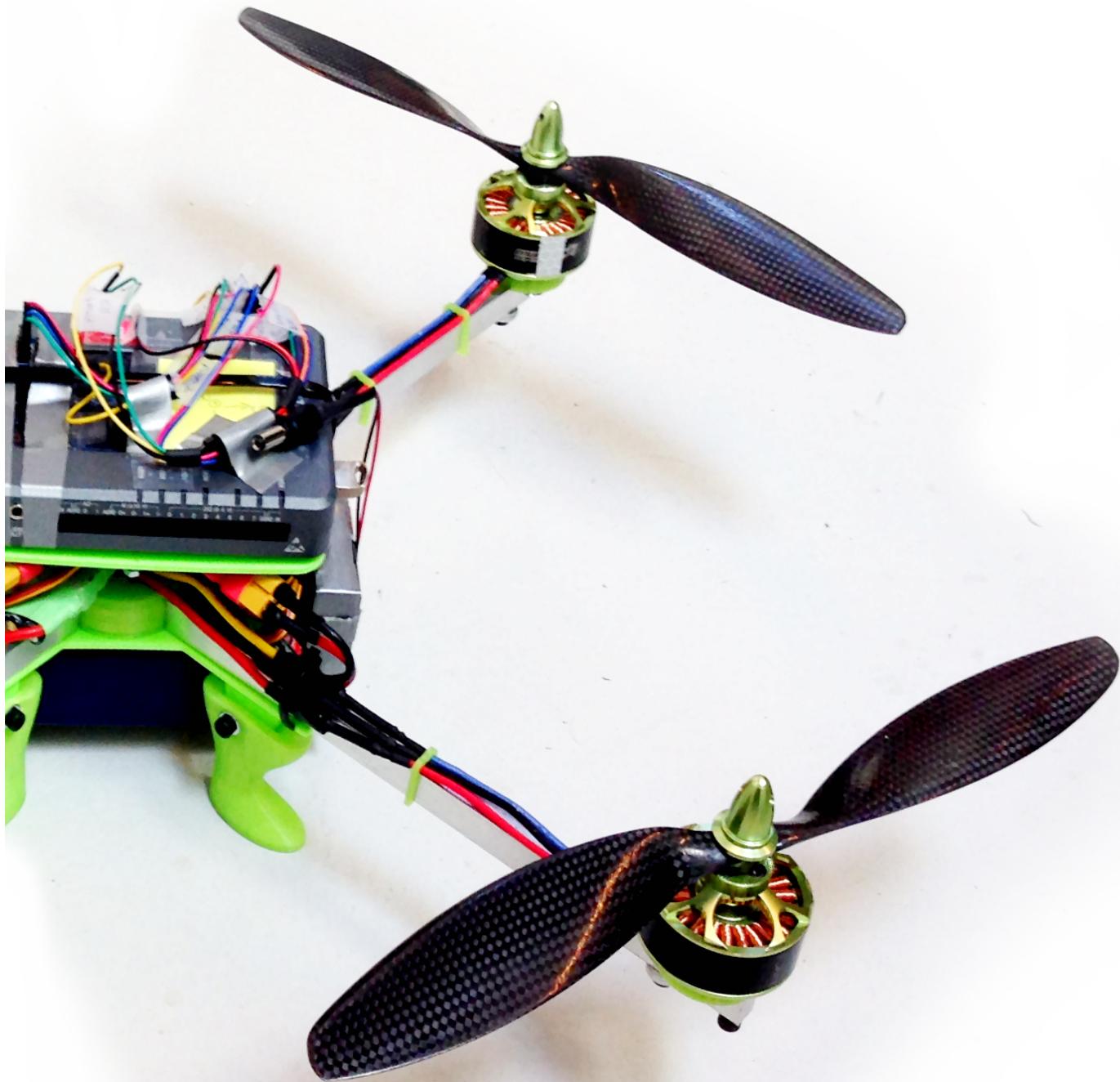


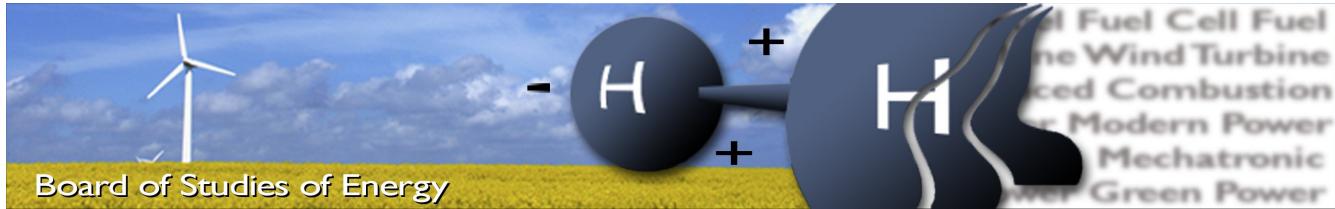
Design, Construction, and Modeling of Quadcopter

Energy, Bachelor Project 2015
MCE6-623



AALBORG UNIVERSITY
DENMARK





Title: Design, Construction, and Modeling of Quadcopter
Semester: 6th
Semester theme: Mechatronic System Design
Project period: 02.02.15 to 10.06.15
ECTS: 15
Supervisors: Henrik Clemmensen Pedersen
Per Johansen
Project group: MCE6-623

Synopsis:

The purpose of this bachelor project was to design and build a drone. Based on initial examination it was chosen to build a quadcopter. The problem statement of the project was as follows:

How can a quadcopter be designed, modeled, and optimized for energy efficient use while remaining stable?

Through an optimization process the structural design of the quadcopter was determined to meet requirements which includes a minimum lift capacity of 2 [kg] besides its own weight, and structural robustness. A National Instrument myRIO was used to control the quadcopter and an accelerometer and gyroscope was used to determine the pitch and roll of the quadcopter. The myRIO was programmed using LabVIEW.

Furthermore, a dynamic model of the quadcopter was made and implemented in MATLAB and Simulink. The model was used to simulate the behavior of the quadcopter and test a designed regulator. The regulator was designed to meet the requirement that the quadcopter should be able to hover stably.

The first experiments on the constructed quadcopter were made without the motors spinning to compare the regulator outputs from the controller and the simulated model. The experiments showed similar results, which made basis for further experiments of the quadcopter. Initial experiments with the motors spinning showed that one of the Electronic Speed Controllers was defect. The problem could not be resolved within the time frame of the project. The problem will be striven to be resolved before the project examination, where additional experimental results can be presented.

Lars Stenstrup Bruntse

Mathias Galberg

Nicklas Koldkjær Jensen

Torben Ejnar Løvbjerg Jørgensen

Copies: 7
Pages, total: 134
Appendices: 4

By signing this document, each member of the group confirms that all participated in the project work and thereby all members are collectively liable for the content of the report.
Furthermore every group member acknowledges that there is no plagiarism in the report.

PREFACE

This report is produced at *Aalborg University, Department of Energy Technology*, for the bachelor project during *6th semester* of the *Mechatronic Control Engineering* education. The report is a product of the semester theme *Mechatronic System Design*.

We would like to recognize Simon Lennart Sahlin and 3D Eksperten for providing a 3D printer and help setting up. We would also like to recognize UASworks for providing components used to characterize motor and propeller.

READING GUIDE

In the beginning of the report an abbreviation and a nomenclature list is found of the different terms used throughout the report.

The Vancouver referencing system is used for references in the report, where a number in square brackets refers to the bibliography found at the end of the report. Furthermore, internal references to appendices, chapters, and sections are used and are stated in the text by numbers or letters. Figures, tables, and equations are labeled by a number or letter.

Attached is a CD-ROM, which includes data from experiments, MATLAB scripts, Simulink models, and LabVIEW files made for the project, as well as guides of how to run the specific files. In addition the report can be found in digital form for closer inspection of figures.

At the end of the report are the appendices, which supports work done in the project but is not directly relevant to the project.

Lars Stenstrup Bruntse

Mathias Galberg

Nicklas Koldkjær Jensen

Torben Ejnar Løvbjerg Jørgensen

Nomenclature

Nomenclature	Unit	Description
B_m	[kg m s ⁻³]	Mechanical damping in the motor
F_d	[N]	Drag force
F_g	[N]	Gravitational force
F_T	[N]	Thrust force
i_m	[A]	Current through motor
J_m, J_p	[kg m ²]	Mass inertia of motor and propeller
L_m	[H]	Inductance through motor
R_m	[Ω]	Electrical resistance through motor wire
S	[m]	Global position
v_m	[V]	Voltage across motor
ϕ, θ, ψ	[rad]	Elements of the Euler angles χ
ρ	[kg m ⁻³]	Mass density of material
ν	[rad]	Local angular velocity
χ	[rad]	Euler angles
B_{lin}	-	Linear drag force coefficient
B_{ang}	-	Angular drag torque coefficient
ω_n	[rad s ⁻¹]	Angular Velocity of Propeller n
m	[kg]	Quadcopter Mass
$\tau_{d,n}$	[N m]	Propeller drag torque
τ_{gyro}	[N m]	Gyroscopic motion torque
$\underline{\underline{J}}_{pr}$	-	Propeller inertia matrix
$J_{x,pr}, J_{y,pr}, J_{z,pr}$	-	Propeller inertia around each axis
$\tau_{p,n}$	[N m]	Total propeller torque on propeller center
τ_{CG}	[N m]	Total torque on CG
$\underline{\underline{J}}$	-	Quadcopter inertia matrix
J_x, J_y, J_z	-	Quadcopter inertia around each axis
τ_m	[s]	Motor settling time
ω_{ref}	[rad s ⁻¹]	Reference propeller angular velocity
K_v	[RPM/V]	Motor constant
$F_{l,n}$	[N]	Lift force of propeller n
k_l	-	Lift force coefficient
D_c	-	Duty Cycle
ϕ_{accel}	[rad]	Approx. angle by accelerometer
$\dot{\phi}_{gyro}$	[rad s ⁻¹]	Quad angular velocity from gyroscope
x, y, z	[m]	Linear position
x_L, y_L, z_L	-	Local frame xyz axis
$\underline{\underline{R}}$	-	Rotation Matrix
$\underline{\underline{Q}}$	-	Euler rate transformation matrix
p, q, r	[rad s ⁻¹]	Local frame angular velocity
$F_{l,L}$	[N]	Local total lift
$F_{l,G}$	[N]	Global total lift
ω_{ss}	[rad s ⁻¹]	Steady State propeller angular velocity
K_{DC}	-	Duty cycle transfer function gain
L_n	[m]	Length vector from CG to propeller center

Subscripts

Subscript	Description
d	Drag
G	Global frame
g	Gravitational
i	Typical index of summation
L	Local frame
l	Lift
m	Motor
p	Propeller
q	Quadcopter

Abbreviation

Abbreviation	Description
BLDC	Brushless DC
DC	Direct Current
CG	Center of Gravity
UAV	Unmanned Aerial Vehicle
ASME	American Society of Mechanical Engineers
SubVI	Sub Virtuel Instrument (LabVIEW subsystem)
GPS	Global Positioning System
ESC	Electronic Speed Control
RPM	Revolutions pr. Minute
PPM	Pulse Position Modulation
PWM	Pulse Width Modulation
RC	Radio Control
IMU	Inertial Measurement Unit
LiPo	Lithium Polymer
NiMH	Nickel-Metal Hydride
NI	National Instruments
USB	Universal Serial Bus
I2C	Inter-Integrated Circuit
SPI	Peripheral Interface-bus
PID	Proportional-Integral-Derivative
SCL	Serial Clock Line
SDA	Serial Data Line
Pm	Phase margin
Gm	Gain margin
DE	Differential Evolution
CFD	Computation Fluid Dynamics
FEM	Finite Element Methods
SI	System International
CAD	Computer aided design

Table of Contents

Preface	v
Nomenclature	viii
Abstract	xv
1 Introduction	1
2 Problem Analysis	3
2.1 Structure	3
2.2 Components	4
2.3 Motion of the Quadcopter	8
2.4 Conclusion	9
3 Problem Specification	11
3.1 Problem Statement	11
3.2 Problem Delimitation	11
4 Mechatronic System Design	13
4.1 Requirements	13
4.2 Structure	13
4.3 Electrical Components	14
4.4 Characteristics	20
4.5 Implementation	21
5 System Modeling	35
5.1 Equations of Motion	36
5.2 Motor Dynamics	44
5.3 Conclusion	48
6 Control System	49
6.1 Simplifications	50
6.2 Linearization	53
6.3 Decoupling	55
6.4 Regulator Design	57
6.5 Simulation of Controllers	63
7 Regulator Optimization	67
7.1 Objective Function	67
7.2 Constraints	67
7.3 Solving Method	68
7.4 Optimized Regulator Response	69
8 Experiment	71
8.1 Purpose	71

TABLE OF CONTENTS

8.2 Setup	71
8.3 Data Processing	72
9 Conclusion	75
Bibliography	78
Appendix	
A Lighter than Air	81
B Lift and Drag Estimations	85
C Design of Quadcopter	89
D Thrust and Inertia Characteristics	103

List of Figures

2.1	Different drone types	3
2.2	Pictures of BLDC motor and a dismantled BLDC motor	4
2.3	Propellers used for multicopter drones	5
2.4	Turnigy ESC for drones	6
2.5	Illustration of pitch, roll, and yaw angles.	8
2.6	Different motor speeds for roll, pitch, and yaw.	9
4.1	Connection diagram of the quadcopter	16
4.2	PmodGYRO 3-Axis Digital Gyroscope	17
4.3	Overview of pin connection between the PmodGYRO and myRIO.	17
4.4	Turnigy 30 Amp ESC	18
4.5	Turnigy BLDC motor	18
4.6	Carbon fiber propeller	19
4.7	Turnigy nano-tech LiPo battery	19
4.8	Picture of the Anself T Plug Power Distribution Board.	20
4.9	Diagram of the essential parts in the LabVIEW framework.	21
4.10	Main LabVIEW VI	22
4.11	Illustrations of single axis sensing of the accelerometer.	23
4.12	Pitch and roll angle determined by accelerometer	23
4.13	Implementation of the accelerometer in the LabVIEW program.	24
4.14	Implementation of the approximation of pitch and roll using the accelerometer.	25
4.15	Implementation of the gyroscope in the LabVIEW program.	26
4.16	Illustration of the complementary filter.	27
4.17	Illustration of the implemented complementary filter.	28
4.18	Implementation of the filter in LabVIEW.	29
4.19	Filter coefficients of 0.2, 0.5, and 0.8	30
4.20	Filter coefficients of 0.7, 0.8, and 0.9	30
4.21	Implementation of the control in the LabVIEW programming.	31
4.22	Implementation of PWM signals for the ESCs in the main block diagram.	32
4.23	Implementation of data logging in the main block diagram.	33
5.1	Quadcopter model structure	35
5.2	The local frame (in blue) is fixed to the quadcopter and moves around in the global frame.	36
5.3	The angles ϕ , θ , and ψ defines pitch, roll, and yaw between global and local frame.	37
5.4	Two dimensional frame rotation by angle ψ (yaw) and the equivalent vector rotation.	38
5.5	Angular velocity to Euler Rates in Simulink based on Equation 5.9	39
5.6	The acting forces on the quadcopter.	39
5.7	Force system in Simulink	41
5.8	Non-uniform geometry with four applied forces	42
5.9	Torque estimation in Simulink	44
5.10	Rotor angular velocity from duty cycle steps.	45
5.11	Zoom of the highlighted steps.	45
5.12	Step on the motor compared to step on the approximated first order transfer function $G_\omega(s)$	47
6.1	Control System.	49

6.2	Block diagram representing the pitch state.	53
6.3	The control loop for pitch.	58
6.4	The simplified control loop for pitch.	59
6.5	Bode diagram of the open loop pitch system.	60
6.6	Bode diagram of the open loop yaw system.	62
6.7	Bode diagram of the open loop altitude control system.	63
6.8	Individual simulations for a step on ϕ , θ , ψ , and z .	64
6.9	Angular velocity on all motors for each step	65
6.10	Altitude and angular position response when subjected to disturbances. Note that the angular position is multiplied by 2, for viewing purposes.	66
7.1	Optimized controller response	69
7.2	Yaw response	70
8.1	Data collected for comparison of regulator outputs between the Simulink model and the LabVIEW program, when applying a pitch angle to the quadcopter. Blue line = Simulink model, and Red line = LabVIEW program.	72
8.2	Data collected for comparison of regulator outputs between the Simulink model and the LabVIEW program, when applying a roll angle to the quadcopter. Blue line = Simulink model, and Red line = LabVIEW program.	73
8.3	Comparison between pitch and roll angles	73
8.4	Comparison between motor 3 and motor 4 without propeller. Motor 4 behaves as motor 1 and motor 2 while motor 3 behaves irregularly	74
B.1	The propeller blade with an illustrated segment.	86
B.2	Drag Coefficient as a function of angle of attack for the NACA 0012 airfoil. [3]	87
B.3	Lift Coefficient as a function of angle of attack for the NACA 0012 airfoil. [3]	87
C.1	Blade clearance constraint for initial optimization	91
C.2	Cross sectional geometries considered for optimization	92
C.3	Moment acting upon the arm	92
C.4	Stress distribution inside profile	92
C.5	Cantilever setup for determining the natural frequency of the entire system	93
C.6	9 points randomly distributed within \underline{L} and \underline{U}	95
C.7	Flowchart of one generation optimization using DE	96
C.8	Population and mutant constraint violation with 2 different mutants (v_1, v_2) which can be compared to the member of the population (x)	97
C.9	The three different types of constrain violations being added to the object function f	98
C.10	Effect of scaling r when using internal penalty	99
C.11	Effect of scaling r and n when using external penalty	100
D.1	Experimental setup to determine the thrust and angular velocity of the motor and propeller.	105
D.2	Effect of Savitzky-Golay filter on the thrust when a constant duty cycle is applied. The data is a sample of the entire dataset during one experiment cycle	106
D.3	Thrust exerted by the propeller as a function of the angular velocity of the propeller. At around 5000-6000 [RPM] the setup made significantly more noise than at lower and higher angular velocities which may have caused the spike on the 14.8 [V] and 15.8 [V].	107
D.4	Collected data of angular velocity [RPM] and duty cycle [-] with different applied voltages along with the linear fit for each voltage	108
D.5	Experimental setup for test of moment of inertia around the revolving axis of an object.	109
D.6	Bifilar pendulum experiments. Due to the perspective of the photographs the strings look like they are not parallel, however, they are parallel when at rest.	110
D.7	Overview of system when at rest and displaced	111
D.8	Forces acting on the object and resulting reaction	111
D.9	ZL plane sketch of F_g and F_s	112

D.10 Varying the inertia of the object within the range at 20 different points	113
D.11 Varying the radius of the object within the range at 20 different points	114
D.12 Varying the mass of the object within the range at 20 different points	114
D.13 Varying the gravitational pull within the range at 20 different points	115
D.14 Varying the length of the string within the range at 20 different points	115
D.15 Varying the starting angle within the range at 20 different points	116

List of Tables

4.1	Design variables found using optimization	14
4.2	Design variables for the constructed quadcopter	14
4.3	myRIO specifications	15
4.4	Overview of pins used on the NI myRIO.	15
4.5	PmodGYRO Specifications	17
6.1	Table showing effect increasing constants	57
6.2	PID controller constants	63
7.1	Table of the optimized controller constants	69
9.1	PID controller constants	75
9.2	Table of the optimized controller constants	76
C.1	Cross sectional properties of arm profiles	92
D.1	Constants and ranges for simulation variables	113

ABSTRACT

The focus in this project was to design and construct a drone. It was needed to examine the necessary aspects to consider when building a drone. Based on the examination it was chosen to design and construct a quadcopter in an X-formation. The project statement of the project was as follows:

How can a quadcopter be designed, modeled, and optimized for energy efficient use while remaining stable?

Additionally the quadcopter was to fulfill requirements, which includes a 2 [kg] lift capacity beside the capacity needed to lift its own weight, the structure of the quadcopter should be robust enough to handle small crashes, and the quadcopter should be able to hover stably.

An optimization on the mechatronical design was made on the frame structure and the components needed to meet the requirements, meaning the components needed should have a minimum power consumption while still being able to meet the requirements, and the frame structure should not be over dimensioned which would add unnecessary weight. Based on the results from the optimization the frame of the quadcopter was fabricated using aluminum and 3D printed parts, and the electrical components was obtained. A National Instruments myRIO, provided by Aalborg University, was implemented as the controller of the quadcopter and handled sensor inputs, calculations, and signal outputs to be able to control the quadcopter. The sensors used included the on-board three-axis accelerometer in the myRIO, and an external three-axis gyroscope. The myRIO was programmed using LabVIEW.

Furthermore, a dynamic model of the quadcopter was made by reviewing and examining relevant theory regarding the motion of the quadcopter, rotation between frames, and the dynamics of the rotors and propellers. Expressions used to described the quadcopter globally and locally was derived, together with forces and torques acting on the quadcopter. The model was implemented in MATLAB and Simulink, which made it possible to simulate and analyze the behavior of the quadcopter. Regulators were designed and optimized using the open-loop transfer function for the system and tested in the MATLAB and Simulink model. The regulators were designed to meet the requirement of stable hovering of the quadcopter.

Experiments were conducted to analyze the performance of the constructed quadcopter. Initially tests were made to examine the regulator output of the controller, without the motors spinning, and compared to the model. The tests showed similar responses which made the basis for more comprehensive tests. When experiments were made with the motors spinning, it was discovered that one of the Electronic Speed Controls was defect, meaning no further experiments were conducted. The problem was not able to be resolved within the time frame of the project. It is striven to resolve the problem before the examination. If the problem is resolved the quadcopter can be fully tested and new experimental data can be presented.

CHAPTER 1

INTRODUCTION

In recent years there has been a rising demand and popularity of Unmanned Aerial Vehicles (UAVs), also commonly known as drones, in different areas e.g. hobby, military, and industry. The military uses them for surveillance, search and rescue, and for specific warfare in active war zones. Amazon has been investigating the usage of quadcopters as a package delivery service[6]. Furthermore, fire departments could use drones to spot potential survivors or observe the fire and provide information about how to fight the fire effectively. Part of the reason for this increase in popularity is the progress in electronic stabilization and control, since a drone is near impossible to control without it. The electronic control and electrical motors have gotten smaller and more efficient, making them more suitable for civil and commercial drones. This makes the drones more practical to use while being economically viable. [7]

With the increasing popularity of drones both for industry and hobby purposes, additional guidelines and regulations have been made for the use of drones. The use of drones in Denmark is regulated by *Trafikstyrelsen*, and restrictions submitted includes e.g. minimum flight distances to public areas and buildings, military areas, and areas of accidents [8].

An example of a drone design is the quadcopter, which is a multicopter with four equally spaced propellers usually in a cross (X) or plus (+) formation. Each propeller is driven by an electrical motor and is fixed-pitched, meaning that the angle of the blades cannot be changed, as opposed to a regular helicopter. A helicopter uses a swashplate to adjust the rotor and blades to maneuver around. A helicopter then needs a small side rotor which produces no upwards thrust but counteracts the torque produced by the main rotor. The quadcopter maneuvers in the air by adjusting the speed on the four motors individually, where two of the propellers spin clockwise and the other two spin counterclockwise, preventing it from pirouetting. [9]

The purpose of this project is to design and build a drone, which includes analysis of the motion of the drone, determination of structural design, and choice of components. As inspiration, the American Society of Mechanical Engineers (ASME) hosts a student design contest, where the task for 2014 edition was to design an UAV, which fulfilled the requirements. The requirements includes that the UAV

- must be able to navigate through gates on a course
- must be able to carry cargo through a course
- must be designed and assembled by the participants

Further information about the ASME competition can be found in Appendix A [10].

Based on the ASME requirements research about drones is necessary before the design process can begin. To help the proceedings the following initial problem is made

What must be considered when designing a drone?

The research on the initial problem is described in Chapter 2.

CHAPTER 2

PROBLEM ANALYSIS

In this chapter, the initial problem which will be analyzed and lead to the problem specification, where a problem statement and delimitations will be presented.

When designing a drone the purpose of the drone has to be taken into account. Drones used for hobby purposes with focus on the possibility of acrobatic flying will typically be small and light weight, and have smaller propellers with motors that operates well at high angular velocities. Another use for drones could be for surveillance and in this case bigger propellers, lift capacity, stability, and more flight time is desirable. This will make the drones able to maneuver steadily in the air with e.g. a camera, which could be useful for military use, where a drone would be able to give precise information about essential locations without exposing anyone to potential harm.

Another important aspect to take into account, when designing a drone, is the price. The demand and purpose of the drone must be specified beforehand to be able to design, optimize, and find the parts that suits demands of drone at the most favorable price possible.

SECTION 2.1

STRUCTURE

Drones can have different kind of base structures, depending on their purpose. A typical base form for drones used for hobby and commercial usage consist of a frame, with a number of arms mounted on it. The center frame holds most, if not all, the electronic components except the motors, which arm mounted on at the end of the arms. Examples of drone design types includes quadcopters (4 propellers), hexacopters (6 propellers), and octacopters (8 propellers).



Figure 2.1: Examples of drone design types. Quadcopter (left) [11], Hexacopter (middle) [12], and Octacopter (right) [13].

Typical materials used for building the frame are carbon fiber, aluminum, plastic, and wood varieties such as plywood. Carbon fiber is the most expensive but vibration absorbent and rigid [14]. Aluminum is light weight, rigid and affordable, but the damping effect is not as good as with the carbon fiber. Plastic is a popular material to use for the protection of the components as it is light weight and fairly rigid. Further on, wood is a cheap material to use and good at absorbing vibrations, but it is not as rigid and can thereby break more easily if the drone crashes [14]. Therefore, when choosing the material for the frame, it is important to focus on the weight, robustness, ability to damp vibrations, and costs to get the optimal solution according to ones preferences.

SECTION 2.2**COMPONENTS**

There are some essential components needed in addition to the frame when building a drone. Moreover it is possible to implement additional accessories depending on the purpose for the drone, e.g. a camera, a LED kit, and a Global Position System (GPS) transmitter. An overview of the essential components are listed below and followed by a description of them. [15]

- Motors
- Propellers
- Electronic Speed Controllers (ESC)
- Sensors
- Flight and electronics controller
- Battery
- Additional accessories

MOTOR

Regular Direct Current (DC) motors or a Brushless Direct Current (BLDC) motors are often chosen when building a drone [16]. BLDC motors are similar to regular DC motors in the way of driving the shaft using magnets and coils. A constructional example of a BLDC motor includes a fixed number of coils mounted on the stator of the motor, while a number of permanent magnets are mounted on the rotor, where the driving shaft is attached. Because the coils are fixed, there is no need for brushes, as the wires delivering power to the motor can go directly to the coils. Regular DC motors are inexpensive, but require periodic maintenance to clean the brushes. Furthermore, at higher speeds the brush friction increases which decreases efficiency of the motor. [17]



Figure 2.2: An example of a BLDC motor for a small drone, and an open motor which shows the coils and the magnets.

The BLDC motor generally uses less power than a DC motor at same speeds, and has a better speed/torque trade-off [17]. Furthermore, the BLDC motor is more expensive than the regular DC motor, but it is typically more energy efficient, as it does not have any friction from the brushes moving. Since the brushes on a regular DC motor is

often the first thing to wear out, the BLDC motor generally has a longer lifespan.

Since BLDC motors exist in many different variants, it is important to consider the size and the power consumption of the motors. The motors have to match the kind of propeller, which is going to be used, and be able to deliver the needed power to make the propellers spin and achieve the desired thrust. One specification in particular is the K_v -rating, which indicates how many Revolutions Per Minutes (RPMs) the motor will do as a function of applied voltage. It does not take any loads into account, which is applied when attaching a propeller.

PROPELLER

Propellers come in many variations, where the diameter, pitch, or number of blades differ. The pitch is of unit length per revolution, and is a measure of how far the propeller would go during one revolution. Depending on the rotational direction of the motor, propellers are made with different directions of tilt angle so the thrust is generated in the desired direction.



Figure 2.3: An example of a set of propellers used for drones.

Considerations about the size of the propellers and which motors are going to be used are necessary to make. Generally, propellers with larger diameter and pitch generates more thrust, but more power is required to be able to drive the propellers [18]. Therefore, smaller propellers are preferable when using a motor driven at high speed, and bigger propellers are used when using a motor driven at lower speeds. It is important to find the right balance between the motor and propeller, to ensure that the drone is able to lift its own weight and maneuver in the air, and possibly lift additional loads. The dimensions of the propellers could also influence the size of the drone frame, since the propellers must have space to spin around unaffected.

ESC

BLDC motors are multi-phased, often 3 phases, which means the motors will not work with direct current. To use the motors an ESC can be used, which makes three different signals with equally shifted phases, that will make the motor spin. The ESC has a battery as power input as well as a Pulse-Position Modulation (PPM) signal, which gives a three phase output to the motor. A PPM signal is similar to a Pulse Width Modulation (PWM) signal, and a typical pulse high-time for Radio Controlled (RC) electronics is in the range of 1-2 [ms] [19]. The frequency of the signal varies between 200 [Hz] and 8 [kHz].



Figure 2.4: An example of an ESC made by Turnigy to drones for hobby usage. [20]

Many different ESCs are available on the market, but the most important factor to consider, when choosing the ESC is its ability to handle the maximum current drawn by the motor.

SENSORS

Sensors are needed to be able to stabilize and control the drone. Measurements of translational accelerations and angular velocities is often used to control the drone, where an accelerometer and a gyroscope can be used respectively. Additionally sensors can be used to determine the orientation of the drone, such as a magnetometer, Global Positioning System (GPS), barometer, and lasers.

Inertial Measurement Units (IMUs) are often used, which are electrical devices consisting of multiple sensors which for example make measurements to determine gravitational forces, velocities, and orientation of a drone. Types of IMUs used for small drones for hobby usage typically includes an accelerometer, a gyroscope, and a magnetometer. [21]

Different types of accelerometers can be found depending on the size, accuracy, and number of sensing axes. A three-axis accelerometer is often preferable to use for drones to be able to approximate an accurate orientation. An accelerometer senses, on each axis, the projection of the gravitational vector often in the gravitational acceleration unit [g]. It is sensitive to vibrations on the drone are big the data measured becomes distorted and the drone could become unstable. Therefore accelerometers are often used in combination with other sensors e.g. a gyroscope.

A gyroscope measures angular velocities, and depending on type and size they can be found in a wide range. It is preferable to use three-axis gyroscope to be able to approximate full orientation of the drone. The measured angular velocities is often integrated to obtain an angular displacement, which means it possible to get an accumulated error from integration and therefore an inaccurate measure of angle.

In addition, a magnetometer could be used, which measures the direction and strength of a magnetic field. It could be used to determine the direction of Earth's magnetic poles and thereby determine in which direction the front of the drone is facing.

The possibilities to determine the orientation of drones are wide and it is therefore necessary to assess the need of

sensors and how they cooperate with each other depending on the application of the drone.

CONTROLLER

One of the problems that needs to be considered when designing a drone is the ability to keep it stable during flight. To achieve this a controller has to be programmed and fitted onto the drone. The controller is the brain of the drone, and it is possible to find different kinds of controller boards on the market specifically made for drone use, where some have IMU sensors integrated on the board. [22]

When choosing a controller it is important to consider if it can handle a high enough sampling frequency from the sensor measurements and deliver a high enough signal frequency to the ESCs, while being able to process the data and calculate new outputs. The controller has to regulate and compensate for errors between the wanted and the current angles of the drone by increasing or decreasing the angular velocity of the motors individually.

Other features, which could be implemented when designing a drone, includes e.g. a return home function or a fail-safe/off switch. If the drone flies out of line of sight or too far away from the operator, or if the drone battery reaches a critical voltage level a build in return home function would be useful. If something on the drone should malfunction a fail-safe/off switch is good to have implemented for safety reasons.

Typically a RC transmitter is implemented, which is used to control the drone remotely. The RC receiver receives a signal transmitted by the transmitter equipped on a remote controller.

BATTERY

The battery is the power source of the drone. Lithium Polymer (LiPo) batteries are often used but other types e.g. Nickel-metal hydride (NiMH) batteries can be used. It is possible to find LiPo batteries in single cell, with a voltage at 3.7 [V], up to packs of over 10 cells connected in series [23]. When choosing the battery, it is important to consider how much power the motors will draw, how long flight time is needed, and the weight of the battery.

A parameter that is helpful when choosing a battery, is the C-value, which is the discharge rate. The C-value can be used to indicate how much current can be drawn from the battery with a given capacity by using Equation 2.1.

$$\text{Max Current} = \text{Discharge Rate} \cdot \text{Battery Capacity} \quad [\text{A}] \quad (2.1)$$

A big battery would generally increase the possible flight duration, but it will also add weight to the drone which will affect the power consumption while flying. Furthermore, the battery has to be able to deliver the max current needed for the motors to ensure the motors deliver the wanted rotational speed.

2.2.1 ACCESSORIES

Depending on the purpose of the drone additional accessories can be implemented. It is popular to implement cameras on drone, which transmits a live feed back to the operator of the drone, which makes it possible to get an orientation of the location the drone is operated in. Furthermore it can be used to make videos and take pictures in accordance the demand of the user.

2.2.2 PARTIAL CONCLUSION

The purpose and demand of the drone has to be analyzed when designing the drone. Drones can be constructed in different sizes and designs, which means the requirements has to be taken into account if an adequate drone, to a favorable price, is wanted. The different components has to be selected thoughtfully so they interact in an proper manner to meet the demands of the drone.

Based on the findings regarding the many different design possibilities of drones and all the consideration needed to be taken into account when designing the whole system, the further work in the project will be based on building a drone with the base structure of a quadcopter. It has to be considered how the drone is controlled and kept stabilized during flight, and it is therefore necessary to analyze the motion of the quadcopter.

SECTION 2.3

MOTION OF THE QUADCOPTER

To be able to describe the motion of the quadcopter in three-dimensional space, terms for the movement around the three axis of motion are used, where the orientation of the quadcopter is determined. For this project, the front of the quadcopter is defined to be in the positive y-direction, while the upwards change in altitude is defined in z-direction. As seen on Figure 2.5, pitch describes the motion around the x-axis, roll describes the motion around the y-axis, and yaw describes the motion around the z-axis.

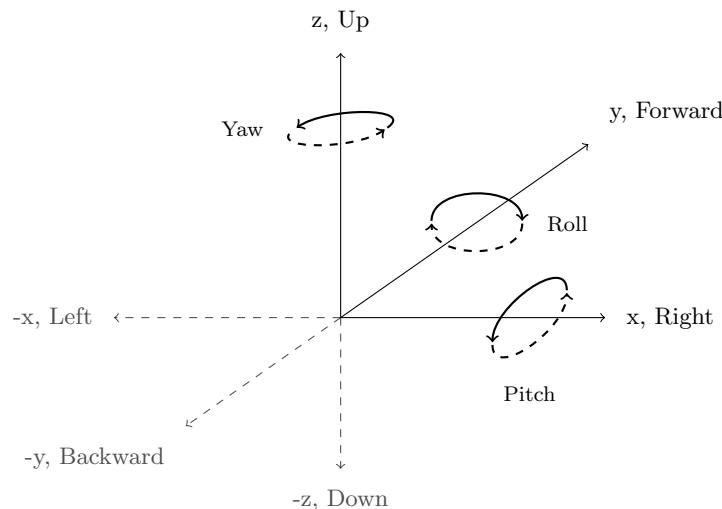


Figure 2.5: Illustration of pitch, roll, and yaw angles.

When combining the three degrees of motion together with the possibility to change the altitude in z-direction, depending on applied thrust, the quadcopter has the ability to move in all directions in three-dimensional space.

When designing a quadcopter, the rotational direction of each motor has to be chosen. Diagonal motors has to spin in the same direction as that which is seen in Figure 2.6, while the remaining motors has to spin in the opposite direction. Additionally, the base structure of the quadcopter has to be determined. Often a symmetrical base structure is used, where the motors and arms are either placed in an (X) formation or (+) formation. The motion of the

quadcopter is controlled by changing the speed of the rotors according to the desired motion. Generally, there are two ways to control the quadcopter:[14]

1. Increase the angular speed of those motors placed in such a manner, they will decrease the error measured
2. Increasing and decreasing all four motors accordingly to the desired effect

Analyzing the last option shows that a roll motion is made by increasing the speed of motor 1 and 4 and decreasing the speed of motors 2 and 3, which is illustrated on Figure 2.6 (A). This configuration will result in the quadcopter rotating around the y-axis and move in the x-direction. In a similar way, a movement and rotation in the opposite direction can be achieved. Illustrated on Figure 2.6 (B), a pitch motion in the y-direction is achieved by decreasing the speed of motors 1 and 2 and increasing the speed of motors 3 and 4.

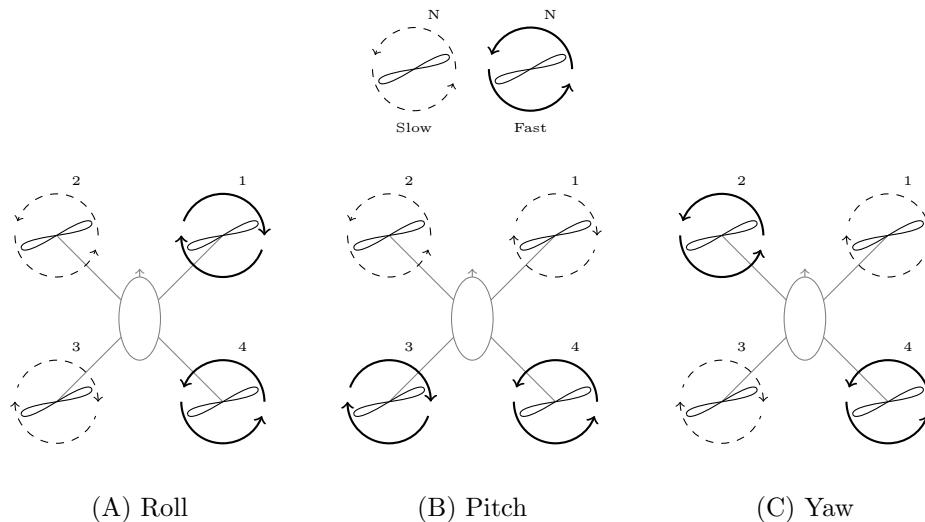


Figure 2.6: Different motor speeds for roll, pitch, and yaw.

As seen on Figure 2.6 (C), a yaw motion is achieved by increasing the speed of opposite motors and decreasing the speed of the remaining two motors. In the figure it is motor 2 and 4 where the speed has been increased, this results in the quadcopter turning counter clockwise in the x,y-plan depending on the chosen spinning direction of each propeller. The quadcopter turns clockwise if the speed of motor 1 and 3 had increased.

SECTION 2.4

CONCLUSION

A quadcopter structure has been selected for the base design of the drone, which is to be build. The quadcopter will be designed in a (X) formation and option (2) is going to be used. To aid the design process a dynamic model of the quadcopter can be useful, where design parameters and control options can be tested. The structure of the quadcopter will be constructed while selected components will be obtained based on the requirements for the quadcopter.

The purpose of the project will be formulated in Chapter 3.

CHAPTER 3

PROBLEM SPECIFICATION

In this chapter a problem statement and delimitation will be made on the basis of problem analysis.

SECTION 3.1

PROBLEM STATEMENT

A problem statement is made, which gives a baseline for the work to be made and the overall direction for this project. Consequently, relevant theories and methods are needed to be explored. The problem statement is as follows:

How can a quadcopter be designed, modeled, and optimized for energy efficient use while remaining stable?

SECTION 3.2

PROBLEM DELIMITATION

Limitations to the problem statement are made in order to make a coherent report within the time frame of this project.

For the development of the quadcopter design, the main focus will be to make it energy efficient and stable while hovering in the air. By stable, should it be understood, the quadcopter should be able stay in the air at the desired angular position, without wobbling significantly. Furthermore, the quadcopter should be dimensioned to meet specific requirements, with effective use of power to drive the motors while not adding unnecessary weight the structure.

Criteria for the mechatronical system design is made with inspiration of the ASME student design competition 2014 which are found in Appendix A. The following criteria are made:

- The quadcopter should be capable of lifting 2 [kg] besides its own weight.
- The quadcopter must be robust and able to handle minor crashes.
- The quadcopter must be able to hover stably.

A mechatronical system optimization will be made to be able to find the right components to fulfill the design criteria and to make an initially system design.

A model of the quadcopter will be made and implemented in MATLAB and Simulink to simulate the motion of the quadcopter and testing a controller for hovering. Additionally, the model will be compared with the constructed physical system.

To test the developed quadcopter, experiments will be conducted and compared to simulations of the model. Data acquisition from the quadcopter will be used, to be able to verify the made and implemented model of the quadcopter.

MECHATRONIC SYSTEM DESIGN

In this chapter, the structure of the quadcopter is presented, which includes the construction of the frame and the process of determining relevant parameters for the design. Furthermore, electrical components are described together with the essential programming of the system.

SECTION 4.1

REQUIREMENTS

Requirements for the quadcopter design is set. The quadcopter should

- have a lift capacity of 2 [kg] besides its own weight
- have individual control of each motor
- be controlled via a wireless connection
- be battery powered
- have sensors to accurately approximate the pitch and roll angle of the quadcopter
- be able to collect data during flight for analysis

SECTION 4.2

STRUCTURE

This section will contain information about the mechanical structure of the quadcopter.

