UNIVERSITY OF SOUTHERN DENMARK

Faculty of engineering

BEng Mechatronics

Mechatronics Semester Project 4

Ladybug

Supervisor:
Lars Duggen Davi Goncalves Accioli

Group: 7
Tobias Blaabjerg Karlsen
Artis Fils
Thor Uerkvitz
Choyon Mainul Hasan
Johan Paul
Ruta Miglava

Year 2023

1 Background

Possessing the ability of flight and minimising effort and casualties has always been desirable for the utility flight can provide. The first unmanned aircrafts can be dated back to 1849, where Austria seemingly had utilised unmanned air balloons with stuffed explosives to attack Venice. [?] Ever since an unmanned aircraft vehicle (UAV), is one that is flown by technological means or as a pre-programmed flight without pilot control, as defined by the ECAA Transport Agency [?], nowadays called drones, have risen in popularity.

Because of this, UAVs come in a wide range of sizes and weights. UAVs often include multirotor, radio-controlled miniature helicopters, and aeroplanes [?]. As a result, there are several methods to categorize drones. The performance parameters of UAVs, such as weight, wingspan, wing load, flight range, maximum flying altitude, speed, and production cost, are typically used to categorize them [?]. According to how the lift is produced, drones may also be divided into fixed-wing and rotating-wing types. According to the drone code category, the European Aviation Safety Agency (EASA) categorizes unmanned aircraft by weight. The EASA regulations for open categories, or drones without an EASA class designation, are summarized succinctly and simply in Figure 1 [?].

Self-built drones weighing up to 250 g, as described in Figure 1, may be used without registration if the drone is a toy or the drone is not equipped with a camera, the remaining drones must be registered, and the pilot must pass examinations [?]. In this paper, self-built rotary drones with four wings or propellers are the objective, making weight-based classification suitable.

UAS	Operation		Drone operator/pilot		
Max weight	Subcategory	Operational restrictions	Drone operator registration	Remote pilot competence	Remote pilot minimum age
< 250 g	A1 (can also fly in subcategory A3)	— No flight expected over uninvolved people (if it happens, overflight should be minimised) — No flight over assemblies of people	No, unless camera / sensor on board **and** the drone is not a toy	— No training required	No minimum age
< 500 g			Yes	— Read carefully the user manual — Complete the training and pass the exam defined by your national competent authority or have a 'Proof of completion for online training' for A1/A3 'open' subcategory	16*

Figure 1: Classification and restrictions for non-EASA class drones [?]

When it comes to the state-of-the-art project, PULP-DroNet is a deep learning-powered visual navigation engine that enables autonomous navigation of a pocket-size quadrotor in a previously unseen environment. Thanks to PULP-DroNet the nano-drone can explore the environment, avoiding collisions also with dynamic obstacles, in complete autonomy – no human operator, no ad-hoc external signals, and no remote laptop! This means that all the complex computations are done directly aboard the vehicle and very fast. The visual navigation engine is composed of both a software and a hardware part. [?]

When it comes to the future, the simulated pollination of agricultural plants by means of nano copter can provide collecting and delivering pollen in the mode of automatic control. A design of nano copter for pollination can be made on the basis of innovative modification of existing model by its reprogramming with regard to its flight controller that is to be fully adapted to computer interface. The robotic system is offered specially for artificial pollination in conditions of greenhouses and minor agricultural enterprises. [?]

2 Problem statement

The utility of smaller drones are immense, where it can be used in surveillance, toys and potentially to also be part of a swarm of drones. Although, there are smaller drones existing in the current market, we would like to challenge ourselves to build one ourselves, where certain goals ranging from functionality to budget are listed below.

2.1 Primary goals

- Net maximum weight of the drone is 250 grams. Weight under 250 grams ensures it falls under A1 category in EU regulations. 1
- Flight time of 20 seconds.
- Stress of the structural system should not exceed rupture point. System does not experience fracture.

2.2 Secondary goals

- Flight time of minimum one minute.
- Can land with acceleration less than 9.8 m/s2.
- Stress of the drone system should not exceed the yield point. System does not experience plastic deformation.
- Drone is remote controllable.
- Drone can fly in formation with another identical drone.
- Total production cost of the drone is under 500 DKK (Not including remote controller).
- Drone can play audio.

2.3 Constraints

- Budget for entirety of project is 2000 DKK.

- Time available to finish the project is 4 months.
- Drone should have a minimum hover time of 5 seconds.
- Drone should be fully functional and able to take off again after landing.
- No use of flight controller software or unmanned vehicle Autopilot software Suite, capable of controlling autonomous vehicles.

3 Test Specifications

3.1 Primary goals:

- To test this, the drone will be weighed with a scale of a precision on 0,1 grams.
- In order to test the flight time, a stopwatch will be started from the moment the drone leaves the ground and is stopped as soon as it lands.
- This goal will be the tested trough FEM, ensuring that the chosen material for the drones body, will not rupture.

3.2 Secondary goals

- This will be tested with the same method as primary goals tests point 2.
- This will be tested with a mobile phone, recording the drones landing, using the drones position compared to the timestamp of the video.
- This will be tested with the same goals as primary goals test point 3.
- This will be tested by the possibility of sending wireless signals to the drone, with the drone reacting to those send signals.
- This will be tested purely by ear, listening to the drones output.
- This will be tested by mobile phone video, looking at the drones positions at given timestamps.

- This will be tested trough summing the price for each single part, ensuring that it doesn't exceed 500 DKK.

3.3 Constraints

- This will be done with the same method as the secondary goals test, though ensuring the project cost is over 2000 DKK.
- To evaluate the time constraint point of the project, the goal fullfilments will be evaluated in the end of the project period. In the case that all primary goals are fulfilled, the constraint is succeeded.
- This will be tested with a stopwatch, ensuring that the hover time is atleast 5 seconds.
- This will be tested with making the drone take off right after a landing, making sure that the drone is fully operational at the second take-off.
- This will fulfilled by not employing any of the aforementioned in the drone.

4 Risk Assesment

To ensure the safety under operation and development of the drone, a risk assessment is created for the drone project, based on the Fine & Kinney method [**?**].

#	Risk	Description	Actions to minimize risk
1	**Safety risks in a operational perspective**		
1.a	**Collision** [R=1.5]	Collisions by the drone flying into either items or persons	Ensuring that the drone is only operated in enclosed spaces or under restraints, so that the drone is unable to fly into unwanted areas.
1.b	**Wear/breakage of components** [R = 0.3]	E.g. by a propeller being operated in a broken condition	Inspecting the propellers and other vital components before every flight or test session.
1.c	**Battery charging malfunctions, fire** [R = 1.75]	Charging the battery with incorrect charger or under undesirable conditions	Ensure that the batteries are only charged with the correct charger.
2	**Safety risks concerning the development phase**		
2.a	**Collision** [R=1.5]	Collisions caused to malfunctions in sensor readings or adjustments of control software giving unwanted results	Ensuring that the drone is only operated in enclosed spaces or under restraints, so that the drone is unable to fly into unwanted areas.
2.b	**Wear/breakage of components** [R = 1.5]	Incorrect mounting of components causing breakage or failure	Checking components after mounting, ensuring correct mounting
2.c	**Battery charging malfunctions** [R = 1.75]	Charging the battery with incorrect charger or under undesirable conditions	Ensure that the batteries are only charged with the correct charger.
2.d	**Shorting** [R = 6]	Having unwanted connections in the produced PCB or a wire connection soldered onto an incorrect pad.	Checking all connections compared to the schematic and checking for unwanted shorts with a multimeter in continuity mode
2.e	**Cuts** [R = 0.1]	Getting cut by propellers	Never operating the drone unrestrained and close to people.

