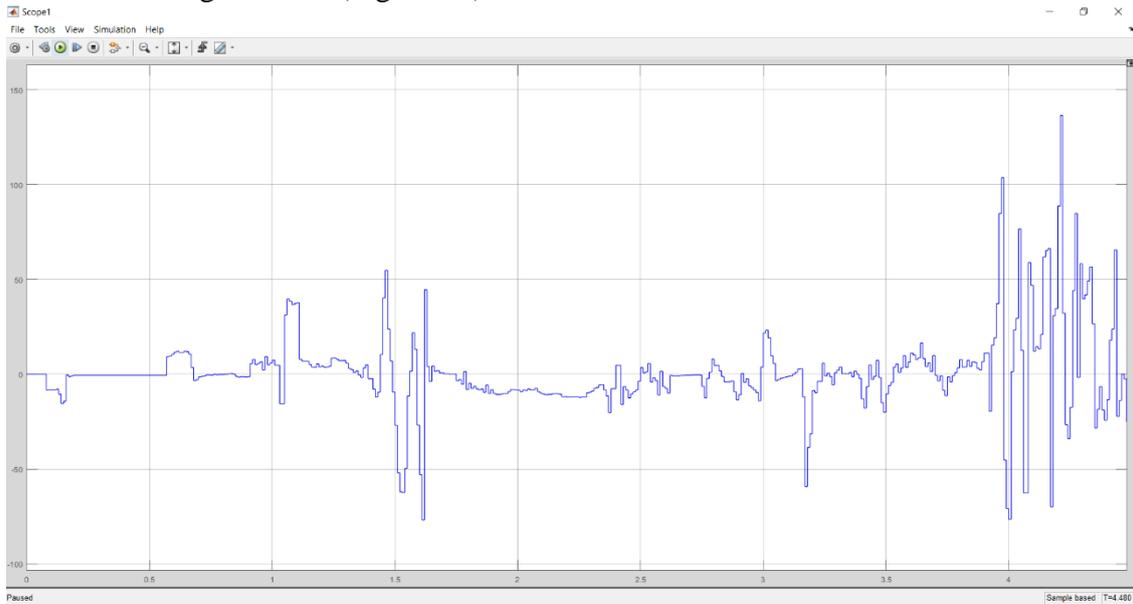


Figure 19. Readings of Pitch angle sensor.

- Yaw angle readings (Figure 20).

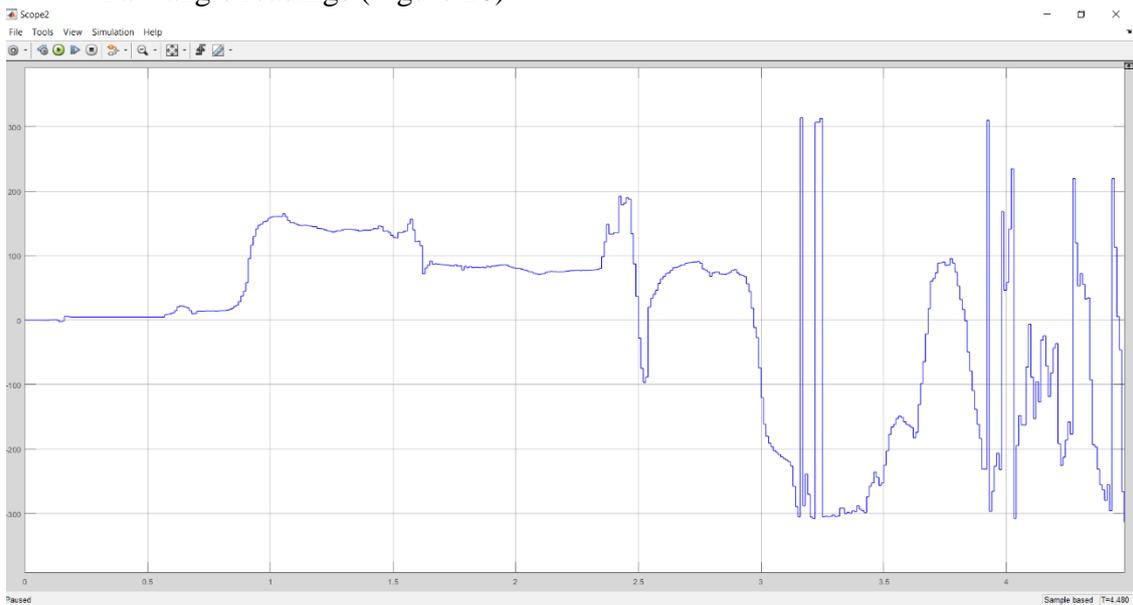


Figure 20. Readings of Yaw angle sensor.