To construct a quadcopter for energy efficient flight, the design has been optimized to use as little energy as possible while still being capable of producing enough thrust to lift itself and a load of 2 [kg]. Additionally, the structure should robust and not yield under the stress of itself due to forces or resonance.

An object function is made and minimized to obtain optimized parameters for the structure of the quadcopter. The object function consists of the torque induced by the drag of the propeller and the angular velocity of the rotor and propeller. An expression for the torque was found in Appendix B. The object function is given by

$$P = \tau\omega \quad [\text{W}] \quad (4.1)$$

where P is the power used by one motor, τ is the torque on the motor induced only by drag on the propeller, and ω is the angular velocity of the rotor and propeller.

Besides the object function, constraints for the optimization are made, which are:

1. Lifting capability (Equation C.3)
2. Geometric limitations (Equation C.2)
3. Stress in the arm (Equation C.4)
4. Resonance of the arm, motor, and propeller (Equation C.7)
5. Relation between width and height of arm profile must be less than or equal to 2

4.3. ELECTRICAL COMPONENTS

Derivation of these constraints can be found in Appendix C. The design is optimized using the evolutionary method called differential evolution which has the benefit of having a better chance of finding a global minimum of the object function. The specific algorithm has been explained in depth in Section C.6.

The result of the optimization for a rectangular profile is found to be

Propeller diameter	15	[in]
Propeller pitch	6.557	[in]
Angular velocity of propeller	1,458	[RPM]
Length of arm	0.283	[m]
Width of arm	14.2	[mm]
Height of arm	7.1	[mm]

Table 4.1: Design variables found using optimization

It is further observed that the diameter of the propeller is limited to a maximum of 15 [in] due to commercial availability. It is noted that constrain three, four, and five are all heavily influenced by the width and height of the arm. From the optimized parameters found it is observed that the fifth constrain is at its limit meaning a more feasible point could be found by changing this constraint and allowing the ratio between width and height to be greater. The width is primarily increasing due to the fourth constrain which can be omitted if the rotors and propellers are properly balanced.

Due to the time frame limitation of the project some components were ordered before the optimization had ended, however, the tendency of a large propeller had already become apparent as well as keeping the mass of the arms at a minimum. The chosen component were manually checked and fulfilled the constraints of the problem except the fourth constraint. The components were also chosen with regards to availability and price.

The quadcopter which has been built has the following characteristics

Propeller diameter	12	[in]
Propeller pitch	3.8	[in]
Length of arm	0.234	[m]
Width of arm	8.0	[mm]
Height of arm	12.0	[mm]

Table 4.2: Design variables for the constructed quadcopter

The arms were made of aluminium and the rest of the quadcopter frame was made of 3D printed PLA plastic.

SECTION 4.3

ELECTRICAL COMPONENTS

The electrical components acquired to build the quadcopter will be presented as well as a description of how they are connected.

The chosen components is as follows:

- National Instruments (NI) myRIO (with internal accelerometer)
- Gyroscope
- ESC
- BLDC motor
- LiPo battery
- Power distribution board

- 9V battery

The sensors chosen are a accelerometer and gyroscope which can be used to approximate the pitch and roll angle fairly accurately. To be able to make the quadcopter hover accurate feedback of the pitch and roll is needed.

The connections between the components are illustrated in Figure 4.1. As seen in the diagram, the LiPo battery is connected to the power distribution board (positive (red), and negative (black)), whose function is to distribute the power from the battery to the motors through the ESCs. The ESCs are connected to the battery through the power distribution board (positive (red), and negative (black)). Additionally, the ESCs are connected to the myRIO through the signal wires (white, blue, green, and yellow on the distribution board, and orange and black on the ESCs). Each ESC is connected to a BLDC motor through a three-phase signal (red, yellow, and black). Furthermore, the gyroscope is connected to the myRIO through a positive 3.3 [V] (red) and ground (black) connection together with a clock (yellow), data (green) and an interrupt (blue) signal. The myRIO is connected with a 9 [V] battery to power it. An independent power supply is chosen for the myRIO, as the voltage of a fully charged LiPo battery (16.8 [V]) exceeds the maximum input voltage to the myRIO (16 [V]).

NATIONAL INSTRUMENTS MYRIO

A National Instruments myRIO-1900 is used as the main controller of the quadcopter, where it handles signals for the ESCs and the sensor measurements. The myRIO is a portable re-configurable I/O device, which is programmed using LabVIEW. A 9 [V] battery is used to power the myRIO on the quadcopter, which is sufficient as the myRIO has power supply range of 6-16 [V] DC. The myRIO has a maximum power consumption of 14 [W], which at an applied current of 9 [V] yields a maximum current drawn at 1.56 [A].

The myRIO has a built in three-axis accelerometer, where continuous samples of each axis are made. The specifications of the accelerometer is given in Table 4.3.

Number of axes	3 (translational)
Range	± 8 [g]
Resolution	12 bit
Sample rate	800 [S/s]
Noise	$3.8 \cdot 10^{-2}$ [$(m/s^2)_{rms}$] typical at 25 °C

Table 4.3: Specifications of the accelerometer in the myRIO. [4]

The myRIO outputs PWM signals to the ESCs and collect data from the gyroscope, meaning a number of I/O ports has to be configured. Table 4.4 shows an overview of the port connection to the myRIO.

MXP	Pin	Function	Wire Color	
A	11	DIO0	Gyroscope	Yellow
A	30	DGND		Black
A	32	DIO14 / I ₂ C.SCL		Green
A	33	+3.3 V		Red
A	34	DIO15 / I ₂ C.SDA		Blue
A	27	DIO8 / PWM0 (front left)	Motors	Blue
A	29	DIO9 / PWM1 (front right)		White
B	27	DIO8 / PWM0 (Rear left)		Green
B	29	DIO9 / PWM1 (Rear right)		Yellow
B	30	DGND		Black

Table 4.4: Overview of pins used on the NI myRIO.

Furthermore, to be able to communicate with the myRIO, the Wi-Fi feature on the myRIO is utilized, where the

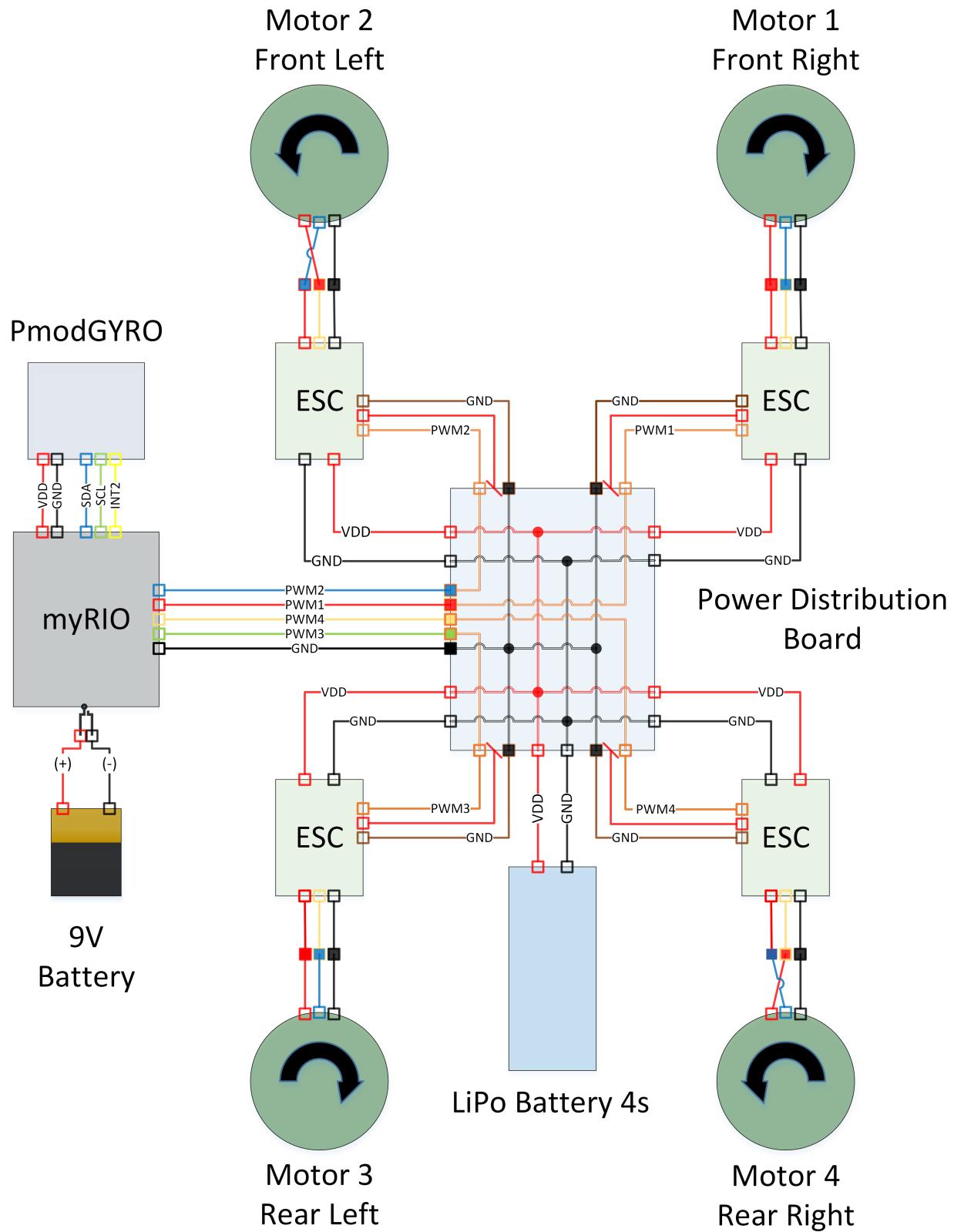


Figure 4.1: An overview of the connections between the electrical components of the quadcopter.

connection to the host computer is made through a Wi-Fi hotspot. In addition, data is saved on a Universal Serial Bus (USB) connected to the USB port in the myRIO.

GYROSCOPE

The gyroscope used on the quadcopter is a PmodGYRO 3-Axis Digital Gyroscope featuring the STMicroelectronics® L3G4200D MEMS motion sensor. The L3G4200D provides the three-axis digital output gyroscope with a built in temperature sensor. The PmodGYRO has 12-pin and 8-pin connections and communicates via Serial Peripheral Interfacebus (SPI) or Inter-Integrated Circuit (I2C), with I2C as default. The PmodGyro is programmed in



Figure 4.2: Picture of the PmodGYRO 3-Axis Digital Gyroscope. [1]

Number of axes	3
Range	± 250 [dps]
Resolution	16 bit
Serial Clock rate	100 [kHz]
Output Frequency	100 [Hz]
Noise density (BW=50 Hz)	0.03 [dps/ $\sqrt{\text{Hz}}$]

Table 4.5: Specifications for the PmodGYRO, ([dps] = degrees per second). [5]

LabVIEW using I2C communication, where Table 4.5 shows an overview of the specifications. The pins used on the PmodGYRO, is a power supply (3.3 [V]), power supply ground, serial clock line (SCL), serial data line (SDA), and data ready interrupt. The pins used on the PmodGyro and their connections with the myRIO are shown in Figure 4.3.

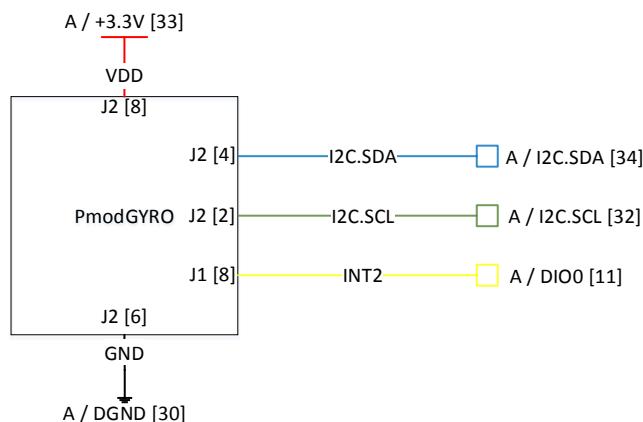


Figure 4.3: Overview of pin connection between the PmodGYRO and myRIO.

ESC

Four Turnigy Multistar 30 Amp Multi-rotor brushless ESC 2-4S are used to control the BLDC motors. The maximum steady-state current the ESCs can handle is 30 [A] and can be used with 2-4 cell LiPO batteries. The ESCs have two inputs for the power supply (positive and ground) and three outputs for the motor. In addition it has a 5.5 [V] output for a RC receiver, a ground signal, and a PWM signal input. In the construction of this quadcopter the 5.5 [V] is not being used as the PWM signals to the ESCs are controlled by the myRIO.



Figure 4.4: Picture of Turnigy Multistar 30 Amp Multi-rotor brushless ESC 2-4S. [2]

The PWM signals to the ESCs can be sent in a frequency range of 50-400 [Hz], where the default non to full throttle range is at a active time of 900-2000 [μ s]. Meaning the rotors achieve their maximum rotational speed, when the ESCs receive a PWM signal with an active time of 2000 [μ s], and they do not spin at an active time of 900 [μ s].

Furthermore, the rotational direction of the motor is determined by how the three phases between the ESCs and motors are connected. The rotational direction can be changed by swapping any two of the three phases, as illustrated on Figure 4.1, where motor 2 and 4 rotate in the opposite direction.

BLDC MOTOR

Four 4112-485KV Turnigy Multistar 22 Pole Brushless Multi-Rotor Motors are used for the quadcopter. The motors can be used with 4-8 cells LiPo batteries and draws a maximum current of 28 [A]. Each motor has 22 poles and a K_v value of 485 [RPM/V].



Figure 4.5: Picture of 4112-485KV Turnigy Multistar 22 Pole Brushless Multi-Rotor Motor.

PROPELLER

Carbon Fiber Propellers are used for the quadcopter with a diameter of 12 inches (30.48 [cm]) and a pitch of 3.8 inches (9.65 [cm]) per revolution.



Figure 4.6: Picture of a Carbon Fiber Propeller used on the quadcopter.

A characteristic of the ESC, motor, and propeller has been made and is found in Appendix D. The results is presented in Section 4.4.

BATTERY

A Turnigy nano-tech 5000 mAh 4s 35~70C Lipo Pack is used. It is a 4S1P battery, meaning all the cells are in series. Each cell has a voltage range of 3.7 - 4.2 [V] depending on the battery being fully charged or not.

The battery has a capacity of 5000 [mAh], and has a C-rating of 35C at constant discharging, while 70C for burst discharging. As mentioned in Chapter 2, the C-rating is a measure of the maximum safe discharge rate of the battery pack. For example for continuous discharge of the battery pack, the maximum discharge current is given by

$$5[\text{Ah}] \cdot 35[\text{h}^{-1}] = 175[\text{A}] \quad (4.2)$$

In the same manner the burst discharging maximum current can be found to 350 [A].



Figure 4.7: Picture of the Turnigy nano-tech 5000 mAh 4s 35~70C Lipo Pack.

POWER DISTRIBUTION BOARD

The power from the battery is distributed to the ESCs with a power distribution board. The quadcopter uses an Anself T Plug Power Distribution Board, which also cluster the PWM signal, 5.5 [V] output, and ground connectors from each ESC.

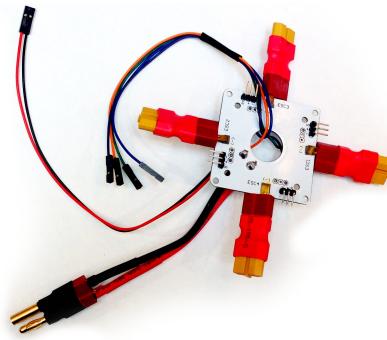


Figure 4.8: Picture of the Anself T Plug Power Distribution Board.

SECTION 4.4

CHARACTERISTICS

An characterization of the propeller, motor, and ESC has been made and is described in Appendix D. Based on this analysis expressions for the thrust and the steady-state angular velocity is found. The thrust at a specific angular velocity for propeller n can be defined by

$$F_{l,n} = k_l \omega^2 \quad [N] \quad (4.3)$$

$$k_l = (2.11 \times 10^{-4}) \quad (4.4)$$

where ω is the angular velocity of the propeller around the center axis and k_l is an experimentally determined constant for the characterized propeller. The angular velocity can be found by

$$\omega = (k_1 V - k_2) D_c + k_3 V + k_4 \quad [\text{rad s}^{-1}] \quad (4.5)$$

$$k_1 = 29.8 \quad (4.6)$$

$$k_2 = 66.3 \quad (4.7)$$

$$k_3 = 12.6 \quad (4.8)$$

$$k_4 = 131 \quad (4.9)$$

where V is the input voltage to the ESC coming from the battery, D_c is the PWM signal coming from the myRIO, and k_1 , k_2 , k_3 , and k_4 is experimentally determined constants specific for the ESC, motor, and propeller.

SECTION 4.5

IMPLEMENTATION

The implementation in LabVIEW is described in the following with a description of how the sensors (accelerometer and gyroscope) are implemented, calibrated, and filtered. For this application a complementary filter is used to approximate the pitch and roll angle of the quadcopter by using data from the accelerometer and gyroscope. Furthermore, the control of the quadcopter, including the regulators used, is presented together with how the signals are sent to the ESCs. In addition, the data logging is specified. Figure 4.9 shows the essential parts of the LabVIEW application and Figure 4.10 shows the main block diagram of the LabVIEW program, which can also be found on the appended CD.

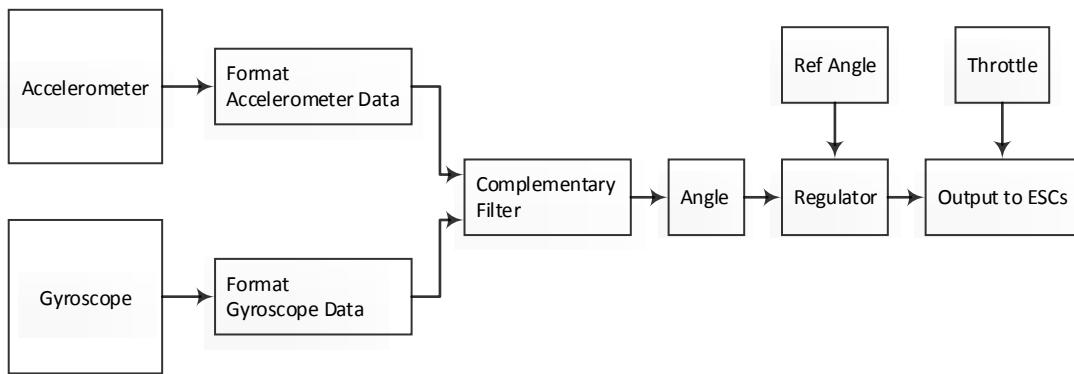


Figure 4.9: Diagram of the essential parts in the LabVIEW framework.

Data from the accelerometer and the gyroscope is sampled at 100 [Hz] in the first loop (1). Moreover, the output signals to the ESCs and the data logging is made in the same loop. A time loop is chosen to assure a continuous loop frequency of 100 [Hz]. In addition, the priority of the loop is set to 3, which means the loop will have the highest priority of the three implemented loops, and will therefore execute first.

All the calculations utilized and the regulators used in the LabVIEW program are implemented in the second loop (2). This loop is also a time loop running at frequency at 100 [Hz]. The priority of the loop is set to 2, meaning this loop will execute secondly.

The third loop (3) is used for the front panel interface on the host computer, where it handles data to be displayed on graphs . A time loop is used, which runs at frequency at 2 [Hz] and has priority 1, the lowest of the loops. There is no need to execute the loop with a higher frequency, as the loop only handles data for graphs, which otherwise would take up processing power needed for more important tasks.

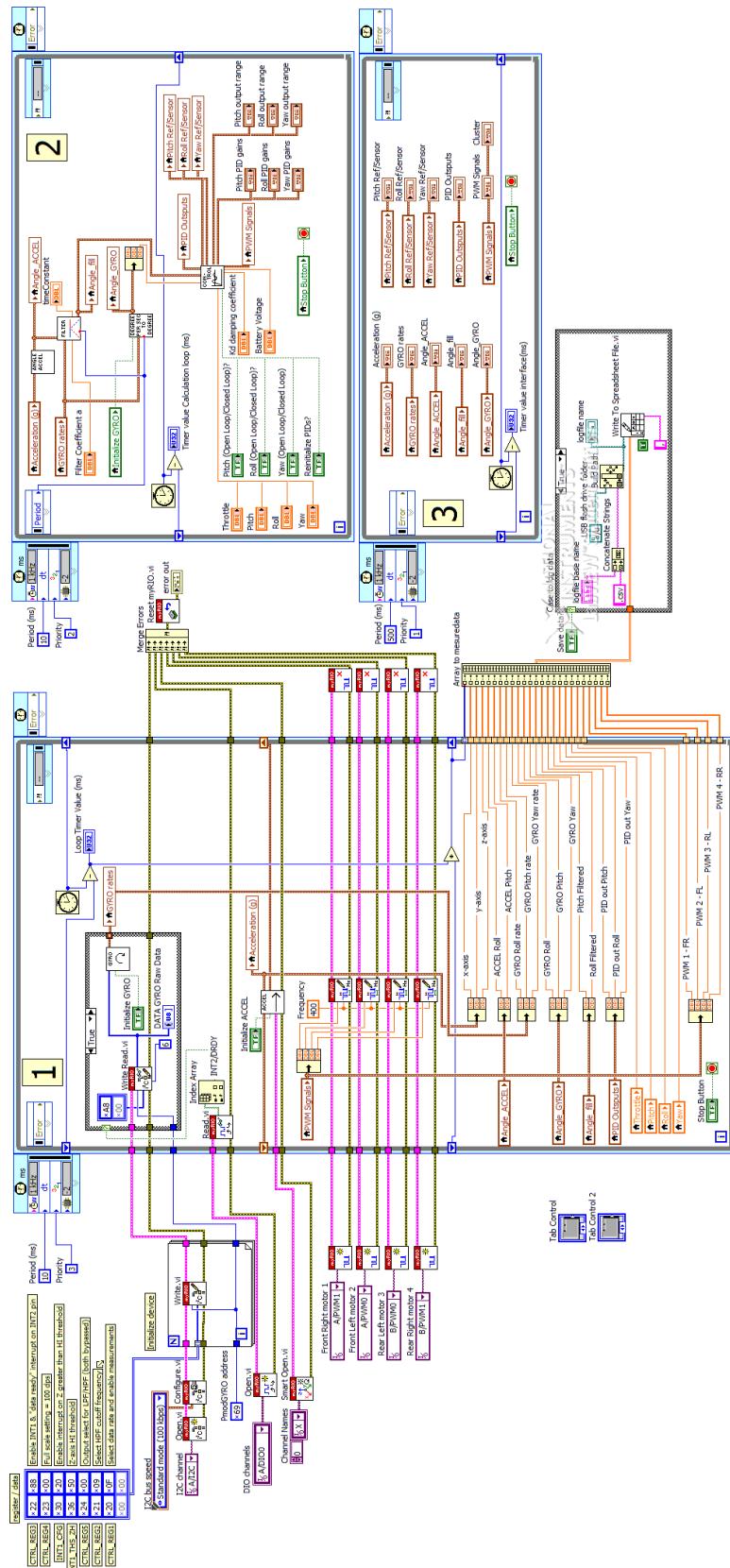


Figure 4.10: The main block diagram of the LabVIEW program.

UTILIZATION OF THE ACCELEROMETER

An accelerometer measures acceleration forces in gravitational acceleration ($[g]$). The forces can be static or dynamic caused by moving or vibrating the accelerometer. Figure 4.11 illustrates a single sensing axis.

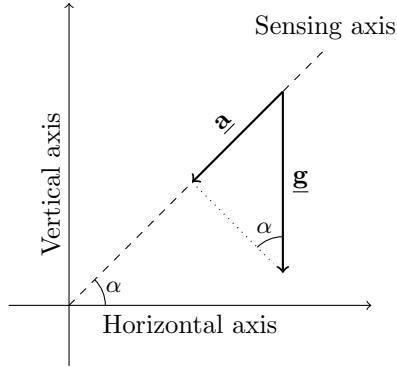


Figure 4.11: Illustrations of single axis sensing of the accelerometer.

The accelerometer measures the projection of the gravity vector on each of the sensing axes. The amplitude of measured acceleration changes as sinus function of the angle α between the sensing axis and the horizontal plane, which is expressed using Equation 4.10.

$$\underline{a} = \underline{g} \sin(\alpha) \quad [g] \quad (4.10)$$

The pitch and roll angles can be approximated by using trigonometry and all three sensing axis of the accelerometer, expressed in Equations 4.11 and 4.12. [24]

$$\text{Pitch} \quad \phi = \tan^{-1} \left(\frac{y^2}{\sqrt{x^2 + z^2}} \right) \quad [\text{rad}] \quad (4.11)$$

$$\text{Roll} \quad \theta = \tan^{-1} \left(\frac{x^2}{\sqrt{y^2 + z^2}} \right) \quad [\text{rad}] \quad (4.12)$$

By using the *Pythagorean theorem*, Equation 4.13, and the tangent relation of trigonometry, Equation 4.14, the equations can be derived.

$$a^2 + b^2 = c^2 \quad (4.13)$$

$$\tan(\alpha) = \frac{\text{opposite}}{\text{adjacent}} \quad (4.14)$$

Figure 4.12 illustrates how the pitch angle is found for a case where the accelerometer is held at both a roll and pitch angle.

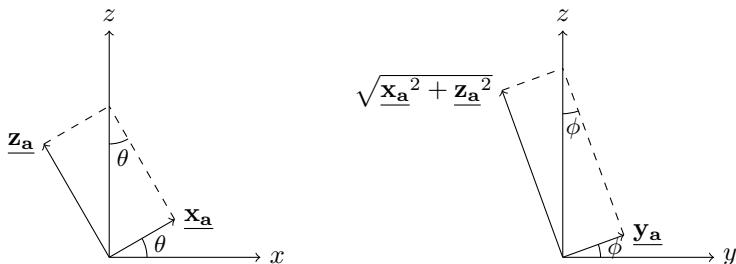
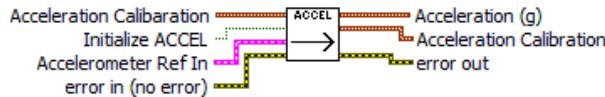


Figure 4.12: Illustrations of finding the pitch angle for a case where the accelerometer is at a roll and pitch angle. Roll (left), and pitch (right).

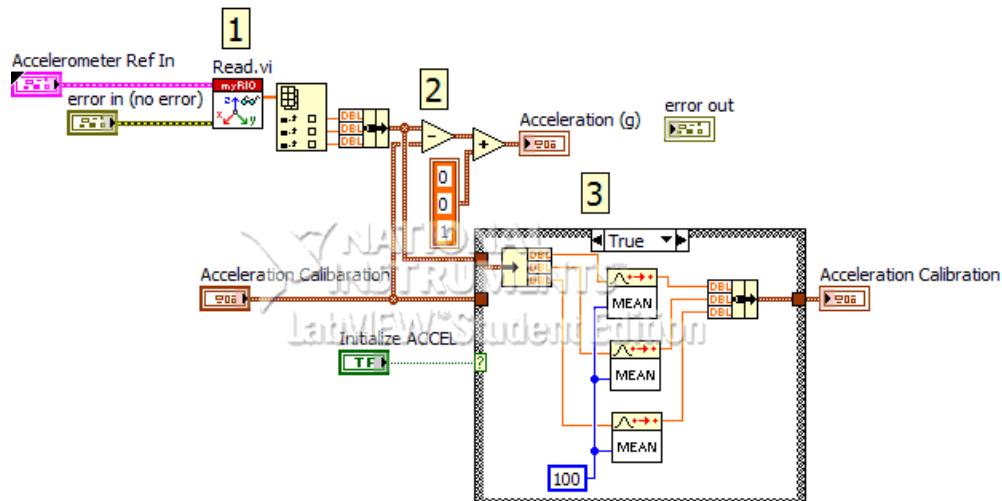
4.5. IMPLEMENTATION

The diagonal of the triangle spanned by the sensed accelerations \underline{z}_a and \underline{x}_a has to be found (left), as it is the vector the sensed acceleration \underline{y}_a forms a pitch angle with (right). The roll angle, for the same case, can be found by a similar procedure.

The collection of the accelerometer data in the LabVIEW program is done in the LabVIEW subsystem (subVI) called ACCEL, as seen in Figure 4.13a.



(a) ACCEL subVI in the main block diagram.



(b) Block diagram of the ACCEL subVI.

Figure 4.13: Implementation of the accelerometer in the LabVIEW program.

In the subVI the data from the accelerometer is called from the myRIO (1), as seen on Figure 4.13b.

The offset needs to be found, so the reading is zero when the accelerometer is level. The offset is found by seeing the average value of sensor reads, when it is horizontal and stationary, thereafter it is subtracted from the read data (2).

The offset is found when *Initialize ACCEL* turns true, the value is then stored in a shift register (3). This process is executed manually in the beginning of each initialization of the program.

From the collected data the pitch and roll are calculated in the subVI called ANGLE ACCEL placed in the second loop in the main block diagram, seen in Figure 4.14a.

The implementation of Equations 4.11 and 4.12 in the ANGEL ACCEL subVI is seen on Figure 4.14b. The *ATAN2* block is used to be able to calculate the angles in any of the four quadrants of the plane.

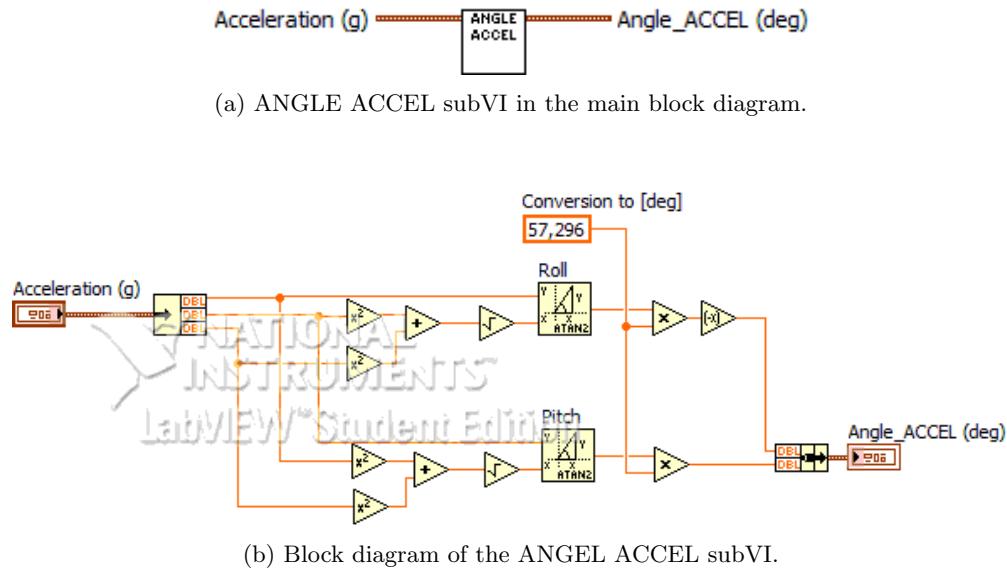


Figure 4.14: Implementation of the approximation of pitch and roll using the accelerometer.

UTILIZATION OF THE GYROSCOPE

The gyroscope measures angular velocity when it is rotated around any of the three axes, which is commonly in degrees per second. The gyroscope itself needs to be initialized before any data can be read. This is done by programming the gyroscope through register addresses, which enables the three-axis outputs, sets the range, and the output frequency. Furthermore, the pin used for the gyroscope has to be enabled on the myRIO. The data is formatted in the GYRO subVI, which is seen on Figure 4.15a.

The data from the gyroscope is being read in the GYRO subVI (1).

The data is formatted from six bytes into three 16-bit integers, which each contains the angular velocity of one of the three axis (2).

The data is then being converted into [dps] (3). When the range of the gyroscope is set to ± 250 [dps], the sensitivity is 0.00875 [dps/digit] [5].

The offset is subtracted from the data (4).

The offset is found by seeing the average value of the sensor reads, while the gyroscope is horizontal and stationary, when the *Initialize GYRO* is manually turned true (5).

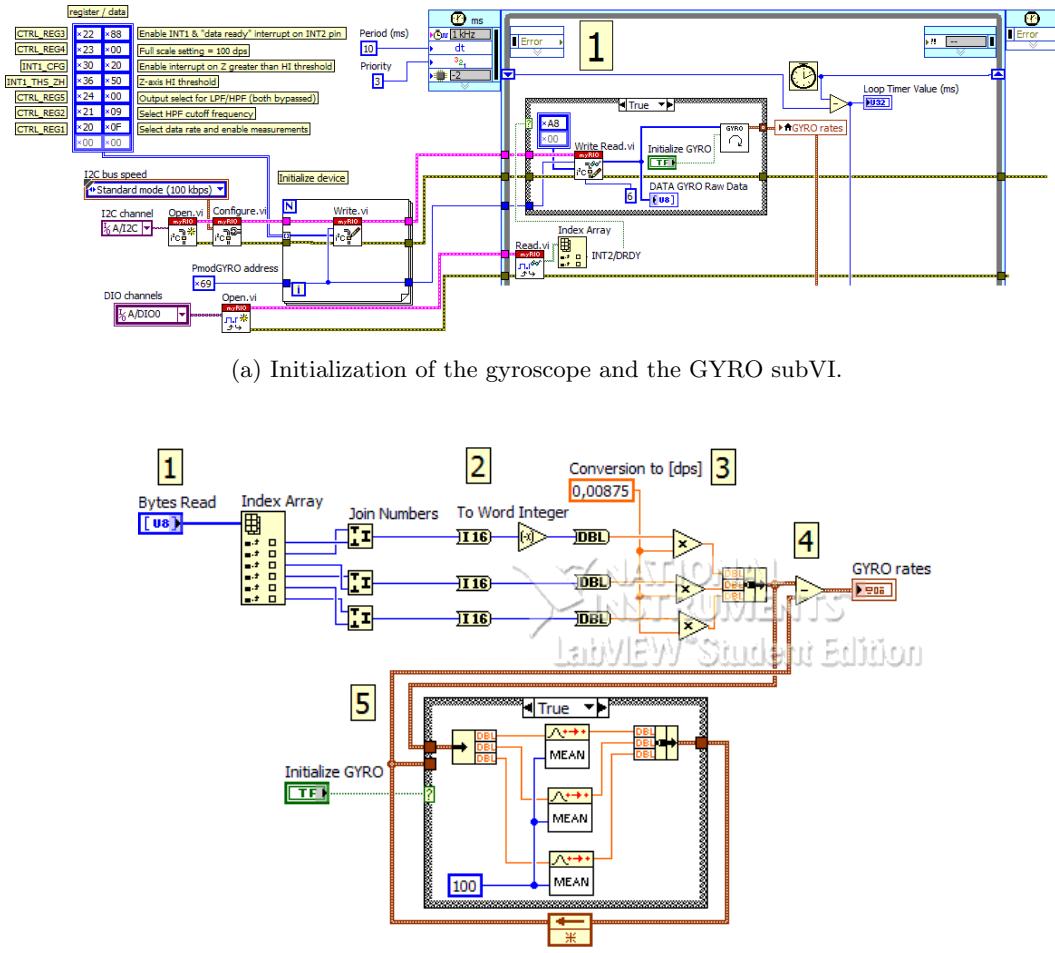


Figure 4.15: Implementation of the gyroscope in the LabVIEW program.

COMPLEMENTARY FILTER

This section is based on the presentation [25] and the lecture [26].

The accelerometer and gyroscope can be used together to estimate the pitch and roll. The tilt angle calculated from the accelerometer is sensitive to quick changes, while the angle obtained from the gyroscope by integration is subjected to drift over a period of time. The accelerometer is useful in the long term while the gyroscope is useful in the short term.

The complementary filter is designed so the strength of both sensors is used to overcome the weaknesses of the sensors, and thereby complement each other. The filter uses the gyroscope in the short period, and then the accelerometer is used to correct the drift of the angle over a long period of time. This will result in a fast responding calculation of the angles and minimal drift of the quadcopter. Drift means that the gyroscope reading will increase or decrease over time, even when held at a constant position. Figure 4.16 illustrates the complimentary filter.

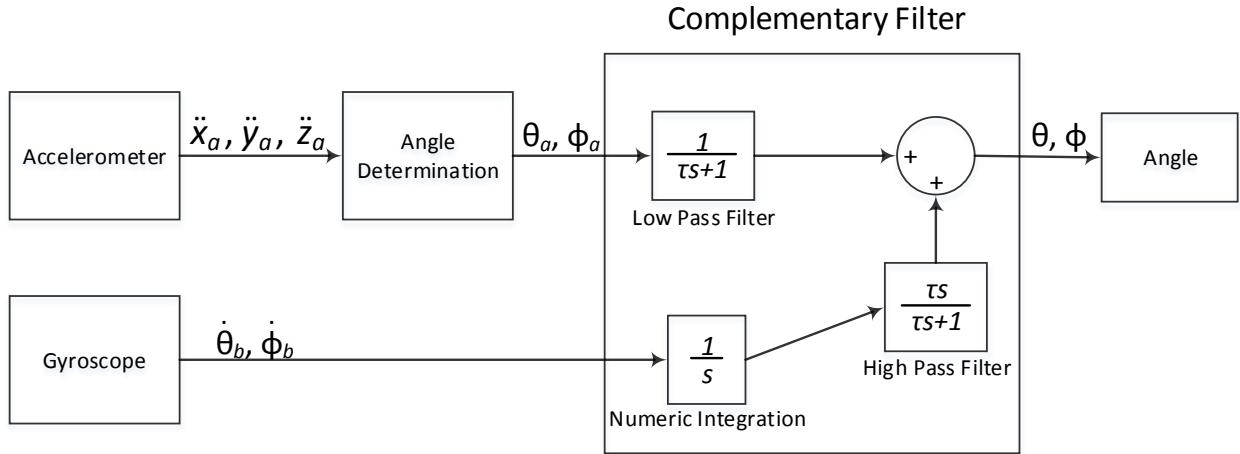


Figure 4.16: Illustration of the complementary filter.

The pitch and roll approximated by the accelerometer, is passed through a *low pass filter*, which will only let low frequency signals pass through and filter out high frequency fluctuations, which could be caused by vibrations.

The angular velocities from the gyroscope is integrated to determine angular position and then passed through something which resembles a *high pass filter*. The effect will be similar to the *low pass filter* caused by the integration as

$$\frac{1}{s} \frac{\tau s}{\tau s + 1} = \frac{\tau}{\tau s + 1} \quad (4.15)$$

The time constant τ will be approximately the same as the *low pass filter* [25]. If the gyroscope changes angle quickly, it will, through integration, estimate the angles fairly accurately, however, the angles will begin to drift over a period of time, when held at a constant position due to accumulated stationary error.

The implementation of the filter can be derived by Equation 4.16. The derivation is done on the pitch angle only as the procedure is the same for the roll angle.

$$\phi = \phi_{accel} \frac{1}{\tau s + 1} + \dot{\phi}_{gyro} \frac{1}{s} \frac{\tau s}{\tau s + 1} \quad (4.16)$$

where ϕ is the filtered angle, ϕ_{accel} is the angle approximated by the accelerometer, and $\dot{\phi}_{gyro}$ is the angular velocity measured by the gyroscope.

$$\phi = \phi_{accel} \frac{1}{\tau s + 1} + \dot{\phi}_{gyro} \frac{\tau}{\tau s + 1} \quad (4.17)$$

$$\phi(\tau s + 1) = \phi_{accel} + \dot{\phi}_{gyro} \tau \quad (4.18)$$

$$\phi \tau s + \phi = \phi_{accel} + \dot{\phi}_{gyro} \tau \quad (4.19)$$

Inverse Laplace and Euler Backward discretization gives

$$\dot{\phi} \tau + \phi = \phi_{accel} + \dot{\phi}_{gyro} \tau \quad (4.20)$$

$$\frac{\phi_k - \phi_{k-1}}{dt} \tau + \phi_k = \phi_{accel} + \dot{\phi}_{gyro} \tau \quad (4.21)$$

Where the subscripts k and $k - 1$ denotes the current and previous angle respectively, and dt is the time increment. Isolating for the current filtered angle ϕ_k yields

$$\phi_k \tau - \phi_{k-1} \tau + \phi_k dt = \phi_{accel} dt + \dot{\phi}_{gyro} \tau dt \quad (4.22)$$

$$\phi_k (\tau + dt) = \phi_{accel} dt + \dot{\phi}_{gyro} \tau dt + \phi_{k-1} \tau \quad (4.23)$$

$$\phi_k = \phi_{accel} \frac{dt}{\tau + dt} + \dot{\phi}_{gyro} dt \frac{\tau}{\tau + dt} + \phi_{k-1} \frac{\tau}{\tau + dt} \quad (4.24)$$

By defining a filter coefficient $\alpha = \frac{\tau}{\tau + dt}$ yields

$$\phi_k = (1 - \alpha) \phi_{accel} + \alpha (\dot{\phi}_{gyro} dt + \phi_{k-1}) \quad (4.25)$$

The implementation of the filter is thereby derived. The total filter for pitch and roll is illustrated in Figure 4.17 and general expressed in Equation 4.26 [25].

$$\text{Estimated Angle} = (\alpha)(\text{Previous Angle} + \text{Gyro} \cdot dt) + (1 - \alpha)(\text{Accelerometer}) \quad (4.26)$$

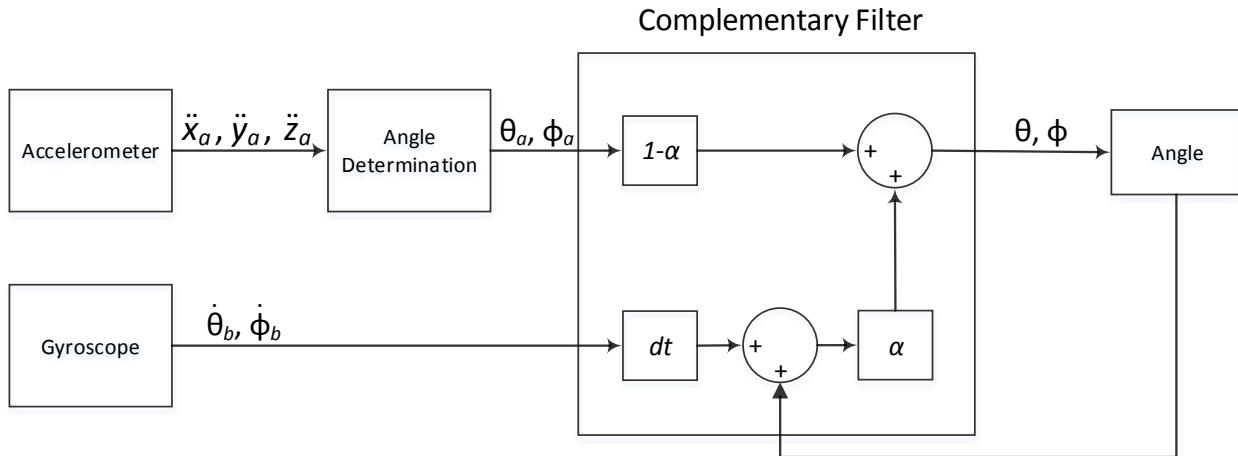


Figure 4.17: Illustration of the implemented complementary filter.