Figure 2: Motor control with MOSFET.

Marked in all the risks are the risk scores calculated by the Fine & Kinney method of Risk, Risk score = Probability * Exposure * Consequence. These risk scores are then compared in the risk clusters of the method, with five defined classes. After calculating the proposed risk ratings of the found risks, it's apparent that all risks fall under the lowest factor of the Fine & Kinney method of scores the: 'Risk; Perhabs acceptable category'.

5 GitHub Interaction

Each semester project has a main focus-point for educational purposes. The main focus-point for SPRO4 is control engineering which covers implementations such as IMU combined with programming and PID for the drone. This means a lot of the project development will be software related.

Working together in a large group for a single deterministic purpose can be cumbersome, so in order to better track progress, a GitHub repository has been made to share files and collaborate more effectively. GitHub requires little to no extra effort to implement and use, and will only require a small change in work culture to use correctly, potentially saving a lot of time.

The GitHub repository is an easy way to share and keep track of the software development progress, and gives many advantages when working with embedded systems.

5.1 Advantages of Using GitHub

- GitHub works by having one main code or branch which is shared between everyone. Users can then create their own branches and work on something individually, before it is sent back to the main. This means past revisions of work can be saved and reviewed by everyone, and changes can easily be reverted if a fundamental mistake occurs- or is spotted during development. Mistakes in the software are also easier to identify and can be solved without having to change the main program at all. This also means that changes can be made without interfering with other peoples workflow, giving a clear advantage when working/collaborating on similar problems in the software. [**?**] With GitHub, open source software becomes easily available and secure, while also being compatible with other GitHub repositories. This means GitHub can also be used to keep dependencies up to date for already existing projects, and provide live updates to multiple libraries and dependencies automatically and safely. [**?**]

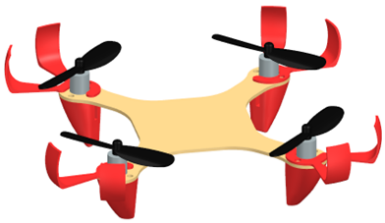
5.2 Usefulness Beyond the Project

GitHub is very standard among many engineering companies, and is one of the most widespread UML service. Its used by many companies in the industry, so its very useful to familiarize with GitHub before entering jobs, especially in software development. And with GitHub's wide usage and accessibility it has quickly become the worlds largest software collaboration tool in the world with over 100 million developers. [**?**] GitHub is also owned by Microsoft, and works with Microsoft programs, which is very standard for many work environments, providing high compatibility.

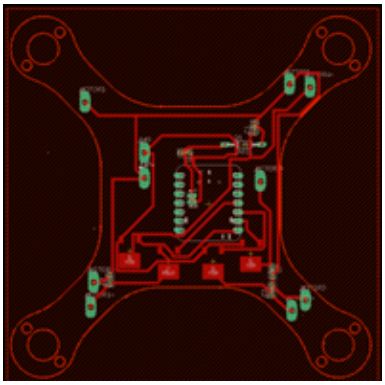
All in all, GitHub seems to be a necessary skill to have in the future when as an engineer.

6 Design and manufacturing of quadrotor

The main body of the drone is created as a PCB, in order for the body to host both the structural element and the electrical connections of the drone. Since the project formulation's main goals dictates the drones size and mechanical properties, the drones size was chosen to be $100mm^2$, so that the PCB could accommodate all the needed components of the drone. The PCB shape itself was designed in Siemens NX as a simple outline with holes for the motor mounts. This outline was exported in a DXF format into Autodesk Eagle were the actual PCB was designed from the needed electrical circuit. The electrical schematic was created in Eagle's schematic designer and then automatically converted to a PCB board with the auto-router feature. The trace width of the PCB is specifically chosen to be 30 mills so it can supply the needed 1.72 amps [**?**] for the motors at maximum rpm. The microcontroller of the drone is mounted by pin headers, the motors by the motor mounts and the remaining components by soldering onto the top layer. The position of the microcontroller is offset from the center, in order to ensure that the In-built IMU is centered to the exact middle of the PCB.



(a) CAD View



(b) PCB View

Figure 3: Drone model

In order to control the speed of the motors, one of the potential options would be to use an H-bridge to control the DC motors. However that would be excessive, as the motors does not need to run in reverse as contain many components As a less power consuning simple and lighter option, a single N-channel mosfet was used for each motor.

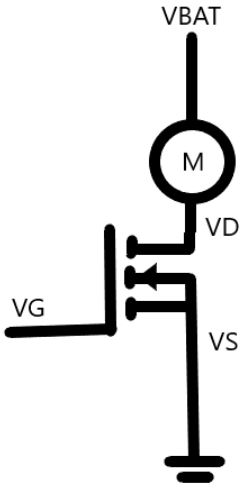


Figure 4: Motor control with MOSFET.

As illustrated in figure 4, the drain of the MOSFET is connected to the motor, which is supplied by the battery, and the source of the MOSFET is grounded. Meanwhile, the VG pins are connected to one of the PWM pins of the nRF52840 MCU, where the opening of the gate is proportional to the PWM. Thus, when VG (simulated by PWM) is smaller than the V threshold of the MOSFET, the motors are static, and if VG surpasses V threshold, then the motors start spinning with higher RPM as PWM is increased. The MOSFETs used for this situation are FDD8896 [**?**], as it has a low threshold voltage of 2.5V (MCU pins can supply up to 3.3V), and can handle up 94A in continuous drain current.

As there is high currents being drawn from the motors, one of the things needed to be configured was protection against overcurrent being drawn from the microcontroller. When the motors are switched on or off, a surge current arises which can result in additional overcurrent being pulled from the microcontroller if the battery is not alone capable of supplying the current.

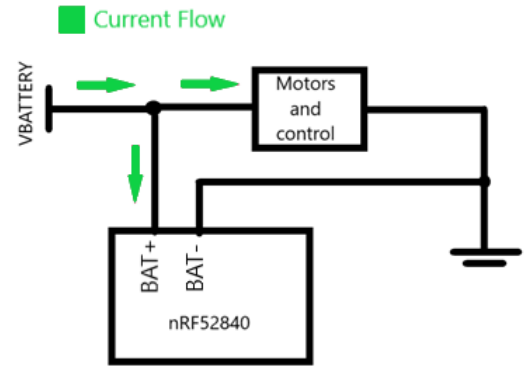


Figure 5: Current flow during continuous operation.

