

Bachelor Project



**Czech
Technical
University
in Prague**

F3

**Faculty of Electrical Engineering
Department of Measurement**

Design and realization of a Convertiplane

**Supervisor: Ing. Martin Šipoš Ph.D.
January 2024**

I. Personal and study details

Student's name: **Broul Marek**

Personal ID number: **507656**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Measurement**

Study program: **Cybernetics and Robotics**

II. Bachelor's thesis details

Bachelor's thesis title in English:

Design and realization of Convertiplane

Bachelor's thesis title in Czech:

Návrh a realizace konvertoplánu

Guidelines:

Be familiar with VTOL and convertible UAVs . Design and realize the model of convertible UAV that will be capable of vertical take-off, transition to horizontal flight mode and subsequent vertical landing. Design and implementation will include the central control unit, which will control the UAV and further enable the measurement of data from sensors and systems: GNSS, barometric altimeter, inertial measurement unit. The measured data will be stored on an SD card for subsequent offline data processing. The realized aircraft should be evaluated under real flight tests and verified the measured data. Implement a demo firmware that will tilt the rotors and allow the propellers to spin at lowest possible RPM.

Bibliography / sources:

- [1] Barnhart R. K., Hottman S. B., Marshall D.M., Shappee E.: Introduction to Unmanned Aerial Systems, CRC Press, 2012, ISBN: 978-1-4398-3520-3
[2] Nonami K., Kendoul F., Suzuki S., Wang W., Nakazawa D.: Autonomous Flying Robots: Unmanned Aerial Vehicles and Micro Aerial Vehicles, Springer, 2010, ISBN: 978-4-431-53856-1
[3] Nelson, Flight stability and automatic control, Springer, 2003, ISBN: 978-0070462731. Stevens, Lewis, Aircraft simulation and control. Wiley, 2003, ISBN: 978-0471371458.

Name and workplace of bachelor's thesis supervisor:

Ing. Martin Šipoš, Ph.D. katedra m ení FEL (13138)

Name and workplace of second bachelor's thesis supervisor or consultant:

Date of bachelor's thesis assignment: **05.02.2024** Deadline for bachelor thesis submission: **24.05.2024**

Assignment valid until:

by the end of summer semester 2024/2025

Ing. Martin Šipoš, Ph.D.
Supervisor's signature

Head of department's signature

prof. Mgr. Petr Páta, Ph.D.
Dean's signature

III. Assignment receipt

The student acknowledges that the bachelor's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor's thesis, the author must state the names of consultants and include a list of references.

Date of assignment receipt

Student's signature

Acknowledgements

I would like to thank my supervisor, Ing. Martin Šipoš, Ph.D., for his guidance and support throughout this project.

A special thanks to my family and my girlfriend for their constant encouragement and support during my studies.

Finally, I would like to thank my classmates for their help and the enjoyable moments we shared during the last 3 years.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

In Prague, on the 23rd of May 2024,
Marek Broul



Abstract

This thesis presents the design and realization of a model of a tilt-rotor convertiplane. The primary objective was to construct a functional model of a VTOL-capable aircraft. To achieve the primary goal, I have constructed a lightweight airframe with functional rotor tilting and elevator mechanisms to control the aircraft.

A flight controller was designed using a custom-designed PCB, microcontroller, and IMU modules. In this thesis, I also derive a mathematical model for this type of tilt-rotor aircraft. This model is linearised and used to design nested PI and PID controllers to achieve stable dynamics of the model. The controller was tested during several experiments with partial success.

A simple aerodynamic model is derived to simulate the system dynamics in horizontal flight using CFD simulations. Based on the aerodynamic model, this thesis proposes closed-loop control, which is validated by simulations.

Keywords: convertiplane, control theory, modeling of dynamic system, aerodynamics, VTOL, CFD

Supervisor: Ing. Martin Šipoš Ph.D.
Fakulta elektrotechnická,
Technická 1902/2,
16200 Praha 6

Abstrakt

Tato bakalářská práce představuje návrh a realizaci modelu konvertoplánu s naklápěcími rotory. Primárním cílem bylo postavit funkční model letounu schopného vertikálního vzletu a přistání (VTOL).

K dosažení hlavního cíle jsem postavil lehký drak letadla s funkčními mechanismy naklápění rotorů a výškovky pro ovládání letadla. Letový kontrolér byl navržen a vytvořen z desky plošného spoje, mikrokontroléru a IMU modulu.

V této práci jsem také odvodil matematický model tohoto typu letounu, který je po linearizaci použit k návrhu vnořených PI a PID regulátorů k dosažení stabilní dynamiky modelu.

Regulátor byl testován během několika experimentů s částečným úspěchem, ale přechod do horizontálního letu nebyl dosažen.

Pro simulaci dynamiky systému v horizontálním letu, jsem odvodil jednoduchý aerodynamický model pomocí CFD simulací. Na základě aerodynamického modelu, tato práce navrhuje řízení v uzavřené smyčce, které je ověřeno pouze simulačně.

Klíčová slova: konvertoplán, teorie řízení, modelování dynamických systémů, aerodynamika, VTOL, CFD

Překlad názvu: Realizace modelu konvertoplánu

Contents

1 Introduction	1	7 Control	49
1.1 Structure	1	7.1 Controller design methods	49
1.2 History of development	2	7.1.1 Pole placement	50
1.3 Present and Future development	3	7.2 Vertical mode	52
1.4 Similar designs of VTOL capable aircraft	4	7.2.1 Roll	52
1.4.1 Tiltwing	4	7.2.2 Pitch	55
1.4.2 Jet-powered thrust vector propulsion	5	7.2.3 Yaw	56
2 Construction	7	7.3 Horizontal mode	57
2.1 Fuselage	7	7.3.1 Roll	57
2.2 Wings	9	7.3.2 Pitch	58
2.3 Rotor tilting mechanism	10	7.3.3 Yaw	58
2.4 Stabilizer	11	7.4 Transition mode	59
3 Electronics	13	8 Experiments	61
3.1 Powertrain	13	8.1 Tests in suspension	61
3.1.1 Motor linearization	14	8.2 Free flight	62
3.2 Flight controller	16	9 Conclusion	67
3.2.1 MCU	16	9.1 Future	67
3.2.2 Sensors	17	9.1.1 Further improvements	67
3.2.3 PCB	19	9.1.2 Potential use-cases	68
3.2.4 Radio Control	20	Bibliography	69
4 Flight controller software	23	A Flight controller schematics	73
4.1 Sensor reading	23		
4.1.1 Calibration	23		
4.1.2 GY-87	24		
4.1.3 NEO-6M	24		
4.1.4 MicroSD card module	24		
4.2 Wireless data transfer	26		
5 Inertial measurement unit	27		
5.1 Kalman Filter	29		
5.2 Accelerometer, Gyroscope data fusion	30		
5.2.1 Results	31		
6 Mathematical model	33		
6.1 Coordinate systems	33		
6.1.1 Transformations	34		
6.2 Vertical flight	36		
6.2.1 Translation	36		
6.2.2 Rotation	36		
6.2.3 Linearization	38		
6.3 Horizontal flight	42		
6.3.1 Aerodynamics	43		
6.3.2 Linearization	46		

Figures

1.1 Transcendental Model 1 G[5]	2	5.3 Kalman filter demonstration of gyroscope drift elimination	32
1.2 Bell XV-3[6]	2	5.4 Kalman filter demonstration of noise reduction	32
1.3 Bell Boeing V-22 Osprey[9]	2	6.1 Simulink 3D environment	33
1.4 Leonardo AW609[11]	3	6.2 Coordinate systems	34
1.5 Bell V-280 Valor[12]	3	6.3 Torque — pitch	37
1.6 Vertol VZ-2[18]	4	6.4 CFD of a simplified 3D model - Velocity magnitude	43
1.7 Lockheed Martin F-35B[25]	5	6.5 Deflected aileron -15° - Static pressure	45
2.1 Fuselage	7	6.6 Coefficient of lift as a function of aileron deflection	45
2.2 Constructed aircraft	8	6.7 Coefficient of lift as a function of elevator deflection	46
2.3 Open fuselage	8	7.1 Nested controller	49
2.4 Right wing	9	7.2 Single controller	49
2.5 Coefficient of lift to AoA function for Reynolds number 50,000[26]	9	7.3 Roll rate system response in vertical mode	53
2.6 Tilting mechanism model	10	7.4 Roll system response in vertical mode	55
2.7 Tilting mechanism	10	7.5 Yaw rate system response in vertical mode	56
2.8 Rotational joint	10	7.6 Roll rate system response in horizontal mode	57
2.9 Stabilizer model	11	7.7 Pitch rate system response in horizontal mode	58
2.10 Elevator - rotational joint	11	8.1 Indoor experiment	61
2.11 Elevator linkage	11	8.2 Indoor experiment - yaw control	63
2.12 Stabilizer	11	8.3 Flight 1 - yaw rate	63
3.1 Force characteristics of the left motor	14	8.4 Flight 2 - pitch	64
3.2 Force characteristics of the right motor	14	8.5 Flight 3 - pitch rate	65
3.3 Power supply schematic	15	8.6 Flight 3 - outdoor testing	65
3.4 ESP32 Microcontroller	16		
3.5 ESP32 schematic	16		
3.6 GY-87	17		
3.7 microSD card module	17		
3.8 NEO-6M	18		
3.9 PCB top board	19		
3.10 PCB bottom board	19		
3.11 RC transmitter - Hitec Optic 6 Sport[31]	20		
3.12 Assembled flight controller	21		
4.1 Simulink PID tuning block scheme	26		
5.1 Accelerometer and Gyroscope orientation estimation - sensor at rest	28		
5.2 Accelerometer and Gyroscope orientation estimation - induced vibrations	29		

Tables

3.1 Linear approximation coefficients of motor force function	14
3.2 NMEA-0183 Reference manual[30]	18
6.1 Coefficient of lift as a function of aileron deflection	45
6.2 Coefficient of lift as a function of elevator deflection	46

Chapter 1

Introduction

This work focuses on the design and realization of a model of a tilt-rotor convertiplane. A convertiplane is a type of aircraft heavier than air, which uses rotor power for vertical take-off and landing (VTOL), with the ability to change its configuration to transition to a conventional horizontal flight using fixed-wing lift[1]. A tilt-rotor convertiplane uses tiltable rotors angled upwards, mounted at the end of a fixed wing to produce lift in the vertical phase of the flight. After the transition to horizontal flight, the rotors are tilted forward and are used for propulsion only, and the fixed wing provides the lift. Compared to a fixed-wing airplane, a convertiplane can operate without a runway and has higher maneuverability at lower speeds. There are a lot of tilt-rotor aircraft designs. Some use 4 or more rotors in a quadcopter configuration and tilt them only for transition to horizontal flight, and stability and control are achieved using differential thrust. Other models, which are more true to real-life tilt-rotor aircraft, use rotors with cyclic and collective control to stabilize the aircraft, and the tilt of the rotors is also used only to transition between flight modes.

1.1 Structure

This chapter summarises VTOL aircraft's past, present, and future development.

Chapter 2 talks about the design of the physical model, and the used electronics are described in chapter 3.

In chapter 4, the code running onboard the flight controller is discussed and the implementation of an IMU is shown in chapter 5.

In chapter 6, a mathematical model of the dynamics of this tilt-rotor aircraft is derived and later used in chapter 7 to design PI and PID regulators to achieve stable flight.

Chapter 8 shows the experiments taken to tune the controllers and attempts of flight, and in chapter 9, future development and potential applications are discussed. Appendix A displays the schematics of the flight controller.

1.2 History of development

The idea of using a tilt-rotor aircraft for VTOL capabilities originated in the 1930s, but it took another 20 years until the first working prototypes were constructed. Transcendental Model 1-G took flight in 1954 but never achieved a complete transition. In 1955, the aircraft suffered a control system failure, resulting in a crash[2, 3]. The program was later abandoned in favor of the Bell XV-3, which was the first tilt-rotor convertiplane to achieve a complete transition to and from a horizontal flight in 1958[2, 4].



Figure 1.1: Transcendental Model 1 G[5]



Figure 1.2: Bell XV-3[6]

In 1972, Bell Helicopter Textron started developing the XV-15 to research military and civil applications of the tilt-rotor design, which first flew in 1977[2]. Later, in 1981, Bell, in collaboration with Boeing Helicopters, began development of the V-22 Osprey, a military tilt-rotor aircraft for the U.S. Air Force and the U.S. Marine Corps to replace the Boeing CH-46 Sea Knight. The first working prototypes of the Osprey flew in 1989, both in the vertical and horizontal modes, but it took another 18 years before the Osprey was introduced into service in 2007. Today, the aircraft is operated by the U.S. Air Force, U.S. Marine Corps, U.S. Navy, and Japan Ground Self-Defense Force and is currently the only tilt-rotor convertiplane in operation with more than 400 manufactured[7, 8].



Figure 1.3: Bell Boeing V-22 Osprey[9]

1.3 Present and Future development

Another tilt-rotor aircraft derived from the XV-15, the Leonardo AW609, took its first flight in 2003 and is currently in the final stage of certification testing with the expected introduction into the civil aviation market later in 2024[2, 10].



Figure 1.4: Leonardo AW609[11]



Figure 1.5: Bell V-280 Valor[12]

In 2013, Bell Helicopters unveiled the V-280 Valor tilt-rotor aircraft as a contender in the Future Vertical Lift program, which aims to develop a family of helicopters for the United States Armed Forces to replace the UH-60 Black Hawk, AH-64 Apache, CH-47 Chinook and OH-58 Kiowa helicopters. The Valor demonstrator first took flight in 2017 and was chosen in 2022 as the winner to replace the UH-60. The aircraft is expected to be introduced into operation in 2030[13, 14].

In 2021, Bell announced a High-Speed Vertical Take-Off and Landing (HSV-TOL) concept, a stopped rotor convertiplane. The concept uses two rotors in the same configuration as a tilt-rotor with the possibility to stop and stow the rotors after the transition to horizontal flight. After the transition, a jet engine located in the tail propels the aircraft[15].

On February 8th, Bell showcased a successful demonstration of the transition from propeller to jet-powered propulsion[16].

1.4 Similar designs of VTOL capable aircraft

1.4.1 Tiltwing

Another type of VTOL-capable aircraft is a tilt-wing design, which can rotate the entire wing with the rotor. This type is generally more efficient in vertical flight than tilt-rotors because the thrust of the rotors in the vertical position encounters less interference from the wing. Furthermore, unlike a tilt-rotor design, a tilt-wing aircraft can begin the transition to horizontal flight at zero forward speed. A notable design of this type is the Vertol VZ-2, the first tilt-wing to take flight in vertical mode in 1957. A year later, in 1958, the aircraft achieved its first transition to horizontal mode[17].



Figure 1.6: Vertol VZ-2[18]

Between 2016 and 2019, Airbus developed the Airbus Vahana, an electric-powered single-seater tilt-wing aircraft with the vision to offer a cost-comparable replacement for short-range urban transportation. By the end of the project in 2019, the aircraft had finished 138 test flights both in vertical and horizontal modes[19].

Currently, NASA is developing the GL-10 Greased Lightning, a hybrid UAV powered by 10 electric-powered propellers and 2 diesel generators[20].

1.4.2 Jet-powered thrust vector propulsion

Jet-powered thrust vector propulsion is another approach for achieving VTOL capabilities. Technically, the mentioned aircraft types in this section belong to the Vertical or Short Take-Off and Landing (V/STOL) category because, although they are capable of vertical take-off and landing, they can only do so in partial armament and fuel load. This category of aircraft includes Hawker Siddeley Harrier, which is a jet-powered subsonic attack aircraft. The Harrier completed its first flight in 1967, and since 1969, it has entered service in the U.S. Marine Corps, British Royal Air Force, German Air Force, and in Spanish, Italian, Indian, and Royal Thai navies[21].

The Harrier fleets are being replaced by a more modern V/STOL aircraft, the Lockheed Martin F-35B Lightning II. The F-35B is a variant of a single-seat stealth multirole combat aircraft developed by several NATO countries in the Joint Strike Fighter program. The F-35B achieved its first vertical landing in 2010 and entered service with the U.S. Marine Corps in 2015. Since then, this variant of the F-35 has been operated by the Italian Air Force and Navy, Japan Air Self-Defence Force, and the British Royal Air Force and Royal Navy[22, 23, 24].



Figure 1.7: Lockheed Martin F-35B[25]

Chapter 2

Construction

The airframe is constructed from 2 aluminum rods connected in a T shape, so one rod creates a wing, and the other creates the tail. These rods function as a chassis on which other components are attached.

2.1 Fuselage

The fuselage consists of 2 3D printed parts, which can be seen in figure 2.1.

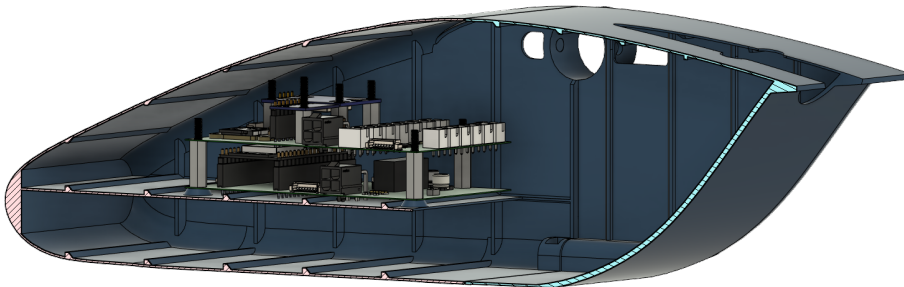


Figure 2.1: Fuselage

Both the parts are printed with the walls only 2 perimeters wide to reduce mass and reinforced with thicker beams to increase rigidity. The two parts are joined together using 4 screws, two of which also connect both of the parts to the wing rod. The constructed fuselage with electronics inside can be seen in figure 2.3 The fuselage houses the flight controller, battery, and ESCs. Additional weights can be placed in the nose of the fuselage to shift the location of the center of gravity, which should be located directly below the wing rod. The constructed convertiplane can be seen in figure 2.2

2. Construction



Figure 2.2: Constructed aircraft

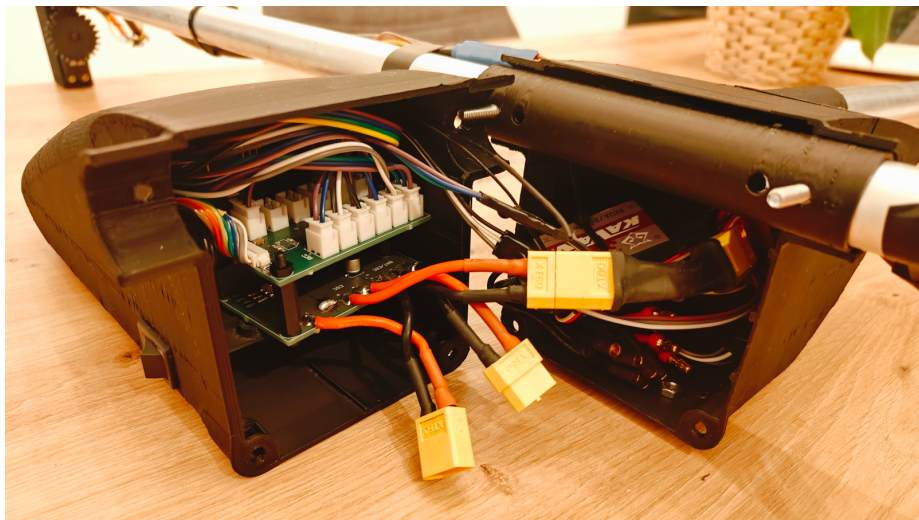


Figure 2.3: Open fuselage

2.2 Wings

The wings of the aircraft are designed to be constructed from several ribs, which should be distributed along the wing rod. The ribs in the middle of the wing are divided into 2 parts, which can revolve, creating an aileron powered by a 12 g servo motor. Figure 2.4 depicts a CAD model of one of the wings.

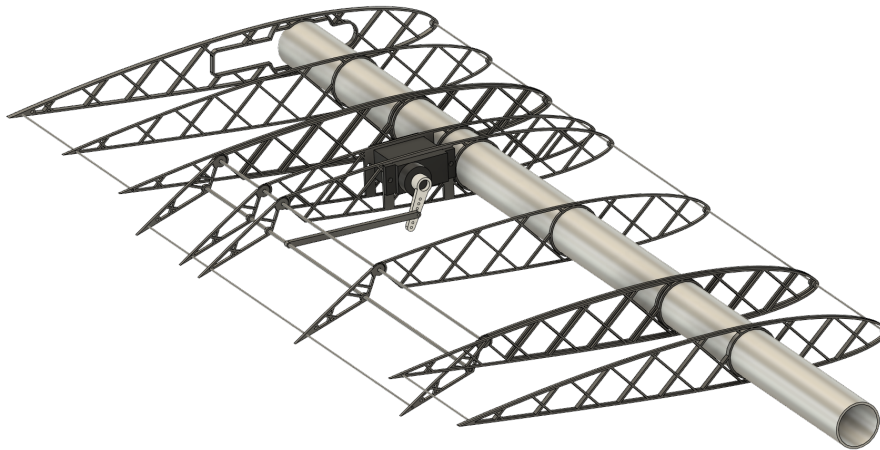


Figure 2.4: Right wing

The ribs are covered with a transparent film, creating an airfoil. I chose the NACA 2414 airfoil because its coefficient of lift to the angle of attack ratio remains linear.