The time increment (dt) is the time that passes between each loop of the LabVIEW program. It is the reciprocal value of the sampling frequency.

The time constant τ defines the boundary of thrusting the gyroscope or the accelerometer. The time constant can be calculated using Equation 4.27 [25].

$$\tau = \frac{\alpha \cdot dt}{1 - \alpha} \quad [\text{s}] \quad (4.27)$$

For time periods, which is shorter than the time constant, the gyroscope has precedence and the noisy accelerations are filtered out. For time periods longer than the time constant, the angles approximated by the accelerometer is given more weight than the gyroscope, which may have begun to drift. The filter coefficient is determined in the following section.

The filter is implemented in the main block diagram in the subVI called Filter, seen in Figure 4.18a.

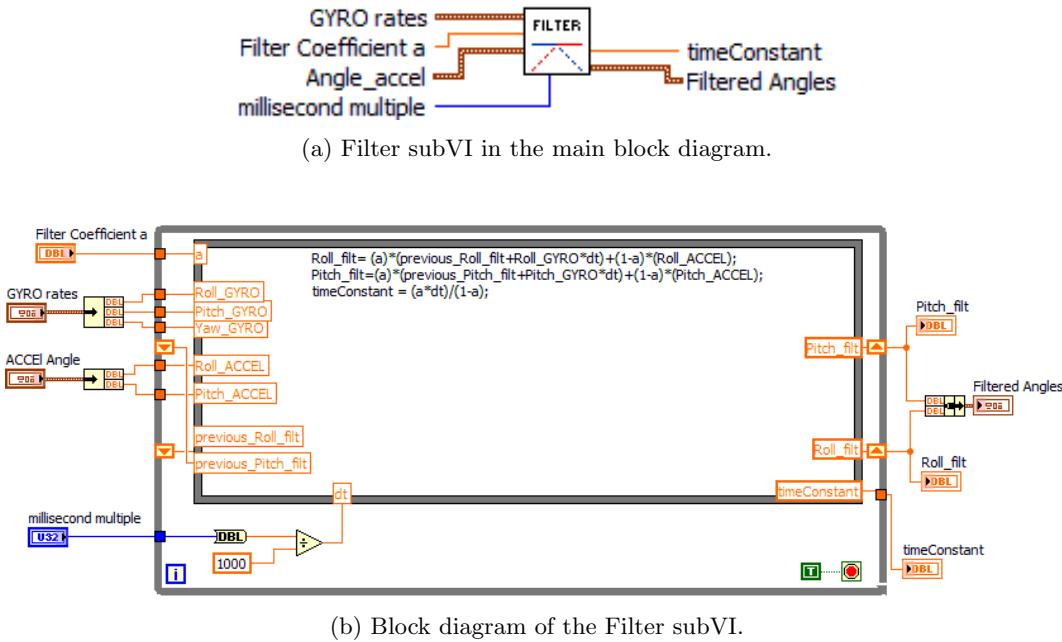


Figure 4.18: Implementation of the filter in LabVIEW.

The block diagram of the Filter subVI is shown in Figure 4.18b, where the complementary filter from Equation 4.26 is implemented to calculate both the pitch and roll.

DETERMINATION OF THE FILTER COEFFICIENT

The filter coefficient is determined by collecting a data set on the roll angle consisting of the angle approximated by the accelerometer, the angle approximated by angular velocity measured by the gyroscope, and sampling time. While collecting data, the quadcopter is tilted manually in a sequence of roll angles. First a negative roll and then back over a period of about 2 seconds, then again a negative roll and then back over a period of about 0.5 [s]. The collected data is then filtered with different filter coefficients using Equation 4.26. The value of the filter coefficient is in the range of 0-1, which specifies the weighting of the accelerometer and gyroscope. If the filter coefficient is zero, the filter outputs the angles approximated by the accelerometer measurements alone, while with a filter coefficient of 1, the filter outputs the angle approximated by the gyroscope measurements alone. Initially calculations with filter coefficients of 0.2, 0.5, and 0.8 is selected, where the performance of the filtered angle is analyzed. The collected and filtered angles is shown in Figure 4.19.

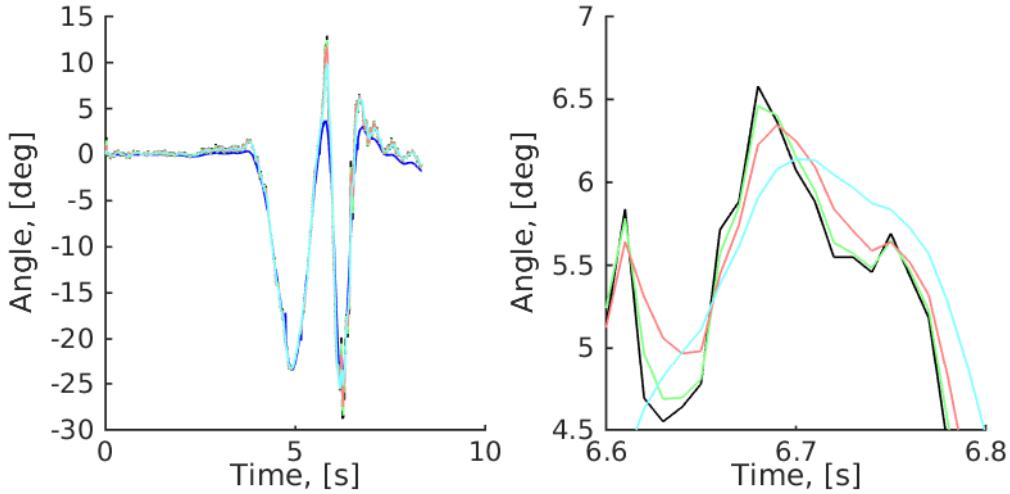


Figure 4.19: Collected and filtered data with filter coefficients of 0.2, 0.5, 0.8. Full period (left) and zoom (right). Black = Angle determined from accelerometer, Blue = Angle determined from gyroscope, Green = Filtered 0.2, Red = Filtered 0.5, Cyan = Filtered 0.8

The filtered data with $\alpha = 0.2$ has the most spikes, which is caused by the accelerometer has the most weighting. The filtered angle with $\alpha = 0.8$ is the smoothest line, while the filtered angle with $\alpha = 0.5$ is in between.

As the filtered angle with $\alpha = 0.8$ has the best performance, additional calculations is made by tuning the filter coefficient. Figure 4.20 shows the collected angles approximated by the accelerometer and gyroscope, and filtered angles where the coefficients is 0.7, 0.8, and 0.9.

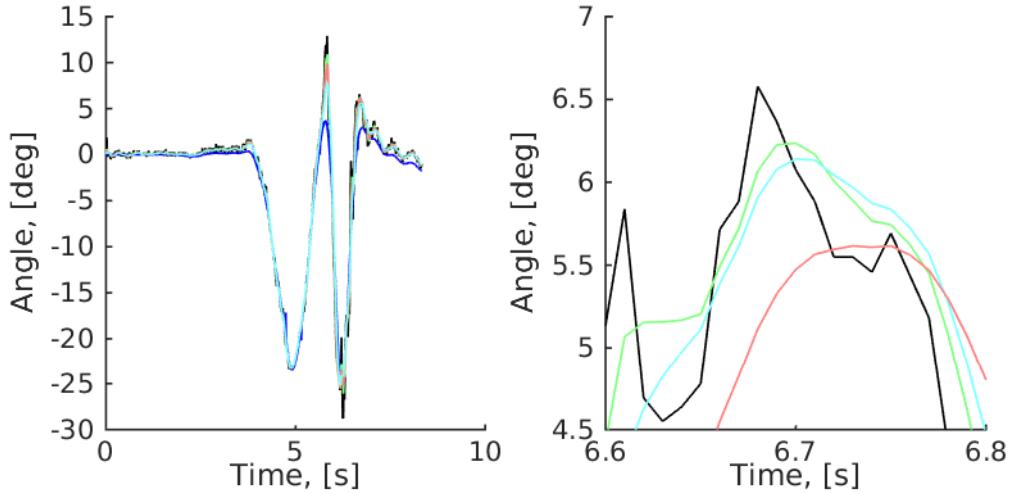


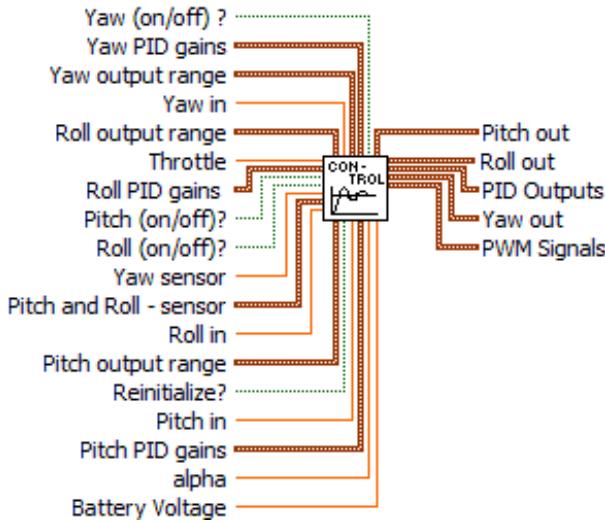
Figure 4.20: Collected and filtered data with filter coefficients of 0.7, 0.8, 0.9. Full period (left) and zoom (right). Black = Angle determined from accelerometer, Blue = Angle determined from gyroscope, Green = Filtered 0.7, Cyan = Filtered 0.8, Red = Filtered 0.9

The angle with $\alpha = 0.7$ has tendency to have more spikes than the other filtered angles. In addition the angle with $\alpha = 0.9$ has begun drifting, if comparing it to the mean value of the angle approximated by the accelerometer, meaning the weighting of the gyroscope is to significant. The angle with $\alpha = 0.8$ has a smooth curve and follows

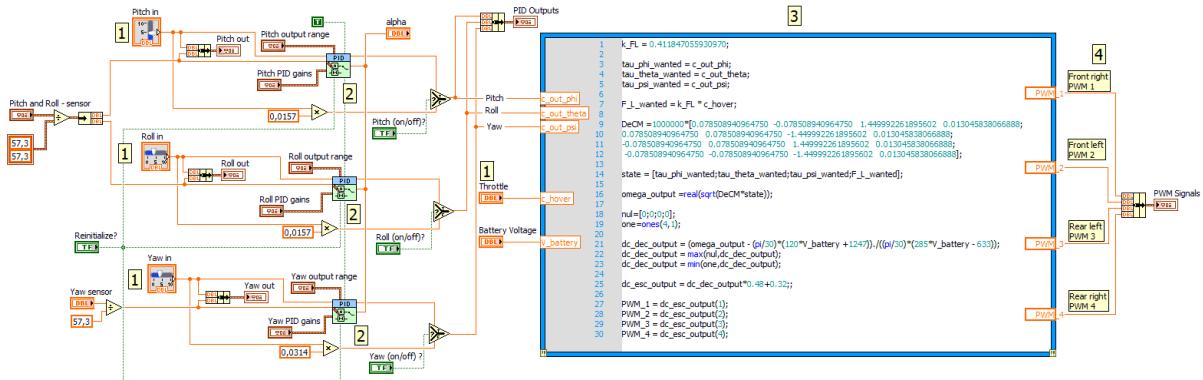
the mean value of the angle determined by the accelerometer and is therefore selected as the filter coefficient for the complimentary filter.

CONTROL

The control of the quadcopter is made in the subVI called Control, seen in Figure 4.21a. The control is made of manual inputs of Throttle, Pitch, Roll, and Yaw together with regulators for the pitch, roll, and yaw angle. Based on the made regulators described in Chapter 6.4 a Proportional-Integral-Derivative (PID) regulator has been chosen to be implemented. As the filtering of data is only made on the pitch and roll angles, it would be assessed if the yaw regulator would be used during experiments. The block diagram of the Control subVI is seen on Figure 4.21b.



(a) Control subVI in the main block diagram.



(b) Block diagram of the Control subVI.

Figure 4.21: Implementation of the control in the LabVIEW programming.

A value for the Throttle, Pitch, Roll, and Yaw can be inputted manually to be able to control the quadcopter (1).

PID regulators for the pitch, roll, and yaw are implemented (2). The *PID Advanced* block is used, which has the ability to damp the effect of the derivative term of the regulator if needed. It will likely be necessary to use damping to reduce the effect of noise from the input signal to the PID regulators. After the PID regulators have

been implemented the necessity of a damping effect will be assessed.

With the output of the regulators the output to the ESCs are then calculated in the *math script node*, based on the derived expression in Section 6.3 (3).

The *math script node* outputs the new duty cycles to the ESCs (4).

OUTPUT TO THE MOTORS

After the new outputs to the ESCs has been found, they are send to each ESC. Figure 4.22 shows the implementation in the main block diagram.

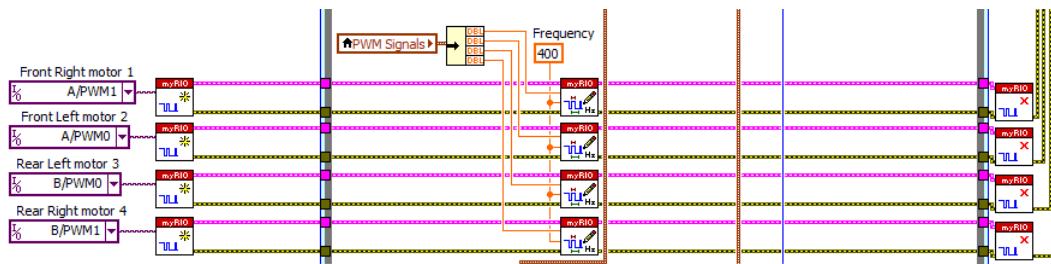


Figure 4.22: Implementation of PWM signals for the ESCs in the main block diagram.

As described earlier, each ESC is connected to the myRIO through four individual pins. The output frequency of the PWM signal is set to 400 [Hz].

DATA LOGGING

To be able to analyze the flight of the quadcopter, data from the experiments is logged using a flash drive connected to the myRIO through its USB port. The logged data includes raw sensor data, filtered data, PID regulator outputs, and manually inputs. The implementation of data logging in the main block diagram is seen on Figure 4.23.

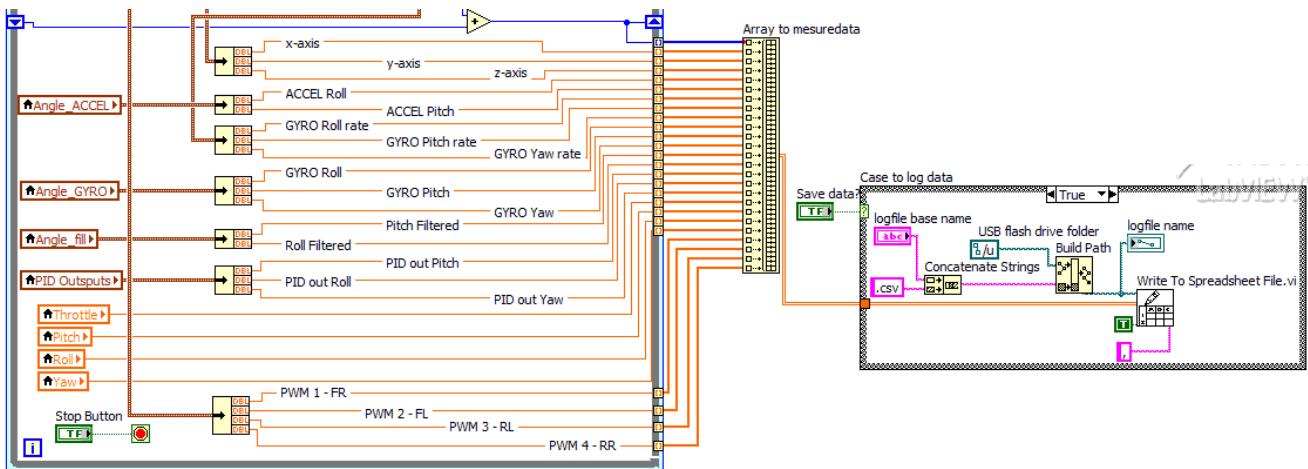


Figure 4.23: Implementation of data logging in the main block diagram.

SYSTEM MODELING

In this chapter the mathematical model of the quadcopter is derived and explained. The model is implemented in Simulink with the purpose of simulating the quadcopter dynamically, and with the ability to change parameters and variables between simulations.

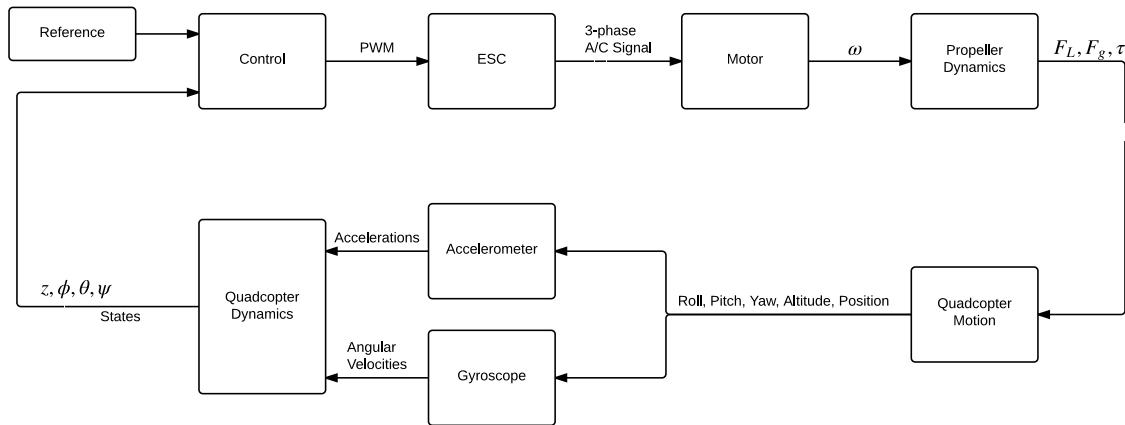


Figure 5.1: Quadcopter model structure

Figure 5.1 shows a block-diagram of the fundamental model structure. The "Quadcopter Motion" block represents the current angular (pitch, roll, yaw) and linear (altitude, position) states of the quadcopter, and of these the accelerometer measures linear accelerations, and the gyroscope sensor measures the angular velocities. The "Quadcopter Dynamics" block contains the dynamic equations for the quadcopter, translating the output of the sensors to linear positions x , y , z , and angular positions ϕ (pitch), θ (roll), and ψ (yaw) states.

In the "Control" block the errors are calculated between reference state from the "Reference" block and the states from the sensors (x , y , z , ϕ , θ , ψ), and the errors are fed into individual controllers for each state yielding some correction output. These outputs are utilized in a series of control algorithms finally yielding a PWM signal for each ESC. The "ESC" block uses the PWM signal into a three-phase AC signal for the motors.

The "Motor" block contains the BLDC motor model, which, based on the ESC signal, gives an angular velocity ω . Finally the "Propeller Dynamics" block yields a new lift and torque based on ω , and the motion of the quadcopter is updated accordingly.

EQUATIONS OF MOTION

To describe the motion of the quadcopter mathematically, it is defined in two three-dimensional frames; global frame and local frame. The local frame has origo at the Center of Gravity (CG) of the quadcopter, and the axes follow the movement of the quadcopter as illustrated in Figure 5.2. Therefore, the thrust from the propellers is always generated in the z_L -direction of the local frame regardless of the angular and linear position of the quadcopter in the global frame. The global frame is the ground-fixed frame wherein the local frame (quadcopter) moves around, and is also illustrated in Figure 5.2. The general definitions for quadrants in two-dimensional cartesian coordinate systems is followed, and propeller 1 and 2 is thereby defined to be the front of the quadcopter in the positive y_L -direction, and propeller 3 and 4 is defined in the negative y_L -direction.

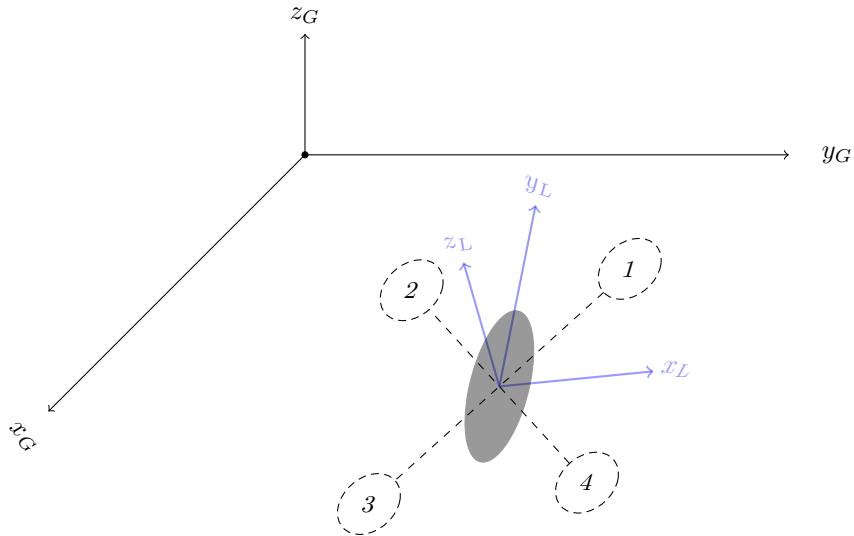


Figure 5.2: The local frame (in blue) is fixed to the quadcopter and moves around in the global frame.

The angular position (pitch, roll and yaw) of the quadcopter is defined by the Euler angles ϕ , θ , and ψ in the vector $\underline{\chi}$. The global frame linear positions x , y and z are defined as linear position vector \underline{S} .

$$\underline{\chi} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}, \quad \underline{S} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (5.1)$$

Furthermore, the local frame angular velocity is defined as the vector

$$\underline{\nu} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad [\text{rad s}^{-1}] \quad (5.2)$$

where p , q , and r is the angular velocity around the x_L -, y_L -, and z_L -axis of the local frame.

In order to derive the equations of motion for the quadcopter, the acting forces and torques in the local frame must be defined in the global frame, which is done by rotating one frame into the other. The global frame is rotated by three counter-clockwise rotations: first rotation is about the x_G -axis by Euler angle ϕ (pitch), second rotation is about the y_1 -axis by Euler angle θ (roll), and the third is about the z_2 -axis by Euler angle ψ .

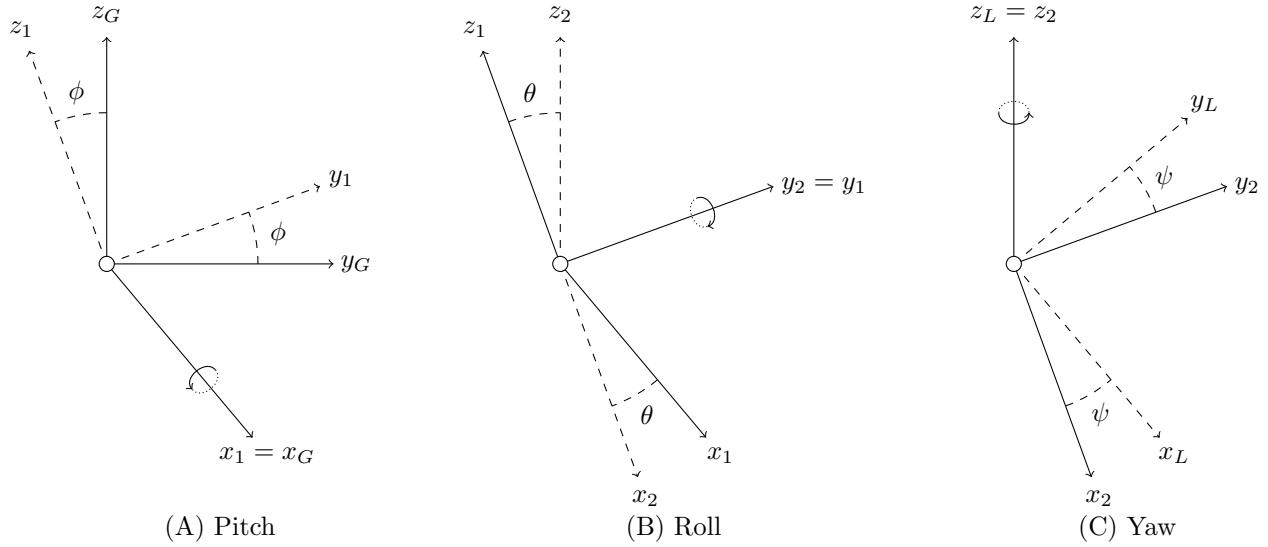


Figure 5.3: The angles ϕ , θ , and ψ defines pitch, roll, and yaw between global and local frame.

Figure 5.3 shows the three counter-clockwise rotations by the three Euler angles. The equivalent rotation matrix for each separate rotation is

$$\underline{\underline{\mathbf{R}_x}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi & c_\phi \end{bmatrix} \quad (5.3)$$

$$\underline{\underline{\mathbf{R}_y}} = \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \quad (5.4)$$

$$\underline{\underline{\mathbf{R}_z}} = \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.5)$$

where $\underline{\underline{\mathbf{R}_x}}$, $\underline{\underline{\mathbf{R}_y}}$, and $\underline{\underline{\mathbf{R}_z}}$ is the counter-clockwise matrix rotations around the x -, y -, and z -axis, and c_ψ is $\cos(\psi)$, s_ϕ is $\sin(\phi)$ etc. [27]. For the reversed rotation, when rotating the local frame to the global frame, the direction and order of rotations must be reversed compared to the order shown in Figure 5.3, which means the first rotation is clockwise around the z_L -axis, the second is clockwise around the y_2 -axis, and the third and last rotation is clockwise around the x_1 -axis.

A combined rotation matrix for all rotations between the two frames is wanted such that a vector defined in the local frame multiplied by the matrix yields the vector rotated to the global frame.

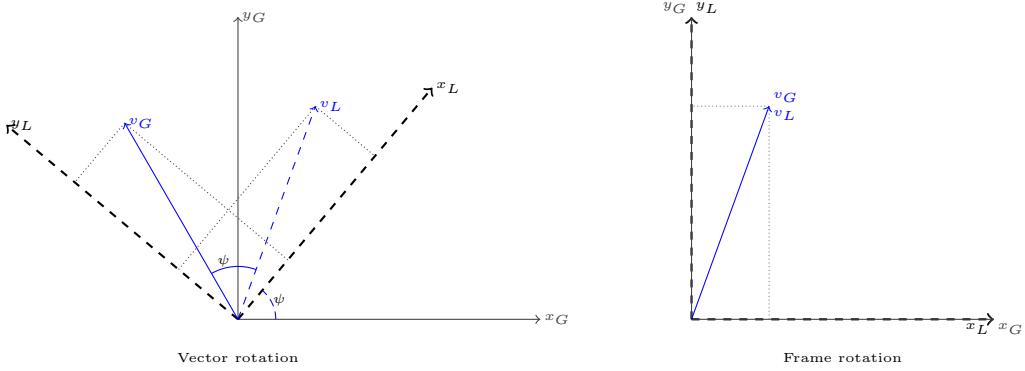


Figure 5.4: Two dimensional frame rotation by angle ψ (yaw) and the equivalent vector rotation.

When a vector is defined in the local frame, and the frame undergoes the reversed rotations clockwise to the global frame as described, it is the equivalent to rotating the vector counter-clockwise within the frame, as it is illustrated in Figure 5.4. Since the order of rotations is the same as for the frame rotations, the vector rotation from the local to the global frame must be counter-clockwise in the order Z-Y-X, so the rotation matrix for the complete rotation of a vector from the local frame to the global frame is

$$\underline{\underline{R}} = \underline{\underline{R}}_x \underline{\underline{R}}_y \underline{\underline{R}}_z = \begin{bmatrix} c_\theta c_\psi & -c_\theta s_\psi & s_\theta \\ c_\phi s_\psi + s_\phi s_\theta c_\psi & c_\phi c_\psi - s_\phi s_\theta s_\psi & -s_\phi c_\theta \\ s_\phi s_\psi - c_\phi s_\theta c_\psi & s_\phi c_\psi + c_\phi s_\theta s_\psi & c_\phi c_\theta \end{bmatrix} \quad [-] \quad (5.6)$$

Multiplying this rotation matrix with a vector defined in the local frame, the vector is rotated into the global frame.

The Euler rates $\dot{\chi}$ are not orthogonal, since the rotations by θ and ψ are made on an already rotated frame. The transformation matrix $\underline{\underline{Q}}$ transforms the Euler rates to obtain the orthogonal local frame angular velocity vector, and is given by Equation 5.7 [27].

$$\underline{\underline{Q}} = \begin{bmatrix} c_\theta c_\psi & s_\psi & 0 \\ -c_\theta s_\psi & c_\psi & 0 \\ s_\theta & 0 & 1 \end{bmatrix} \quad [-] \quad (5.7)$$

The transformation from Euler rates to angular velocity is carried out by

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} c_\theta c_\psi & s_\psi & 0 \\ -c_\theta s_\psi & c_\psi & 0 \\ s_\theta & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad [\text{rad s}^{-1}] \quad (5.8)$$

and the inverse transform is found to be

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \underline{\underline{Q}}^{-1} \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \frac{1}{c_\theta} \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi c_\theta & c_\psi c_\theta & 0 \\ -c_\psi s_\theta & s_\psi s_\theta & c_\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad [\text{rad s}^{-1}] \quad (5.9)$$

where the angular velocities are transformed to the Euler rates [27]. The implementation in Simulink is seen in Figure 5.5 below.

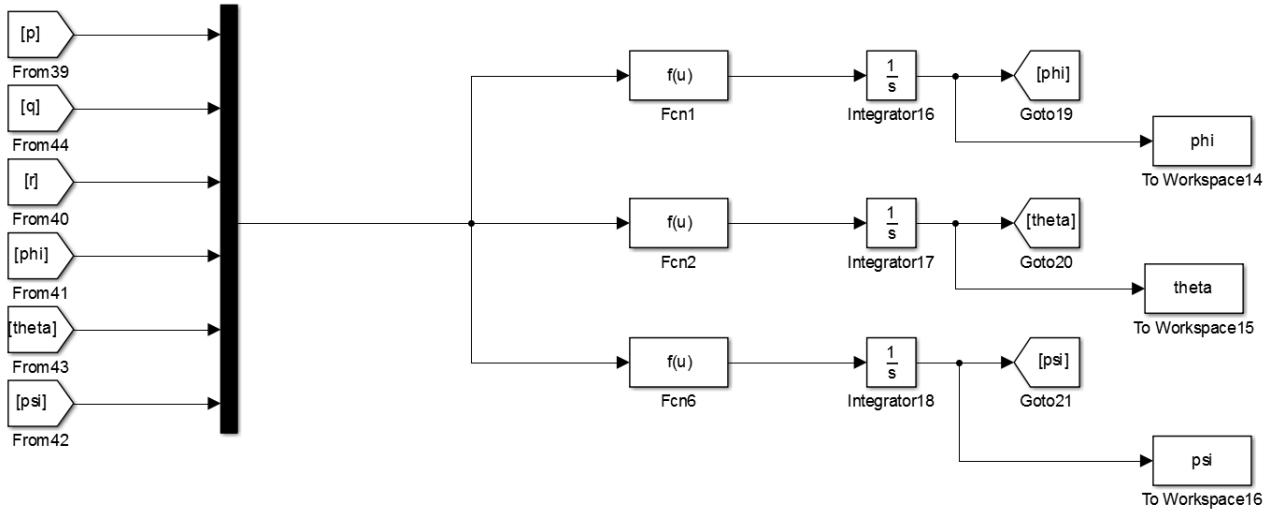


Figure 5.5: Angular velocity to Euler Rates in Simulink based on Equation 5.9

The three functions in Figure 5.5 defines the x -, y -, and z -component of the resultant vector from Equation 5.9. With the rotations between the frames and relevant transformations defined, the acting forces on the quadcopter is analyzed.

5.1.1 FORCES

The acting forces on the quadcopter are the four propeller thrusts ($\mathbf{F}_{I,n}$, $n = 1, 2, 3, 4$), the gravitational force (\mathbf{F}_g) as illustrated in Figure 5.6. The linear drag force ($\mathbf{F}_{d,\text{lin}}$), and the drag torque induced when rotating (τ_d) are not illustrated.

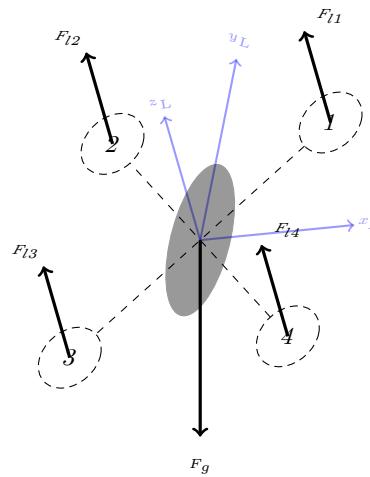


Figure 5.6: The acting forces on the quadcopter.

The many factors involved in determining the drag makes most simplified methods for calculating these rough estimates, and they are in general very hard to determine without experimental measurements. It was not possible

to experimentally determine the drag effects in this project, so in order to include drag effects in the model, some estimations are made to determine these:

The linear drag force is assumed to be proportional to the linear velocity by some constant B_{lin} , shown in Equation 5.10. B_{lin} is chosen based on the rough estimation that the linear drag force on the quadcopter is 10 [gram] when the velocity is one meter per second, and any change in the drag force due to a change in the angle at which the wind passes the quadcopter is neglected.

$$\underline{\mathbf{F}_{d,lin}} = \begin{bmatrix} F_{d,x} \\ F_{d,y} \\ F_{d,z} \end{bmatrix} = \begin{bmatrix} B_{lin}x\dot{x}_L \\ B_{lin}y\dot{y}_L \\ B_{lin}z\dot{z}_L \end{bmatrix} \quad [\text{N}] \quad (5.10)$$

The drag torque τ_d is exerted about CG. The drag torque is similarly assumed to be proportional to the angular velocity by some constant B_{ang} , shown in Equation 5.11. B_{ang} is determined from the rough estimation that the drag torque, when rotating 1 [rad s^{-1}], is approximately 5×10^{-3} [N m].

$$\underline{\boldsymbol{\tau}_d} = \begin{bmatrix} \tau_{d,\phi} \\ \tau_{d,\theta} \\ \tau_{d,\psi} \end{bmatrix} = \begin{bmatrix} B_{ang}p \\ B_{ang}q \\ B_{ang}r \end{bmatrix} \quad [\text{N m}] \quad (5.11)$$

These are, as mentioned, rough estimates which excludes any geometric analysis and non-linear contributions, and are estimated to include some kind of drag effects to the model, which will prove necessary for the control system to be stable.

In order to derive equations for the propeller thrusts, the angular velocity of the propeller is firstly defined to be

$$\underline{\boldsymbol{\omega}_n} = \begin{bmatrix} 0 \\ 0 \\ \omega_{z,n} \end{bmatrix} \quad [\text{rad s}^{-1}] \quad (5.12)$$

where $\omega_{z,n}$ is the spin rate of the propeller in the local xy -plan. The local frame propeller thrust $\underline{\mathbf{F}_{l,n}}$ is derived in Equation B.9 in Appendix B using blade element theory, and is summed for an expression of the combined lift from all propellers to be

$$\underline{\mathbf{F}_{l,L}} = \sum_{n=1}^4 \underline{\mathbf{F}_{l,n}} = k_l \sum_{n=1}^4 \underline{\boldsymbol{\omega}_n}^2 = \begin{bmatrix} 0 \\ 0 \\ k_l \sum_{n=1}^4 (\omega_{z,n}^2) \end{bmatrix} \quad [\text{N}] \quad (5.13)$$

where $\underline{\mathbf{F}_{l,L}}$ is the resultant thrust vector from all four propellers which is always in the positive local frame z_L -direction. The resultant thrust is rotated into the global frame by

$$\underline{\mathbf{F}_{l,G}} = \underline{\mathbf{R}} \underline{\mathbf{F}_{l,L}} \quad [\text{N}] \quad (5.14)$$

where $\underline{\mathbf{F}_{l,G}}$ is the resultant thrust from the four propellers in the global frame. The linear drag force is rotated to the global frame by

$$\underline{\mathbf{F}_{d,lin,G}} = \underline{\mathbf{R}} \underline{\mathbf{F}_{d,lin}} \quad [N] \quad (5.15)$$

where $\underline{\mathbf{F}_{d,lin,G}}$ is the linear drag force in the global frame. The gravitational force $\underline{\mathbf{F}_g}$ is known in the global frame z -direction as

$$\underline{\mathbf{F}_g} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} m \quad [N] \quad (5.16)$$

where g is the gravitational acceleration constant and m is the mass of the quadcopter. The motion of the quad is then defined in the global frame from Newton's second law of motion to be

$$\begin{aligned} \ddot{\underline{\mathbf{S}}}m &= \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} m = \underline{\mathbf{F}_{1,G}} - \underline{\mathbf{F}_{d,lin,G}} + \underline{\mathbf{F}_g} \\ &= \underline{\mathbf{R}} \begin{bmatrix} 0 \\ 0 \\ k_l \sum_{n=1}^4 \omega_{z,n}^2 \end{bmatrix} - \underline{\mathbf{R}} \begin{bmatrix} B_{lin}x_L \dot{z} \\ B_{lin}y_L \dot{z} \\ B_{lin}z_L \dot{z} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} m \end{aligned} \quad [N] \quad (5.17)$$

where \ddot{x} , \ddot{y} and \ddot{z} are the accelerations of the quadcopter along the x , y , and z -axes in the global frame. Isolating for $\ddot{\underline{\mathbf{S}}}$ yields the first equation of motion, which is

$$\ddot{\underline{\mathbf{S}}} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \underline{\mathbf{R}} \begin{bmatrix} 0 \\ 0 \\ k \sum_{n=1}^4 \omega_{z,n}^2 \end{bmatrix} \frac{1}{m} - \underline{\mathbf{R}} \begin{bmatrix} B_{lin}x_L \dot{z} \\ B_{lin}y_L \dot{z} \\ B_{lin}z_L \dot{z} \end{bmatrix} \frac{1}{m} + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \quad [\text{m s}^{-2}] \quad (5.18)$$

and its implementation in Simulink is illustrated in Figure 5.7

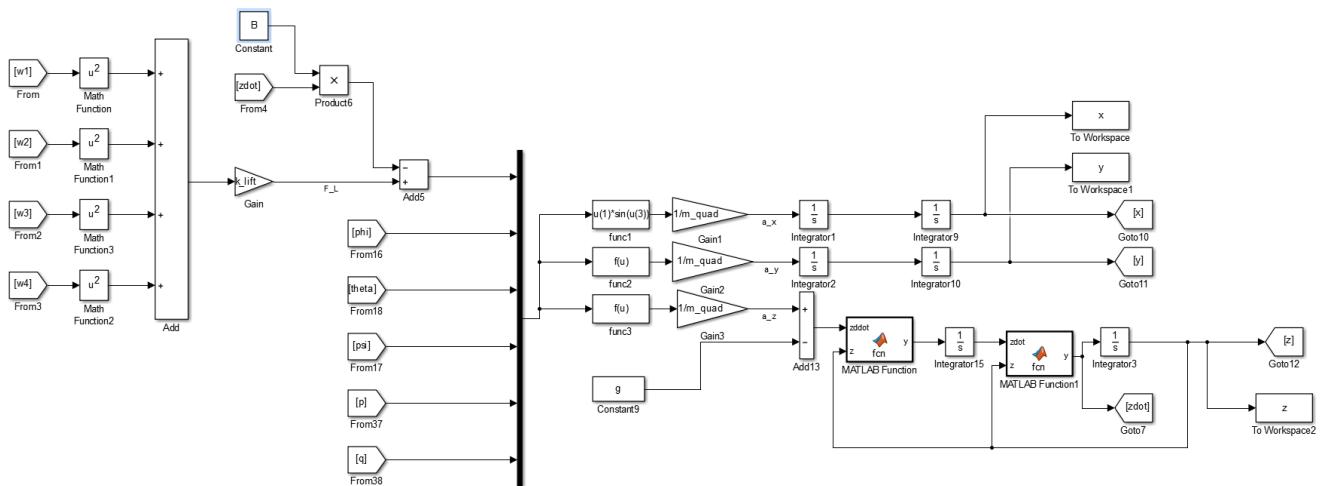


Figure 5.7: Force system in Simulink

5.1.2 TORQUES

With the linear equation of motion defined, the angular motion of the quadcopter can be described. In addition to the thrust, the propeller also applies a torque on CG of the quadcopter. The direction and magnitude of the torque depends entirely on the thrust from and drag on each propeller and the distance to CG. In general, when a torque and a force is applied at a point P , it is related to CG by

$$\underline{\tau}_{CG} = \underline{\tau}_P + \underline{r} \times \underline{F} \quad [\text{N m}] \quad (5.19)$$

where $\underline{\tau}_P$ is the torque vector at the point of applied force P , \underline{r} is the length vector from CG to P , and \underline{F} is the applied force vector [28]. Figure 5.8 illustrates a geometry with four applied forces with all angles and lengths relevant for the torque calculations shown.