As was predicted, the quadcopter is highly unstable. Each input signal makes quadcopter extremely wobbly and uncontrollable. It was not possible to stabilize it at all and it resulted in crash.

In second test, all of PID values were tuned down (Figure 28).

And results were:

- Roll angle sensor readings (Figure 21).

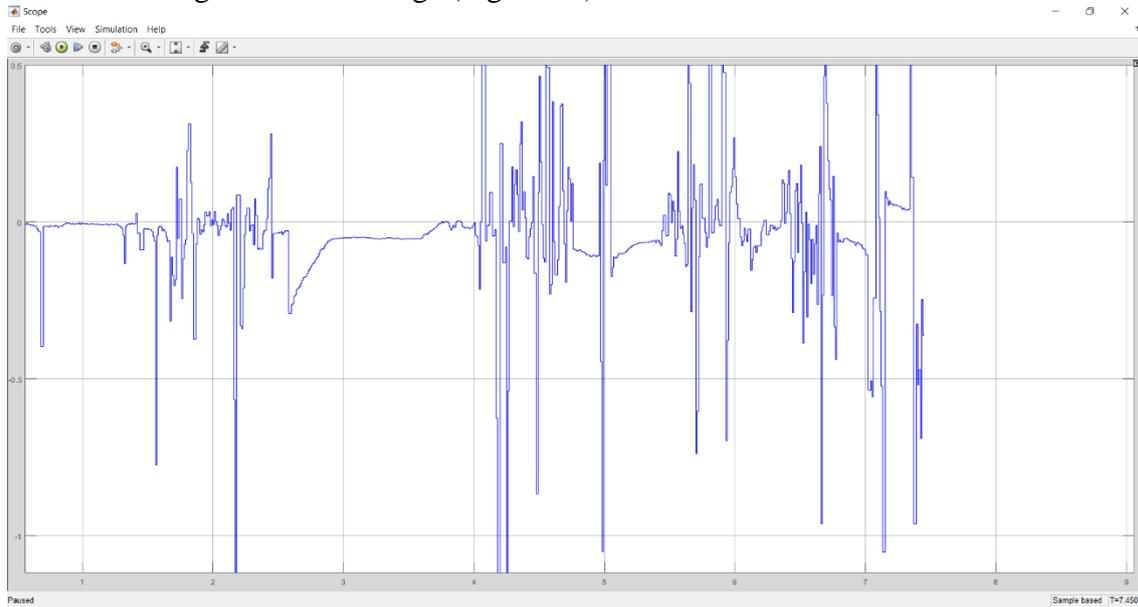


Figure 21. Readings of Roll angle sensor.

- Pitch angle sensor readings (Figure 22).