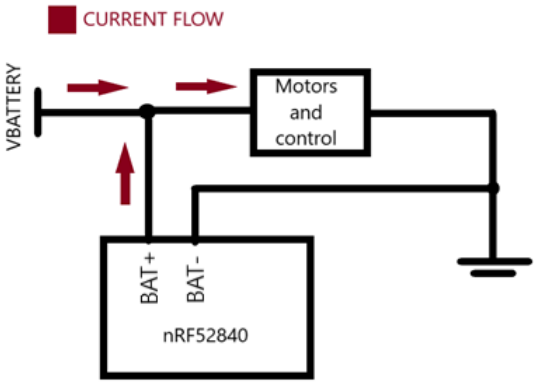


Figure 6: Potential current flow due to surge current.

In order to prevent reverse current from the MCU, during a surge, one of the potential options to utilise would be a diode placed in the following configuration.

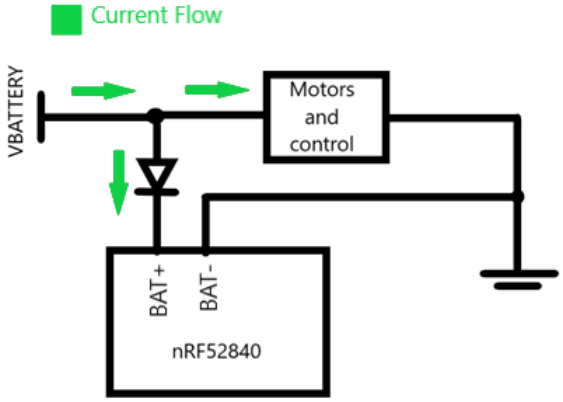


Figure 7: Diode added to prevent reverse current flow from MCU.

Additionally decoupling capacitors were added in parallel to the positive and negative motor pins and the MCU BAT+ and BAT- pins.

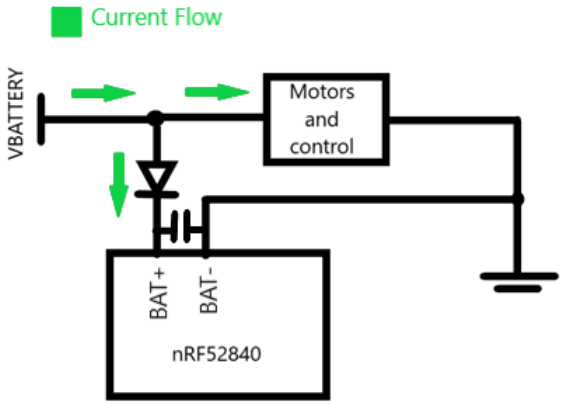


Figure 8: Example of decoupling capacitor for stable voltage to MCU.

The diode utilised is 1N5818 [**?**], as it has a low forward voltage of under 0.5V, resulting in low power losses. Moreover, the decoupling capacitors are cermaic capacitors as they generally smaller in size, and they are rated at 100 nF, which follows the general guidelines of decoupling capacitor values. [**?**]

With the combination of the CAD-outline and the electrical schematic, PCB Gerber files could be output for production. The only excess parts needed for the drone would be motor mounts and prop guards (prop guards solely for testing,

not for final use). The Motor mounts act as both landing legs and motor mounts. They would be designed in order for the motors to press fit into the central hole, mounted onto the PCB by screws into the motor mount going through the PCB. The designed mounting parts was printed in PLA on a Prusa MK3S+ 3D printer. The Parts were printed with a single perimeter and 3% infill in order for each leg to weigh 1.1 grams while still having the nescessary structure to have a stable press-fit.

(a) Motor Mount (b) Mounted Motor (c) Prop Guard

Figure 9: 3D-Printed drone parts

The project formulation states that the "Stress of the drone's structural system should not exceed rupture point. System does not experience fracture". This was tested through Ansys Mechanical Static Structural Analysis. Using the CAD model of the drone, the following boundary conditions were applied: Fixed constraint on the bottom area (excluding the drones arms), Z direction forces (upwards) was applied on each of the motor mount screw holes according to the found motor thrust, a single force vector in the Z direction (downwards) was applied to the top face (excluding the drones arms), to resemble the gravity force caused by the weight of the drone. Besides the boundary condition, the setup contained a mesh refinement of 3 steps, a material assignment of FR4 Fiber Glass Epoxy Board and a solution setting of equivalent stress. Below is a picture of the stress simulation of the initial design, which showed clear stress concentrations in the corners where the motor arms runs into the body section. The stress concentration had a magnitude of 1.6839 MPa which is satisfactory regarding the requirement of fracture, since the tensile strength of the given material is 320 MPa [**?**].

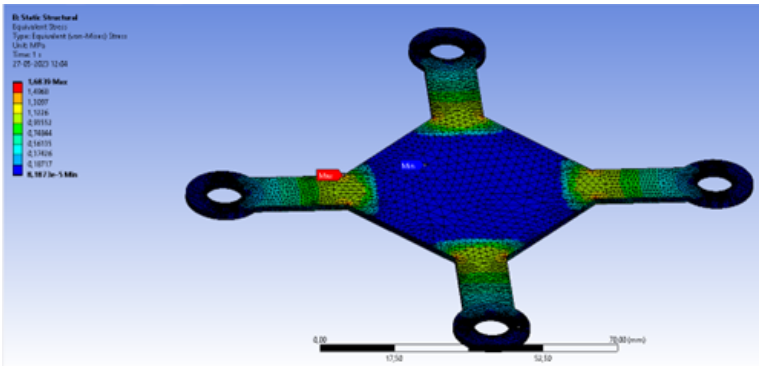


Figure 10: Initial Ansys Simulation

Although it was clear that the drone body would be strong enough to operate under max power, the stress concentrations were undesirable. Therefore 20 mm radii was added to the stress all corner points of the drone body. Repeating the simulation with the same boundary conditions, it was clear that the added radii removed the stress concentrations and instead distributed the stress over a broader area of the motor arms. Furthermore, it was found that the maximum stress was found to be 0.87154 MPa, which is approximately half the maximum strength of the initial model that had the stress concentrations.

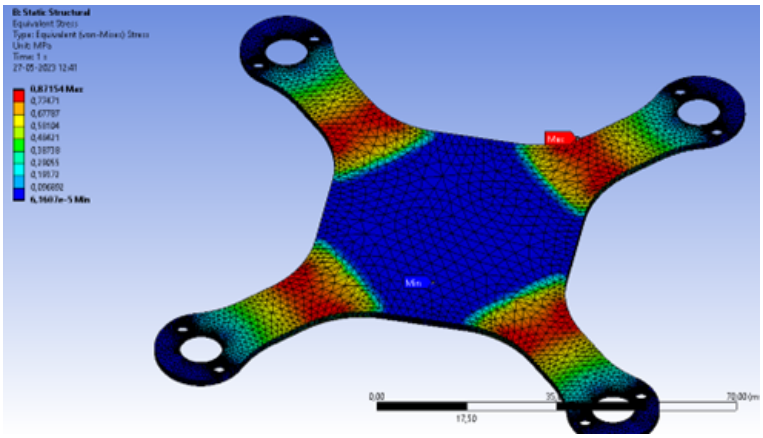


Figure 11: Final Ansys Simulation

To power the drone a single cell Turnigy Li-Po battery is used, at 3.7 - 4.2 V and 300 mAh. The battery is chosen specifically to be able to supply the needed current of the drone. With a C-rating of 35C, the battery is able to discharge

continously at 10.5 amps which is sufficent for the electrical circuit. The motors used are 8520 brushed Coreless DC-motors, which properties are characterized in the control chapter under thrust testing.