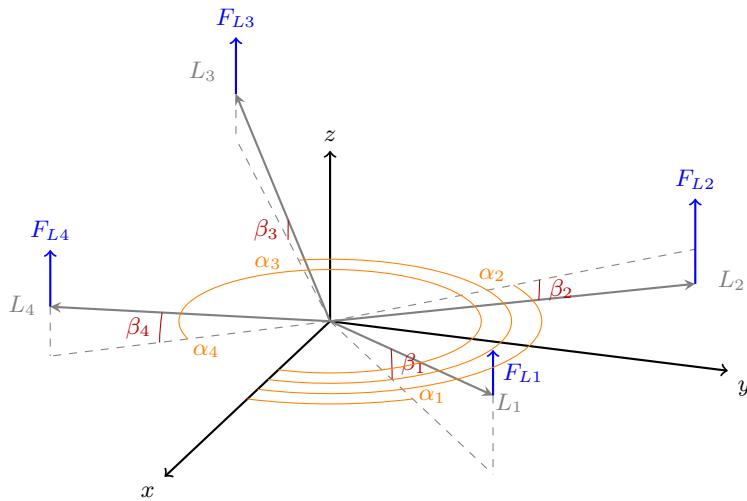


Figure 5.8: Illustration of a non-uniform geometry with four applied forces as well as the relevant lengths and angles.

The geometry illustrated in Figure 5.8 is used as the template for defining the torques on the quadcopter, where the length vectors L_n are the arm length vectors from CG to the propeller center, and the force vectors F_n are the propeller lifts. The local frame torque from propeller n on CG can be derived from Equation 5.19 to be

$$\underline{\tau}_{CG,n} = \underline{\tau}_{p,n} + \underline{L}_n \times \underline{F}_n \quad [\text{N m}] \quad (5.20)$$

where $\underline{\tau}_{p,n}$ is the torque on the propeller shaft by propeller n , $\underline{\tau}_{CG,n}$ is the torque on CG from propeller n , and \underline{L}_n is the length vector from CG to the center of propeller n . The angles α_n and β_n illustrated in Figure 5.8 are utilized to define the x -, y -, and z -component of \underline{L}_n to be

$$\begin{aligned} L_{x,n} &= ||\underline{L}_n|| \cos(\alpha_n) \\ L_{y,n} &= ||\underline{L}_n|| \sin(\alpha_n) \\ L_{z,n} &= ||\underline{L}_n|| \sin(\beta_n) \end{aligned} \quad [\text{m}] \quad (5.21)$$

where $L_{x,n}$, $L_{y,n}$, and $L_{z,n}$ is the x -, y -, and z -component of the length vector to propeller n , respectively. The propeller torque on the shaft ($\underline{\tau}_{\mathbf{p},n}$) is defined by an expanded Euler equation of rotational motion, which includes the drag torque and gyroscopic motion torque of the propeller. The drag torque on the propeller is defined in Equation B.11 in Appendix B using blade element theory, as

$$\underline{\tau}_{\mathbf{d},n} = b_n \underline{\omega}_n^2 \quad [\text{N m}] \quad (5.22)$$

where τ_d is the drag torque exerted on the propeller, b is a drag constant and ω_n is the propeller rotational rate in the local frame xy -plan defined in Equation 5.12. The gyroscopic torque is a product of the propellers being rotated around a different axis (x_L - and y_L -axis) than the axis it spins around (z_L -axis). In other terms, when the propellers are rotated with the quadcopter by the angular velocity $\underline{\nu}$, and additionally when they are spinning by a rate of $\underline{\omega}_n$, a gyroscopic motion torque is produced, which is given by Equation 5.23 [28].

$$\underline{\tau}_{\mathbf{gyro}} = \left(\underline{\underline{\mathbf{J}}}_{\mathbf{pr}} \underline{\nu} \right) \times \underline{\omega}_n \quad [\text{N m}] \quad (5.23)$$

The inertia matrix $\underline{\underline{\mathbf{J}}}_{\mathbf{pr}}$ is the combined inertia of the propeller and rotor, and is defined based to be

$$\underline{\underline{\mathbf{J}}}_{\mathbf{pr}} = \begin{bmatrix} J_{x,pr} & 0 & 0 \\ 0 & J_{y,pr} & 0 \\ 0 & 0 & J_{z,pr} \end{bmatrix} \quad [\text{kg m}^2] \quad (5.24)$$

where $J_{x,pr}$, $J_{y,pr}$, and $J_{z,pr}$ is the inertia around each axis, which is determined by the pendulum experiment in Appendix D. It is clear from Equation 5.23, that since the spin of the propellers $\underline{\omega}_n$ is only defined around the z_L -axis, the quadcopter angular velocity around z (r) has no contribution to the gyroscopic motion torque, while pitching by positive p applies a negative torque around the y_L -axis and rolling by positive q applies a positive torque around the x_L -axis. The expanded Euler equation of rotational motion becomes

$$\underline{\tau}_{\mathbf{p},n} = \underline{\underline{\mathbf{J}}}_{\mathbf{pr}} \dot{\underline{\omega}}_n + b_n \underline{\omega}_n^2 - \left(\underline{\underline{\mathbf{J}}}_{\mathbf{pr}} \underline{\nu} \right) \times \underline{\omega}_n \quad [\text{N m}] \quad (5.25)$$

where the first part is the torque due to acceleration of the propeller, the second is the drag torque from Equation 5.22, and the third part is the gyroscopic torque from the propeller spinning in the local frame xy -plan [28]. Inserting the expression for $\underline{\tau}_{\mathbf{p},n}$ in Equation 5.20 and summing for all four propellers, the total torque produced by all propellers on CG is

$$\underline{\tau}_{\mathbf{CG}} = \sum_{n=1}^4 \left(\underline{\underline{\mathbf{J}}}_{\mathbf{pr}} \dot{\underline{\omega}}_n + b_n \underline{\omega}_n^2 - \underline{\underline{\mathbf{J}}}_{\mathbf{pr}} \underline{\nu} \times \underline{\omega}_n + \underline{\mathbf{L}}_n \times \underline{\mathbf{F}}_n \right) \quad [\text{N m}] \quad (5.26)$$

With an expression of the torque produced on CG by all propellers defined, the torque is now related to the motion of the local frame (quadcopter). Once again, Euler's equation of rotational motion is utilized, where the torque contribution from the angular drag is included. The equation for the quadcopters rotational motion is given as

$$\underline{\tau}_{\mathbf{CG}} = \underline{\mathbf{J}} \dot{\underline{\nu}} + \underline{\nu} \times (\underline{\mathbf{J}} \underline{\nu}) + \underline{\tau}_{\mathbf{d}} \quad [\text{N m}] \quad (5.27)$$

5.2. MOTOR DYNAMICS

where $\dot{\underline{\nu}}$ is the local frame angular acceleration, and $\underline{\mathbf{J}}$ is the inertia matrix of the quadcopter, which is of same diagonal structure as the propeller inertia, and is experimentally determined in Appendix D. By isolating $\dot{\underline{\nu}}$ in Equation 5.27, the equation becomes

$$\dot{\underline{\nu}} = \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \underline{\mathbf{J}}^{-1}(\underline{\tau}_{CG} - \underline{\nu} \times (\underline{\mathbf{J}}\underline{\nu}) - \underline{\tau}_d) \quad [\text{rad s}^{-2}] \quad (5.28)$$

The calculated local frame angular accelerations \dot{p} , \dot{q} and \dot{r} are now to be related to the global frame angular accelerations $\ddot{\phi}$, $\ddot{\theta}$ and $\ddot{\psi}$. From Equation 5.9 it is clear that

$$\begin{aligned} \ddot{\underline{x}} &= \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \frac{d}{dt} \left(\underline{\mathbf{Q}}^{-1} \underline{\nu} \right) \\ &= \left(\frac{d}{dt} \underline{\mathbf{Q}}^{-1} \right) \underline{\nu} + \underline{\mathbf{Q}}^{-1} \dot{\underline{\nu}} \end{aligned} \quad [\text{rad s}^{-2}] \quad (5.29)$$

and the second equation of motion is thereby obtained. The implementation of the torque estimations in Simulink is illustrated in Figure 5.9.

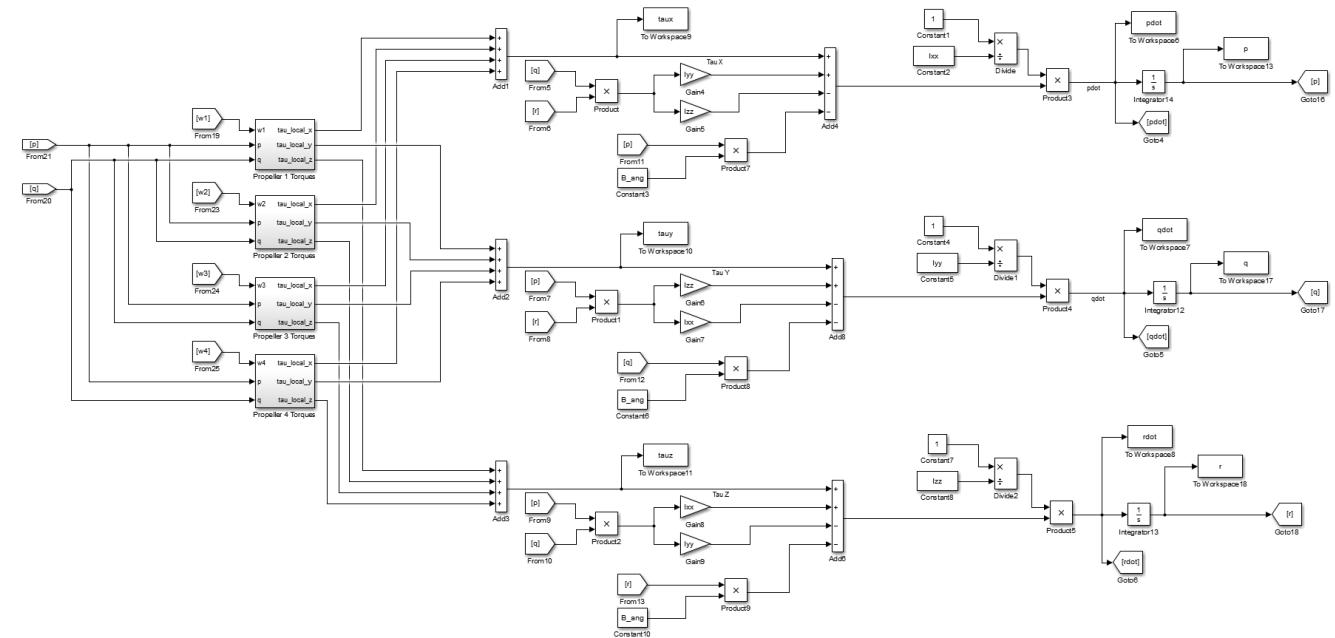


Figure 5.9: Torque estimation in Simulink

SECTION 5.2

MOTOR DYNAMICS

This section will describe a simplified model of the motors and ESCs to describe its dynamic response. A transfer function for the motors is needed, which is approximated from analysis of the step response of the motor.

In the same experimental setup as the one used for characterization of the motors, small steps is applied to the motor to analyze the angular velocity response from these steps. The steps can be seen in Figure 5.10

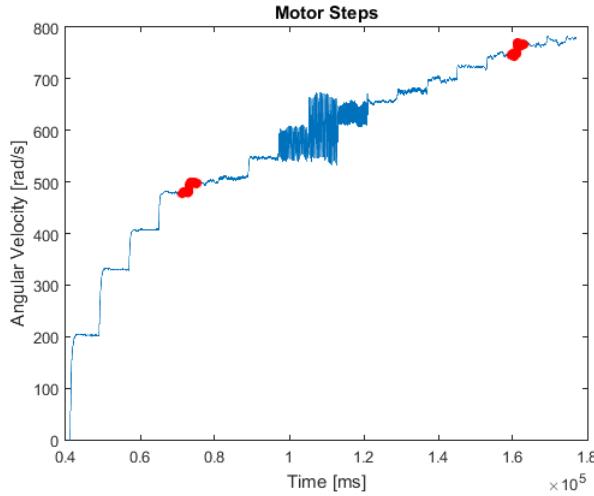
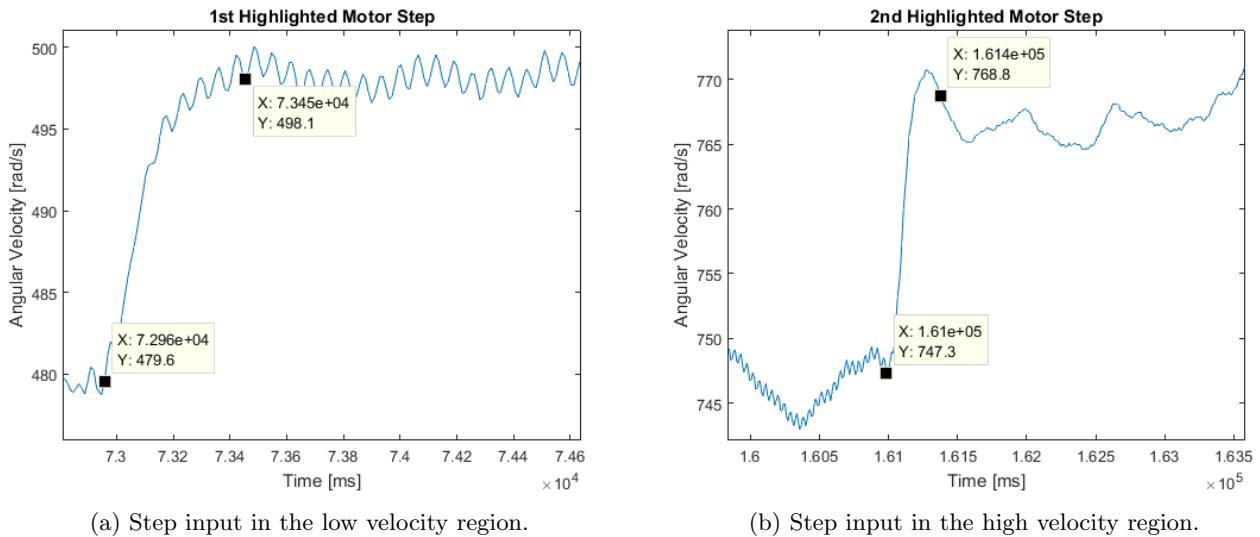


Figure 5.10: Rotor angular velocity from duty cycle steps.

It is seen on the figure that two steps has been highlighted in red. The step response varies from the high to the low angular velocity region, which is seen on the zoom of the two highlighted steps in Figure 5.11a and 5.11b. The highlighted choice of step to model further within the lower region is made based on Figure D.4 in Appendix D, where it is seen that the linear approximation of the angular velocity in steady state, as a function of duty cycle, only fits for angular velocities higher than 4500 [RPM] ($470 \text{ [rad s}^{-1}\text{]}$) for a battery voltage of 15.8 [V]. Since the dynamics of the motor will be related to a duty cycle as input when the motor controller is designed, the first step to be analyzed is chosen around $500 \text{ [rad s}^{-1}\text{]}$, as illustrated in Figure 5.10. The second step highlighted in red is simply chosen at the highest possible angular velocity, where the step data is still feasible.



(a) Step input in the low velocity region.

(b) Step input in the high velocity region.

Figure 5.11: Zoom of the highlighted steps.

Figure 5.11 illustrates the two highlighted steps in Figure 5.10. The step in the low velocity region shows no visible overshoot, but rather a small continuous oscillation after settling. In the high velocity region there is too much noise to identify a small overshoot, if one exists.

The transfer functions approximating these steps will therefore be chosen to be of first order, and are given by

$$G_{1.order}(s) = \frac{1}{\tau s + 1} \quad (5.30)$$

where τ is the time constant, which is the time from the step start to the time where the function reaches 63 [%] of the step value [29]. The settling time T_s is the time where the function has settled within 1.8 [%] of the step value, and is calculated from Equation 5.31 [29].

$$T_s = 4\tau \quad [\text{s}] \quad (5.31)$$

In the two steps shown in Figure 5.11, the two datapoints on each figure are:

- The approximated start time of the step
- Point where the angular velocity has settled

Reading from the graphs, the step in Figure 5.11a settles in $T_{s,step1} = 0.49$ [s], and the step in Figure 5.11b settles in $T_{s,step2} = 0.4$ [s]. The corresponding time constants are

$$\begin{aligned}\tau_1 &= \frac{T_{s,step1}}{4} = 0.1225 \\ \tau_2 &= \frac{T_{s,step2}}{4} = 0.1\end{aligned}$$

and the first order transfer functions are given in Equation 5.32.

$$\begin{aligned}G_{step1}(s) &= \frac{1}{\tau_1 s + 1} \\ G_{step2}(s) &= \frac{1}{\tau_2 s + 1}\end{aligned} \quad (5.32)$$

The transfer function G_{step1} has a single pole located at -8.16, and similarly G_{step2} has a pole at -10. For future modeling and controller design, the slowest case is chosen, such that the time response requirements for the pitch, roll, yaw, and altitude controllers are met at all times. The chosen transfer function is

$$G_\omega(s) = \frac{\omega}{\omega_{ref}} = \frac{1}{0.1225s + 1} = \frac{1}{\tau_m s + 1} \quad (5.33)$$

where the input is the reference rotor angular velocity ω_{ref} and the output is the actual rotor angular velocity ω .

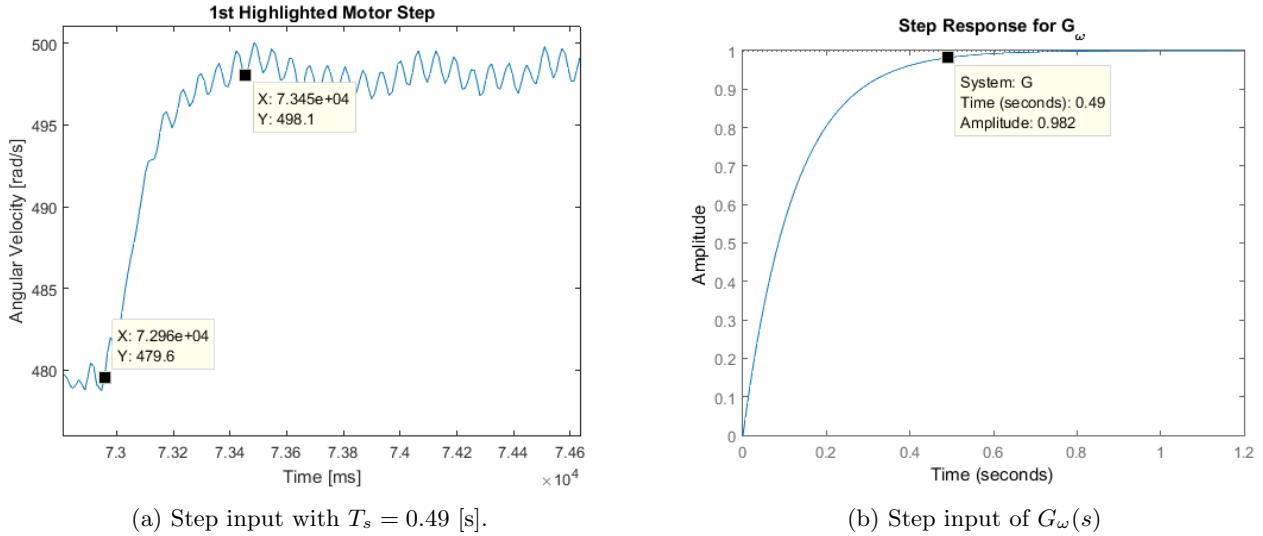


Figure 5.12: Step on the motor compared to step on the approximated first order transfer function $G_\omega(s)$.

Figure 5.12 illustrates the step response of $G_\omega(s)$ compared to the chosen step to derive the first order transfer function from. It is seen that $G_\omega(s)$ settles as it should at 0.49 [s], and is assumed to be a feasible approximation for the motor dynamics.

In Section 4.3 the ESCs and their function on the quadcopter is described, and it is recalled that the input to the ESCs, which will control the angular velocity of the motors, is a duty cycle. The wanted angular velocity input of $G_\omega(s)$ is to be related to the duty cycle in steady state, such that the motor transfer function will be of duty cycle input and actual angular velocity output.

The relation between duty cycle and angular velocity in steady state was characterized in Appendix D.3 in Equation D.7. Since this expression has the battery voltage as an additional variable, which varies from 16.8 [V] when fully charged to 14.8 [V] when depleted, which was chosen to be the lowest accepted voltage for operation, the mean voltage (15.8 [V]) is chosen for further calculations. Inserting the chosen voltage in Equation D.7 and solving for the duty cycle gives

$$D = \frac{\omega_{ss} - 329.13}{405.265} = \frac{1}{405.265} \omega_{ss} - 0.812 \quad (5.34)$$

where ω_{ss} is the steady state rotor angular velocity from the duty cycle. To obtain a transfer function with angular velocity ω as input and duty cycle D as output, Equation 5.34 is linearized around the constant -0.812 and the equation becomes

$$D = \frac{1}{405.265} \omega_{ss} \quad (5.35)$$

where D is observed to be proportional to ω . The equation is divided by ω and is Laplace transformed to obtain the transfer function G_D in Equation 5.36.

$$G_D(s) = \frac{D(s)}{\omega_{ss}(s)} = \frac{1}{405.265} = K_{DC} \quad (5.36)$$

The wanted motor transfer function with the duty cycle D as input can now be calculated by multiplying G_D to the input ω_{ref} , which is done in Equation 5.37.

$$G_{motor}(s) = \frac{\omega(s)}{D(s)} = G_D(s)^{-1}G_\omega(s) = \frac{K_{DC}^{-1}}{\tau_m s + 1} \quad (5.37)$$

A simple transfer function describing the dynamics of the angular velocity of the rotor from a duty cycle input is derived. This is to be utilized for designing the different controllers for the system in the following chapter.

SECTION 5.3**CONCLUSION**

A model has been made and implemented into Simulink which simulates the behavior of a quadcopter with four propellers. The model makes it possible to test controllers and their effect on the system while getting different information about the angular velocity of each propeller during flight. The model takes into account the dynamics of the specific ESC, motor, and propeller obtained for the project and thereby serves as an especially good indicator for the response of the constructed quadcopter in different scenarios when a controller is implemented.

CHAPTER 6

CONTROL SYSTEM

In Chapter 5 the fundamental structure of the quadcopter model was illustrated in Figure 5.1, and this chapter will focus on the contents and design of the "Control" block. A block-diagram of the control system to be designed is seen in Figure 6.1. Chapter 5 also provided a mathematical model of the quadcopter which has been implemented in Simulink with the purpose of simulating the motion of the quadcopter in different scenarios with different controller designs. A controller is to be designed based on this mathematical model, which will be implemented in Simulink to investigate the performance of said controller. The main requirement of the controller is to effectively keep the quadcopter hovering in the air at a specific altitude.

In order to do so, the controller must handle the following states:

- Pitch (ϕ)
- Roll (θ)
- Yaw (ψ)
- Altitude (z)

By controlling the angular position of the quadcopter, it is possible to keep it stable, and by additionally controlling the altitude, the quadcopter can be kept still and hovering at a specific altitude.

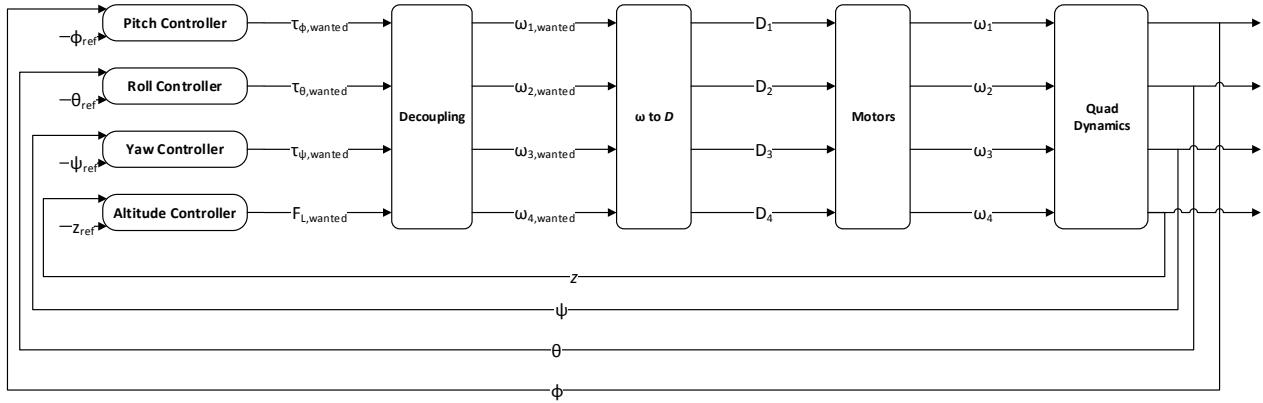


Figure 6.1: Control System.

Figure 6.1 illustrates four individual controllers to be designed, controlling pitch, roll, yaw and altitude. The inputs to each controller is the error calculated between the current state derived from the sensors and the reference signal. The output of the controllers is a wanted torque (or wanted lift for the altitude controller) for the system to be corrected by. The "Decoupling" block contains the set of equations relating the controller outputs to a wanted angular velocity ω for each motor, which will be described in a later section. The " ω to D " block converts the wanted angular velocity into a duty cycle for the motors to which a transfer function was derived in Section 5.2 in Equation 5.36. The "Motor" block contains the four identical transfer functions for the four motors, also derived in Section 5.2, and yields the rotor angular velocities which, through the quadcopter dynamics, gives the pitch, roll, yaw, and altitude state.

SECTION 6.1

SIMPLIFICATIONS

Initial simplifications can be made for designing the controller, since there is very little motion involved in hovering the quadcopter in the air.

In Section 5.1.2 the total torque on CG was derived in Equation 5.26 to be

$$\underline{\tau}_{\text{CG}} = \sum_{n=1}^4 \left(\underline{\mathbf{J}_{\text{pr}}} \underline{\omega_n} + b_n \underline{\omega_n}^2 - \underline{\mathbf{J}_{\text{pr}}} \underline{\nu} \times \underline{\omega_n} + \underline{\mathbf{L}_n} \times \underline{\mathbf{F}_n} \right)$$

where the four expressions in the summation are the torque due to accelerating the propeller, the drag torque, the gyroscopic torque and the torque due to propeller thrusts. When the quadcopter is hovering, the motors constantly make small changes to the rotors angular velocities to correct for errors in ϕ , θ , ψ , and z , and thereby keeping the quadcopter stationary. The accelerations and decelerations of the rotors are assumed to be small when hovering and are therefore neglected.

The first expression ($\underline{\mathbf{J}_{\text{pr}}} \underline{\omega_n}$) of Equation 5.26 is therefore neglected. Since the angular velocity of the quadcopter ν is similarly assumed to be small when hovering, the cross product (third expression) of Equation 5.26 is assumed to be small and is therefore neglected. Equation 5.26 is simplified to

$$\underline{\tau}_{\text{CG}} = \sum_{n=1}^4 (b_n \underline{\omega_n}^2 + \underline{\mathbf{L}_n} \times \underline{\mathbf{F}_n}) = \begin{bmatrix} k_l (l_{y,1} \omega_{z,1}^2 + l_{y,2} \omega_{z,2}^2 + l_{y,3} \omega_{z,3}^2 + l_{y,4} \omega_{z,4}^2) \\ -k_l (l_{x,1} \omega_{z,1}^2 - l_{x,2} \omega_{z,2}^2 - l_{x,3} \omega_{z,3}^2 - l_{x,4} \omega_{z,4}^2) \\ b_n (\omega_{z,1}^2 + \omega_{z,2}^2 + \omega_{z,3}^2 + \omega_{z,4}^2) \end{bmatrix} \quad (6.1)$$

From the symmetry in the geometry of the quadcopter, the lengths can be simplified to

$$|l_{y,1}| = |l_{y,2}| = |l_{y,3}| = |l_{y,4}| = l_y \quad (6.2)$$

$$|l_{x,1}| = |l_{x,2}| = |l_{x,3}| = |l_{x,4}| = l_x \quad (6.3)$$

and since b_n is negative for motor two and four, Equation 6.1 becomes

$$\underline{\tau}_{\text{CG}} = \begin{bmatrix} k_l l_y (\omega_{z,1}^2 + \omega_{z,2}^2 - \omega_{z,3}^2 - \omega_{z,4}^2) \\ k_l l_x (-\omega_{z,1}^2 + \omega_{z,2}^2 + \omega_{z,3}^2 - \omega_{z,4}^2) \\ |b_n| (-\omega_{z,1}^2 + \omega_{z,2}^2 - \omega_{z,3}^2 + \omega_{z,4}^2) \end{bmatrix} \quad (6.4)$$

The torque is related to the angular velocity of the quadcopter in Equation 5.28, which was

$$\dot{\underline{\nu}} = \underline{\mathbf{J}}^{-1} (\underline{\tau}_{\text{CG}} - \underline{\nu} \times (\underline{\mathbf{J}} \underline{\nu}) - \underline{\tau}_{\text{d}}) \quad (6.5)$$

Due to the angular velocity of the quadcopter being small when hovering, the cross product ($\underline{\nu} \times (\underline{\mathbf{J}} \underline{\nu})$) is neglected. The drag torque $\underline{\tau}_{\text{d}}$ from the angular velocity of the quadcopter is, however, not neglected, which will prove to be necessary if damping is to be present in the system (further described in Section 6.2). Equation 5.28 becomes

$$\dot{\underline{\nu}} = \underline{\mathbf{J}}^{-1} (\underline{\tau}_{\text{CG}} - \underline{\tau}_{\text{d}}) = \begin{bmatrix} (k_{ll_y}(\omega_{z,1}^2 + \omega_{z,2}^2 - \omega_{z,3}^2 - \omega_{z,4}^2) - B_{ang}p) / J_x \\ (k_{ll_x}(-\omega_{z,1}^2 + \omega_{z,2}^2 + \omega_{z,3}^2 - \omega_{z,4}^2) - B_{ang}q) / J_y \\ (|b_n|(\omega_{z,1}^2 - \omega_{z,2}^2 + \omega_{z,3}^2 - \omega_{z,4}^2) - B_{ang}r) / J_z \end{bmatrix} \quad (6.6)$$

It is clear from Equation 6.6 and Equation 5.29 in Section 5.1.2 that the states $\ddot{\phi}$, $\ddot{\theta}$, and $\ddot{\psi}$ is a function of the torques on CG, ie. τ_ϕ , τ_θ , and τ_ψ , and furthermore it is seen in Equation 5.18 in Section 5.1.1 that the state \ddot{z} is a function of the propeller thrusts. From this, four parameters are defined for convenience to be

$$\begin{aligned} u_1 &= \tau_\phi &= k_{ll_y}(\omega_{z,1}^2 + \omega_{z,2}^2 - \omega_{z,3}^2 - \omega_{z,4}^2) \\ u_2 &= \tau_\theta &= k_{ll_x}(-\omega_{z,1}^2 + \omega_{z,2}^2 + \omega_{z,3}^2 - \omega_{z,4}^2) \\ u_3 &= \tau_\psi &= |b_n|(\omega_{z,1}^2 - \omega_{z,2}^2 + \omega_{z,3}^2 - \omega_{z,4}^2) \\ u_4 &= \|F_L\| &= k_l(\omega_{z,1}^2 + \omega_{z,2}^2 + \omega_{z,3}^2 + \omega_{z,4}^2) \end{aligned} \quad (6.7)$$

which is used to express the state equations. Writing Equation 5.29 in Section 5.1.2 with the simplified $\dot{\underline{\nu}}$ in Equation 6.6 in terms of u_1 , u_2 , and u_3 , the angular acceleration states ($\ddot{\phi}$, $\ddot{\theta}$, and $\ddot{\psi}$) can be written separately as

$$\begin{aligned} \ddot{\phi} &= \frac{c_\psi}{J_x c_\theta} (u_1 - B_{ang}p) - \frac{\dot{\theta} (\dot{\psi} c_\theta - \dot{\theta} s_\theta + \dot{\theta} c_\psi s_\theta + \dot{\theta} c_\psi s_\psi s_\theta)}{c_\theta^2} - \frac{s_\psi}{J_y c_\theta} (u_2 - B_{ang}q) + \frac{(\dot{\phi} \dot{\theta} s_\psi s_\theta (c_\psi + s_\psi))}{c_\theta^2} \\ \ddot{\theta} &= \frac{s_\psi}{J_x c_\theta} (u_1 - B_{ang}p) + \frac{c_\psi}{J_y c_\theta} (u_2 - B_{ang}q) + \dot{\phi} \dot{\psi} c_\theta \\ \ddot{\psi} &= -\frac{c_\psi s_\theta}{J_x c_\theta} (u_1 - B_{ang}p) + \frac{s_\psi s_\theta}{J_y c_\theta} (u_2 - B_{ang}q) + \frac{1}{J_z} (u_3 - B_{ang}r) - \frac{\dot{\phi} \dot{\theta}}{c_\theta} + \frac{\dot{\psi} \dot{\theta} s_\theta}{c_\theta} \end{aligned} \quad (6.8)$$

where the states ϕ , θ , and ψ are obtained by integrating twice with respect to time. Similarly the linear acceleration \ddot{z} can be written in terms of u_4 from Equation 5.18 in Section 5.1.1, which is done below.

$$\ddot{z} = \frac{(c_\phi c_\theta (u_4 - B_{lin}\dot{z}) - B_{lin}\dot{z}(c_\psi s_\phi + c_\phi s_\psi s_\theta) - B_{lin}\dot{z}(c_\phi s_\psi - c_\phi c_\psi s_\theta) - gm)}{m} \quad (6.9)$$

6.1.1 SMALL ANGLE APPROXIMATION

The simplifications made so far are based on approximations of certain variables when the quadcopter is only hovering, and similarly the Euler angles can be approximated for this case.

When the quadcopter is hovering, the controllers will try to level the quadcopter in the air, making the Euler angles vary slightly around 0 [rad]. It is therefore assumed, that the Euler angles in the state equations can be approximated to be 0, while still being feasible for small angles.

Inserting $\phi\theta\psi = 0$ into the rotation matrix $\underline{\mathbf{R}}$ and the equivalent transformation matrix $\underline{\mathbf{Q}}$ derived in Chapter 5 gives

$$\underline{\mathbf{Q}}_{\phi\theta\psi=0} = \underline{\mathbf{R}}_{\phi\theta\psi=0} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where it is noted, that multiplying a vector with any of the matrices equals the same vector. This is also confirmed when inserting $\phi\theta\psi = 0$ into the simplified state equations in 6.8 and 6.9, which yields

$$\left. \begin{array}{l} \ddot{\phi}_{\phi\theta\psi=0} = \frac{u_1 - B_{ang}p}{J_x} = \frac{u_1 - B_{ang}\dot{\phi}}{J_x} \\ \ddot{\theta}_{\phi\theta\psi=0} = \frac{u_2 - B_{ang}q}{J_y} = \frac{u_2 - B_{ang}\dot{\theta}}{J_y} \\ \ddot{\psi}_{\phi\theta\psi=0} = \frac{u_3 - B_{ang}r}{J_z} = \frac{u_3 - B_{ang}\dot{\psi}}{J_z} \\ \ddot{z}_{\phi\theta\psi=0} = \frac{u_4 - B_{lin}z - gm}{m} \end{array} \right\} \quad (6.10)$$

where all sines, cosines, and Euler rates have disappeared. To examine the assumption the angles ϕ , θ , and ψ are all set to 5 [deg] and the resultant two matrices, $\underline{\underline{\mathbf{R}}}$ and $\underline{\underline{\mathbf{Q}}}$, is written below.

$$\left. \begin{array}{l} \underline{\underline{\mathbf{R}}} = \begin{bmatrix} 0.992 & -0.087 & 0.087 \\ 0.094 & 0.992 & -0.087 \\ -0.079 & 0.094 & 0.992 \end{bmatrix} \\ \underline{\underline{\mathbf{Q}}} = \begin{bmatrix} 0.992 & 0.087 & 0 \\ -0.087 & 0.996 & 0 \\ 0.087 & 0 & 1 \end{bmatrix} \end{array} \right\} \quad (6.11)$$

It is apparent that the entries in the matrices varies by less than 0.1 compared to the case for $\phi\theta\psi = 0$. Since the variables are all values from 0 to 1 (from cosines and sines), the entries change in the matrices is thereby less than 10 [%], and the resultant small change in the state equations from these angles (compared to $\phi\theta\psi = 0$) is assumed to be negligible. The Euler angles are assumed to be less than 5° at all times when hovering, it is assumed that the small angle approximation holds up, and that the state equations in 6.10 is usable within the boundaries of controlling the quadcopter when hovering.

SECTION 6.2

LINEARIZATION

The intention is to control ϕ , θ , ψ , and z with linear control. The aim of this section is therefore to find the linearized relations between the accelerations $\ddot{\phi}$, $\ddot{\theta}$, $\ddot{\psi}$, \ddot{z} and the terms u_1 , u_2 , u_3 , u_4 .

In general, linearization of a nonlinear function $f(x)$ about a point x_0 is done by a Taylor series expansion given by Equation 6.12 [29].

$$f(x) \approx f(x_0) + \frac{\partial f(x)}{\partial x}|_{x=x_0} \cdot (x - x_0) \quad (6.12)$$

Equation 6.9 is linearized using Equation 6.12 around the global frame hovering lift force $u_{4,0} = mg$ as well as $\dot{z}_0 = 0$, and the result is written in Equation 6.13.

$$\Delta\ddot{z} = \frac{\Delta u_4 - B_{lin}\Delta\dot{z}}{m} \quad (6.13)$$

Similarly linearization can be done for the state equations $\ddot{\phi}$, $\ddot{\theta}$, and $\ddot{\psi}$ around the torques $u_{1,0} = u_{2,0} = u_{3,0} = 0$ and around the Euler rates $\dot{\phi} = \dot{\theta} = \dot{\psi} = 0$, which are based on hovering conditions of the quadcopter. The linearized angular accelerations $\ddot{\phi}$, $\ddot{\theta}$, and $\ddot{\psi}$ is written below.

$$\left. \begin{array}{l} \Delta\ddot{\phi} = \frac{\Delta u_1 - B_{ang}\Delta\dot{\phi}}{J_x} \\ \Delta\ddot{\theta} = \frac{\Delta u_2 - B_{ang}\Delta\dot{\theta}}{J_y} \\ \Delta\ddot{\psi} = \frac{\Delta u_3 - B_{ang}\Delta\dot{\psi}}{J_z} \end{array} \right\} \quad (6.14)$$

Each state can now be written in their respective Laplace transforms, which is done below.

$$\left. \begin{array}{l} \Delta\phi s^2 = \frac{\Delta u_1(s) - B_{ang}\Delta\phi(s)s}{J_x} \\ \Delta\theta s^2 = \frac{\Delta u_2(s) - B_{ang}\Delta\theta(s)s}{J_y} \\ \Delta\psi s^2 = \frac{\Delta u_3(s) - B_{ang}\Delta\psi(s)s}{J_z} \\ \Delta z s^2 = \frac{\Delta u_4(s) - B_{lin}\Delta z s}{m} \end{array} \right\} \quad (6.15)$$

It is now possible to derive the transfer functions for the states ϕ , θ , ψ , and z by integrating and drawing the block diagram representing the states, which is done for the pitch state in Figure 6.2. The remaining three states have identical diagram stricture.

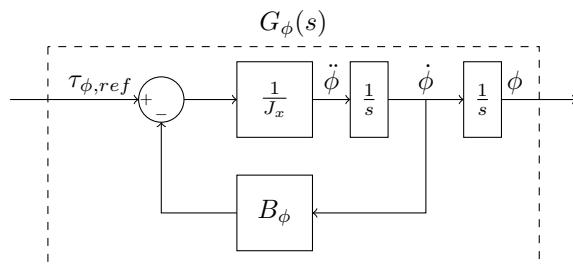


Figure 6.2: Block diagram representing the pitch state.