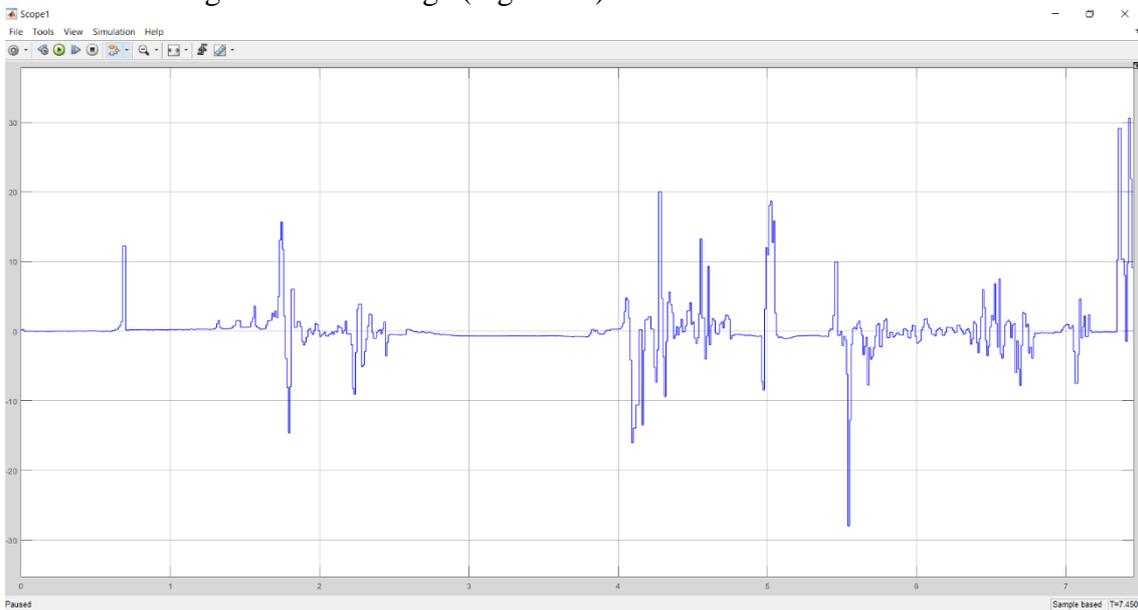


Figure 22. Readings of Pitch angle sensor.

- Yaw angle readings (Figure 23).

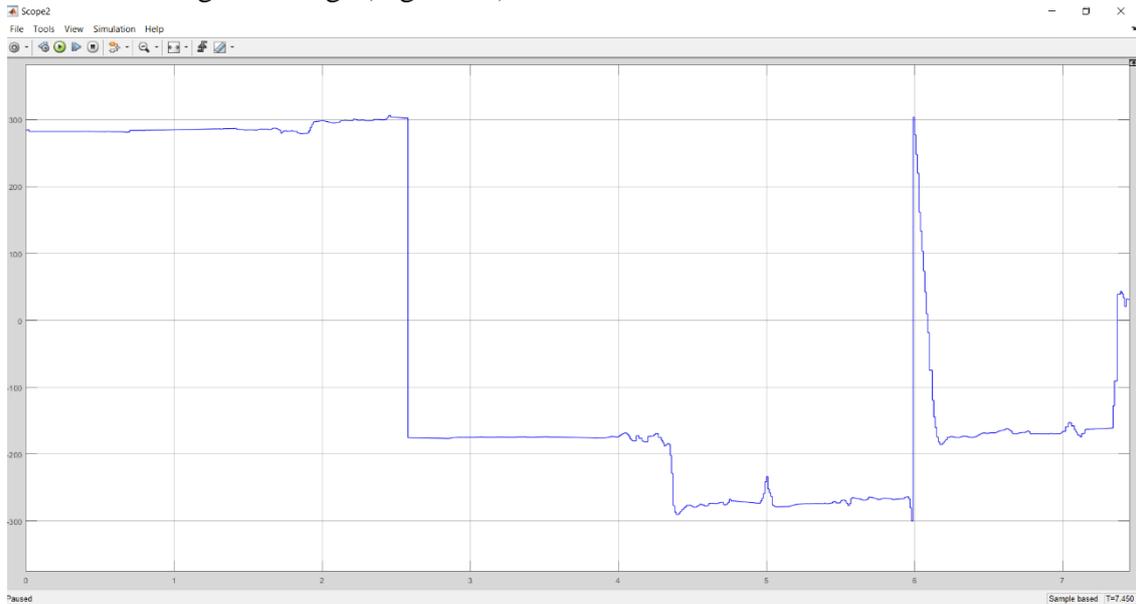


Figure 23. Readings of Yaw angle sensor.

As we can see, the results are much more satisfactory. The drone can already hold stable position and hover. Input signals still produce small vibrations and fluctuations, but the PID controller can level them out.

8 Summary

In this thesis, a quadcopter was designed and built based on a research of quadcopter's operation principles and how each separate component should work. Each component was researched in detail and composed in a single working 4-rotor drone.

Interfacing quadcopter's microcontroller with Matlab environment was the next step. This quadcopter's APM 2.8 microcontroller was connected with a WiFi module and then to a laptop's WiFi network. Sending and receiving TCP messages between the quadcopter and Matlab required message deserializing and for that purpose *aditya00j_simulink_mavlink* Matlab module was used, which compiled message deserializing sfunction for Simulink. Using this sfunction, Simulink scheme was created which initiated TCP connection, deserialized messages from the quadcopter and visualized data from drone to graphs. Mission planner software was used to modify PID controller values.

