



TALLINN UNIVERSITY OF TECHNOLOGY
SCHOOL OF ENGINEERING
Department of Electrical Power Engineering and Mechatronics

**CONSTRUCTION OF VTOL DRONE WITH TANDEM
EDF MOTOR AND DEVELOPMENT OF AN
ALGORITHM FOR AUTOMATIC STABILIZATION
IN HOVER FLIGHT**

**TANDEM VENTILAATORMOOTORIGA VTOL DROONI
KONSTRUEERIMINE JA AUTOMAATSE RIPPLENNUS
STABILISEERIMISE ALGORITMI VÄLJATÖÖTAMINE**

MASTER THESIS

Student: Tõivo Nerep
Student code: 204817MAHM
Supervisor: Kristjan Pütsep, Lecturer

(On the reverse side of title page)

AUTHOR'S DECLARATION

Hereby I declare, that I have written this thesis independently.

No academic degree has been applied for based on this material. All works, major viewpoints and data of the other authors used in this thesis have been referenced.

"18" May 2022

Autor: Tõivo Nerep

/ signature /

Thesis is in accordance with terms and requirements

"18" May 2022

Supervisor: Kristjan Pütsep

/ signature /

Accepted for defence

".....".....2022

Chairman of thesis defence commission:

/ name and signature /

Non-exclusive Licence for Publication and Reproduction of Graduation Thesis¹

I, Tõivo Nerep (date of birth: 21.08.1991)

Hereby

1. grant Tallinn University of Technology (Taltech) a non-exclusive licence for my thesis

"Construction of VTOL Drone With Tandem EDF Motor and Development of an Algorithm for Automatic Stabilization in Hover Flight",

supervised by

Kristjan Pütsep – Lecturer, Tallinn University of Technology

- 1.1 reproduced for the purposes of preservation and electronic publication, incl. To be entered in the digital collection of TalTech library until the expiry of the term of copyright;
 - 1.2 published via the web of TalTech, inc. To be entered in the digital collection of TalTech library until the expiry of the term of copyright;
 - 1.3 I am aware that the author also retains the rights specified in clause 1 of this license.
- 2 I confirm that granting the non-exclusive licene does not infringe third persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from ohter legislation.

¹ *Non-exclusive Licence for Publication and Reproduction of Graduation Thesis is not valid during the validity period of restriction on access, except the university`s right to reproduce the thesis only for preservation purposes.*

_____ (signature)

18. May 2022 (date)

Department of Electrical Power Engineering and Mechatronics

THESIS TASK

Student: Tõivo Nerep, 204817MAHM

Study programme: MAHM, Mechatronics

Supervisor: Lecturer, Kristjan Pütsep

Thesis topic:

(in English) Construction of VTOL Drone With Tandem EDF Motor and
Development of an Algorithm for Automatic Stabilization in
Hover Flight

(In Estonian) Tandem ventilaatormootoriga VTOL drooni konstrueerimine ja
automaatse ripplennus stabiliseerimise algoritmi väljatöötamine

Thesis main objectives:

1. Tail-sitter VTOL drone platform design and construction
2. Test bench design and construction
3. Hover flight automated stabilization algorithm development

Thesis tasks and time schedule:

No.	Task description	Deadline
1.	Gathering and processing information	Sept. 2021 – Oct. 2021
2.	Developing design of UAV and test bench	Oct. 2021 – Jan. 2022
3.	Constructing UAV and test bench	Nov. 2021 – Mar 2022
4.	Developing control algorithm and performing tests	Dec. 2021 – Apr. 2022
5.	Writing Thesis	Nov. 2021 – May 2022

Language: English

Deadline for submission of thesis: „18“ May 2022

Student: Tõivo Nerep „18“ May 2022
/signature/

Supervisor: Kristjan Pütsep „18“ May 2022
/signature/

Head of study programme: Anton Rassõlkin “....”.....2022
/signature/

*Terms of thesis closed defence and/or thesis restricted access conditions to be
formulated on the reverse side*

CONTENTS

PREFACE	7
LIST OF ABBREVIATIONS.....	8
LIST OF FIGURES	10
LIST OF TABLES	12
1. INTRODUCTION.....	13
2. LITERATURE REVIEW	15
2.1 Tandem and single EDF engine	15
2.2 VTOL concept solutions	17
2.3 Aircraft aerodynamics	19
2.4 VTOL control methods.....	22
2.5 Existing solutions	24
2.6 The purpose of the thesis	26
2.7 Conclusion of the chapter	27
3. CONSTRUCTION AND DESIGN PROCESS	29
3.1 Flying platform	29
3.2 Flying platform hardware	32
3.2.1 Actuators	32
3.2.2 Power distribution	35
3.2.3 Control boards	37
3.3 Flying platform construction	39
3.5 Test bench design and construction	42
4. ALGORITHM AND TESTING	46
4.1 Algorithm development	46
4.1.1 Used libraries	47
4.1.2 Global variables and setup	48
4.1.3 RC function	49
4.1.4 Sensor and PID function	52
4.1.5 Servo function and main loop	54
4.2 Testing and PID Tuning	55
4.2.1 Thrust map test.....	55
4.2.2 PID gain tuning	57
4.3 Conclusions	60
SUMMARY.....	62
KOKKUVÕTE	64
LIST OF REFERENCES	66
APPENDICES	69

Appendix 1 Technical drawings of flying platform	70
Appendix 2 Prototype electrical scheme	78
Appendix 3 Technical drawings of test bench	79
Appendix 4 Control algorithm program code.....	83
Appendix 5 Global variable array descriptive table	91

PREFACE

My interest in aviation inspired the idea of this project. The concept was thought first in 2019, which aimed to use a centrifugal compressor instead of a tandem EDF engine as a thrust motor. It soon became apparent that my knowledge of control logic and programming was insufficient to complete the project. That was one of the reasons why I decided to study mechatronics at Tallinn University of Technology. I revived the project by combining my previous experience in aerodynamics and aviation with mechatronics. The idea to use a tandem EDF engine instead of a centrifugal compressor comes from my friend and colleague, who wrote a thesis on such an engine concept at Estonian Aviation Academy. The idea for a test bench is inspired by the gyroscope gimbal principle widely used in aviation.

The whole project is based on self-financing, which forced decisions on materials to be made for financial rather than rational reasons. The project not only taught me a broad understanding of systems but also patience and determination. Every successful development process involves failures. This project's wrong decisions and setbacks will provide a stronger foundation for my future career.

I would like to thank my managers and colleagues in Estonian Police and Border Guard Aviation Group for doing everything to allow me to focus on my studies in recent years. Special thanks to Martin Lihulinn, who gave me consultation regarding the tandem EDF concept and Tambet Viljalo, who helped me with ongoing electronics problems.

I am also very grateful to my supervisor Kristjan Pütsep, Lecturer at Tallinn University of Technology, who supported the whole thesis development process.

Keywords: VTOL, PID control, Tandem EDF, non-destructive testing, UAV, IMU, Embedded system, C++

LIST OF ABBREVIATIONS

µs	microsecond
3D	Three dimension
A	Ampere
ABS	Acrylonitrile Butadiene Styrene
Ah	Ampere hour
BEC	Battery Elimination Circuit
C	Battery C-rating (current which battery is charged or discharged)
C°	degree Celsius
CAD	Computer Aided Design
DC	Direct Current
DoF	Degrees of Freedom
EDF	Electric Ducted Fan
ESC	Engine Speed Controller
EUR	Euro
g	Gram
GNU	General Public Licence (also known as GNU GPL or simply GPL)
Hz	Hertz
I/O	Input/Output
I2C	Inter Integrated Circuit
IMU	Inertial Measurement Unit
KB	Kilobyte
Kg	Kilogram
KV	Constant velocity of a motor (RPM when one volt is applied with no load)
LiPo	Lithium Polymer
LQR	Linear Quadratic Regulator
mA	Milliampere
MHz	Megahertz
MIT	Massachusetts Institute of Technology
mm	Millimetre
Nm	Newton meter
P	Proportional
PD	Proportional Derivative
PETG	Polyethylene Terephthalate Glycol
PID	Proportional Integral Derivative
PLA	Polylactic Acid
PWM	Pulse Width Modulation

Rad/s	Radians per second
RC	Remote Control
RPM	Rotations per minute
UAV	Unmanned Aerial Vehicle
V	Volt
VTOL	Vertical Take-off and Landing
W	Watt

LIST OF FIGURES

Figure 1.1 VTOL Lockheed XFV -1 aircraft [1].....	13
Figure 2.1 The first part of the test apparatus used to analyze engine parameters [2]	16
Figure 2.2 Design example of Tail-sitter; 'Puffin' [4]	18
Figure 2.3 Design example of Tiltrotor; 'Bell V-22' [4].....	18
Figure 2.4 Design example of Tiltwing aircraft; "GL-10" [4].....	19
Figure 2.5 Force vector directions, acting on the object	20
Figure 2.6 Moving airflow toward the airfoil; Relative airflow is shown as blue lines and the direction is from left to right; Airfoil is shown in grey [7]	21
Figure 2.7 Force vectors are shown for fixed-, and rotary-wing aircraft. The left image shows the aircraft force vectors of the rotary-wing in forward flight; the right part shows the fixed-wing aircraft that is tilted [6] [7]	21
Figure 2.8 Typical flight modes of the tail-sitter VTOL UAV: 1 – 2 represent the transition from hover to forward flight, 3 - 4 represent the transition from forward flight to hover [12].....	22
Figure 2.9 Illustration of tandem EDF VTOL concept [17].....	25
Figure 2.10 Illustration of single EDF VTOL concept used in [18], [19], and [20]	25
Figure 2.11 Illustration of double tandem EDF VTOL concept with state input variables [21]	26
Figure 3.1 Platform 3D model (with cut view of engine section) and its coordinate system	29
Figure 3.2 Tandem EDF motor assembly.....	30
Figure 3.3 Control frame assembly exploded view.....	31
Figure 3.4 Battery compartment assembly.....	32
Figure 3.5 Servo actuator dimensions; Cyan – standard design servo actuator; Red – slim design servo actuator	34
Figure 3.6 Power distribution scheme	35
Figure 3.7 Flying platform assembly	40
Figure 3.8 Special hinge bolt drawing with dimensions	41
Figure 3.9 Hardware connection block scheme	41
Figure 3.10 Test bench rotating frame axis placement.....	43
Figure 3.11 Proposed gimbal type frame with measurements, and arc arrows showing rotating joints.....	43
Figure 3.12 Raw material profile; $H1 = H2 = 35\text{ mm}$ and $B = 66\text{ mm}$ [30]	44
Figure 3.13 Inner frame to outer frame attachment	44
Figure 3.14 Inner frame and drone attachment adjustability	45

Figure 3.15 Flying platform attached to test bench	45
Figure 4.1 Algorithm flowchart	46
Figure 4.2 Unfiltered PWM signal with pulse length of 1000 μ s	50
Figure 4.3 RC signal input response with different filter constant values: blue - unfiltered signal, green – 70 % filtered signal, red – 99 % filtered signal	51
Figure 4.4 Program code in Arduino IDE that returns correction signal for pitch actuator	54
Figure 4.5 Throttle mapping test equipment.....	56
Figure 4.6 Throttle map graph: dashed line (blue) – trendline, black line - actual test result	56
Figure 4.7 PID controller response for too high proportional gain [13 p. 152]	57
Figure 4.8 PID controller response for too small proportional gain [13 p. 152].....	57
Figure 4.9 PID controller response for too small integral gain [13 p. 152]	58
Figure 4.10 PID controller response for too large integral time [13 p. 153]	58
Figure 4.11 PID controller response for too large derivative time [13 p. 153]	58
Figure 4.12 Roll PID controller disturbance rejection graph; black line – actual position measured by inertial sensor; red line – setpoint.....	60

LIST OF TABLES

Table 2.1 Disclaimers and acknowledgements of references	27
Table 3.1 EDF motor specifications [22]	33
Table 3.2 ESC specification comparison [22]	34
Table 3.3 ESC and EDF compatibility	34
Table 3.4 Servo models characteristics [22]	35
Table 3.5 Battery and EDF characteristics [22]	36
Table 3.6 DSWY2596 DC to DC voltage converter characteristics [23]	37
Table 3.7 Arduino Nano specifications [24]	38
Table 3.8 Adafruit BNO055 specifications [26]	38
Table 3.9 Selection of available printing parameters in Ultimaker Cura 4.12.1	39
Table 3.10 Filament comparison	40
Table 3.11 3D printed parts and printing time	40
Table 4.1 Arduino variable types [37]	48
Table 4.2 PWM signal different reading method comparison [38]	49
Table 4.3 Throttle mapping test results	56

1. INTRODUCTION

In general, aerial vehicles can be divided into fixed- and rotary-wing. These both have their advantages and disadvantages. The operation of fixed-wing aircraft depends on the availability of a sufficient runway. Therefore, to operate a fixed-wing drone, site selection is critical. On the other hand, the flight range of such a concept is significantly longer than rotary-wing aircraft can provide. Rotary wing aircraft can take off vertically and hover, but they have higher power consumption. Due to both disadvantages, scientists have been exploring the possibilities of figuring out a hybrid version that can take off vertically, hover, and transit into a more economical forward flight. Such a design is called vertical take-off and landing (VTOL) aircraft. The first known VTOL aircraft made its first flight in 1954 (Figure 1.1). Only one unit was produced, and the project was cancelled in 1955. The aircraft could fly, but since the pilot had poor visibility during vertical take-off and landing, it was considered too dangerous. Further VTOL aircraft were primarily developed with a design where not the aircraft itself was tilted to achieve translational flight, but instead, engines and wings were tilted, or engine airflow deflected downwards. These solutions for control logic and moving mechanical systems have been complex and challenging. [1]



Figure 1.1 VTOL Lockheed XFV -1 aircraft [1]

Due to the rapid development of technology, unmanned aerial vehicles have found their way into commercial use. They are useful in different fields, including firefighting, transportation and surveillance. With the growing use of unmanned aerial vehicles, there is also a need for more sophisticated and capable machines. Thanks to availability of advanced technology, such automated aircraft can now be developed. This could lead to a safe flight control solution for the initial design of the VTOL aircraft. The primary benefit is the simplicity of mechanical design, which improves durability and helps lower maintenance and production costs. A novel unmanned aerial vehicle can be designed by integrating modern technological advantages. Many pieces of research have already been conducted on the topic, but much remains to be discovered and developed.

This paper seeks a solution for the tail-sitter VTOL automated hovering algorithm. The problem is that the existing solutions are multifunctional, trying to solve control logic for many different concepts. Even if some solutions are dedicated to a specific model, then these solutions are highly theoretical and validated only by simulation. This leads to a loss in the performance and speed of the controller. The aim is to ensure one goal-focused robust, simple, and failproof control logic for tail-sitter VTOL automatic hovering phase. A VTOL drone with a tandem EDF motor will be constructed during the process to test the algorithm solution. In addition, one test bench with three degrees of freedom (3 DoF) for non-destructive testing and another fixture for thrust force mapping are developed.

The second chapter focuses on existing solutions and a literature review to give overview of subject. The third chapter will describe the design and construction process of the drone and test bench. The final chapter describes algorithm development and test procedures.

2. LITERATURE REVIEW

2.1 Tandem and single EDF engine

EDF engines are known as powerful but with relatively high energy consumption motors. Integrating two counter-rotating EDF motors into one tandem engine will provide a powerful engine with specific benefits. These potential benefits were studied in the final thesis of the Estonian Aviation Academy [2]. In the research, author describes the theory behind EDF engines and the principles of counter-rotating propellers. The components of the EDF engine are a DC motor, ESC, ventilator, and housing. These are described in detail, together with design benefits and flaws.

Shortcomings of EDF engine:

- Vortex footprint generated by the high rotational speed
- Angular momentum generated by electrical motor
- High energy consumption
- The profile of the surrounding housing generates parasite drag.
- Profile of surrounding housing adds weight
- Lower efficiency at low speeds

Advantages of EDF engine:

- Smaller dimensions
- Higher thrust
- Higher ventilator efficiency factor at higher flight speeds
- Higher static thrust at smaller dimensions of the propeller
- Lower noise level
- Safer operation due to lack of exposed propellers
- Good prerequisites to use in VTOL applications
- Higher exhaust air flow speed

These flaws and benefits are presented in comparison to the conventional propeller. Following research [2], many previously stated shortcomings could be reduced by using two-stage counter-rotating fans. Using this solution, some additional benefits are as follows:

- Higher power rating without increasing the ventilator diameter
- Possibility to eliminate vortex footprint
- Increase in efficiency up to 16 %
- Possibility to eliminate angular momentum

To validate theoretical aspects, author constructed Tandem- and single-fan EDF engines. Thrust, energy consumption, noise, and vortex footprint were investigated during the experiments. The author constructed a fixture that allowed direct thrust into scale placed below to measure static thrust. Energy consumption was measured with an ammeter, and the noise was measured with a noise level meter. All the tests were carried out for single EDF and tandem EDF. The test apparatus is shown in Figure 2.1.



Figure 2.1 The first part of the test apparatus used to analyze engine parameters [2]

Separate tests were carried out to obtain characteristics of the vortex footprint. To visualize the results, a steam generator was used. During tests, airflow speed measurements were also taken. As a conclusion of both experiments, the author discovered that the tandem EDF solution produces up to 84 % of the thrust that two separate EDF engines could produce. The most significant advantage was that the tandem engine has up to 1.53 times higher exiting airflow speed in conjunction with reduced rotational momentum [2].

2.2 VTOL concept solutions

Guillaume J.J. Ducard and Mike Allenspach have written a paper that analyses different VTOL concepts [3]. The paper aims to help other researchers to develop control systems for VTOL aircraft. In the first part of the paper, the authors give a historical overview of manned and unmanned VTOL configurations. An overview is divided into two tables, where models are listed together with a short description. In the first table, the selection of manned VTOL projects are described. In the second table, similar descriptions of unmanned projects are shown. Short conclusions of these projects are provided as follows:

- The most well-known manned VTOL designs have been developed for military purposes. Many projects did not meet expectations, and successful flights were limited. One of the manned VTOL projects that the authors describe was tail sitter. All other projects were more complex designs with tiltable engines, rotors, wings, or adjustable airflow direction. Due to complexity, most of these projects were cancelled. The only projects that are still flying are F35-B SVTOL, Bell-Boeing V-22, and Hawker P1127/Harrier. F35 and Harrier use a similar concept of deflecting airflow under the aircraft to obtain hover flight. Bell-Boeing uses tilting rotors to transit from one flight phase to another. These aircraft are relatively expensive to produce and maintain under strict aviation standards.
- The most-known unmanned VTOL designs have been more successful, but only a few have been completed. The tables provided by the authors clearly show that the projects are confidential and that there is a lack of information available. The design of most projects is complex, leading to costly maintenance. Although the shortcoming of tail-sitter VTOL, regarding pilot comfort and visibility, is eliminated (unmanned aircraft pilot remains on the ground), most of the reviewed projects are developed under different design concepts.

During further research, authors provide descriptions and comparisons of different VTOL concepts. In [4], authors describe a similar view of conceptual division. Unmanned VTOL aircraft are divided into three categories: Tail-sitter, tiltrotor, and tiltwing. A short description of each concept is provided in the following list:

- Tail-sitter (Figure 2.2) concept contributes to simple mechanical design. Rotors are attached rigidly to the fuselage, and to obtain hover flight, the whole aircraft is tilted. This design is more complex to control in comparison to the others. Due to the tilt of the whole aircraft, a large wing area is exposed to winds during the hover phase. This will generate significant disturbances that actuators try to compensate for, leading to power loss due to the stabilization process. [3], [4]

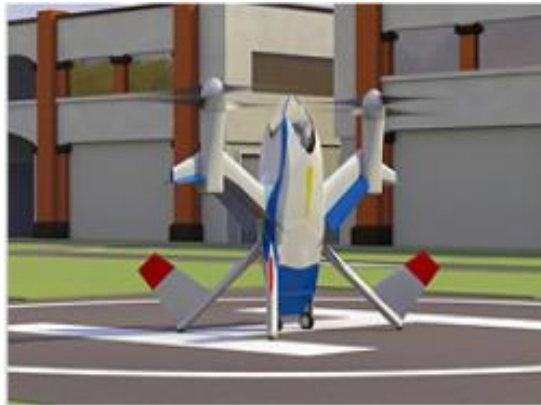


Figure 2.2 Design example of Tail-sitter; 'Puffin' [4]

- The tiltrotor aircraft (Figure 2.3) will maintain fuselage orientation regardless of the flight phase. This is done by tilting only the rotors to obtain hover flight. The major disadvantage is that part of the wing will be in propeller downwash during hover flight, which reduces potential propeller capability. Another shortcoming is associated with the design of the wings. As heavy tiltable rotors are mounted to the wings, producing wings shorter and thicker is necessary. This leads to reduced cruise flight efficiency. Additionally, this design has increased the mechanical complexity and deadweight of actuators. [3], [4]



Figure 2.3 Design example of Tiltrotor; 'Bell V-22' [4]

- Tiltwing aircraft (Figure 2.4) will maintain its fuselage similarly to the tiltrotor concept, but some of the shortcomings are eliminated. The rotors are rigidly attached to the wing, and the entire wing is tilted. This allows it to use potential propeller capability to a greater extent. Another benefit gained is that flight controls will be rotated together with wings, making it possible to use them in hover flight. This concept has a complex wing-tilting mechanism and deadweight associated with it. [3][4]



Figure 2.4 Design example of Tiltwing aircraft; "GL-10" [4]

2.3 Aircraft aerodynamics

Before understanding the aerodynamics of hybrid aircraft, it is essential to understand conventional aerial vehicles. Therefore, relevant topic reviews are provided, associated with fixed- and rotary-wing aircraft. The aerodynamics of traditional fixed-wing aircraft is described in [5] and [6]. The same topics for conventional rotary-wing aircraft is provided in [7], [8], and [9].

Aerodynamics is the branch of mechanics that deals with the motion of air and other gases and the effects of such motion on objects in the medium [10]. Regarding the aerodynamics of the object, it is affected by two main forces: lift and drag. The lift acts perpendicular to the relative wind and opposes another force called weight. Drag is parallel to the relative wind and opposes the force called thrust. This adds another two forces necessary to understand the body in flight. All these four forces are shown in Figure 2.5 and described as follows:

- The lift is the component that forces the object to become airborne. It is produced by airflow moving across an airfoil, which is designed to generate a low-pressure area above the wing and a higher pressure area below the wing. The pressure difference is the reason behind the lift. The lift direction is not always perpendicular to the weight, as shown in Figure 2.5. The direction depends on the relative wind.
- Weight is a load component that pulls the object downward due to the force of gravity. Therefore, the lift must overcome weight to allow the object to become airborne. This is the only vector that will sustain its direction regardless of relative wind direction
- Thrust is the force produced by the rotor opposing the force of drag
- Drag is the force that acts against movements caused by airflow disruption. This vector opposes thrust and is parallel to the relative wind. This component can be divided into induced drag and parasitic drag.

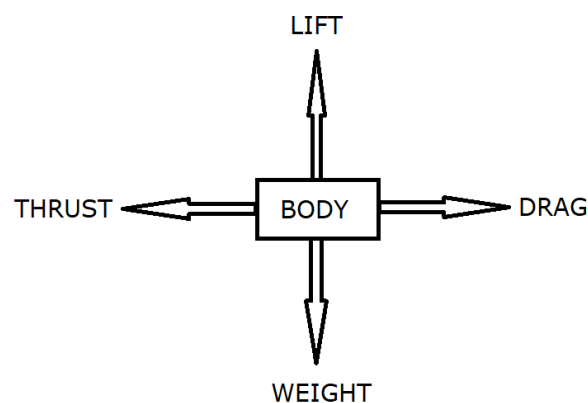


Figure 2.5 Force vector directions, acting on the object

Newton's three laws of motion and Bernoulli's principle can explain the flight phenomenon. Bernoulli's principle states that if the velocity of moving fluid increases, the pressure of fluid decreases. By applying the same principle to an airfoil (Figure 2.6) a shape similar to a diffuser is achieved depending on the airfoil angle. Bernoulli's principle states that the increased airflow speed above the airfoil results in a decrease in pressure.

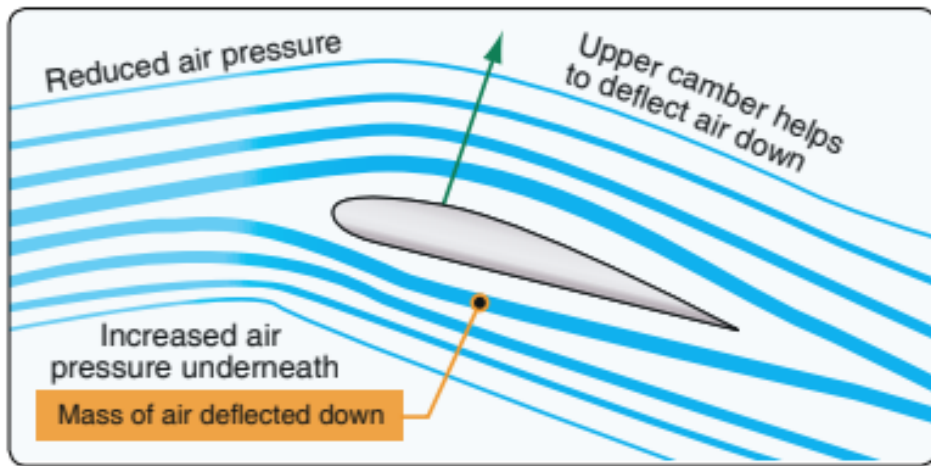


Figure 2.6 Moving airflow toward the airfoil; Relative airflow is shown as blue lines and the direction is from left to right; Airfoil is shown in grey [7]

All that affects both fixed- and rotary-wing aircraft. Figure 2.7 shows the associated vectors for fixed and rotary-wing aircraft in a specific flight regime. The difference is in how a sufficient relative wind is produced. This difference proves why aeroplanes are not recognized as aircraft with hovering capability, but helicopters are. Fixed-wing aircraft move through the air with thrust provided by a powerplant. The movement is axial, and lift will be produced when sufficient speed is achieved. This principle is also valid for helicopters, where an airfoil will produce lift if the relative wind and angle of attack are sufficient. The difference being that the aircraft's fuselage remains stationary, whereas the airfoil rotates through the air to generate lift. This enables helicopters hovering and performing manoeuvres that fixed-wing aircraft are not capable of.