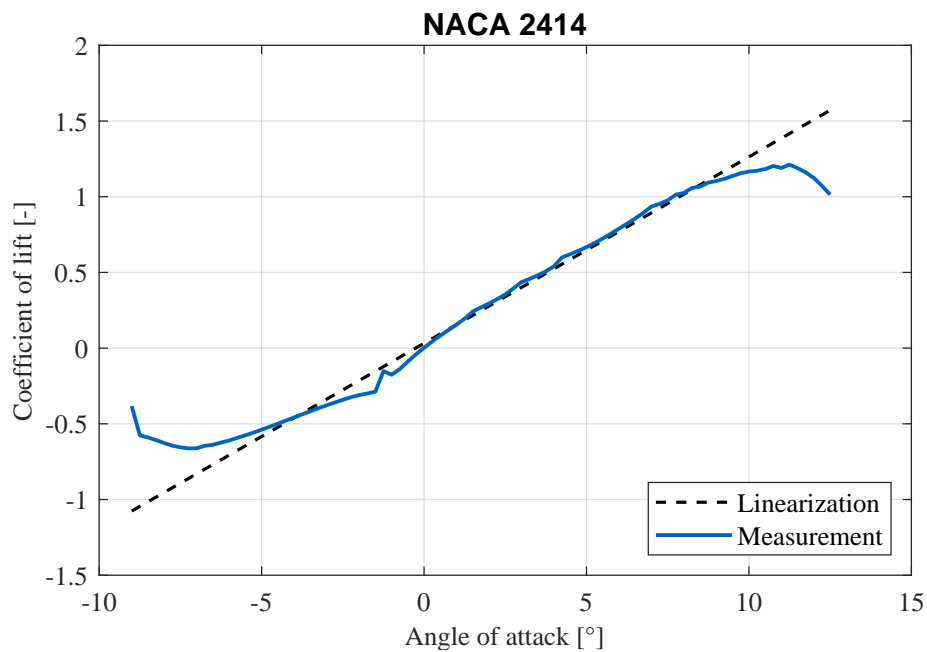


Figure 2.5: Coefficient of lift to AoA function for Reynolds number 50,000[26]

2.3 Rotor tilting mechanism

Figure 2.6 displays the model of the mechanism used to tilt the rotors by a servo motor and figure 2.7 shows the constructed mechanism. The gear displayed on the right side is rigidly connected to the wing, and the other gear is connected to the servo motor. The mechanism can rotate around the rotational joint displayed in figure 2.8.

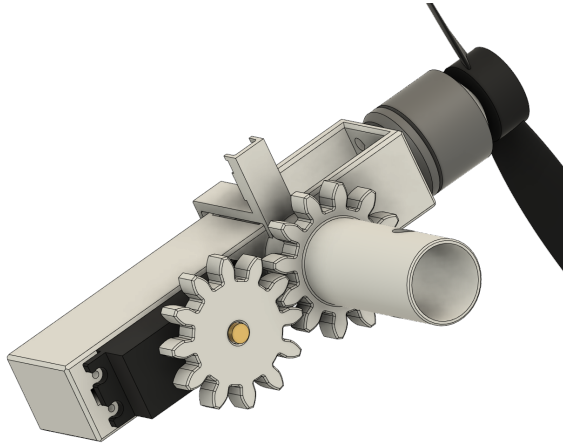


Figure 2.6: Tilting mechanism model

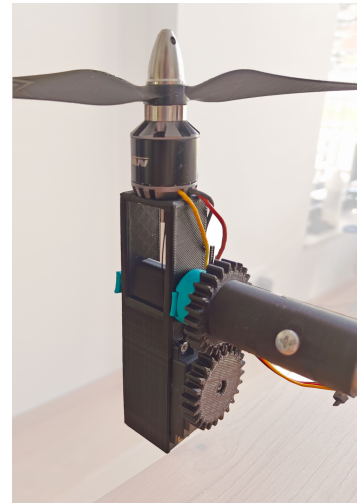


Figure 2.7: Tilting mechanism

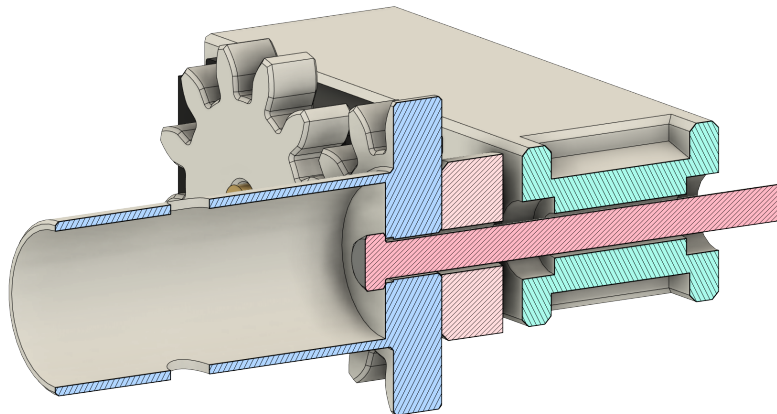


Figure 2.8: Rotational joint

2.4 Stabilizer

The stabilizer houses a servo motor, which can control the elevator using a linkage. The vertical and horizontal stabilizer surfaces are inserted and glued into slots in the main body, and the elevator is separated into 2 parts, which are connected to the body using a rotational joint, which can be seen in figure 2.10. The mechanism to deflect the elevator can be seen in figures 2.11 and 2.12.

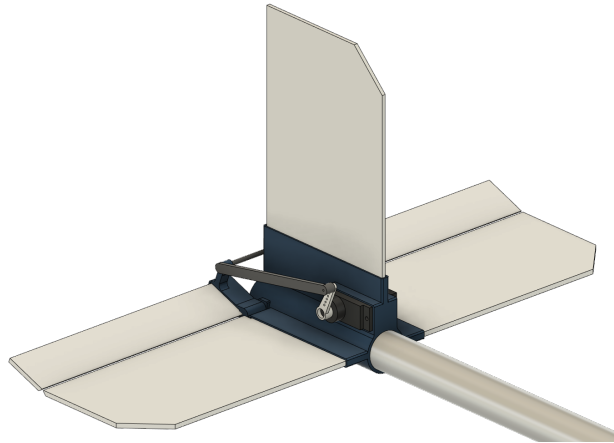


Figure 2.9: Stabilizer model

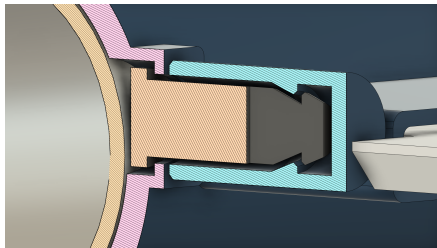


Figure 2.10: Elevator - rotational joint

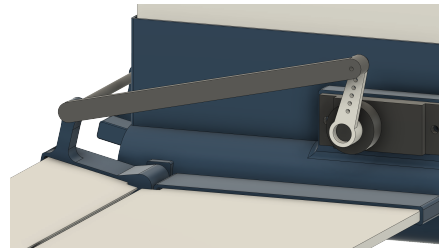


Figure 2.11: Elevator linkage



Figure 2.12: Stabilizer

Chapter 3

Electronics

3.1 Powertrain

When choosing the components of the convertiplane's powertrain, I had to choose components powerful enough to sustain a hover. The minimal needed power is calculated using the following formula

$$P = \frac{F^{\frac{3}{2}}}{\sqrt{2\rho A}} [27], \quad (3.1)$$

where F is the gravitational force, $F = mg$, m is the mass of the aircraft and g is the gravitational acceleration, ρ is the air density and A is the rotor disk area. To account for an error in my initial mass assumption, an additional mass that might be added in the future and a power reserve for stable control, I chose the powertrain to supply double the calculated power. That means 2 brushless KV1300 motors with 2 10x5 propellers, two ESCs, each rated up to 30 A, and a 3S 11.1 V 2200 mAh lithium-polymer battery.

The maximum sustained time in hover t can be estimated using the following equations.

$$t = \frac{Q}{I} \quad (3.2)$$

$$I = \frac{P}{U} \quad (3.3)$$

$$t = \frac{QU}{P}, \quad (3.4)$$

where Q is the battery capacity, P is the required power to sustain hover, and U is the battery voltage. The original mass estimate was 1450 g, consuming 156 W, resulting in a hover time of 9 minutes and 23 seconds.

The final mass, however, increased to 1.74 kg, and the aircraft now requires 201 W, resulting in a decreased hover time of 7 minutes and 17 seconds.

3.1.1 Motor linearization

I measured the output force of the motors at different speeds. The measured data from both motors with linear regression is displayed in figures 3.1 and 3.2.

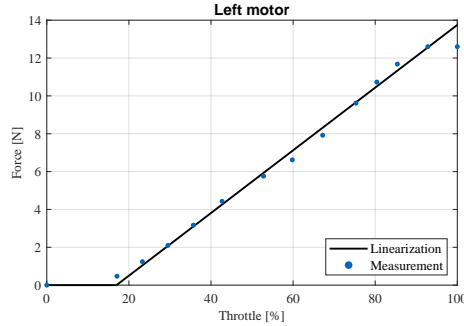


Figure 3.1: Force characteristics of the left motor

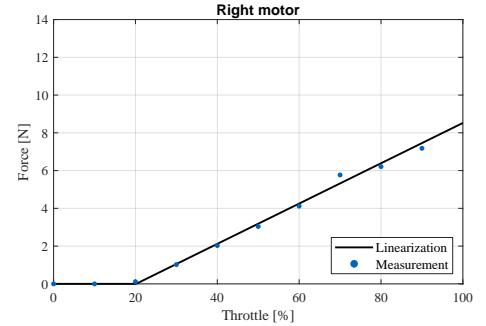


Figure 3.2: Force characteristics of the right motor

From the measured data, it is clear that there is a dead zone from 0 to 20% of the throttle. The aircraft will need to produce around 80% of its maximum power to sustain flight where the dead zone has no effect. However, before take-off, when the motors are turned off or in this dead zone, the integral part of the regulators will wind up. This problem is solved by setting the integral error of all regulators to 0 when the throttle is below a certain value, where the motors are guaranteed to spin. The measurement shows that the right motor is less powerful than the left. The output of the controller is an amplitude of the required force. This force is converted to motor input using separate linear functions for each motor, which negates the effect of different motor performances. Using integral control will also help negate the effect, but the difference will still limit the maximum performance of the left motor. Otherwise, the force difference at the maximum throttle of both motors would induce a roll. The linearized functions are

$$F(u) = a_0 + a_1 u, \quad (3.5)$$

where F is the force produced, u is the throttle level in %, and a_0 and a_1 are linear function coefficients. The coefficient values can be seen in the table 3.1.

	Left motor	Right motor
a_0 [N]	-2.81	-2.15
a_1 [N · % ⁻¹]	0.17	0.11
u [%] ∈	[16.99, ∞]	[20.15, ∞]

Table 3.1: Linear approximation coefficients of motor force function

The chosen ESCs are equipped with BEC (Battery Elimination Circuit) - a step-down regulator that decreases the voltage to 5 V and can be used to power the servo motors.

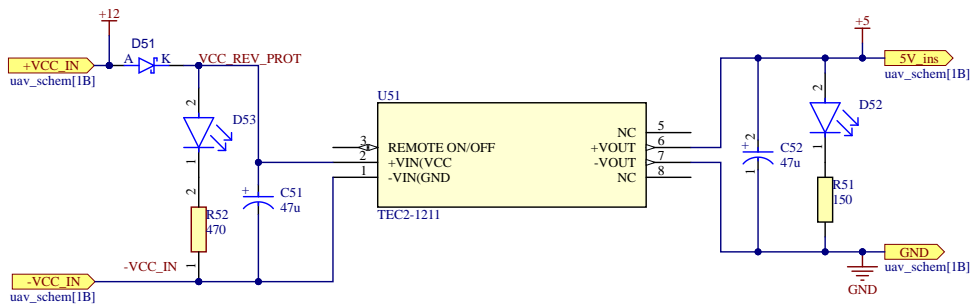


Figure 3.3: Power supply schematic

The microcontroller is supplied by a separate insulated step-down voltage regulator to ensure a stable voltage level. The schematic can be seen in figure 3.3.

3.2 Flight controller

The flight controller is the main computer used to read and log sensor measurements and radio transmitter input. From the measurements, the flight controller calculates the orientation of the aircraft, and based on this information, it adjusts rotor tilt, motor speed, and control surfaces to achieve a stable flight.

3.2.1 MCU

The chosen Microcontroller Unit for a flight controller is an ESP32 microcontroller with a dual-core 32-bit microprocessor running at 240 MHz, with 34 programmable GPIOs, 18 12-bit ADCs, 4 SPI, 2 I²C, and 3 UART interfaces and integrated Wi-Fi and Bluetooth connectivity[28]. Compared to other available microcontrollers, I chose the ESP32 mainly because it offers a dual-core 32-bit microprocessor and Wi-Fi connection, which can be used for real-time wireless controller tuning. The MCU and its schematic can be seen in figures 3.4 and 3.5.

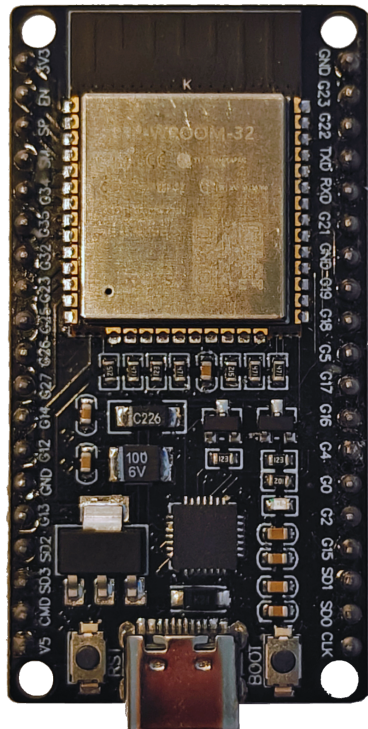


Figure 3.4: ESP32 Microcontroller

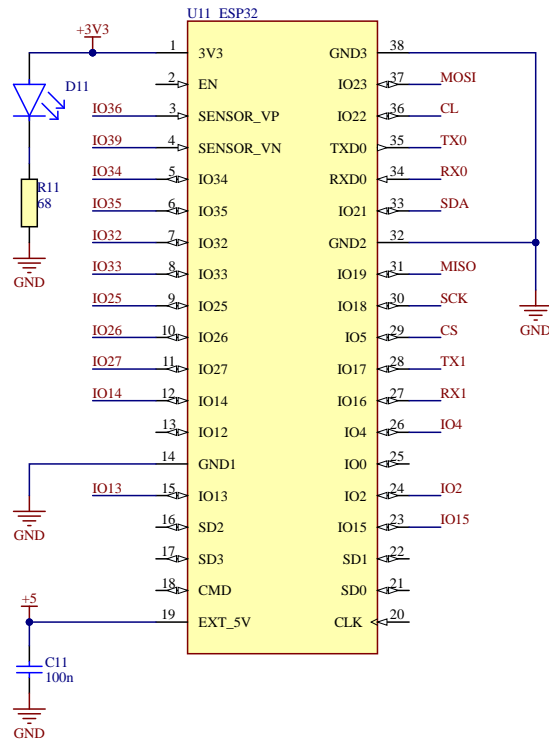


Figure 3.5: ESP32 schematic

■ 3.2.2 Sensors

■ GY-87

GY-87 is a sensor module on a separate PCB, which uses an MPU6050 accelerometer with a gyroscope to measure linear acceleration and angular velocities and a BMP085 barometer to measure atmospheric pressure.

Both the accelerometer and gyroscope have adjustable ranges. The accelerometer can be set to measure the force with a range from $\pm 2g$ to $\pm 16g$ and the gyroscope range can be set from $\pm 250^\circ \cdot s^{-1}$ to $\pm 2000^\circ \cdot s^{-1}$. The values are sent to the MCU as 16-bit integer values. Therefore, an increased range of the measurement decreases the resolution of the sensors.

After experimentation, I have decided to use the $\pm 4g$ range for the accelerometer with a resolution of $8192 \frac{\text{LSB}}{g}$ and the $\pm 250^\circ \cdot s^{-1}$ range for the gyroscope with resolution of $131 \frac{\text{LSB}}{\text{m}\cdot\text{s}^{-1}}$ as these were the highest resolution without reaching saturation.

Using the barometric formula [29], the altitude of the aircraft can be calculated from the measured pressure

$$h(p) = -\frac{p_0 \ln \frac{p}{p_0}}{\rho_0 g}, \quad (3.6)$$

where p is the measured pressure, $p_0 = 1013.25 \text{ hPa}$ is the standard barometric pressure, $\rho_0 = 1.225 \text{ kg} \cdot \text{m}^{-3}$ is the air density at sea level and $g = 9.81 \text{ m} \cdot \text{s}^{-2}$ is the gravitational constant.

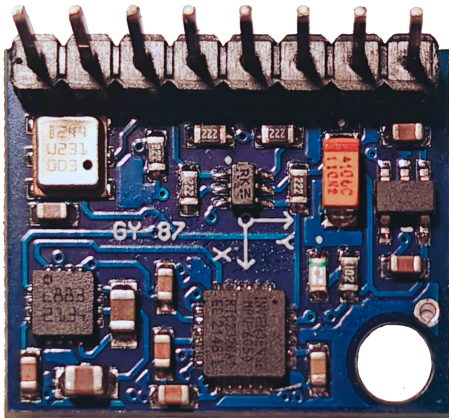


Figure 3.6: GY-87

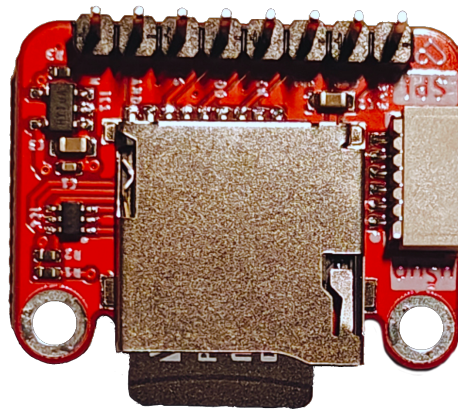


Figure 3.7: microSD card module

■ Micro SD card module

All in-flight data is saved on a micro SD card using a module displayed in figure 3.7 using an SPI bus. The format and process of data logging are mentioned in chapter 4.

■ GPS module

A NEO-6M GPS module uses NMEA-0183 messages to send the measured data to the MCU serial port. Below is an example of a GGA message

```
$GPGGA,115739,4158.84,N,09147.44,W,4,13,0.9,255,M,32.00,M,01,00*6E
```

The message is written in the following format

```
$NID,UTC,LAT,N/S,LON,E/W,FX,NS,HP,ALT,M,HG,M,A,CID*CSM.
```

The abbreviations are explained in table 3.2.

Abbreviation	Definition
NID	NMEA message identifier
UTC	Coordinated universal time
LAT	Latitude
N/S	North/South
LON	Longitude
E/W	East/West
FX	Fix type
NS	Number of satellites tracking
HP	Horizontal dilution of precision
ALT	Altitude
M	meters
HG	Height of the geoid above sea level
M	meters
A	Age of correction data
CSM	Checksum calculated from data between \$ and * noninclusive

Table 3.2: NMEA-0183 Reference manual[30]



Figure 3.8: NEO-6M

3.2.3 PCB

As mentioned before, the flight computer is built from modules on separate PCBs. This way, it was possible to debug the flight computer on a breadboard before the manufacture of a printed circuit board. However, the connectors connected to a breadboard are prone to disconnection. I have, therefore, decided to design a PCB that serves as a motherboard for the components. The PCB is separated into 2 boards, which are placed on top of each other to decrease the area of the board. The signal connections between these 2 boards are ensured by Molex PicoBlade connectors, and voltage supply and ground are connected using Molex MiniFitJr connectors.

The 3S lithium-polymer battery is connected to the board, where the voltage is stepped down to 5V using an insulated DC/DC converter. This converter powers the ESP32 and the RC receiver. The ESP has an inbuilt 3.3V LDO regulator, which supplies the sensors and microSD card reader. Each of the voltage supplies is connected to a separate LED to indicate whether the supply is operational. The PCB also provides a connection to the servos and ESCs. The boards can be seen in figures 3.10 and 3.9, the assembled flight controller can be seen in figure 3.12 and the schematics of the PCB are included in the appendix A.