(a) Turnigy Battery (b) 8520 DC-motor

Figure 12: Motor and Battery used on the drone

7 Control prototyping

7.1 Plant Model Equations

This section of the report will explain how the plant model equations are obtained. Followed by how these equations are implemented in MATLAB/Simulink.

The plant model is a series of equations with multiple variables, which uses inputs, states, and parameters to get a certain output. For this project, the plant model is designed to take a height input the drone should hover at, paired with the inputs from the sensors, to achieve the desired hover height and stability.

7.1.1 State Equations

The drone possesses 6 degrees of freedom, enabling it to navigate in various directions and perform rotations. It can move along the X, Y, and Z axes, while also being capable of pitch (rotation around the Y axis), roll (rotation around the X axis), and yaw (rotation around the Z axis). We can identify the precise angle and position of the drone at any time by defining the axis of rotation and its accompanying state variables. [?]

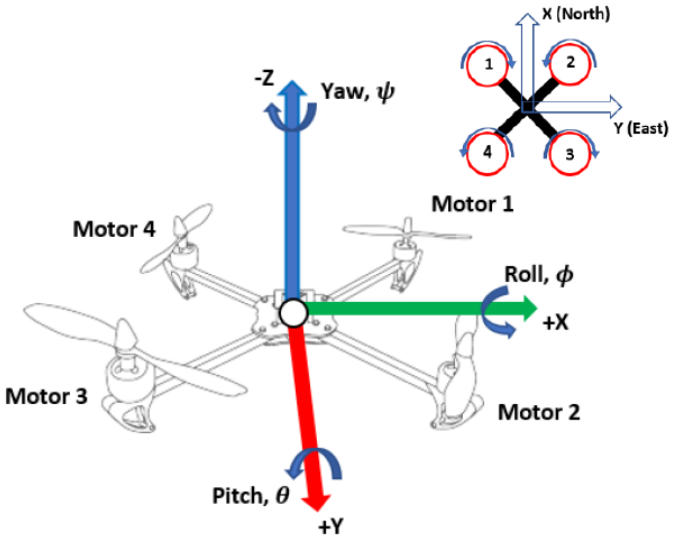


Figure 13: Quadcopter 'x' configuration

The drone's states include critical physical properties like as acceleration, velocity, and position, which are all detected by sensors. A gyroscope is used to record the angular changes of the drone. An accelerometer is a device that measures acceleration along the x, y, and z axes. Since the gyroscope drifts and the accelerator has measurement spikes when subjected to high acceleration, a complementary filter will be needed, which will be discussed in the "IMU" subsection. A dedicated infrared sensor is also used to measure distances in the z-direction, allowing for exact location information.

Symbol	Description	Unit	Observability
θ	Pitch Euler Angle	rad	Stereo Vision
$\dot{\theta}$	Pitch Euler Angular Velocity	rad/s	Gyroscope
$\ddot{\theta}$	Pitch Euler Angular Accel.	rad/s^2	-
ϕ	Roll Euler Angle	rad	Stereo Vision
$\dot{\phi}$	Roll Euler Angular Velocity	rad/s	Gyroscope
$\ddot{\phi}$	Roll Euler Angular Accel.	rad/s^2	-
ψ	Yaw Euler Angle	rad	Stereo Vision
$\dot{\psi}$	Yaw Euler Angular Velocity	rad/s	Gyroscope
$\ddot{\psi}$	Yaw Euler Angular Accel.	rad/s^2	-
X	Position in X	m	Stereo Vision
$\dot{X}$	Velocity in X	m/s	-
$\ddot{X}$	Accel. in X	m/s^2	Accelerometer
Y	Position in Y	m	Stereo Vision
$\dot{Y}$	Velocity in Y	m/s	-
$\ddot{Y}$	Accel. in Y	m/s^2	Accelerometer
Z	Position in Z	m	Stereo Vision
$\dot{Z}$	Velocity in Z	m/s	-
$\ddot{Z}$	Accel. in Z	m/s^2	Accelerometer
Ω_1	Motor 1 Angular Velocity	rad/s	Voltage/Current/Optical
Ω_2	Motor 2 Angular Velocity	rad/s	Voltage/Current/Optical
Ω_3	Motor 3 Angular Velocity	rad/s	Voltage/Current/Optical
Ω_4	Motor 4 Angular Velocity	rad/s	Voltage/Current/Optical
θ-Theta (/they-tuh/), ϕ-Phi (/fi/), ψ-Psi (/sai/), Ω-Omega (/oh-mega/)			

Figure 14: State table

Before plant model equations can be generated, parameters for the drone need to be calculated. The parameters are what explains the drone mathematically and is an integral part of modelling the drone. The more accurate these values are the more realistic the plant model will be. The parameters needed can be seen in the table below: [?]

Symbol	Description	Unit
I_{xx}, I_{yy}, I_{zz}	Inertia moments	$Kg\ m^2$
J_r	Rotor inertia	$Kg\ m^2$
l	Rotor axis to copter center distance	m
b	Thrust coefficient	N/s^2
d	Drag coefficient	Nm/s^2

Figure 15: State table

7.1.2 Inertia Moments

A moment of inertia is a constant value that explains the force/torque required to rotate an object in the direction of the moment. A desired angular velocity around a rotation axis can be found. The moments of inertia are found using estimation. If it is assumed that the drone is symmetrical in all axis, the moments of inertia can be found using parallel axis theorem to approximate the moments for each individual part of the drone in the x, y, z directions. For the calculations it's assumed that the centre of mass for the drone is perfectly in the centre of the drone's body. The equations used for each individual part of the drone can be found in. [?]

For the approximation smaller electrical components were not considered, such as wires and mosfets, since their collective total weight is less than a gram, they are deemed negligible for the accuracy estimation.

Motor MoI	$I_{motor,x} = 2\left(\frac{1}{12}m(3r^2 + h^2)\right) + 2\left(\frac{1}{12}m(3r^2 + h^2) + md_{a2a}^2\right)$ $I_{motor,y} = 2\left(\frac{1}{12}m(3r^2 + h^2)\right) + 2\left(\frac{1}{12}m(3r^2 + h^2) + md_{a2a}^2\right)$ $I_{motor,z} = 4(\frac{1}{2}mr^2 + md_{a2a}^2)$
Arms MoI	$I_{Arm,x} = 2\left(\frac{1}{12}m(w^2 + h^2)\right) + 2\left(\frac{1}{12}m(h^2 + d^2) + md_{a2a}^2\right)$ $I_{Arm,y} = 2\left(\frac{1}{12}m(w^2 + h^2)\right) + 2\left(\frac{1}{12}m(h^2 + d^2) + md_{a2a}^2\right)$ $I_{Arm,z} = 4(\frac{1}{12}m(w^2 + d^2) + md_{a2a}^2)$
Battery MoI	$I_{Battery,x} = \frac{1}{12}m(w^2 + h^2)$ $I_{Battery,y} = \frac{1}{12}m(h^2 + d^2)$ $I_{Battery,z} = \frac{1}{12}m(w^2 + d^2)$
Flight Controller MoI	$I_{FC,x} = \frac{1}{12}m(w^2 + h^2)$ $I_{FC,y} = \frac{1}{12}m(h^2 + d^2)$ $I_{FC,z} = \frac{1}{12}m(w^2 + d^2)$
Rotor moment of inertia	$J_r = \frac{1}{2}mR^2$