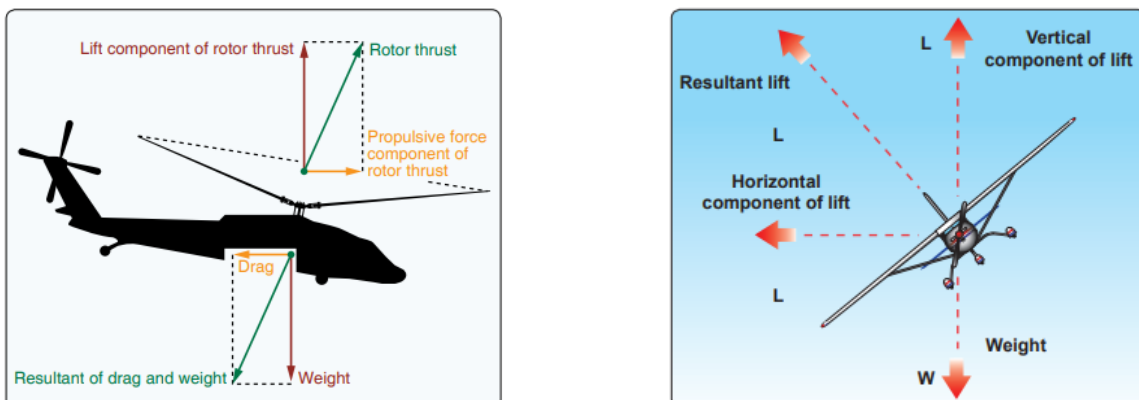


Figure 2.7 Force vectors are shown for fixed-, and rotary-wing aircraft. The left image shows the aircraft force vectors of the rotary-wing in forward flight; the right part shows the fixed-wing aircraft that is tilted [6] [7]

This was a simplified summary of the aerodynamics that affect helicopters and aeroplanes. The [11] and [8] repeatedly highlight that helicopter aerodynamics is too complex to rely only on calculations. Therefore, it is reasonable to perform multiple physical experiments to obtain results. In [7], [11] and [8], more detailed information about aerodynamics affecting rotary-wing aircraft is provided. In [5] and [6] more detailed description for fixed-wing aircraft aerodynamic effects is described.

2.4 VTOL control methods

In [3], possible control methods are described for tail-sitter, tiltrotor, and tiltwing. The authors found that automatic control of hybrid UAVs is an ongoing challenge. The reason is that the VTOL dynamics are highly non-linear. The most challenging is the transition manoeuvre from hover mode to forward flight mode and vice versa. These transitions for the tail-sitter VTOL drone are shown in Figure 2.8. During this literature review, only tail-sitter concept control methods are being reviewed.

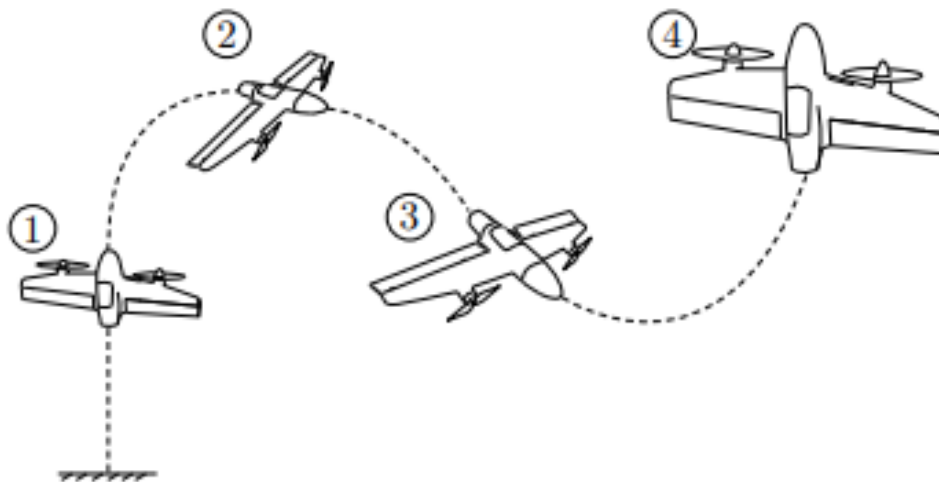


Figure 2.8 Typical flight modes of the tail-sitter VTOL UAV: 1 - 2 represent the transition from hover to forward flight, 3 - 4 represent the transition from forward flight to hover [12]

The authors divide tail-sitter control methods into two categories based on approaches. The focus is only on methodologies that can be used across the entire flight envelope. Methods focusing exclusively on hover phase control are excluded. Several solutions are identified in both categories:

- Scheduled Control Approach
 - Divide and Conquer with P/PD/PID
 - Divide and Conquer with LQR
 - Divide and Conquer with Backstepping
- Unified Control Approach
 - Robust Control
 - Direct Gain Scheduling with P/PD/PID
 - Dynamic Inversion with P/PD/PID
 - Dynamic Inversion with LQR
 - Dynamic Inversion with Backstepping

Scheduled control approaches are identified as a promising concept for VTOL aircraft. This policy linearizes the system around a finite set of trim-points and solves each set with the dedicated independent controller. The most used approach is Divide and Conquer together with a well-tuned controller. Flight phases are divided into categories, and each is solved as an individual problem. This approach switches discretely between phases, but only one of the controllers is running at the time. Regarding controller types, the most used is stated to be P/PD/PID control. Such a controller is relatively easy to configure, and it requires only limited knowledge of the system. The general formula of independent PID is following [13]:

$$u_{PID}(t) = K_p e(t) + K_I \int_0^t e(t) dt + K_D \frac{de(t)}{dt} \quad (2.1)$$

Where: K_p – proportional gain

K_I – integral gain

K_D – derivative gain

$e(t)$ – error vector

Difference between desired position and actual position is error, and the equation output can be tuned by configuring three gains. Proportional gain specifies how quickly the error will be eliminated linearly, integral gain will eliminate steady-state error, and derivative gain is added to provide error corrections depending on the rate of change of the error. [3][12][13]

Another frequently used scheduled control law is LQR (Linear Quadratic Regulator). The controller optimizes the control of the linear system. The authors point out that this controller has good robustness properties, but modelling errors can drastically affect results. This means that the complex dynamic model of a hybrid UAV must be

calculated precisely [3]. A more detailed comparison of the tail-sitter's PID and LRQ control is provided in [12]. Overall, since the number of linear points during the entire flight envelope is finite, each control structure is limited to a specific envelope area. This may lead to unstable control at transition points of the flight.

Unified control approaches offer a solution that provides one controller, instead of multiple, for the entire flight envelope. This solution is applied to overcome changing aerodynamic effects. In [14], the authors review the Robust Control method and state that successful control has been achieved over the entire flight envelope. This method does not react to uncertainties but creates robustness against them. The shortcoming is that such controllers often need additional hardware to achieve satisfactory results [3]. A more advanced solution is the dynamic inversion method. This transforms the nonlinear system so that the controller can act on it linearly. The solution has shown promising results but is highly dependent on the precision of the model [3]. On the other hand, in [15], a dynamic inverse with neural network control is proposed. The authors mention that such a solution is usually model free. The mathematical model was constructed only to make the simulations more convincing and to validate the essential functions of the controller [15]. Another project that covers the neural network controller is [16].

2.5 Existing solutions

Numerous researches have been carried out on tail-sitter VTOL aircraft. The [12] is a paper comparing model-based and model-free control approaches. In addition to simulation flight and analysis, the paper describes aerodynamic preliminaries and theoretical control background. The authors have found that a model-free controller has better rejection of disturbances than a model-based controller.

The authors of [17] propose the concept of UAV (Figure 2.9) to assist in automated maintenance procedures for infrastructure systems. This concept presents a double propeller ducted fan tail-sitter VTOL which is highly similar to a concept under research in the current project. The authors describe a detailed dynamic model and propose a control solution which is then validated with the aid of simulation. Testing in a physical environment is not included in [17]. Authors point out that counterrotating propellers cancel out reaction pairs generated by the gyroscopic precession torque effect. Further, this solution allows designing controls so that yaw dynamics are solved

purely by using ducted fans. The [18], [19] and [20] cover similar projects (Figure 2.10), where a solution is sought for the VTOL concept powered by a ducted fan. The difference is that in these studies, the architecture of UAVs contributes only to a single propeller design. This leads to higher actuation expectations for other flight controls as they need to overcome rotational momentum generated by a single thrust propeller.

In [21], authors have constructed a novel tandem ducted fan vehicle. This solution is not intended to be VTOL aircraft but uses two thrust engines with counter-rotating and axially positioned propellers. This concept is shown in Figure 2.11, where design is presented together with state input variables. This research includes nonlinear modelling, attitude control system design, and simulations. In conclusion, the authors stated that additional research regarding speed and position control must be carried out together with various flight experiments [21].

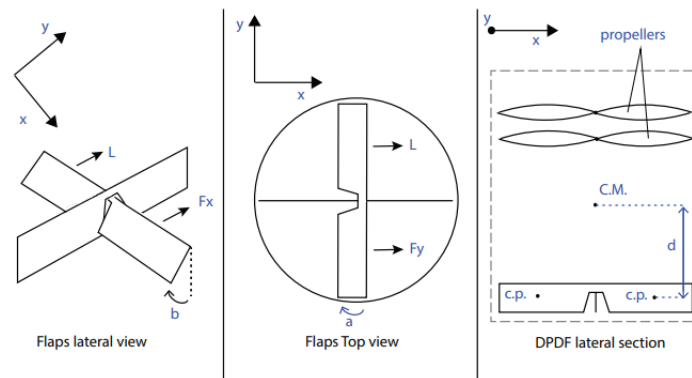


Figure 2.9 Illustration of tandem EDF VTOL concept [17]

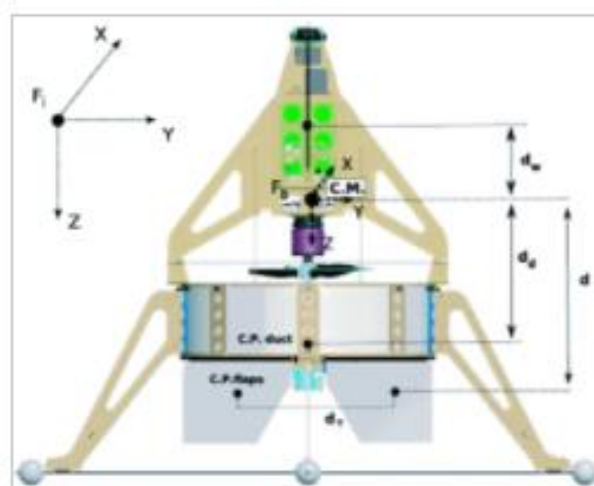


Figure 2.10 Illustration of single EDF VTOL concept used in [18], [19], and [20]

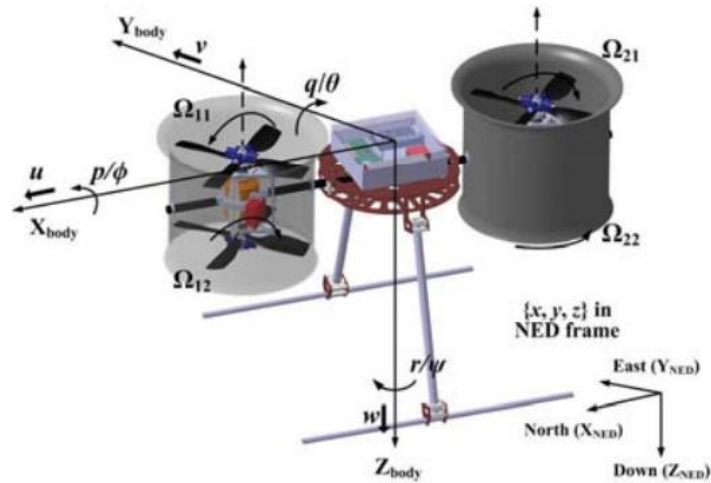


Figure 2.11 Illustration of double tandem EDF VTOL concept with state input variables [21]

2.6 The purpose of the thesis

Problem statement

- The Solution is intended to provide an automatic hovering attitude stabilization algorithm for the tandem EDF-powered VTOL UAV concept. The solution is an algorithm that controls two counter-rotating fans and two flight control surfaces simultaneously. To perform corrective actions, an Inertial Measurement Unit (IMU) is used to detect the current position of the aircraft. IMU will send position information to the microprocessor that compares new data with setpoints. If an error is detected, corrective action will be sent to the corresponding actuator.
- What should be the solution to provide automatically stabilized hover flight for tandem EDF-powered VTOL UAV?
- During development, the aim is to create an algorithm system that controls the attitude of the constructed UAV concept in the hover flight phase.
- The solution may be of interest to companies that are implementing ways to take advantage of hybrid UAVs. Such concept could be implemented into different fields: military, transportation, surveillance etc.
- The result of the thesis is an attitude controlling algorithm on the functional VTOL UAV.

2.7 Conclusion of the chapter

The literature analysis revealed numerous studies focusing on VTOL aircraft. Among others, there are several tail-sitters with ducted fan solutions, but only one research was found using the same principle of concept. The conclusion of reviewed literature is provided in Table 2.1.

Table 2.1 Disclaimers and acknowledgements of references

Research paper reference	Disclaimer(s)	Acknowledgement(s)
[2]- Tandem EDF engine	Only covers engine characteristics Not implemented on physical UAV	Benefits and flaws of Tandem EDF systems Design characteristics of Tandem EDF systems
[3], [4] – review of VTOL systems	Not dedicated to any tests nor simulations Only analytical research	Description of most VTOL concepts and most existing VTOL control methods
[5], [6], [7], [8], [9] [11] – Aerodynamics of conventional aircraft	N/A	Principles of aerodynamics
[10] - Dictionary	N/A	Scientific Definition for aerodynamics
[12] – Model free or model based design	Results are validated only in simulation	comparison of LRQ and PID controller dedicated to tail-sitter Model free design can give better results
[14] -control of tiltrotor UAV	Needs additional hardware to achieve its maximum performance	Robust control can achieve satisfactory results over full flight envelope
[15], [16] – Neural Network control	The VTOL concept contributes only from one propeller	Successful control methods based on neural network Results validated in simulation and real environment
[17] -	Not VTOL Results are validated only in simulation	Possible control method for tandem EDF engine solution Dynamic model of axially placed tandem EDF engine
[18],[19],[20]	The VTOL concept contributes only from one propeller Results are validated only in simulation	Design and model of similar VTOL concept
[21]	Not VTOL EDF engines are not placed axially (concept is different) Results are validated only in simulation	Possible control method for tandem EDF fan solution Characteristics of tandem EDF engine

This review of existing research revealed that the topic needs further development and analytical data from tests carried out in a physical environment. Most existing papers use simplified dynamic models and validate solutions only in simulation. The problem is that dynamic model for such a concept is highly complex to calculate precisely. On

the other hand, there are already analysed dynamical models and promising results validated in simulators. Therefore, this paper concentrates on:

- constructing an unmanned VTOL tail-sitter
- constructing a test bench for non-destructive testing
- proposing a control solution with the dedicated algorithm on an embedded system platform.
- Validating solutions in realistic environment (avoiding dynamic model calculations and simulations)

3. CONSTRUCTION AND DESIGN PROCESS

3.1 Flying platform

The design concept is first compiled into a 3D model for verification. This model can also be used directly for additive manufacturing process. Software used for this purpose is Solid Edge 2021 Student edition. During this paper, basic tail-sitter is constructed. As only an algorithm for stable hover flight will be developed, the design of wings is not necessary. Fuselage components are manufactured using 3D printing technology. Modifications to purchased parts are made using basic tools. Figure 3.1 shows an overall model with coordinate system. Dynamics regarding coordinates are as follows:

- Rotation around y – roll axis
- Rotation around x – pitch axis
- Rotation around z – yaw axis

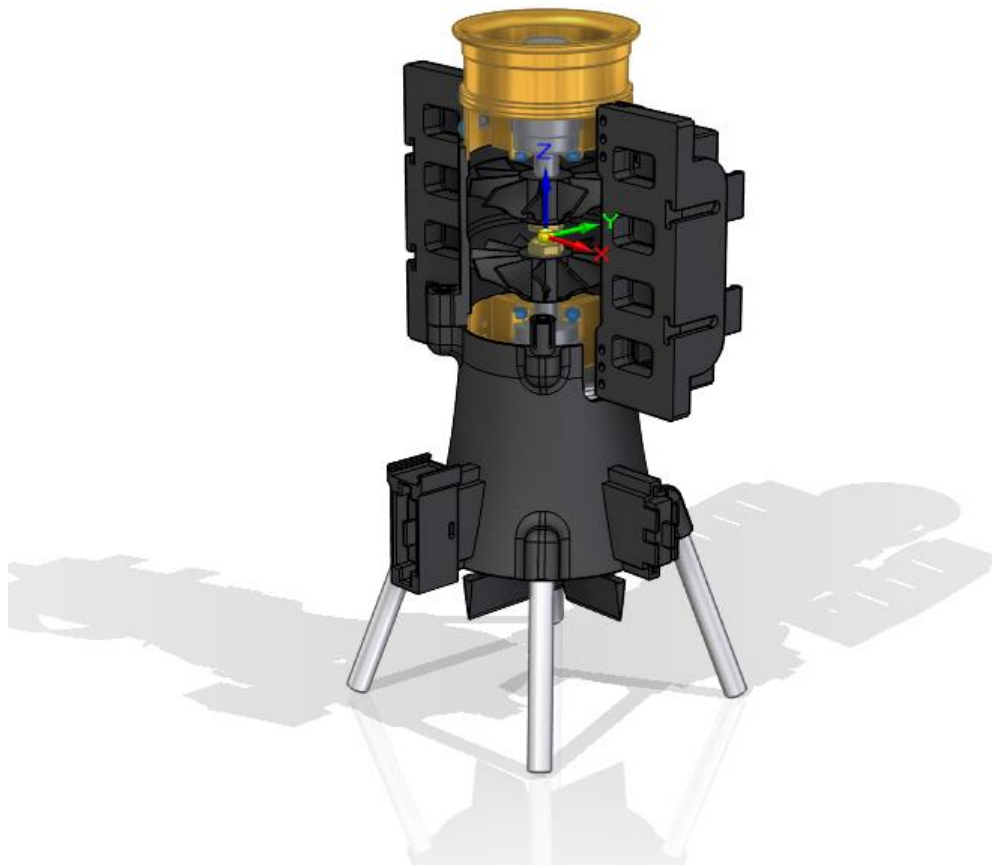


Figure 3.1 Platform 3D model (with cut view of engine section) and its coordinate system

The process started by choosing suitable EDF motors that set initial conditions for further design. Motors were disassembled, and each part was measured to model using Solid Edge software. This created a base for modelling 3D assembly from which unnecessary motor details could be removed and modified on an ongoing basis as needed. Further, housing (Figure 3.2 item number 7) was designed to fix two motors together, completing a tandem EDF motor assembly. As seen in Figure 3.2, this assembly consists of 9 different items. Most of them are original parts from Dr. Mad Thrust 70 mm EDF motor. Completed thrust motor is counter-rotating tandem EDF motor. This assembly is responsible for generating lift force and controlling yaw angle.

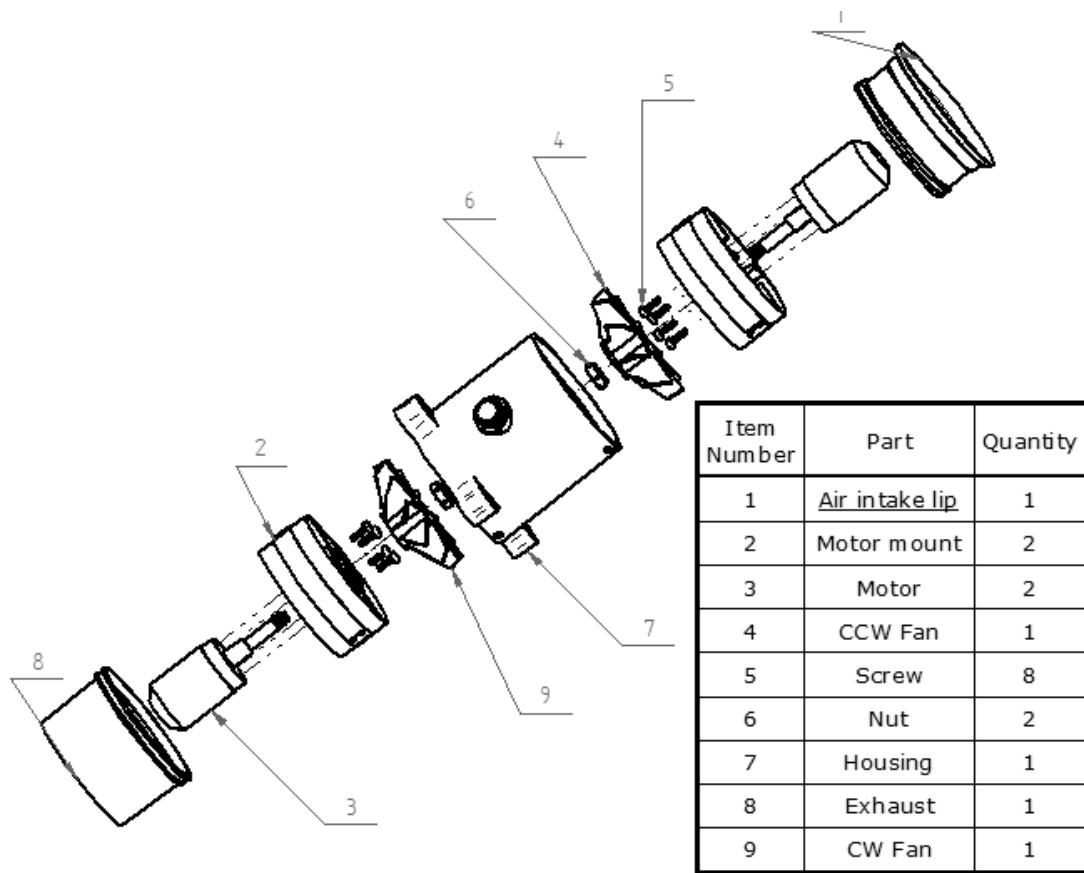


Figure 3.2 Tandem EDF motor assembly

Next, roll and pitch control assembly design was modelled (Figure 3.3). This assembly consists of housing (Figure 3.3 Item number 1), servo actuators (Figure 3.3 Item number 2), control surfaces (Figure 3.3 Item number 3 and 4), landing gear (Figure 3.3 Item number 5) and special hinge bolts (Figure 3.3 Item number 6). Servo actuators are responsible for moving control surfaces, which in turn are responsible for controlling roll and pitch angles. Housing mounts all these parts together and connects control frame assembly to tandem EDF motor assembly. It also supports aluminium tubes acting as a landing gear.

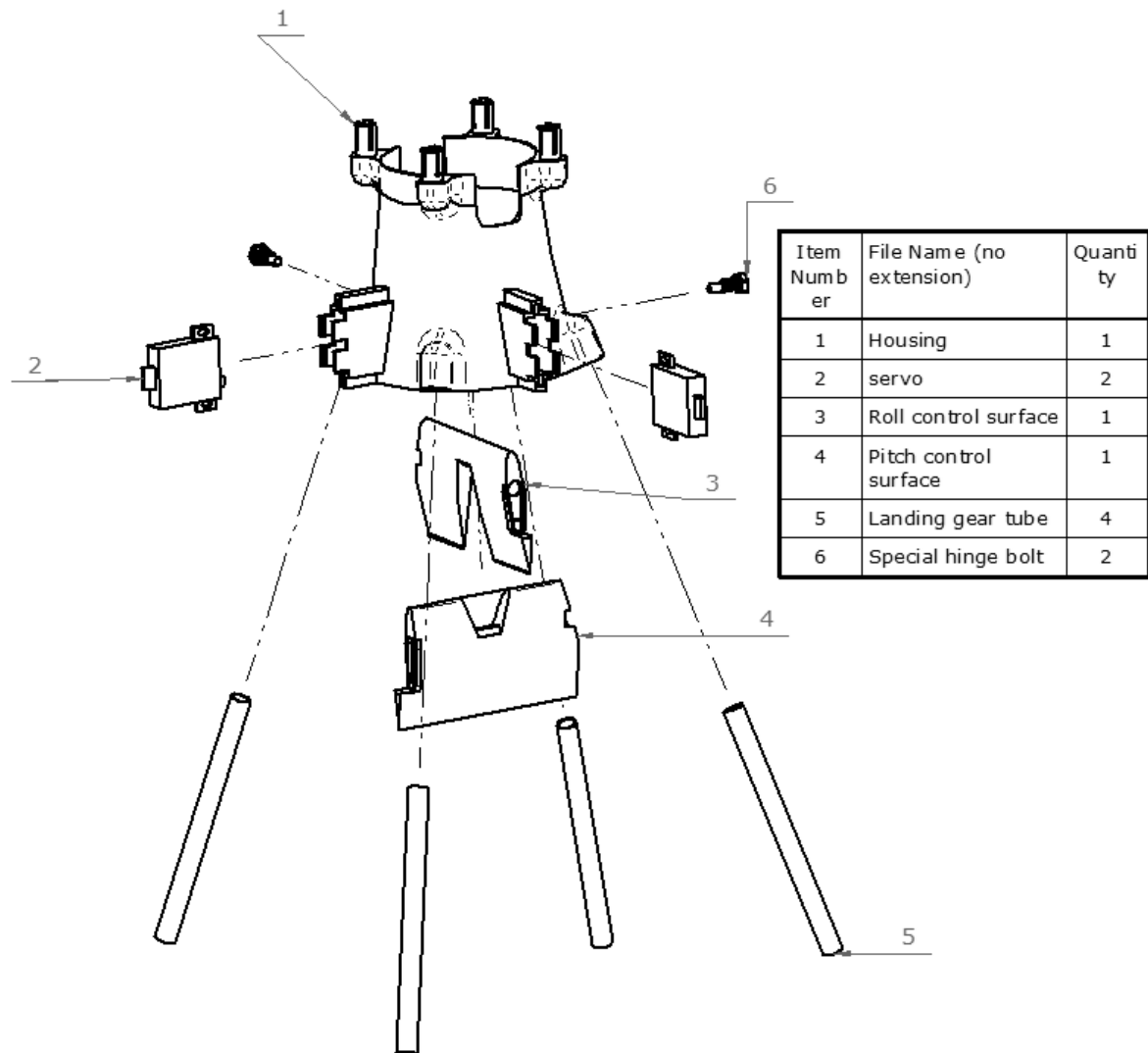


Figure 3.3 Control frame assembly exploded view

Final subassembly of designed fuselage is battery compartment assembly (Figure 3.4). This assembly consists of reinforcement (Figure 3.4 item number 1), battery compartment (Figure 3.4 item number 2), housing attachments (Figure 3.4 item number 3) and fasteners (Figure 3.4 item number 4, 5 and 6). The fuselage assembly consists of two battery compartments. Battery dimensions were main design criteria for this subassembly as it needs to fit battery inside the compartment. It also acts as a test bench attachment link through reinforcement (Figure 3.4 item number 1). Housing attachments (Figure 3.4 item number 3) fix battery compartment assembly and tandem EDF motor assembly parts together.