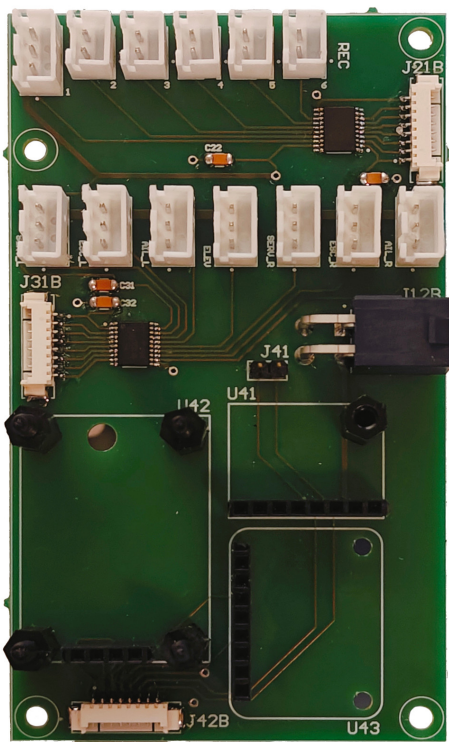


Figure 3.9: PCB top board

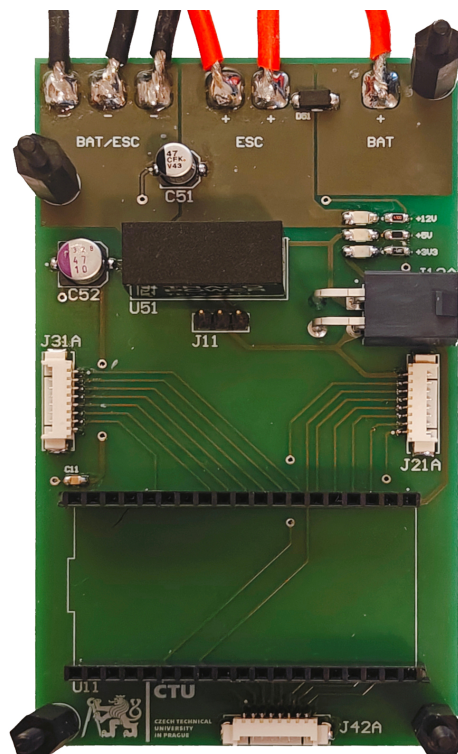


Figure 3.10: PCB bottom board

3.2.4 Radio Control

The aircraft is controlled via a 6-channel RC transmitter shown in figure 3.11. The 4 stick channels are used for roll, pitch, throttle, and yaw input. A 2-position switch (SW1) is used to enable or disable the motor rotation, and a 3-position switch (SW3) is used to switch between vertical, transition and horizontal modes. These modes are further described in chapter 7.

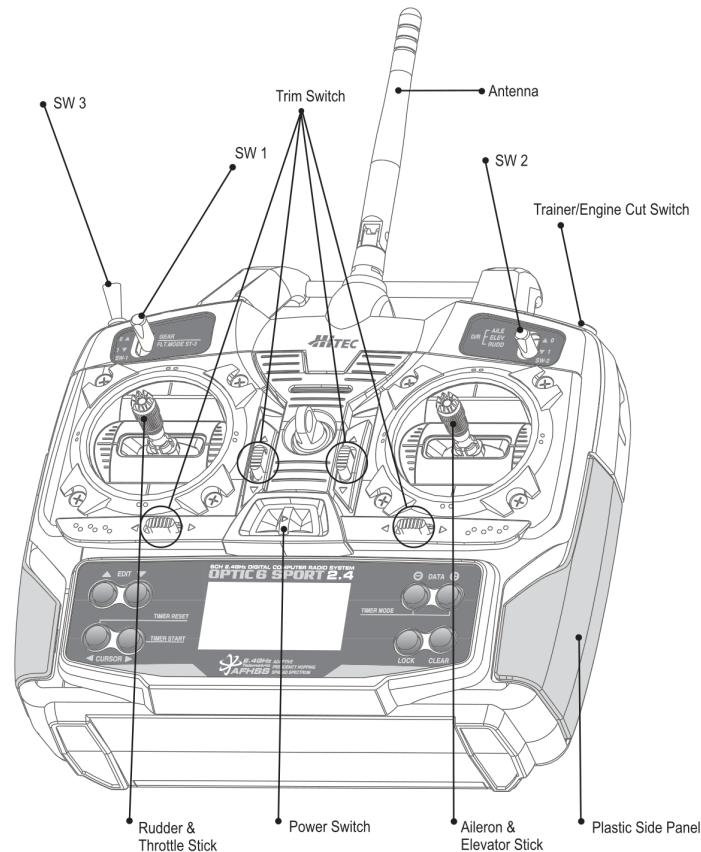


Figure 3.11: RC transmitter - Hitec Optic 6 Sport[31]

A compatible RC receiver is used to receive the inputs. The output of this receiver is a PWM signal with a frequency of 50 Hz. The information about the stick position can be read from the duty cycle, which ranges from 5 to 10%. The signal from a single channel is read using an interrupt pin that is triggered each time the logical value of the sensor changes from low to high. After the trigger, the time from the last change is measured. This interval ranges from 1000 to 2000 μs . Each of the channels is read by a different interrupt pin. The switches can be read the same way as the axes; only the duty cycle can reach 2 or 3 discrete values instead of the full range.

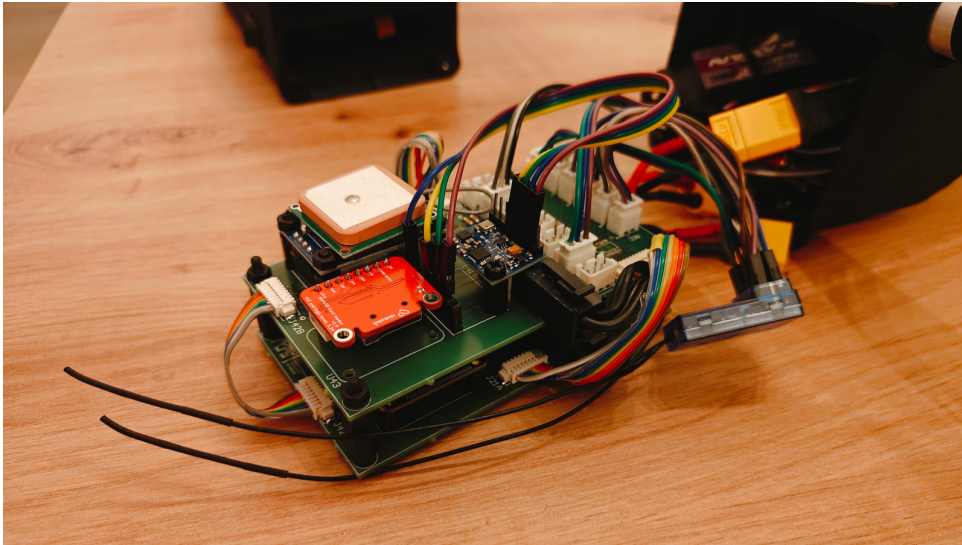


Figure 3.12: Assembled flight controller

Chapter 4

Flight controller software

A flight controller requires robust software that can sustain a failure of multiple non-critical systems and still allow the user to continue flight, land, or, in the worst case, turn off motors to reduce the risk of an injury or third-party property damage. To ensure the independence of each system, I have decided to separate the code into several parallel tasks, which run simultaneously. Each task can run at a different frequency, and more critical tasks can be assigned higher priority than others. For example, the accelerometer offers a refresh rate of up to 1 kHz, and the GPS only 1 Hz. It is, therefore, beneficial that these sensors can be read at different rates.

The code is written in Arduino, which is a variation of the C++ language, using freeRTOS to achieve parallelization. FreeRTOS also enables the use of both microprocessor cores, increasing the computational power of the microcontroller. All variables shared between the tasks are protected using mutexes to prevent race conditions.

4.1 Sensor reading

4.1.1 Calibration

After each power-up of the flight controller, the gyroscope and the output of the Kalman filter are calibrated, to reduce the effect of the gyroscope offset and to set the initial orientation of the IMU. Once communication with the sensors is established, the MCU reads and averages 200 samples from each gyroscope axis. These values are subtracted from each subsequent measure to reduce the drift of the gyroscope. While this calibration is running, the rest of the system is on standby. This is achieved using an event group. The event group is a shared variable set to 0 by default that can be read or written to by each task. After a calibration task is completed, it sets its assigned bit of the event group to 1. The tasks waiting for the end of the calibration are only allowed to read this variable, and once all required bits are assigned to 1, the tasks can start. Specifically, the IMU completes the gyroscope calibration and sets the LSB of the event group to 1. The Kalman filter task reads that the value of LSB is now 1 and starts its calibration. Once this calibration is complete, the Kalman filter task sets the second bit of the event group to 1,

allowing all the other tasks to run.

During this calibration, the motors remain turned off, and all servos are in their center positions, disregarding any user input from the receiver.

■ 4.1.2 GY-87

The GY-87 module communicates with the MCU via I²C bus using the Wire, BMP085, and MPU6050 Arduino libraries. After the flight controller is powered on, the MCU starts testing the connection and halts until successful communication is established with the MPU6050 and the BMP085. These accelerometer and gyroscope measurements are used to calculate the orientation as will be described in chapter 5. The barometer measurements are used to calculate the aircraft's altitude and may be used for altitude hold control in the future.

■ 4.1.3 NEO-6M

The NEO-6M GPS module used UART to transfer data to the MCU. The MCU attempts to read the UART buffer regularly. Once it is filled from the GPS module, the MCU first calculates and confirms the checksum of the message. If the checksum is correct and the NMEA message identifier is GGA, the message is parsed into a predefined struct.

Because the GPS information is non-critical, the rest of the tasks do not have to wait until a fix from the GPS is received.

■ 4.1.4 MicroSD card module

The microSD card module records sensor measurements, user inputs, calculated orientation, and motor and servo control inputs. All entries are saved in the Comma-Separated Values (CSV) format, which can be easily read and analyzed in Matlab. The module uses the SPI bus to communicate with the MCU. I have used the SD and SPI libraries to write to and read from the files stored on the SD card. After powering up, the MCU creates a new log file. The name of the file is initiated as `log_001.csv`. The number is incremented if an existing file with the same name exists until the name is collision-free.

After the file is created, a header is written on the first row containing column names separated by commas. The columns are organized in the following order:

- GY-87 data
 - accelerometer in all 3 axes
 - gyroscope in all 3 axes
 - pressure
 - temperature
- GPS data
 - UTC
 - latitude
 - longitude
 - altitude
 - GPS fix
- Kalman filter data
 - pitch
 - roll
- motor control inputs
 - left motor
 - right motor
 - left tilt servo
 - right tilt servo
 - left aileron
 - right aileron
 - elevator
- RC receiver data
 - roll input
 - pitch input
 - throttle input
 - yaw input
 - motor on
 - flight mode

The microSD card module is also not critical for the operation; therefore, the rest of the tasks are allowed to run without a successful initialization of the module.

4.2 Wireless data transfer

Using the WiFi library, I have programmed the MCU to create a WiFi access point. After connecting to this local network, it is possible to send data to and from the MCU over TCP with Simulink. I have used this primarily to tune the PID controllers in real-time without the need to disassemble the aircraft, and to receive the IMU measurements during experiments. A Simulink scheme used to tune PID controllers is displayed in figure 4.1.

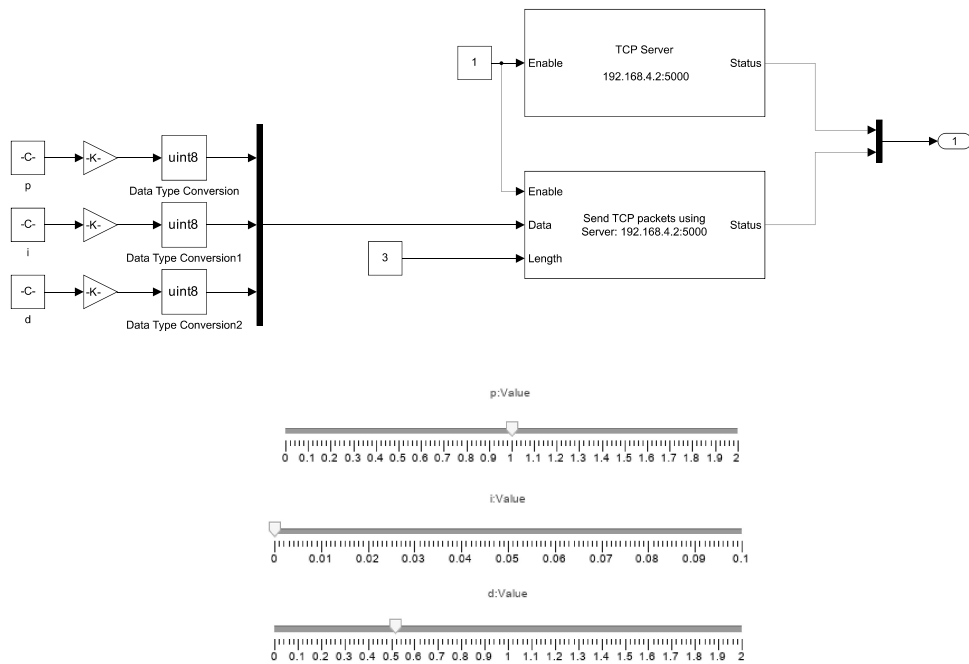


Figure 4.1: Simulink PID tuning block scheme

Chapter 5

Inertial measurement unit

This chapter describes the process of acquiring the aircraft's orientation from the accelerometer and gyroscope and uses the following notation[32].

$$\mathbf{x}_k - \text{real state of a system} \quad (5.1)$$

$$\tilde{\mathbf{x}}_k - \text{measurement of a state} \quad (5.2)$$

$$\hat{\mathbf{x}}_k^- - \text{prediction of a state before taking the measurement} \quad (5.3)$$

$$\hat{\mathbf{x}}_k - \text{updated prediction of a state after taking the measurement} \quad (5.4)$$

Because the accelerometer does not measure acceleration, but rather the forces acting on a suspended weight inside the sensor, the aircraft's pitch and roll can be calculated using the gravitational force vector

$$\tilde{\phi} = \arctan \frac{\tilde{a}_x}{\sqrt{\tilde{a}_y^2 + \tilde{a}_z^2}} \quad (5.5)$$

$$\tilde{\theta} = \arctan \frac{\tilde{a}_z}{\tilde{a}_x}, \quad (5.6)$$

where $\tilde{\phi}$ is the roll measurement, $\tilde{\theta}$ is the pitch measurement, and $\tilde{a}_{x,y,z}$ are measurements of the acceleration in the respective axes. The accelerometer measurement is, however, burdened by noise, and the calculation also assumes zero linear acceleration. If a force acts upon the sensor, accelerating it in the x axis, the a_x reading will increase, and the calculated pitch will change without the sensor physically rotating.

Using the onboard gyroscope, the pitch and roll can be estimated from the system dynamics. The gyroscope measures angular velocity around all three axes. With set sampling period Δt , the angle estimate is

$$\hat{\phi}_k = \hat{\phi}_{k-1} + \tilde{\omega}_{x,k-1} \Delta t, \quad (5.7)$$

where $\hat{\phi}_k$ is the estimate of a configuration using the previous estimated value $\hat{\phi}_{k-1}$ and the measurement $\tilde{\omega}_{x,k-1}$ is the gyroscope reading from step $k - 1$. In reality, the gyroscope will have a small bias, indicating non-zero angular velocity even at rest. This bias may be small, but because the measurements are integrated over time, the error will drift the estimate further and further from real orientation.

In figure 5.1 is the calculated orientation from both these sensors during an experiment when the sensor was at rest. Even though the sensor was not moving, the gyroscope was initiated with wrong initial conditions and then slowly drifted away. Furthermore, the orientation calculated from accelerometer data is burdened by noise. To reduce the noise and effect of vibrations, I implemented a low-pass filter. However, the low-pass filter has limited capabilities, as filtering at a high frequency will reduce the noise only partially, but filtering at a lower frequency will affect the measurements of the system's faster dynamics.

The initial conditions and gyroscope bias are slightly compensated using calibration during the power-up of the system, when the aircraft should remain at rest with the motors turned off. After the first 200 samples, the average values of angular velocities are calculated and then subtracted from each subsequent measurement. The same calibration process zeros the initial pitch and roll. Data from an experiment where fast oscillations were induced while rotating the sensor can be seen in figure 5.2.

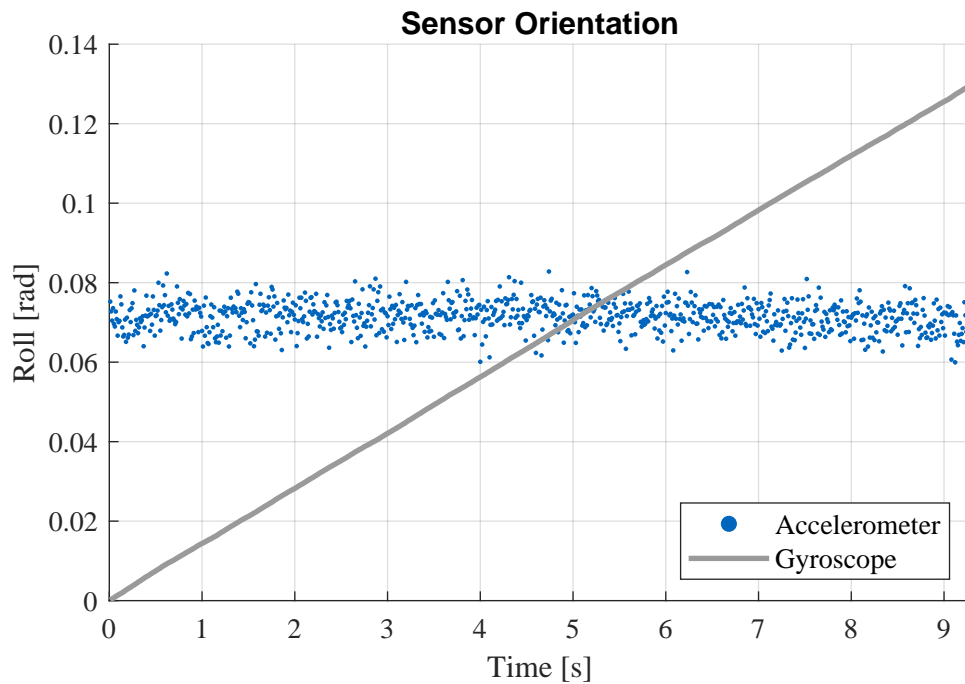


Figure 5.1: Accelerometer and Gyroscope orientation estimation - sensor at rest

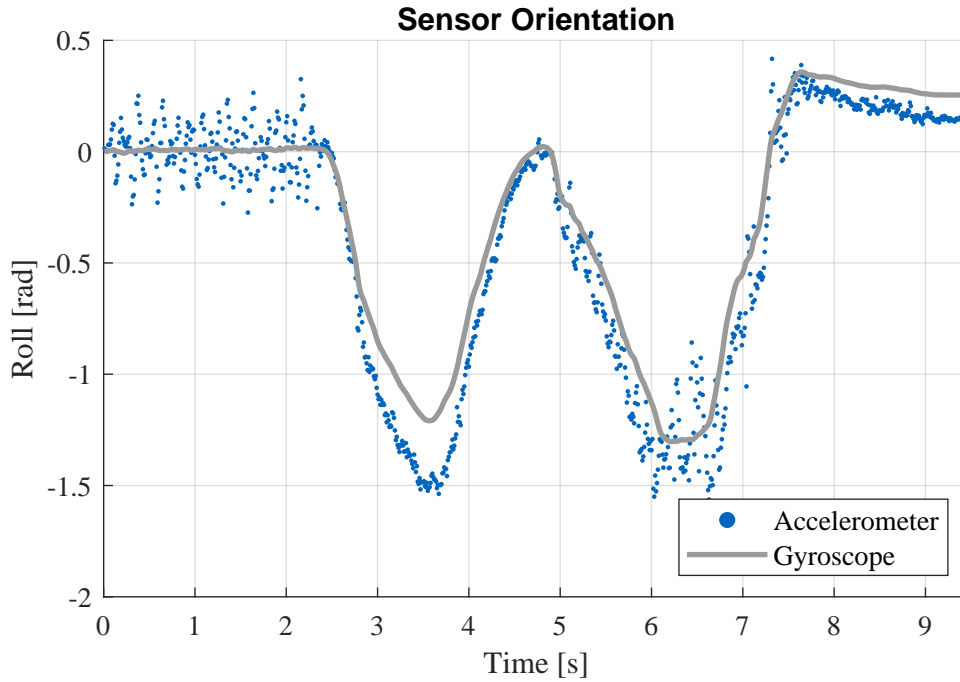


Figure 5.2: Accelerometer and Gyroscope orientation estimation - induced vibrations

5.1 Kalman Filter

Using the Kalman filter, the data can be fused from both sensors to get more accurate and smoother readings. In the next steps, an implementation of the Kalman filter by Prof. D. Simon[32] is used.

The linear model of the stochastic system is

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{w}_{k-1} \quad (5.8)$$

$$\mathbf{y}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k, \quad (5.9)$$

where \mathbf{x}_k is the state vector, \mathbf{F} is the system dynamics matrix, \mathbf{y}_k is the output vector, \mathbf{H} , is the output matrix, \mathbf{w}_{k-1} is the vector of process noise, and \mathbf{v}_k is the vector of measurement noise.