By block-diagram theory the transfer function for each state is written in Equation 6.16.

$$\left. \begin{aligned} G_\phi(s) &= \frac{\Delta\phi}{\Delta u_1} = \frac{1}{J_x s^2 + B_{ang}s} \\ G_\theta(s) &= \frac{\Delta\theta}{\Delta u_2} = \frac{1}{J_y s^2 + B_{ang}s} \\ G_\psi(s) &= \frac{\Delta\psi}{\Delta u_3} = \frac{1}{J_z s^2 + B_{ang}s} \\ G_z(s) &= \frac{\Delta z}{\Delta u_4} = \frac{1}{m s^2 + B_{lin}s} \end{aligned} \right\} \quad (6.16)$$

Equations 6.16 shows the final transfer functions which are going to be used in designing the controllers.

SECTION 6.3

DECOUPLING

Based on the simplifications made, it is assumed that a torque around one axis does not cause ant torque around another axis. This is supported bu Equation 6.15, where the states are independent of each other. Based on tis assumption it is assumed that the wanted torque and lift can be decoupled to an angular velocity on each of the four motors.

The torques (u_1 , u_2 , and u_3) around the three axes and lift force (u_4) are given by:

$$\left. \begin{array}{l} u_1 = k_l l_y (\omega_1^2 + \omega_2^2 - \omega_3^2 - \omega_4^2) \\ u_2 = k_l l_x (-\omega_1^2 + \omega_2^2 + \omega_3^2 - \omega_4^2) \\ u_3 = |b| (\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \\ u_4 = k_l (\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \end{array} \right\} \quad (6.17)$$

where each parameter is a function of different sums of the squared angular velocity of the motors. The four equations in 6.17 with the four unknowns can be expressed on matrix form as

$$\begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \\ F_L \end{bmatrix} = \underline{\underline{\text{DCM}}} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \quad (6.18)$$

where **DCM** is a 4x4 matrix given in Equation 6.19.

$$\underline{\underline{\text{DCM}}} = \begin{bmatrix} k_l l_y & k_l l_y & -k_l l_y & -k_l l_y \\ -k_l l_x & k_l l_x & k_l l_x & -k_l l_x \\ b & -b & b & -b \\ k & k & k & k \end{bmatrix} \quad (6.19)$$

The decoupling matrix is identified as the inverse of

$$\underline{\underline{\text{DCM}_d}} = \begin{bmatrix} 1 & 1 & -1 & -1 \\ -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad (6.20)$$

since the inverse of **DCM** effectively decouples the sum of squared angular velocities corresponding to each state into a squared angular velocity for each motor, which is shown in the relation below.

$$\begin{bmatrix} \omega_1^2 + \omega_2^2 - \omega_3^2 - \omega_4^2 \\ -\omega_1^2 + \omega_2^2 + \omega_3^2 - \omega_4^2 \\ \omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2 \\ \omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & -1 & -1 \\ -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \quad (6.21)$$

Solving Equation 6.18 for the angular velocity vector gives the relation between the wanted torque/lift force and the corresponding angular velocity on each of the four motors.

$$\begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} = \underline{\underline{\text{DCM}}}^{-1} \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \\ F_L \end{bmatrix} \quad (6.22)$$

To obtain the angular velocity on the motors the square root have to be taken. Since the quadcopter is operating at hover angular velocities on the motors, the square root will not yield an imaginary number. The angular velocity on the four motors are related to an equivalent duty cycle to each of the ESCs.

SECTION 6.4

REGULATOR DESIGN

All the relevant transfer functions for the control system illustrated in Figure 6.1 has been derived, and the pitch, roll, yaw, and altitude regulators are now to be designed from these.

PID REGULATOR

A short introduction will be made of a PID regulator, and the reasons behind the choice of this regulator will be explained. Proportional-Integral-Derivative (PID) control is one of the most commonly used control algorithms in industrial control [30]. Reasons why the PID regulator is widely used includes a relative robust performance and more flexibility compared to e.g. phase-lag and phase-lead regulators [29]. The PID regulator consists of a proportional, integral, and derivative term as seen in the transfer function of a PID regulator in Equation 6.23.

$$G_{PID}(s) = K_p + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + K_p s + K_i}{s} \quad (6.23)$$

Each term of the regulator has its own effect on the system [31]:

Effects of proportional term:

- Increased system gain if $K_p > 1$, but a more oscillatory system is obtained and the gain margin and phase margin are reduced
- Faster transient response/higher bandwidth.
- No change in phase.

Effects of integral term:

- Adds a pole to the system (increases system type by 1) and improves stationary response.
- Slows the transient response.
- Adds -90° to the phase, which will have a destabilizing effect.
- Reduces high frequency noise since the error signal is integrated.

Effects of differential term:

- Adds 90° to the phase, which will have a stabilizing effect.
- Faster transient response.
- Only contributes for varying signals.
- The differential gain yields infinity high frequency gain. Very sensitive to high frequency noise.

Some of the effects on the closed loop system from the PID constants are summarized in Table 6.1.

CL Response	Rise Time	Overshoot	Settling Time	SS error
K_p	Decrease	Increase	Small change	Decrease
K_i	Decrease	Increase	Increase	Eliminate
K_d	Small change	Decrease	Decrease	No change

Table 6.1: Table showing effect increasing constants

The four transfer functions for the four states defined in Equation 6.16 in Section 6.2 all contain a free integrator, which is also illustrated in the pitch example in Figure 6.2 of the same section. The free integrator makes the systems of type one, and also has the effect of eliminating the steady state error of the closed loop system without the help of an integral term in the regulator. This implies that the integral term of the PID regulator is unnecessary, since the other effects of the integral term, e.g. the slowed transient response, are unwanted for the case of this project. However, since the differential term is sensitive to high frequency noise in the signal (which is to be expected from the sensors in the physical system), a PD regulator with no integral term is assumed to be too sensitive to high frequency noise, since any noise in the sensor signals merely gained to the output. While a PID regulator still has the noise-sensitive derivative term, it is less dominant in the regulator output signal with the presence of the additional, and much less noise-sensitive, integrator term. The PID regulator is therefore chosen for this project.

6.4.1 PITCH PID

The pitch control loop is illustrated in Figure 6.3

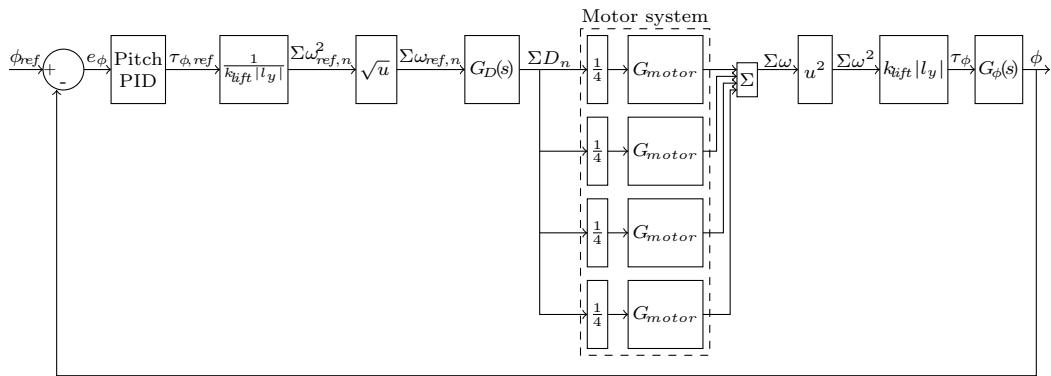


Figure 6.3: The control loop for pitch.

As seen in Figure 6.3, the regulator output is the reference torque wanted to correct for the error e_ϕ . The gain $(k_{lift} |l_y|)^{-1}$ converts the reference torque to the sum of squared reference angular velocities $\sum \omega_{ref,n}^2$, which is supported by Equation 6.17 in Section 6.3. Note that these reference angular velocities represents the wanted change to the current angular velocity of each motor in the control loop. The motor transfer functions require duty cycle as input, so the square root is taken of $\sum \omega_{ref,n}^2$, and the resultant $\sum \omega_{ref,n}$ is converted to a sum of duty cycles $\sum D_n$ by $G_D(s)$. Once the motor system has given the actual rotor angular velocities $\sum \omega_n$, the procedure of unit conversion is made like before, but inversely, to obtain the actual pitch torque used in G_ϕ . It is clear that

$$unit_{conversion} \cdot MotorSystem \cdot unit_{conversion}^{-1} = MotorSystem \quad (6.24)$$

and therefore all blocks except Pitch PID, $G_D(s)$, Motor System and $G_\phi(s)$ can be excluded which yields the same plant in Figure 6.4 below. It is observed in the block diagram of Figure 6.3 that the highlighted "Motor System" contains the four identical transfer functions for the four identical motors. While the wanted change to the angular velocities in $\sum \omega_{ref,n}$ are of the same magnitude, some of them are negative e.g. motor three and four when pitching, as described in Section 6.3. The dynamics of changing the angular velocity in either direction remain the same. The sum is therefore divided into four quarters to get the motor dynamics from a wanted change in angular velocity for each motor. The motor output is then summed again to obtain the actual angular velocities $\sum \omega_n$, which is used to determine the actual pitch torque τ_ϕ , as illustrated in Figure 6.3.

It is clear that the highlighted "Motor System" in Figure 6.3 consist of four quarters of equal transfer functions, and is easily simplified to G_{motor} , as it is done in Figure 6.4. Figure 6.4 illustrates the simplified control loop of which the pitch PID controller will be designed.

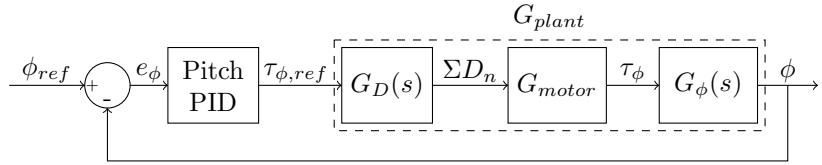


Figure 6.4: The simplified control loop for pitch.

The pitch transfer function $G_\phi(s)$ was derived in Section 6.2 to be

$$G_\phi(s) = \frac{\Delta\phi(s)}{\Delta u_1(s)} = \frac{1}{J_x s^2 + B_{ang} s} \quad (6.25)$$

and the motor transfer function $G_{motor}(s)$ was derived in Section 5.2 to be

$$G_{motor}(s) = \frac{\omega(s)}{D(s)} = \frac{K_{DC}^{-1}}{\tau_m s + 1} \quad (6.26)$$

where it is noted, that the input is a duty cycle (D). The wanted angular velocity ω_{ref} is converted to a wanted duty cycle for the motors by the gain $G_D(s)$, which is re-written below.

$$G_D(s) = \frac{D(s)}{\omega_{ref}(s)} = K_{DC} \quad (6.27)$$

The pitch PID is initially designed by analyzing the frequency response. The compensator frequency ω_1 , which will be used to derive the controller constants $K_{i\phi}$, $K_{p\phi}$, and $K_{d\phi}$, are chosen from the following three design specifications:

- A minimum phase margin (pm) of 45° . This choice of phase margin is based on an allowed phase shift of 45° to the system before becoming unstable. A good phase margin for stability typically ranges from 30° to 60° . A phase margin of 45° is therefore chosen which helps to compensate for inaccuracies in the model as well as works as a safety margin [31]. The compensator frequency ω_1 must be higher than the one at which the phase margin is achieved, i.e. $\angle G(j\omega_1)H(j\omega_1) < -180^\circ + pm$.
- The magnitude must be less than $0[\text{dB}]$. ω_1 is chosen from $|G(j\omega_1)H(j\omega_1)| < 0\text{dB}$. Similar to the phase margin, this specification works as a safety margin, ensuring that the magnitude will not cross the border at which the system becomes unstable. At this crossing, the gain margin (the maximum gain allowed to the system input while the system is still stable) is 0, and becomes negative once the magnitude in dB becomes positive. In general a positive phase margin *and* gain margin is required for the system to be stable [29].
- A settling time of maximum $5[\text{s}]$. ω_1 is chosen from $\omega_1 > \frac{8}{T_{settle} \tan(pm)}$.

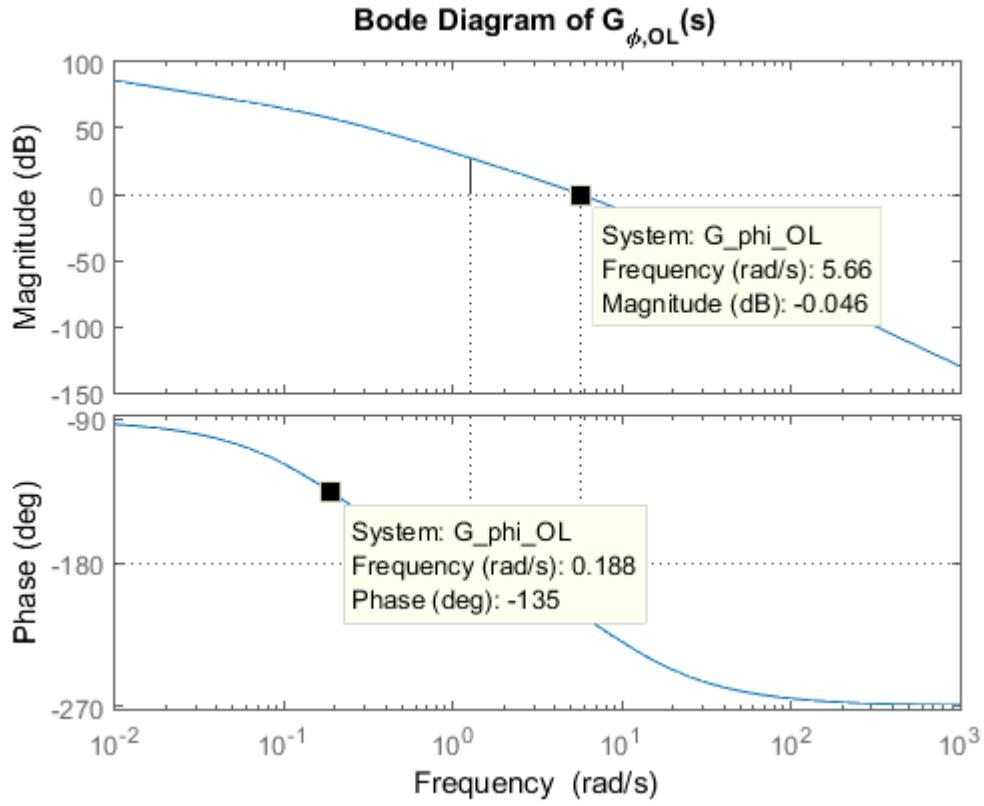


Figure 6.5: Bode diagram of the open loop pitch system.

Figure 6.5 shows a bode plot of the open loop pitch system. The points corresponding to the two specifications for choice of compensator frequency based on magnitude and phase margin are highlighted by markers. The phase margin is achieved at $\omega_1 = 0.188[\text{rad s}^{-1}]$, the magnitude of 0[dB] is achieved at $\omega_1 = 5.66[\text{rad s}^{-1}]$, and the compensator frequency related to the settling time is calculated to be $\omega_1 = 1.6[\text{rad s}^{-1}]$. The compensator frequency is chosen to be $\omega_1 = 6[\text{rad s}^{-1}]$ for further design.

From the determined ω_1 , the PID proportional constant Kp_ϕ is calculated from

$$Kp_\phi = \frac{\cos(\theta_c)}{|G_{plant,\phi}(j\omega_1)H_\phi(j\omega_1)|} \quad (6.28)$$

where H_ϕ is non-existent and therefore set to 1 [29]. The angle θ_c , which is the angle of the controller evaluated at the chosen compensator frequency, is calculated from Equation 6.29 [29].

$$\theta_c = \angle G_{PID}(j\omega_1) = -180^\circ + pm + \angle G_{plant,\phi}(j\omega_1)H_\phi(j\omega_1) \quad (6.29)$$

The calculated Kp_ϕ is utilized in Equation 6.30 to calculate the integral tuning constant Ki_ϕ [29].

$$Ki_\phi = 0.1\omega_1 Kp_\phi \quad (6.30)$$

Additionally Kd_ϕ is derived from Equation 6.31 [29].

$$Kd_\phi\omega_1 - \frac{Ki_p hi}{\omega_1} = \frac{\sin(\theta)}{|G_{plant,\phi}(j\omega_1)H_\phi(j\omega)|} \quad (6.31)$$

The results are listed in Table 9.1.

The design procedure for the roll PID (and for the yaw and altitude PID) is the same as for the pitch PID, and the control loops are similar to the one illustrated in Figure 6.4.

6.4.2 ROLL PID

Since the inertia about the local frame y -axis $J_y = 0.0257$ is nearly the same as the inertia about the local frame x -axis $J_x = 0.0258$, the transfer function and thereby the bode plot and corresponding choice of compensator angle ω_1 is very similar to that of the pitch PID. The chosen compensator angle from the three specifications is listed in Table 9.1.

The constants Kp_θ , Ki_θ , and Kd_θ are derived by same procedure as before, and the results are listed in Table 9.1.

6.4.3 YAW PID

The transfer function $G_\psi(s)$ was derived in Section 6.2 to be

$$G_\psi(s) = \frac{\Delta\psi(s)}{\Delta u_3(s)} = \frac{1}{J_z s^2 + B_{ang}s} \quad (6.32)$$

The structure of the yaw control loop is the same as the structure of the other pitch and roll control loop which is seen in Figure 6.4.

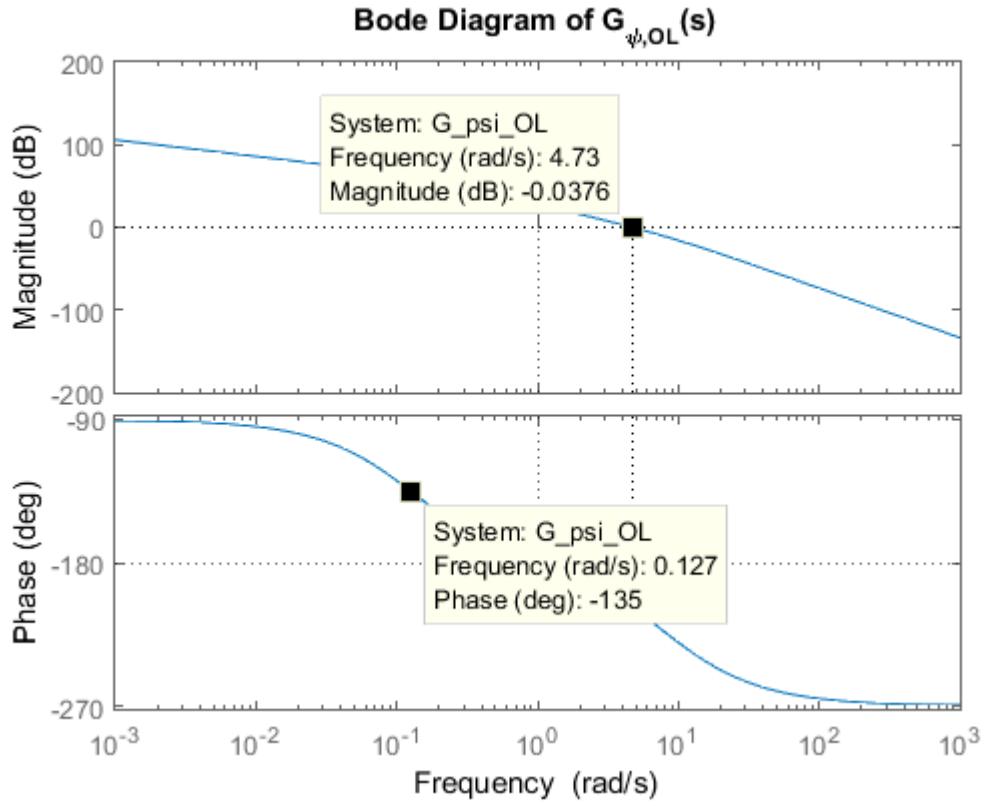


Figure 6.6: Bode diagram of the open loop yaw system.

Figure 6.6 shows a bode plot of the open loop yaw system. The specifications for choosing the compensator frequency ω_1 are the same as with the pitch and roll PID, and the points corresponding to the phase margin and magnitude specifications for choice of compensator frequency are highlighted by markers. The phase margin is achieved at $\omega_1 = 0.127[\text{rad s}^{-1}]$, the magnitude of $0[\text{dB}]$ is achieved at $\omega_1 = 4.73[\text{rad s}^{-1}]$, and the settling time is achieved at $\omega_1 = 1.6[\text{rad s}^{-1}]$. The compensator frequency is chosen to be $\omega_1 = 5[\text{rad s}^{-1}]$ for further design.

The PID constants are found again by same procedure, and the results are listed listed in Table 9.1 with the other.

6.4.4 ALTITUDE PID

The transfer function G_z was derived in Section 6.2 to be

$$G_z(s) = \frac{\psi}{\tau_\psi} = \frac{1}{ms^2 + B_{lin}s} \quad (6.33)$$

The control loop is of the same structure as the one illustrated in Figure 6.4. The bode plot of the open loop altitude control system is seen in Figure 6.7.

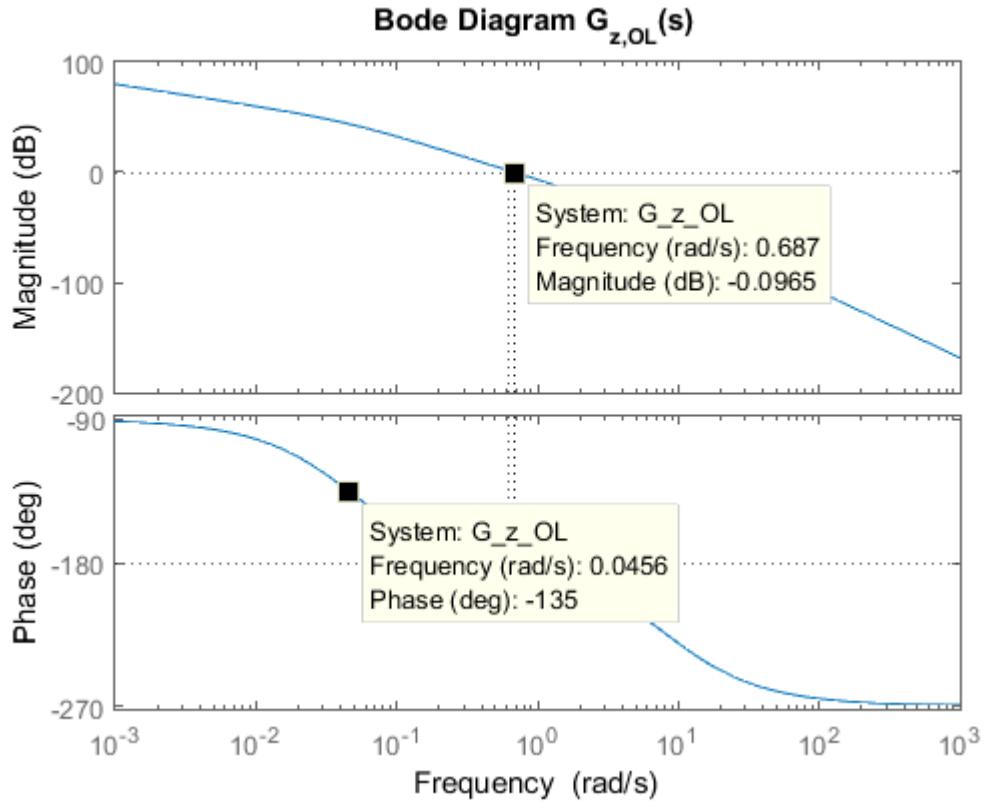


Figure 6.7: Bode diagram of the open loop altitude control system.

Two points are marked according to specifications for choice of compensator frequency. The phase margin is achieved at $\omega_1 = 0.0456[\text{rad s}^{-1}]$, the magnitude of 0 is achieved at $\omega_1 = 0.687[\text{rad s}^{-1}]$, and the settling time is achieved at $\omega_1 = 1.6[\text{rad s}^{-1}]$. The compensator frequency is chosen to be $\omega_1 = 2[\text{rad s}^{-1}]$ for further design.

The same procedure is used to determine Kp_z , Ki_z , and Kd_z and the results are listed in Table 9.1.

Controller	Transfer Function Constant	Drag constant B	$\omega_{1,chosen}$	Kp	Ki	Kd
Pitch PID	$Jx = 0.0258$	$B_{ang} = 0.0982$	$6[\text{rad s}^{-1}]$	0.2108	0.1265	0.1925
Roll PID	$Jy = 0.0257$	$B_{ang} = 0.0982$	$6[\text{rad s}^{-1}]$	0.2108	0.1265	0.1925
Yaw PID	$Jz = 0.0387$	$B_{ang} = 0.0982$	$5[\text{rad s}^{-1}]$	0.2936	0.1468	0.2251
Altitude PID	$m = 2.132[\text{kg}]$	$B_{lin} = 0.005$	$2[\text{rad s}^{-1}]$	4.7257	0.9451	3.9377

Table 6.2: PID controller constants

SECTION 6.5

SIMULATION OF CONTROLLERS

The analytically found controllers will be tested in the developed Simulink model in this section, to see the system response when given a step input on each state as well as step inputs on multiple states simultaneously.

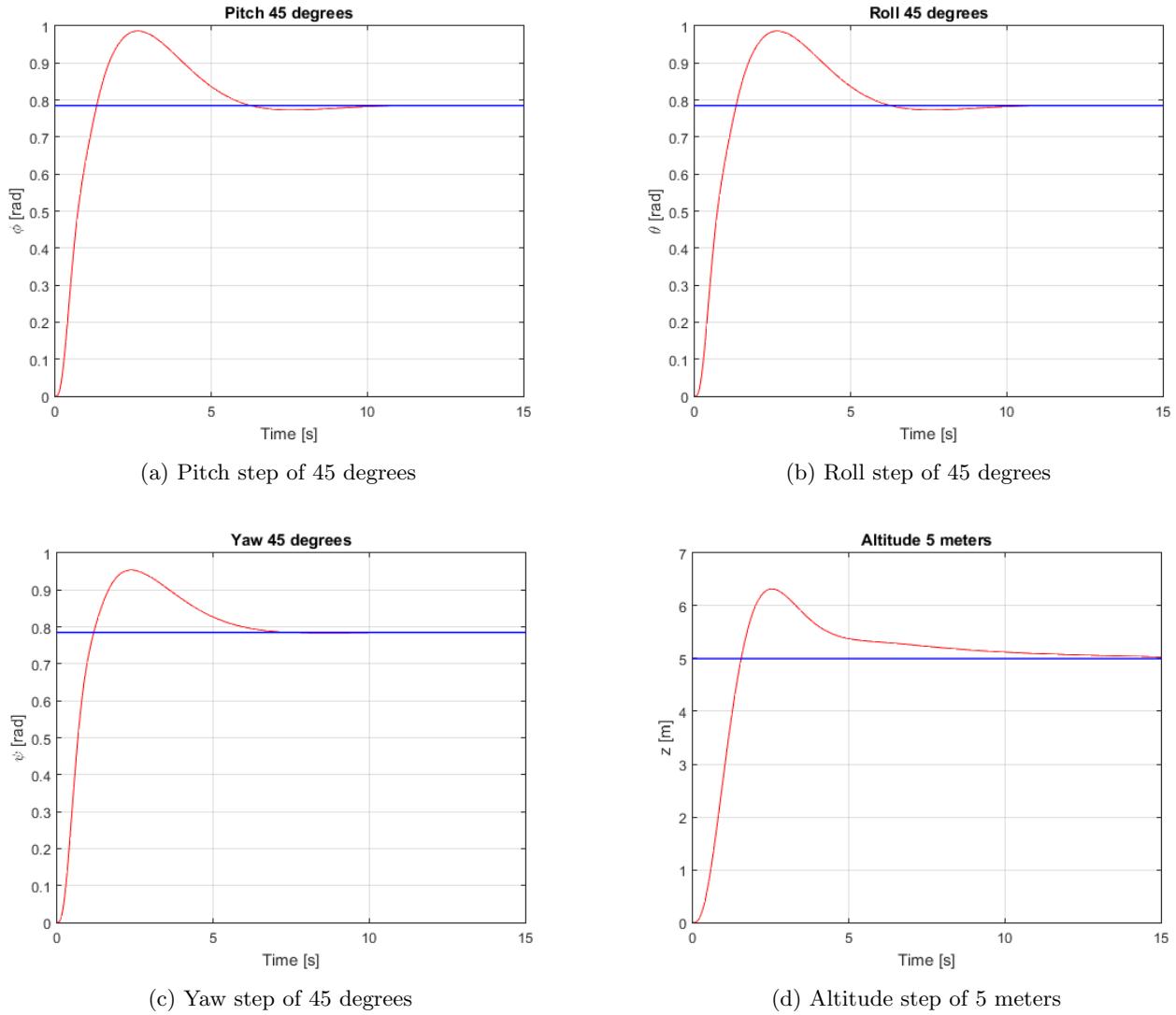

 Figure 6.8: Individual simulations for a step on ϕ , θ , ψ , and z .

Figure 6.8 shows the step response when given a step input of 45° for pitch, yaw, and roll, as well as a step input of 5m altitude. It is observed that the step response of ϕ and θ are near identical, with approximately the same settling time of $T_s = 6[\text{s}]$ and an overshoot of approximately 25[%]. This is quite high considering the effects it will have on the motion of the quadcopter. The yaw step is similar with a slightly higher settling time, but with an overshoot closer to 20[%].

In general, the settling time of all the steps are noted to be higher than the chosen settling time for designing the PID's, which was 5[s]. This is specially the case for the altitude step, which takes more than 10[s] to settle, and with an overshoot closer to 30[%]. As previously mentioned, many simplifications has been made to the system in order to derive linear transfer functions, along with the assumption of decoupling (the change in one state has no effect on the remaining states). These can all contribute by some degree to a change in the step response of the non-linear model, effectively slowing the dynamics of the step response compared to the step response of the linear transfer functions. Each step is however stable, and the controller constants will be further optimized in Chapter 7 to try and reduce the settling time and overshoot.

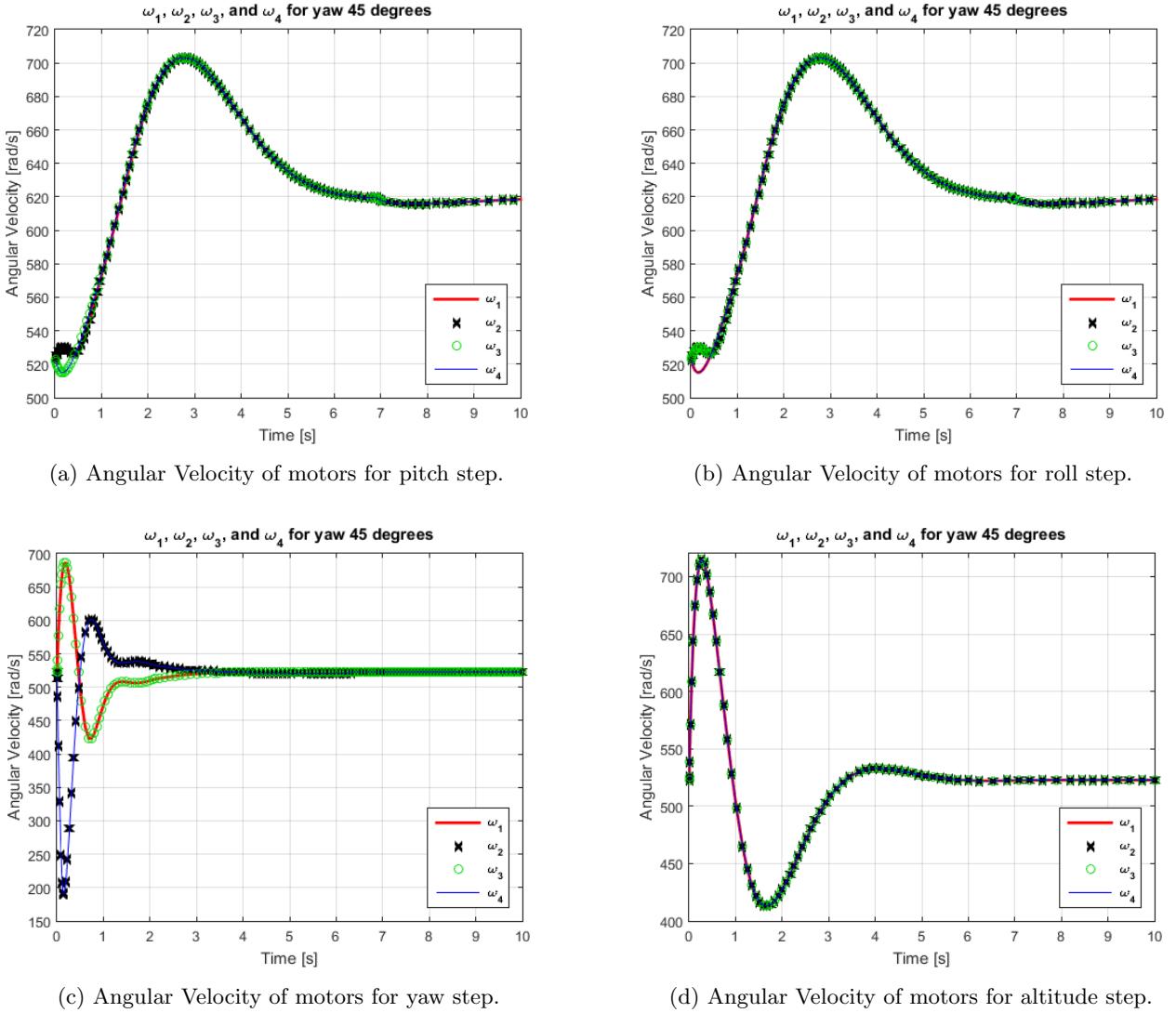


Figure 6.9: Angular velocity on all motors for each step

Figure 6.9 shows the angular velocities of the motors during the step input on each state. In Figure 6.9a it is seen that the angular velocity on motor one and two rises while the velocity of motor three and four drops in the beginning, which is expected for a positive pitch angle and around $t = 0.6[\text{s}]$ motor three and four gain higher velocity which corresponds with the overshoot on 6.8a. Afterwards the angular velocity of all motors settles at approximately $620 [\text{rad s}^{-1}]$, which is the velocity needed to keep the altitude of the quadcopter while pitched 45° .

In Figure 6.9b the angular velocities of motor one and four is observed to drop, while the angular velocity of motor two and three rises. This similarly corresponds to a positive roll angle, and as with the pitch, motor two and three falls while motor one and four rises at approximately $t = 0.6\text{s}$ to stop the quadcopter from rotating at $\theta = 45^\circ$. The motors settle at the same angular velocity as with the pitch, which is expected since the global frame lift force is proportional to $\cos(\phi) \cos(\theta)$.

In Figure 6.9c the angular velocities is seen to change much more aggressively than they did for pitching and rolling. This is expected since the torque produced from propeller drag is significantly smaller than the torque produced from the propeller lifts. Initially the angular velocity of motor one and three rises while the velocity of motor two and four falls, which corresponds to a positive yaw. When the rotation is stopped at $\psi = 45^\circ$, the velocity of

motor 2 and 4 increase, while the velocity of motor 1 and motor 3 decrease. In this case, however, the angular velocity of motor 2 and 4 falls significantly more than motor one and three rises. While the angular velocity of the motors are limited to 7000 [RPM], the yaw torque is not limited to the point where there is no loss in altitude, which means the entire range of angular velocity of the motors is used in order to produce the biggest possible yaw torque when it is required, but with the consequence of loss of altitude. Additionally, the angular velocity of the four motors from the altitude step in Figure 6.9d are seen to be equal, and the overshoot in Figure 6.9d is also seen in the dive in angular velocity.

As an additional test for the analytically determined controllers, a series of torque disturbances, of different magnitude and at different times are, added in the Simulink model using the signal builder block. For this simulation, a step input of 5[m] at the simulation start is chosen, while the steps on all Euler angles are 0[°]. The disturbances are the following:

- A pitch torque of 0.1[N m] for one second at $t = 30[\text{s}]$, and of 0.2[N m] for one second at $t = 60[\text{s}]$.
- A roll torque of $-0.1[\text{N m}]$ for one second at $t = 30[\text{s}]$, and of 0.3[N m] for one second at $t = 70[\text{s}]$.
- A yaw torque of $-0.2[\text{N m}]$ for one second at $t = 60[\text{s}]$

Figure 6.10 shows the altitude and angular position response of the quadcopter from these disturbances.

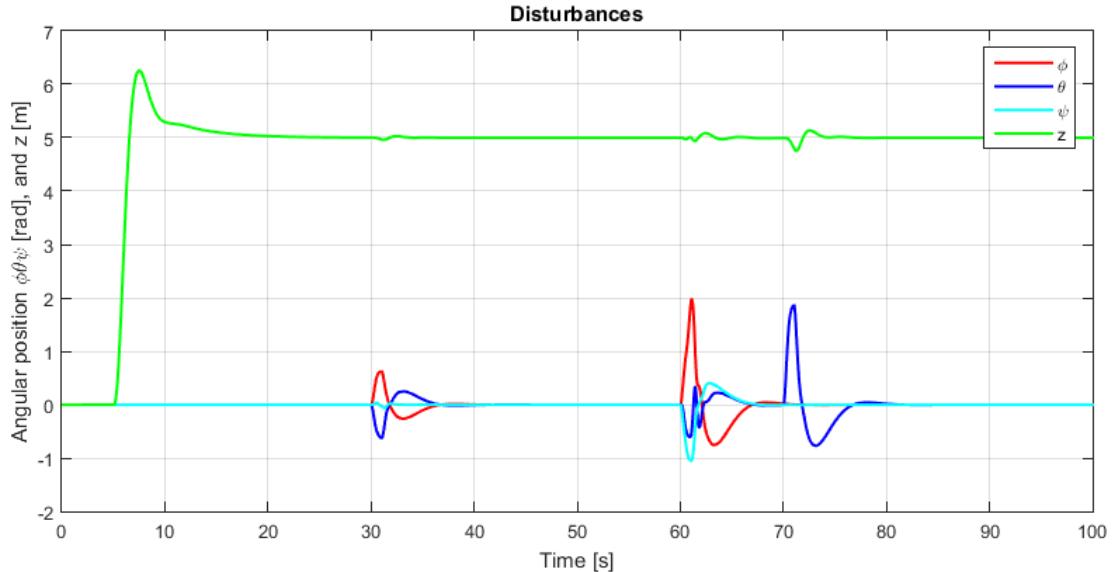


Figure 6.10: Altitude and angular position response when subjected to disturbances. Note that the angular position is multiplied by 2, for viewing purposes.

Figure 6.10 shows that the regulators corrects the angle error satisfactorily with no steady-state error, even after being subjected to disturbances of up to $0.3[\text{N m}]$ of torque for one second. The disturbances is seen to have a small effect on the altitude of the quadcopter, which is corrected by the altitude regulator satisfactorily. It is also seen that when a pitch and yaw torque is applied simultaneously (at $t = 60[\text{s}]$), a gyroscopic motion torque appears on the y_L -axis, which causes the change in the angle θ . This is also seen at $t = 30[\text{s}]$, where a pitch and roll torque of $0.1[\text{N m}]$ is applied simultaneously, and thereby causing a gyroscopic motion torque around z_L causing a small yaw of the quadcopter. Despite the exclusion of the gyroscopic motion torques in the controller design, it is seen that the resulting change in angle from these is quickly corrected by the respective regulator. The tests of the analytically developed controllers shows, that they settle in a reasonable time and with no steady-state error, though with some overshoot, which will be attempted to be improved through optimization of the controller constants.

REGULATOR OPTIMIZATION

In this chapter a numerical optimization of the controller constants (K_d , K_p and K_i), for the PID controllers will be made using Matlab, for both ϕ , ψ . This chapter is based on [32] and fifth lecture in Mechatronic system design (MCE6) with Michael M. Bech [34].

One of the requirements of this project is to construct and control a quadcopter making it able to hover steadily. Step response made on the analytically developed controllers showed a significant overshoot, when given a step input on ϕ and θ . The main goal for the optimization of the controllers, is to reduce the overshoot and settling time of the analytically developed controllers while maintaining no or a negligible small steady state error.

SECTION 7.1

OBJECTIVE FUNCTION

The object function to minimize, is chosen based on an equivalent PID controller structure as the one implemented in LabVIEW, which is of the form shown in equation 7.1.

$$G_{PID}(s) = K \left(1 + \frac{1}{T_i s} + \frac{T_d s}{\frac{T_d}{N} s + 1} \right) \quad (7.1)$$

where K is the gain, T_i is the time constant for the integrator term, T_d is the time constant for the derivative term, and N is the max high-frequency gain which should be in the range 2-20. Here chosen to be 20 based on [33] and [34]. The design vector is defined in 7.2

$$\underline{\underline{x}} = \begin{bmatrix} x1 \\ x2 \\ x3 \end{bmatrix} = \begin{bmatrix} K \\ T_i \\ T_d \end{bmatrix} \quad (7.2)$$

The chosen objective function to minimize is the gain K :

$$f(x) = x1 \quad (7.3)$$

SECTION 7.2

CONSTRAINTS

A series of constraints is made to obtain an optimized regulator without the regulator making the system unstable. These constraints can be found in Equation 7.4.

$$\left. \begin{array}{l} c(1) = 0 - x(1) \\ c(2) = 0 - x(2) \\ c(3) = 0 - x(3) \\ c(4) = 45 - \text{PhaseMargin} \\ c(5) = \text{SettlingTime} - 2 \\ c(6) = \text{Overshoot} - 8 \end{array} \right\} \quad (7.4)$$

The first three inequality constraints is just to ensure that the design vector is positive. The fourth ensures that the phase margin of the open loop transfer function with controller is minimum 45° or above. The fifth constraint is to obtain a settling time within 2 [s]. The last inequality constraint limits the overshoot, compared to the overshoot in the analytical regulator. Data of the design vector, open loop transfer function and closed loop transfer function needed for satisfying the constraints, are saved in a global state variable which gets updated and can be updated every iteration.