Item Number	File Name (no extension)	Quantity
1	Reinforcement	1
2	Battery compartment	1
3	Housing attachment	2
4	Screw	4
5	Washer	4
6	Nut	4

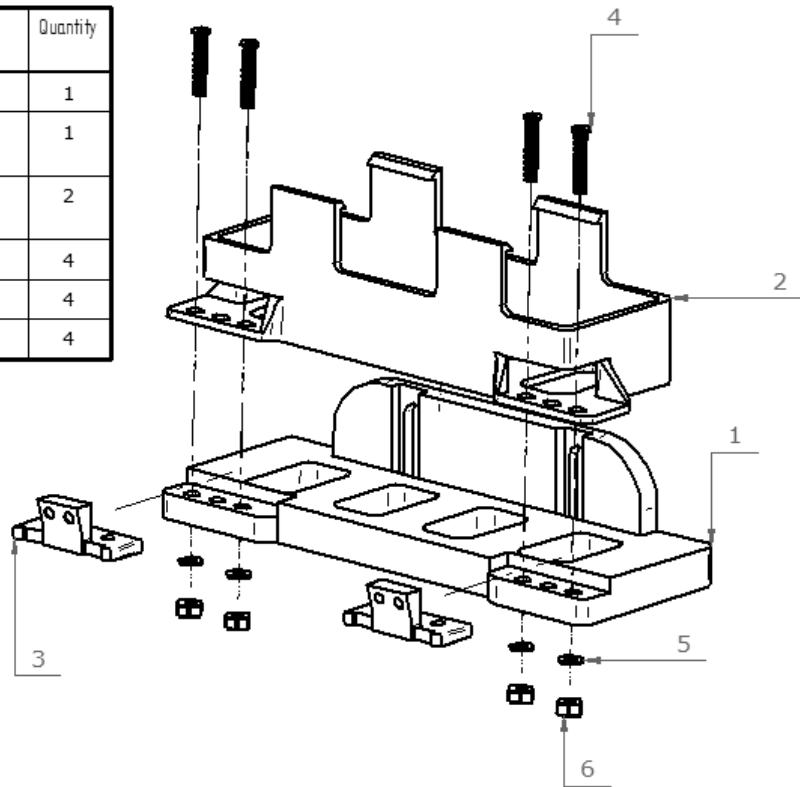


Figure 3.4 Battery compartment assembly

3.2 Flying platform hardware

Importance of choosing components is vital because without considering the characteristics, risk remains that the designed platform will not produce sufficient lift. Another factor is that components must work as a set. Therefore, component selection was performed prior to fuselage design process (paragrah 3.3). Further in this paragraph, hardware characteristics and component selection reasoning are described.

3.2.1 Actuators

Thrust motors are a major part of aircraft. This is the first decision that sets criteria for other components. Dr. Mad Thrust 70 mm 10 blade Alloy EDF 3000 KV motors were chosen. These motors have good quality and high performance. As EDF motors are pre-assembled, they are also dynamically balanced by manufacturer. Aluminium alloy housing acts as heat dissipator and provides rigidity that motors with plastic housing do not have. Specifications are shown in Table 3.1. Dimensions set the fuselage design criteria. Motor type defines maximum speed, and that in turn defines

maximum voltage. Simplified relation between them is shown in Equation 3.1. Actual rotational speed will also depend on the batteries' charge level (fully charged 4s LiPo battery voltage is 16,8 V) and load on the motor. Therefore, values in Table 3.1 are not exactly corresponding to values resulting in calculation. Considering motor type, maximum speed, maximum voltage and power ratings, batteries, and electrical speed controllers (ESC) could be chosen.

$$\omega = Kv * V_{max} \quad (3.1)$$

Where: ω – Maximum motor speed [RPM]

Kv – Motor type [Kv]

V_{max} – Maximum voltage [V]

Table 3.1 EDF motor specifications [22]

No.	Parameter	Value
1.	EDF inner diameter [mm]	69,00
2.	EDF outer diameter [mm]	72,00
3.	Number of blades	10,00
4.	Maximum speed [RPM]	48000,00
5.	Motor Type [KV]	3000,00
6.	Continuous Power [W]	1200,00
7.	Maximum Voltage [V]	14,80
8.	Maximum Current [A]	83,00
9.	Maximum Thrust [g]	2300,00
10.	Weight [g]	256,00

ESC decision is important to get best possible motor performance. Main hardware components were ordered from same retailer to simplify logistics. The decision was made between two similar products. Specifications of both are provided in Table 3.2 and it shows that Tunigy Plush has smaller dimensions and weighs less than HobbyKing ESC. Downside is that most programming cannot be done without programming card. BEC (battery elimination circuit) characteristics are not considered relevant during decision making, as solution has its own voltage regulator and power distribution circuit. Therefore, due to simplified programming and availability, chosen product is HobbyKing speed controller. Additionally, it has overvoltage, overheating, and abnormal input protections, which contribute to more reliable operation. To finally validate the decision, compatibility of EDF motor and ESC parameters is shown in Table 3.3, which reveals that ESC parameters meet conditions set by EDF motors.

Table 3.2 ESC specification comparison [22]

No.	HobbyKing 80A (2 – 6S) ESC 4A SBEC	Tunigy Plush – 32 80A (2-6S) w/BEC
Continuous current [A]	80,00	80,00
Burst current [A]	100,00	Not specified
No of battery cells	2 - 6	2 - 6
Voltage range [V]	5,60 – 25,20	8,40 – 25,20
BEC output [-]	5,50 V @ 4,00 A	5,50 V @ 5,00 A
BEC type	SBEC	SBEC
Size [mm]	67,00 x 32,00 x 23,00	50,50 x 31,50 x 20,0
Weight [g]	95,00	56,50
Programming	Can be done without program card	Needs program card
Availability	Only backorder	In warehouse
Price [EUR]	37,99	32,54

Table 3.3 ESC and EDF compatibility

No.	HobbyKing 80A (2 – 6S) ESC 4A SBEC	Dr. Mad Thrust EDF motor
Burst current [A]	100,00	83,00
No of battery cells	2 - 6	4
Voltage range [V]	5,60 – 25,20	11,20 – 16,80

Control surface servo actuators are difficult to select prior to design and testing phase. At this point, it is challenging to estimate magnitudes of force required to move control surfaces. Rotational range of 180 degrees is sufficient for servo application in this project. Another criterion is reliability, which can be increased by choosing servo actuator with metal gears. There is wide selection of actuators with different speeds, sizes, and torque that meet these two criteria. As a result, decision was made to use servos with slim design. Dimensions for standard and slim servo are shown in Figure 3.5. The parameters of the suitable servo models are given in Table 3.4. As other hardware components tolerate up to 5 V supply voltage, Corona DS – 239HV will not be considered a suitable servo actuator. Due to its properties, it needs an additional voltage regulator to ensure the correct input voltage. Table 3.4 reveals that all Turnigy models provide better speed and torque parameters but draw considerably more current. The need for speed and torque parameters is unknown before the test results. Therefore, it is reasonable to select the cheapest solution, bearing in mind that it can be easily replaced later with more capable servo. As a result, servo actuators used for flying platform will be Corona CS – 239MG.

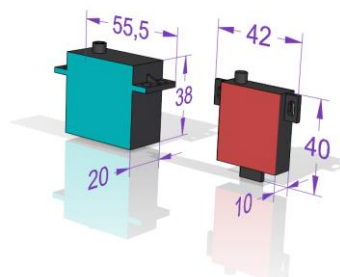


Figure 3.5 Servo actuator dimensions; Cyan – standard design servo actuator; Red – slim design servo actuator

Table 3.4 Servo models characteristics [22]

Model	Operating voltage [V]	Max operating current [A]	Max torque [Nm]	Weight [g]	Max speed [rad/s]	Price [EUR]
Turnigy TGY - A55H	4,80 / 6,00	1,20	0,84	28,40	8,73	51,59
Turnigy TGY - 811	4,80 / 6,00	Not specified	0,80	30,00	8,73	22,99
Turnigy - 813	4,80 / 6,00	Not specified	0,88	30,00	11,63	31,99
Corona DS - 239HV	6,00/ 7,40	0,40	0,45	22,00	8,06	9,86
Corona CS - 239MG	4,80 / 6,00	0,40	0,45	22,00	7,48	10,64

3.2.2 Power distribution

Power distribution circuit is responsible for ensuring sufficient voltage and current for components. From paragraph 3.2.1 and 3.2.3, it is found out that two voltage levels are necessary to provide proper function for final circuit. Servo actuators, microcontroller and sensor are powered with low voltage (5 V), and EDF motors and ESCs are powered with high voltage (14,8 V). Based on that, a power distribution circuit was constructed. Although selected ESCs have also BECs and they could also be used as low voltage sources, they are not used. If the flight platform is further developed into horizontal flight capable VTOL, the control surfaces are expected to remain functional in the event of ESC failure. For this purpose independent voltage regulator board was used. Power supply wiring diagram is shown in Figure 3.6.

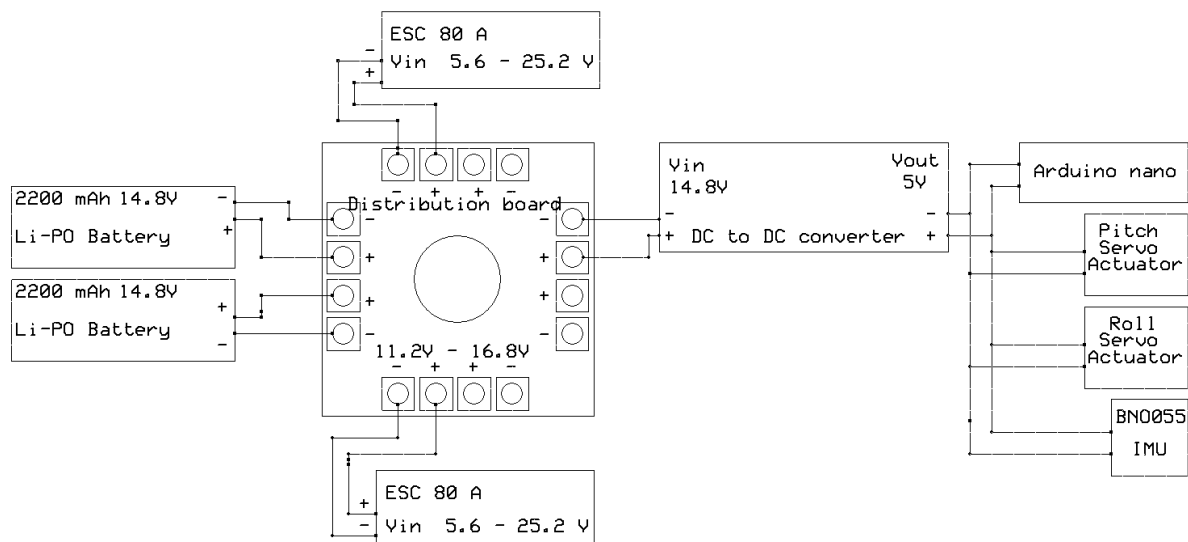


Figure 3.6 Power distribution scheme

Two batteries act as a power source. The decision for batteries were based on EDF motor characteristics. Table 3.1 shows that these motors use current up to 83 A at input voltage of 14,8 V. This means that four cell lithium polymer battery would be sufficient. It also defines maximum load condition and sets criteria for battery C -

rating, which indicates how fast battery can be safely discharged. The chosen batteries are Turnigy 2,2 Ah 60 – 120C LiPo. Characteristics compared to EDF motor are shown in Table 3.5. A maximum discharge rate of 120C means that whole battery capacity can be discharged within 30 seconds. It is calculated as follows:

$$t_{bat} = \frac{C_{bat}}{C_{rating} * C_{bat}} * 3600 = \frac{1}{C_{rating}} * 3600 \quad (3.2)$$

$$t_{bat} = \frac{1}{120} * 3600 = 30$$

Where: t_{bat} – minimum time to discharge battery [s]

C_{bat} – battery capacity [Ah]

C_{rating} – battery c – rating

This calculation also indicates that even if one battery fails, the other battery can safely power both motors in maximum load conditions. Although batteries can be discharged at a high rate, the powered time will be limited by capacity. Following equation provides calculation for actual discharging time for maximum load condition:

$$t_{min} = \frac{C_{bat}}{I_{EDF}} * 3600 \quad (3.3)$$

$$t_{min} = \frac{2,2}{83} * 3600 = 95,42$$

Where: t_{min} – Minimum discharge time [s]

C_{bat} – Battery capacity [Ah]

I_{EDF} – EDF motor maximum current [A]

To conclude, chosen batteries can power motors in case of maximum load for 95 seconds. It must be noted that calculations are made with maximum current conditions. In actual application, highest currents may be drawn only in hover mode. During horizontal flight phase (not developed during this paper) current will decrease considerably.

Table 3.5 Battery and EDF characteristics [22]

No.	Turnigy LiPo	Dr. Mad Thrust EDF motor
Burst current [A]	-	83,00
No of battery cells	4	4
Voltage range [V]	11,2 – 16,8	11,20 – 16,80
Maximum discharging c-rating	60 - 120	-
Capacity [Ah]	2,2	-
Maximum discharge or maximum continuous load [A]	264	83

Power distribution board shown in Figure 3.6 is responsible for connecting two batteries in parallel and providing voltage for consumers. This results in an overall system capacity of 4,4 Ah and a nominal voltage of 14,8 V. This is considered a high voltage level during this paper. It is used by EDF motors and DC-DC voltage regulator.

Low-level voltage is supplied by DC-DC converter. As seen in Figure 3.6, this voltage is used by servo actuators, IMU sensor, and microcontroller. Therefore, the overall current consumption of these devices will set the maximum current criteria for converter. Microcontroller's maximum supply current is 200 mA, two servo actuators together use maximum of 800 mA, and IMU sensor uses 12,3 mA. Therefore, overall maximum current is 1012,3 mA. Considering the possibility that servo actuators may need to be replaced, the maximum current could increase to 2412,3 mA (based on Table 3.4). Selected circuit board is DSWY2596 which uses LM2596 chip. Characteristics of this DC-DC voltage regulator are shown in Table 3.6. As the table shows, reserve for further developments is available.

Table 3.6 DSWY2596 DC to DC voltage converter characteristics [23]

Parameter	Value
Input voltage [V]	4 - 35
Output voltage [V]	1,23 - 30
Output max current [mA]	3000
Temperature range [C°]	-45 ... +85
Weight [g]	13

3.2.3 Control boards

Control is provided by an embedded system where a microcontroller performs controlling tasks. Inertial measurement unit (IMU) provides position feedback and acts as a sensor system. The system's heart is an Arduino Nano microcontroller, and sensing element is Adafruit BNO055 IMU. Selection of actuating elements is validated in paragraph 3.2.1.

Arduino Nano is an open-source hardware and official webpage [24] provides schematics on which the circuit board can be assembled. However, for current purpose, the board is purchased as a complete assembly. The microcontroller is responsible for receiving information from the IMU sensor via I2C interface, calculating correct inputs and sending them to actuators. Four digital output pins and two I2C pins are necessary for those purposes. However, remote control capability can be implemented for further development. For this purpose, additional three digital input

pins are considered. As this paper proposes solution for configuring the PID controller, three additional analogue input pins for potentiometers are also necessary. These are used to change PID gain parameters in real-time during experiment. More thorough control logic description is provided in paragraph 3.3. Arduino Nano specifications are given in Table 3.7. As seen from the table, number of I/O pins meet the set conditions. Furthermore, the operating voltage is suitable for power distribution circuit described in paragraph 3.2.2. Overall design also contributes to its relatively small dimensions and weight. As BNO055 is sending its data with a frequency of 100 Hz (see Table 3.8), clock frequency of microcontroller is assumed to be sufficient. If it turns out to be insufficient during further development, it could be replaced with a more powerful microcontroller. For example, Teensy 4.0 microcontroller has the same dimensions and is compatible with same compiler. It uses ARM Cortex-M7 processor, which provides clock frequency of 600 MHz. This allows replacing microcontroller without any changes in algorithm or design of fuselage attachments.

Adafruit BNO055 absolute orientation sensor uses from Bosch BNO055 chip. It uses ARM-Cortex-M0 microcontroller to process accelerometer, magnetometer, and gyroscope measurement data. It processes and sends filtered data in quaternions, Euler angles or vectors [25]. The advantage over many other sensors is that the data is already processed and ready for host controller. This reduces the load on the host and allows it to focus on the primary control functions.

Table 3.7 Arduino Nano specifications [24]

Microcontroller chip	ATmega328
Operating voltage [V]	5
Available flash memory [KB]	30
Clock speed [MHz]	16
Number of analog input pins	8
DC current per I/O pins [mA]	40
Input voltage [V]	7 - 12
Number of digital I/O pins	22
Number of PWM output pins	6
Power consumption [mA]	19
Dimensions (Length x Width) [mm]	45 x 18
Weight [g]	7

Table 3.8 Adafruit BNO055 specifications [26]

Microcontroller chip	BNO055
Chip operating voltage [V]	3,6
Board input voltage [V]	3,3 - 5,0
Data sample Frequency [Hz]	100
Power consumption [mA]	12,3
Dimensions (Length x Width) [mm]	27 x 20
Weight [g]	3

3.3 Flying platform construction

To construct proposed flying platform (design shown in Figure 3.1), all corresponding parts must be fabricated. Most of these are manufactured with 3D printer. Parts are modelled in Solid Edge 2021 software, sliced with Ultimaker Cura 4.12.1 software, and printed with Anet ET4 3D printer. While Solid Edge 2021 is a CAD software, Ultimaker Cura provides possibility of converting CAD models into g-code file format recognized by 3D printers. It applies specified printing parameters to model. Main parameters that can be configured for best printing performance are shown in Table 3.9. Additionally, slicing in Ultimaker Cura attaches file with movement coordinates, printing/travelling speed etc. Comparison of three different filament materials is given in Table 3.10. It reveals that ABS is not printable with available equipment due to specific needs. Although PLA would give better surface finish and overall aesthetic look, PETG was chosen to provide more durable parts. List of manufactured parts with printing time and material consumption is given in Table 3.11. Overall printing time is 42 hours and 9 minutes. This does not include design process and equipment preparation time. Failed attempts and repetitive prints due to on-going design development process are excluded from overall consumption and price calculations. All technical drawings are presented in Appendix 1.

Additionally, modifications to purchased hardware and manufacturing of special attachments were performed manually. Two special hinge bolts (Figure 3.7 Item number 6) were manufactured from a standard M6 bolt. Detail drawing is shown in Figure 3.8. Landing gear tubes (Figure 3.3 Item number 5) are manufactured by cutting 100 mm pieces from aluminium tube purchased from building material store. Modifications to used EDF motors include shortening shaft and removing unnecessary housings. Fuselage is completed after mounting subassemblies and parts together as shown in Figure 3.7.

Table 3.9 Selection of available printing parameters in Ultimaker Cura 4.12.1

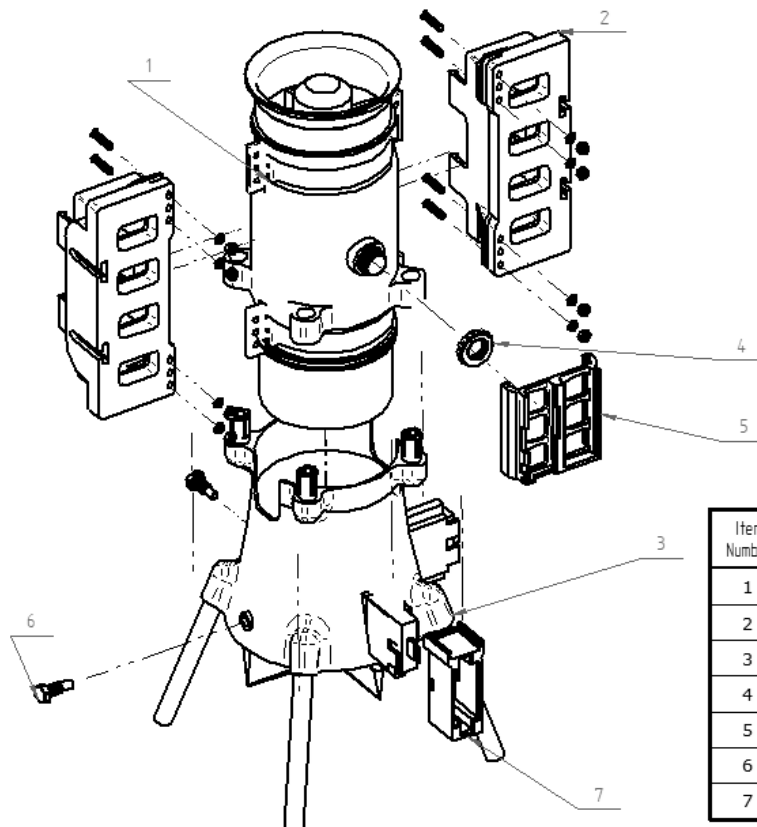
Parameter	Affects
Layer height	Surface roughness
Initial layer height	Core strength
Wall thickness	Core strength
Top/Bottom thickness	Core strength
Infill density	Overall strength
Infill pattern	Material cost, printing time, strength distribution
Printing temperature	Quality
Build plate temperature	Detail attachment strength to build plate
Support parameters	Quality
Build plate adhesion parameters	Detail attachment strength to build plate

Table 3.10 Filament comparison

Material	Strength	Printing temperature [C°]	Printable with Anet ET4
PLA	Less strength and durability	190 - 230	Yes
ABS	Strong and durable	230 - 240	No
PETG	Strong and durable	230 - 250	Yes

Table 3.11 3D printed parts and printing time

Part no	Description	Printing time [hh:mm]	Material consumption [g]
1	Motor assembly housing	07:48	73
2	Control frame connection ring	01:53	15
3	Power distribution board nut	00:08	1
4	Control frame assembly housing	14:48	115
5	Roll control surface	03:16	30
6	Pitch control surface	03:00	27
7	Battery compartment	03:38	29
8	Battery compartment reinforcement	04:28	46
9	Microcontroller attachment rack	01:58	14
10	IMU sensor attachment rack	01:12	10
Overall Consumption [g]			360
Price [EUR]			7,56



Item Number	Part name	Quantity
1	EDF motor assembly	1
2	Battery compartment assy	2
3	Control frame assy	1
4	Power distribution board nut	1
5	Controller attachment rack	1
6	Special bolt	2
7	IMU attachment rack	1

Figure 3.7 Flying platform assembly

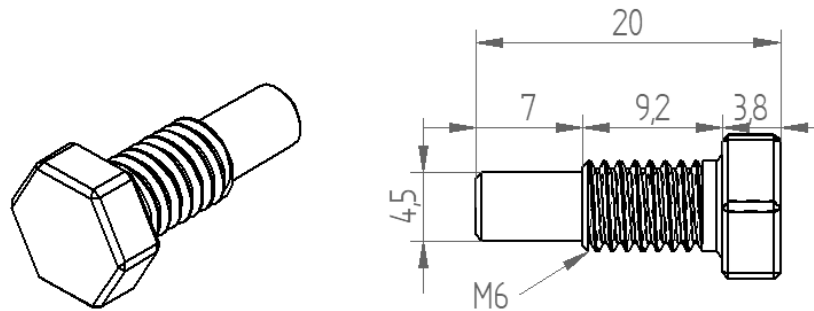


Figure 3.8 Special hinge bolt drawing with dimensions

An electrical circuit was completed to prepare a platform for algorithm implementation. Hardware connections and dedicated pins were equipped with connectors by soldering. The general block scheme of the control system is shown in Figure 3.9, where green lines represent PWM signal, purple lines represent I2C, red lines are low voltage supply, and blue lines are high voltage supply connections. More elaborate scheme is available in Appendix 2. Electrical design solutions and issues addressed in this paper are solved by information available in [27], [28], and [29].

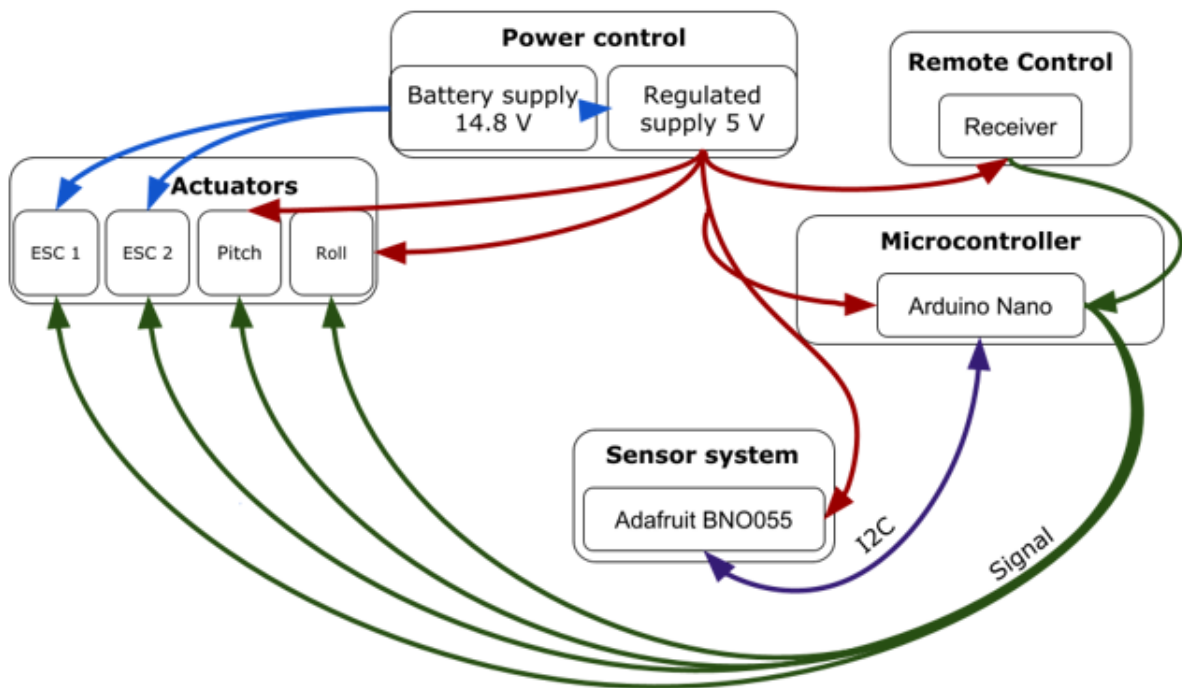


Figure 3.9 Hardware connection block scheme

Before constructing the aircraft, theoretical flight capability was assessed. For this purpose, all component weights were added together. Weight for manufactured parts was accessible from Ultimate Cura software that estimates the material consumption. Final product was expected to weigh approximately 1600 g. From Table 3.1 it is known that one EDF produces maximum thrust of 2300 g. Based on that, the manufacturing

and assembly processes were carried out. Due to not including attachment parts and wiring weight in calculations, it was expected that final prototype would weigh up to 150 g more than theoretically calculated. Completed prototype's gross weight is 1760 g.