Figure 16: Inertia moments equations

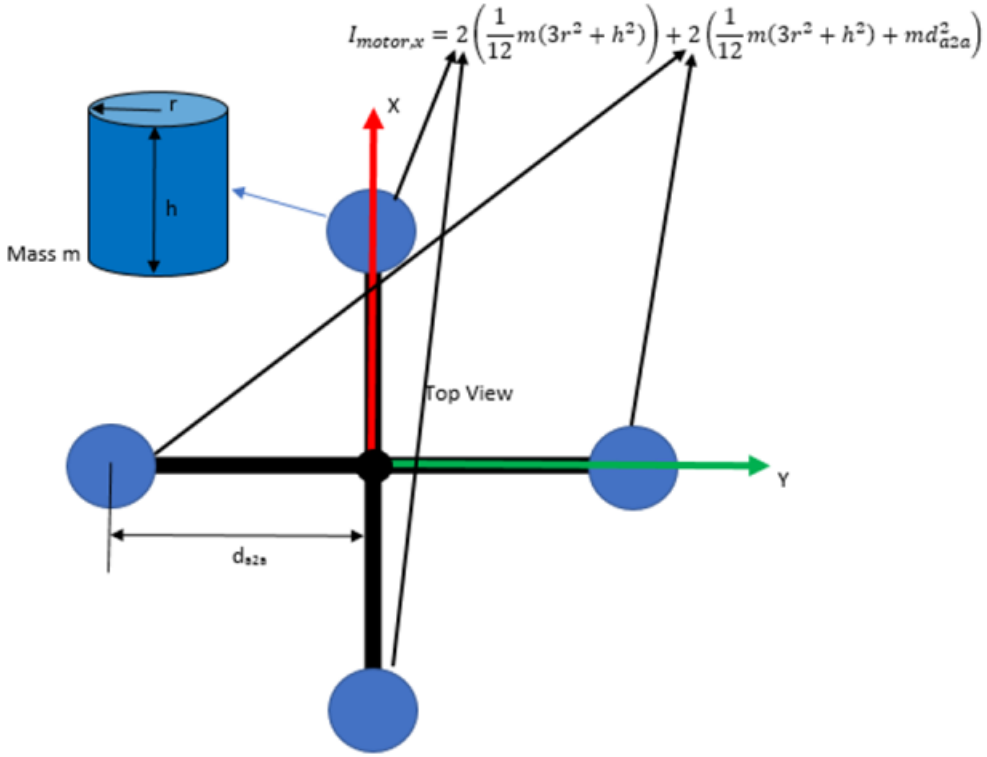


Figure 17: Top view of quadcoptor

A mathematical document can be found in the appendix as "SPRO4math". Which shows the detailed process of calculation. The mathematical equations are set up in maple, allowing quick changes to values in case a part is changed or removed. This feature proved useful since the weight estimate of the drone has changed rapidly over the course of the project period, as well as new components being added, which could then easily be integrated in the calculations.

$$I_{rotor} := 729$$

$$I_x := 48932.24391$$

$$I_y := 49929.46266$$

$$I_z := 99299.32284$$

Figure 18: Moments of inertia in the x, y, z axis, in $g * mm^2$ as calculated from equations, used in the final version of the quadcopter

7.1.3 Characterization of Motor

Some drone parameters can not be determined due to a lack of data concerning the motors used in the project. The motors are of a commercial product, but with no available data sheet. This means that motor constants and thrust coefficients needs to be determined through testing. Other motors within the same weight class were available, but those other alternatives did not provide the thrust desired compared to their weight, since the weight requirement set for the project is around 40 grams. For the same reason the motors currently in use are good, since they have a good thrust to weight ratio, with the caveat of not having a data sheet
The next part of the report is a small summary of the tests done to determine the data needed for the motors. The full reports concerning testing and acquiring these values can be found in the appendix.

- Thrust testing can be found in appendix "Motor Thrust Report"
- Motor constants testing can be found in appendix "Motor Testing"

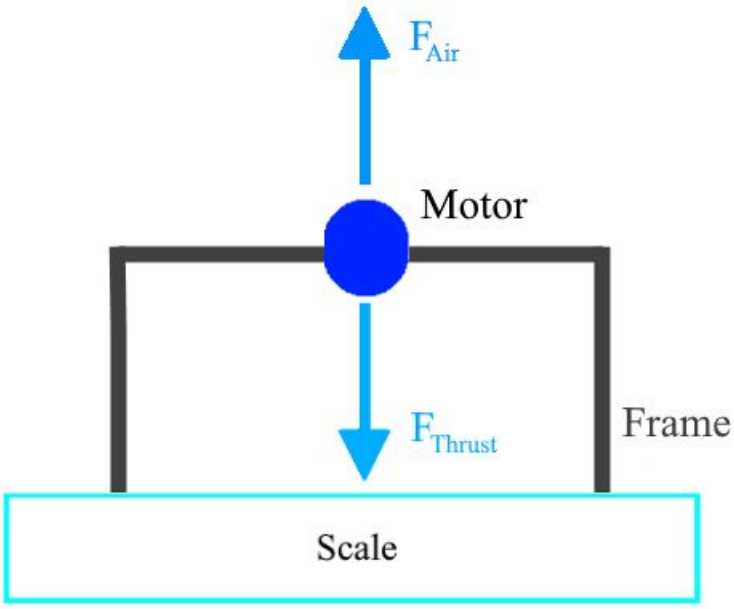


Figure 19: Thrust testing setup

7.1.4 Thrust testing

For thrust testing the motor is placed in a frame on a scale to measure the change in weight when the motor is operating. The motor used had an operating voltage window of 0-4,2 volts, therefore the motor was tested between 1-4,2 volts incremented with 0,1 volts, and the weight change noted for each increment. Characterizing the motor yielded the results seen in the figure below. As a general rule of thumb, provided by project supervisors, the motor's need to produce double the thrust of the total weight of the drone to operate ideally. At maximum voltage level of the battery, a single motor produces 22.2 grams of thrust. In conclusion, at maximum voltage level the drone can ideally weigh 44.4 grams in total. The battery will slowly discharge meaning that it's impossible to always have max output, so it should be a little less than 44.4 grams in practice.