SECTION 7.3

SOLVING METHOD

Based on the description of the different solvers in MATLABs Optimization Toolbox, found in Appendix C Fmincon was chosen with active-set as the algorithm option. Active-set does not require a gradient and respects the bounds. The optimization was run with different initial points and bounds, where the chosen initial point, lower- and upper bounds can be seen in 7.5

$$\left. \begin{array}{l} \underline{\underline{\text{prob.xo}}} = \begin{bmatrix} 0.1 \\ 20 \\ 1 \end{bmatrix} \\ \underline{\underline{\text{prob.lb}}} = \begin{bmatrix} 0.02 \\ 0.02 \\ 0.02 \end{bmatrix} \\ \underline{\underline{\text{prob.ub}}} = \begin{bmatrix} 600 \\ 600 \\ 600 \end{bmatrix} \end{array} \right\} \quad (7.5)$$

The open loop transfer function is put in series with the controller. The design vector at the regulator gets updated every iteration and is then fed back with a unity gain to get the closed loop transfer function of the plant with optimized controller. Values of the optimized design vectors can be found in 7.6

$$\left. \begin{array}{l} \underline{\underline{\mathbf{x}_\phi}} = \begin{bmatrix} 0.0097 \\ 408.14 \\ 4.15 \end{bmatrix} \\ \underline{\underline{\mathbf{x}_\psi}} = \begin{bmatrix} 0.024 \\ 323.0543 \\ 2.80 \end{bmatrix} \end{array} \right\} \quad (7.6)$$

These values are then related to K_p , K_i and K_d values, used in the Simulink model by Equation 7.7 [31]. No optimization have been done to the controller constants for θ , since the open loop transfer function is similar to ϕ due to the symmetry, the same controller constant found for ϕ will be used for θ as well.

$$\begin{aligned} K_p &= K \\ K_i &= \frac{K}{T_i} \\ K_d &= KT_d \end{aligned} \tag{7.7}$$

	ϕ	θ	ψ
K_p	0.0097	0.0097	0.0240
K_i	2.3767e-05	2.3767e-05	7.4291e-05
K_d	0.0403	0.0403	0.0672

Table 7.1: Table of the optimized controller constants

Based on data from the global structure of the open loop transfer function with the optimized controller. The phase margins $\phi = 60.4^\circ$ and for $\psi = 57^\circ$ were found. This also satisfies the constraints set up for the phase margin.

SECTION 7.4

OPTIMIZED REGULATOR RESPONSE

This section is going to show the step response of ϕ and ψ with the new optimized controllers from Table 9.2. An optimization of the regulator was not made because no z regulator was implemented in the physical quadcopter system. The θ response is not shown because of the similarity with ϕ . Figure 7.1 shows the step response for ϕ and ψ .

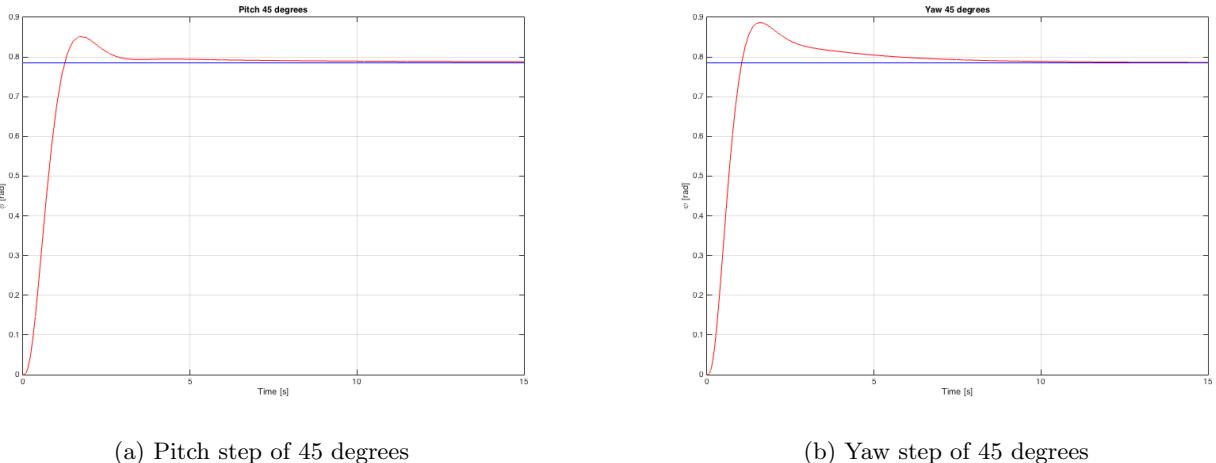
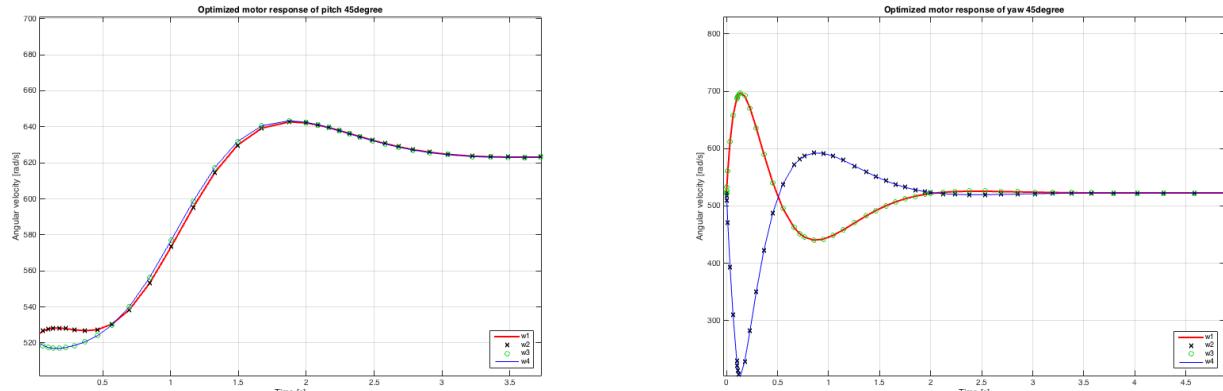


Figure 7.1: Optimized controller response

As seen on the step response for ϕ in Figure 7.1a, the overshoot is reduced significantly to around 8 % which also satisfies the constraint. The settling time is also faster compared to the analytical controller around 2.7 seconds based on stepinfo, exceeds the settling time constraint by a small but acceptable amount. Figure 7.1b shows the step response of ψ , both the overshoot (14%) and the settling time around 5.7 [s] is larger than for ϕ . This violates the constraints set up for the optimization, however, it is still assessed to be usable, since the rise time is a little faster and therefore reaches it in range of the final value fast enough. The same tendencies are also reflected on the angular velocity of the motors, which can be found in Figure 7.2.

7.4. OPTIMIZED REGULATOR RESPONSE



(a) Angular Velocity of motors for pitch step

(b) Angular Velocity of motors for yaw step

Figure 7.2: Yaw response

On Figure 7.2a it is seen that the angular velocity of motor one and two increases, while three and four correspondingly decreases in response to the step input of 45° . At around 0.75 seconds and until approximately 2 seconds, the angular velocity of motor three and four exceeds one and two, which corresponds fine with the overshoot. When the right angle is achieved the velocity on all motors follow each other and settle at around 2.7 seconds. The same tendency can be seen on Figure 7.2b here motor one and three increases while two and four decreases to give a positive yaw followed by an overshoot.

The optimized controllers found by using fmincon in Matlab are slightly faster and reduces the overshoot significantly compared to the analytically found controllers. The optimized controllers are assessed to be better than the ones determined analytically and are therefore used and implemented in LabVIEW for controlling the quadcopter.

EXPERIMENT

The regulators which have been designed are implemented on the myRIO to test the effect on the physical system. To test the regulator implementation four experiments will be conducted

1. Comparison between regulator output on the myRIO and in Simulink by manually tilting the quadcopter
2. Speed of the four motors when propellers are mounted and the individual control of each motor
3. Test of sensitivity to noise impacting the measurement of the angle when all four motors are spinning at hovering speed
4. Flying the quadcopter and observing response to minor disturbances

SECTION 8.1

PURPOSE

The purpose of testing the regulator in the physical system is to validate control theory and the behavior of the quadcopter when hovering and experiencing mild disturbances during flight. Several factors are not taken into account and when modeling the system and simplifications has been made in areas which are complicated to model or outside the scope of this project and therefore the stability of the system with all these factors will be determined experimentally.

SECTION 8.2

SETUP

8.2.1 REGULATOR OUTPUT COMPARISON

The pitch and roll regulators will be tested both separately and simultaneously. During this test both the regulator outputs and the angle will be acquired to be capable of comparing the models response to the scenario which the quadcopter had experienced.

The quadcopter will be tilted, by hand with no motors running, to approximately 10 [deg] two consecutive times. First slowly to about 10 [deg] and back to horizontal and then more rapidly to 10 [deg] and back to horizontal level.

8.2.2 INDIVIDUAL SPEED CONTROL OF THE FOUR MOTORS

The angular velocity is measured using a tachometer and the speed at different duty cycles is recorded for all four motors.

8.2.3 SENSITIVITY TO NOISE WITH SPINNING ROTORS AND PROPELLERS

Testing how much the regulator is affected by the vibrations coming from the 4 rotating rotors and propellers. This will be tested at different speed to find the speed of which the vibrations are the worst and has the highest chance of causing problems during flight.

8.2.4 HOVERING QUADCOPTER

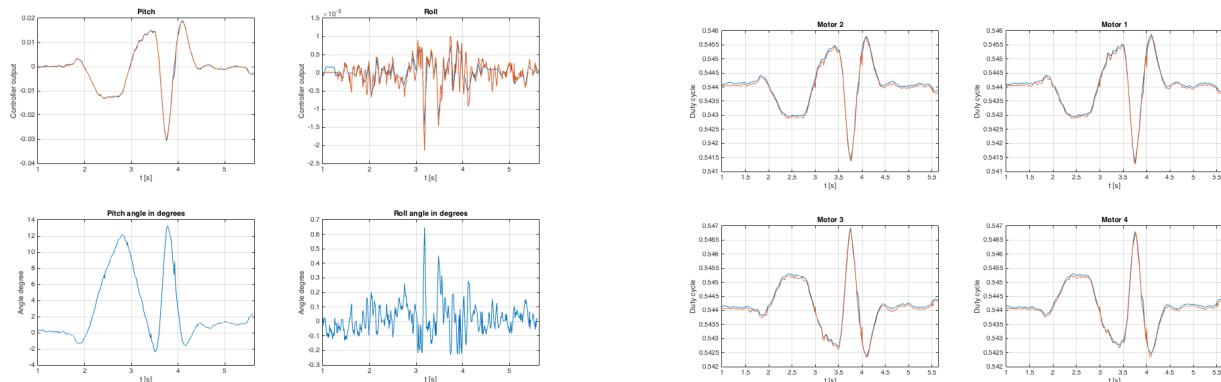
The quadcopter will be flown to a hovering state and disturbed to observe and measure the response of the regulator and how it affects the quadcopter in flight.

SECTION 8.3

DATA PROCESSING

8.3.1 REGULATOR OUTPUT COMPARISON

This section will compare the regulator outputs of the developed Simulink model and the LabVIEW program. The experiment was conducted by pitching and rolling the quadcopter manually by hand in angles of different magnitude. Figure 8.1 shows a comparison of both duty cycle and controller output between the Simulink model- and LabVIEW data, the blue line is the Simulink model data and the red line is data from the LabVIEW program.



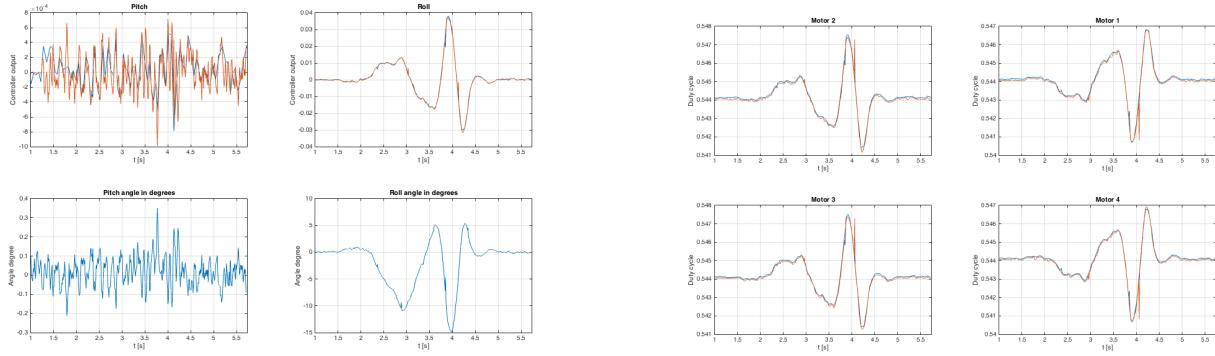
(a) Controller output comparison with small pitch angles

(b) Duty cycle comparison with small pitch angles

Figure 8.1: Data collected for comparison of regulator outputs between the Simulink model and the LabVIEW program, when applying a pitch angle to the quadcopter. Blue line = Simulink model, and Red line = LabVIEW program.

The data points for the regulator outputs for both the Simulink model and LabVIEW program lies close together and follows the same tendency for pitch, as seen on the figure 8.1. The data for roll is more oscillating and noisy for the LabVIEW data, but the y-axis scale is also correspondingly smaller, as no roll angle was manually applied to the Quadcopter. When the quadcopter has a positive pitch angle at around 2 seconds, the output is negative and the duty cycle to motor one and two decreases and correspondingly increases on three and four, as expected to

meet the reference angle of 0° . The first second of data have been filtered away on the graphs due to calibration of the quadcopter being plane and at 0° on all axes.

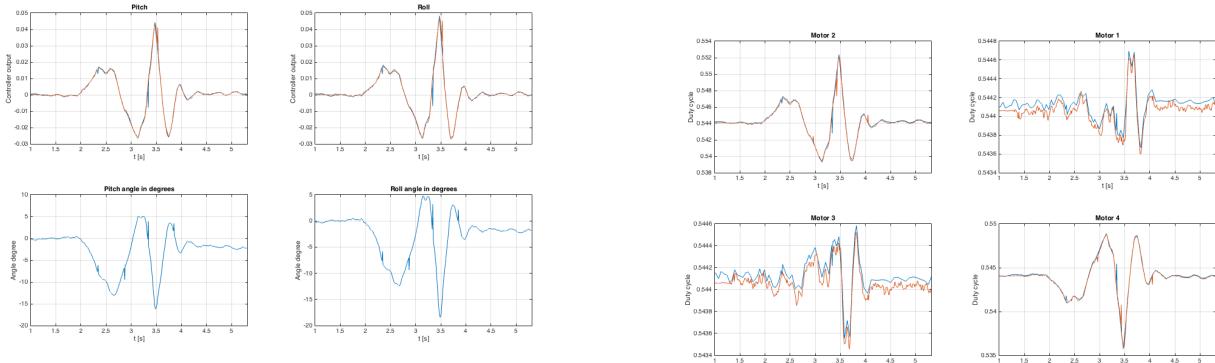


(a) Controller output comparison with small roll angles

(b) Duty cycle comparison with small roll angles

Figure 8.2: Data collected for comparison of regulator outputs between the Simulink model and the LabVIEW program, when applying a roll angle to the quadcopter. Blue line = Simulink model, and Red line = LabVIEW program.

Figure 8.2 shows the regulator comparison, when a roll angle is applied to the quadcopter. Figure 8.1a shows the same tendencies as for pitch. Which is also expected since the inertia around the pitch and roll axes are close to identical and therefore have similar transfer functions and controller designs.



(a) Controller output comparison with small roll and pitch angles

(b) Duty cycle comparison with small roll and pitch angles

Figure 8.3: Comparison between pitch and roll angles

Figure 8.3 shows data for the experiment of the quadcopter being both pitched and rolled at the same time, this is done to test if the controllers works as expected and corrects the error even when both being pitched and rolled. The quadcopter has a negative pitch- and roll angle simultaneously at 2.25 seconds, which means that the quadcopter should increase the angular velocity on motor two and decrease on motor four, while motor one and three should remain almost unchanged. This is the case as seen on 8.3b.

The comparison between the Simulink model and LabVIEW program data shows that tendencies of the regulators are correct, the duty cycle to the right motors gets either increased or decreased accordingly to what would be expected when the quadcopter is tilted in a specific angle. Only the tendency of the controller can be verified through this comparison, since the Simulink model only gets angle inputs from the experiments conducted on the quadcopter.

8.3.2 INDIVIDUAL SPEED CONTROL OF THE FOUR MOTORS

When conducting this experiment a significantly higher angular velocity was observed on motor 3 at low duty cycles both with and without propellers. The cause of this problem was inspected and was found to be a faulty ESC. The angular velocity on motor 3 and motor 4 can be seen in Figure 8.4 without the propeller on, however, the effect with a propeller on was similar

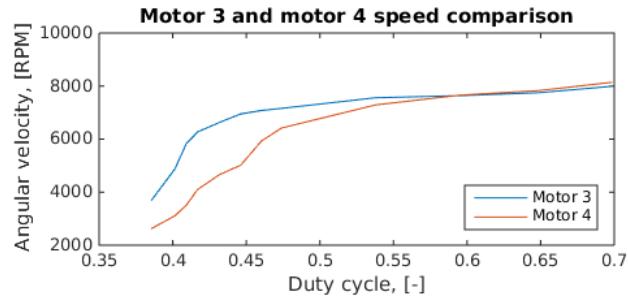


Figure 8.4: Comparison between motor 3 and motor 4 without propeller. Motor 4 behaves as motor 1 and motor 2 while motor 3 behaves irregularly

This issue could be resolved by making a new fit of the angular velocity for only motor 3 and using this for only controlling motor 3. This was not done due to the time it would take to do it and the concern that any problem within the ESC would get worse and potentially put the structure of the quadcopter in danger.

8.3.3 SENSITIVITY TO NOISE WITH SPINNING ROTORS AND PROPELLERS

This experiment was not conducted due to the problems found with one of the ESCs which was found and discussed in Section 8.3.2.

8.3.4 HOVERING QUADCOPTER

This experiment was not conducted due to the problems found with one of the ESCs which was found and discussed in Section 8.3.2.

CONCLUSION

In this Chapter a conclusion to the problem specifications for the project will be made.

At the beginning of the project the following problem statement was made:

How can a quadcopter be designed, modeled, and optimized for energy efficient use while remaining stable?

Based on the problem statement, a structural optimization was made using Differential evolution to find the most energy efficient quadcopter design. The objective function to minimize was therefore chosen to be the power needed to spin the propeller at constant angular velocity given by the drag on the propeller an angular velocity. The object function was given by:

$$P = \tau_m \omega \quad (9.1)$$

By minimization of the objective function with the made constraints in C, design variables were found. Due to limitations of available components, which could be found and put together within the time frame of following design variables were found to give the most energy efficient design with regards to the power loss of the propeller due to drag which by nature will try to minimize the drag on the propeller as well as the angular velocity. The arm dimensions ensures that the natural frequency and stress in the arm does not increase to level which could harm the quadcopter during optimization.

$$D = 15 \quad [\text{in}] \quad (9.2)$$

$$p = 6.557 \quad [\text{in}] \quad (9.3)$$

$$RPM = 1.458 \quad [\text{RPM}] \quad (9.4)$$

$$L = 0.283 \quad [\text{m}] \quad (9.5)$$

$$w = 14.2 \quad [\text{mm}] \quad (9.6)$$

$$h = 7.1 \quad [\text{mm}] \quad (9.7)$$

A Simulink model of the quadcopter was made together with an analytically developed PID regulator to help keep the quadcopter stable as stated in the problem statement. The analytic regulators was designed based on criteria in 6.4 and the values can be seen in the following table

Controller	Transfer Function Constant	Drag constant B	$\omega_{1,chosen}$	Kp	Ki	Kd
Pitch PID	$I_x = 0.0258$	$B_{ang} = 0.0982$	$6[\text{rad s}^{-1}]$	0.2108	0.1265	0.1925
Roll PID	$I_y = 0.0257$	$B_{ang} = 0.0982$	$6[\text{rad s}^{-1}]$	0.2108	0.1265	0.1925
Yaw PID	$I_z = 0.0387$	$B_{ang} = 0.0982$	$5[\text{rad s}^{-1}]$	0.2936	0.1468	0.2251
Altitude PID	$m = 2.132[\text{kg}]$	$B_{lin} = 0.005$	$2[\text{rad s}^{-1}]$	4.7257	0.9451	3.9377

Table 9.1: PID controller constants

The overshoot of the analytical regulators for ϕ and θ were around 25% and 20% for ψ which was higher than desired. The settling time were around 6s for both ϕ , θ and ψ , which were slightly than expected. The PID regulator for z is left out because that it is not used in LabVIEW, there the altitude is controlled manually by a throttle. The analytically developed regulators had too high overshoot and settling time to get the wanted stability, an optimization was therefore made on the regulators to get a better performance.

The objective function used in the optimization of the analytical regulators was based on a PID regulator similar to the one implemented in LabVIEW and was as shown below:

$$G_{PID}(s) = K \left(1 + \frac{1}{T_i s} + \frac{T_d s}{\frac{T_d}{N} s + 1} \right) \quad (9.8)$$

Where the objective function is to minimize the gain K, and with the design vector:

$$\underline{\underline{x}} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} K \\ T_i \\ T_d \end{bmatrix} \quad (9.9)$$

The optimized regulators found with the given constraints mentioned in 7 is given in the table below:

	ϕ	θ	ψ
K_p	0.0097	0.0097	0.0240
K_i	2.3767e-05	2.3767e-05	7.4291e-05
K_d	0.0403	0.0403	0.0672

Table 9.2: Table of the optimized controller constants

The optimized regulators on ϕ and θ satisfied the constraints for overshoot on 8%, the settling time exceeds the constraints by a small but negligible amount. The phase margin for ϕ and θ was 60.4° and 57° for ψ , which satisfied the constraints. The step response of the optimized controllers were improved compared to the analytical and therefore used in the LabVIEW program for controlling the quadcopter.

It was not possible to make experiments with the quadcopter flying because of a defect ESC. The only test which was possible to conducted was by examining the performance of the regulator without the quadcopter flying. This was done by pitching and rolling the quadcopter manually by hand and use the sensor data of the angles in the Simulink model to see the regulator- and duty cycle outputs. The experiment showed similar regulator performance and the right duty cycle response to different error angles.

FUTURE WORK

This chapter describes different possibilities of future work to be done on the quadcopter.

Experiments to test the quadcopter while flying is needed to check the full performance of the controller and to determined how stable it is while hovering. This also includes examination of the effect of high frequency noise on the regulators and the need to reduce it. The experiments would have to be simulated in the model to be able to compare and assess the accuracy of the full model.

Furthermore the next step to improve the quadcopter would be to implement additional sensors to advance the control of the quadcopter. A magnetometer can be implemented in the system to approximated a more accurate yaw angle by measuring a magnetic field vector and compared it the Earth's magnet field. If an accurate yaw angle of the quadcopter can be obtained it would be possible to use a regulator on the angle, and thereby make the quadcopter more stable during flying.

In addition a barometer can be implemented in quadcopter, which measures both temperature and air pressure. It can be used to estimate an altitude of the quadcopter, and depending on the type used, an resolution of 10 [cm] in the altitude can be obtained [35]. With an accurate approximation of the altitude, and regulator for the altitude position can be implemented to the controller. With an altitude regulator it will possible to get the quadcopter up

to and keep a desired altitude.

Furthermore the implementation of a R/C controller would make the quadcopter more easily to control compared to current setup, where a host computer is used. A R/C receiver is needed to be implemented to the controller which would be able to receive the transmitted signals from the R/C controller. The controller has to be programmed to read and operate accordingly to the transmitted signal.

BIBLIOGRAPHY

- [1] PmodGYRO 3-Axis Digital Gyroscope, Digilent Std. [Online]. Available: <http://www.digilentinc.com/Products/Detail.cfm?Prod=PMOD-GYRO>
- [2] Turnigy Multistar 30 Amp Multi-rotor Brushless ESC 2-4S (OPTO), Hobbyking Std. [Online]. Available: http://www.hobbyking.com/hobbyking/store/_27776_Turnigy_Multistar_30_Amp_Multi-rotor_Brushless_ESC_2_4S_OPTO_.html
- [3] M. E. Dreier, Introduction to Helicopter and Tiltrotor Flight Simulation, J. A. Schetz, Ed. American Institute of Aeronautics and Astronautics, 2007.
- [4] USER GUIDE AND SPECIFICATIONS NI myRIO-1900, National Instruments, 2013. [Online]. Available: <http://www.ni.com/pdf/manuals/376047a.pdf>
- [5] L3G4200D MEMS motion sensor: ultra-stable three-axis digital output gyroscope, STMicroelectronics, 2010. [Online]. Available: <http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/CD00265057.pdf>
- [6] Amazon, “Amazon prime air.” [Online]. Available: <http://www.amazon.com/b?node=8037720011>
- [7] Forbes, “What makes the quadcopter design so great for small drones?” 2013. [Online]. Available: <http://www.forbes.com/sites/quora/2013/12/23/what-makes-the-quadcopter-design-so-great-for-small-drones/>
- [8] Regler vedrørende flyvninger med droner i Danmark, droner.dk Std. [Online]. Available: <http://droner.dk/regler-for-droneflyvning/>
- [9] A. Gibiansky. (2013, November) Quadcopter dynamics and simulation. [Online]. Available: <http://andrew.gibiansky.com/blog/physics/quadcopter-dynamics/>
- [10] A. S. of Mechanical Engineers, “Lighter than air,” 2014. [Online]. Available: <https://www.asme.org/getmedia/c9569a01-86cc-4dc7-888b-02311ed2bed4/Lighter-than-Air-UAV-FY14.aspx>
- [11] SteadyDrone QU4D Ready To Fly Quadcopter, Kinguiruano Std. [Online]. Available: <http://www.kinguiruano.com/steadydrone-qu4d-ready-to-fly-quadcopter-p-1724.html>
- [12] GPS - Hexacopter flat S800, MULTICOPTER Shop Std. [Online]. Available: <http://www.multicopter-shop.de/copter/hexacopter/hexacopter-flat-head-s800-dji-naza-m-v2-gps.html>
- [13] Octocopter, service-drone Std. [Online]. Available: <http://www.service-drone.com/en/octocopter>
- [14] O. Liang, Build A Quadcopter From Scratch - Hardware Overview, Internet page, OscarLiang.net Std., June 2013, last seen 26-02-15. [Online]. Available: <http://blog.oscarliang.net/build-a-quadcopter-beginners-tutorial-1/>
- [15] Build Your Own Quadcopter Beginner Guid - Part 1, Robot Kingdom Std., August 2013. [Online]. Available: <http://robot-kingdom.com/build-your-own-quadcopter-beginner-guide-part-1/>
- [16] Drone Power, Edison Tech Center Std. [Online]. Available: <http://www.edisontechcenter.org/Drones.html>
- [17] The Electric Motor, Edison Tech Center Std. [Online]. Available: <http://www.edisontechcenter.org/electricmotors.html>
- [18] Sam, How to choose the right motor for your multicopter, DroneTrest Std., October 2014. [Online]. Available: <http://www.dronetrest.com/t/how-to-choose-the-right-motor-for-your-multicopter/568>
- [19] Guid to PWM and PPM, Endurance R/C Std. [Online]. Available: <http://www.endurance-rc.com/ppmtut.php>
- [20] T. Jespersen, An example of an ESC used for quadcopters, TKJ Electronics Std., March 2012, last seen 02-03-2015. [Online]. Available: <http://blog.tkjelectronics.dk/2012/03/quadcopters-how-to-get-started/>

- [21] D. Calin, Accelerometer, Gyroscope and IMU Sensors - Tutorials, Introrobotics Std., July 2013. [Online]. Available: <http://www.introrobotics.com/accelerometer-gyroscope-and-imu-sensors-tutorials/>
 - [22] A. Fabio, DRONING ON: CHOOSING A FLIGHT CONTROLLER, HACKDAY Std., June 2014. [Online]. Available: <http://hackaday.com/2014/06/06/droning-on-flight-controller-round-up/>
 - [23] D. Wang, QuadcopterParts: What are they and what do they do?, Quadcopter Academy Std., February 2015. [Online]. Available: <http://www.quadcopteracademy.com/quadcopter-parts-what-are-they-and-what-do-they-do/>
 - [24] C. J. Fisher, Using an Accelerometer for Inclination Sensing, Analog Devices Std., 2010, last seen: 18-05-2015. [Online]. Available: <http://www.analog.com/media/en/technical-documentation/application-notes/AN-1057.pdf>
 - [25] S. Colton, “The balance filter - a simple solution for integrating accelerometer and gyroscope measurements for a balancing platform,” June 2007, last seen: 04-06-15. [Online]. Available: <https://b94be14129454da9cf7f056f5f8b89a9b17da0begoogledrive.com/host/0B0ZbiLZrqVa6Y2d3UjFVWDhNZms/filter.pdf>
 - [26] P. J. P. How, “Inertial sensors - complementary filtering - simple kalman filtering,” Fall 2004, last seen: 04-06-15. [Online]. Available: http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-333-aircraft-stability-and-control-fall-2004/lecture-notes/lecture_15.pdf
 - [27] P. E. Nikravesh, Computer-Aided Analysis of Mechanical Systems. Simon & Schuster, 1988.
 - [28] K. Meriam, Engineering Mechanics Dynamics, 6th ed. Wiley & Sons, 2008.
 - [29] C. Phillips, Feedback control systems / Charles L. Phillips, John M. Parr, fifth edition ed. Boston: Prentice Hall, 2011.
 - [30] N. Instruments, “Pid theory explained,” March 2011. [Online]. Available: <http://www.ni.com/white-paper/3782/en/>
 - [31] H. C. Pedersen, “Grundlaeggende regulering og modellering lektion 14,” March 2014. [Online]. Available: https://www.moodle.aau.dk/pluginfile.php/217034/mod_resource/content/0/Files-lecture-14/MM14_slides.pdf
 - [32] J. Arora, Introduction to Optimum Design, fifth edition ed. Boston, MA: Academic Press, 2011.
 - [33] C. I. of Technology, “Chapter 8 pid control,” March 2014. [Online]. Available: http://www.cds.caltech.edu/~murray/courses/cds101/fa04/caltech/am04_ch8-3nov04.pdf
 - [34] M. M. B. AAU, “Mechatronic system design lecture 5,” March 2014. [Online]. Available: https://www.moodle.aau.dk/pluginfile.php/469883/mod_resource/content/1/L5%20Slides%20and%20Exercise.pdf
 - [35] MS5611-01BA03 Barometric Pressure Sensor, with stainless steel cap, measurement SPECIALITIES, October 2012. [Online]. Available: <http://www.meas-spec.com/downloads/MS5611-01BA03.pdf>
 - [36] A. Ragheb, Wind Turbine Gearbox Technologies. INTECH Open Access Publisher, 2011. [Online]. Available: <http://cdn.intechopen.com/pdfs-wm/16241.pdf>
 - [37] T. Irvine, “Bending frequencies of beams, rods, and pipes,” November 2012. [Online]. Available: <http://www.vibrationdata.com/tutorials/beam.pdf>
 - [38] K. Price, Differential evolution a practical approach to global optimization. Berlin New York: Springer, 2005.
- Appendix

APPENDIX A

LIGHTER THAN AIR

Lighter than Air UAV

The use of Unmanned Air Vehicles (UAV) has proved to be very efficient. They not only save the pilot from being placed in harm's way but the expense is much lower. There are uses of UAV's that have been identified. An example of one of those uses would be to observe a forest fire to provide information so the Forrest Service can determine the best way to fight the fire.

Design a small UAV to carry a cargo through two gates, see below, drop a payload and return to the starting point. This is an initial proof of concept prototype. You must design and build, at a minimum, the propulsion and control system for the UAV yourself. You cannot purchase and modify an existing commercially available vehicle. The device must be able to maneuver around and through obstacles, change height, and pass through a hoop for sizing.

A hoop approximately 0.71-m (28-in) in diameter will be used by the judges to limit the size of the vehicle. It is radio controlled.

Vehicle Requirements

1. The vehicle must be powered by batteries.
2. The device must be controlled through a wireless transmitter/receiver radio link. The following requirements pertain specifically to the device controller:
 - A radio transmitter may have its own batteries rechargeable or non-rechargeable.
 - The transmitter/receiver radio link may be any commercially available model controller.
 - During the trial, the device must be completely controlled via the radio link no other contact, interaction or influence is permitted.
 - One team member must control the device throughout the trial.
 - All radio controllers will be impounded and shut off during the competition, except during the team's run.
3. All devices must have a readily accessible and clearly labeled master shut-off switch.

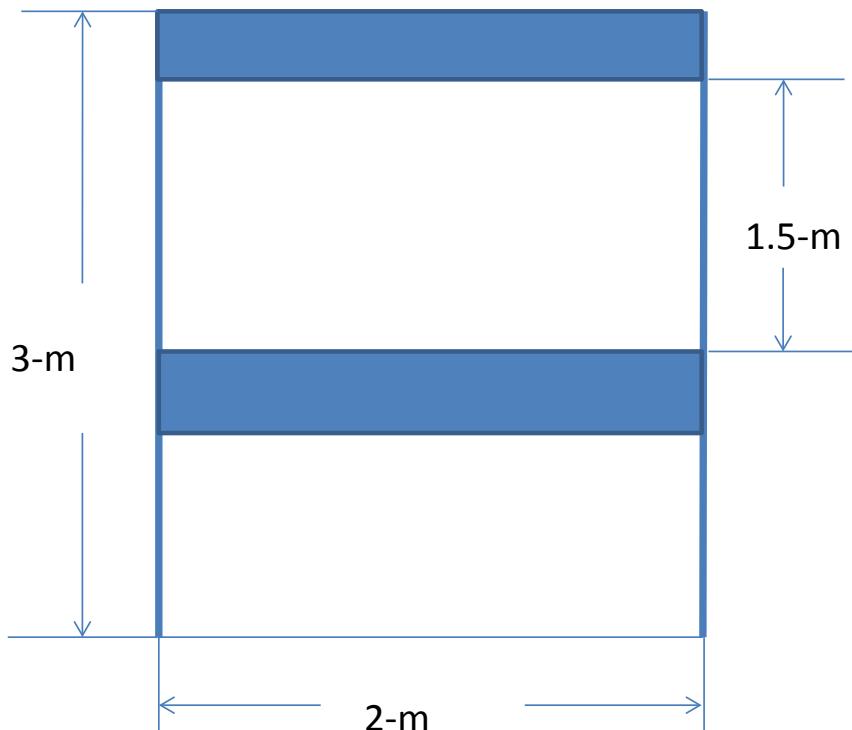
Course Description

The **test course** will consist of a flat, level section of floor marked off with masking tape and the corresponding airspace above that floor. The rectangular course, with the maneuver gates, will be in a space of 5-m by 7.75-m in size. The air space for maneuver will be the room of the venue. There will be no penalty for maneuvering outside the gates course. This will not be consistent district to district. The minimum will be 2-m between gates. The gates can be placed anywhere in the marked off area.

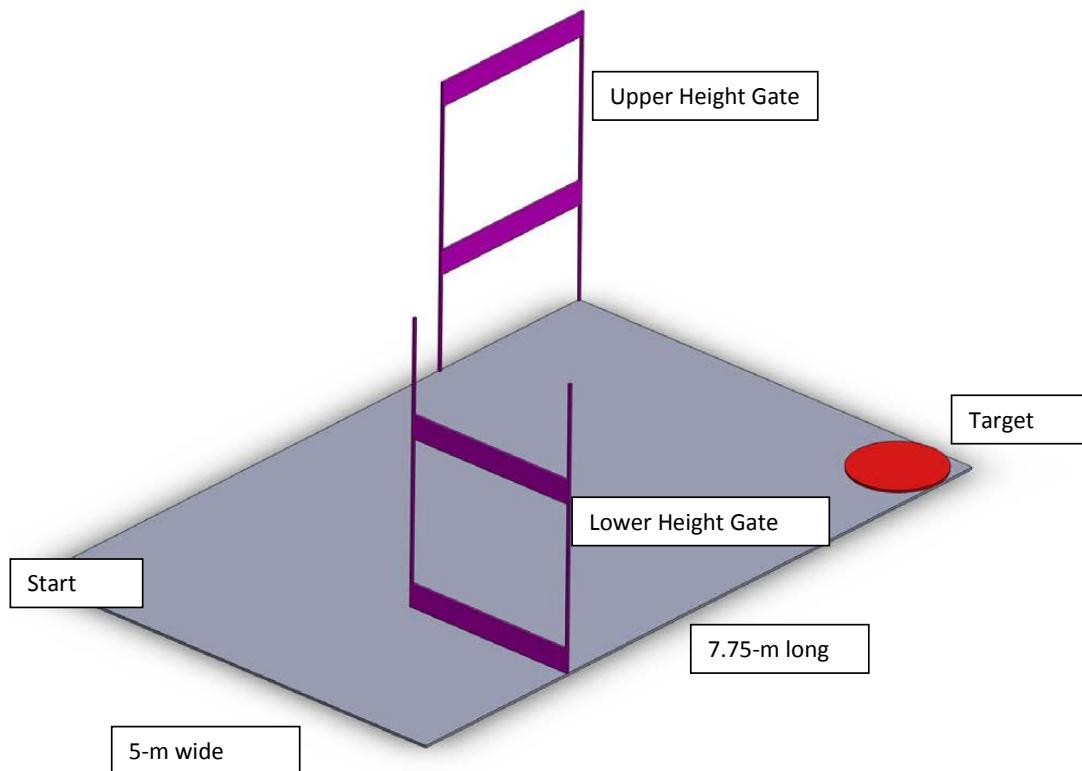
The UAV must go through two gates drop a payload and return to the starting point. Extra points will be awarded for going through the same two gates after dropping the payload. Each

gate will be 2M wide and 3M in height with a 1.5-M high opening. One gate will have the opening high and the other gate will have the opening low. See schematic of the gate.

Example of Gate and Opening



Example Layout of Course



Tasks to be accomplished.

1. Navigate through the gates in the fastest time.
2. Teams will be scored on the maximum cargo carried.
3. Bonus: Release a simulated 1-gm water bladder. (Note: Use a bag of sand.)
4. Bonus: Does the canister hit the intended fire? Target is 1-m in diameter.
5. Hitting or touching the gates will incur a penalty.
6. Provide photographic visual evidence of the construction of your vehicle.
7. Signed Ethical Statement that you constructed the vehicle.
8. One page Design Calculations.

Run Score = Max(300 s – Trial Time, 0)
+ (Number of gates successfully negotiated)x200
+ (Number of grams carried) x50
+(release of bladder)x20
+(bladder hits target)x100
+(Lighter than air)x100
-(number of gates hit)x20
-(unacceptable design calculations)x100

APPENDIX B

LIFT AND DRAG ESTIMATIONS

This Appendix contains the derivation of the drag and lift effects of the propellers, which are used in Chapter 5 for determining the equations of motion of the quadcopter. A local nomenclature is used for this Appendix, which is illustrated below.

Nomenclature	Unit	Description
A	[m ²]	Cross sectional area of arm profile
D	[m]	Diameter of propeller
f_n	[Hz]	Natural frequency of arm with motor and propeller attached at the end
F_D	[N]	Drag force of propeller when hovering
F_L	[N]	Lifting force from one propeller when hovering
$F_{L,max}$	[N]	Lifting force when lifting twice the weight as when hovering
J_a	[kg m ²]	Mass moment of inertia of the arm around the center of the quadcopter
J_m	[kg m ²]	Mass moment of inertia of the motor around the motor shaft axis
J_p	[kg m ²]	Mass moment of inertia of the propeller around the motor shaft axis
J_s	[m ⁴]	Area moment of inertia of the arm
L	[m]	Length of arm from center of quadcopter to center of motor axis
p	[m]	Pitch of propeller
P_m	[W]	Power needed to turn the propeller at ω_m
M	[kg]	Total mass of the quadcopter
M_a	[kg]	Mass of arm connecting motor to center
M_a	[kg]	Mass of central hub including battery, MCU, sensors, and ESCs
M_m	[kg]	Mass of motor
M_p	[kg]	Mass of propeller
N	[RPM]	Angular velocity of rotor when hovering
T	[N]	Thrust of motor and propeller
σ_{mat}	[Pa]	Yield strength of material used for the arm
σ_{max}	[Pa]	Maximum stress in the arm when lifting force is $F_{L,max}$
τ_S	[N m]	Torque applied on the total system pitch, roll, or yaw it
τ_m	[N m]	Torque around the motor axis coming from drag on the propeller
ω_m	[rad s ⁻¹]	Angular velocity of rotor when hovering
$\omega_{m,max}$	[rad s ⁻¹]	Angular velocity of rotor when lifting twice the weight of the copter
ω_n	[rad s ⁻¹]	Natural frequency of arm, motor, and propeller

To estimate the drag and lift of the propeller, some assumptions calculating these are made:

- Drag and lift of propeller is approximated using blade element theory
- Propeller does not resonate at any angular velocity
- Pitch is constant at any distance r for when $r > 0$, however, it has a maximum angle of attack of 45 [°] to the flow of air
- Motors has no internal friction
- Motor torque constant is independent of the angular velocity of the motor
- Density of air is constant

Blade Element Theory is thereby used for determining the lift and drag from the propellers. The blade is divided into small segments of length dr and blade width h , as illustrated in Figure B.1. The length from the edge of the propeller hub to each segment is r_i .