3.5 Test bench design and construction

Research revealed that controllers could be tuned in different ways. During this paper, controller was expected to be tuned through experiments. Although a preliminary dynamic model description is introduced in paragraph 3.1, further mathematical model calculations are excluded from the scope of current thesis. This approach requires a test platform that allows performing experiments safely. Due to tests carried out in real environment, more precise controller could be achieved. In addition, significant time could be saved as future research projects can use the same test platform. The assumptions made to set dynamics requirements are based on the study results. Main criterion is that test bench must provide three degrees of rotational freedom. Test subject must have freedom to rotate around yaw, pitch, and roll axis. Another condition to meet is that flying platform attachments to test bench must be designed so that centre of gravity is adjustable. Additionally, the prototype should remain mobile and easy to upgrade during ongoing tests. To ensure mobility, dimensions must be designed to fit through standard building doors without disassembling. It should weigh 3 to 5 kg to ensure that a single person can operate it. Chosen design is a gimbal type. The difference is in frame assembly location, as it is expected that middle frame does not disturb airflow produced by drone. This is to ensure more realistic results in testing phase. Proposed frame assembly is shown in Figure 3.11. Assembly is divided into four subassemblies:

- Stationary support frame assembly
- Outer frame assembly
- Inner frame assembly
- Attachment assembly

All technical drawings for manufactured parts are available in Appendix 3. Stationary frame is base assembly that provides support for others. Outer frame assembly is directly attached to it through bearing. This provides rotation around yaw axis. Inner frame assembly is attached to outer frame assembly through two bearings to provide rotation around roll axis. Final attachment assembly acts as connection link between

flying platform and test bench. It is connected to the inner frame through two bearings to provide rotational freedom for pitch axis. Figure 3.10 shows attached flying platform into test bench and illustrates its dynamics.

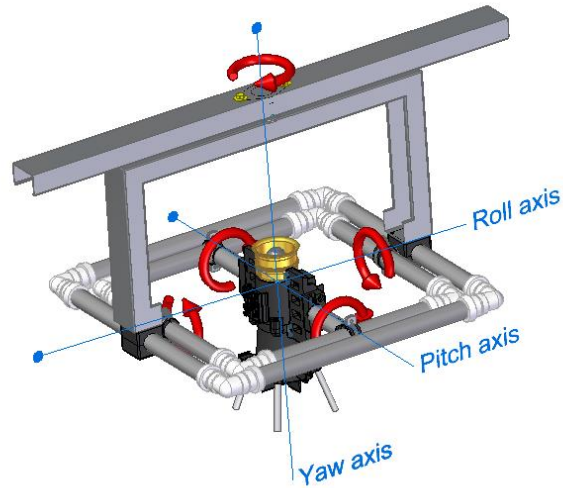


Figure 3.10 Test bench rotating frame axis placement

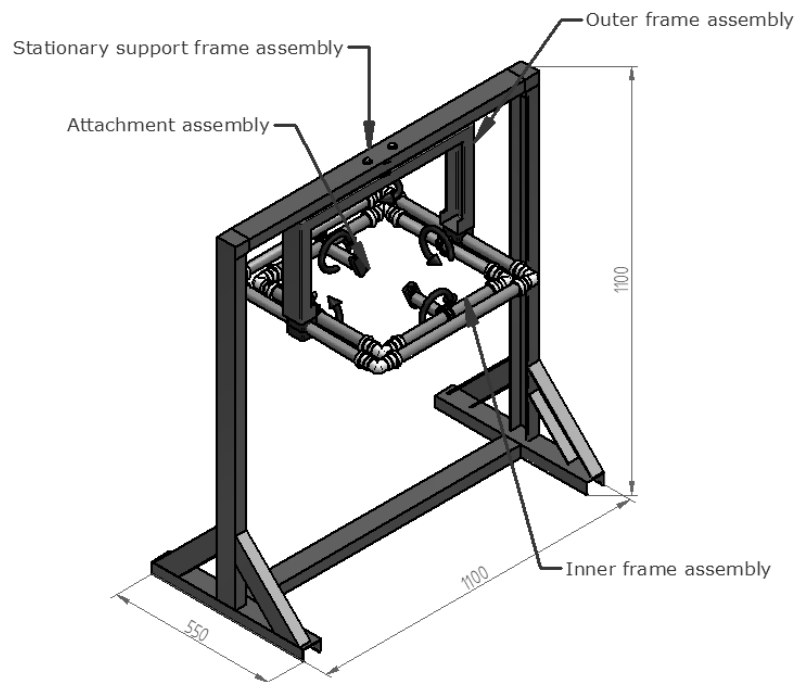


Figure 3.11 Proposed gimbal type frame with measurements, and arc arrows showing rotating joints

Stationary support frame is constructed from standard metal frame available in local construction retailer. The profile is shown in Figure 3.12. Parts are cut into length in accordance with design drawings and fastened using rivets and screws. Outer frame assembly is constructed using industrial electrical wiring protection tube (32 mm

diameter), construction metal frame and 3D printed bearing attachments. Tubular plastic material is lightweight, cheap and easy to process. To connect tubular pieces, plumbing angle connections with seals are used. They provide sufficient friction and maintain detachability for modifications during on-going testing phase. Same materials are used for inner frame assembly. Raw material is cut into length and assembled by design drawings. Additionally, commercial tube clamps are sufficient to attach assemblies with each other. This also allows to adjust the centre of gravity by axially changing the position of the clamp on the pipe. Figure 3.13 illustrates inner frame to outer frame attachment joint. Only one attachment is shown in the figure. The concept includes two attachments at opposite ends of the frame. These joints will also provide rotational freedom in roll axis (shown in Figure 3.10). Rotational freedom is achieved by use of bearing and 3D printed housing. This bearing assembly is designed so that the attachment position to the outer frame could also be adjusted. High adjustability can prevent collisions between frames in case of centre of gravity adjustment without extending overall frame dimensions.

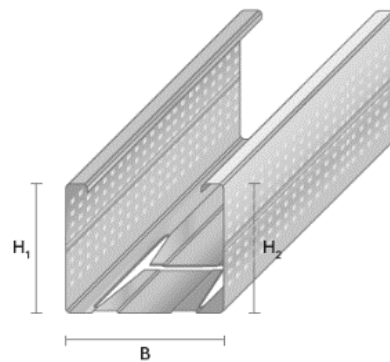


Figure 3.12 Raw material profile; $H_1 = H_2 = 35 \text{ mm}$ and $B = 66 \text{ mm}$ [30]

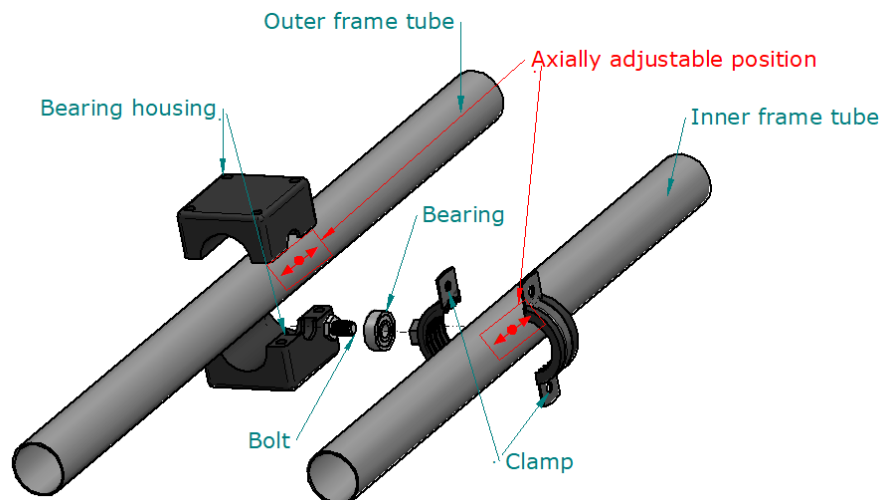


Figure 3.13 Inner frame to outer frame attachment

Attachment assembly is designed by following same principles. It must provide adjustability of centre of gravity, support for flying platform, and rotational freedom over pitch axis. Figure 3.14 illustrates joint adjustability, which ensures possibility to change centre of gravity during tests. The connection joint design is like conventional cross-member table of a milling machine. Additionally to axial movements shown in Figure 3.14, assembly has two bearings to provide rotational freedom over pitch axis.

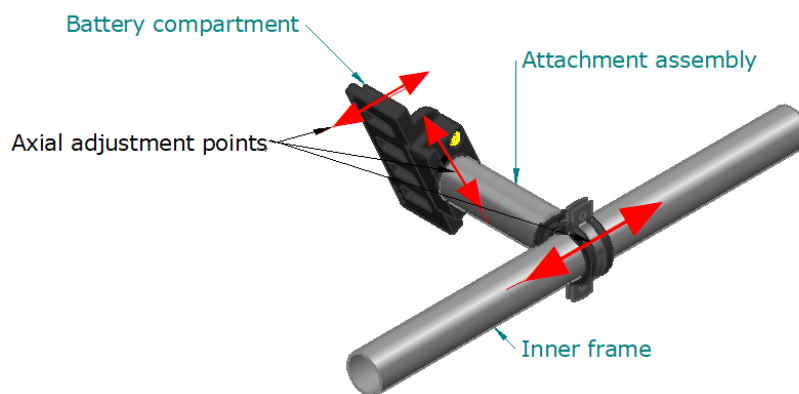


Figure 3.14 Inner frame and drone attachment adjustability

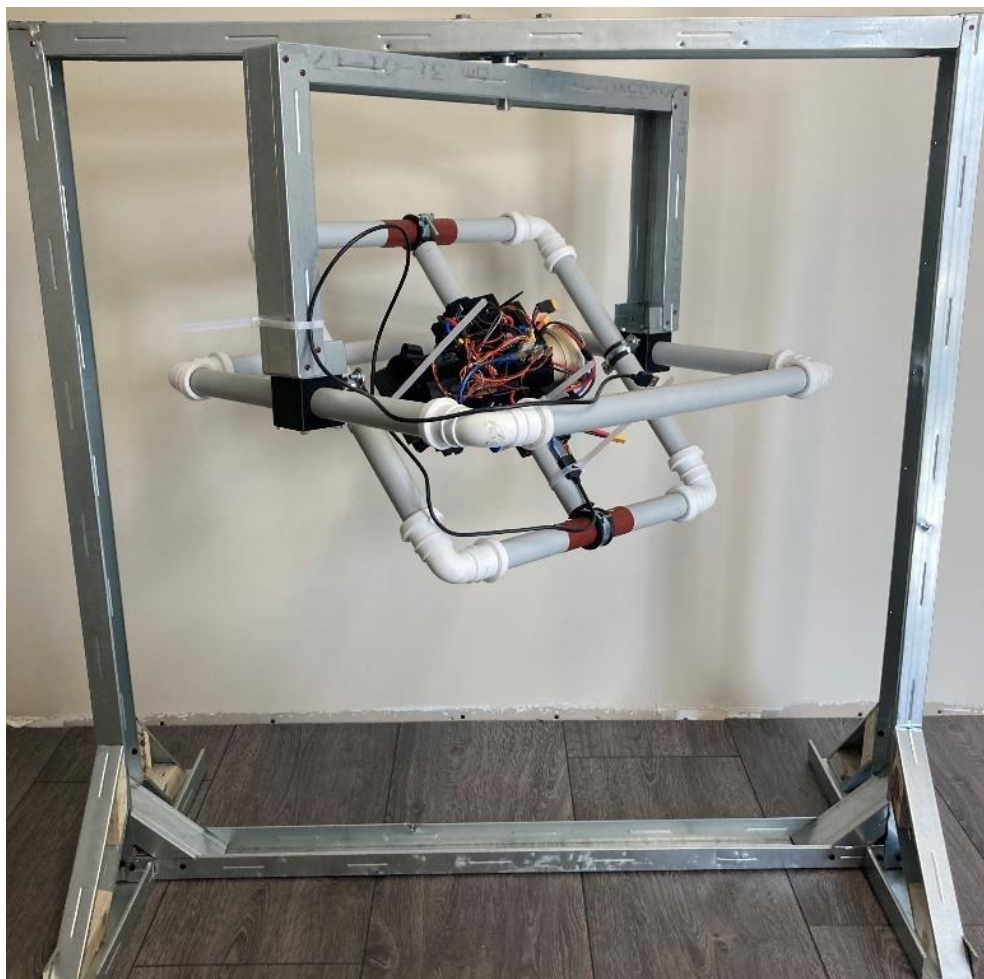


Figure 3.15 Flying platform attached to test bench

4. ALGORITHM AND TESTING

4.1 Algorithm development

Control algorithm is responsible for controlling and connecting hardware. It connects microcontroller, remote control system, sensor system and actuation system. Algorithm is written in free software Arduino 1.8.19. The code is written in C++ programming language with some unique additional methods and functions specific to Arduino only. General programming skills are learned from [31] and [32]. To validate functions, General Public Licenced software called SerialPlot 0.12.0 is used. This allows to plot data in more understandably. A general flowchart describing the contents of the algorithm is shown in Figure 4.1. The whole code is provided in appendix 4.

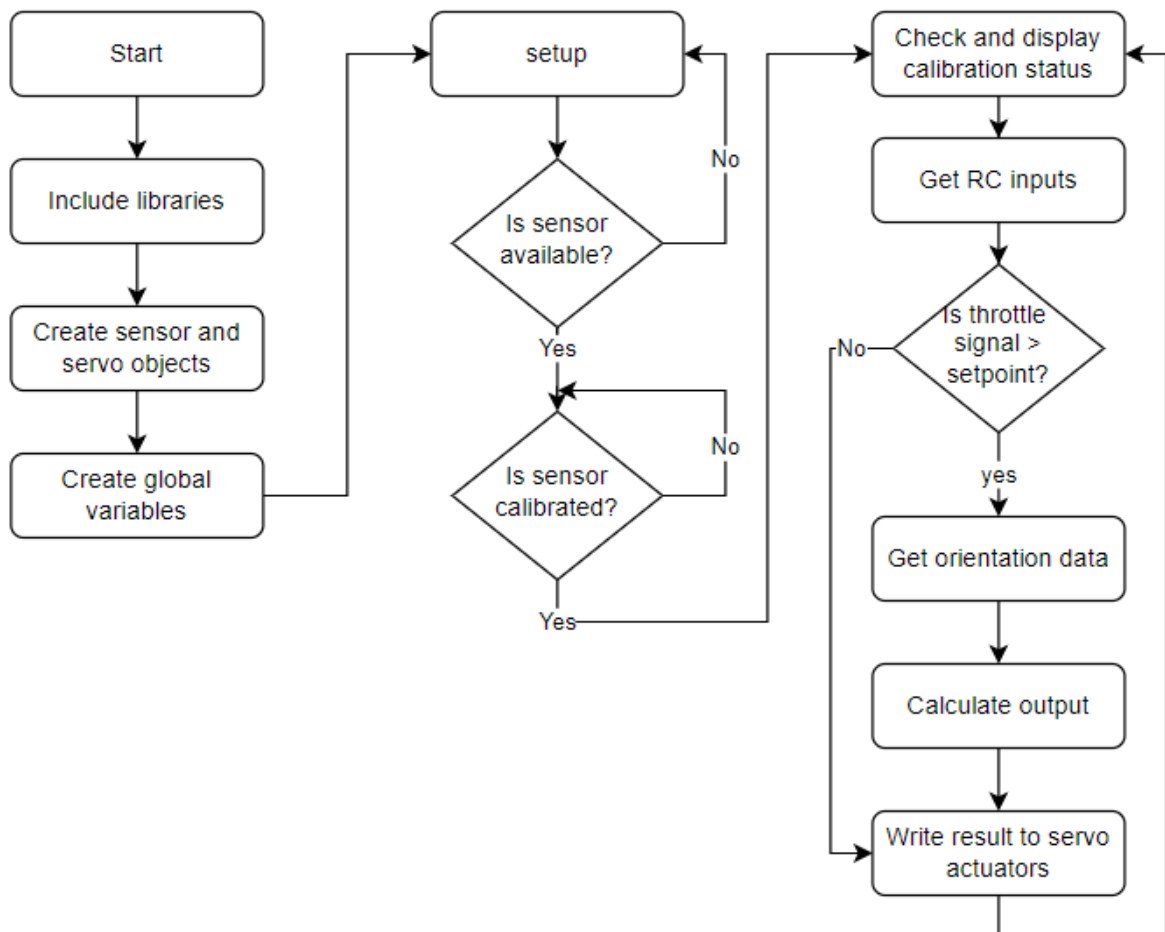


Figure 4.1 Algorithm flowchart

4.1.1 Used libraries

In Arduino, libraries are auxiliary files that provide certain functionalities. By using such files, programmers can use unified solutions to save time during development process. These files can be provided by hardware manufacturer, in which case they are referred as hardware drivers. They also could be algorithm files written by someone to provide universal solutions for common issues in field. Libraries included during current thesis are either under open-source licence or provided by hardware manufacturer. Algorithm uses six different libraries which are:

- PinChangeInterrupt library – Published under free MIT licence. For more details see [33]. Functionalities of this library are used to detect rising and falling edges of receiver signal. By determining time of each event, pulse width modulation signal will be calculated to achieve RC functionality
- Servo library – Published under free GNU Lesser General Public Licence v2.1. For more details see [34]. This library is used to communicate with actuators. It includes functions to read signals from actuators as well as to pass commands to them
- Wire library – Built in library to Arduino software. This allows to communicate over I2C devices. During current thesis it will be used to pass sensor data between microcontroller and Adafruit BNO055 sensor. For more information see [35]
- Adafruit_Sensor library – This is library provided to control various Adafruit sensors. It is good solution from hardware manufacturer as it collects all the sensor drivers into one library file. For more information see [36]
- Adafruit_BNO055 library – This is library provided by Adafruit hardware manufacturer. It can be said to be sub library for Adafruit_Sensor library.
- Utility/math library – Also library provided by Adafruit, dedicated to calculus. It can be called sub library for Adafruit_BNO055 library.

Overall, first library in list above is used to develop RC functions and second to develop servo control functionalities. All others are used to get information from sensor system which can be identified as sensor drivers.

4.1.2 Global variables and setup

After microprocessor is powered and libraries included, the global variables must be declared to ensure necessary functionality. Global variables are variable types that can be accessed by any part of program, therefore they are declared outside the functions. Each variable will be stored in microprocessor memory. Table 4.1 contains specifications of different variable types used in Arduino. As seen from this table, the storage space for same type of data can be extremely different in size. This in turn affects the performance of microcontroller. Therefore, it is reasonable to carefully analyse the maximum values for dedicated data and choose the smallest suitable variable type.

Table 4.1 Arduino variable types [37]

Name	Type	Size and range
byte	Round number	1 byte (0 to 255)
Int	Round number	2 bytes (-32768 to 32767)
long	Round number	4 bytes (-2147483648 to 2147483647)
unsigned int	Unsigned	2 bytes (0 to 65535)
unsigned long	Unsigned	4 bytes (0 to 4294967295)
float	Float numbers	4 bytes (3,4028235E+38 to -3,4028235E+38)
double	Float numbers	8 bytes (double float range)
char	Text data type	1 byte (-12 to 127)
unsigned char	Text data type	1 byte (0 to 255)
String	Text data type	N/A
bool/boolean	Binary true or false	1 bit (0 or 1)

These variables can be stored in arrays, allowing the same variable name with multiple values. For example, algorithm calculates three different PID equations which are identical in form. This means that there are three of each variable dedicated to each formula. One method would be to declare them as a separate variable, and another to declare variable as array with three elements. The last method keeps program clean and understandable. For clarity, variables are divided into two categories: RC function variables and PID function variables. Due to storing variables in arrays, RC function uses 5 different variable arrays instead of 15 separate variables. PID function uses 14 variable arrays instead of 37 separate variables. The table of variable arrays is provided in Appendix 5.

The setup function is responsible for starting and setting connections. It will start serial connection to ensure program capability to send data over serial port. Further it will allocate hardware connection pins, check if sensor is available, and set sensor parameters. It also writes initial values to each actuator. Pitch and Roll actuator centre positions can be adjusted by modifying initial values.

4.1.3 RC function

This function is dedicated to get signal information from receiver. Receivers are connected to remote (transceiver) wirelessly and connected to microcontroller by signal cables. Through these lines PWM (Pulse-Width Modulation) signal is sent to processor. There are different methods to measure these signals. All of them will measure the time that signal remains in high state. Based on information in Table 4.2, decision for suitable method was chosen. As current version of algorithm contributes three input PWM signals (Throttle, Pitch and Roll), only Pin Change Interrupt method is sufficient. In addition, it provides an opportunity for further development.

Table 4.2 PWM signal different reading method comparison [38]

Function type	Principle	Pros	Cons
Pulse In function	waits pin to go high → starts timing → waits pin to go low → stops timing → returns length of pulse	No need for external library files; Measures and returns PWM signal; simple to use	Processor is occupied and cannot be used while waiting pulse
External Interrupts	Triggers interrupt when rising or falling edge of signal is detected → calls specified interrupt service routine (ISR)	No need for external library files; Measures and returns PWM signal Better use of processor	Number of pins is limited (only two pins can be used)
Pin Change Interrupts	Triggers interrupt when rising or falling edge of signal is detected → calls specified interrupt service routine (ISR)	All 20 pins can be used; Measures and returns PWM signal; Processor can be used for other functions while waiting signal changes	Needs external library file;

Function is written based on examples included to used library. Function will measure three different signals (Throttle, Pitch and Roll). Pulse length is expected to be in range of 1000 μs to 2000 μs , where the lowest value is seen as zero. This is common range for various servo actuators. First attempts revealed that the signal contained unacceptable interferences. As seen from Figure 4.2, signal (blue signal) interferences during 10 second measurement are significant. Signal length floats from 870 μs to 1110 μs . Pulse length is expected to be stable with length approximately 1000 μs . Signal present in Figure 4.2 would trigger the throttle actuators between 0 and 10 % range, which is unacceptable. Y-axis on graph represents pulse length and x-axis represents time.