Throughout this work, 2 separate tests were conducted. In the first part of testing all PID values were set to their maximum. This test showed that the quadcopter became extremely unstable and impossible to control. In the second part of drone testing, results from first test were used to properly tune PID values in a series of experiments. As a result, more optimum PID values were discovered and the quadcopter became more stable and controllable.

Though interfacing with Matlab already brought satisfactory results in PID tuning, it can be further improved by doing more detailed testing and using different stabilization algorithms to control the drones.

Kokkuvõtte

Käesolevas töös kavandati ja ehitati nelikopter, üldiselt kopteri ja iga eraldi komponendi tööpõhimõtete uuringute najal. Iga komponenti uuriti üksikasjalikult ning selle alusel valiti sobilikud 4-rootoriga kopteri kokku ehitamiseks.

Järgmine samm oli nelikopteri mikrokontrolleri MATLAB-keskkonnale ühendamine. Kasutatud nelikopteri APM 2.8 mikrokontroller oli ühendatud WiFi mooduliga ja seejärel sülearvuti WiFi võrguga. TCP-sõnumite saatmine ja vastuvõtmine nelikopteri ja MATLABi vahel nõudis sõnumi deserialiseerimist, selleks oli kasutatud `aditya00j_simulink_mavlink` MATLAB moodul, mis koostas sõnumi deserialisatsiooni-funktsiooni Simulinkile. Kasutades seda funktsiooni, loodi Simulink skeem, mis algatas TCP-ühenduse, deserialiseeris sõnumid nelikopterilt ja visualiseeris andmeid graafiku kujul. PID kontrolleri väärtuste muutmiseks kasutati mission planner tarkvara.

Selle töö käigus viidi läbi 2 eraldi testi PID väärtuste muutumisel ning kopteri käitumise hindamisel. Esimeses testis määrati kõik PID väärtused maksimaalseteks. See test näitas, et nelikopter muutub äärmiselt ebastabiilseks ja juhtimatuks. Kopteri teises testis kasutati esimese katse tulemusi PID väärtuste õigeks häälestamiseks. Selle testi tulemusena saavutati optimaalseid PID väärtusi, kui nelikopter muutub stabiilsemaks ja kontrollitavamaks.

MATLABiga ühinemine ning optimaalsete väärtuste leidmine on juba tõhusad PID-häälestuse tulemused, kuigi seda saab veelgi täiustada, näiteks teostades täiendavaid testimised ning kasutades erinevaid stabiliseerimisalgoritme, selleks et saavutada paremad juhtimis seadistused kopterite paremaks kontrollimiseks.

References:

1. Ardupilot.org. (n.d.). *ArduPilot Open Source Autopilot*. [online] Available at: <http://ardupilot.org/> [Accessed 25 Apr. 2019].
2. Learn.parallax.com. (2018). *Understanding the Physics of Multirotor Flight | learn.parallax.com*. [online] Available at: <https://learn.parallax.com/tutorials/robot/elev-8/understanding-physics-multirotor-flight> [Accessed 13 Jun. 2018].
3. Altax US Consultancy. (2018). *Altax Consulting*. [online] Available at: <https://altax.net/> [Accessed 13 Jun. 2018].
4. Wang, D. (2015). *Quadcopter Parts: What are they and what do they do? – Quadcopter Academy*. [online] Quadcopter Academy. Available at: <http://www.quadcopteracademy.com/quadcopter-parts-what-are-they-and-what-do-they-do/> [Accessed 25 Apr. 2019].
5. Flynt, J. (2017). *21 Types of Drones: The Ultimate List of Drone Types - 3D Insider*. [online] 3D Insider. Available at: <https://3dinsider.com/types-of-drones/> [Accessed 25 Apr. 2019].
6. Drone and Quadcopter. (n.d.). *How to Build a Quadcopter - Drone and Quadcopter*. [online] Available at: <https://droneandquadcopter.com/how-to-build-quadcopter/> [Accessed 25 Apr. 2019].
7. aliexpress.com. (2019). *F14893 G DIY RC Drone Quadrocopter X4M380L Frame Kit QQ Super Motor ESC Props-in Parts & Accessories from Toys & Hobbies on Aliexpress.com | Alibaba Group*. [online] Available at: <https://tinyurl.com/diplomaframex4m380l> [Accessed 24 Apr. 2019].
8. Jefimenko, O. and Valone, T. (2011). *Electrostatic motors*. Beltsville, MD: Integrity Research Institute, pp.22-45.
9. Chaudhari, D. (2019). *9 Difference between Electric Motor and Generator [Including Block Diagram] - DipsLab.com*. [online] DipsLab.com. Available at: <https://dipslab.com/difference-electric-motor-generator/> [Accessed 24 Apr. 2019].
10. Drone Nodes. (n.d.). *Drone Motor Fundamentals – How Brushless Motor Works*. [online] Available at: <http://dronenodes.com/drone-motors-brushless-guide/> [Accessed 24 Apr. 2019].
11. Rozum Robotics. (n.d.). *Brushless AC motor vs. brushed motion technology*. [online] Available at: <https://rozum.com/brushless-motor/> [Accessed 24 Apr. 2019].
12. YouTube. (2012). *Brushless DC Motors & Control - How it Works (Part 1 of 2)*. [online] Available at: <https://www.youtube.com/watch?v=ZAY5JInyHXY&t=540s> [Accessed 13 Jun. 2018].
13. Rcgroups.com. (2005). *RC Groups - View Single Post - Calculating thrust?*. [online] Available at: <https://www.rcgroups.com/forums/showpost.php?p=4277601&postcount=2> [Accessed 13 Jun. 2018].
14. YouTube. (2015). *RC Basics - Understanding Electronic Speed Controllers (ESC)*. [online] Available at: <https://www.youtube.com/watch?v=OZNxbxL7cdc> [Accessed 13 Jun. 2018].

15. YouTube. (2017). *ESC electronic speed controller with arduino ALL EXPLAINED*. [online] Available at: <https://www.youtube.com/watch?v=8LXPcJD6hEA> [Accessed 13 Jun. 2018].
16. Nave, R. (n.d.). *Faraday's Law*. [online] Hyperphysics.phy-astr.gsu.edu. Available at: <http://hyperphysics.phy-astr.gsu.edu/hbase/electric/farlaw.html> [Accessed 8 May 2019].
17. (FAA), F. (2013). *Helicopter Flying Handbook 2012*. Chicago: Aviation Supplies & Academics, Inc., Chapter 2.
18. Droney Bee. (2018). *Quadcopter blade rotation and lift : How and why - Droney Bee*. [online] Available at: <http://www.droneybee.com/quadcopter-blade-rotation-lift/> [Accessed 13 Jun. 2018].
19. Kadamatt, V. (2017). *Choosing motors and propellers for multi-rotors - Droney Bee*. [online] Droney Bee. Available at: <http://www.droneybee.com/choosing-motors-and-propellers-for-multirotors/> [Accessed 29 Apr. 2019].
20. Oscar Liang. (2017). *How to choose LiPo Battery for Mini Quad, Drones and Quadcopters - Oscar Liang*. [online] Available at: <https://oscarliang.com/lipo-battery-guide/> [Accessed 13 Jun. 2018].
21. DroneTrest. (2015). *Power Distribution Boards - How to choose the right one*. [online] Available at: <https://www.dronetrest.com/t/power-distribution-boards-how-to-choose-the-right-one/1259> [Accessed 13 Jun. 2018].
22. Norman, A. (n.d.). *Drone Transmitter and Receiver – Radio Control System Guide*. [online] Drone Nodes. Available at: <http://dronenodes.com/drone-transmitter-receiver-fpv/> [Accessed 29 Apr. 2019].
23. (2019). *Understanding Spread Spectrum for Communications - National Instruments*. [online] Available at: <http://www.ni.com/en-us/innovations/white-papers/06/understanding-spread-spectrum-for-communications.html> [Accessed 29 Apr. 2019].
24. Roberts, R. (2012). *ABCs of Spread Spectrum - A Technology Introduction and Tutorial*. [online] Sss-mag.com. Available at: <http://sss-mag.com/ss.html> [Accessed 29 Apr. 2019].
25. DroneTrest. (2015). *RC radio control protocols explained: PWM, PPM, PCM, SBUS, IBUS DSMX, DSM2*. [online] Available at: <https://www.dronetrest.com/t/rc-radio-control-protocols-explained-pwm-ppm-pcm-sbus-ibus-dsmx-dsm2/1357> [Accessed 13 Jun. 2018].
26. Radiolink.com.cn. (n.d.). *Radiolink R8EF 2.4G 8 channels receiver - RadioLink Electronic Limited*. [online] Available at: <http://www.radiolink.com.cn/doce/product-detail-128.html> [Accessed 24 Apr. 2019].
27. Position Estimation using Inertial Measurement Unit (IMU) on a Quadcopter in an Enclosed Environment. (2016). *International Journal of Computing, Communication and Instrumentation Engineering*, 3(2), pp.332 -336.
28. Sparton. (2015). *What is an IMU?*. [online] Sparton. Available at: <https://www.spartonnavex.com/imu/> [Accessed 13 Jun. 2018].
29. DroneZon. (n.d.). *Drone Gyro Stabilization, IMU And Flight Controllers Explained*. [online] Available at: <https://www.dronezon.com/learn-about-drones-quadcopters/three-and-six-axis-gyro-stabilized-drones/> [Accessed 29 Apr. 2019].
30. Ardupilot.org. (2016). *Archived:APM 2.5 and 2.6 Overview — Plane documentation*. [online] Available at: <http://ardupilot.org/plane/docs/common-apm25-and-26-overview.html> [Accessed 13 Jun. 2018].