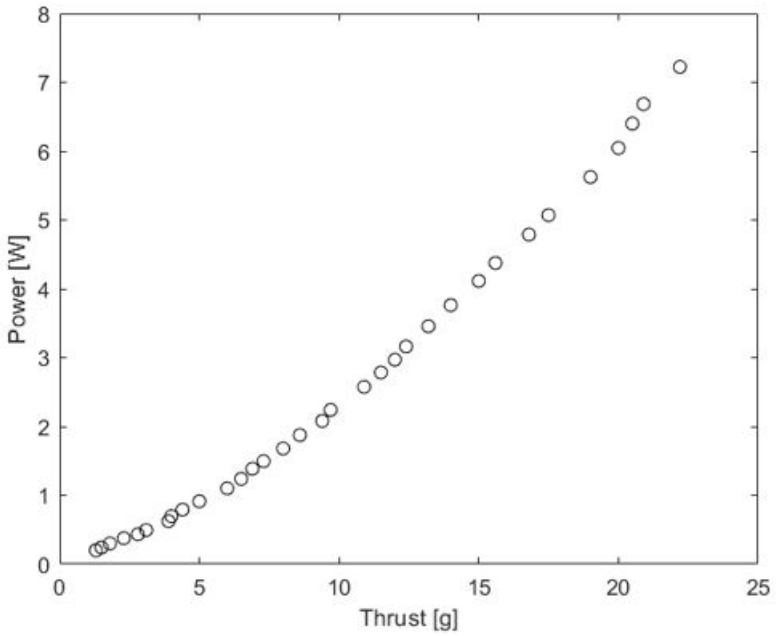


Figure 20: Thrust test data graphed

7.1.5 Motor Constants, and Thrust Coefficient

Motor constants relevant for this project include, "Motor velocity constant" (K_v) and "Motor torque constant" (K_T) With a simple setup the motor's RPM can be tested proportionally to PWM. The motor was tested in 20 increments, using a photo sensor to measure RPM. Using the data from the test, we get the following result. [?]

$$K_{\mathrm{v}} = \frac{\omega_{\text{no-load}}}{V_{\text{peak}}} \qquad K_{\mathrm{T}} = \frac{\tau}{I_{\mathrm{a}}} = \frac{60}{2\pi K_{\mathrm{v(RPM)}}}$$

Figure 21: Equations used for calculating thrust- and velocity constants

$$K_v := \frac{\omega}{V} = 8943.333333 \frac{RPM}{V}$$

$$K_\tau := \frac{60}{2 \cdot \pi \cdot K_v} = 0.001067755861 \frac{N \cdot m}{A}$$

Figure 22: Motor Constants

Combining the two test data sets it is possible to determine the thrust coefficient of the motor. Using the graph from the thrust test the thrust can be calculated by taking the voltage drop over the motor. This thrust is then divided with the angular velocity to get the thrust coefficient ($TC = 3.59E - 05$).

The constants/coefficients are calculated around 3.0V inputs, which is the voltage needed to achieve the theoretical hover thrust. This is sensible since most of the time the drone will be operating around these values.

7.1.6 Environmental Forces

The drone is very small, weighing only approximately 55grams as per requirement and with little surface area. Because of this, aerodynamic effects have been neglected for the modelling. Environmental forces mainly include but are not limited to wind resistance and liftoff turbulence.

7.2 Plant Model Equations

Now with the all the states and parameter defined, plant model equations can be generated. First, we will look at how the motors should operate together to move the quadcopter in the desired direction. A quadcopter can move in x and y directions using pitch and roll. When the quadcopter rotates around one of its axis, all the rotor blades will generate thrust at an angle, moving the drone in a direction. It is important to control the individual motors so the drone won't tilt too much, causing it to flip in the air. On the picture below the different cases of directional movement for the quadcopter can be seen.

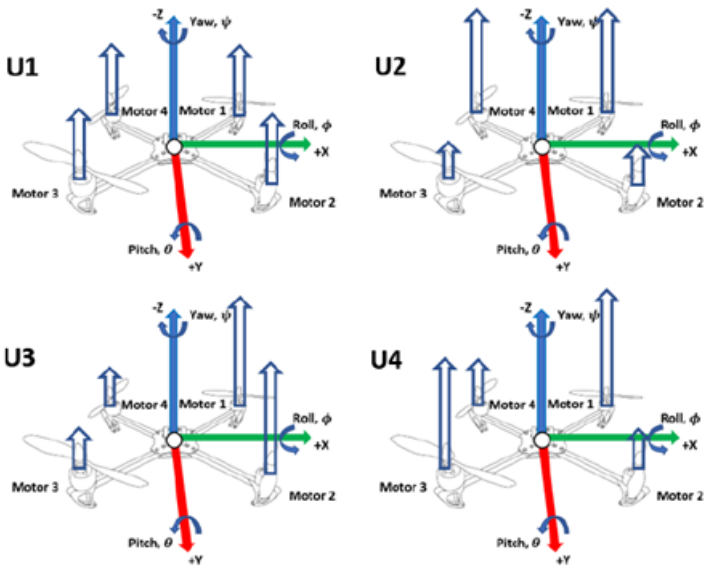


Figure 23: Visual representation of thrust equations [?]

Here U1 is the general thrust behaviour, providing thrust equally and symmetrically to move only the Z axis. U2 creates a roll, thrusting the drone in the Y direction. U3 creates a pitch, thrusting the drone in the X direction. U4 increases the thrust for motors turning the same direction and decreases the thrust of the motors rotating the opposite direction, allowing the drone to rotate in the Z axis e.g., yaw due to non-symmetrical rotational forces created by the moments of the motors. [?]

Each motors thrust is then defined as angular rotational speed to acquire the following equations.

Input	Thrust Equation	Description	
$U1_x$	$b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2)$	General Thrust	(2.8)
$U2_x$	$b \sin\left(\frac{pi}{4}\right)(\Omega_1^2 - \Omega_2^2 - \Omega_3^2 + \Omega_4^2)$	Roll Thrust	(2.9)
$U3_x$	$b \sin\left(\frac{pi}{4}\right)(\Omega_1^2 + \Omega_2^2 - \Omega_3^2 - \Omega_4^2)$	Pitch Thrust	(2.10)
$U4_x$	$d(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2)$	Yaw Thrust	(2.11)

Figure 24: Thrust equations [?]

The plant equations for this project were gotten from one of the sources. For a more in depth description of how the plant equations are found see [?]