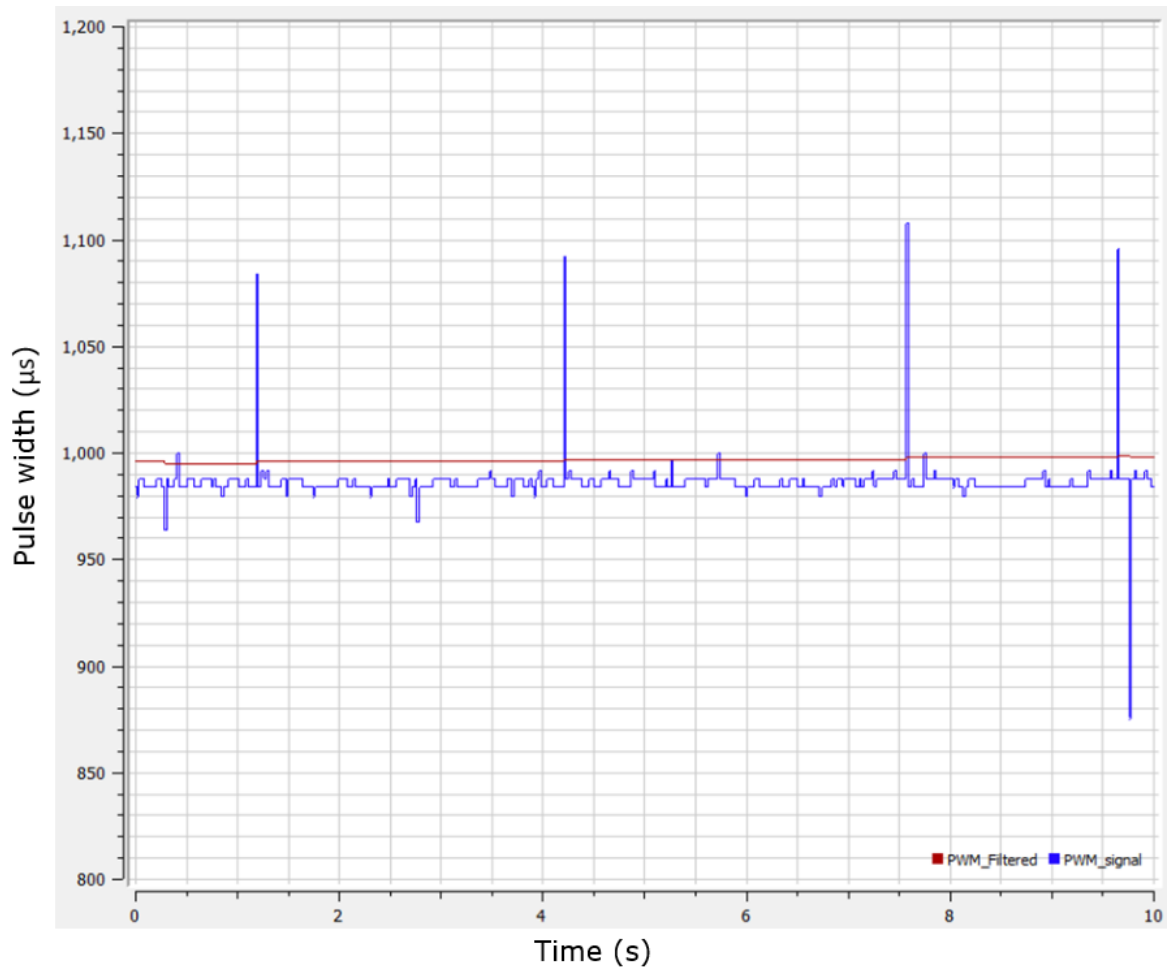


Figure 4.2 Unfiltered PWM signal with pulse length of 1000 μs

To provide more stable signal, algorithm based low pass filter is used. For this purpose, past signal value must be stored in memory. Then the program calculates (Equation 4.1) final value. The principle is based on amount of trust given to measured signal value. In Equation 4.1 constants M_1 and M_2 values added together must give result of 1. For throttle signal constants $M_1 = 0,99$ and $M_2 = 0,01$ are used. Old value is accounted for 99 % and new value only 1 %. This approach allows to adjust values based on specific circumstances. The consequence for aiming highly accurate and stable signal is that the speed of signal response will decrease. Figure 4.3 shows three graphs, each representing throttle pulse (full range from 0 % to 100 %); one unfiltered signal; and two filtered signals. It is clear from the graphs that the finer the filter, the slower the response. It is reasonable for throttle to give up speed of response to achieve a more accurate signal. For roll and pitch RC signals identical filter can be used with fine-tuned individual constants. They are not explained during current thesis as they are irrelevant to achieving the objectives.

$$PWM_{FINAL} = M_1 * PWM_{OLD} + M_2 * PWM_{FINAL} \quad (4.1)$$

Where: PWM_{FINAL} – PWM signal returned by function [μs]

PWM_{OLD} – Previous PWM signal returned by function [μs]

M_1 – Constant for previous signal

M_2 – Constant for current signal

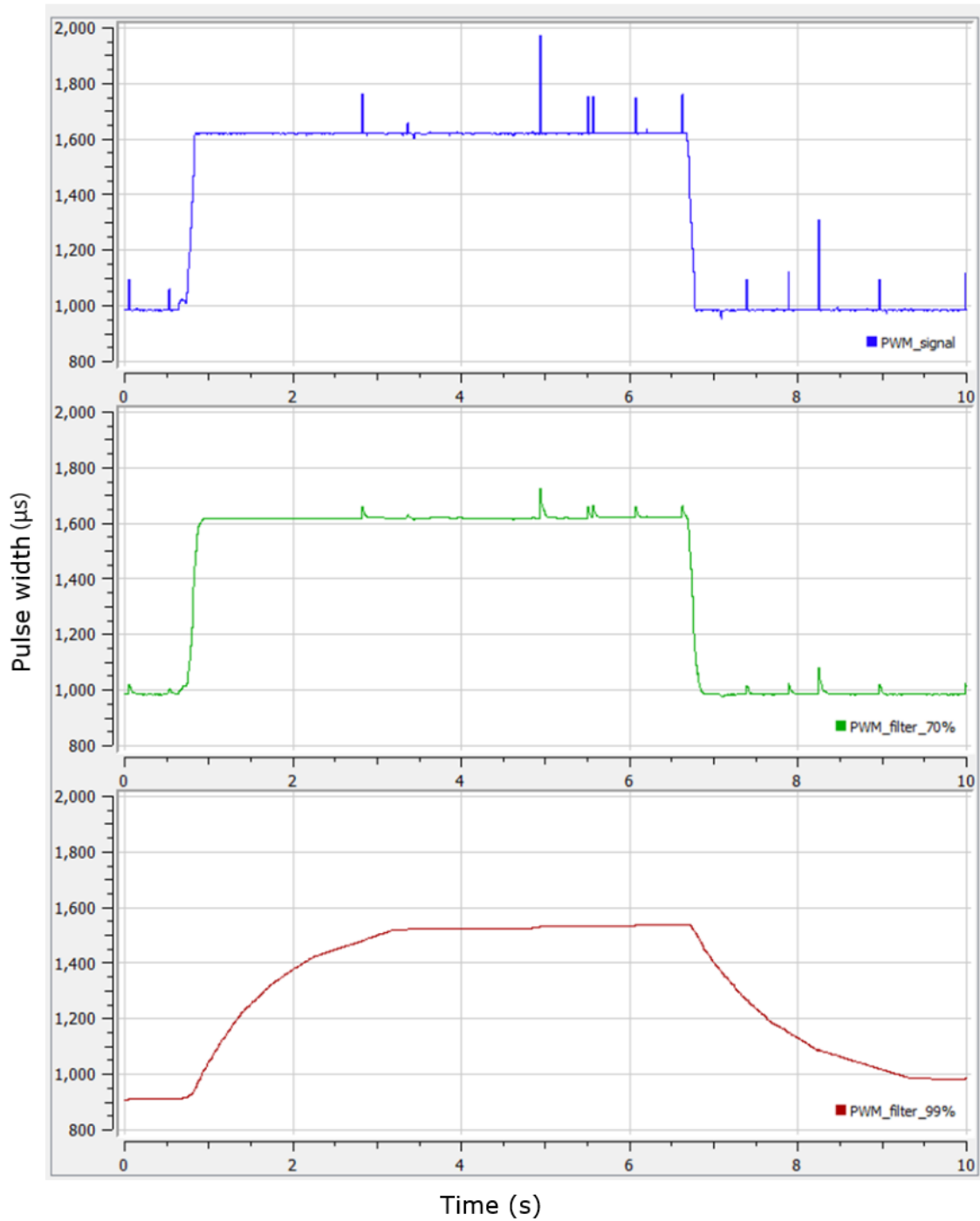


Figure 4.3 RC signal input response with different filter constant values: blue - unfiltered signal, green – 70 % filtered signal, red – 99 % filtered signal

4.1.4 Sensor and PID function

Sensor function uses Adafruit unified sensor library functions to get information from sensor. This library (explained in paragraph 4.1.1) makes it simple to achieve understandable data directly from hardware. To reach the goals of thesis, only orientation information is necessary. It allows to track actual angle of pitch and roll axis, which is used for further calculations. The sensor also provides angular velocity, acceleration (rotational and linear), magnetic field, and temperature information, which widens possibilities to improve prototype in further research. The function also measures time between previous and current measurements. It is used in PID equation and to assess overall program performance.

PID stands for proportional integral derivative controller. These controllers are chosen to provide correction signals for stabilization. Such solutions have been used widely in industry. PID controllers are simple, robust, and with good performance [13 p. 111]. This principle can be used as P (proportional), PI (proportional integral), PD (proportional derivative) or PID controller. General formula is shown in Equation 2.1, which can be separated into three different equations. Development of function is based on [13 pp. 112 - 113]. Equation parameters are replaced in algorithm by the names of the actual program variables. For clarity following equations are shown with actual program variables. Equation 4.2 is proportional component of formula, Equation 4.3 is integral component of formula, and Equation 4.4 is derivative component of formula. As seen from Equation 4.2, proportional component provides constant correction for error. This means that the error will be eliminated with equal steps depending on error and proportional gain. As a result, linear graph is drawn. Regarding current application, error component is calculated using Equation 4.5. Proportional controller alone will not eliminate steady state error and is usually slow controller. Integral component is added to eliminate steady state error. This means that if the proportional force is insufficient, the integral component will increase the force. To do so, error itself and time that it has persisted are accounted for. Steady state error is calculated using Equation 4.6. Derivative component is responsible for reducing overshoots. It will dampen the proportional component by reducing its rate as it approaches to zero. Putting all together, results in Equation 4.8 that solves overall correction signal for actuators.

$$P = K_1 * Error \quad (4.2)$$

Where: P – Proportional component

K_1 – Proportional gain

$Error$ – Difference between desired position and actual position

$$I = K_2 * steadyerrorArea \quad (4.3)$$

Where: I – Integral component

K_2 – Integral gain

$steadyerrorArea$ – steady error

$$D = K_3 * Slope \quad (4.4)$$

Where: D – Derivative component

K_3 – Derivative gain

$Slope$ – error change per time

$$Error = Target - Actual \quad (4.5)$$

Where: $Target$ – desired position

$Actual$ – position measured by sensor

$$steadyerrorArea = steadyerrorArea + Error * dt \quad (4.6)$$

Where: dt – time between current and previous calculation

$$Slope = \frac{ErrorDif}{dt} \quad (4.7)$$

Where: $ErrorDif$ – difference between current and previous error

$$PWM = PWM + K_1 * Error + K_2 * steadyerrorArea + K_3 * Slope \quad (4.8)$$

Where: PWM – correction signal

The actual function that performs all previously introduced calculations is shown in Figure 4.4. As described in section 4.1.2, variables are stored as arrays and number in parentheses behind each variable indicates the position of desired value in array. The IF function after PID calculation, will ensure that in case actuators are not providing enough force (i.e. in case of insufficient airflow), the correction value is not increasing over specified value. This is necessary to provide accurate and rapid response from controller. Such phenomenon is also called integral windup. Every actuator has its limit values where actuators become ineffective. As a result, integral part of controller continues to increase [13 pp. 129 - 135]. Although [13] offers various solutions for anti-windup, in this work it is decided to map overall output to solve the issue. The map function converts values in range of -1000 to 1000 into values that can be directly written to servo actuator. For pitch, these values are calibrated to range from 65 to 95 (actual degree values). It ensures that servo will not move into its mechanical design limits. This gives pitch control surface 30 degrees of range. Control program for roll actuator is the identical.

```

//PID calculation function//
void PIDfunction() {
  // Pitch error PID equation
  ErrorOld[1] = Error[1];
  Error[1] = Target[1] - Actual[1];
  ErrorDif[1] = Error[1] - ErrorOld[1];
  Slope[1] = ErrorDif[1] / dt;
  steadyerrorArea[1] = steadyerrorArea[1] + Error[1] * dt;

  PWM[1] = PWM[1] + k1[1] * Error[1] + k3[1] * Slope[1] + k2[1] * steadyerrorArea[1];
  if (PWM[1] > 1000) {
    PWM[1] = 1000;
  } else if (PWM[1] < -1000) {
    PWM[1] = -1000;
  }
  PWM[1] = map(PWM[1], -1000, 1000, 65, 95);
}

```

Figure 4.4 Program code in Arduino IDE that returns correction signal for pitch actuator

4.1.5 Servo function and main loop

Servo function is responsible for sending commands to servo actuators. Due to Arduino servo library file (described in section 4.1.1) this function is relatively short and simple. It uses built in commands to either write to specified servos or to read current signal of servos. For scope of thesis, only write commands are used. Library allows to write different types of signals to servos. For EDF motors, the signal is passed directly from RC function and is written in microseconds (range 1000 to 2000).

As motors are three phase DC motors, direction of rotation is not set in program code, but using wiring to select different phase setup. Roll and pitch actuator commands are recognized in range of 180 degrees, which is made compatible with the output of the PID function.

In main loop, all previously described functions are called in correct order. First the calibration function (not described in detail as it is built in function for sensor libraries) will check if the sensor system is still calibrated. Further sensor function gets position information, passes it to PID function and finally servo function sends command to actuators. This sequence will be performed over and over until the microcontroller is powered. For tuning and prototyping main loop also contains commands that send data to serial port. This enables to monitor specific parameters during live trials.

4.2 Testing and PID Tuning

To validate solution, two tests were carried out. It is essential to assess VTOL platform capabilities and test bench suitability for PID gain tuning. After completion of the tests, it is expected that the flight platform removed from the test bench will be ready for field testing.

4.2.1 Thrust map test

The first test was carried out to map thrust of the engine. Based on [2] and theoretical specifications of individual EDFs, constructed tandem EDF motor is expected to provide more than 3000 g of thrust. To validate expectations, test environment was constructed. For this purpose, scale with measurement range of 0 to 5000 g and tolerance of 1 g was used. Flying platform to scale attachment consists of 3D printed side supports and aluminium composite plate. Overall setup is shown in Figure 4.5. Throttle signal was limited in each phase by mapping corresponding variable in algorithm. Pitch and roll servos were deactivated to ensure stability. Each test phase was dedicated to certain throttle percentage. As a result, throttle map in respect to thrust was composed. Results are shown in Table 4.3 and throttle map is shown in Figure 4.6. Test revealed that expectations were optimistic and actual maximum thrust was approximately 2100 g. Reasoning behind difference in theoretical and actual thrust values are not investigated during this paper. Results are sufficient to

achieve aims of the thesis. Thrust map (Figure 4.6) function is nearly linear, which allows implementation of separate PID controller gain parameters for each region. Such controller approach would give accurate and stable flight characteristics.

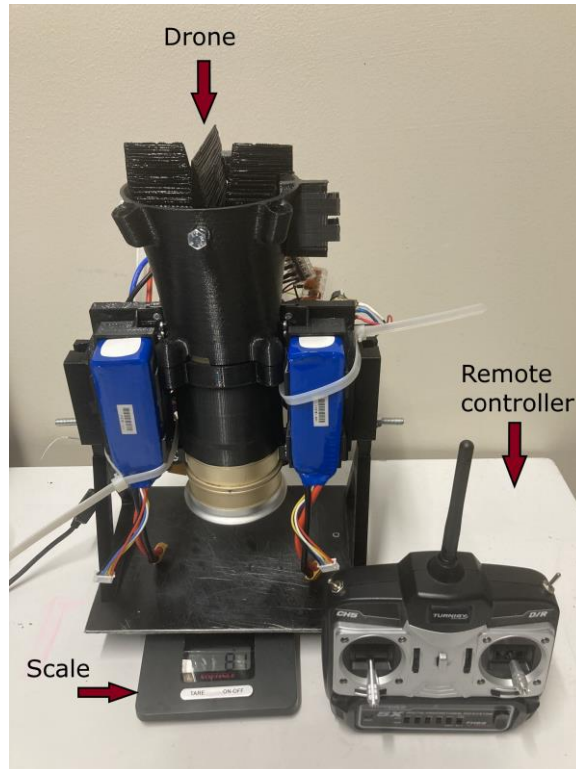


Figure 4.5 Throttle mapping test equipment

Table 4.3 Throttle mapping test results

Throttle [%]	Thrust [g]
10	143
20	425
30	673
40	902
50	1094
60	1253
70	1424
80	1602
90	1808
100	2093

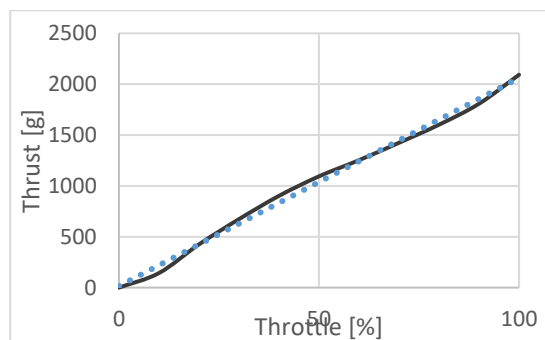


Figure 4.6 Throttle map graph: dashed line (blue) – trendline, black line - actual test result

4.2.2 PID gain tuning

To achieve sufficient control algorithm performance, PID equation gain parameters must be tuned. As previously explained, parameters are aimed to be tuned manually in this thesis. General rule for manual tuning (trial and error method) is that P-gain will be increased until plant enters state where output oscillates constantly. Secondly, I-gain value could be set so that oscillations are reduced. Finally, D-gain can be set so that system response is rapid. It is assumed that PID controller can show mainly five different dynamic behaviours. Graph in Figure 4.7, Figure 4.8, Figure 4.9, Figure 4.10, and Figure 4.11 show characteristics of each behaviour. Based on these scenarios, tuning decisions were made during the tuning process. More detailed description is available in [13 pp. 151 - 154].

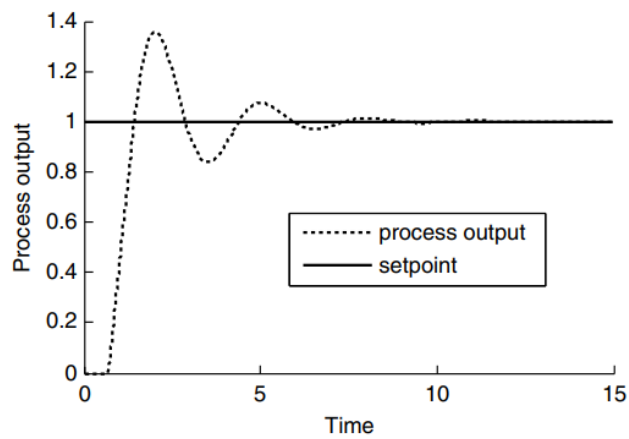


Figure 4.7 PID controller response for too high proportional gain [13 p. 152]

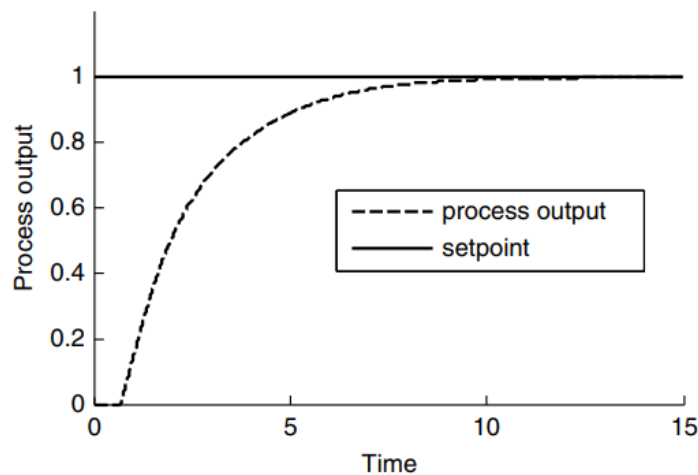


Figure 4.8 PID controller response for too small proportional gain [13 p. 152]

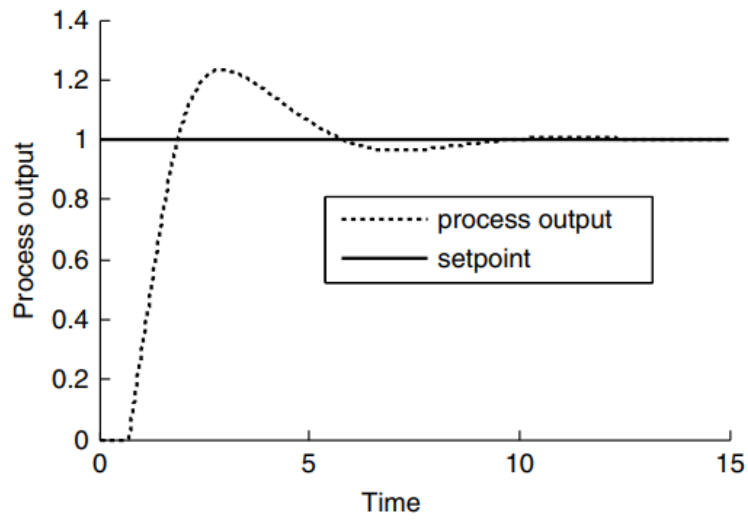


Figure 4.9 PID controller response for too small integral gain [13 p. 152]

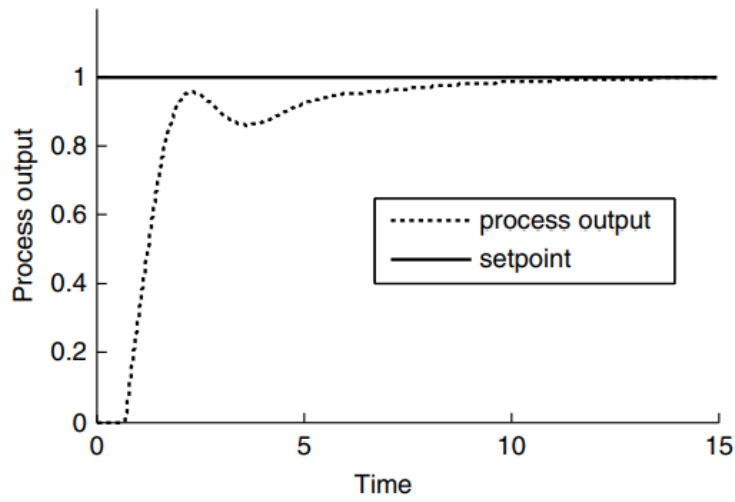


Figure 4.110 PID controller response for too large integral time [13 p. 153]

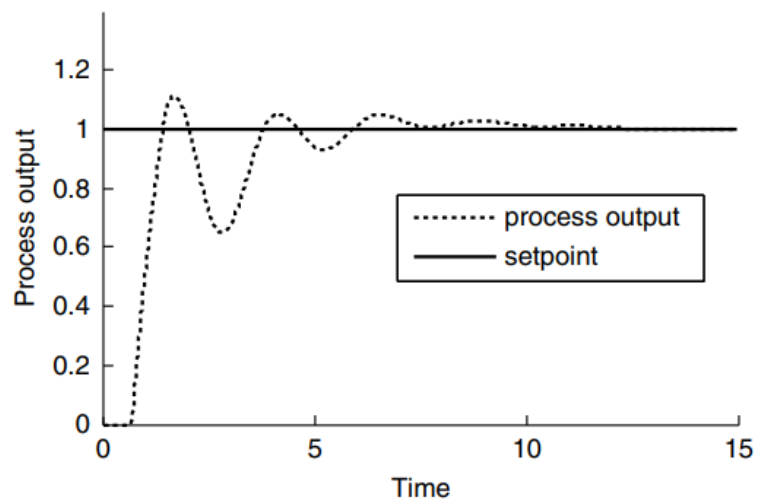


Figure 4.11 PID controller response for too large derivative time [13 p. 153]

Changing gain parameters by uploading algorithm with new values after each attempt is time consuming. Therefore, another function was added to algorithm and potentiometer was to hardware. Additional function maps the range of potentiometer values to specified gain variable. As gain values are set one by one, only one potentiometer will be used. This allows to fully tune one parameter before uploading values for another. To provide simultaneous parameter tuning, a separate potentiometer for each parameter could be considered for further development.

During this paper, only gain parameters for 80 % throttle were tuned. This is expected to be sufficient (based on thrust map) to validate proposed solution. The tuning process for each axis controller was identical. Therefore, only roll axis controller tuning process is described.

Preliminary conditions were:

- Test bench must be on stable floor
- Drone must be attached to test bench
- Drone centre of gravity must be set so that it does not affect test results
- Tuning functions in algorithm, must be turned on (by uncommenting dedicated functions) and set as desired
- Serial connection established with PC to monitor parameters
- SerialPlot software tuned to monitor actual position and setpoint in respect to time
- Other axis movements must be fixed

Next, remote controller and drone were turned on, throttle set to 80% and SerialPlot graph was monitored during trials. To evaluate PID response performance, disturbance must be applied. Each iteration, disturbance shall be identical to ensure testing repeatability. During tuning process, disturbance is selected to be 60 degrees from setpoint. Trial and error method requires multiple iterations to detect magnitude of gain parameters. Tuned graph for Roll controller response is shown in Figure 4.12. It indicates a too large proportional gain parameter when comparing with five dynamic behaviours (described above). Overall, graph shows that system stabilizes eventually.

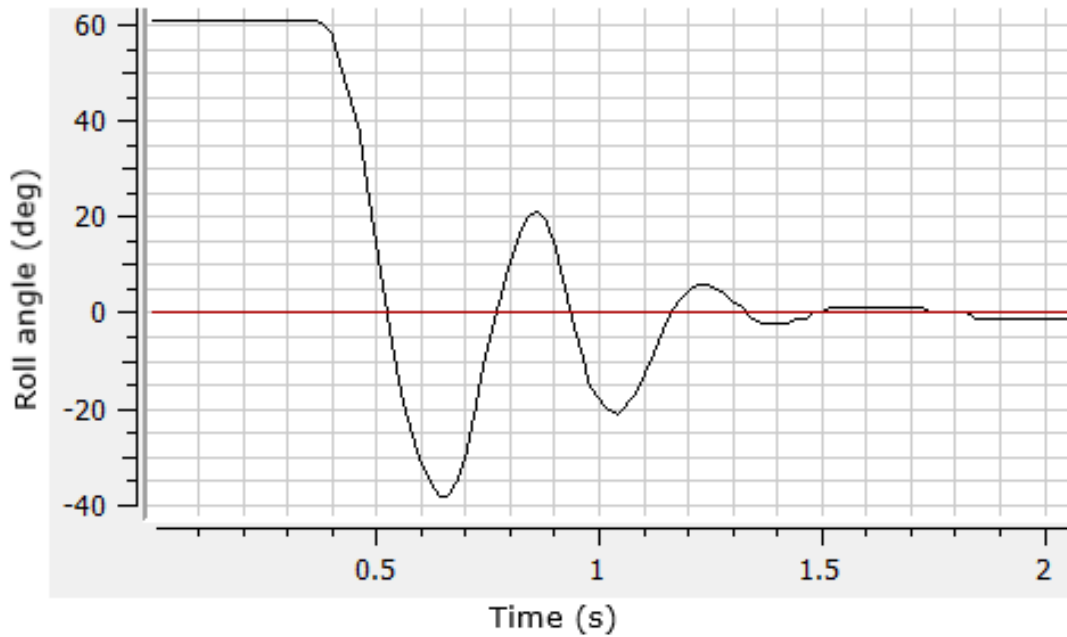


Figure 4. 12 Roll PID controller disturbance rejection graph; black line – actual position measured by inertial sensor; red line – setpoint

4.3 Conclusions

Based on study [2] and characteristics of EDF engines, the expected thrust was 3000 g or more. Thrust mapping trials revealed that the flying platform is sufficient for take-off, but the maximum thrust force did not meet expectations. The gross weight of drone is 1760 g and maximum thrust achieved by testing was 2093 g. According to the tests, additional 410 g of payload can be carried. EDF motors operate most efficiently with an axially moving intake airflow, which cannot be ensured by constructed test fixture. Therefore, it is assumed that actual thrust force is greater than test results revealed. This setback did not affect the achievement of the goals of thesis. If higher thrust is desired, it is possible to design a fixture that increases distance between scale and inlet to provide testing with more axial airflow. The experiment also showed that thrust force in respect to throttle level is linear. This creates a good precondition for the further development of multi-level controller.