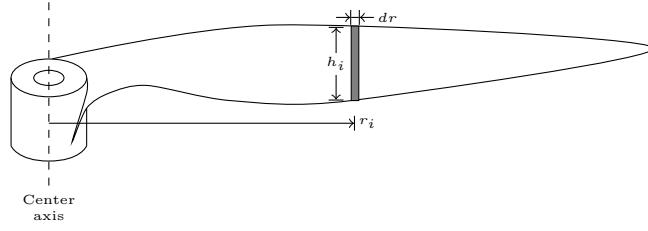


Figure B.1: The propeller blade with an illustrated segment.

The lift and drag forces acting on each element can be found using

$$\text{Lift: } \underline{\mathbf{F}_{l,i}} = \frac{1}{2} C_{l,i} A_i \rho \underline{\mathbf{v}_i}^2 \quad [\text{N}] \quad (\text{B.1})$$

$$\text{Drag: } \underline{\mathbf{F}_{d,i}} = \frac{1}{2} C_{d,i} A_i \rho \underline{\mathbf{v}_i}^2 \quad [\text{N}] \quad (\text{B.2})$$

where $F_{l,i}$ and $F_{d,i}$ are the respective lift and drag force on element i , $C_{l,i}$ and $C_{d,i}$ are the lift and drag coefficient of each element, ρ is the density of air and $\underline{\mathbf{v}_i}$ is the velocity of element i [36]. Based on the geometry of the propellers, the lift and drag are assumed to be produced only from the propeller spinning in the xy -plan. The velocity of each element is therefore expressed from the spin rate of the propeller in the local frame xy -plan only, and is

$$\underline{\mathbf{v}_i} = \underline{\omega_n} r_i \quad [\text{m s}^{-1}] \quad (\text{B.3})$$

where $\underline{\omega_n}$ is the spin of the propeller in the local frame xy -plan, which was defined in Equation 5.12 in Section 5.1.1.

The area A_i of each element is defined from Equation B.4.

$$A_i = h_i dr \quad (\text{B.4})$$

DRAG AND LIFT COEFFICIENTS

The drag and lift coefficients depend on many factors, and are for the most part experimentally decided. Until a characterization of the propellers are made, data from experimental tests on a NACA 0012 airfoil is used, where the drag and lift coefficients are expressed from the blade's angle of attack [3], which is the angle of the blade relative to the local frame xy -plan. This is a rough estimate of coefficients based on a geometry differing from the geometry of the propellers, and is only to be used for modelling purposes. To calculate the angle of attack, the circumference of each element (the distance traveled by the element in one revolution) is needed, which is

$$C_i = 2\pi r_i \quad (\text{B.5})$$

where C_i is the circumference related to the element at r_i . The angle of attack of each element is then given by

$$\alpha_i = \tan^{-1} \left(\frac{P}{C_i} \right) \quad (\text{B.6})$$

where P is the blade pitch and α_i is the angle of attack of element i .

Figure B.2 shows the drag coefficient experimental data on the NACA 0012 airfoil, and Figure B.3 shows the lift coefficient data.

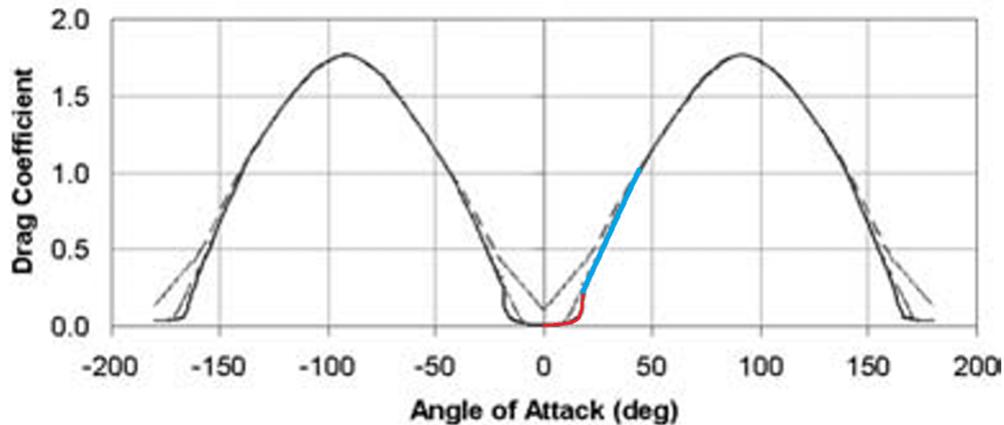


Figure B.2: Drag Coefficient as a function of angle of attack for the NACA 0012 airfoil. [3]

It is seen that the function is split into two sections. The first section from 0° to 15° angle of attack (colored red) is expressed by a quadratic function, and the second section in the interval 15° to 45° (colored blue) is expressed by a linear function.

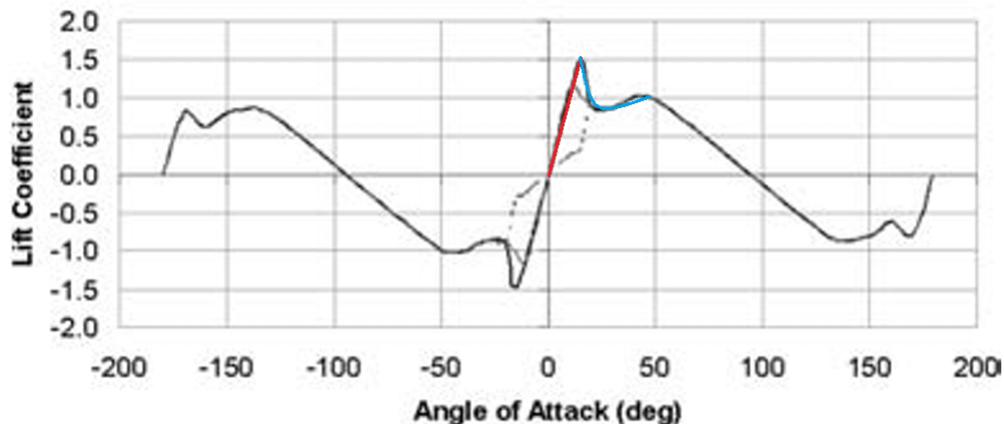


Figure B.3: Lift Coefficient as a function of angle of attack for the NACA 0012 airfoil. [3]

The function of the lift coefficient is similarly split into two sections. The first section in the interval 0° to 15° angle of attack (colored red) is expressed with a linear function, and the second section (colored blue) in the interval 15° to 45° is expressed with a quadratic function.

The interpolation yields the functions of the drag and lift coefficient of each element to be

$$\begin{aligned}
C_{l,i} &= 5.72958 \alpha_i, & 0 \leq \alpha_i \leq 15 \frac{\pi}{180} \\
C_{l,i} &= 9.3013 \alpha_i^2 - 10.5042 \alpha_i + 3.6125, & 15 \frac{\pi}{180} \leq \alpha_i \leq 45 \frac{\pi}{180} \\
C_{d,i} &= 3.2828 \alpha_i^2 + 0.0573 \alpha_i + 0.005, & 0 \leq \alpha_i \leq 15 \frac{\pi}{180} \\
C_{d,i} &= 1.6234 \alpha_i - 0.175, & 15 \frac{\pi}{180} \leq \alpha_i \leq 45 \frac{\pi}{180}
\end{aligned} \tag{B.7}$$

To find the total lift ($F_{l,n}$) of propeller n , the forces on each element are summed for all the elements on one blade and multiplied by the number of blades on the propeller, shown in Equation B.9

$$\underline{\mathbf{F}_{l,n}} = N_{bl} \sum_i \underline{\mathbf{F}_{l,i}} = N_{bl} \sum_i \frac{1}{2} C_{l,i} A_i \rho r_i^2 \underline{\omega_n}^2 = k \underline{\omega_n}^2, \quad [\text{N}] \tag{B.8}$$

$$k = N_{bl} \sum_i \frac{1}{2} C_{l,i} A_i \rho r_i^2 \tag{B.9}$$

The torque on the propeller shaft is, unlike the thrust, dependent on the direction of rotation of the propeller. Propeller 1 and 3 rotate clockwise (negative direction), and propeller 2 and 4 rotate counter-clockwise (positive direction). The drag torque is produced in the opposite direction of the propeller spin direction, and is for each element given by

$$\underline{\tau_{d,i}} = (-1)^{n+1} r_i \underline{\mathbf{F}_{d,i}} \quad [\text{N m}] \tag{B.10}$$

where the $(-1)^{n+1}$ expression yields -1 for propeller 2 and 4, ensuring negative torque (clockwise) from positive spin direction (counter-clockwise). This gives a total torque $\underline{\tau_{d,n}}$ of propeller n defined by Equation B.11.

$$\begin{aligned}
\underline{\tau_{d,n}} &= (-1)^{n+1} N_{bl} \sum_i \underline{\mathbf{F}_{d,i}} r_i = (-1)^{n+1} N_{bl} \sum_i \frac{1}{2} C_{d,i} A_i \rho r_i^3 \underline{\omega_n}^2 = b \underline{\omega_n}^2, \\
b &= (-1)^{n+1} N_{bl} \sum_i \frac{1}{2} C_{d,i} A_i \rho r_i^3 \quad [\text{N m}]
\end{aligned} \tag{B.11}$$

APPENDIX C

DESIGN OF QUADCOPTER

The propeller dimension and arm geometry is optimized for when the quadcopter is not accelerating, for energy efficient purposes, by dimensioning the propeller, its angular velocity, and dimensioning the arm for it to withstand the stress and oscillations while being long enough to not collide with the propellers.

SECTION C.1

INTRODUCTION

The optimized design variables will be, such that the quadcopter will be able to fly with an arm strength and natural frequency able to withstand small crashes and minor imbalances in the motor and propeller which could cause instability in the structure due to the resonance of the spinning components. These requirements should be fulfilled with the least amount of energy spent and with the only energy loss in the system being the drag on the propeller.

The propeller presumed to be defined only by its diameter, pitch, and angular velocity, whereas the arm is defined by its length and the cross sectional dimensions which can be seen in Figure C.2.

SECTION C.2

NOMENCLATURE

Symbol	Unit	Description
r, t, a, b, w, h	[m]	Profile dimensions seen in Figure C.2
A	[m^2]	Cross sectional area of the profile
c	[m]	Clearance between propeller tips
d_A	[m]	Distance from CG to center of thrust
E	[Pa]	Young's modulus for the material of the arm
D	[m]	Diameter of propeller
F_A	[N]	Force stressing the arm
F_l	[N]	Lifting force
g	[m s^{-2}]	Gravitational acceleration
J	[m^4]	Area moment of inertia
J_A	[m^4]	Area moment of inertia in the direction of F_A
J_x	[m^4]	Area moment of inertia in the x direction
J_y	[m^4]	Area moment of inertia in the direction perpendicular to x
M	[kg]	Mass of the propeller and motor combined
M_A	[N m]	Moment acting on the arm due to F_a at distance d_A
M_Q	[kg]	Total mass of the quadcopter
P	[W]	Power consumption
r_A	[m]	Distance from centerline of the arm to the surface of the arm in the direction of F_A
$S_{F\sigma}$	[—]	Safety factor for strength of arm
$S_{F\omega}$	[—]	Safety factor for resonance
α	[rad]	Angle from central coordinate system to arm (See Figure C.1)
ρ	[kg m^{-3}]	Mass density of the material of the arm
σ_{max}	[Pa]	The maximum stress produced by quadcopter in the arm
σ_{YS}	[Pa]	Yield strength of the material of the arm
τ_m	[N m]	Torque induced by drag only on the propeller
ω	[rad s^{-1}]	Angular velocity of the motor around the motor axis
$\omega_{n,arm}$	[rad s^{-1}]	Natural frequency of the arm

SECTION C.3

OBJECT FUNCTION

The only energy loss considered when optimizing the system, is the drag on the propeller which produces a torque opposing the direction of the spin of the propeller. This means that the power needed to keep the propeller spinning at constant speed can be described by

$$P = \tau_m \omega \quad [\text{W}] \quad (\text{C.1})$$

and will serve as the object function for this optimization.

This does not take into consideration any movement capability reduction caused by having a bigger structure or propeller.

The torque produced by the drag on the propeller is expressed using Equation B.11.

SECTION C.4

CONSTRAINTS

Besides the object function the system has several constraints which will be evaluated.

C.4.1 PROPELLER POSITION AND SIZE CONSTRAINT

To ensure the propeller blade tips do not collide when the quadcopter is flying, there must be a minimum distance between two adjacent propellers, which using Figure C.1 can be described by

$$d_A \cos(\alpha) > \frac{D + c}{2} \quad [\text{m}] \quad (\text{C.2})$$

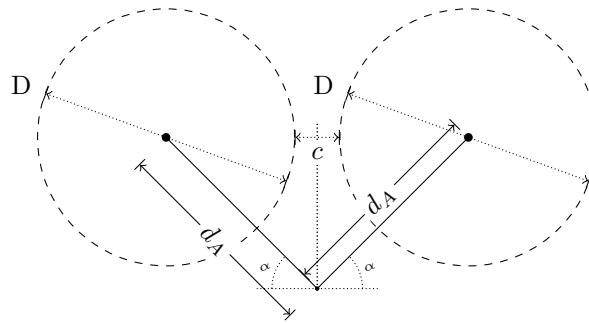


Figure C.1: Blade clearance constraint for initial optimization

C.4.2 LIFT CONSTRAINT

It is essential for the quadcopter that it is capable of lifting itself off the ground, meaning the lifting force from all propellers must be greater than the gravitational pull on the quadcopter. For the quadcopter to also maneuver around without losing altitude the lifting force must be greater than the gravitational pull by some margin. This margin is set to 2 which is significantly more, than what is needed for the quadcopter to lift off and should also correct for the simplifications made when estimating the lifting force.

$$4F_l \geq 2M_Qg \quad [\text{N}] \quad (\text{C.3})$$

where F_l is the lifting force of one propeller, M_Q is the total mass of the quadcopter, and g is the gravitational constant. The lifting force is calculated in Appendix B.

C.4.3 ARM STRENGTH CONSTRAINT

To ensure enough strength in the arm to not exceed the yield stress σ_{YS} of the arm material as well as a safety factor of 2 ($S_{F\sigma}$) which corrects for the simplified calculation of stress given by Equation C.4. If a more precise estimation was needed a Finite Element Analysis would have to be made using software like Solidworks or equal.

$$\sigma_{YS} > S_{F\sigma}\sigma_{max} \quad [\text{Pa}] \quad (\text{C.4})$$

The strength of the arm (σ_{max}) depends on the material used, the geometry of the profile, the force acting on the profile and the position inside the profile. For the optimization there are several profiles considered which are shown in Figure C.2.

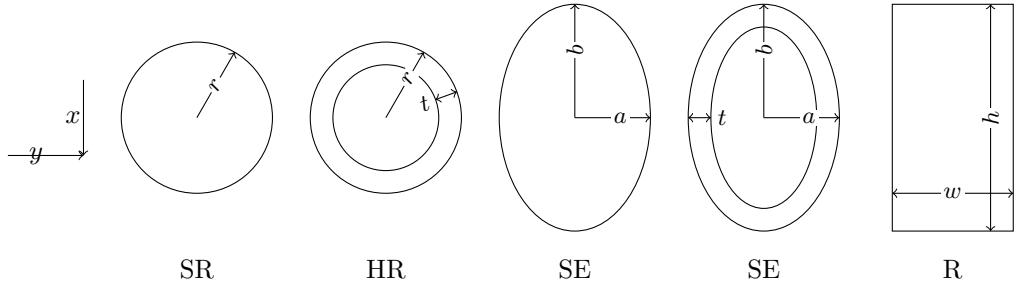


Figure C.2: The five shapes considered for the arm cross sectional shape when doing the initial optimization of the quadcopter

These geometries can be described in terms of cross sectional area by Table C.1

Profile	A	J_x	J_y
Solid rod (SR)	$2\pi r^2$	$0.25\pi r^4$	$0.25\pi r^4$
Hollow rod (HR)	$2\pi(r^2 - (\bar{r} - \bar{t})^2)$	$0.25\pi(\bar{r}^4 - (\bar{r} - \bar{t})^4)$	$0.25\pi(\bar{r}^4 - (\bar{r} - \bar{t})^4)$
Solid ellipse (SE)	$2\pi ab$	$0.25\pi ab^3$	$0.25\pi a^3 b$
Hollow ellipse (HE)	$2\pi(ab - (a - t)(b - t))$	$0.25\pi(ab^3 - (a - t)(b - t)^3)$	$0.25\pi(a^3 b - (a - t)^3(b - t))$
Rectangle (R)	wh	$(1/12)wh^3$	$(1/12)w^3 h$

Table C.1: Table for the calculation of the cross sectional area (A), the area moment of inertia in the x direction (J_x) and the direction perpendicular to x (J_y)

With these area moments of inertia, the stress acting upon the arm holding a force (F_A) at a distance (d_A) to the fastening point of the arm. The moment applied to the arm by this force can be described by

$$M_A = F_A d_A \quad [\text{N m}] \quad (\text{C.5})$$

This moment can be illustrated using Figure C.3

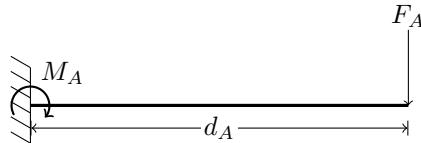


Figure C.3: Resulting moment (M_A) acting upon the arm by the force (F_A) at distance (d_A)

Which is then used to describe the maximum static stress in the arm in direction r_A by

$$\sigma_{max} = \frac{M_A r_A}{J_A} \quad [\text{Pa}] \quad (\text{C.6})$$

where the term r_A is the distance from the center of the arm to the surface of the arm which can be seen in Figure C.4, and J_A is the area moment of inertia in direction r_A

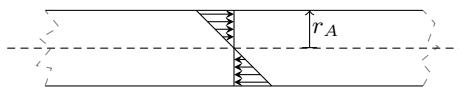


Figure C.4: The stress distribution inside the profile where some force to the right of the picture is acting upon the arm. It is seen that the stress increases as the distance to the center line increases.

C.4.4 NATURAL FREQUENCY CONSTRAINT

The natural frequency of the arm is a factor which must be taken into account as well when dimensioning the arm and the wanted angular velocity of the motor and propeller. Having these two frequencies near each other can cause any imbalance to start resonating and cause instability in the structure. Having the natural frequency of the arm well above the frequency of the motor will resolve this problem. This constraint can be defined by

$$\omega_{n,arm} > S_{F\omega}\omega \quad [\text{rad s}^{-1}] \quad (\text{C.7})$$

where $\omega_{n,arm}$ is the natural frequency of a cantilever beam, which can be seen in Figure C.5, and $S_{F\omega}$ is the safety factor for the angular velocity to further ensure that the angular velocity of the rotor is not near the natural frequency of the system. The cantilever is fixed at one end while free at the other with a mass attached to the free end as well as the mass of the arm taken into account which can be estimated by [37]

$$\omega_{n,arm} = \sqrt{\frac{3EJ}{(0.2235\rho Ad_A + M)d_A^3}} \quad [\text{rad s}^{-1}] \quad (\text{C.8})$$

Where E is Young' modulus for the specific material, J is the area moment of inertia in the direction of which the natural frequency is wanted known, ρ is the mass density of the material in the arm, A is the cross sectional area of the arm found in Table C.1, d_A is the length of the arm, and M is the mass of the object at the end of the arm.

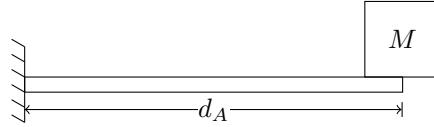


Figure C.5: Cantilever setup for determining the natural frequency of the entire system

With these constraints and the object function defined the optimization process and algorithm can be assessed.

SECTION C.5

OPTIMIZATION ALGORITHMS

There are many different optimization algorithms to find both the local minimum and to have a good chance of finding the global minimum. Amongst the algorithms to find local variables the MATLAB Optimization Toolbox provides several methods to find the minimum of a function for various object functions and different types of constraints. These algorithms can be adapted to solve for the global minimum instead of the local minimum. There are also other methods like Differential Evolution (DE) which takes a very different approach to find the global minimum of the object function.

Amongst the most popular functions in the MATLAB Optimization Toolbox are the *fminunc*, *fmincon*, and *fminsearch* algorithms. These algorithms combined with DE gives a broad range of tools to optimize different problems.

C.5.1 FMINUNC

Fminunc is used to find the minimum of a smooth nonlinear function which has no constraints of any type. A constraint can be added to the object function by introducing a penalty term when some design variable exceeds a boundary or a property of multiple design variables exceeds some value.

C.5.2 FMINCON

Fmincon adds the ability to make constraint of various types to object functions which are smooth, however, it does not handle non-smooth functions.

C.5.3 FMINSEARCH

Fminsearch adds the ability to find a local minimum of non-smooth object functions, however, it does not handle constraints and it too only finds the local minimum which depends on the initial guesses of design variables. There are ways that constraints can be added to the optimization which

C.5.4 DIFFERENTIAL EVOLUTION

DE is a subclass of the evolutionary methods and is radically different from fminunc, fmincon, and fminsearch since it does not use the typical optimization method where the derivative is found. Instead it operates with a number of designs and mixes these designs to find new designs after which it deletes the worst designs and thereby finds a local minimum within the bounded area which likely is the global minimum and fulfill the constraints. The disadvantage of this method is that it takes more calculations to find the minimum within the bounded area.

SECTION C.6

DESCRIPTION OF THE DIFFERENTIAL EVOLUTION ALGORITHM

As mentioned the DE optimization method does not use the typical optimization where the derivative is found, and an initial starting point is used. This section is largely based on the book [38].

This section uses the symbols

Symbol	Description
Cr	Crossover constant
F	Scaling factor
j_{rand}	Randomly chosen design variable in the mutant set which the next generation must inherit
k	Constraint
$\underline{L}, \underline{U}$	Lower and upper bound for initialization
Np	Number of members in the population
\underline{v}	Mutated population
\underline{x}	Population

C.6.1 OVERVIEW

The idea behind DE is to gradually evolve the entire population which are all the sets of design variables, by mixing other members of the population, to then be compared to the current population and the best designs are accepted into the next generation. The algorithm starts by creating random variables within a lower boundary (\underline{L}) and an upper boundary (\underline{U}), e.g. the lower boundary for the propeller diameter could be 0 [in] and the upper boundary could be 15 [in] since this is a good range to find a good variety of propellers regarding pitch. This lower and upper bound is made for all design variables. From here Np designs are made within the lower and upper bound. This can be seen in Figure C.6

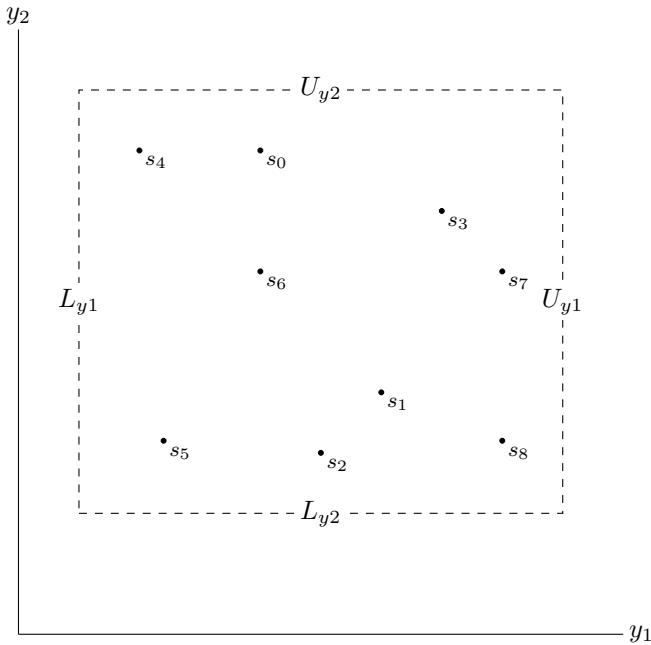


Figure C.6: 9 points randomly distributed within $\underline{\mathbf{L}}$ and $\underline{\mathbf{U}}$

A typical population size would be 10 times the number of design variables.

From here the actual optimization can start which follow the algorithm C.7. From Figure C.7 it is seen that there are 3 operations being done to get from one generation to selecting the members of the next generation

1. Generating the mutated population from the current generation
2. Crossover
3. Evaluation

GENERATING THE MUTATED POPULATION

One mutated member of the population is generated from 3 random members of the current population. From the example in Figure C.7 it is seen that

$$\underline{\mathbf{x}}_{0,g} = \text{round} \left(\underline{\mathbf{x}}_{2,g} + F \left(\underline{\mathbf{x}}_{3,g} - \underline{\mathbf{x}}_{Np-2,g} \right) \right) \quad (\text{C.9})$$

which can be generalized to

$$\underline{\mathbf{x}}_{a,g} = \text{round} \left(\underline{\mathbf{x}}_{b,g} + F \left(\underline{\mathbf{x}}_{c,g} - \underline{\mathbf{x}}_{d,g} \right) \right) \quad (\text{C.10})$$

where it is important to note that $\{a, b, c, d\} \in [0, Np - 1]$ and $a \neq b \neq c \neq d$. The F is for scaling the effect of the two sets being subtracted and depends on Np and Cr which will be explained when describing the crossover.

As a rule of thumb the F should have a value between 0.9 and 0.3 where the 0.9 is with a small population and 0.3 is with a big population. If any of the design variables are discrete this design variable is rounded to the nearest allowed value of the design variable.

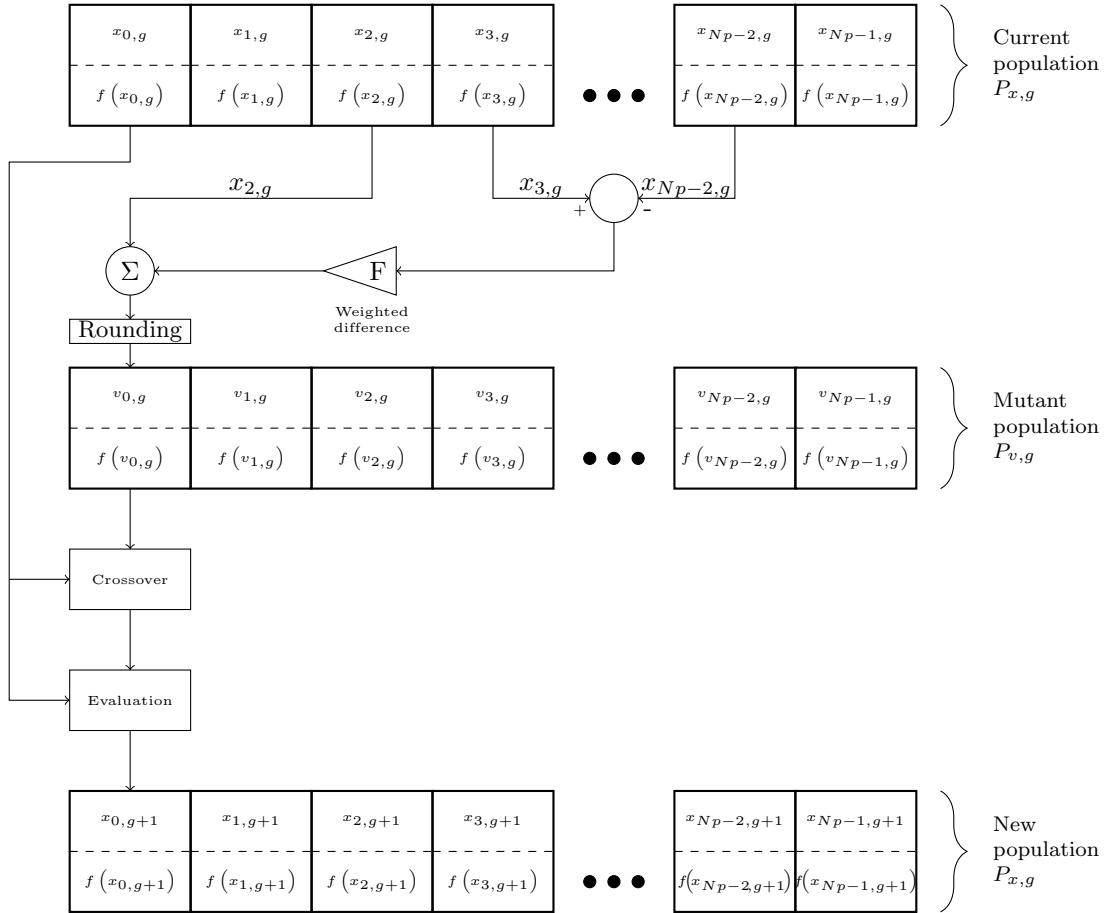


Figure C.7: Flowchart of one generation optimization using DE

CROSSOVER

Crossover or discrete recombination is a way of further mutating the mutant population member by making it inherit some quality from the member of the population which it is being compared to. This can be written as

$$\underline{v}_{i,g} = v_{j,i,g} = \begin{cases} v_{j,i,g} & \text{if } \text{rand}_j(0,1) \leq Cr \text{ or } j = j_{rand} \\ x_{j,i,g} & \text{otherwise} \end{cases} \quad (\text{C.11})$$

Which means that if the number generated randomly by $\text{rand}_j(0,1)$ is less than or equal to Cr the mutant inherits its own value, where Cr typically has a value of 0.9 meaning 90 % of the time it will inherit its own characteristic. The other case which can be true is if $j = j_{rand}$ which is implemented so that the mutant can never, due to crossover, inherit all the characteristics of the population member. j_{rand} is a random characteristic in the set of the mutant which cannot be changed due to crossover.

If neither of these cases are true the specific characteristic will be inherited from the member of the population of which it is being compared to. As an example a cross over situation where $Cr = 0.9$ can be given where

$$\underline{v}_{i,g} = \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} \quad \underline{x}_{i,g} = \begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} \quad \underline{r} = \text{rand}_j(0, 1) = \begin{bmatrix} 0.4 \\ 0.2 \\ 0.95 \end{bmatrix} \quad j_{rand} = 2$$

From this it is seen that

$$\underline{r} \leq Cr = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \quad j_{\text{rand}} \Rightarrow \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

where 1 means that the mutant characteristic is inherited from the mutant and if there is only zeros in the row of the characteristic, as is the case with characteristic 3, the characteristic is inherited from the population member. This means that the mutant after crossover is defined by

$$\underline{\mathbf{v}_{i,g}} = \begin{bmatrix} v_a \\ v_b \\ x_c \end{bmatrix}$$

EVALUATION

In the evaluation the mutant is compared to the member of the population, to find which set will be carried to the next generation. If there are no constraints this evaluation is a straight forward comparison of which object function is the smallest. However, when implementing the constraints this decision becomes significantly more complicated. There are many different ways of handling these constraint violations if the mutant has a lower object function

1. Only choose mutant if no constraints are violated
2. Only choose mutant if all constraint violations are less violated than the violations of the member of the population it is being compared to
3. Only choose mutant if the number of constraints violated are smaller than that of the population
4. Estimate if the mutant is violated more than the population even if one or more violations are bigger than the population

In the case of Figure C.8 it is seen that the x violates k_1 and k_2 while v_1 only violates k_2 and not k_1 . v_2 only violates k_2 and fulfills k_1 , however it violates k_2 with more than the population.

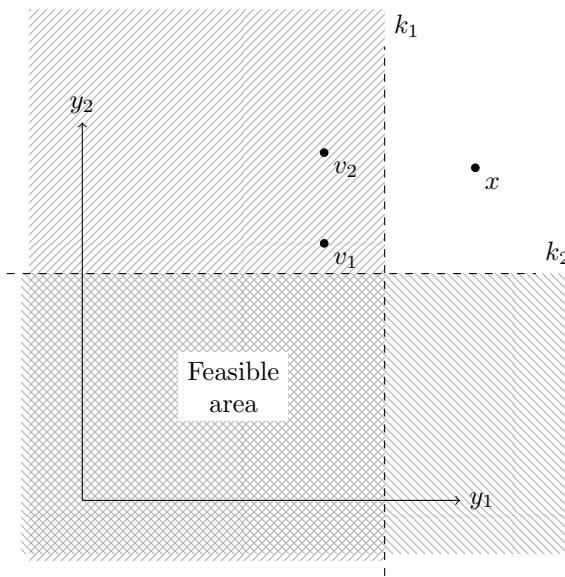


Figure C.8: Population and mutant constraint violation with 2 different mutants (v_1, v_2) which can be compared to the member of the population (x)

This means that using method 1 x will proceed and both v_1 and v_2 will lose. Using method 2 v_1 will proceed when comparing to x , however, v_1 will lose to x , since k_2 is violated a little more even though k_1 is improved. In method 3 x will lose, to both mutants since x violates two constraints and both v_1 and v_2 only violate one constraint each. Method 4 requires a further definition of how the value of a constraint violation can be compared to other constraint violations. Typically this is done by adding a penalty term to the object function. There are 3 main types of penalties

1. Brick wall penalty
2. Internal penalty
3. External penalty

These 3 types of penalty can be assessed using the example where the object function (f) is defined by $f = -y + 10$ and a constrain (k) is set at $k = 7$ so that $y \geq k$ is violating the constrain which when transformed into the typical notation of constraints is defined by $0 \geq g \Leftrightarrow 0 \geq y - k$. The effect of the three penalties can be seen in Figure C.9

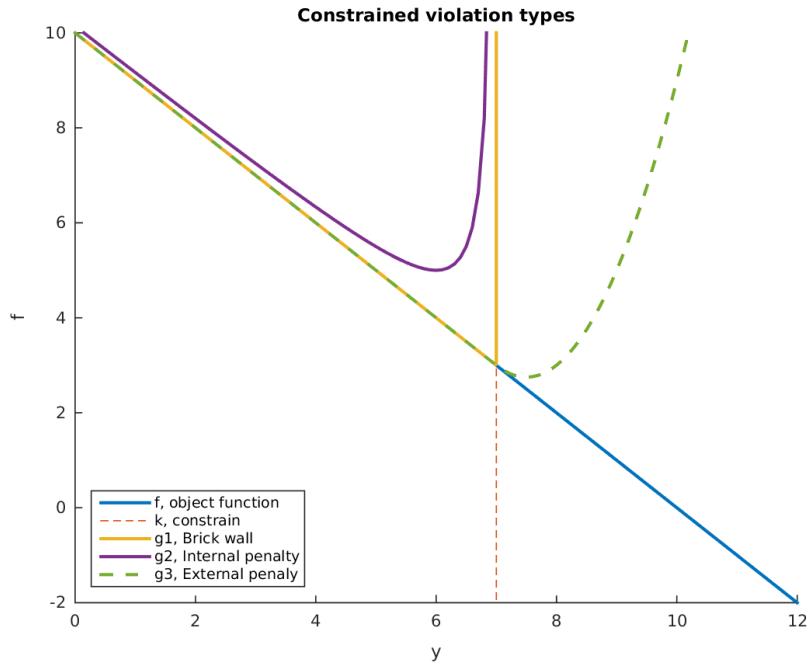


Figure C.9: The three different types of constrain violations being added to the object function f

Brick wall penalty This approach adds a constant with value infinite to the object function if the constrain is being violated. This can be written as

$$f = \begin{cases} f & \text{if } g \leq 0 \\ f + \infty & \text{otherwise} \end{cases} \quad (\text{C.12})$$

Though it is easy to implement this has the disadvantage of making the function non-smooth and therefore some solver types cannot be used.

Internal penalty This penalty type keeps the object function smooth at the cost of finding a local minimum which in reality is not the actual local minimum if the actual local minimum lies on the constraint. The internal penalty can be defined by

$$f = \begin{cases} f - \frac{1}{rg} & \text{if } g \leq 0 \\ f + \infty & \text{otherwise} \end{cases} \quad (\text{C.13})$$

where r scales how sharp this penalty will be and thereby how close it will come to the constraint which can be seen in Figure C.10. It is possible for the object function to be assessed in the infeasible zone to the right of k and if this happens, the penalty will be negative and therefore the second case is added where, if $g > 0$ the object function will have a value of ∞ .

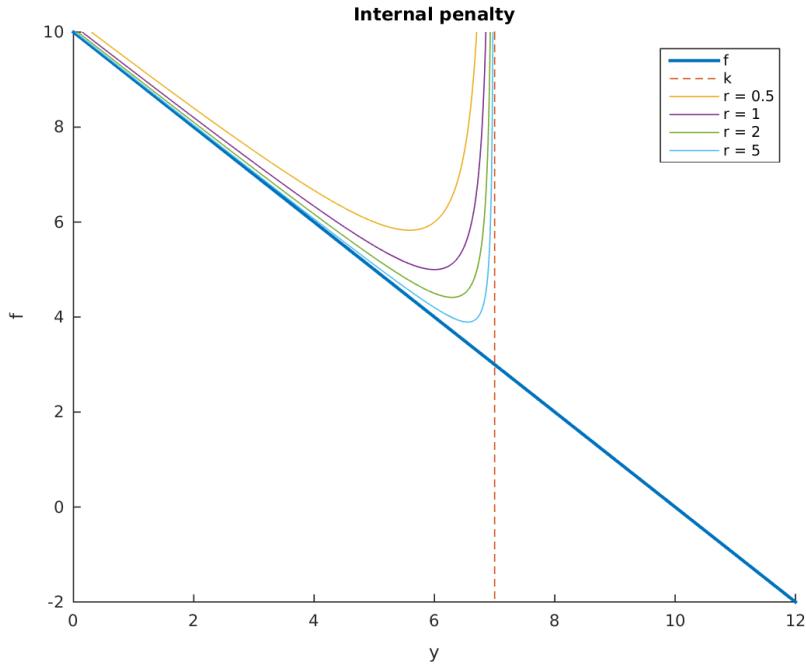
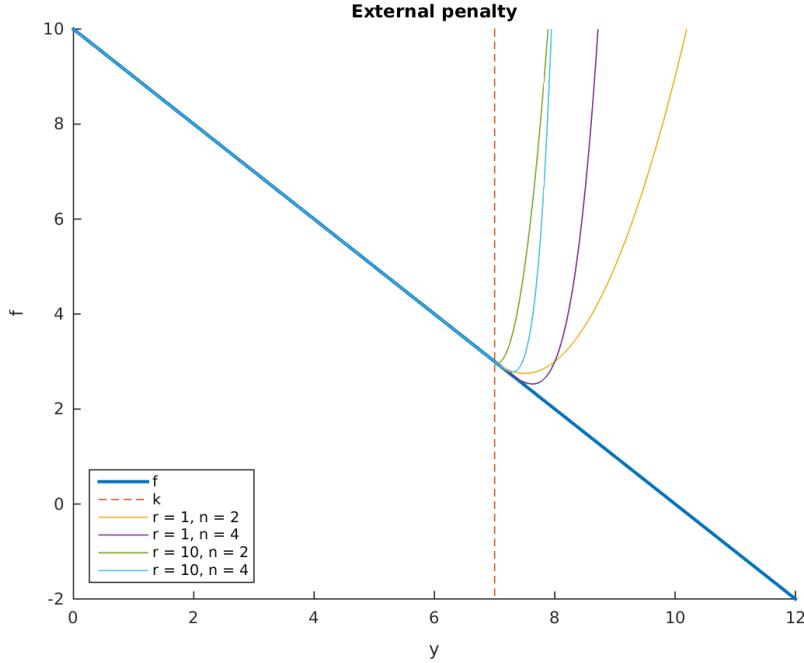


Figure C.10: Effect of scaling r when using internal penalty

External penalty This penalty type has the benefit of not effecting the object function while the constraint is not violated, however, when the constraint is violated the penalty is very small in the beginning and then takes effect. This delay can be tweaked by adjusting the parameters r and n in Equation C.14 and the effect can be seen in Figure C.11.

$$f = f + r \max(0, g)^n \quad (\text{C.14})$$


 Figure C.11: Effect of scaling r and n when using external penalty

It is seen on Figure C.11 that especially external penalties with a low r has a minimum that far exceeds the constrain at $y = 7$.

Choosing Constraint Violation Method When choosing how to handle any constraint violation method, it is not only important how well the constraint is handled, but also how long it takes to compute whether the mutant or the existing population should proceed to the next population. The best way to handle these constraint violations depends on the specific problem. In general the longer time it takes to calculate the value of the mutant the more time should be spent on making the correct evaluation. This becomes especially clear when convergence is needed to get the value of the object function, as is the case with Computation Fluid Dynamics (CFD) and Finite Element Methods (FEM) which can take a long time to compute. Typically DE is not used for models which takes a long time. This is due to the number of calculations typically being higher in DE than when using other optimization methods.