The Kalman filter is initiated as

$$\hat{\mathbf{x}}_0 = \mathbf{x}_0 \quad (5.10)$$

$$\mathbf{P}_0 = \mathbb{E} \left[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T \right], \quad (5.11)$$

where \mathbf{P} is the covariance matrix, and \mathbb{E} is the expected value operator. Next, the covariance matrix is estimated for the next step

$$\hat{\mathbf{P}}_k^- = \mathbf{F}\mathbf{P}_{k-1}\mathbf{F}^T + \mathbf{Q}, \quad (5.12)$$

where \mathbf{Q} is the covariance matrix of the process

$$\mathbf{Q} = \begin{bmatrix} \sigma_1^2 & 0 & \cdots \\ 0 & \sigma_2^2 & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}, \quad (5.13)$$

where σ_i is the standard deviation of the noise w_i . The Kalman gain \mathbf{K} is estimated as

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R})^{-1}, \quad (5.14)$$

where \mathbf{R} is the covariance matrix of the measurement

$$\mathbf{R} = \begin{bmatrix} \sigma_1^2 & 0 & \cdots \\ 0 & \sigma_2^2 & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}, \quad (5.15)$$

where σ_i is the standard deviation of the noise v_i . The state can be estimated and updated using the measurement \mathbf{z}_k .

$$\hat{\mathbf{x}}_k^- = \mathbf{F} \hat{\mathbf{x}}_{k-1} \quad (5.16)$$

$$\hat{\mathbf{x}}_k = \mathbf{K}_k (\mathbf{z}_k - \mathbf{H} \hat{\mathbf{x}}_k^-). \quad (5.17)$$

The last step is to update the estimate of the covariance matrix

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \mathbf{P}_k^-. \quad (5.18)$$

5.2 Accelerometer, Gyroscope data fusion

The model for data fusion is based on a lecture by Prof. T. Moir [33, 34]. For accelerometer and gyroscope fusion, the state vector is

$$\mathbf{x} = \begin{bmatrix} \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \end{bmatrix}, \quad (5.19)$$

and the measurement vector is

$$\mathbf{z} = \begin{bmatrix} \tilde{\phi} \\ \tilde{\omega}_x \\ \tilde{\theta} \\ \tilde{\omega}_y \end{bmatrix}. \quad (5.20)$$

The discrete linear system matrices can be derived from (5.7).

$$\mathbf{F} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.21)$$

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (5.22)$$

The initial conditions are expected to be

$$\hat{\mathbf{x}}_0 = \mathbf{0} \quad (5.23)$$

$$\mathbf{P}_0 = \mathbf{0}. \quad (5.24)$$

Using the data specified in the accelerometer and gyroscope datasheet, the covariance matrix \mathbf{R} is

$$\mathbf{R} = \begin{bmatrix} 0.23 & 0 & 0 & 0 \\ 0 & 0.12 & 0 & 0 \\ 0 & 0 & 0.23 & 0 \\ 0 & 0 & 0 & 0.12 \end{bmatrix}. \quad (5.25)$$

The covariance matrix \mathbf{Q} can be obtained by tuning the diagonal elements until a satisfactory result is reached.

■ 5.2.1 Results

The use of the Kalman filter is demonstrated in figures 5.3 and 5.4. The drift of the gyroscope is completely compensated, and the noisiness of the accelerometer is mostly reduced. The noise can be further reduced using higher values for \mathbf{Q} matrix, but that will also result in a slight unwanted delay.

In these particular examples, the covariance matrix is

$$\mathbf{Q} = 10^{-4} \begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix}. \quad (5.26)$$

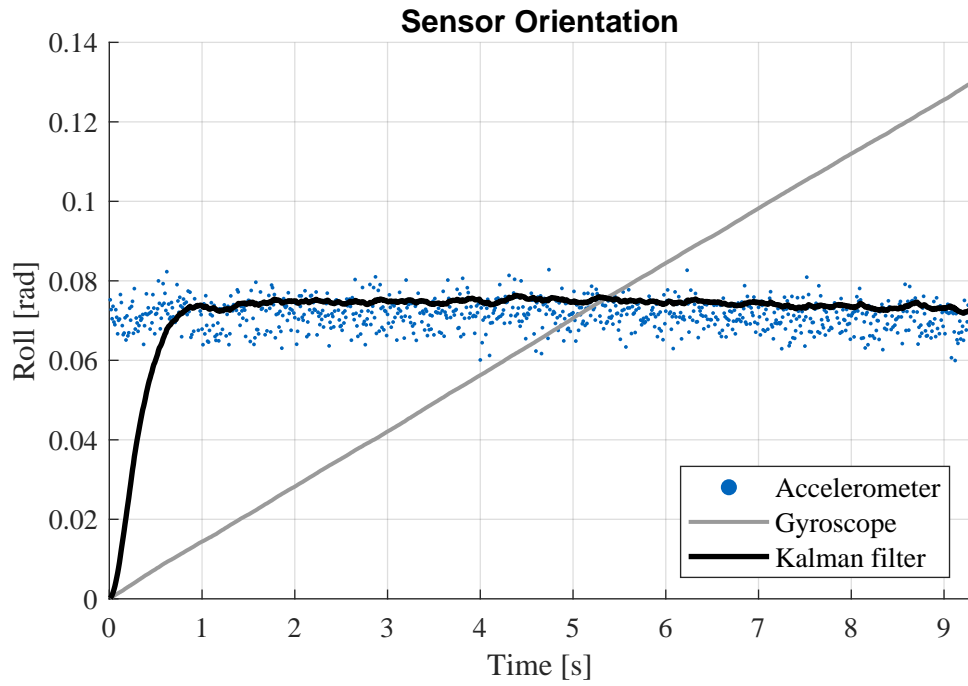


Figure 5.3: Kalman filter demonstration of gyroscope drift elimination

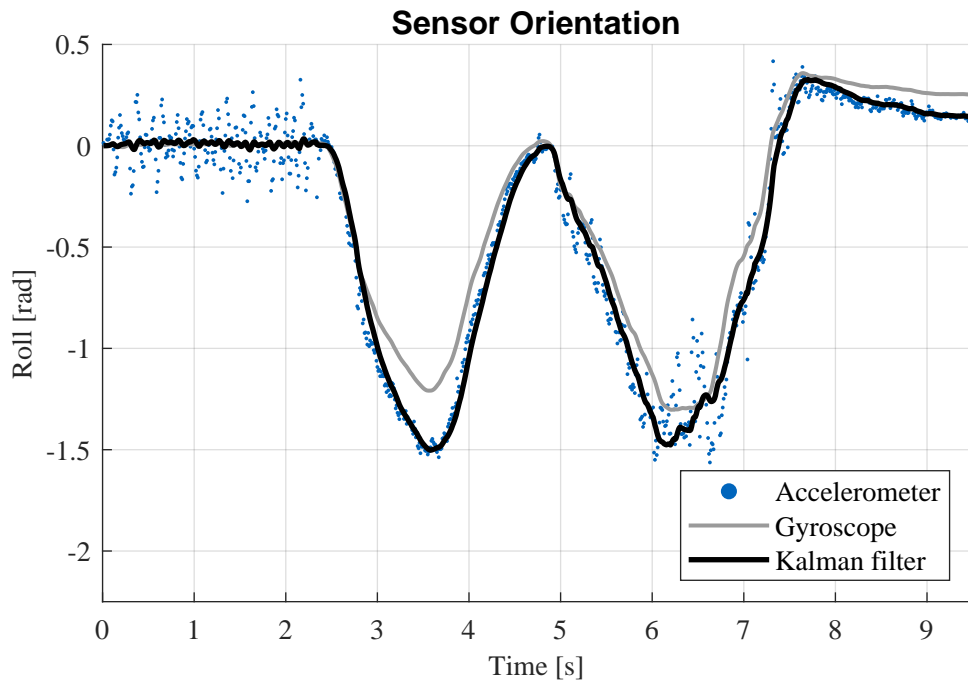


Figure 5.4: Kalman filter demonstration of noise reduction

Chapter 6

Mathematical model

A mathematical model of the system is a powerful tool that allows me to simulate the dynamics of the system to design and tune regulators to achieve stable flight without the risk of damaging the physical model or its surroundings. The acquired differential equations can be simulated using a numerical solver. I use Simulink, as it enables me to create 3D animations of the simulations. Simulink 3D environment can be seen in figure 6.1.

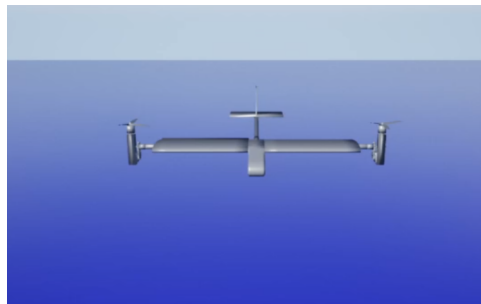


Figure 6.1: Simulink 3D environment

6.1 Coordinate systems

Several coordinate systems need to be defined to model the dynamic behavior of the model. The world coordinate system is rigidly connected to the ground and is referred to with index g . This coordinate system is only used in simulations.

The coordinate system with index 0 is located in the center of gravity of the aircraft, which should be located below the center of a rod that creates the wing. This coordinate system is connected to the aircraft only by translation and rotation around the z -axis, which is oriented from the center of gravity upwards, perpendicular to the ground plane.

Another coordinate system, referred to using index 1, is located in the center of coordinate system 0, but it is fully connected to the aircraft translationally and rotationally. The y -axis is oriented from the center to the left wing, the x -axis forward, and the z -axis upwards. The roll ϕ of the aircraft is defined as the angle between y_0 and y_1 axes (rotation around the x_1 axis). The pitch θ is defined as the angle between x_0 and x_1 axes (rotation around y_1 axis). Finally, the yaw ψ is defined as the angle between x_0 and x_g .

There are then two more coordinate systems, each located in the center of rotation of each rotor module indexed with two on the left wing and three on the right wing. z_2 and z_3 are both perpendicular to the rotor disk of corresponding motors. The $y_{2,3}$ axes are both running through the y_1 -axis with the same orientation, and $x_{2,3}$ are oriented in a way so that all coordinate systems are right-handed as shown in figure 6.2.

The coordinate systems 2 and 3 can only rotate around their y axes. The angles between the y_1, z_1 plane and z_2 and z_3 are called γ_L and γ_R .

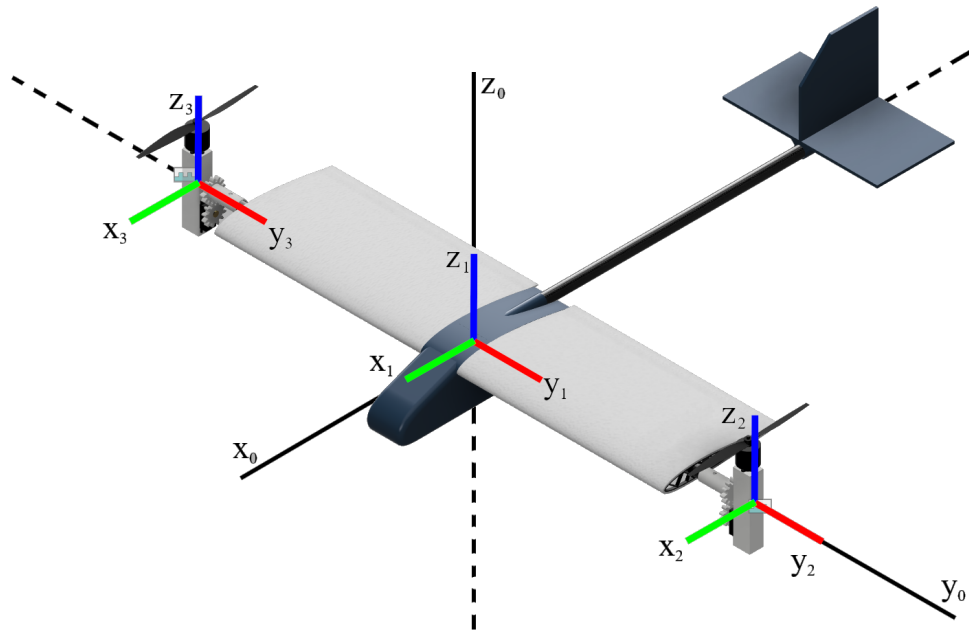


Figure 6.2: Coordinate systems

6.1.1 Transformations

Using these coordinate systems, expressing the forces and torques acting on the aircraft simplifies the modeling process. To express a vector from one coordinate system in a different one, I use transformations in the form of $SO(3)$ objects. An $SO(3)$ object is represented by a rotational matrix. In the case of transformation from coordinate system 2 to coordinate system 1, the rotation of a rotor module happens around the y axis, and therefore the

rotational matrix is

$$\mathbf{R}_{ij} = \begin{bmatrix} \cos \gamma & 0 & -\sin \gamma \\ 0 & 1 & 0 \\ \sin \gamma & 0 & \cos \gamma \end{bmatrix}, \quad (6.1)$$

where i is the index of the next coordinate system, and j is the index of the current coordinate system.

For example, because I chose the z -axis of the second coordinate system perpendicular to the rotor disk, the force vector of magnitude F_l expressed in this coordinate system is

$$\mathbf{F}_2 = \begin{bmatrix} 0 \\ 0 \\ F_l \end{bmatrix}. \quad (6.2)$$

If the rotor module is rotated by an angle γ_L , the force vector expressed in coordinate system 1 is

$$\mathbf{F}_1 = \mathbf{R}_{12}\mathbf{F}_2 \quad (6.3)$$

$$\mathbf{F}_1 = \begin{bmatrix} \cos \gamma_L & 0 & -\sin \gamma_L \\ 0 & 1 & 0 \\ \sin \gamma_L & 0 & \cos \gamma_L \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ F \end{bmatrix} \quad (6.4)$$

$$\mathbf{F}_1 = \begin{bmatrix} -\sin \gamma_L F \\ 0 \\ \cos \gamma_L F \end{bmatrix}. \quad (6.5)$$

Since there are no restrictions for the relative movement between the world coordinate system and the first coordinate system, the transformation is not as straightforward as the previous one. The onboard sensors measure pitch and roll. These can be expressed using rotation matrices.

$$\mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \quad (6.6)$$

$$\mathbf{R}_y = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \quad (6.7)$$

$$\mathbf{R}_{01} = \mathbf{R}_x \mathbf{R}_y. \quad (6.8)$$

Finally, the transformation from coordinate system 0 to g can be expressed as a rotation around the z_g axis.

$$\mathbf{R}_{g0} = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (6.9)$$

$$(6.10)$$

6.2 Vertical flight

6.2.1 Translation

The two rotors can apply forces on the aircraft. The left motor applies force with the magnitude F_L . Expressed in the coordinate system of the left rotor, the force vector is

$$\mathbf{F}_{L2} = \begin{bmatrix} 0 \\ 0 \\ F_L \end{bmatrix}. \quad (6.11)$$

Using the transformation matrix \mathbf{R}_{12} , the force in coordinate system 1 can be expressed as

$$\mathbf{F}_{L1} = \mathbf{R}_{12}\mathbf{F}_{L2}. \quad (6.12)$$

The force produced by the right rotor is

$$\mathbf{F}_{R1} = \mathbf{R}_{13}\mathbf{F}_{R3} = \mathbf{R}_{13} \begin{bmatrix} 0 \\ 0 \\ F_R \end{bmatrix}, \quad (6.13)$$

where F_R is the magnitude of the force.

The sum of these forces with the gravitational force, expressed in coordinate system 0, can be used to simulate the translational dynamics of this model

$$\mathbf{F}_{L0} = \mathbf{R}_{01}\mathbf{F}_{L1} = \mathbf{R}_{01}\mathbf{R}_{12}\mathbf{F}_{L2} \quad (6.14)$$

$$\mathbf{F}_{R0} = \mathbf{R}_{01}\mathbf{F}_{R1} = \mathbf{R}_{01}\mathbf{R}_{13}\mathbf{F}_{R3} \quad (6.15)$$

$$\mathbf{F}_0 = \mathbf{F}_{L0} + \mathbf{F}_{R0} \quad (6.16)$$

$$\mathbf{F} = \mathbf{F}_{L0} + \mathbf{F}_{R0} + \mathbf{F}_g. \quad (6.17)$$

Using the second Newton's Law of Motion, it is possible to derive a set of differential equations, which can be used to simulate the translational movement of the model.

$$\ddot{\mathbf{x}}_{\text{tr}} = \frac{\mathbf{F}}{m}, \quad (6.18)$$

6.2.2 Rotation

To simulate the rotational dynamics of the model, torques acting on the model can be calculated as

$$\tau = Fd, \quad (6.19)$$

where d is the distance between the center of gravity and the line generated by the force vector. The torque is

$$\tau_y = \frac{1}{2}F_{L1z}d_w - \frac{1}{2}F_{R1z}d_w \quad (6.20)$$

$$\tau_z = \frac{1}{2}F_{L1x}d_w - \frac{1}{2}F_{R1x}d_w, \quad (6.21)$$

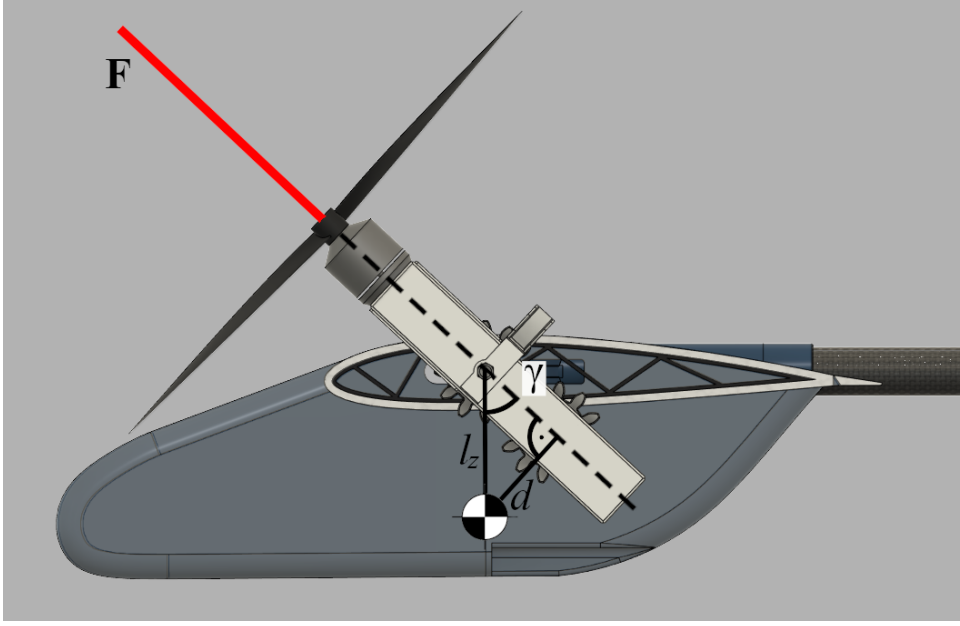


Figure 6.3: Torque — pitch

where d_w is the wingspan of the airplane. Pitch roll is given by the distance d_{cg} , which is a function of angle γ of each motor as displayed in 6.3.

$$d_{cg} = \sin \gamma l_z, \quad (6.22)$$

where l_z is the distance between the CG and the axis of rotation of rotors. The torque is then

$$\tau_x = F_L \sin \gamma_L l_z + F_R \sin \gamma_R l_z. \quad (6.23)$$

The torque vector is

$$\boldsymbol{\tau}_1 = \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} \quad (6.24)$$

$$\boldsymbol{\tau}_0 = \mathbf{R}_{01} \boldsymbol{\tau}_1. \quad (6.25)$$

Using Newton's second law for rotation, the torques can be expressed in coordinate system 0 to derive the differential equations for the rotational dynamics of the system.

$$\boldsymbol{\tau} = \mathbf{J} \ddot{\boldsymbol{\varphi}}, \quad (6.26)$$

where \mathbf{J} is the moment of inertia matrix

$$\mathbf{J} = \begin{bmatrix} J_{xx} & J_{xy} & J_{xz} \\ J_{yx} & J_{yy} & J_{yz} \\ J_{zx} & J_{zy} & J_{zz} \end{bmatrix}. \quad (6.27)$$

This matrix can be obtained from CAD software. The elements of the matrix are

$$\mathbf{J} = \begin{bmatrix} 0.089 & 3.4 \cdot 10^{-5} & 3.5 \cdot 10^{-7} \\ 3.4 \cdot 10^{-5} & 0.13 & 0.0066 \\ 3.5 \cdot 10^{-7} & 0.0066 & 0.2 \end{bmatrix} \left[\text{kg} \cdot \text{m}^2 \right]. \quad (6.28)$$

For simplification, I assume that non-diagonal elements are zero. The angular acceleration is then

$$\ddot{\boldsymbol{\varphi}} = \begin{bmatrix} \frac{\tau_{0x}}{J_{xx}} \\ \frac{\tau_{0y}}{J_{yy}} \\ \frac{\tau_{0z}}{J_{zz}} \end{bmatrix}. \quad (6.29)$$