PID controller tuning trials revealed that proposed principle for controller tuning is simple and effective, although further trials are necessary. It can be seen from Figure 4.12 that overshoot and oscillation are still present. The dynamic weight of cable connected between computer and drone affects dynamics of the model. Therefore, results differ for each iteration and the final parameters cannot be achieved. After

cable connection improvement, further controller tuning must be performed. Overall, trials validate that actuator system and algorithm are sufficient to control process output behaviour in desired way and system can be stabilized with two PID controllers. Experiments also identified several issues, that need to be eliminated before actual flight attempts. These issues were not addressed in the thesis, but are shortly described as follows:

- Although the monitored RC signals are clean after filtering, there are slight noise errors in the actuation system. Noise may be caused by DC to DC converter
- It is inconvenient to disconnect the power supply system. Power switch should be added for faster switching and safety
- Trial time is relatively short due to high power consumption. Additional battery to provide longer trial time should be considered
- The algorithm does not cover exceptions such as loss of remote controller signal etc.

SUMMARY

Due to the complexity of control mechanisms, VTOLs have not found their way into widespread commercial use. Thanks to the availability of advanced technology, it is possible to develop simpler systems. Such systems could be used in transportation, military field, surveillance etc. The thesis aimed to propose an automatic stabilization control algorithm solution for VTOL hover flight phase. During development process, a test bench for non-destructive tests and a VTOL drone platform were built as precondition for achieving main goal of thesis.

The second chapter is devoted to a critical analysis of existing research on a similar topic. It reveals numerous papers dedicated to control design and simulation of VTOL aircrafts. Many authors acknowledged that accurate dynamic models are too complex for realistic simulations. None of the existing research papers offered a solution for VTOL with tandem EDF engine. This thesis focuses on such innovative solution that makes simple and robust control design possible.

Learning from the shortcomings of other studies, it was decided to design a flight platform and a suitable test bench instead of a simulation method. Third chapter is dedicated to design, manufacturing, and construction process descriptions. It also includes component and solution selection description. Flight platform actuator system consist of two counterrotating EDF thrust motors and two servos that control pitch and roll movements. Positioning system relies on Adafruit BNO055 inertial sensor. The selected controller for the system is Arduino Nano. Test bench is designed to offer three degrees of freedom for non-destructive testing. Apart from purchased parts, most of complex details were manufactured by the 3D printing method. Software used during the process were SolidEdge 2021 student edition and Ultimaker Cura 4.12.1.

The fourth chapter describes algorithm development, structure and experiments that confirm the proposed solutions. Program code was written in Arduino IDE software and tests were monitored via serial port with SerialPlot software. The algorithm sequence principle is following:

- Read present condition from sensor
- Calculate controller output considering present and previous conditions
- Send commands to actuator system

Controller calculations are based on PID equation. Two experiments were carried out to validate concept: thrust vector force mapping, and PID gain parameter tuning. First test revealed linear dependency between throttle level and thrust force. It also confirmed that maximum thrust is sufficient to lift the platform. However, the maximum thrust value did not meet theoretical expectations. For the second experiment, a convenient solution was proposed to provide PID gain parameter adjustability during real-time experiments. This speeded up the traditionally time-consuming trial and error method. The original solution included three PID controllers. One for each axis, but first tests showed that counterrotating motors cancel out gyroscopic effects better than expected. This allowed to simplify the program code. Overall, trials validate the aims of thesis as drone output behaviour is automatically stabilized after disturbance. However, several challenges emerged which point to the need for further development.

The results of thesis provide a solid foundation for developing of the VTOL drone. The test bench prototype confirms a concept that can be further used on various flight platforms. On this basis, further development activities should be divided into two:

- Creating a larger in size test bench with integrated smart functionalities
 - With sensors that detect force vectors
 - With servos that allow to generate constant disturbances
- Further development of drone
 - Fine tuning PID parameters for each throttle level
 - Development of multi-level controller based on finite state machine
 - Field experiments
 - Wing design and construction
 - Horizontal flight phase development
 - Power efficiency analyse

KOKKUVÕTE

Keerukate juhtimismehhanismide tõttu ei ole VTOL õhusõidukid leidnud laialdast kommertskasutust. Tänu tehnoloogia arengule ja kättesaadavusele on tekkinud võimalus välja töötada lihtsamaid juhtimismehhanisme. VTOL kontseptsiooniga õhusõidukeid saab kasutada transpordi-, millitaar-, õhuseire- ja teistes valdkondades. Lõputöö eesmärk oli pakkuda välja automaatne stabiliseerimise lahendus VTOL ripplennu faasiks.

Teine peatükk sisaldab kriitilist analüüsi varasemalt tehtud uuringutele, mis käsitlevad sarnaseid teemasid. Analüüsi käigus selgus, et on tehtud mitmeid VTOL õhusõidukite juhtimissüsteemide loomisele keskenduval uuringuid. Enamik nende autoritest valideerib tulemusi simulatsiooni teel. Samas viitavad and asjaolule, et simulatsiooni eelduseks olev dünaamiline mudel on keerukas. Seetõttu ei ole simulatsioonis saadud tulemused võrreldavad reaalses keskkonnas sooritatud katsetustega. Lisaks selgus, et ükski analüüsitud uuring ei pakkunud lahendust kontseptsioonile, mis kasutaks tandem EDF mootorit. Käesolev lõputöö keskendub just sellisele innovatiivsele lahendusele, mis võimaldab luua lihtsa ja robustse juhtimismehhanismi.

Teiste uuringute puuduste põhjal otsustati simulatsioonimeetodi asemel valideerida pakutav lahendus füüsilises testkeskkonnas. Selleks konstrueeriti testpink ja VTOL alusplatvorm. Kolmas peatükk kirjeldabki disaini- ja konstrueerimise protsessi. Selles peatükis on lisaks välja toodud komponentide valiku põhjendused. VTOL alusplatvormi aktuaatorsüsteemi moodustavad kaks vastupidiselt pöörlevat EDF mootorit, mis tagavad tõukejõu, ning kaks servo mootorit, mis liigutavad juhtpindasid. Juhtpinnad tagavad kontrolli õhusõiduki kaldenurga (*pitch angle*) ja rullumisnurga (*roll angle*) üle. Õhusõiduki asendi tuvastamiseks kasutatakse Adafruit BNO055 inertsiandurit. Kogu süsteemi juhtajuna on kasutusel Arduino Nano mikrokontroller. Testpink on kavandatud selliselt, et see võimaldaks õhusõidukile kolmes teljes pöörlemiseks vabadusastet. Selline kontseptsioon võimaldab konfigurereida kontrollereid ning valideerida lahendust mittepurustavate katsete teel. Lahenduste tootmiseks kasutati 3D printimise meetodit. Kasutatavad tarkvarad tootmisprotsessi käigus olid SolidEdge 2021 student edition ja Ultimaker Cura 4.12.1.

Neljas peatükk kirjeldab algoritmi välja töötamise ja katsete protsesse. See peatükk sisaldab katsete tulemuste analüüsi. Juhtprogramm kirjutati Arduino IDE tarkvaraga ja katsete jälgimiseks kasutati serial porti koostöös SerialPlot tarkvaraga. Algoritmi põhimõtte on järgnev:

1. Andurilt loetakse positsiooni info
2. Mikrokontroller kalkuleerib vajaliku korrektsiooni
3. Mikrokontroller saadab kalkuleeritud käsklused aktuaatoritele
4. Andurilt loetakse uue positsiooni info (ring algab algusest ning jätkub kuni protsess peatatakse kasutaja poolt)

Kalkulatsioonid põhinevad klassikalisel PID võrrandil. Kontseptsiooni valideerimiseks teostati kaks erinevat katsete seeriat: tõukejõu kaardistamine ja PID parameetrite konfigureerimine. Esimene seeria näitas, et tõukejõu ja mootorite pöörlemiskiiruse vahel on lineaarne seos. See tõestas ka, et maksimaalne tõukejõud on piisav, et platvorm õhku tõsta. Siiski ei vastanud maksimaalsed tulemused teoreetilistel arvutustel põhinevatele ootustele. Teise katse sooritamiseks pakub käesolev töö välja mugava lahenduse, mis võimaldab PID parameetrite vahetut reguleerimist katsete käigus. Selline lahendus kiirendab aeganõudvat *trial and error* meetodit. Kuigi algne algoritmi lahendus koosnes kolmest PID kontrollerist (iga telje juhtimiseks eraldi controller), tõestasid esimesed testid, et vastupidi pöörlevad mootorid tühistavad güroskoopilised jõud oodatust paremini. Sellest tulenevalt on eesmärkide saavutamiseks vajalikud vaid kahe telje kontrollerid. Kokkuvõttes, katsed valideerivad lõputöö eesmärgid ning droon käitus testide käigus oodatult, tasakaalustades end automaatselt. Sellest olenemata selgus ka mitmeid väljakutseid pakkuvaid probleeme, mis viitavad edasise arendustegevuse vajalikkusele.

Lõputöö tulemused loovad tugeva aluse pakutud kontseptsiooniga VTOL drooni valmimisele ning valminud testpingi prototüüp kinnitab, et sellise kontseptsiooniga on võimalik testida erinevaid lennumasinaid. Edasised arendustegevuse suunad saab jagada kahte kategooriasse:

- Suurema ja integreeritud tarkade funktsioonidega testpingi loomine
 - Koos vektoriaalseid jõudusid tuvastavate anduritega
 - Koos servomootoritega, mis tagavad identsed häired testimiseks
- Edasised arendustegevused seoses drooniga
 - Igale pöörlemissageduse vahemikule PID parameetrite konfigureerimine
 - Mitmetasandilise PID kontrolleri loomine
 - Lennukatsed testpingi väliselt
 - Tiiva disaini ja konstrueerimisprotsessi teostamine
 - Juhtimissüsteem horisontaalsesse lennufaasi sisenemiseks
 - Kontseptsiooni energiatarbe analüüsi teostamine

LIST OF REFERENCES

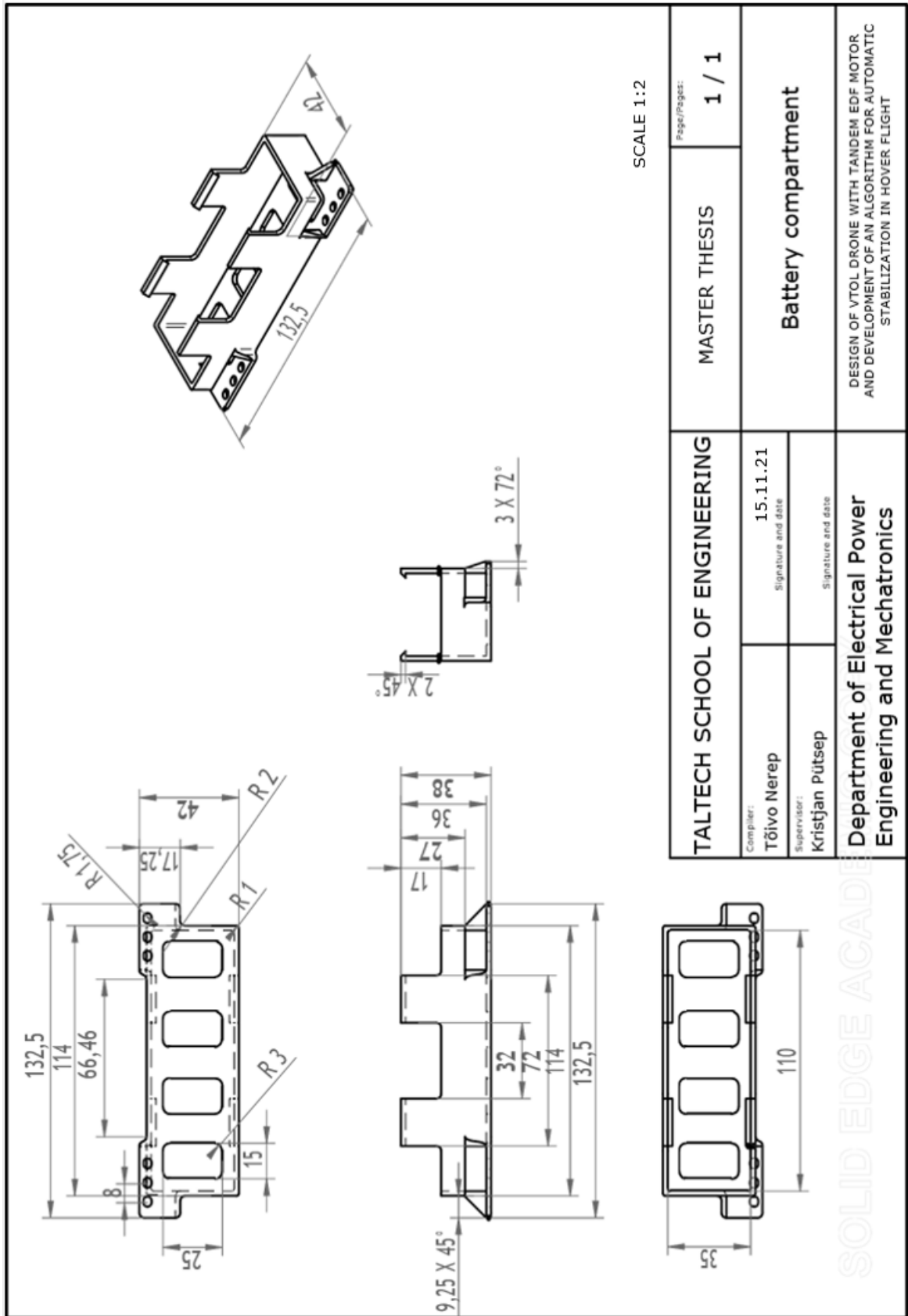
- [1] 'Lockheed Xfv-1 VTOL | MilitaryImages.Net'. [Online]. Available: <https://www.militaryimages.net/media/lockheed-xfv-1-vtol.7074/>. [Accessed: Jan. 05, 2022]
- [2] M. Lihulinn, 'Construction of Tandem EDF engine', Estonian Aviation Academy, Tartu, 2018.
- [3] G. J. J. Ducard and M. Allenspach, 'Review of designs and flight control techniques of hybrid and convertible VTOL UAVs', *Aerosp. Sci. Technol.*, vol. 118, p. 107035, Nov. 2021, doi: 10.1016/j.ast.2021.107035.
- [4] Y. Zhou, H. Zhao, and Y. Liu, 'An evaluative review of the VTOL technologies for unmanned and manned aerial vehicles', *Comput. Commun.*, vol. 149, pp. 356–369, Jan. 2020, doi: 10.1016/j.comcom.2019.10.016.
- [5] C. Notebook, 'Higher Education', *Aerodynamics & Performance*, 2022. [Online]. Available: <https://www.cfinotebook.net/notebook/aerodynamics-and-performance/aerodynamics-and-performance>. [Accessed: May 01, 2022]
- [6] F. A. Administration, 'Aerodynamic Factors', in *FAA-H-8083-15B, Instrument Flying Handbook*, U.S. Department of Transportation, 2012, p. 4.1-4.17 [Online]. Available: https://www.faa.gov/sites/faa.gov/files/regulations_policies/handbooks_manuals/aviation/FAA-H-8083-15B.pdf. [Accessed: May 01, 2022]
- [7] F. A. Administration, 'Aerodynamics of Flight', in *Helicopter Flying Handbook*, U.S. Department of Transportation, 2019, p. 2.1-2.26 [Online]. Available: https://www.faa.gov/sites/faa.gov/files/regulations_policies/handbooks_manuals/aviation/helicopter_flying_handbook/helicopter_flying_handbook.pdf. [Accessed: May 01, 2022]
- [8] A. T. Conlisk, 'Modern helicopter rotor aerodynamics', *Prog. Aerosp. Sci.*, vol. 37, no. 5, pp. 419–476, Jul. 2001, doi: 10.1016/S0376-0421(01)00011-2.
- [9] C. Rotaru and M. Todorov, 'Helicopter Flight Physics', in *Flight Physics - Models, Techniques and Technologies*, K. Volkov, Ed. InTech, 2018 [Online]. Available: <http://www.intechopen.com/books/flight-physics-models-techniques-and-technologies/helicopter-flight-physics>. [Accessed: May 01, 2022]
- [10] 'Dictionary.com', *Dictionary.com*. [Online]. Available: <https://www.dictionary.com/>. [Accessed: May 01, 2022]
- [11] A. T. Conlisk, 'Modern Helicopter Aerodynamics', *Annu. Rev. Fluid Mech.*, vol. 29, no. 1, pp. 515–567, 1997, doi: 10.1146/annurev.fluid.29.1.515.
- [12] J. M. O. Barth *et al.*, 'Fixed-wing UAV with transitioning flight capabilities: Model-Based or Model-Free Control approach? A preliminary study', in *2018*

- International Conference on Unmanned Aircraft Systems (ICUAS)*, Jun. 2018, pp. 1157–1164, doi: 10.1109/ICUAS.2018.8453404.
- [13] S. W. Sung, J. Lee, and I.-B. Lee, *Process identification and PID control*. Singapore: Wiley, 2009.
- [14] Z. Liu, Y. He, L. Yang, and J. Han, 'Control techniques of tilt rotor unmanned aerial vehicle systems: A review', *Chin. J. Aeronaut.*, vol. 30, no. 1, pp. 135–148, Feb. 2017, doi: 10.1016/j.cja.2016.11.001.
- [15] Z. Cheng, H. Pei, and S. Li, 'Neural-Networks Control for Hover to High-Speed-Level-Flight Transition of Ducted Fan UAV With Provable Stability', *IEEE Access*, vol. 8, pp. 100135–100151, 2020, doi: 10.1109/ACCESS.2020.2997877.
- [16] J. Xu *et al.*, 'Learning to fly: computational controller design for hybrid UAVs with reinforcement learning', *ACM Trans. Graph.*, vol. 38, no. 4, p. 42:1-42:12, Jul. 2019, doi: 10.1145/3306346.3322940.
- [17] A. Bonci, A. Cervellieri, S. Longhi, G. Nabissi, and G. Antonio Scala, 'The Double Propeller Ducted-Fan, an UAV for safe Infrastructure inspection and human-interaction', in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Sep. 2020, vol. 1, pp. 727–733, doi: 10.1109/ETFA46521.2020.9212035.
- [18] R. Naldi, L. Gentili, L. Marconi, and A. Sala, 'Design and experimental validation of a nonlinear control law for a ducted-fan miniature aerial vehicle', *Control Eng. Pract.*, vol. 18, no. 7, pp. 747–760, Jul. 2010, doi: 10.1016/j.conengprac.2010.02.007.
- [19] R. Naldi, L. Marconi, and A. Sala, 'Modelling and control of a miniature ducted-fan in fast forward flight', in *2008 American Control Conference*, Jun. 2008, pp. 2552–2557, doi: 10.1109/ACC.2008.4586875.
- [20] L. Marconi, R. Naldi, and A. Sala, 'Modeling and analysis of a reduced-complexity ducted MAV', in *2006 14th Mediterranean Conference on Control and Automation*, Jun. 2006, pp. 1–5, doi: 10.1109/MED.2006.328770.
- [21] Y. Zhang, C. Xiang, B. Xu, X. Wang, and W. Fan, 'Comprehensive nonlinear modeling and attitude control of a novel tandem ducted fan vehicle', in *2016 IEEE International Conference on Aircraft Utility Systems (AUS)*, Oct. 2016, pp. 50–56, doi: 10.1109/AUS.2016.7748019.
- [22] 'Radio Control Planes, Drones, Cars, FPV, Quadcopters and more', *Hobbyking*. [Online]. Available: https://hobbyking.com/en_us/. [Accessed: Oct. 10, 2021]
- [23] ON Semiconductor, 'LM2596 - 3.0 A, Step-Down Switching Regulator' [Online]. Available: <https://www.onsemi.com/pdf/datasheet/lm2596-d.pdf>. [Accessed: Dec. 15, 2021]

- [24] 'Arduino Nano', *Arduino Official Store*. [Online]. Available: <http://store.arduino.cc/products/arduino-nano>. [Accessed: Dec. 15, 2021]
- [25] Adafruit, 'Adafruit BNO055 Absolute Orientation Sensor', *Adafruit Learning System*. [Online]. Available: <https://learn.adafruit.com/adafruit-bno055-absolute-orientation-sensor/overview>. [Accessed: Dec. 15, 2021]
- [26] B. Sensortec, 'Intelligent 9-axis absolute orientation sensor', 2014 [Online]. Available: https://cdn-shop.adafruit.com/datasheets/BST_BNO055_DS000_12.pdf. [Accessed: Dec. 15, 2021]
- [27] J. O. Attia, *Circuits and electronics: hands-on learning with analog discovery*. Boca Raton [etc]: CRC Press, 2021.
- [28] T. L. Floyd, *Electronics fundamentals: circuits, devices, and applications*, 3rd ed. Englewood Cliffs (N.J.); Columbus (Ohio): Prentice Hall, 1995.
- [29] T. L. Floyd, *Principles of electric circuits: conventional current version*, 9th ed., International ed. Boston (Mass.) [etc.]: Pearson, 2010.
- [30] 'Profiline metallkarkass kipsseinte ja -lagede ehituseks', *Favor*. [Online]. Available: <https://favor.ee/profiline/metallkarkass/>. [Accessed: May 10, 2022]
- [31] M. McGrath, *C++ programming in easy steps*, Fifth edition. Leamington Spa, Warwickshire, United Kingdom: In Easy Steps, 2017.
- [32] M. R. Gregoire, *Professional C++*, Fourth edition. Indianapolis: Wrox, 2018.
- [33] 'PinChangeInterrupt - Arduino Reference'. [Online]. Available: <https://www.arduino.cc/reference/en/libraries/pinchangeinterrupt/>. [Accessed: Feb. 10, 2022]
- [34] 'Servo - Arduino Reference'. [Online]. Available: <https://www.arduino.cc/reference/en/libraries/servo/>. [Accessed: May 10, 2022]
- [35] 'Arduino - Wire'. [Online]. Available: <https://www.arduino.cc/en/reference/wire>. [Accessed: Feb. 10, 2022]
- [36] *Adafruit Unified Sensor Driver*. Adafruit Industries, 2022 [Online]. Available: https://github.com/adafruit/Adafruit_Sensor. [Accessed: May 10, 2022]
- [37] 'Arduino Variable Types [Complete Guide]', May 03, 2021 [Online]. Available: <https://roboticsbackend.com/arduino-variable-types-complete-guide/>. [Accessed: May 10, 2022]
- [38] 'Three Ways To Read A PWM Signal With Arduino | BenRipley.com' [Online]. Available: <https://www.benripley.com/diy/arduino/three-ways-to-read-a-pwm-signal-with-arduino/>. [Accessed: Feb. 10, 2022]

APPENDICES

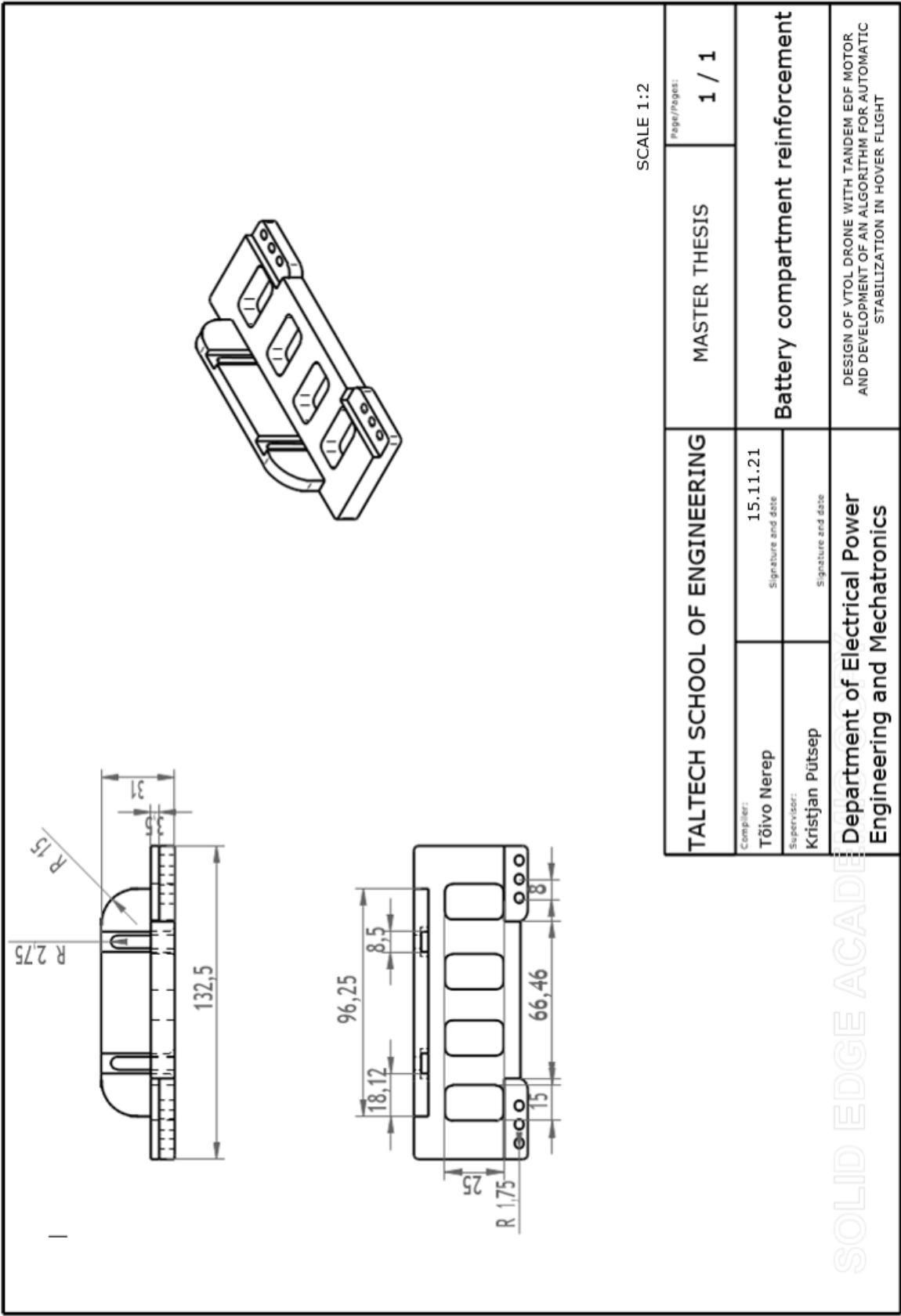
Appendix 1 Technical drawings of flying platform