In general method two, which accepts the mutant if and only if all violated constraints are violated less than they were in the population that it is being compared to, is often chosen due to the little computing time and ensuring that the design never gets worse than it was before.

SECTION C.7

RESULT OF OPTIMIZATION

The optimization has been run for the rectangular profile, only due to the other shapes being difficult to handle and built around, with boundaries given by:

	D [in]	p [in]	ω [rev/min]	d_A [m]	w [m]	h [m]
Lower	4	4	0	0	0	0
Upper	15	10	10,000	1	0.1	0.1

The propeller dimension boundaries (D, p) are chosen around the commercially available propellers. Further a constraint for the width-height ratio is made which was observed to be approximately 2:1 which gives results of

$$D = 15 \quad [\text{in}] \quad (\text{C.15})$$

$$p = 6.557 \quad [\text{in}] \quad (\text{C.16})$$

$$RPM = 1.458 \quad [\text{RPM}] \quad (\text{C.17})$$

$$L = 0.283 \quad [\text{m}] \quad (\text{C.18})$$

$$w = 14.2 \quad [\text{mm}] \quad (\text{C.19})$$

$$h = 7.1 \quad [\text{mm}] \quad (\text{C.20})$$

From this it is seen that the propeller diameter is maximized to the boundary meaning there is likely a more feasible point at higher diameters, however, these are not typically commercially available at normal web shops at sizes this big. It is further observed that the ratio between the width and height is at 2:1 which is the constraint set. This indicates that the area moment of inertia in the y direction is the limiting factor which is realistic due to the low stress exerted by the propeller during take off.

With this observation it can be considered if the constraint for natural frequency is necessary since the propeller and rotor can be balanced after installation to make sure the resonance, if it occurs, will be small.

APPENDIX D

THRUST AND INERTIA CHARACTERISTICS

Characteristics of the motor and propeller are made to be able to implement it in the model of the quadcopter for simulations, and to design the regulator for the physical quadcopter.

SECTION D.1

NOMENCLATURE

For this appendix, a local nomenclature is defined which should not be confused with the nomenclature of the main report.

Symbol	Unit	Description
a_0, ω_F, a_n, b_n		Fourier variables and coefficients
A, B, C	[m]	Position of object at rest, when displaced by an angle θ , and the revolving axis
D_c	[–]	PWM duty cycle input to the ESC
F_g	[N]	Gravitational pull on the object
F_l	[N]	Lifting force of propeller
\mathbf{F}_o	[N]	Opposing force by the object to force the string around the revolving axis instead of through the circle peripheral spanned by the object
$\mathbf{F}_s, \mathbf{F}_{s,L}, F_{s,z}$	[N]	String force along the string, along L , along the vertical axis
g	[m s^{-2}]	Gravitational constant
H	[m]	Seen in Figure D.5 as the length of the string
J	[kg m^2]	Moment of inertia of the object around the revolving axis
k_{1-5}	[–]	Exponents for the general bifilar equation
K	[–]	Constant for the expression for the moment of inertia
K_1, K_2		Constants for the equation defining angular velocity as a function of input voltage and duty cycle
L	[m]	Projection of the string onto the horizontal plane at the height of point B
m	[kg]	Total mass of propeller and rotor
r	[m]	Distance from the revolving axis to the point of which the string is attached to the object
RPM	[rev min^{-1}]	Angular velocity of rotor and propeller (proportional to ω)
R_t	[N]	Resulting force on the object at point B making the object turn towards the resting position
t	[s]	Time
T	[s]	Oscillation period for one cycle
Thr	[gram]	Thrust of the propeller (proportional to F_l)
V	[V]	Supply voltage to the ESC from a battery or power supply
α	[rad]	Angle between string and L axis
θ	[rad]	Position of the object
$\ddot{\theta}$	[rad s^{-2}]	Angular acceleration of the object
θ_a	[rad]	Angle between \mathbf{F}_o and $\mathbf{F}_{s,L}$
τ	[N m]	Torque around the revolving axis
ω	[rad s^{-1}]	Angular velocity of the rotor and propeller

SECTION D.2

PURPOSE

The purpose of characterizing the motor and propeller is to find expressions for the dynamics of the rotor and propeller. The purpose of the experimental characterization is to get expressions for:

1. Thrust and steady state angular velocity of rotor and propeller as a function of the input signal to the ESC
2. Mass moment of inertia of the rotor and propeller around the rotor axis

The angular velocity as function of input signal to the ESC, and the thrust as a function of the angular velocity, is determined by one experiment, while the effect of applied voltage levels is tested over a series of experiments conducted in the same manner.

SECTION D.3

STEADY STATE ANGULAR VELOCITY AND THRUST

The specific purpose of this experiment is to find the angular velocity of the propeller and rotor as a function of the pulse width to the ESC and the thrust as function of angular velocity when at steady state.

METHOD

The experiment is made by fastening one of the motors with an attached propeller on to a fixed arm, which prevents the motor from moving when the propeller is spinning. The motor is connected with one of the ESCs which is connected to a power supply and a Arduino Duemilanove. The Arduino is provided by UASworks together with code, and is used to input a PWM signal to the ESC and also to make measurements of thrust and current. A strain gauge is applied to the fixed arm, and is used to measure the thrust produced by the spinning rotor and propeller. A current transducer is used to measure the current in the motor. Furthermore an infrared tachometer is used to determine the steady state angular velocity of the rotor. The experiment is conducted by a setup with a surrounding shield for protection. The setup for the experiment is shown on Figure D.1.

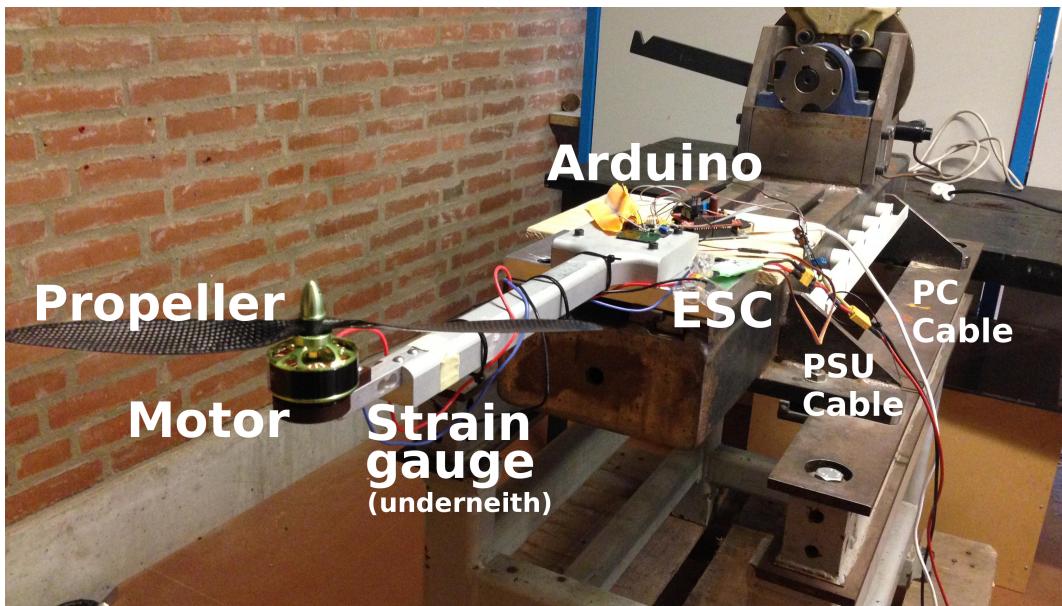


Figure D.1: Experimental setup to determine the thrust and angular velocity of the motor and propeller.

Items used to conduct the experiment is as follows:

- Fixed arm for the motor to be fastened to
- Brushless DC-Motor (same kind as used for the quadcopter)
- Propeller (same kind as used for the quadcopter)
- ESC (same kind as used for the quadcopter)
- Power supply (Delta Elektronika SM52-30, 60353)
- Arduino Duemilanove
- Computer to upload the code to the Arduino and log the serial data
- Strain gauge
- Current transducer
- Infrared Tachometer
- Protective shielding

The experiment is carried out by applying PWM signals with different duty cycles to the ESC while the Arduino is logging data, and while the steady state angular velocity measured by the tachometer is noted manually.

The signal sent to the ESC is updated with a frequency of 50 [Hz] and the duty cycle is in the range of 3.5 % to 10 % which is equivalent to the minimum and maximum signal, respectively. For further reference the minimum PWM input (3.5 %) will be noted as 0, and the maximum PWM input (10 %) will be noted as 1.

The voltage level of the power supply is set within the range 14.8 [V] and 16.8 [V] to represent the recommended battery voltage range, when for each cell voltage on the four cell battery is between 3.7 [V] and 4.2 [V].

RESULTS AND DATA PROCESSING

The data from experiments being conducted will be processed in order to find the thrust as a function of the angular velocity, and the angular velocity as a function of the input voltage and duty cycle to the ESC.

FILTERING OF DATA

During the experiment a systematic oscillation was observed on the thrust measurement, which has been filtered using the Savitzky-Golay filter which approximates a low degree polynomial for a limited number of points using the least squares method. The effect of the filter can be seen in Figure D.2.

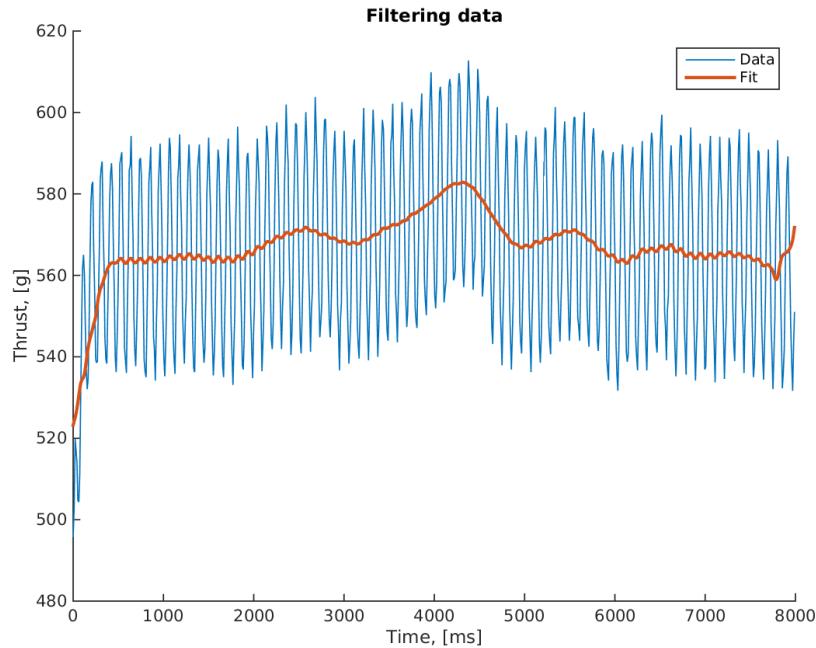


Figure D.2: Effect of Savitzky-Golay filter on the thrust when a constant duty cycle is applied. The data is a sample of the entire dataset during one experiment cycle

THRUST

With the applied filter to the thrust a fit is made using the data from the experiments with

- 2 experiments at 14.8 [V]
- 1 experiment at 15.8 [V]
- 2 experiments at 16.8 [V]

Figure D.3 shows that the thrust at specific angular velocities does not depend on the input voltage. The data from the two 14.8 [V] and the two 16.8 [V] experiments are represented by the average of the two experiments.

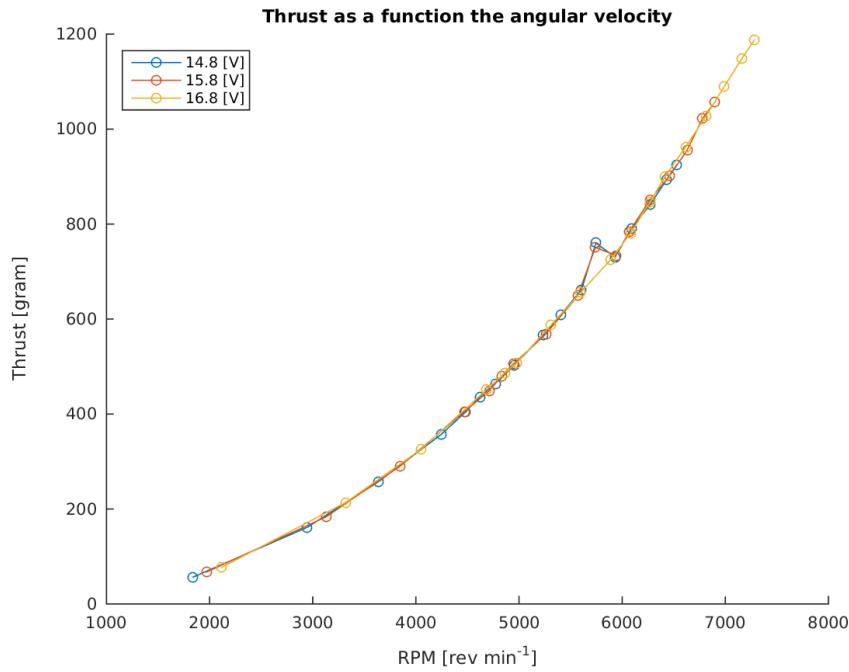


Figure D.3: Thrust exerted by the propeller as a function of the angular velocity of the propeller. At around 5000-6000 [RPM] the setup made significantly more noise than at lower and higher angular velocities which may have caused the spike on the 14.8 [V] and 15.8 [V].

The 2nd degree polynomial fit for this data is described by

$$Thr = (2.15 \times 10^{-5}) RPM^2 \quad [\text{gram}] \quad (\text{D.1})$$

which in SI units can be expressed by

$$F_l = (2.11 \times 10^{-4}) \omega^2 \quad [\text{N}] \quad (\text{D.2})$$

ANGULAR VELOCITY

Figure D.4 shows the measured angular velocity as a function of the duty cycle of the PWM signal with different applied voltages.

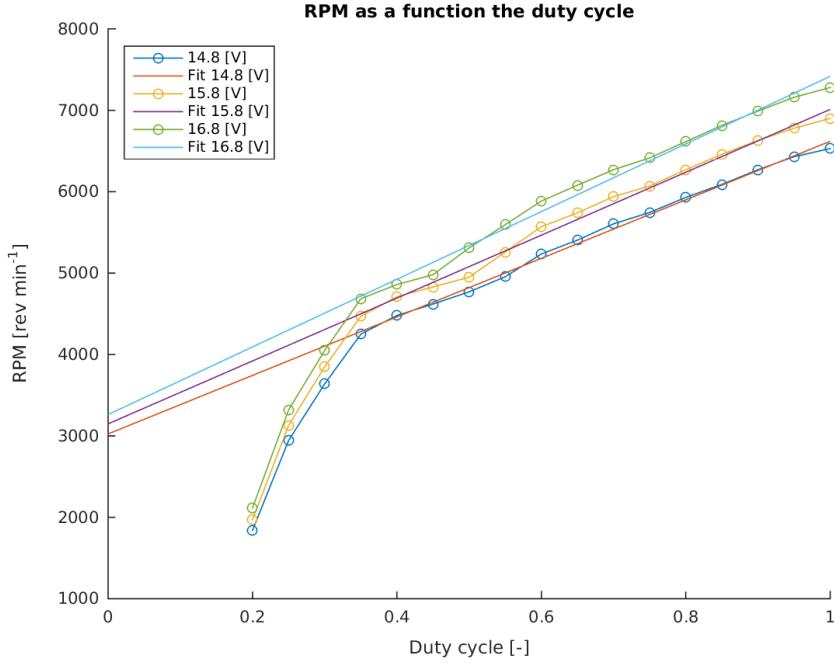


Figure D.4: Collected data of angular velocity [RPM] and duty cycle [-] with different applied voltages along with the linear fit for each voltage

It is seen that the graph changes significantly at a duty cycle of around 0.35. From 0.35 to 1 the behavior for all voltage inputs are approximately linear. When comparing the angular velocity at 0.35, which is around 4500 [RPM], the thrust in Figure D.3 is approximately 400 [gram] which is below the typical operating thrust when the motor is mounted to the quadcopter. This means that the fit on Figure D.4, which is only made for a duty cycle greater than 0.35, is valid for this application.

It is also observed, both on the data and the fit, that a higher voltage input yields a higher angular velocity for the same duty cycle input.

This has to be accounted for, and therefore the function for angular velocity will depend on both the duty cycle input and the voltage input. The fit for the three different voltages can be approximated by

$$RPM = K_1 D_c + K_2 \quad [\text{rev min}^{-1}] \quad (\text{D.3})$$

where D_c is the input duty cycle and the constants K_1 and K_2 for the different voltages are

V [V]	K_1	K_2
14.8	3.59×10^3	3.02×10^3
15.8	3.86×10^3	3.15×10^3
16.8	4.16×10^3	3.26×10^3

where V is the input voltage to the ESC. The fit for the variables K_1 and K_2 from the table above can be estimated by

$$K_1 = K_{1a} V + K_{1b} \quad [\text{rev min}^{-1}] \quad (\text{D.4})$$

$$K_2 = K_{2a} V + K_{2b} \quad [\text{rev min}^{-1}] \quad (\text{D.5})$$

where $K_{1a} = 285$, $K_{1b} = -633$, $K_{2a} = 120$, $K_{2b} = 1,247$. This results in a final equation for the angular velocity as a function of the duty cycle and voltage given by

$$RPM = (K_{1a}V - K_{1b}) D_c + K_{2a}V + K_{2b} \quad [\text{rev min}^{-1}] \quad (\text{D.6})$$

which in SI units is defined by

$$\omega = ((K_{1a}V - K_{1b}) D_c + K_{2a}V + K_{2b}) \frac{2\pi}{60} \quad [\text{rad s}^{-1}] \quad (\text{D.7})$$

CONCLUSION

With the conducted experiments, the propeller, motor, and ESC unit has been tested to find a function for the thrust at different angular velocities. The duty cycle necessary to achieve this angular velocity at steady state has also been found for different voltage inputs which may occur during flight of the final quadcopter.

SECTION D.4

MASS MOMENT OF INERTIA

Finding the moment of inertia of the propeller and rotor is essential when describing how they behave dynamically. However, due to the complex structure and the uneven weight distribution it is difficult to approximate using simpler shapes, or to compute it using a CAD program. Therefore the moment of inertia is found experimentally.

D.4.1 SETUP

The rotor and propeller are suspended in a system as seen in Figure D.5, and then turned by a small angle around its center axis, and the time period for one oscillation is measured by video. The actual setup can be seen in Figure D.6

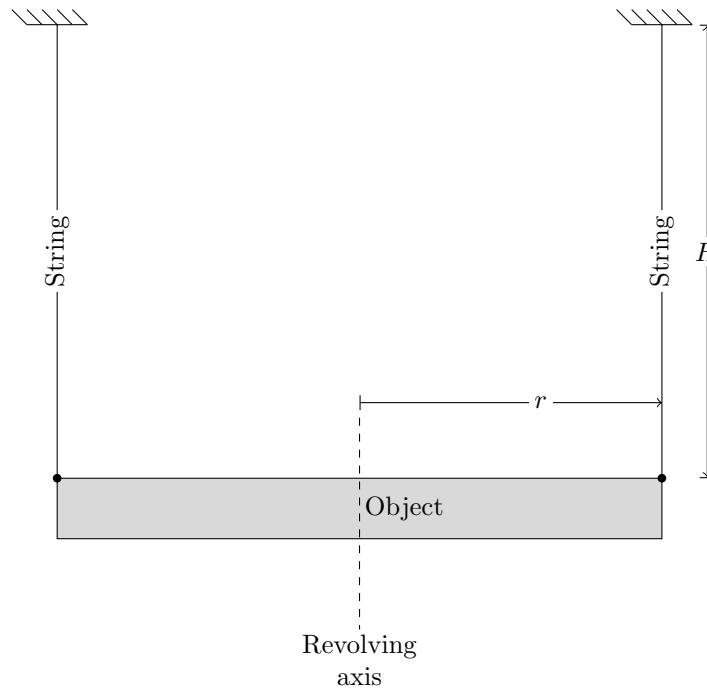


Figure D.5: Experimental setup for test of moment of inertia around the revolving axis of an object.

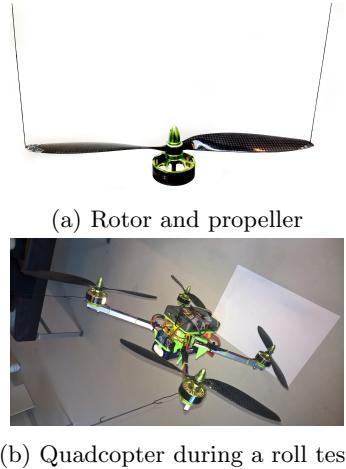


Figure D.6: Bifilar pendulum experiments. Due to the perspective of the photographs the strings look like they are not parallel, however, they are parallel when at rest.

D.4.2 MATHEMATICAL DESCRIPTION OF THE SETUP

In this section an expression for the inertia of a object suspended by two strings will be derived by numerically determining the constants of the general equation

$$T = K J^{k_1} r^{k_2} m^{k_3} g^{k_4} H^{k_5} f(\theta) \quad [\text{kg m}^2] \quad (\text{D.8})$$

where T is the oscillation period, K is some constant, J is the moment of inertia around center axis, r is the distance from the center of revolution to the string, g is the gravitational constant, H is the length of the strings suspending the object from above, $f(\theta_0)$ is a function describing the starting angle's effect on the oscillation period. The power constants k_n for $n = 1, 2, \dots, 5$ will be determined by simulation.

Equation D.8 neglects the damping which will affect the object when it is moving. This damping could come from drag on the object and string, or the way the string is attached to the ceiling, and is assumed to be small enough to have a negligible effect on the system.

D.4.3 PHYSICAL BEHAVIOR

To derive the movement of the object when let go from it's starting angle, the system will be examined and simulated to determine the oscillation period. The system seen from above can be simplified to Figure D.7 where A is the object's position at rest, B is the object's position when displaced by an angle of θ , C is the revolving axis that the object is revolving about, r is the distance from C to A or C to B which are equal, and L is distance between point A and point B .

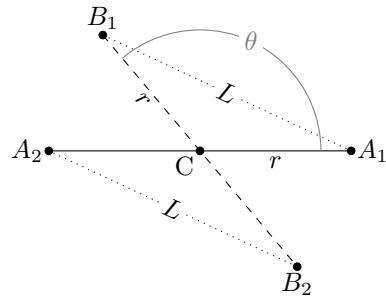


Figure D.7: Overview of system when at rest and displaced

Having the overview of Figure D.7 makes it possible to add the forces acting on the object when it is not at rest. These forces are shown in Figure D.8 where $F_{s,L}$ is the force from the string pulling on the object to return to rest position along the line L , F_o is the force from the object opposing the force of the string, such that the string does not enter the peripheral that the object spans when rotating. R_t is the resulting reaction which turns the object around its revolving axis.

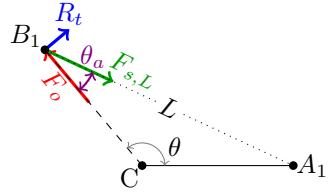


Figure D.8: Forces acting on the object and resulting reaction

The magnitude of F_s originates from the gravitational pull on the object, which can be found by looking at the object in the plane spanned by the vertical axis (z) and the line L in Figure D.9. The z component of F_s must be equal to F_g for the object to not be moving along the vertical axis. When the experiment is conducted, it is further observed that the height difference from initial position to relaxed position is very small and the system is simplified to be described by

$$F_{s,z} = F_g = \frac{mg}{2} \Rightarrow [N] \quad (D.9)$$

$$\|F_s\| = \frac{F_g}{\sin(\alpha)} \quad [N] \quad (D.10)$$

where it is noted that F_g in the point B_1 is only equal to half the typical definition of $F_g = mg$ due to the object being suspended in two strings. The angle α is described by the string length, L , and the right angle near A by

$$\alpha = \frac{\pi}{2} \sin\left(\frac{\sin(\frac{\pi}{2})L}{H}\right) \quad [\text{rad}] \quad (D.11)$$

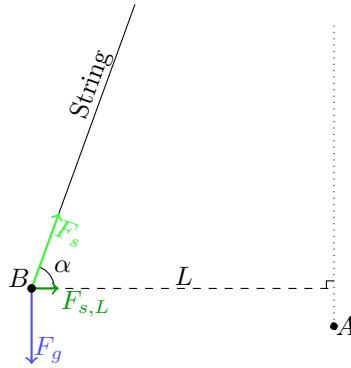


Figure D.9: Gravitational pull on the object and the strings opposing reaction making sure that the object does not fall to the ground. It is also observed that as B is moved away from point A the object has a slight increase in altitude due to the constant string length.

It is further seen that the magnitude of the component of F_s which lies on the L -axis can be described by

$$||F_{s,L}|| = \cos(\alpha)F_s \quad [\text{N}] \quad (\text{D.12})$$

From Figure D.8 it is seen that the magnitude of F_o can be calculated using the angle between $F_{s,L}$ and F_o which is defined by

$$\theta_a = \frac{\pi - |\theta|}{2} \quad [\text{rad}] \quad (\text{D.13})$$

which in turn yields F_o as shown in Equation D.14.

$$||F_o|| = \cos(\theta_a) F_{s,L} \quad [\text{N}] \quad (\text{D.14})$$

With the magnitudes defined the directions of the vectors $\underline{\mathbf{F}_{s,L}}$ and $\underline{\mathbf{F}_o}$ are given by

$$\angle F_{s,L} = \begin{cases} \frac{\theta - \pi}{2} & \text{for } \theta > 0 \\ \frac{\theta + \pi}{2} & \text{for } \theta < 0 \end{cases} \quad [\text{rad}] \quad (\text{D.15})$$

$$\angle F_o = \theta \quad [\text{rad}] \quad (\text{D.16})$$

where the cartesian coordinates can be found, and R_t is found by

$$R_t = \underline{\mathbf{F}_{s,L}} + \underline{\mathbf{F}_o} \quad [\text{N}] \quad (\text{D.17})$$

The torque around the point C is found by

$$\tau = 2R_t r \quad [\text{Nm}] \quad (\text{D.18})$$

where the "2" comes from the object being suspended by two strings. This torque accelerates the object by the relation

$$\ddot{\theta} = \frac{\tau}{J} \quad [\text{rad s}^{-2}] \quad (\text{D.19})$$

From where the angle at any time can be found using integration and the oscillation period is found when $\theta = \theta_0$ [rad] after the release of the object when $t = 0$ [s].

D.4.4 VARIABLE DEPENDENCE

This section will vary a single variable while keeping all other variables constant in order to find the specific variable's impact on the oscillation time. The value of the variables when constant, and the range of it when being varied is found in Table D.1

Variable	Constant	Minimum range	Maximum range
J [kg m ²]	0.01	0.001	1
r [m]	0.1	0.05	1
m [kg]	0.1	0.01	1
g [m s ⁻²]	9.82	1	20
H [m]	2	0.5	4
θ_0	10	1	179

Table D.1: Constants and ranges for simulation variables

The fits are made using the Curve Fitting Toolbox in MATLAB.

INERTIA DEPENDENCE

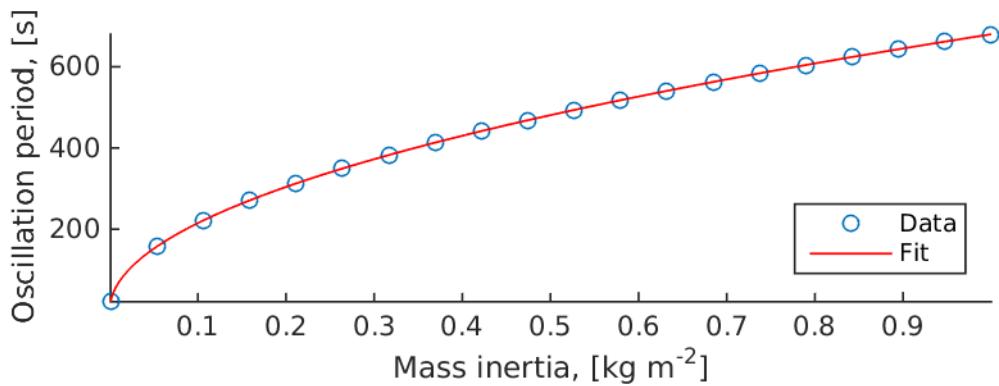


Figure D.10: Varying the inertia of the object within the range at 20 different points

The fit has the form of a power equation given by

$$T = CJ^{0.5}, \quad R^2 = 1 \quad [\text{s}] \quad (\text{D.20})$$

where C is the impact of all other variables on the system.

RADIUS DEPENDENCE

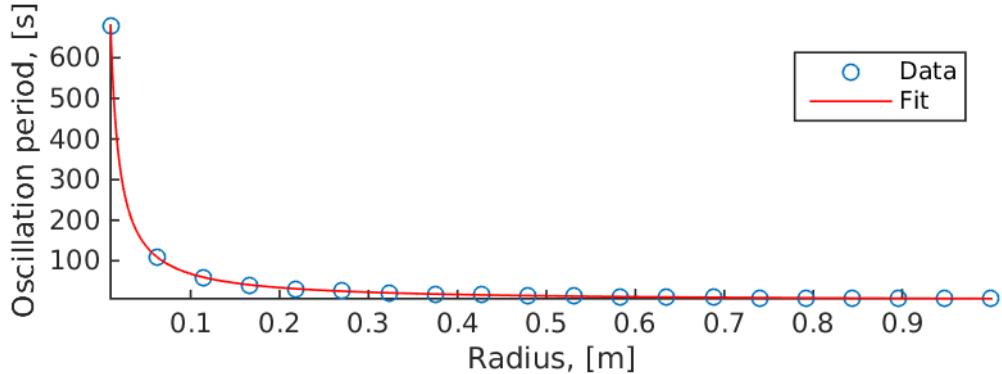


Figure D.11: Varying the radius of the object within the range at 20 different points

The fit has the form of a power equation given by

$$T = CR^{-1}, \quad R^2 = 1 \quad [\text{s}] \quad (\text{D.21})$$

where C is the impact of all other variables on the system.

MASS DEPENDENCE

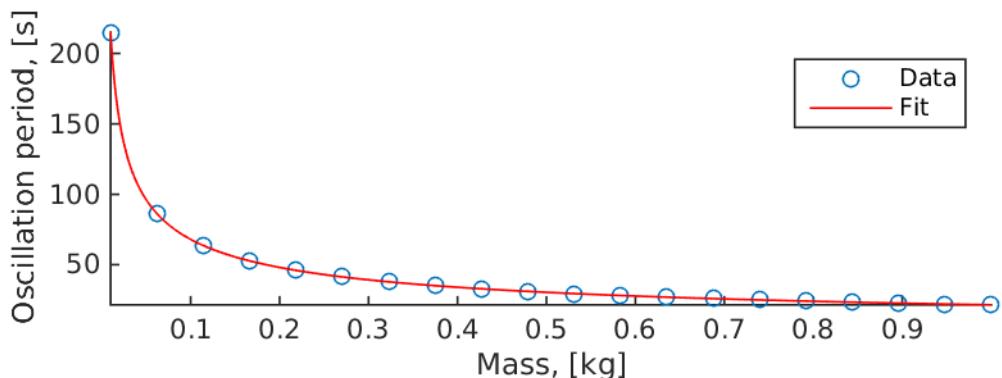


Figure D.12: Varying the mass of the object within the range at 20 different points

The fit has the form of a power equation given by

$$T = Cm^{-0.5}, \quad R^2 = 1 \quad [\text{s}] \quad (\text{D.22})$$

where C is the impact of all other variables on the system.

GRAVITATIONAL PULL DEPENDENCE

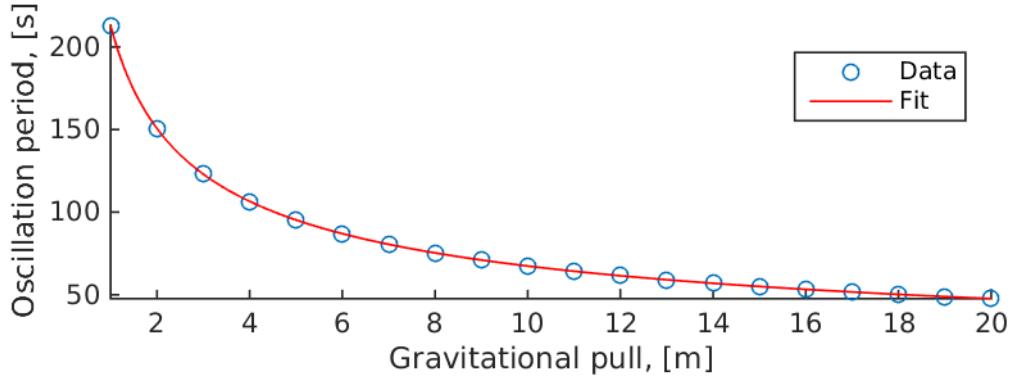


Figure D.13: Varying the gravitational pull within the range at 20 different points

The fit has the form of a power equation given by

$$T = Cg^{-0.5}, \quad R^2 = 1 \quad [\text{s}] \quad (\text{D.23})$$

where C is all other variables impact on the system.

STRING LENGTH DEPENDENCE

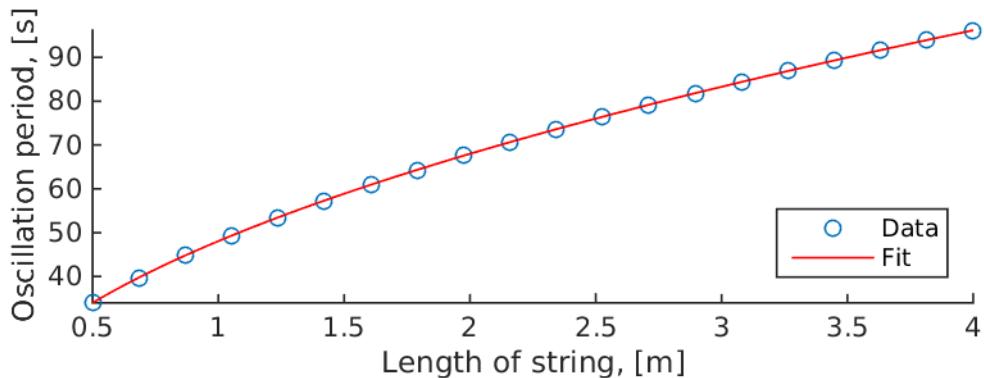


Figure D.14: Varying the length of the string within the range at 20 different points

The fit has the form of a power equation given by

$$T = CH^{-1}, \quad R^2 = 1 \quad [\text{s}] \quad (\text{D.24})$$

where C is the impact of all other variables on the system.

STARTING ANGLE DEPENDENCE

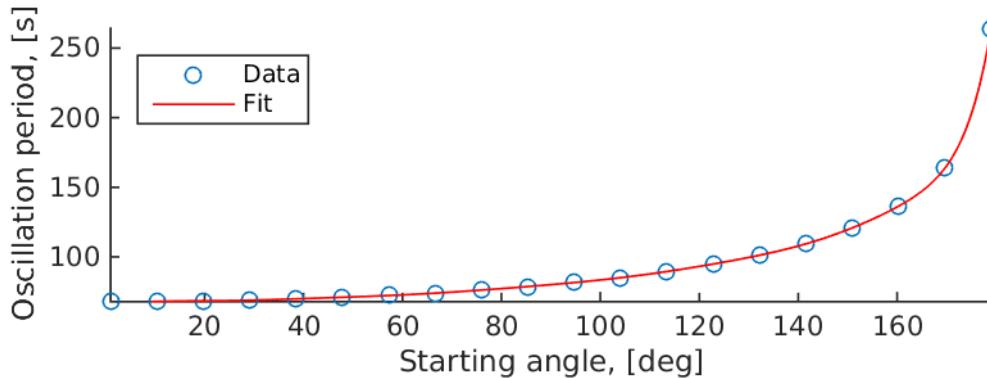


Figure D.15: Varying the starting angle within the range at 20 different points

It is observed that the data cannot be approximated by a power equation and is instead approximated by a sixth order Fourier series which in general is given by

$$T = a_0 \sum_{n=1}^6 a_n \cos(n\theta_0 \omega_F) + b_n \sin(n\theta_0 \omega_F), \quad R^2 = 1 \quad [\text{s}] \quad (\text{D.25})$$

where a , b , and ω_F are given in below.

$$\begin{aligned} a_0 &= 1.098 \times 10^{10} & \omega_F &= 5.032 \times 10^{-3} \\ a_1 &= -1.725 \times 10^{10} & b_1 &= -7.724 \times 10^9 \\ a_2 &= 7.955 \times 10^9 & b_2 &= 8.91 \times 10^9 \\ a_3 &= -1.639 \times 10^9 & b_3 &= -5.154 \times 10^9 \\ a_4 &= -1.881 \times 10^8 & b_4 &= 1.655 \times 10^9 \\ a_5 &= 1.594 \times 10^8 & b_5 &= -2.696 \times 10^8 \\ a_6 &= -2.221 \times 10^7 & b_6 &= 1.571 \times 10^7 \end{aligned}$$

It is further observed that the impact of the starting angle on the oscillation time is very small, which is also observed from $T(1) = 67.9$ [second] and $T(45) = 70.6$ [s].

CONSTANT

The constant (K) is found by solving for K in Equation D.8 which gives

$$K = \frac{T(J, r, m, g, H)}{J^{0.5} r^{-1} m^{-0.5} g^{-0.5} H^{0.5} f(\theta_0)} = 0.3205 \quad [-] \quad (\text{D.26})$$

and with K defined, the expression for the inertia can be derived by solving for J in Equation D.8.

$$J = \frac{mgr^2 T^2}{K^2 H(f(\theta))^2} \quad [\text{kg m}^2] \quad (\text{D.27})$$

This expression is further supported by the doing a unit test on the expression as seen in

$$[\text{N m}] = \frac{[\text{kg}][\text{m s}^{-2}][\text{m}]^2[\text{s}]^2}{[-][\text{m}][\text{rad}]^2} \quad (\text{D.28})$$

D.4.5 ROTOR AND PROPELLER INERTIA

With the setup shown in Figure D.6a the variables are defined by

$$m = 0.071 \quad [\text{kg}] \quad (\text{D.29})$$

$$g = 9.81 \quad [\text{m s}^{-2}] \quad (\text{D.30})$$

$$r = 0.15 \quad [\text{m}] \quad (\text{D.31})$$

$$h = 1.566 \quad [\text{m}] \quad (\text{D.32})$$

$$\theta_0 \approx 20 \quad [\text{deg}] \quad (\text{D.33})$$

When conducting the experiment with 30 oscillations and filming using a 60 FPS camera with implemented timer, the following data is obtained

	Starting time, [s]	Stop time, [s]	Total time, [s]	Number of oscillations, [-]	Average time for one oscillation, [s]
Experiment 1	8.08	28.30	20.22	30	0.674
Experiment 2	3.38	23.61	20.23	30	0.674

From this data the inertia of the rotor and propeller is estimated to be

$$J = 1.15 \times 10^{-4} \quad [\text{kg m}^2] \quad (\text{D.34})$$

D.4.6 QUADCOPTER INERTIA

The quadcopter inertia is found around the pitch, roll, and yaw axes.

PITCH

The experiment for finding the inertia around the pitch axis was conducted with the variables being

$$m = 2.090 \quad [\text{kg}] \quad (\text{D.35})$$

$$g = 9.81 \quad [\text{m s}^{-2}] \quad (\text{D.36})$$

$$r = 0.17 \quad [\text{m}] \quad (\text{D.37})$$

$$H = 1.96 \quad [\text{m}] \quad (\text{D.38})$$

with a starting angle at around 45 [deg] to 90 [deg] which is greater than for the rotor and propeller experiment due to the quadcopter's oscillation being damped much more than the rotor and propeller experiment.

	Starting time, [s]	Stop time, [s]	Total time, [s]	Number of oscillations, [-]	Average time for one oscillation, [s]
Experiment 1	6.38	28.71	22.33	12	1.861
Experiment 2	4.15	26.41	22.26	12	1.855
Experiment 3	4.15	26.35	22.20	12	1.850
Experiment 4	8.55	31.08	22.53	12	1.877

Which gives a moment of inertia of 0.258 [kg m²].

ROLL

The experiment for finding the inertia around the roll axis was conducted with the variables being equal to the variables for the pitch inertia experiment.

	Starting time, [s]	Stop time, [s]	Total time, [s]	Number of oscillations, [-]	Average time for one oscillation, [s]
Experiment 1	10.97	33.43	22.46	12	1.87
Experiment 2	7.73	29.71	22.98	12	1.832
Experiment 3	8.02	30.43	22.41	12	1.867

Which gives a moment of inertia of $0.0257 \text{ [kg m}^2\text{]}$.

YAW

The experiment for finding the inertia around the yaw axis was conducted with most variables being equal to the variables for the pitch and roll inertia experiment except

$$H = 1.90 \quad [\text{m}] \quad (\text{D.39})$$

	Starting time, [s]	Stop time, [s]	Total time, [s]	Number of oscillations, [-]	Average time for one oscillation, [s]
Experiment 1	7.48	52.16	44.68	20	2.234
Experiment 2	6.30	55.56	49.26	22	2.239
Experiment 3	7.07	57.11	50.04	22	2.274

Which gives a moment of inertia of $0.0387 \text{ [kg m}^2\text{]}$.