6.2.3 Linearization

In this section, I obtain a linear state-space model of the system

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \quad (6.30)$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \quad (6.31)$$

Equilibrium point

To linearize the model of the system, I chose the equilibrium point

$$\mathbf{P} = \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{x}_0 \\ \mathbf{y}_0 \end{bmatrix}, \quad (6.32)$$

where \mathbf{u} is the input vector, \mathbf{x} is the state vector and \mathbf{y} is the output vector

$$\mathbf{u}_0 = \begin{bmatrix} F_{L0} \\ F_{R0} \\ \gamma_{L0} \\ \gamma_{R0} \end{bmatrix}. \quad (6.33)$$

The state vector \mathbf{x} is a 12×1 vector

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_{\text{tr}} \\ \mathbf{v}_{\text{tr}} \\ \boldsymbol{\varphi} \\ \boldsymbol{\omega} \end{bmatrix} \quad (6.34)$$

and the output vector is

$$\mathbf{y} = \begin{bmatrix} \mathbf{x}_{\text{tr}} \\ \boldsymbol{\varphi} \end{bmatrix}. \quad (6.35)$$

The second-order differential equations 6.18 and 6.29 can be transformed to double the amount of first-order differential equations.

$$\dot{\mathbf{x}}_{\text{tr}} = \mathbf{v}_{\text{tr}} \quad (6.36)$$

$$\dot{\mathbf{v}}_{\text{tr}} = \frac{\mathbf{F}}{m} \quad (6.37)$$

$$\dot{\boldsymbol{\varphi}} = \boldsymbol{\omega} \quad (6.38)$$

$$\dot{\boldsymbol{\omega}} = \mathbf{J}^{-1}\boldsymbol{\tau} \quad (6.39)$$

In the equilibrium point, the aircraft is expected to have both translational and rotational coordinates and velocities equal to 0. The rotors point upwards, which means that the γ angles are also zero

$$\mathbf{x}_{\text{tr}} = \mathbf{0} \quad (6.40)$$

$$\mathbf{v}_{\text{tr}} = \mathbf{0} \quad (6.41)$$

$$\boldsymbol{\varphi} = \mathbf{0} \quad (6.42)$$

$$\boldsymbol{\omega} = \mathbf{0} \quad (6.43)$$

$$\gamma_{L0} = 0 \quad (6.44)$$

$$\gamma_{R0} = 0. \quad (6.45)$$

The initial velocities are expected to be zero, and therefore the equation 6.37 must be equal to zero.

$$\dot{\mathbf{v}}_{\text{tr}} = \mathbf{0} \quad (6.46)$$

$$\frac{\mathbf{F}}{m} = \mathbf{0} \quad (6.47)$$

$$\mathbf{F} = \mathbf{0}. \quad (6.48)$$

After substituting \mathbf{F} from equation 6.17 the equations are

$$\mathbf{F}_{L0} + \mathbf{F}_{R0} + \mathbf{F}_g = \mathbf{0} \quad (6.49)$$

$$\mathbf{F}_{L0} + \mathbf{F}_{R0} = -\mathbf{F}_g. \quad (6.50)$$

Since the initial conditions are 6.45, substitution of $\boldsymbol{\varphi} = \mathbf{0}$ to the rotational matrices results in identities,

$$\mathbf{R}_{01x}(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos 0 & -\sin 0 \\ 0 & \sin 0 & \cos 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (6.51)$$

The same is true for \mathbf{R}_{01y} , \mathbf{R}_{01z} , \mathbf{R}_{12} and \mathbf{R}_{13} . The matrices \mathbf{R}_{02} and \mathbf{R}_{03} are multiplications of these matrices and therefore

$$\begin{bmatrix} 0 \\ 0 \\ F_{L0} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ F_{R0} \end{bmatrix} = - \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \quad (6.52)$$

$$F_{L0} + F_{R0} = mg. \quad (6.53)$$

The equation 6.43 implies

$$F_{L0} = F_{R0} = \frac{mg}{2}. \quad (6.54)$$

State-space model

To calculate the state-space matrices, the Jacobian of equations (6.36) to (6.39) needs to be calculated.

$$\mathbf{A} = \begin{bmatrix} \frac{\partial \dot{\mathbf{x}}_{\text{tr}}}{\partial \mathbf{x}_{\text{tr}}} & \frac{\partial \dot{\mathbf{x}}_{\text{tr}}}{\partial \mathbf{v}_{\text{tr}}} & \frac{\partial \dot{\mathbf{x}}_{\text{tr}}}{\partial \varphi} & \frac{\partial \dot{\mathbf{x}}_{\text{tr}}}{\partial \omega} \\ \frac{\partial \dot{\mathbf{v}}_{\text{tr}}}{\partial \mathbf{x}_{\text{tr}}} & \frac{\partial \dot{\mathbf{v}}_{\text{tr}}}{\partial \mathbf{v}_{\text{tr}}} & \frac{\partial \dot{\mathbf{v}}_{\text{tr}}}{\partial \varphi} & \frac{\partial \dot{\mathbf{v}}_{\text{tr}}}{\partial \omega} \\ \frac{\partial \dot{\varphi}}{\partial \mathbf{x}_{\text{tr}}} & \frac{\partial \dot{\varphi}}{\partial \mathbf{v}_{\text{tr}}} & \frac{\partial \dot{\varphi}}{\partial \varphi} & \frac{\partial \dot{\varphi}}{\partial \omega} \\ \frac{\partial \dot{\omega}}{\partial \mathbf{x}_{\text{tr}}} & \frac{\partial \dot{\omega}}{\partial \mathbf{v}_{\text{tr}}} & \frac{\partial \dot{\omega}}{\partial \varphi} & \frac{\partial \dot{\omega}}{\partial \omega} \end{bmatrix} \quad (6.55)$$

In equation 6.36, the time derivation of \mathbf{x}_{tr} is, therefore

$$\frac{\partial \dot{\mathbf{x}}_{\text{tr}}}{\partial \mathbf{x}_{\text{tr}}} = \mathbf{0} \quad (6.56)$$

$$\frac{\partial \dot{\mathbf{x}}_{\text{tr}}}{\partial \mathbf{v}_{\text{tr}}} = \mathbf{I} \quad (6.57)$$

$$\frac{\partial \dot{\mathbf{x}}_{\text{tr}}}{\partial \varphi} = \mathbf{0} \quad (6.58)$$

$$\frac{\partial \dot{\mathbf{x}}_{\text{tr}}}{\partial \omega} = \mathbf{0}, \quad (6.59)$$

where \mathbf{I} is a 3×3 identity matrix. The derivations of $\dot{\mathbf{v}}_{\text{tr}}$ are

$$\frac{\partial \dot{\mathbf{v}}_{\text{tr}}}{\partial \mathbf{x}_{\text{tr}}} = \mathbf{0} \quad (6.60)$$

$$\frac{\partial \dot{\mathbf{v}}_{\text{tr}}}{\partial \mathbf{v}_{\text{tr}}} = \mathbf{0} \quad (6.61)$$

$$\frac{\partial \dot{\mathbf{v}}_{\text{tr}}}{\partial \omega} = \mathbf{0}. \quad (6.62)$$

The derivation of $\dot{\mathbf{v}}_{\text{tr}}$ by φ in the equilibrium point is

$$\frac{\partial \dot{\mathbf{v}}_{\text{tr}}}{\partial \varphi} = \begin{bmatrix} 0 & g & 0 \\ g & 0 & 0 \\ 0 & 0 & 0. \end{bmatrix} \quad (6.63)$$

The equation 6.38 defines the time derivation of φ as ω so the partial derivations are

$$\frac{\partial \dot{\varphi}}{\partial \mathbf{x}_{\text{tr}}} = \mathbf{0} \quad (6.64)$$

$$\frac{\partial \dot{\varphi}}{\partial \mathbf{v}_{\text{tr}}} = \mathbf{0} \quad (6.65)$$

$$\frac{\partial \dot{\varphi}}{\partial \varphi} = \mathbf{0} \quad (6.66)$$

$$\frac{\partial \dot{\varphi}}{\partial \omega} = \mathbf{I}. \quad (6.67)$$

The derivations of $\dot{\omega}$ by \mathbf{x}_{tr} , \mathbf{v}_{tr} and φ in the equilibrium point are

$$\frac{\partial \dot{\omega}}{\partial \mathbf{x}_{\text{tr}}} = \mathbf{0} \quad (6.68)$$

$$\frac{\partial \dot{\omega}}{\partial \mathbf{v}_{\text{tr}}} = \mathbf{0} \quad (6.69)$$

$$\frac{\partial \dot{\omega}}{\partial \varphi} = \mathbf{0} \quad (6.70)$$

$$\frac{\partial \dot{\omega}}{\partial \omega} = \mathbf{0}. \quad (6.71)$$

To obtain the matrix \mathbf{B} , the equations have to be derivated by the inputs.

$$\mathbf{B} = \begin{bmatrix} \frac{\partial \dot{\mathbf{x}}_{\text{tr}}}{\partial F_L} & \frac{\partial \dot{\mathbf{x}}_{\text{tr}}}{\partial F_R} & \frac{\partial \dot{\mathbf{x}}_{\text{tr}}}{\partial \gamma_L} & \frac{\partial \dot{\mathbf{x}}_{\text{tr}}}{\partial \gamma_R} \\ \frac{\partial \dot{\mathbf{v}}_{\text{tr}}}{\partial F_L} & \frac{\partial \dot{\mathbf{v}}_{\text{tr}}}{\partial F_R} & \frac{\partial \dot{\mathbf{v}}_{\text{tr}}}{\partial \gamma_L} & \frac{\partial \dot{\mathbf{v}}_{\text{tr}}}{\partial \gamma_R} \\ \frac{\partial \dot{\varphi}}{\partial F_L} & \frac{\partial \dot{\varphi}}{\partial F_R} & \frac{\partial \dot{\varphi}}{\partial \gamma_L} & \frac{\partial \dot{\varphi}}{\partial \gamma_R} \\ \frac{\partial \dot{\omega}}{\partial F_L} & \frac{\partial \dot{\omega}}{\partial F_R} & \frac{\partial \dot{\omega}}{\partial \gamma_L} & \frac{\partial \dot{\omega}}{\partial \gamma_R} \end{bmatrix} \quad (6.72)$$

Since $\dot{\mathbf{x}}_{\text{tr}}$ and $\dot{\varphi}$ are only functions of their respective velocities, the derivatives by all inputs are zero.

$$\frac{\partial \dot{\mathbf{x}}_{\text{tr}}}{\partial F_L} = \frac{\partial \dot{\mathbf{x}}_{\text{tr}}}{\partial F_R} = \frac{\partial \dot{\mathbf{x}}_{\text{tr}}}{\partial \gamma_L} = \frac{\partial \dot{\mathbf{x}}_{\text{tr}}}{\partial \gamma_R} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (6.73)$$

$$\frac{\partial \dot{\varphi}}{\partial F_L} = \frac{\partial \dot{\varphi}}{\partial F_R} = \frac{\partial \dot{\varphi}}{\partial \gamma_L} = \frac{\partial \dot{\varphi}}{\partial \gamma_R} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (6.74)$$

The derivations of $\dot{\mathbf{v}}_{\text{tr}}$ by the inputs are

$$\frac{\partial \dot{\mathbf{v}}_{\text{tr}}}{\partial F_L} = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{m} \end{bmatrix} \quad (6.75)$$

$$\frac{\partial \dot{\mathbf{v}}_{\text{tr}}}{\partial F_R} = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{m} \end{bmatrix} \quad (6.76)$$

$$\frac{\partial \dot{\mathbf{v}}_{\text{tr}}}{\partial \gamma_L} = \begin{bmatrix} \frac{g}{2} \\ 0 \\ 0 \end{bmatrix} \quad (6.77)$$

$$\frac{\partial \dot{\mathbf{v}}_{\text{tr}}}{\partial \gamma_R} = \begin{bmatrix} \frac{g}{2} \\ 0 \\ 0 \end{bmatrix}. \quad (6.78)$$

Derivation of $\dot{\omega}$ results in

$$\frac{\partial \dot{\omega}}{\partial F_L} = \begin{bmatrix} 0 \\ \frac{d_w}{2I_{yy}} \\ 0 \end{bmatrix} \quad (6.79)$$

$$\frac{\partial \dot{\omega}}{\partial F_R} = \begin{bmatrix} 0 \\ -\frac{d_w}{2I_{yy}} \\ 0 \end{bmatrix} \quad (6.80)$$

$$\frac{\partial \dot{\omega}}{\partial \gamma_L} = \begin{bmatrix} -\frac{d_z mg}{2I_{xx}} \\ 0 \\ \frac{d_w mg}{4I_{zz}} \end{bmatrix} \quad (6.81)$$

$$\frac{\partial \dot{\omega}}{\partial \gamma_R} = \begin{bmatrix} -\frac{d_z mg}{2I_{xx}} \\ 0 \\ -\frac{d_w mg}{4I_{zz}} \end{bmatrix}. \quad (6.82)$$

Getting the \mathbf{C} and \mathbf{D} matrices is fairly straightforward since the output vector is the position and rotation vector and \mathbf{D} is $\mathbf{0}$

$$\mathbf{C} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \end{bmatrix} \quad (6.83)$$

■ Transfer functions

The transfer functions can be obtained from the state space model using the following formula

$$\mathbf{G}(s) = \mathbf{C} (s\mathbf{I} - \mathbf{A})^{-1} \mathbf{B} + \mathbf{D}. \quad (6.84)$$

The result of the equation is a 6×4 matrix, where G_{ij} is the transfer function from the j^{th} input to the i^{th} output. These transfer functions are used in chapter 7 to design regulators.

■ 6.3 Horizontal flight

During the horizontal flight, the aircraft transitions to a different configuration, where the rotors are angled directly forward with a set angle of $\gamma_L = \gamma_R = 90^\circ$. And the aircraft orientation is no longer controlled by the tilt of the rotors. The roll of the aircraft is controlled by ailerons, the pitch is controlled using the elevator and the yaw of the aircraft is controlled by differential thrust of motors.

To achieve this control, a different dynamic system incorporating the aerodynamic forces has to be created.

6.3.1 Aerodynamics

At non-zero horizontal forward velocity, the shape of the aircraft, mainly the wing, starts to generate aerodynamical forces. In this model, I only take lift L and drag D into account. These forces are defined using dimensionless aerodynamic coefficients [35] C_D and C_L .

$$D = qSC_D \quad (6.85)$$

$$L = qSC_L \quad (6.86)$$

$$q = \frac{1}{2}\rho v^2, \quad (6.87)$$

where q is the dynamic pressure, $\rho = 1.2250 \text{ kg} \cdot \text{m}^{-3}$ is the air mass density at sea level and S is the nominal wing area (area of the projection of the wing to the $x-y$ plane). The aerodynamic coefficients are complex functions given by the shape of the aircraft, angle of attack and properties of air and are usually computed experimentally in a wind tunnel or using Computational Fluid Dynamics (CFD) simulations.[35] A result of such simulation can be seen in figure 6.4.

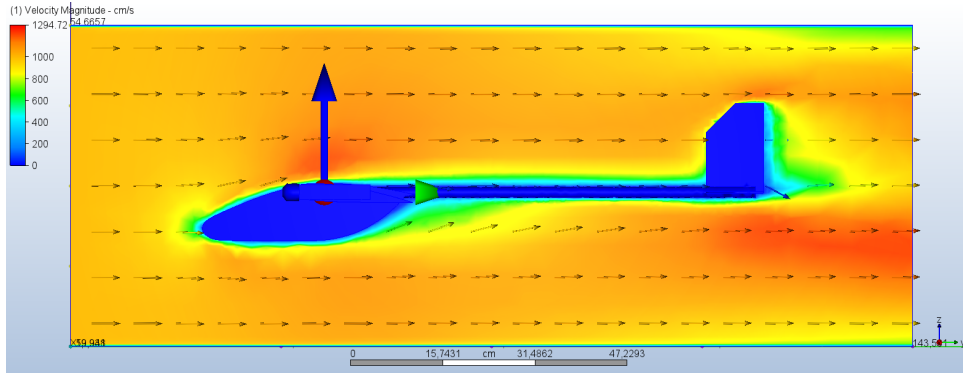


Figure 6.4: CFD of a simplified 3D model - Velocity magnitude

The calculated coefficients are

$$C_D = 0.16 \quad C_L = 0.19. \quad (6.88)$$

Translation

In the aerodynamic model, I assume a constant coefficient of drag regardless of the angle of attack or control surface deflection.

The forces acting upon the aircraft expressed in coordinate system 1 are

$$\mathbf{F}_g = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \quad (6.89)$$

$$\mathbf{L} = \begin{bmatrix} 0 \\ 0 \\ L \end{bmatrix} \quad (6.90)$$

$$\mathbf{D} = \begin{bmatrix} -D \\ 0 \\ 0 \end{bmatrix}. \quad (6.91)$$

Using Newton's second law of motion, a set of differential equations can be obtained

$$\ddot{\mathbf{x}}m = \begin{bmatrix} -\frac{1}{2}C_D S \rho v^2 \\ 0 \\ \frac{1}{2}C_L S \rho v^2 - mg \end{bmatrix}. \quad (6.92)$$

■ Rotation

Using the CFD simulation, I measured the coefficient of lift of the wing cross-section as a function of aileron deflection to simulate the torque acting upon the aircraft and similarly for the elevator. I assume that the center of pressure of the wing is directly above the center of gravity meaning the main wing does not generate any torque. The aerodynamic torques acting upon the aircraft expressed in coordinate system 1 are

$$\boldsymbol{\tau} = \begin{bmatrix} \frac{1}{2}l_{cp}L_L - \frac{1}{2}l_{cp}L_R \\ l_t L_E \\ \frac{1}{2}l_w F_{R1} - \frac{1}{2}l_w F_{L1} \end{bmatrix}, \quad (6.93)$$

where L_L , L_R and L_E are the lift forces caused by corresponding ailerons and the elevator, l_{cp} is the distance between centers of pressure of the ailerons, l_t is the distance between the center of gravity of the aircraft and the center of pressure of the vertical stabilizer and l_w is the distance between the motors. From these forces, another set of differential equations can be calculated.

$$\mathbf{J}\ddot{\boldsymbol{\varphi}} = \begin{bmatrix} \frac{1}{2}l_{cp}qS(C_{La}(\delta_L) - C_{La}(\delta_R)) \\ l_t q S C_{Le}(\delta_E) \\ \frac{1}{2}l_w F_{R1} - \frac{1}{2}l_w F_{L1} \end{bmatrix}, \quad (6.94)$$

where δ_L , δ_R , and δ_E are the angles of deflection of the left and right aileron and the elevator. \mathbf{J} is the matrix of the moment of inertia, and as stated before, I only assume the diagonal elements to be non-zero.

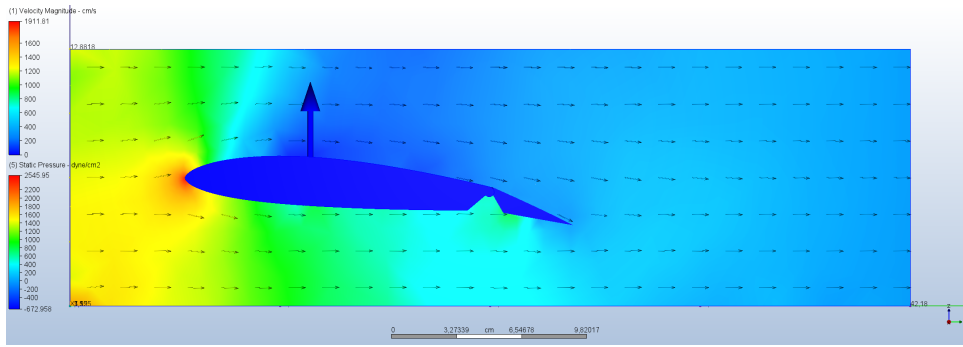


Figure 6.5: Deflected aileron -15° - Static pressure

Figure 6.5 shows a simulation of static pressure around the wing with the aileron deflected 15° downwards. Results of several simulations with different angles of deflection can be seen in table 6.1 and figure 6.6.

δ [$^\circ$]	C_{La}	ΔC_{La}
-20	1.78	0.77
-15	1.54	0.53
-10	1.13	0.12
-5	1.05	0.03
0	1.01	0.00
5	0.73	-0.28
10	0.57	-0.44
15	0.50	-0.52
20	0.44	-0.57

Table 6.1: Coefficient of lift as a function of aileron deflection

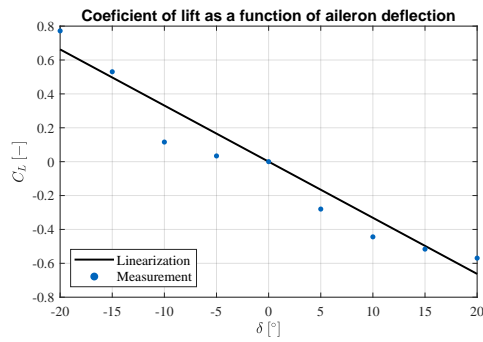


Figure 6.6: Coefficient of lift as a function of aileron deflection

The coefficient of lift as a function of aileron deflection can be linearized as

$$C_{La}(\delta) = a_a \delta, \quad (6.95)$$

where $a_a = -0.033 [\text{deg}^{-1}] = -1.89 [\text{rad}^{-1}]$.