31. Ardupilot.org. (n.d.). *Mission Planner Overview — Mission Planner documentation*. [online] Available at: <http://ardupilot.org/planner/docs/mission-planner-overview.html> [Accessed 24 Apr. 2019].
32. Unbehauen, H. (2009). *CONTROL SYSTEMS, ROBOTICS AND AUTOMATION - Volume II: System Analysis and Control: Classical Approaches II*. Eolss Publishers, pp.58-79.
33. Electronics Hub. (2015). *PID Controller-Working and Tuning Methods*. [online] Available at: <https://www.electronicshub.org/pid-controller-working-and-tuning-methods/> [Accessed 13 Jun. 2018].
34. Urquizo, A. (2011). *PID controller overview*. [online] Commons.wikimedia.org. Available at: https://commons.wikimedia.org/wiki/File:PID_en.svg [Accessed 24 Apr. 2019].
35. Liang, O. (2019). *Quadcopter PID Explained - Oscar Liang*. [online] Oscar Liang. Available at: <https://oscarliang.com/quadcopter-pid-explained-tuning/> [Accessed 30 Apr. 2019].
36. Controlguru.com. (2015). *Integral (Reset) Windup, Jacketing Logic and the Velocity PI Form – Control Guru*. [online] Available at: <https://controlguru.com/integral-reset-windup-jacketing-logic-and-the-velocity-pi-form/> [Accessed 13 Jun. 2018].
37. El-Hussieny, H. (2016). *System Modeling and Linear Systems*. [online] Bu.edu.eg. Available at: http://bu.edu.eg/portal/uploads/Engineering,%20Shoubra/Electrical%20Engineering/823/crs-14145/Files/lecture7_SteadyStateError.pdf [Accessed 24 Apr. 2019].
38. Ardupilot.org. (2016). *Learning the ArduPilot Codebase — Dev documentation*. [online] Available at: <http://ardupilot.org/dev/docs/learning-the-ardupilot-codebase.html> [Accessed 13 Jun. 2018].

Appendix 1

```
float PID::get_pid(float error, float scaler) {
    uint32_t tnow = AP_HAL::millis();
    uint32_t dt = tnow - _last_t;
    float output = 0;
    float delta_time;
    if (_last_t == 0 || dt > 1000) {
        dt = 0;
        reset_I();
    }
    _last_t = tnow;
    delta_time = (float)dt / 1000.0f;
    // Compute proportional component
    _pid_info.P = error * _kp;
    output += _pid_info.P;
    // Compute derivative component if time has elapsed
    if ((fabsf(_kd) > 0) && (dt > 0)) {
        float derivative;
        if (isnan(_last_derivative)) {
            derivative = 0;
            _last_derivative = 0;
        } else {
            derivative = (error - _last_error) / delta_time;
        }
        // discrete low pass filter, cuts out the
        // high frequency noise that can drive the controller crazy
        float RC = 1/(2*M_PI*_fCut);
        derivative = _last_derivative +
            ((delta_time / (RC + delta_time)) *
            (derivative - _last_derivative));
        // update state
        _last_error = error;
        _last_derivative = derivative;
        // add in derivative component
        _pid_info.D = _kd * derivative;
        output += _pid_info.D;
    }
    // scale the P and D components
    output *= scaler;
    _pid_info.D *= scaler;
}
```

```

_pid_info.P *= scaler;
// Compute integral component if time has elapsed
if ((fabsf(_ki) > 0) && (dt > 0)) {
    _integrator += (error * _ki) * scaler * delta_time;
    if (_integrator < -_imax) {
        _integrator = -_imax;
    } else if (_integrator > _imax) {
        _integrator = _imax;
    }
    _pid_info.I = _integrator;
    output += _integrator;
}
_pid_info.desired = output;
return output;
}