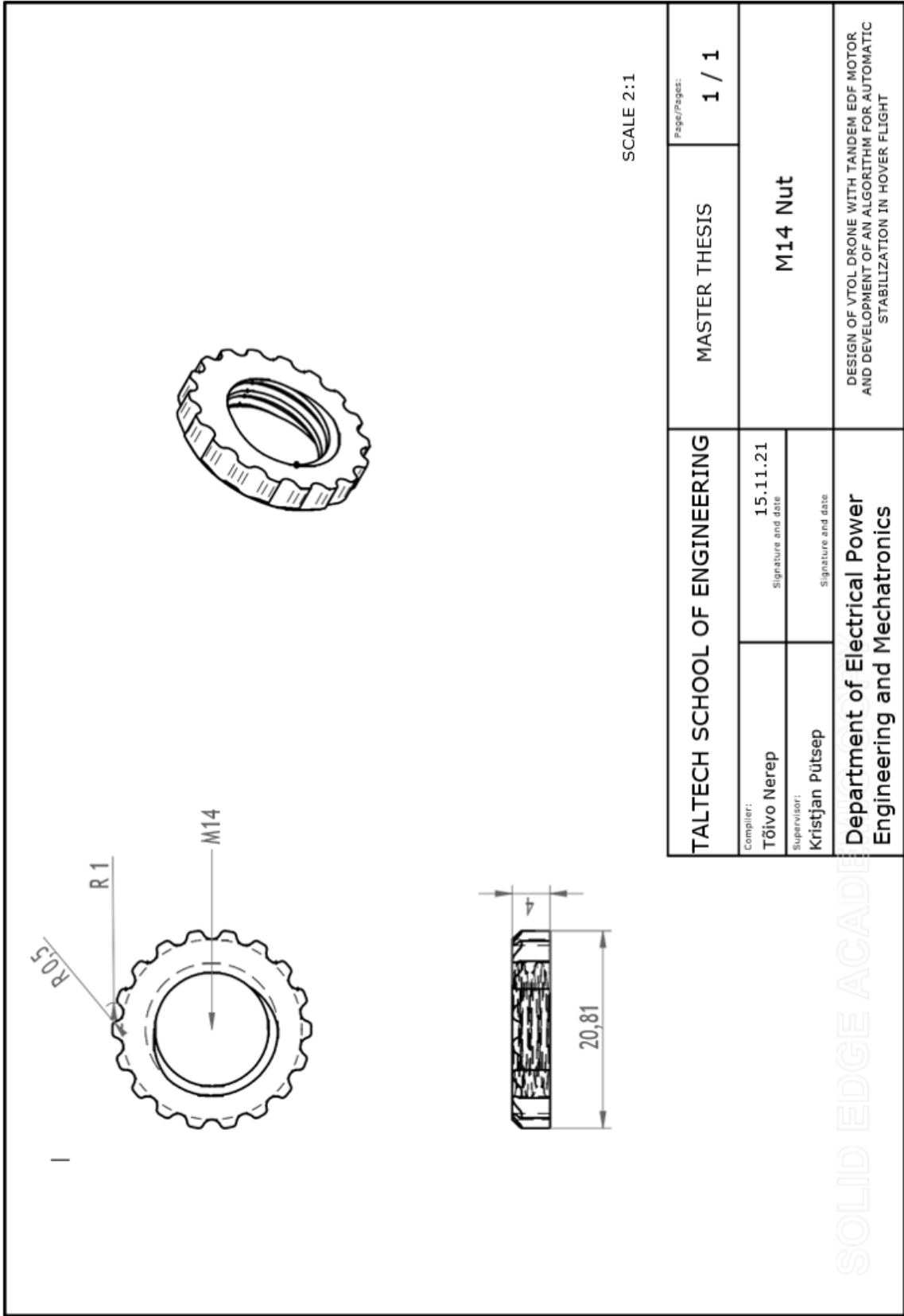
SCALE 1:2

TALTECH SCHOOL OF ENGINEERING		MASTER THESIS	Page/Pages: 1 / 1
Compiler: Tõivo Nerep	Signature and date 15.11.21	Battery compartment	
Supervisor: Kristjan Pütsep	Signature and date		
Department of Electrical Power Engineering and Mechatronics		DESIGN OF VTOL DRONE WITH TANDEM EDF MOTOR AND DEVELOPMENT OF AN ALGORITHM FOR AUTOMATIC STABILIZATION IN HOVER FLIGHT	

SOLID EDGE ACADEMY



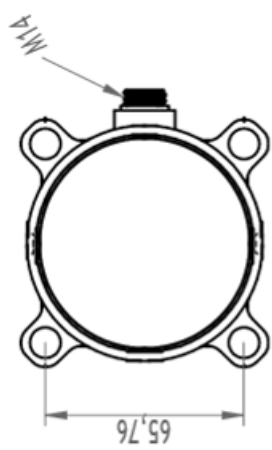
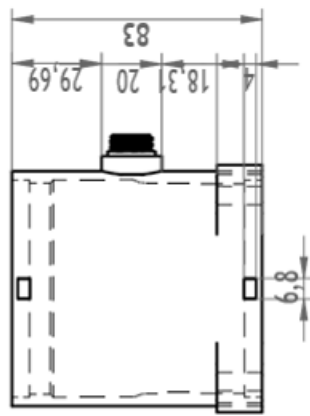
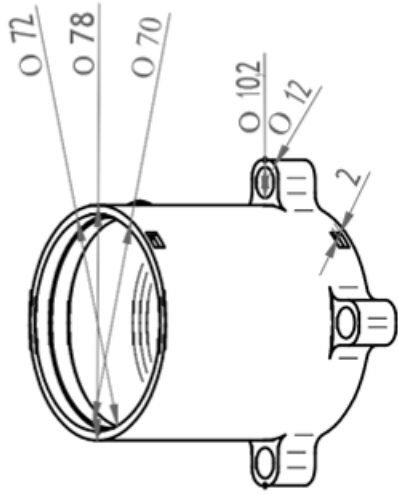
SOLID EDGE ACADEMY



SCALE 2:1

TALTECH SCHOOL OF ENGINEERING		MASTER THESIS	Page/Pages: 1 / 1
Compiler: Tõivo Nerep	15.11.21 Signature and date	M14 Nut	
Supervisor: Kristjan Pütsep	Signature and date		
Department of Electrical Power Engineering and Mechatronics		DESIGN OF VTOL DRONE WITH TANDEM EDF MOTOR AND DEVELOPMENT OF AN ALGORITHM FOR AUTOMATIC STABILIZATION IN HOVER FLIGHT	

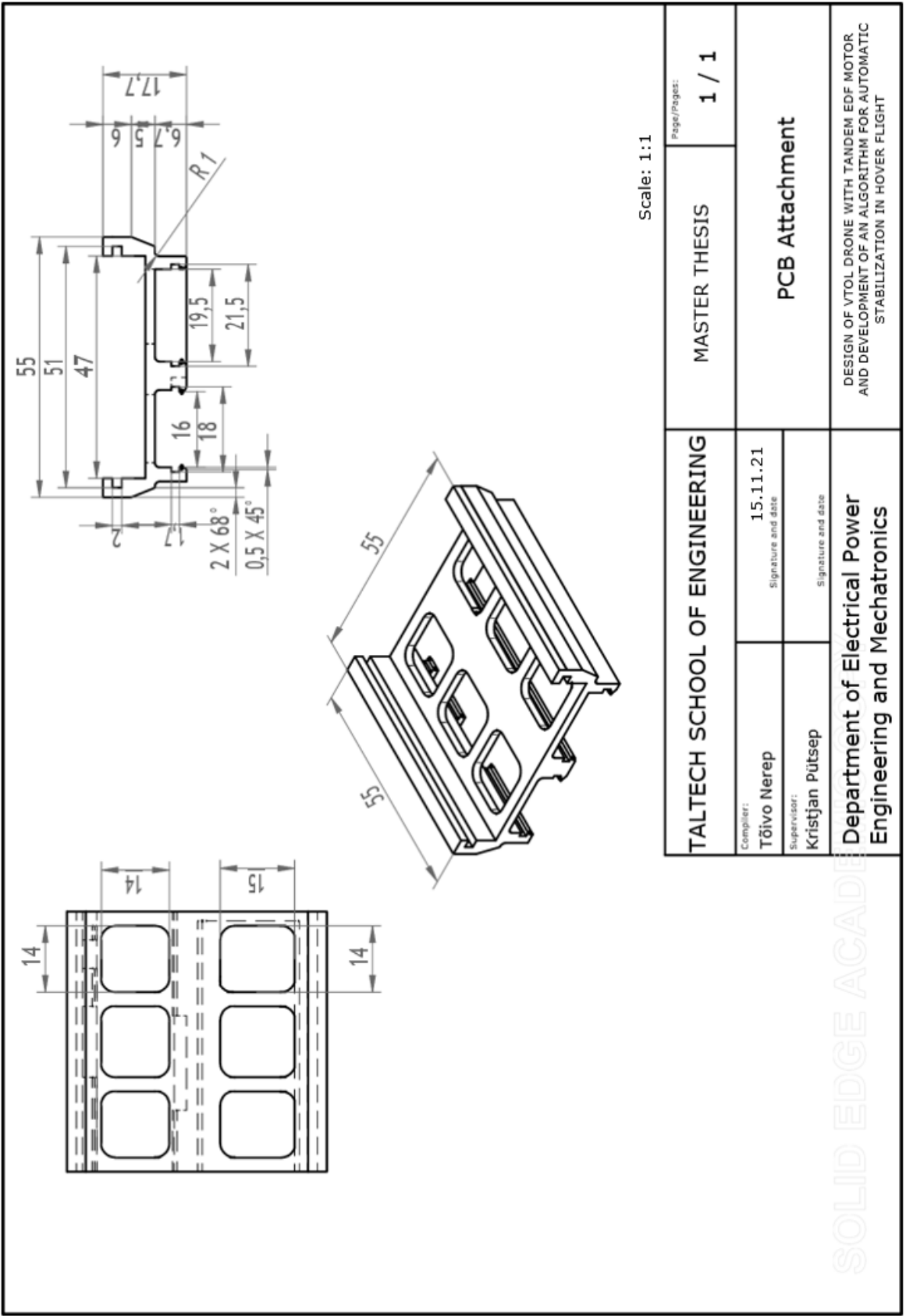
SOLID EDGE ACADEMY

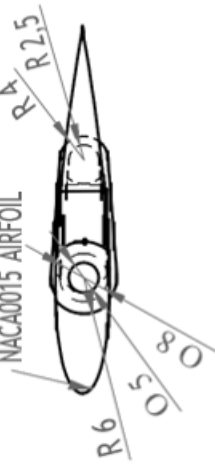
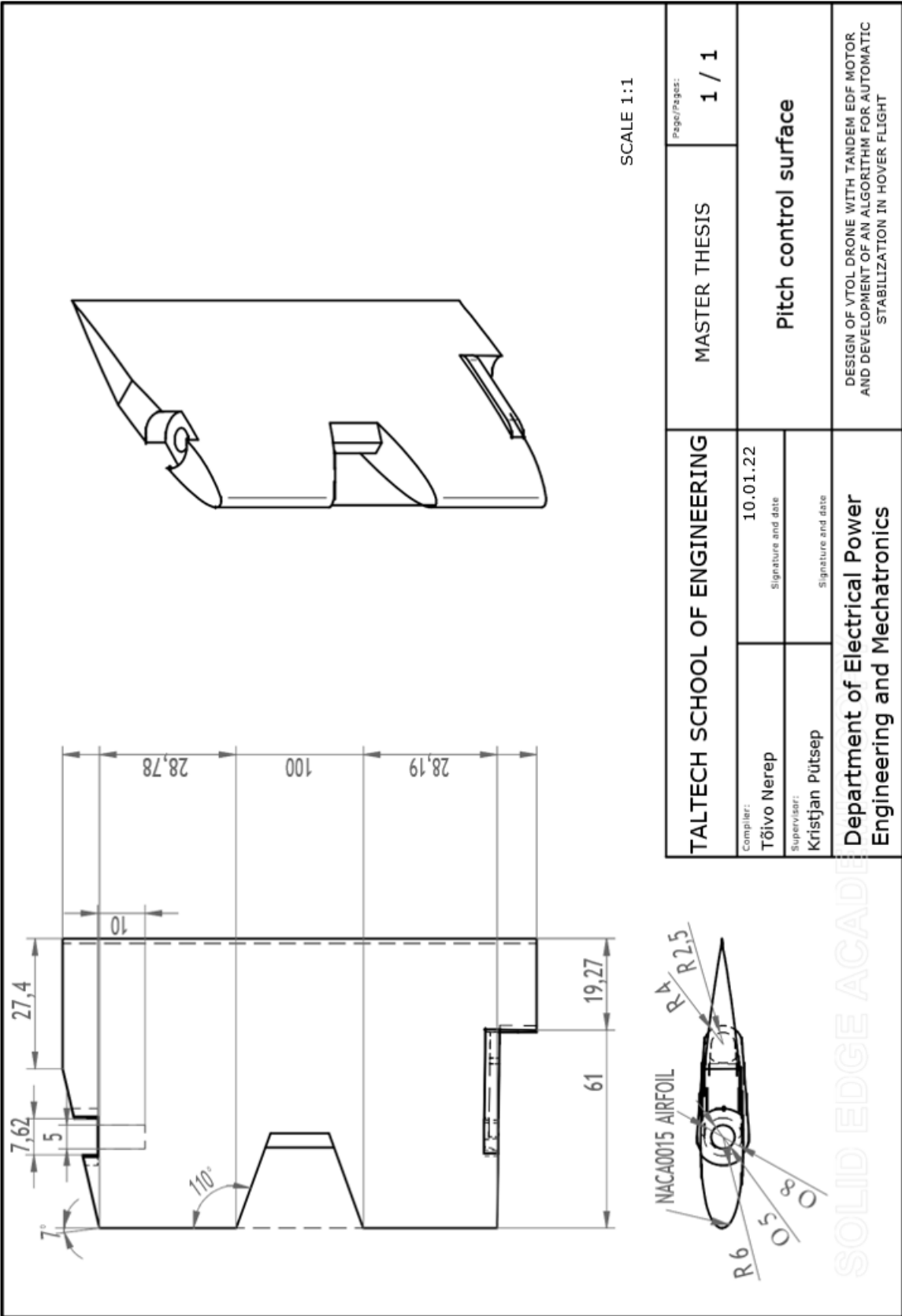


SCALE 1:2

TALTECH SCHOOL OF ENGINEERING		MASTER THESIS	Page/Pages: 1 / 1
Compiler: Tõivo Nerep	15.11.21 Signature and date	Motor housing	
Supervisor: Kristjan Pütsep	Signature and date		
Department of Electrical Power Engineering and Mechatronics		DESIGN OF VTOL DRONE WITH TANDEM EDF MOTOR AND DEVELOPMENT OF AN ALGORITHM FOR AUTOMATIC STABILIZATION IN HOVER FLIGHT	

SOLID EDGE ACADEMY

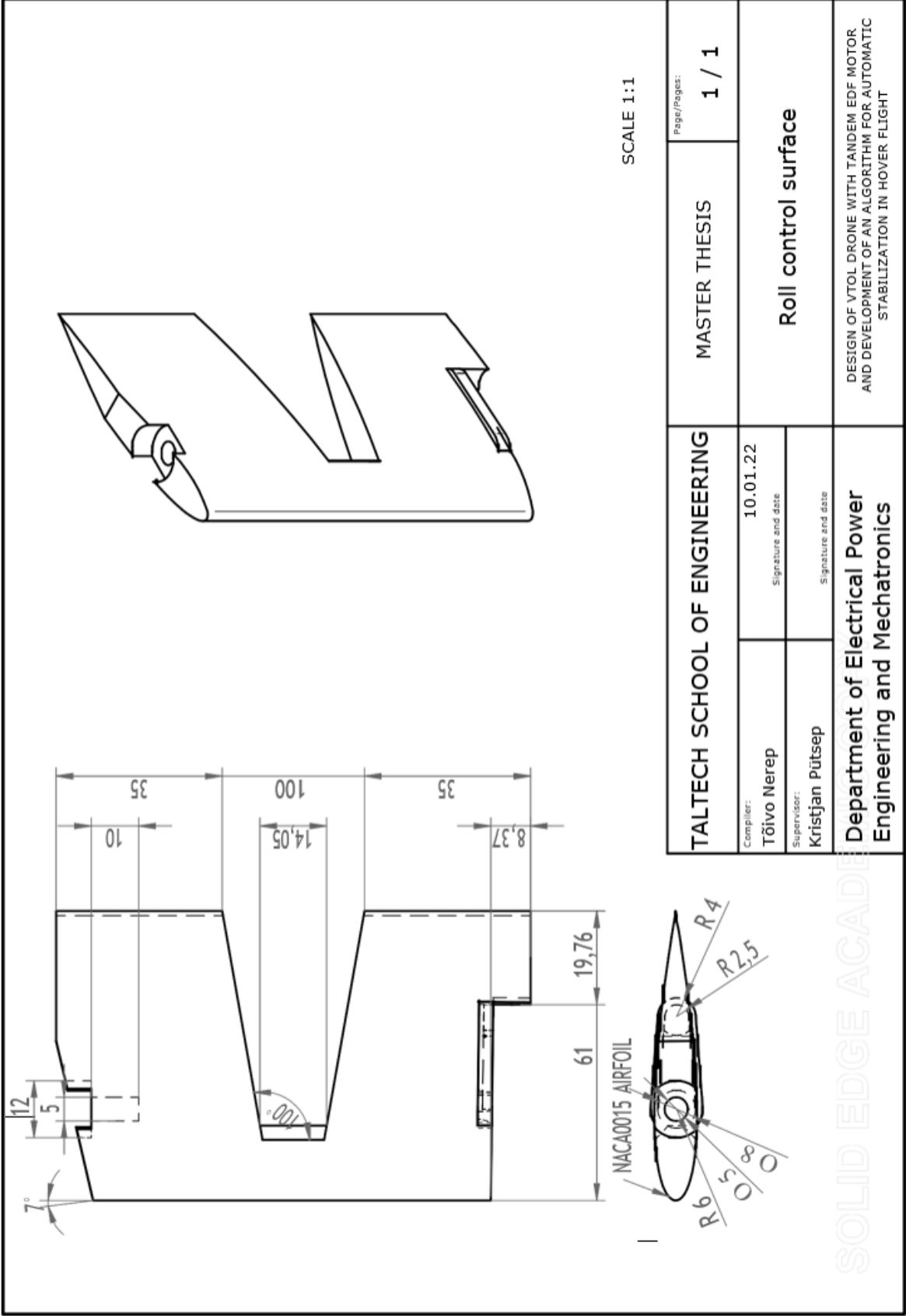




TALTECH SCHOOL OF ENGINEERING	
Compiler: Tõivo Nerep	10.01.22 Signature and date
Supervisor: Kristjan Pütsep	Signature and date

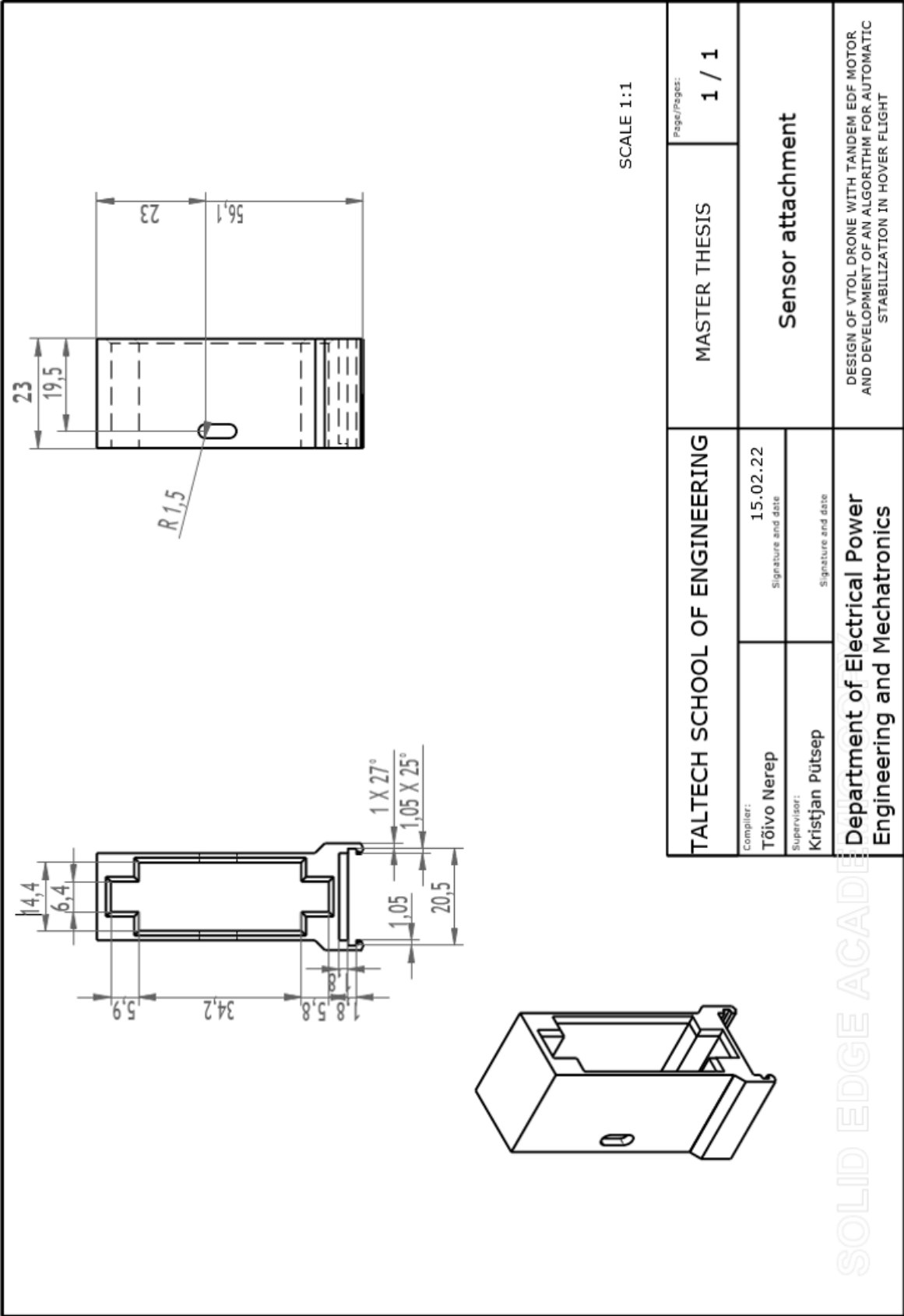
MASTER THESIS	Page/Pages: 1 / 1
Pitch control surface	
DESIGN OF VTOL DRONE WITH TANDEM EDF MOTOR AND DEVELOPMENT OF AN ALGORITHM FOR AUTOMATIC STABILIZATION IN HOVER FLIGHT	

**Department of Electrical Power
Engineering and Mechatronics**



TALTECH SCHOOL OF ENGINEERING		MASTER THESIS	Page/Pages: 1 / 1
Compiler: Tõivo Nerep	10.01.22 Signature and date	Roll control surface	
Supervisor: Kristjan Pütsep	Signature and date		
Department of Electrical Power Engineering and Mechatronics		DESIGN OF VTOL DRONE WITH TANDEM EDF MOTOR AND DEVELOPMENT OF AN ALGORITHM FOR AUTOMATIC STABILIZATION IN HOVER FLIGHT	

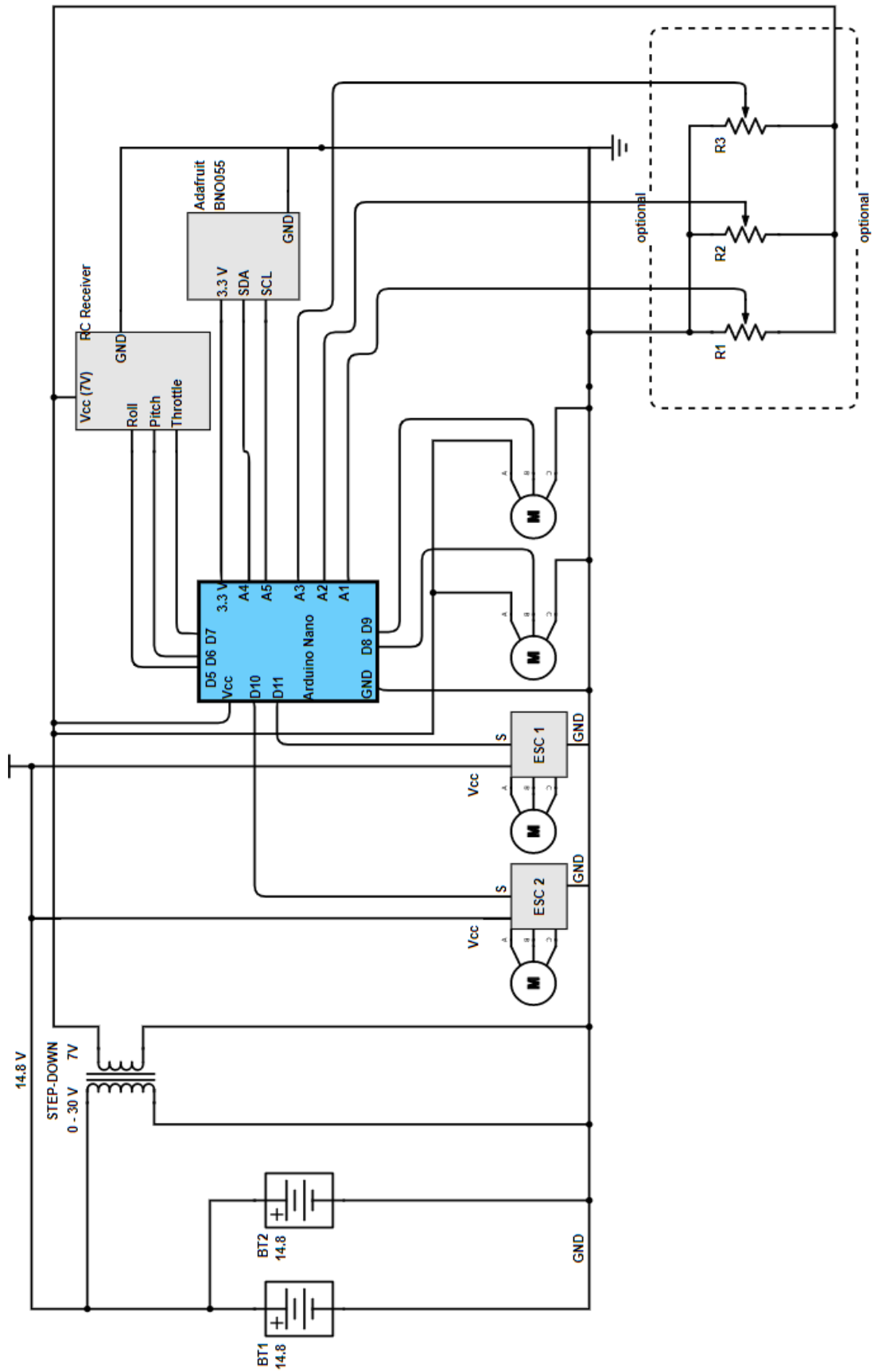
SOLID EDGE ACADEMY



TALTECH SCHOOL OF ENGINEERING		MASTER THESIS	Page/Pages: 1 / 1
Compiler: Tõivo Nerep	15.02.22 Signature and date	Sensor attachment	
Supervisor: Kristjan Pütsep	Signature and date		
Department of Electrical Power Engineering and Mechatronics		DESIGN OF VTOL DRONE WITH TANDEM EDF MOTOR AND DEVELOPMENT OF AN ALGORITHM FOR AUTOMATIC STABILIZATION IN HOVER FLIGHT	