State/Input	Dependencies	Equation
θ	$\dot{\theta}$	$\theta = \int \dot{\theta}$
$\dot{\theta}$	$\ddot{\theta}$	$\dot{\theta} = \int \ddot{\theta}$
$\ddot{\theta}$	$\dot{\phi}, \dot{\psi}, \Omega_r, U3$	$\ddot{\theta} = \frac{\dot{\phi}\dot{\psi}(I_{zz} - I_{xx}) - J_r\dot{\phi}\Omega_r + l(U3)}{I_{yy}}$
ϕ	$\dot{\phi}$	$\phi = \int \dot{\phi}$
$\dot{\phi}$	$\ddot{\phi}$	$\dot{\phi} = \int \ddot{\phi}$
$\ddot{\phi}$	$\dot{\theta}, \dot{\psi}, \Omega_r, U2$	$\ddot{\phi} = \frac{\dot{\theta}\dot{\psi}(I_{yy} - I_{zz}) + J_r\dot{\theta}\Omega_r + l(U2)}{I_{xx}}$
ψ	$\dot{\psi}$	$\psi = \int \dot{\psi}$
$\dot{\psi}$	$\ddot{\psi}$	$\dot{\psi} = \int \ddot{\psi}$
$\ddot{\psi}$	$\dot{\theta}, \dot{\phi}, U4$	$\ddot{\psi} = \frac{\dot{\theta}\dot{\phi}(I_{xx} - I_{yy}) + (U4)}{I_{zz}}$
X	$\dot{X}$	$x = \int \dot{x}$
$\dot{X}$	$\ddot{X}$	$\dot{x} = \int \ddot{x}$
$\ddot{X}$	$\psi, \phi, \theta, U1$	$\ddot{X} = \frac{(\sin\psi \sin\phi - \cos\psi \sin\theta \cos\phi)U1 - A_x\dot{X}}{m}$
Y	$\dot{Y}$	$Y = \int \dot{Y}$
$\dot{Y}$	$\ddot{Y}$	$\dot{Y} = \int \ddot{Y}$
$\ddot{Y}$	$\psi, \phi, \theta, U1$	$\ddot{Y} = \frac{(\cos\psi \sin\phi + \sin\psi \sin\theta \cos\phi)U1 - A_y\dot{Y}}{m}$
Z	$\dot{Z}$	$Z = \int \dot{Z}$
$\dot{Z}$	$\ddot{Z}$	$\dot{Z} = \int \ddot{Z}$
$\ddot{Z}$	$\psi, \phi, U1$	$\ddot{Z} = \frac{mg - (\cos\theta \cos\phi)U1 - A_z\dot{Z}}{m}$
Ω_r	$\Omega_1, \Omega_2, \Omega_3, \Omega_4$	$\Omega_r = \Omega_1 - \Omega_2 + \Omega_3 - \Omega_4$
$U1_x$	$\Omega_1, \Omega_2, \Omega_3, \Omega_4$	$U1_x = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2)$
$U2_x$	Ω_2, Ω_4	$U2_x = b\sin\left(\frac{pi}{4}\right)(\Omega_1^2 - \Omega_2^2 - \Omega_3^2 + \Omega_4^2)$
$U3_x$	Ω_1, Ω_3	$U3_x = b\sin\left(\frac{pi}{4}\right)(\Omega_1^2 + \Omega_2^2 - \Omega_3^2 - \Omega_4^2)$
$U4_x$	$\Omega_1, \Omega_2, \Omega_3, \Omega_4$	$U4_x = d(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2)$

Figure 25: Summary of plant equations [?]

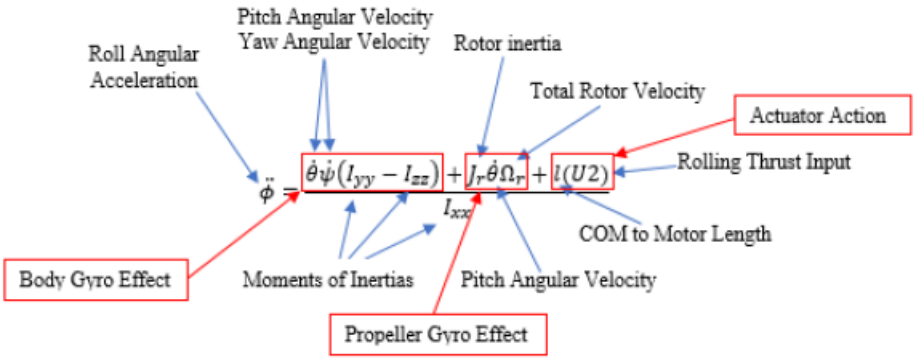


Figure 2.14-Rolling acceleration equation explained

Figure 26: Example of plant equation for phi. [?]

Body gyroscopic effects from changes in the quadcopter orientation. Propeller gyroscopic effects from propeller rotation and angle/orientation Actuation action forces produced by the rotors/propellers.

7.3 SIMULINK/Matlab Implementation

To make the Simulink model the thrust equations must be transferred into the program using the angular speed as inputs and the motor parameters. Below we see the thrust equations in Simulink with the inputs on the left and outputs on the right. (The outputs are U1, U2, U3, and U4, as well as Omega) In SIMULINK the math is done using "blocks", the addition and subtraction is performed first, and then the next blocks take the product. Essentially the math is converted using blocks in SIMULINK.

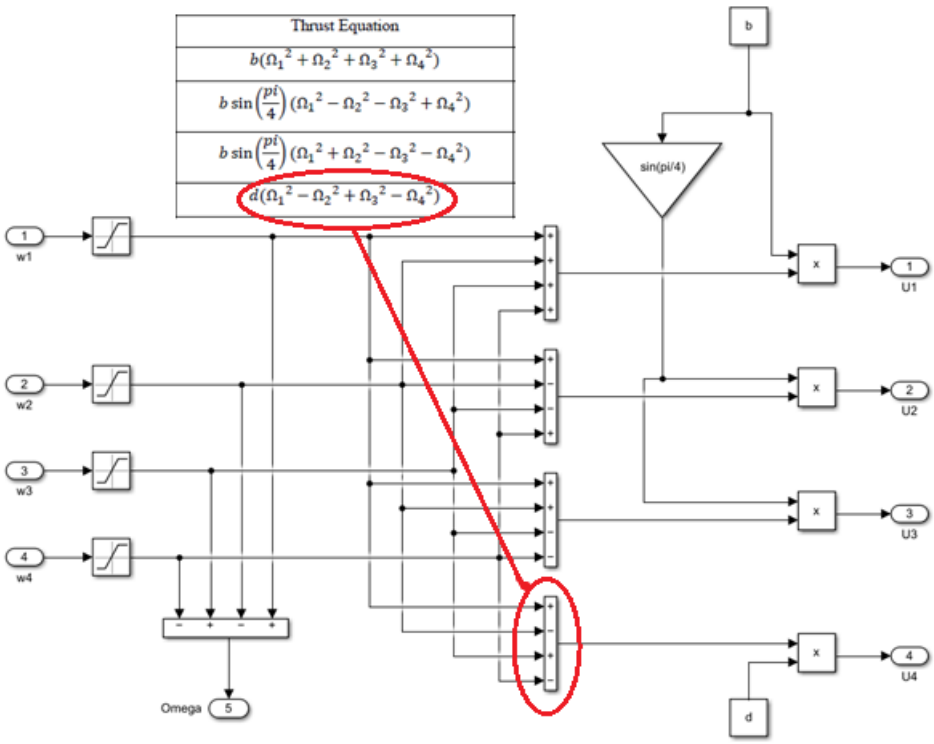


Figure 27: Thrust equations in SIMULINK

The rest of plant equations are then also transferred to SIMULINK using the same process, using blocks to represent the math. The next part of the plant takes the thrust equations as an input to calculate theta, phi, and psi, for position, velocity, and acceleration.

Below is an example for the equation of phi double dot. We use the thrust equations and feedback together with the drone parameters to get the result. Since all the equations need feedback from the other plant equations, an initial value must be set. It makes sense to make the initial value 0 since we haven't moved yet and there has been no change in any direction or angle. Throughout the flight the drone will then fill out the theta, phi, and psi values using the censors to determine the change in position. For example, if the drone is slanted a bit in the direction of motor 4 and 1, the drone will have a change in the roll aka phi value. The censor will detect the magnitude of this change, and U2 will increase to give more thrust to those motors to stabilize. Thrust is increased proportionally to smoothing the

stabilization.