```

Figure 24. C++ function which calculates PID controlling output signal and takes error and scaler parameters as input.

Appendix 2

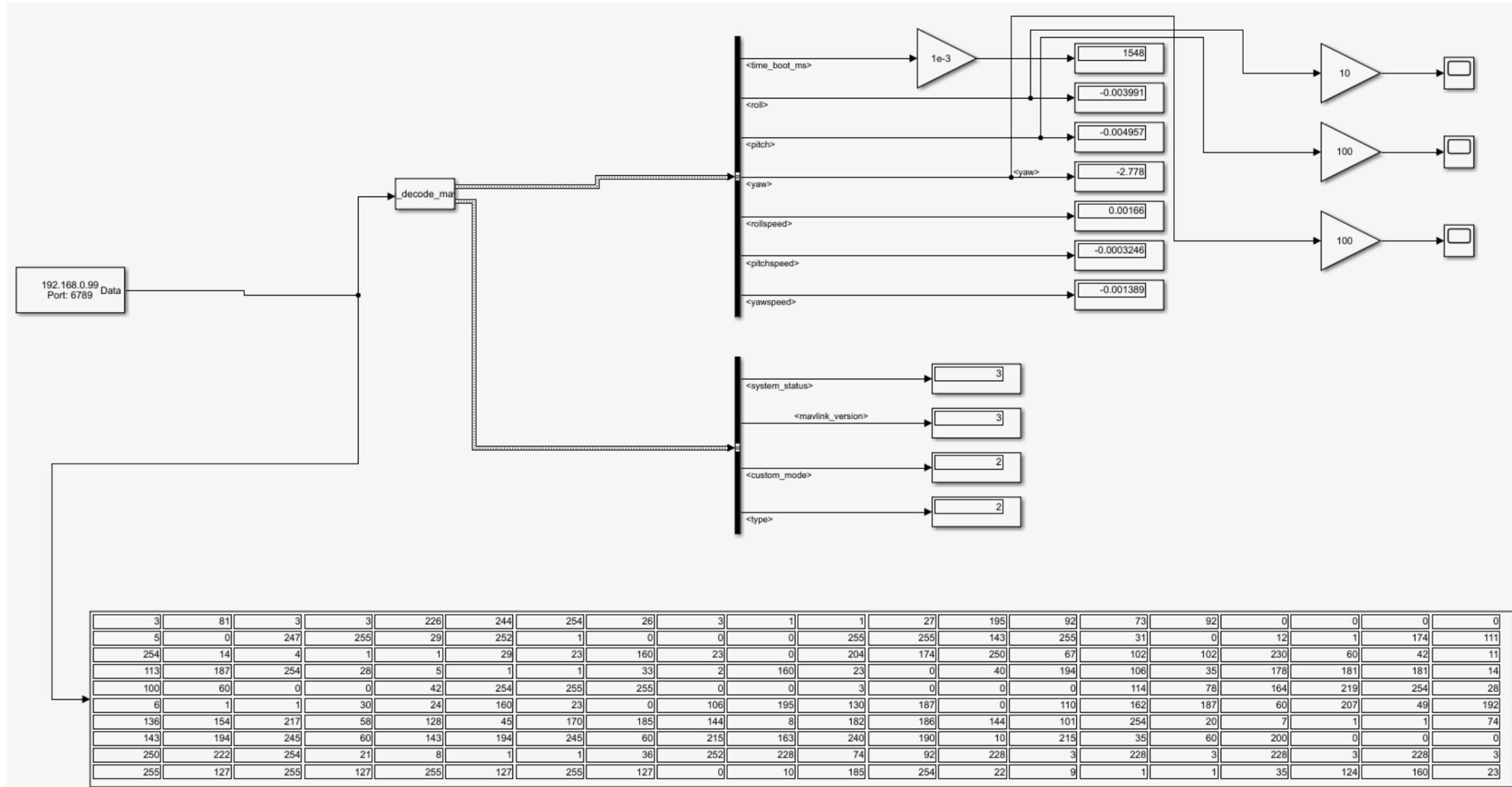


Figure 25. Simulink schematic of MAVLink messages deserializer. Data received from operating quadcopter.

Appendix 3

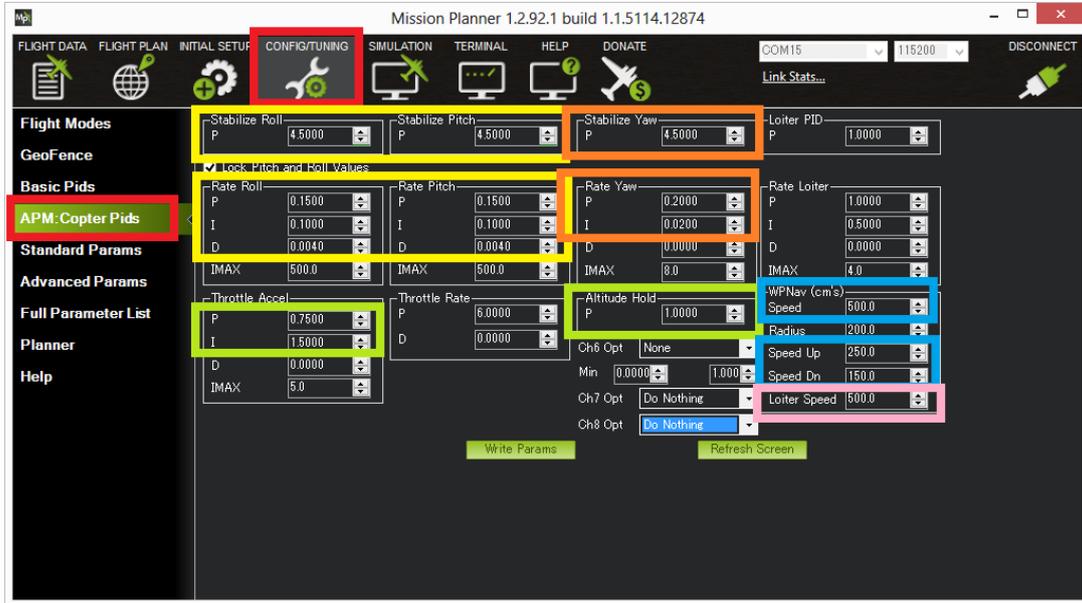


Figure 26. Mission Planner GUI [25].



Figure 27. All quadcopter PIDs' values are set to maximum [25].

Stabilize Roll (Error to Rate) P 1.800	Stabilize Pitch (Error to Rate) P 1.800	Stabilize Yaw (Error to Rate) P 1.800	Position XY (Dist to Speed) P 0.500
<input checked="" type="checkbox"/> Lock Pitch and Roll Values			
Rate Roll P 0.10048 I 0.10048 D 0.004 IMAX 100 FILT 0.000	Rate Pitch P 0.10048 I 0.10048 D 0.004 IMAX 100 FILT 0.000	Rate Yaw P 0.200 I 0.020 D 0.000 IMAX 100 FILT 0.000	Velocity XY (Vel to Accel) P 0.500 I 0.500 D 0.000 IMAX 100
Throttle Accel (Accel to motor) P 0.5765 I 1.153 D 0.000 IMAX 80	Throttle Rate (VSpd to accel) P 5.000	Altitude Hold (Alt to climbrate) P 0.500 Ch6 Opt: None Min: 0.000, 1.000 Ch7 Opt: Do Nothing Ch8 Opt: Do Nothing	WPNav (cm's) Speed 500,000 Radius 200,000 Speed Up 250,000 Speed Dn 150,000 Loiter Speed 500,000
Write Params		Refresh Screen	

Figure 28. New PID values, which are set to more optimal values based on Matlab graphs observation [25].