The same process can be repeated for the elevator.

δ [°]	C_{Le}	ΔC_{Le}
-20	0.17	0.12
-15	0.15	0.11
-10	0.14	0.095
-5	0.072	0.030
0	0.042	0.0
5	0.014	-0.028
10	-0.0087	-0.050
15	-0.029	-0.070
20	-0.043	-0.085

Table 6.2: Coefficient of lift as a function of elevator deflection

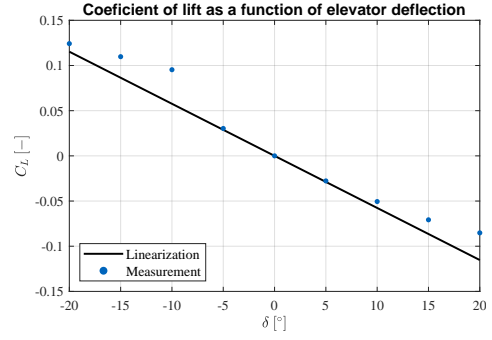


Figure 6.7: Coefficient of lift as a function of elevator deflection

In this case, the slope of the linearized function is

$$a_e = -0.0058 [\text{deg}^{-1}] = -0.33 [\text{rad}^{-1}].$$

6.3.2 Linearization

A linearized model of the system is used to control the orientation of the system in horizontal flight. Control of the position of the system in horizontal flight is too complex for this work, and therefore, this model only describes the rotational dynamics of the system. The state vector is

$$\begin{bmatrix} \dot{\varphi} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \omega \\ \frac{l_{cp}qS}{2J_{xx}} (C_L(\delta_L) - C_L(\delta_R)) \\ \frac{l_tqS}{J_{yy}} C_L(\delta_E) \\ \frac{l_w}{J_{zz}} (F_{R1} - F_{L1}) \end{bmatrix}, \quad (6.96)$$

The state space matrices are obtained as Jacobians of the state space vector.

$$\mathbf{A} = \begin{bmatrix} \frac{\partial \dot{\varphi}}{\partial \varphi} & \frac{\partial \dot{\varphi}}{\partial \omega} \\ \frac{\partial \dot{\omega}}{\partial \varphi} & \frac{\partial \dot{\omega}}{\partial \omega} \end{bmatrix} \quad (6.97)$$

$$\mathbf{B} = \begin{bmatrix} \frac{\partial \dot{\varphi}}{\partial \delta_L} & \frac{\partial \dot{\varphi}}{\partial \delta_R} & \frac{\partial \dot{\varphi}}{\partial \delta_E} & \frac{\partial \dot{\varphi}}{\partial F_L} & \frac{\partial \dot{\varphi}}{\partial F_R} \\ \frac{\partial \dot{\omega}}{\partial \delta_L} & \frac{\partial \dot{\omega}}{\partial \delta_R} & \frac{\partial \dot{\omega}}{\partial \delta_E} & \frac{\partial \dot{\omega}}{\partial F_L} & \frac{\partial \dot{\omega}}{\partial F_R} \end{bmatrix} \quad (6.98)$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (6.99)$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \frac{l_{cp}qS}{2J_{xx}} a_a & -\frac{l_{cp}qS}{2J_{xx}} a_a & 0 & 0 & 0 \\ 0 & 0 & \frac{l_tqS}{J_{yy}} a_e & 0 & 0 \\ 0 & 0 & 0 & -\frac{l_w}{J_{zz}} & \frac{l_w}{J_{zz}} \end{bmatrix} \quad (6.100)$$

■ Equilibrium point

The linearized model assumes the following conditions

$$v = 10 \text{ [m} \cdot \text{s}^{-1}] \quad (6.101)$$

$$\varphi = \mathbf{0} \text{ [}^\circ\text{]}. \quad (6.102)$$

■ State-Space model

Using these conditions, the state space model of the system is

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (6.103)$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -96.6 & 96.6 & 0 & 0 & 0 \\ 0 & 0 & -24.2 & 0 & 0 \\ 0 & 0 & 0 & -5.0 & 5.0 \end{bmatrix} \quad (6.104)$$

$$\mathbf{C} = \mathbf{I} \quad (6.105)$$

$$\mathbf{D} = \mathbf{0}. \quad (6.106)$$

■ Transfer function

Equation 6.84 results in a 6×5 matrix, which contains transfer functions of the horizontal flight control systems.

Chapter 7

Control

This chapter describes the process of designing regulators in vertical and horizontal flight.

7.1 Controller design methods

The dynamic systems in this chapter are represented by their continuous transfer functions derived in chapter 6 and the regulators are designed by placing the closed-loop poles of the systems into desired stable positions. To reduce the order of the systems, 2 nested controllers are used as shown in figure 7.1.

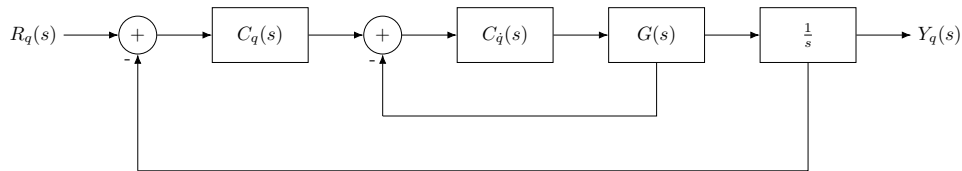


Figure 7.1: Nested controller

In this architecture, $C_{\dot{q}}(s)$ controls the angular velocity, and $C_q(s)$ controls the angular coordinate. This way, instead of one controller for one second-order system, two regulators for two first-order systems have to be designed.

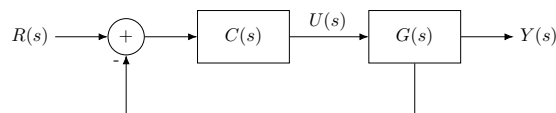


Figure 7.2: Single controller

7.1.1 Pole placement

The transfer functions can be written as fractions

$$G(s) = \frac{b(s)}{a(s)} \quad (7.1)$$

$$C(s) = \frac{q(s)}{p(s)}. \quad (7.2)$$

The transfer function of the closed loop is then

$$F(s) = \frac{\frac{bq}{ap}}{1 + \frac{bq}{ap}} \quad (7.3)$$

$$F(s) = \frac{bq}{ap + bq}, \quad (7.4)$$

where

$$c(s) = ap + bq \quad (7.5)$$

is the characteristic polynomial of the system.

First-order system

In the case of a first-order system, a PI controller $C(s)$ can place the closed-loop poles into desired positions, resulting in a second-order system.

A general first-order system without zeros can be described using the following transfer function

$$G(s) = \frac{b(s)}{a(s)} = \frac{k}{s + v_p}, \quad (7.6)$$

where k is the gain of the system, and v_p is the pole of the system.

The transfer function of a PI controller is

$$C(s) = \frac{k_p s + k_i}{s}. \quad (7.7)$$

Using 7.4 the closed loop transfer function is

$$F(s) = \frac{k(k_p s + k_i)}{s(s + v_p) + k(k_p s + k_i)}. \quad (7.8)$$

The characteristic polynomial of the transfer function is

$$c(s) = s^2 + s(v_p + k k_p) + k k_i. \quad (7.9)$$

The poles of the closed loop can be placed by solving the following equation for k_p and k_i .

$$c(s) = s^2 + 2\zeta\omega_n s + \omega_n^2 \quad (7.10)$$

$$k_p = \frac{2\zeta\omega_n - v_p}{k} \quad (7.11)$$

$$k_i = \frac{\omega_n^2}{k}, \quad (7.12)$$

where ζ is the dampening of the system and ω_n is the system natural frequency. These parameters can be designed using the following equations

$$\zeta = -\frac{\ln OS}{\sqrt{OS^2 + \pi^2}} \quad (7.13)$$

$$\omega_n = \frac{4}{\zeta T_r}, \quad (7.14)$$

where OS is the overshoot and T_r is the rise time: the time it takes the system to stabilize within 2% of its stable-state value k with a step signal as an input[36, 37].

■ Second-order system

A PID controller can place both poles of the closed loop second order system $G(s)$

$$G(s) = \frac{b(s)}{a(s)} = \frac{b_1 s + b_0}{s^2 + a_1 s + a_0}. \quad (7.15)$$

The transfer function of a PID controller is

$$C(s) = \frac{q(s)}{p(s)} = \frac{k_d s^2 + k_p s + k_i}{s}. \quad (7.16)$$

The characteristic polynomial of the closed-loop transfer function is

$$c(s) = (a_2 s^2 + a_1 s + a_0)s + (b_1 s + b_0)(k_d s^2 + k_p s + k_i). \quad (7.17)$$

The desired characteristic polynomial can be written in the following form

$$c_{des}(s) = \beta(s + \alpha\omega_n)(s^2 + 2\zeta\omega_n s + \omega_n^2), \quad (7.18)$$

where $-\alpha\omega_n$ is a third pole of the closed loop. If the pole is placed further to the left from the imaginary axis of the s plane compared to the rest of the poles, the effect of this pole on the fast dynamics of the system will be minimal. A system of equations is obtained by comparing the coefficients of each power in (7.17) and (7.18)

$$\beta = 1 + b_1 k_d \quad (7.19)$$

$$a_1 + b_1 k_p + b_0 k_d = (\alpha\omega_n + 2\zeta\omega_n)\beta \quad (7.20)$$

$$a_0 + b_1 k_i + b_0 k_p = (1 + 2\alpha\zeta)\omega_n^2 \beta \quad (7.21)$$

$$b_0 k_i = \alpha\omega_n^3 \beta. \quad (7.22)$$

Solving this system of equations will result in k_p , k_i , and k_d [37].

7.2 Vertical mode

In vertical mode, the aircraft is able to take off and land upwards with zero horizontal velocity.

7.2.1 Roll

The roll rate $\dot{\phi}$ of the system is controlled by the differential thrust of the motors. The transfer function of this system is $G_{\dot{\phi}}(s)$ is

$$G_{\dot{\phi}}(s) = \frac{4}{s}. \quad (7.23)$$

Using the process described in 7.1.1 with the requirements for the overshoot to be near zero $OS = 1 \cdot 10^{-5}$ and the rise time around $T_r = 2.5$ s and the parameters of the system

$$k = 4 \quad (7.24)$$

$$v = 0, \quad (7.25)$$

the obtained constants are

$$k_p = 0.8 \quad k_i = 0.65. \quad (7.26)$$

The linear model of the system does not take the dynamics and non-linearities of the motors into account. I have, therefore created a simulation model in Simulink and attempted to model these non-linearities to study the designed controllers. After these simulations, I decided to add derivative control and reduce the integral constant to reduce oscillations. I then tuned these parameters to achieve stable system dynamics even with these non-linearities. Furthermore, the integral windup caused by the motor saturation is prevented by using clamping of the integral action of the PID controller. The set of constants derived from these simulations is

$$k_p = 0.8 \quad k_i = 0.05 \quad k_d = 0.5. \quad (7.27)$$

The system response from an experiment where the aircraft was suspended from the ceiling can be seen in figure 7.3.

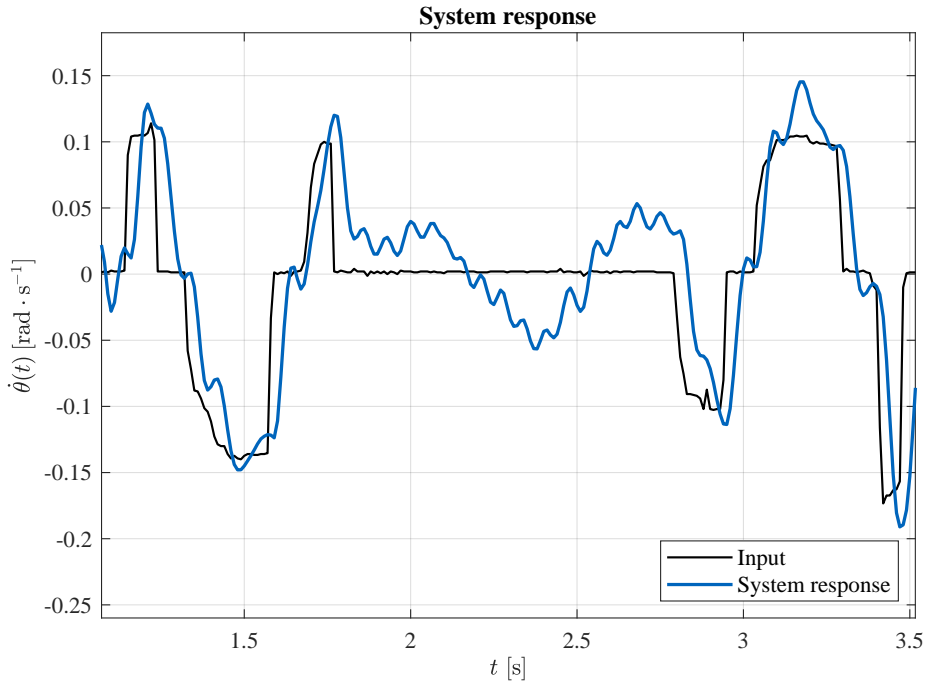


Figure 7.3: Roll rate system response in vertical mode

If I would like to replicate the process for the angular coordinate, the system $G_\phi(s)$ is now

$$G_\phi(s) = F_\phi(s) \frac{1}{s} \quad (7.28)$$

$$G_\phi(s) = \frac{(k_d s^2 + k_p s + k_i)}{(0.25 + k_d) s^2 + k_p s + k_i} \frac{1}{s} \quad (7.29)$$

which is a third-order system.

The adjunct polynomial scheme can be used to approximate a second-order model of the system[38].

Let the transfer function be

$$G(s) = \frac{b_2 s^2 + b_1 s + b_0}{a_3 s^3 + a_2 s^2 + a_1 s + a_0} \quad (7.30)$$

The transient and steady-state gains of the system are defined as

$$K_t = \frac{b_2}{a_3} \quad (7.31)$$

$$K_{ss} = \frac{b_0}{a_0} \quad (7.32)$$

The approximated second-order system using the adjunct polynomial scheme is

$$R(s) = \frac{b_1 s + b_0}{a_2 s^2 + a_1 s + a_0} \quad (7.33)$$

which can be simplified by scaling to

$$R(s) = \frac{s + \frac{b_0}{b_1}}{s^2 + \frac{a_1}{a_2}s + \frac{a_0}{a_2}}. \quad (7.34)$$

The transfer function needs to be modified to maintain the original transient and steady-state gains

$$R(s) = \frac{K_t s + K_{ss} \frac{b_0}{b_1}}{s^2 + \frac{a_1}{a_2}s + \frac{a_0}{a_2}}. \quad (7.35)$$

The second-order approximation for $G_\phi(s)$ is then

$$G_\phi(s) = \frac{k_d s^2 + k_p s + k_i}{(0.25 + k_d)s^3 + k_p s^2 + k_i s} \quad (7.36)$$

$$K_t = \frac{k_d}{0.25 + k_d} \quad (7.37)$$

$$(7.38)$$

$$K_{ss} = \infty \quad (7.39)$$

$$R(s) = \frac{\left(\frac{k_d}{0.25+k_d}\right)s + \frac{k_i}{k_p}}{s^2 + \frac{k_i}{k_p}}. \quad (7.40)$$

A PID controller for this second-order system can be then designed using the pole placement method described in section 7.1.1. The result of the pole-placement method with the same requirements as in 7.2.1 and $\alpha = 20$ are constants

$$k_p = 3.7 \quad k_i = 1.7 \quad k_d = 0.3. \quad (7.41)$$

This regulator was also further tuned to combat the non-linearities of the system. The system response from an experiment can be seen in figure 7.4.

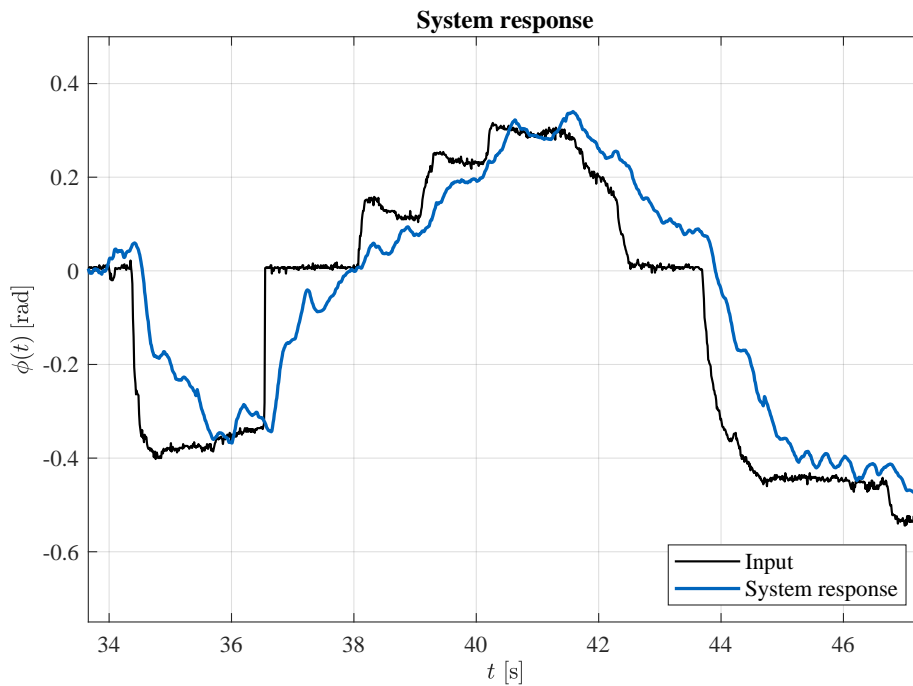


Figure 7.4: Roll system response in vertical mode

7.2.2 Pitch

The pitch and pitch rate of the aircraft are controlled by tilting both rotors in the same direction. The transfer function of pitch rate is

$$G_{\dot{\phi}}(s) = \frac{1.25}{s}. \quad (7.42)$$

The transfer function differs from the roll rate only in gain, and therefore the process designing a regulator is the same as in section 7.2.1.

$$k_p = 0.75 \quad k_i = 0.001 \quad k_d = 0.3. \quad (7.43)$$

7.2.3 Yaw

The flight controller is not equipped with a compass and computing the heading from GPS is not possible unless the aircraft is moving. Therefore, only the angular velocity is controlled using a single PID controller.

The yaw of the aircraft in its vertical configuration can be controlled by tilting each rotor in the opposite direction. The transfer function from rotor angle to yaw rate is

$$G_{\dot{\psi}}(s) = \frac{12.5}{s}. \quad (7.44)$$

Using the same method as in 7.2.1, and after manual tuning, the following constants are obtained

$$k_p = 0.25 \quad k_i = 0.01 \quad k_d = 0.08. \quad (7.45)$$

Figure 7.5 shows a system response to several step inputs.

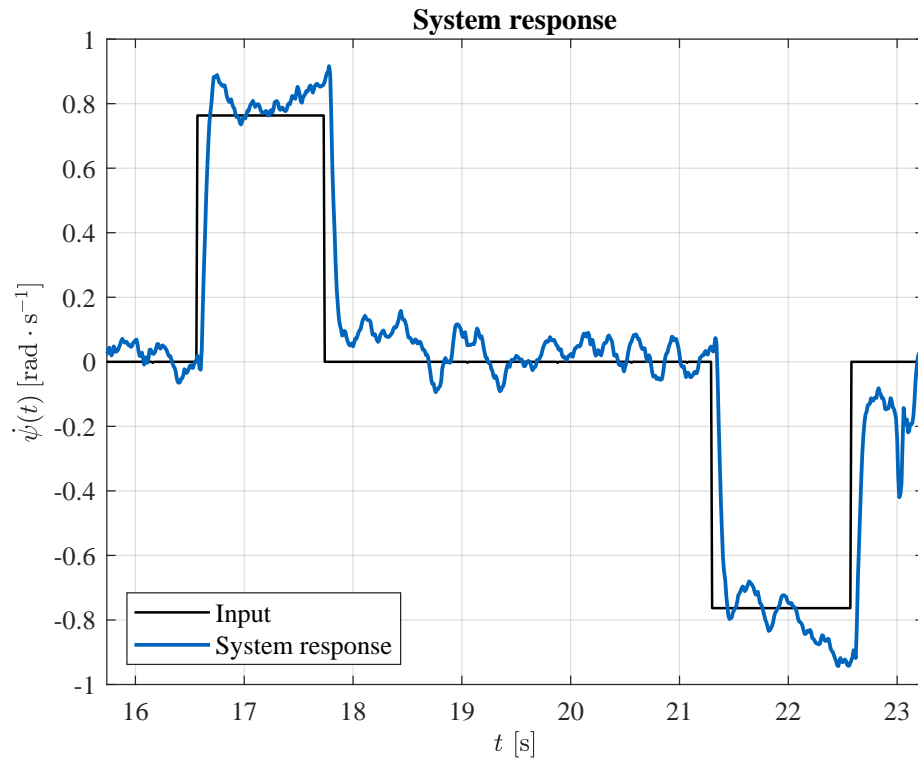


Figure 7.5: Yaw rate system response in vertical mode

7.3 Horizontal mode

In the horizontal mode, the aircraft has the same dynamics as a fixed-wing aircraft, which is naturally stable. Despite the natural stability of the system, a controller can be used to decrease the effect of turbulence and simplify the piloting of the aircraft further.