SOLID EDGE ACADEMY

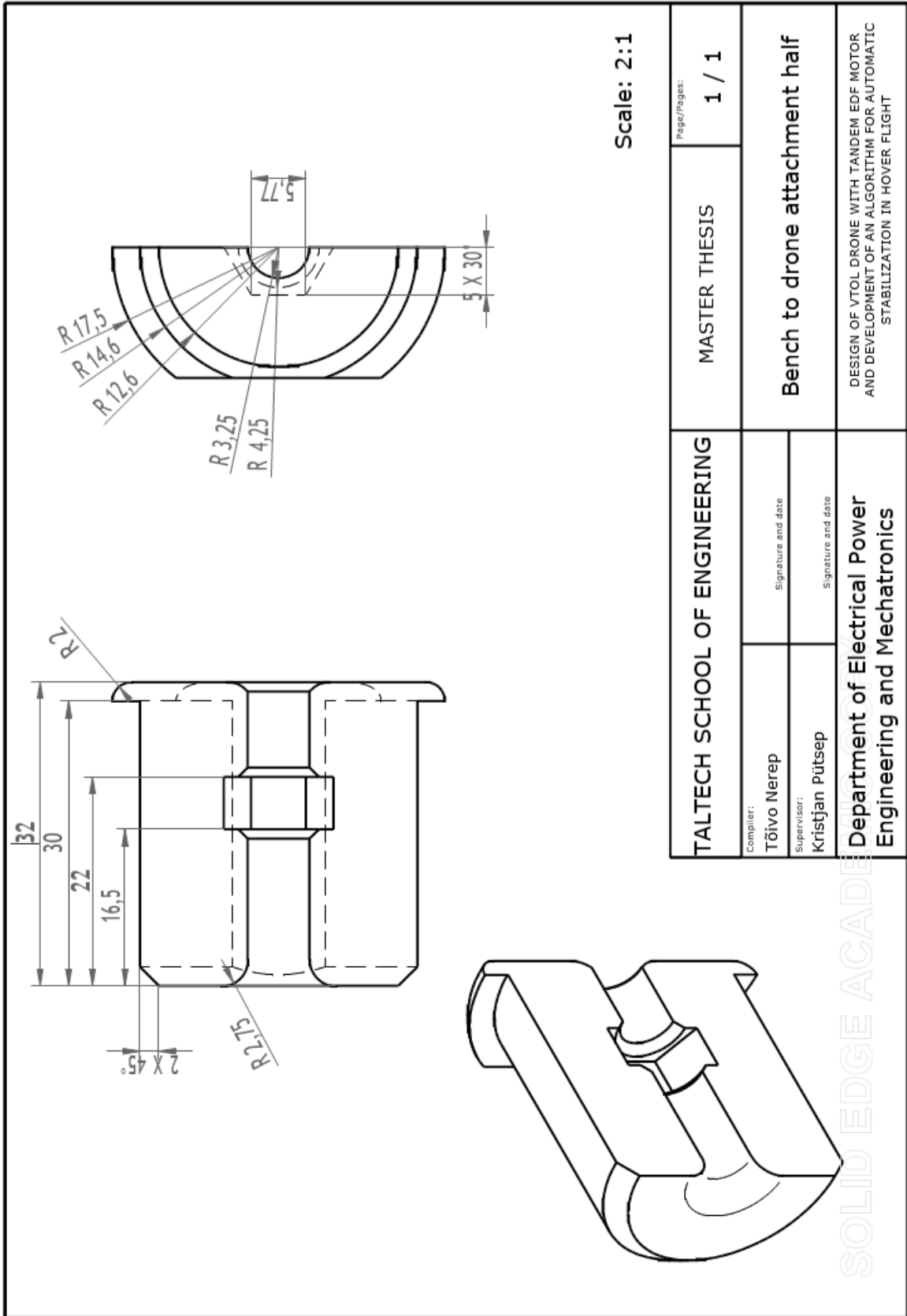
Appendix 2 Prototype electrical scheme



Appendix 3 Technical drawings of test bench

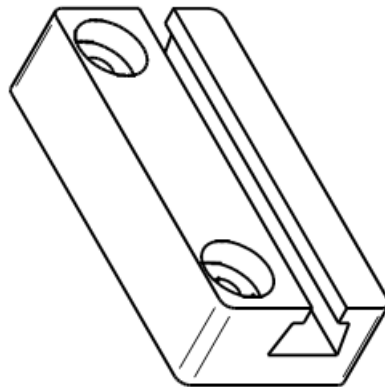
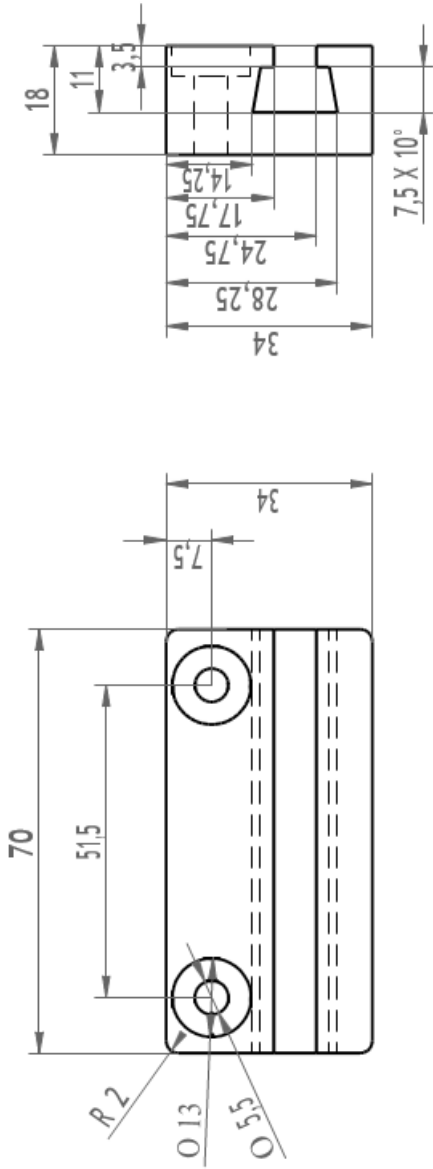
Scale: 2:1

TALTECH SCHOOL OF ENGINEERING		MASTER THESIS		Page/Pages: 1 / 1
Compiler: Tõivo Nerep Signature and date		Pitch bearing attachment half		
Supervisor: Kristjan Pütsep Signature and date		DESIGN OF VTOL DRONE WITH TANDEM EDF MOTOR AND DEVELOPMENT OF AN ALGORITHM FOR AUTOMATIC STABILIZATION IN HOVER FLIGHT		
Department of Electrical Power Engineering and Mechatronics				



TALTECH SCHOOL OF ENGINEERING		MASTER THESIS		Page/Pages: 1 / 1
Compiler: Tõivo Nerep	Signature and date	Bench to drone attachment half		
Supervisor: Kristijan Pütsep	Signature and date			
Department of Electrical Power Engineering and Mechatronics		DESIGN OF VTOL DRONE WITH TANDEM EDF MOTOR AND DEVELOPMENT OF AN ALGORITHM FOR AUTOMATIC STABILIZATION IN HOVER FLIGHT		

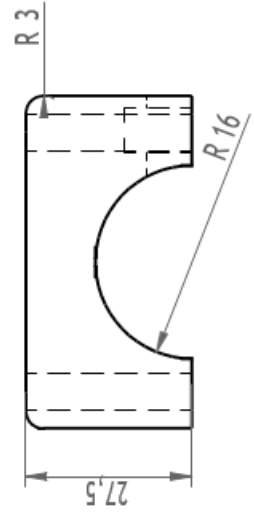
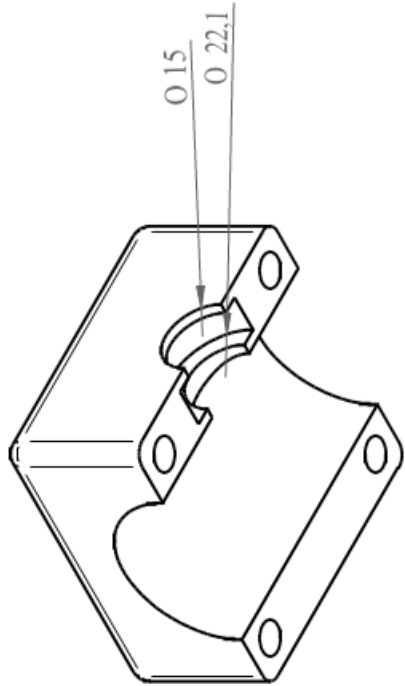
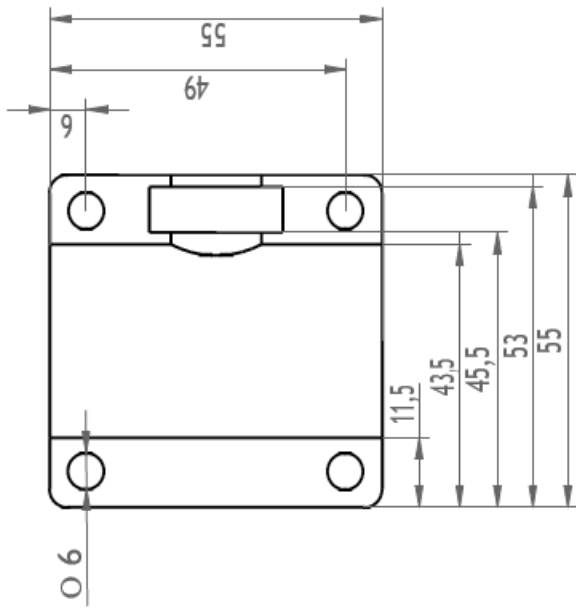
SOLID EDGE ACADEMY



Scale: 1:1

TALTECH SCHOOL OF ENGINEERING		MASTER THESIS	Page/Pages: 1 / 1
Compiler: Tõivo Nerep	Signature and date		Bench to drone attachment slider axial rail
Supervisor: Kristjan Pütsep	Signature and date		
Department of Electrical Power Engineering and Mechatronics		DESIGN OF VTOL DRONE WITH TANDEM EDF MOTOR AND DEVELOPMENT OF AN ALGORITHM FOR AUTOMATIC STABILIZATION IN HOVER FLIGHT	

SOLID EDGE ACADEMY



Scale: 1:1

TALTECH SCHOOL OF ENGINEERING		MASTER THESIS	Page/Pages: 1 / 1
Compiler: Tõivo Nerep	Signature and date	Roll axis bearing housing half	
Supervisor: Kristjan Pütsep	Signature and date		
Department of Electrical Power Engineering and Mechatronics		DESIGN OF VTOL DRONE WITH TANDEM EDF MOTOR AND DEVELOPMENT OF AN ALGORITHM FOR AUTOMATIC STABILIZATION IN HOVER FLIGHT	

SOLID EDGE ACADEMY

Appendix 4 Control algorithm program code

```
// LIBRARY FOR REMOTE CONTROL
#include <PinChangeInterrupt.h>

// LIBRARY FOR SERVOS
#include <Servo.h>

// LIBRARIES FOR SENSOR
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BNO055.h>
#include <utility/imuMaths.h>

// CREATE SENSOR OBJECT //
Adafruit_BNO055 sensor = Adafruit_BNO055(55); // creates sensor object

//CREATE SERVO OBJECTS //
Servo ROLL, PITCH, CW, CCW; // creates servo object for roll and pitch servo

// VARIABLES FOR RC FUNCTION //
const byte channel[3] = {5, 6, 7}; // array of signal pins for receiver channels connected to digital pins
volatile float signal_start[3] = {0, 0, 0}; // variable array where signal start time is stored
volatile float PWM_signal[3] = {0, 0, 0}; // variable array where current PWM signal is stored
volatile float PWM_old[3] = {0, 0, 0}; // variable array where previous PWM signal is stored

// VARIABLES FOR PID FUNCTION //
volatile float k1[3] = {0, 0, 0}; // variable array where P - gain parameters are stored
volatile float k2[3] = {0, 0, 0}; // variable array where I - gain parameters are stored
volatile float k3[3] = {0, 0, 0}; // variable array where D - gain parameters are stored
volatile float Pot_Val = 0;
volatile float Pot_New = 0;

volatile float Target[3] = {0, 0, 0}; // variable array where Setpoint values for PID calculations are stored
volatile float Actual[3] = {0, 0, 0}; // variable array where actual position values are stored
```

```

volatile float Error[3]           = {0, 0, 0};    // variable array where current error values are stored
volatile float ErrorOld[3]        = {0, 0, 0};    // variable array where previous error values are stored
volatile float ErrorDif[3]        = {0, 0, 0};    // variable array where difference between current and previous
error is stored
volatile float Slope[3]           = {0, 0, 0};    // variable array where error difference divided by time
difference is stored
volatile float steadyerrorArea[3] = {0, 0, 0};    // variable array where steady error variable values are stored
volatile float PWM[4]            = {0, 0, 0, 0};  // variable array where servo command signals are stored
volatile float milliPrevious;      // variable where previous measurement time is stored
volatile float milliCurrent;      // variable where current measurement time is stored
volatile float dt;                // variable where time between two measurement points is stored

void setup() {
  Serial.begin(115200);            // opens serial port with specified baud rate

  // SENSOR SETUP //
  if (!sensor.begin()) {          // checks if sensor is working
    Serial.print("error 2 - Sensor not detected"); // feedback to serial port if sensor not detected
    while (1);                    // continue if sensor is working
  }
  delay(1000);
  sensor.setExtCrystalUse(true);  // command to adafruit to use external sensor which is more
accurate than internal

  // PID SETUP //
  milliCurrent = millis();        // stores number of milliseconds passed since program started

  // RECEIVER SETUP //
  pinMode(channel[0], INPUT);     // sets digital pin 5 as input
  pinMode(channel[1], INPUT);     // sets digital pin 6 as input
  pinMode(channel[2], INPUT);     // sets digital pin 7 as input

  attachPCINT(digitalPinToPCINT(channel[0]), channel0, CHANGE); // Attach Pin change interrupt to pin 5
  attachPCINT(digitalPinToPCINT(channel[1]), channel1, CHANGE); // Attach Pin change interrupt to pin 6
  attachPCINT(digitalPinToPCINT(channel[2]), channel2, CHANGE); // Attach Pin change interrupt to pin 7

```

```

// SERVO SETUP //
CCW.attach(10, 900, 2000); // allocates digital pin 10 to CCW serco object
CW.attach(11, 900, 2000); // allocates digital pin 11 to CW servo object
ROLL.attach(8); // allocates digital pin 8 to Roll serco object
PITCH.attach(9); // allocates digital pin 9 to Pitch servo object
//set servos to initial position
CCW.writeMicroseconds(900); // to ensure that motors will not spin during setup
CW.writeMicroseconds(900); // to ensure that motors will not spin during setup
delay(3000); // wait 3 seconds to be sure that ESCs received commands
ROLL.write(80); // to calibrate and set centre point for roll servo
PITCH.write(80); // to calibrate and set centre point for pitch servo
delay(2000); // wait 2 seconds to allow servos to stabilise and reach to positions
}

/*
void displayCalStatus()
{
  /* Get the four calibration values (0..3) */
  /* Any sensor data reporting 0 should be ignored, */
  /* 3 means 'fully calibrated' */ /*
  uint8_t system, gyro, accel, mag;
  system = gyro = accel = mag = 0;
  sensor.getCalibration(&system, &gyro, &accel, &mag);

  /* The data should be ignored until the system calibration is > 0 *//*
  Serial.print("\t");
  if (!system)
  {
    Serial.print("! ");
  }

  /* Display the individual values *//*
  Serial.print("Sys:");
  Serial.print(system, DEC);
  Serial.print(" G:");
  Serial.print(gyro, DEC);
  Serial.print(" A:");
  Serial.print(accel, DEC);
}

```

```

    Serial.print(" M:");
    Serial.println(mag, DEC);
}*/

// FUNCTION TO ADJUST P AND D GAIN PARAMETERS DURING LIVE TRIALS //
// This function uses remote control channels 0 and 1 (trim function only) to manipulate gain parameters //
// function takes as input 1 or 2, which set if pitch or roll gain parameters are under manipulation //
void PD(byte pin){

    Pot_Val = analogRead(A1); // Read potentiometer position from analog pin and store it to memory
variable
    Pot_New = map(Pot_Val, 0, 1023, 0, 100); // map potentiometer range to new value 0 to 100
    k1[pin] = Pot_New/100; // divide range by 100 to provide floating point values 0 to 1 for gain
value
}

// FUNCTION THAT MEASURES SIGNAL LENGHT AND TAKES CARE OF INTERRUPTS //
// This function uses pinchange interrupt library example code to trigger //
// interrupt when signal is rising or falling, stores rising time and //
// subtracts it from falling time to get lenght of signal //
void processPin(byte pin) {

    PWM_old[pin] = PWM_signal[pin]; // sets previous PWM_signal value to old value
    // following sets local variable to each receiver pin (digital pin 5, digital pin 6, digital pin 7)
    uint8_t trigger = getPinChangeInterruptTrigger(digitalPinToPCINT(channel[pin]));

    if (trigger == RISING) { // trigger if signal is rising
        signal_start[pin] = micros(); // stores rising time value
    } else if (trigger == FALLING) { // trigger if signal is falling
        PWM_signal[pin] = micros() - signal_start[pin]; // calculates PWM signal lenght by subtracting falling
time from rising time
    }
    PWM_signal[pin] = .9 * PWM_old[pin] + .1 * PWM_signal[pin]; // Low pass filter to provide more stable signal
    if (PWM_signal[2] > 1600 && PWM_signal[2] < 2100) { // sets condition that if result is more than 2000
        PWM_signal[2] = 1600; // values will not rise over 2000
    } else if (PWM_signal[2] > 2100) { // sets condition that if result is less than 900

```

```

    PWM_signal[2] = 1000;           // values will not decrease less than 900
} else if (PWM_signal[2] < 1000) { // sets condition that if result is less than 900
    PWM_signal[2] = 1000;           // values will not decrease less than 900
}
}

// FOLLOWING THREE FUNCTIONS CALL OUT PREVIOUS FUNCTION WITH EACH PIN //
void channel0() {
    processPin(0);
}
void channel1() {
    processPin(1);
}
void channel2() {
    processPin(2);
}

// SENSOR FUNCTION //
void sensorfunction() {
    sensors_event_t measurement; // Local variable where measurement data is stored (data type for
Adafruit Unified Sensor Driver) // Get new sensor data from sensor
    sensor.getEvent(&measurement);

    Actual[1] = round(measurement.orientation.z); // Store pitch angle to position 1 (second data row) in variable
array "Actual"
    Actual[2] = round(measurement.orientation.y); // Store roll angle to position 2 (third data row) in variable
array "Actual"

    milliPrevious = milliCurrent; // Sets the previous measurement time
    milliCurrent = millis(); // gets current measurement time
    dt = milliCurrent - milliPrevious; // calculates time difference between previous and current
    measurement
}

```

```

// PID CALCULATION FUNCTION //
void PIDfunction() {

    /* // Pitch error PID equation //
    ErrorOld[1] = Error[1]; // Sets previous error value
as old value
    Error[1] = Target[1] - Actual[1]; // Calculates new error
value (Proportional parameter)
    ErrorDif[1] = Error[1] - ErrorOld[1]; // Calculates difference
between previous and current error
    Slope[1] = ErrorDif[1] / dt; // calculates slope
(derivative parameter) or rate in change of error
    steadyerrorArea[1] = steadyerrorArea[1] + Error[1] * dt; // calculates sum of error
over time (integral parameter)

    PWM[1] = PWM[1] + k1[1] * Error[1] + k3[1] * Slope[1] + k2[1] * steadyerrorArea[1]; // signal correction with
PID equation
    if (PWM[1] > 1000) { // sets condition that if
result is more than 1000
        PWM[1] = 1000; // values will not rise over
1000
    } else if (PWM[1] < -1000) { // sets condition that if
result is less than -1000
        PWM[1] = -1000; // values will not decrease
less than -1000
    }
    PWM[1] = map(PWM[1], -1000, 1000, 65, 95); // converts equation result
to reasonable value for servo command

    // Roll error PID equation //
    ErrorOld[2] = Error[2]; // Sets previous error
value as old value
    Error[2] = Target[2] - Actual[2]; // Calculates new error
value (Proportional parameter)
    ErrorDif[2] = Error[2] - ErrorOld[2]; // Calculates difference
between previous and current error
    Slope[2] = ErrorDif[2] / dt; // calculates slope
(derivative parameter) or rate in change of error

```



```

    steadyerrorArea[2] = steadyerrorArea[2] + Error[2] * dt;           // calculates sum of error
over time (integral parameter)

    PWM[2] = PWM[2] - k1[2] * Error[2] + k3[2] * Slope[2] + k2[2] * steadyerrorArea[2]; // signal correction with
PID equation
    if (PWM[2] > 1000) {                                           // sets condition that if
result is more than 1000
        PWM[2] = 1000;                                           // values will not rise over
1000
    } else if (PWM[2] < -1000) {                                   // sets condition that if
result is less than -1000
        PWM[2] = -1000;                                          // values will not decrease
less than -1000
    }
    PWM[2] = map(PWM[2], -1000, 1000, 65, 95);                   // converts equation result
to reasonable value for servo command
}

// SERVO FUNCTION //
// This function communicates with servo actuators //
void servofunction() {
    CCW.writeMicroseconds(PWM_signal[2]); // Command to CCW servo
    delay(15); // delay to allow servo to stabilize
    CW.writeMicroseconds(PWM_signal[2]); // Command to CW servo
    delay(15); // delay to allow servo to stabilize
    // ROLL.write(PWM[2]); // Command to roll servo
    // delay(15); // delay to allow servo to stabilize
    PITCH.write(PWM[1]); // Command to pitch servo
    delay(15); // delay to allow servo to stabilize
}

```

```

// THIS IS THE MAIN LOOP THAT IS PERFORMED OVER AND OVER //
void loop() {

    //displayCalStatus();           // check calibration
    sensorfunction();             // Call out sensor function
    PIDfunction();                // Call out PID function
    servofunction();              // Call out servo function
    PD(1);                         // Activate adjustment for Roll gain parameters (deactivated when commented
out) Only used for PID adjustments
    // PD(0);                       // Activate adjustment for Pitch gain parameters (deactivated when commented
out) Only used for PID adjustments

    // SERIAL PRINTS ARE USED TO GET FEEDBACK VIA SERIAL PORT //
    // parameters to be sent to serial are ment to be changed in accordance with the monitoring needs//
    // After sucessful tuning process, these lines should be commented out//
    Serial.print("1");
    Serial.print(",");
    Serial.print(PWM[1]);
    Serial.print(",");
    Serial.print(PWM_signal[2]);
    Serial.print(",");
    Serial.print(k1[1]);
    Serial.print(",");
    Serial.print(Actual[1]);
    Serial.print(",");
    Serial.print(-Error[1]);
    Serial.print(",");
    Serial.println(Target[1]);

}

```

Appendix 5 Global variable array descriptive table

A5 Table 1.1 Global variable array table

PWM_signal[]	Position	Description
	0	Pitch RC signal
	1	Roll RC signal
	2	Throttle RC signal
Channel[]		
	0	digital pin 5 for RC signal input
	1	digital pin 6 for RC signal input
	2	digital pin 7 for RC signal input
Signal_start[]		
	0	Pitch signal start time
	1	Roll signal start time
	2	Throttle signal start time
PWM_signal[]		
	0	Current Pitch signal
	1	Current Roll signal
	2	Current throttle signal
PWM_old[]		
	0	Pitch old value
	1	Roll old value
	2	Throttle old value
k1		
	0	Pitch Proportional gain
	1	Roll Proportional gain
	2	yaw proportional gain
k2		
	0	Pitch Integral gain
	1	Roll Integral gain
	2	yaw integral gain
k3		
	0	Pitch Derivative gain
	1	Roll Derivative gain
	2	Yaw Derivative gain
Target[]		
	0	Pitch target value
	1	Roll target value
	2	Yaw target value
Actual[]		
	0	Pitch angle from sensor
	1	Roll angle from sensor
	2	Yaw angle from sensor

A5 Table 1.1 Continued

PWM_signal[]	Position	Description
Error[]		
	0	Current Pitch angle difference from Target
	1	Current Roll angle difference from Target
	2	Current Yaw angle difference from Target
ErrorOld[]		
	0	Previous Pitch error
	1	Previous Roll error
	2	Previous Yaw error
ErrorDif[]		
	0	Difference in pitch Error and ErrorOld
	1	Difference in roll Error and ErrorOld
	2	Difference in yaw Error and ErrorOld
Slope		
	0	Pitch ErrorDif divided by time difference
	1	Roll ErrorDif divided by time difference
	2	Yaw ErrorDif divided by time difference
steadyerrorArea[]		
	0	Pitch steady error
	1	Roll steady error
	2	Yaw steady error
PWM		
	0	Pitch servo command
	1	Roll servo command
	2	EDF 1 motor command
	3	EDF 2 motor command