7.3.1 Roll

The roll of the aircraft in horizontal flight is controlled using asymmetric deflection of the ailerons. The transfer function from right aileron deflection to the angular velocity of roll, derived in section 6.3.1, is

$$G(s) = \frac{1.69}{s}. \quad (7.46)$$

The transfer function from the other aileron is the same, only multiplied by -1 . I will be using both ailerons deflected in the opposite direction. Therefore, the effect will be double, and the transfer function will change to

$$G(s) = \frac{3.38}{s}. \quad (7.47)$$

Using the pole placement method 7.1.1 the following constants are obtained

$$k_p = 0.8 \quad k_i = 0.65. \quad (7.48)$$

A simulated step response of roll rate can be seen in figure 7.6

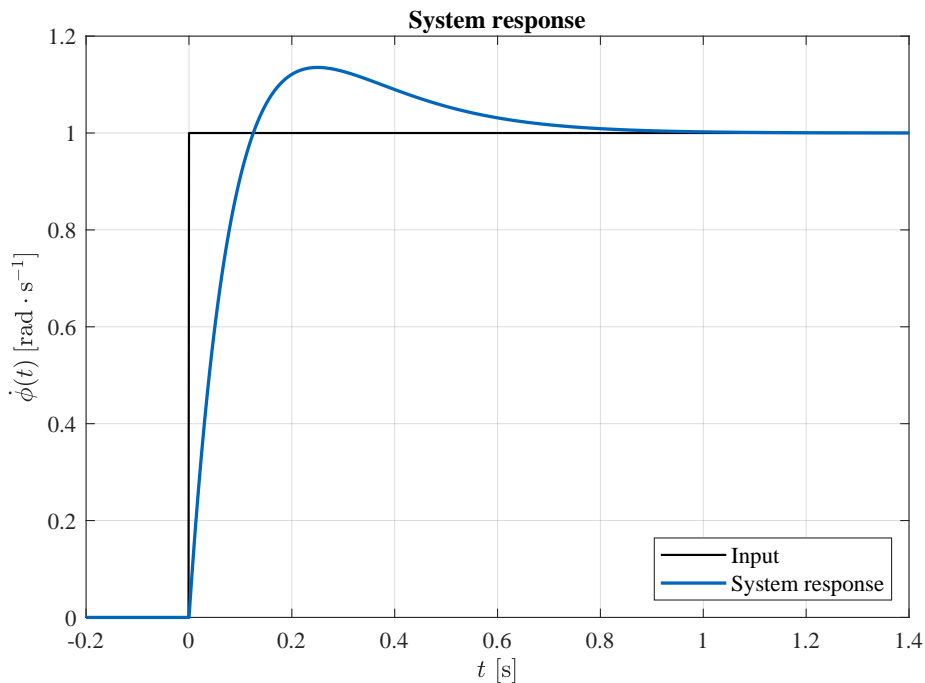


Figure 7.6: Roll rate system response in horizontal mode

7.3.2 Pitch

In the horizontal mode, the pitch and pitch rate are controlled using the elevator. The transfer function from elevator deflection to pitch rate is

$$G(s) = \frac{24.2}{s}. \quad (7.49)$$

Using the pole placement method, I designed a PI controller with the constants

$$k_p = 0.36 \quad k_i = 0.65. \quad (7.50)$$

A simulated step response can be seen in figure 7.7

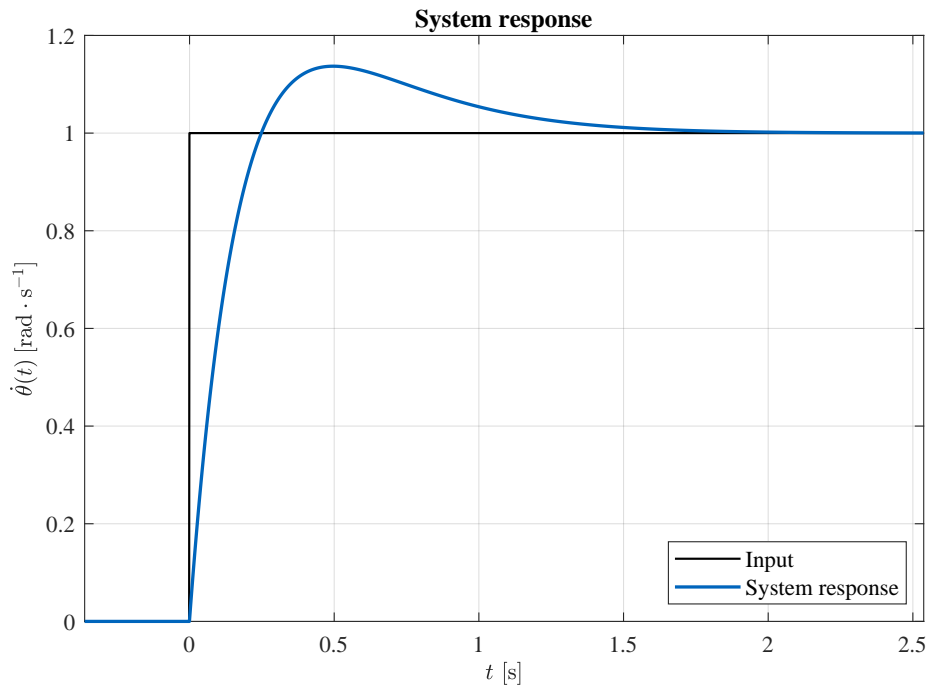


Figure 7.7: Pitch rate system response in horizontal mode

7.3.3 Yaw

In the horizontal mode, the yaw of the aircraft can be controlled by a differential thrust of the motors. However, I have decided not to use a feedback loop to control this rotation because, during a bank, the aircraft naturally starts to yaw in the direction of the bank, causing the aircraft to turn. The control loop would counter this yaw and increase the pilot's load.

■ 7.4 Transition mode

A transition mode combines control from vertical and horizontal modes. The rotors are tilted 45° forward, and the controls are mixed. So, the aircraft's roll will be controlled by ailerons and differential thrust, the pitch by elevator and symmetric rotor tilt, and the yaw by differential thrust and asymmetric rotor tilt. The aircraft operator can manually switch between the modes.

Chapter 8

Experiments

8.1 Tests in suspension

I have performed several experiments to verify and tune the calculated regulators. In the experiments to control the yaw rate, roll rate, and yaw, the aircraft was suspended from the ceiling to simulate a hover without the risk of causing damage to the model. The model was attached to the ceiling in the middle of the wing rod, which slightly changed some aspects of the dynamics of the model. A picture from this experiment can be seen in figure 8.1.



Figure 8.1: Indoor experiment

In free flight, the system rotates around the center of gravity, which should be located directly below the wing rod. In this experiment, however, the center of rotation is the attachment point above the center of gravity. This changes the system's roll behavior from marginally stable to a very underdamped stable system. Despite this minor change, this experiment simulated the dynamics credibly and was used to tune the regulators for roll rate and roll. The dynamic of the yaw rate of the system is not affected by the change of rotation point because the length of the torque lever arm is unchanged.

Figure 8.2 shows a significant difference between the tilt of each rotor used to control the yaw rate of the aircraft.

The largest change in dynamic behavior renders the experiment unusable for simulating the dynamic behavior of pitch rate and pitch. Because the center of rotation is shifted into the center of the wing rod, the lever arm of the rotor force becomes 0 in any rotor tilt configuration. Therefore, the pitch rate and regulators could not be tuned using this experiment, so they had to be tuned in free flight.

Video from an experiment testing the yaw control is available at the following link: <https://youtu.be/d-yejsV6MJs>. and video <https://youtu.be/lpziTAxAoDw> shows an experiment of roll control.

8.2 Free flight

I have also conducted several experiments in free flight. At first, I attempted to take off from the ground, but I later shifted to a hand-launched take-off to limit the windup, which could accumulate before the aircraft lifted off the ground. The other reason was that I wanted to limit the destruction caused by failed attempts by performing the experiments in tall grass, and a conventional take-off was not possible in such environments. In the text below, some of the most notable experiments are described. During these flights, the flight controller logged 28 variables. However, in this chapter, only the data with interesting information is displayed.

Flight 1

In the first semi-successful flight, during the first seconds after take-off, the aircraft remained stable until I induced a change in control input, which caused the aircraft to oscillate in the yaw axis.

This experiment allowed me to measure valuable data and proved that a stable flight could be reached after tuning the control system. The yaw rate can be seen in figure 8.3. A video recording of this flight can be seen using the following link: <https://youtu.be/eXhuAVpNTA8>. This oscillation was probably caused by a different throttle setting used in this experiment compared to the test in suspension. The increased throttle required to take off increased the effect of differential rotor tilt, and the system became unstable. After this test, I decreased the proportional constant for the yaw rate to eliminate the oscillations and validated the new controller in a simulation.



Figure 8.2: Indoor experiment - yaw control

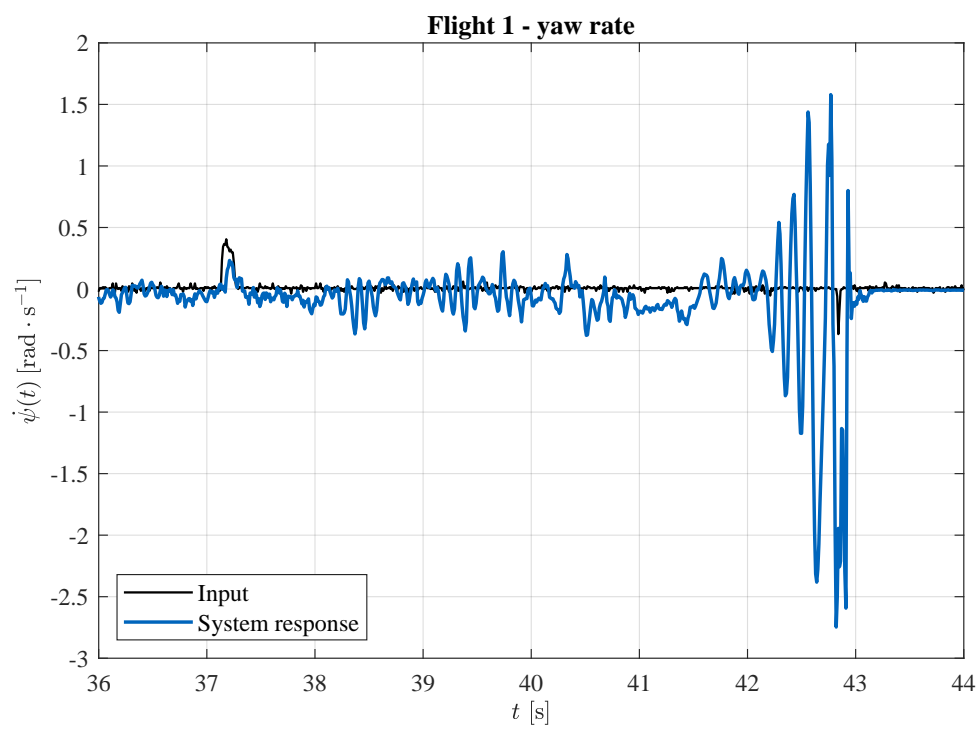


Figure 8.3: Flight 1 - yaw rate

Flight 2

In the next test, the location of the center of gravity was likely shifted backward, which caused the aircraft to pitch back immediately after the release. This problem was later solved by adding additional weights to the nose of the airframe. Figure 8.4 shows the pitch of the aircraft during the experiment and a recording of this flight can be seen here: <https://youtu.be/tJOPukzwjAg>.

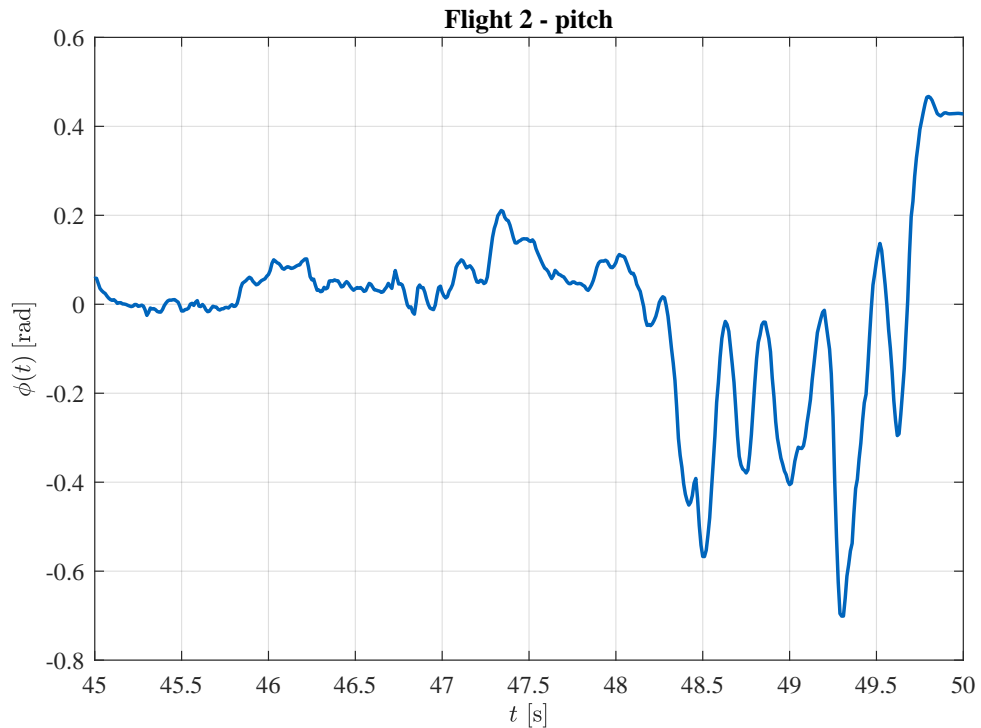
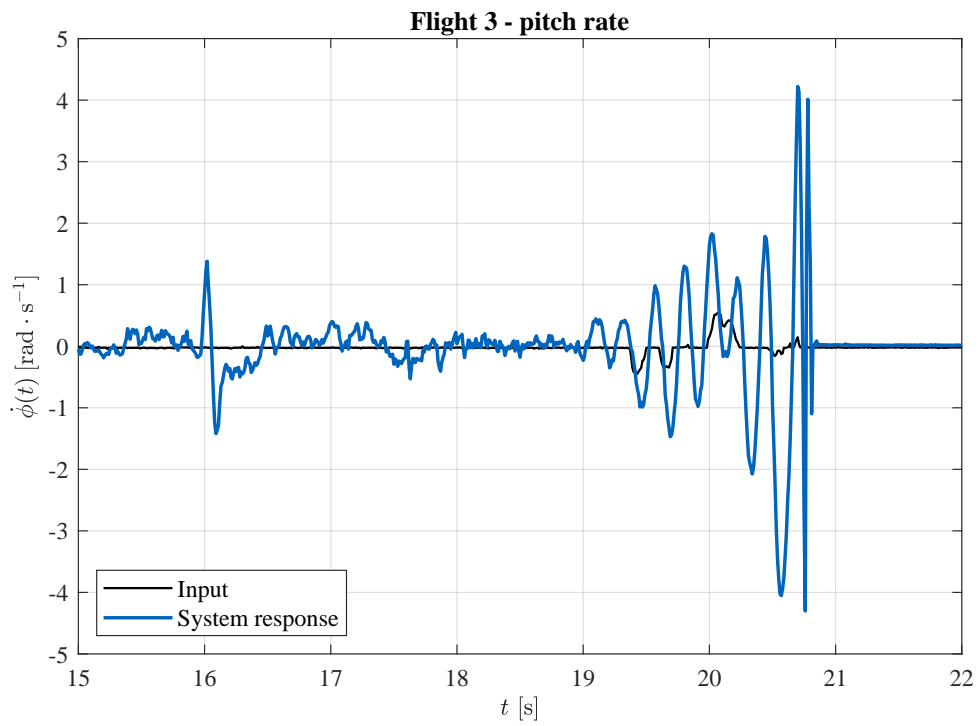


Figure 8.4: Flight 2 - pitch

Flight 3

The next test looked promising at the beginning, but during the experiment, problems with oscillations started to occur. This was probably caused by a large proportional constant of the pitch rate regulator in combination with a limitation of the tilting speed of the rotors. The pitch rate of the aircraft can be seen in figure 8.5 and a recording here: <https://youtu.be/RVyrV09XnUA>.

**Figure 8.5:** Flight 3 - pitch rate**Figure 8.6:** Flight 3 - outdoor testing

Chapter 9

Conclusion

In this work, I have designed and created a model of a tiltrotor convertiplane with functional rotor tilting mechanisms, a printed circuit board to house the flight controller and sensor and also provide connections to ESCs, servo motors, and an RC receiver. For this flight controller, I have written a functional software. I have derived a mathematical model which provides credible simulations of the system dynamics. This model was linearized and used to design feedback regulators to control the yaw rate, pitch rate, roll rate, and roll. The regulators were designed and tuned to achieve stable flight in simulations successfully. The regulator stabilized the aircraft roll and yaw axes but had problems with stability in the pitch axis.

9.1 Future

9.1.1 Further improvements

This project still needs improvements in the control system to achieve stable vertical and horizontal flight. But also further features can be implemented to expand the capabilities of the aircraft, such as:

Structural durability

Currently, the aircraft is very fragile despite several measures taken and described in chapter 2. Almost every experiment results in partial destruction and significantly prolongs intervals between experiments. Increased rigidity may be achieved using different materials or manufacturing processes.

Position control

After achieving stable flight, the next step would be to implement a regulator to achieve a position hold in vertical flight using the GPS data.

Horizontal flight analysis and aerodynamic improvements

After achieving horizontal flight, experiments with different wing sizes or types of airfoils could be performed to analyze the aerodynamic behavior and

increase efficiency.

■ Horizontal flight control

This thesis only suggested a control system for the horizontal part of the flight, but this system would have to be tested and further improved. This system could then be used for autonomous waypoint following.

■ 9.1.2 Potential use-cases

If it were possible to improve and finalize the model and also implement the listed features, the aircraft could be used in several applications.

■ Autonomous cargo delivery

Autonomous cargo delivery systems already exist, which either use quadcopter drones or fixed-wing UAVs have to drop cargo using parachutes. This solution limits the accessible areas and increases the risk of damaging or losing cargo. A tilt-rotor convertiplane solution could offer a combination of these systems with increased range and safer cargo delivery.

■ Aerial photography and monitoring

The increased range of the aircraft can be used to cover large areas, which would be beneficial for assessing crop and field conditions, power line and pipeline inspection, or wildfire monitoring.



Bibliography

- [1] United States Air Force. Transcendental model 1-g. [Online]. Available: https://commons.wikimedia.org/wiki/File:Transcendental_Model_1-G.jpg#/media/File:Transcendental_Model_1-G.jpg
- [2] Bell Helicopters Textron. Bell helicopter xv-3. [Online]. Available: <https://www.nationalmuseum.af.mil/Visit/Museum-Exhibits/Fact-Sheets/Display/Article/195785/bell-helicopter-textron-xv-3/>
- [3] Y. Osakabe. Bell boeing v-22 osprey. [Online]. Available: <https://www.af.mil/News/Photos/igphoto/2002483241/mediaid/5459888/>
- [4] AgustaWestland. Agustawestland aw609. [Online]. Available: <https://newatlas.com/agustawestland-aw609-tiltrotor/21466/>
- [5] W. Cushman. Bell v-280 valor. [Online]. Available: <https://simpleflying.com/bell-v280-valor-complete-guide/>
- [6] Smithsonian National Air and Space Museum. Vertol vz-2. [Online]. Available: https://airandspace.si.edu/collection-media/NASM-A19650279000_DSH01
- [7] L. Milagros. Lockheed martin f-35b lightning ii. [Online]. Available: <https://www.dvidshub.net/image/2095604/first-vertical-landing-f-35b-lightning-ii-march-18-2010>
- [8] “Naca 2414.” [Online]. Available: <http://airfoiltools.com/airfoil/details?airfoil=n2414-il>
- [9] Hitec. Hitec optic 6 sport 2.4 instruction manual. [Online]. Available: https://hitecrd.com/files/Manual_Optic6Sport2.4.pdf
- [10] “Nmea reference manual.” [Online]. Available: <https://www.sparkfun.com/datasheets/GPS/NMEA%20Reference%20Manual-Rev2.1-Dec07.pdf>
- [11] R. J. Woods, “Convertiplanes and other vtol aircraft,” *SAE Transactions*, vol. 65, pp. 150–158, 1957. [Online]. Available: <http://www.jstor.org/stable/44564361>

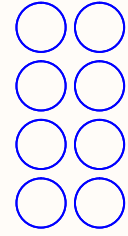
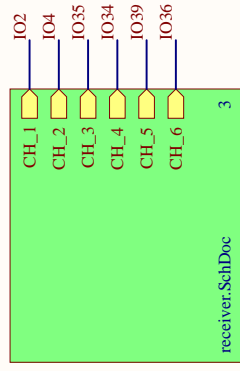
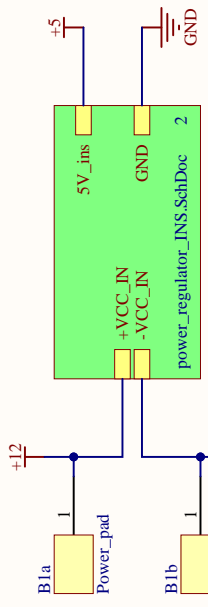
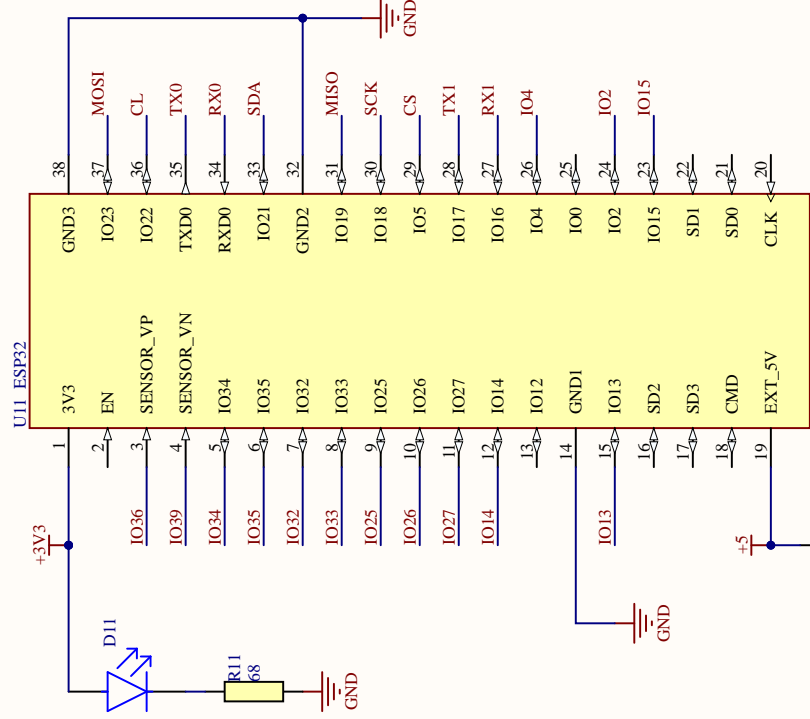
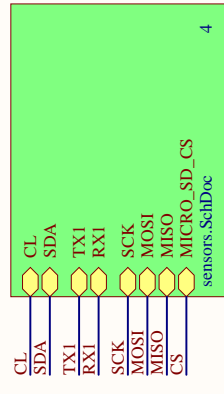
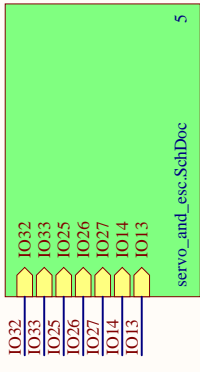
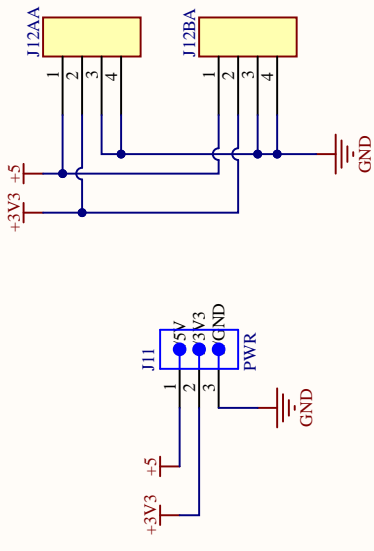
- [24] Wikipedia contributors, “Nasa gl-10 greased lightning — Wikipedia, the free encyclopedia,” 2023. [Online]. Available: https://en.wikipedia.org/w/index.php?title=NASA_GL-10_Greased_Lightning&oldid=1177518631
- [25] “Hawker siddeley - bae harrier.” [Online]. Available: <https://www.baesystems.com/en/heritage/hawker-siddeley-harrier>
- [26] S. Fox, “F-35 performs its first fully vertical landing,” Mar 2010. [Online]. Available: <https://www.popsci.com/technology/article/2010-03/f-35-completes-first-success-fully-vertical-landing/>
- [27] Wikipedia contributors, “Lockheed martin f-35 lightning ii operators — Wikipedia, the free encyclopedia,” 2024. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Lockheed_Martin_F-35_Lightning_II_operators&oldid=1223844318
- [28] —, “Lockheed martin f-35 lightning ii — Wikipedia, the free encyclopedia,” 2024. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Lockheed_Martin_F-35_Lightning_II&oldid=1224762855
- [29] S. Baluta, Oct 2015. [Online]. Available: <http://www.starlino.com/power2thrust.html>
- [30] E. Systems, “Esp32-wroom-32e, esp32-wroom-32ue datasheet,” 2023. [Online]. Available: <https://www.espressif.com/en/products/modules/esp32>
- [31] G. Lente and K. Ósz, “Barometric formulas: various derivations and comparisons to environmentally relevant observations,” *ChemTexts*, vol. 6, no. 2, p. 13, Apr. 2020.
- [32] D. Simon, *Optimal State Estimation - Kalman, H infinity and Nonlinear Approaches*. John Wiley & Sons, Inc., 2006, p. 123–129.
- [33] T. Moir. (2021) Embedded kalman filtering. https://www.youtube.com/watch?v=mpbWQbkl8_g.
- [34] —, *Kalman Filtering*. Cham: Springer International Publishing, 2020. [Online]. Available: https://doi.org/10.1007/978-3-030-34839-7_13
- [35] B. L. Stevens, F. L. Lewis, and E. N. Johnson, *Aircraft Control and Simulation: Dynamics, controls design, and Autonomous Systems*. Wiley, 2016.
- [36] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback control of Dynamic Systems*. Pearson, 2020.
- [37] M. Šebek, “Regulátory,” Lecture, 2024. [Online]. Available: http://www.polyx.com/_ari/slajdy/Lec-ARI-09-Controllers.pdf

- [38] S. N. Deepa and S. Ganesan, “Design of pid controller for higher order continuous systems using mpso based model formulation technique,” *World Academy of Science, Engineering and Technology*, vol. 80, pp. 1274–1280, 08 2011.



Appendix A

Flight controller schematics

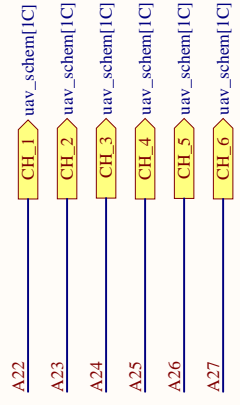


Title: Flight Controller
 Size: A4 Sheet 1 of 5
 Date: 11.03.2024 Location: uav_schem.SchDoc

Author: Marek Broul
 Organization: FEE CTU PRAGUE

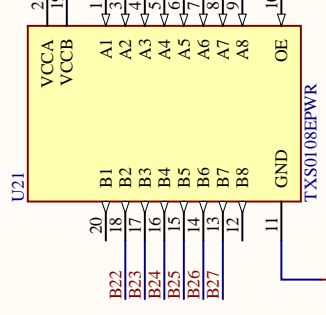
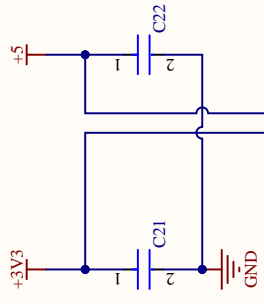
A

A



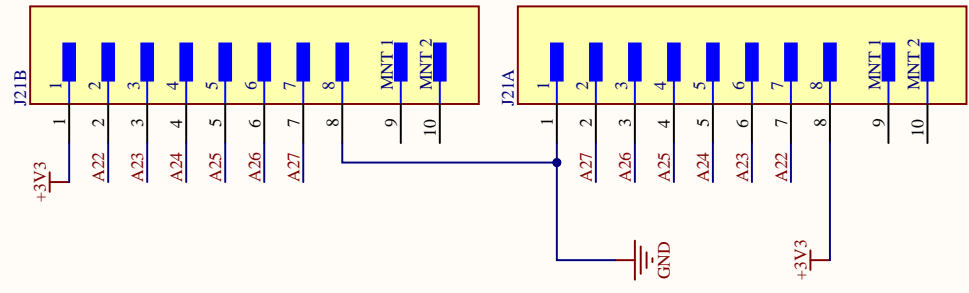
B

B



C

C



D

D



CTU
CZECH TECHNICAL
UNIVERSITY
IN PRAGUE

Title: Receiver

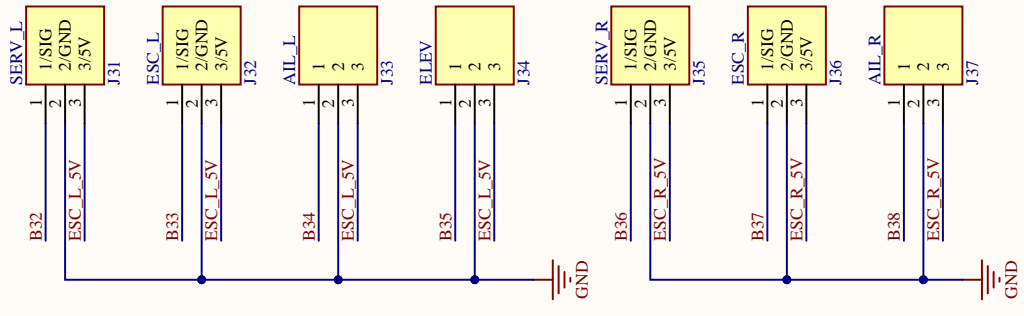
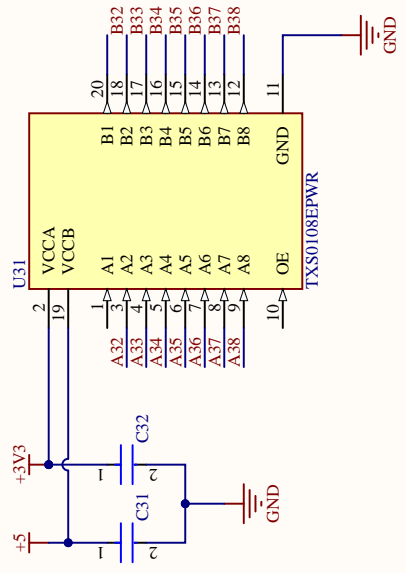
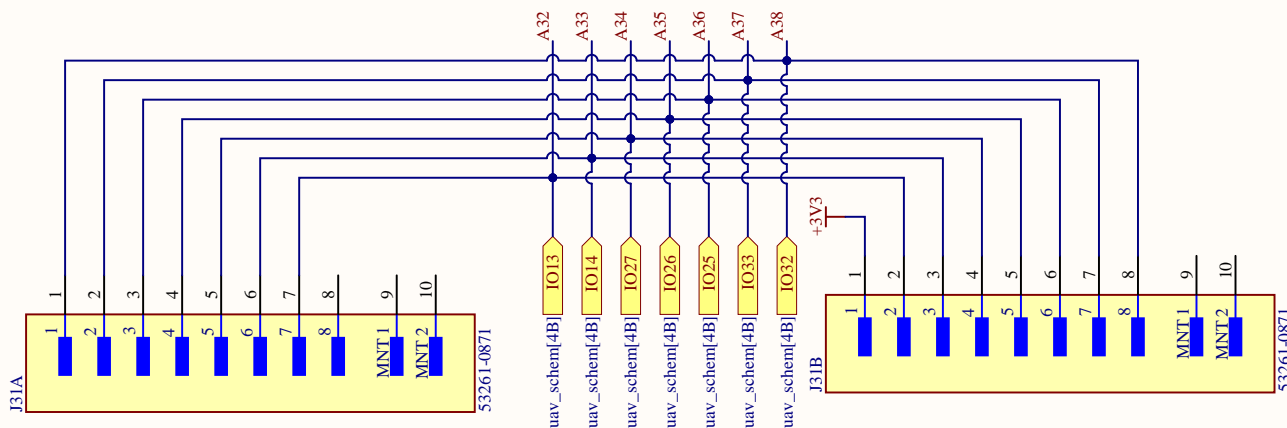
Author: Marek Broul

Size: A4 Sheet 2 of 5

Organization: FEE CTU PRAGUE

Date: 11.03.2024

Location: receiver.SchDoc



A

B

C

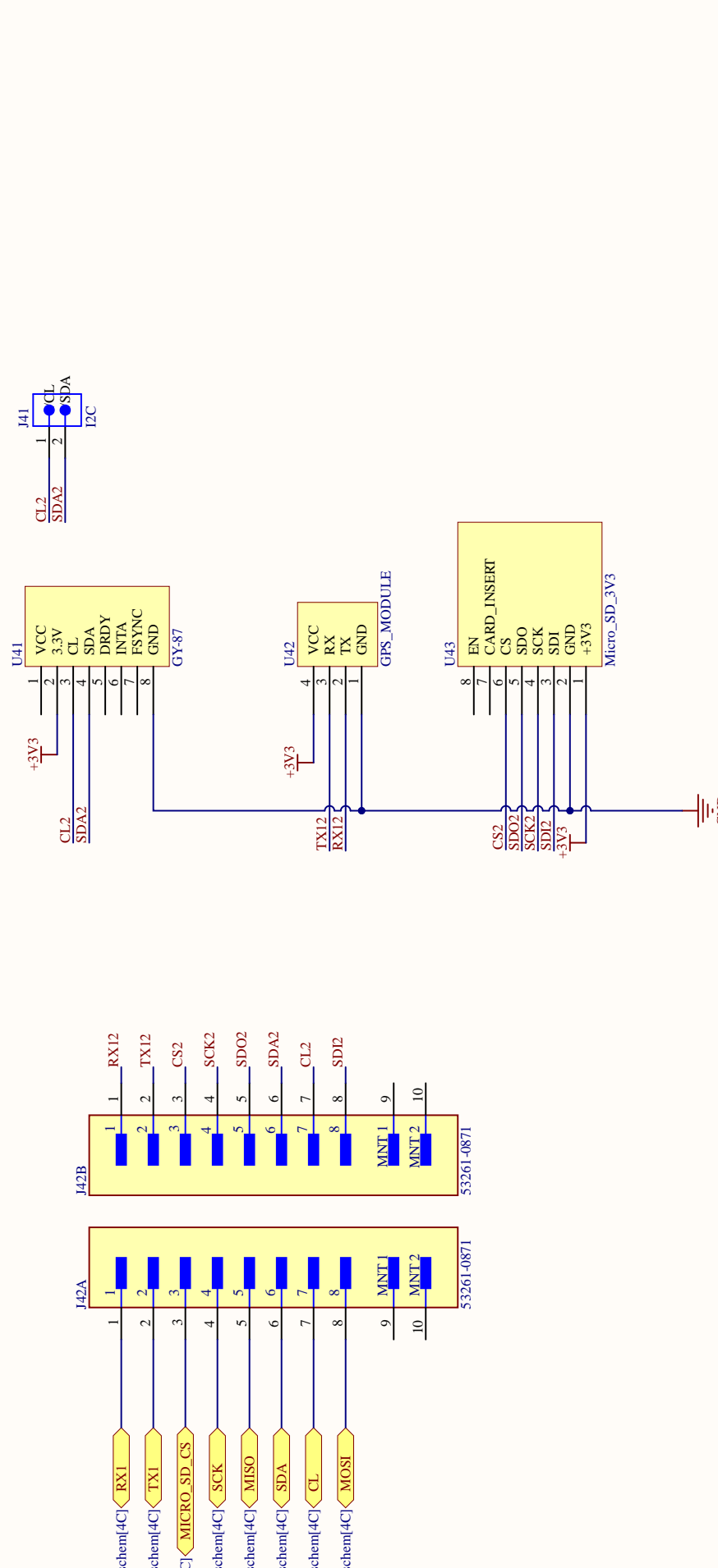
D



CTU
 CZECH TECHNICAL
 UNIVERSITY
 IN PRAGUE

Title: Servos and ESCs
 Size: A4 Sheet 3 of 5
 Date: 11.03.2024 Location: servo_and_esc_SchDoc

Author: Marek Broul
 Organization: FEE CTU PRAGUE



A

B

C

D



Title: Sensors Author: Marek Broul

Size: A4 Sheet 4 of 5 Organization: FEE CTU PRAGUE

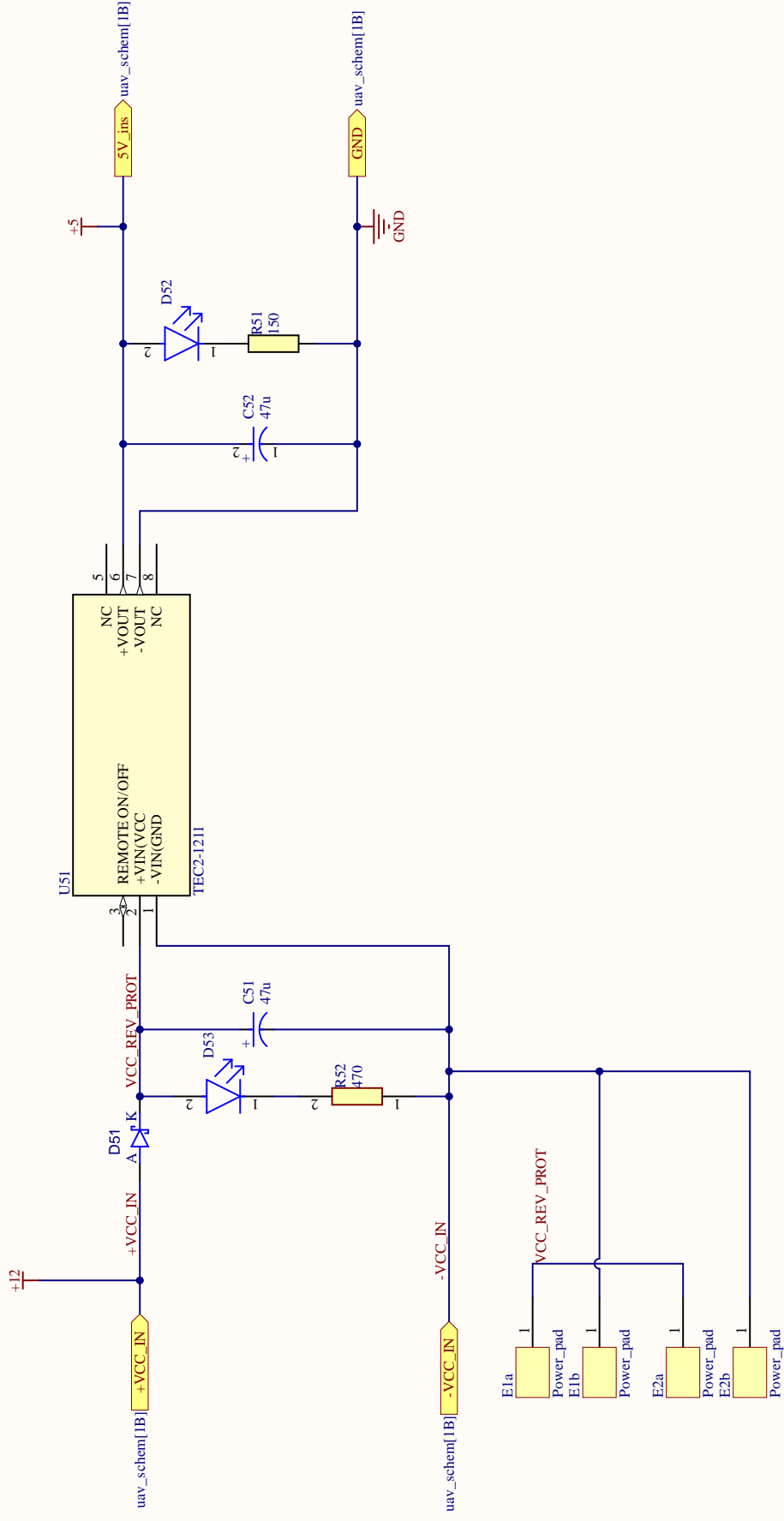
Date: 11.03.2024 Location: sensors.SchDoc

A

B

C

D



Title: 12 to 5 V regulator insulated		Author: Marek Broul	
Size: A4	Sheet 5 of 5	Organization: FEE CTU PRAGUE	
Date: 11.03.2024		Location: power_regulator_INS.SchDoc	

A

B

C

D