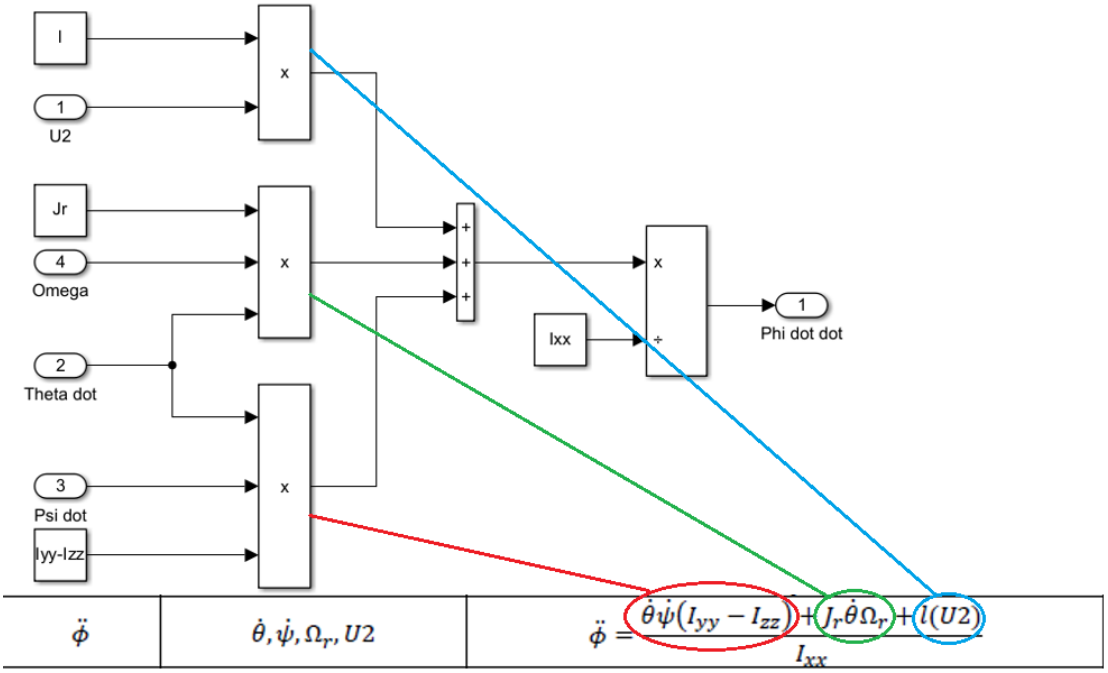


Figure 28: Plant equation for phi double dot

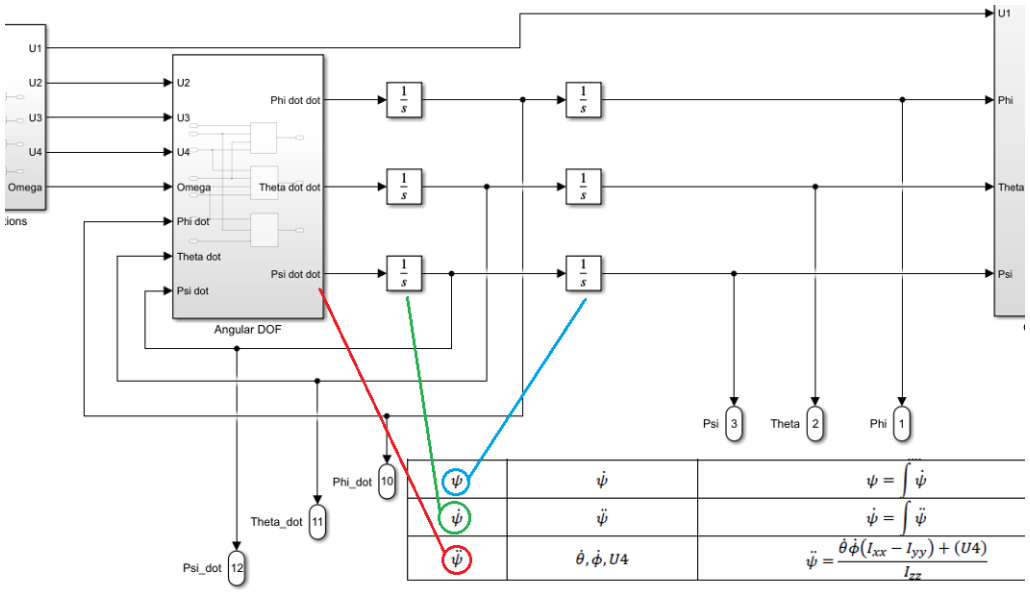


Figure 29: Angular equations overview example

To finish out the plant, the X, Y, and Z values for acceleration, velocity, and position are calculated using the values from theta, phi, and psi, these equations rely on feedback from each other in the same way that angles do, and use the same principles when it comes to determining the initial value.

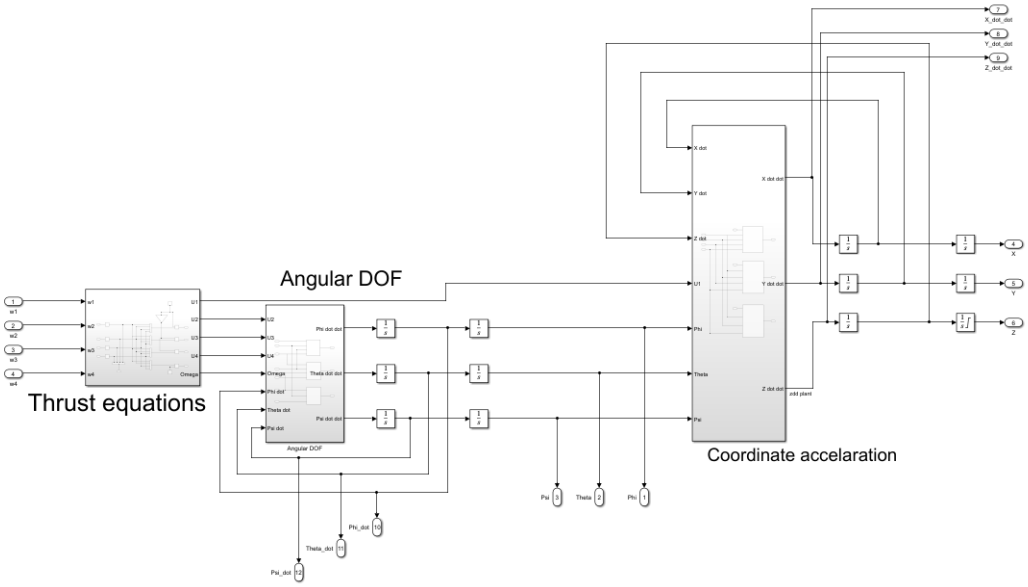


Figure 30: Plant overview in SIMULINK

7.4 UAV Toolbox

To achieve a visual way for checking if the quadrotor plant has been constructed correctly, the UAV Toolbox was used. (See Figure 31) For studying PID control, the MATLAB UAV 3D simulation provides a realistic and interactive environment. The simulation was used to examine the behavior of the UAV system rather than relying exclusively on theoretical concepts and mathematical equations. Experiments with different PID controller parameters were conducted, to see how they affect the quadrotor's reaction and acquire hands-on experience tweaking the controller. This immersive learning experience aid in the integration of theory and practice. Control performance in complex circumstances can be analyzed using the MATLAB UAV 3D simulation. When numerous PIDs are involved, it can become difficult to intuitively understand how one effects another; thus, a visual representation aids in grasping and understanding the concepts.

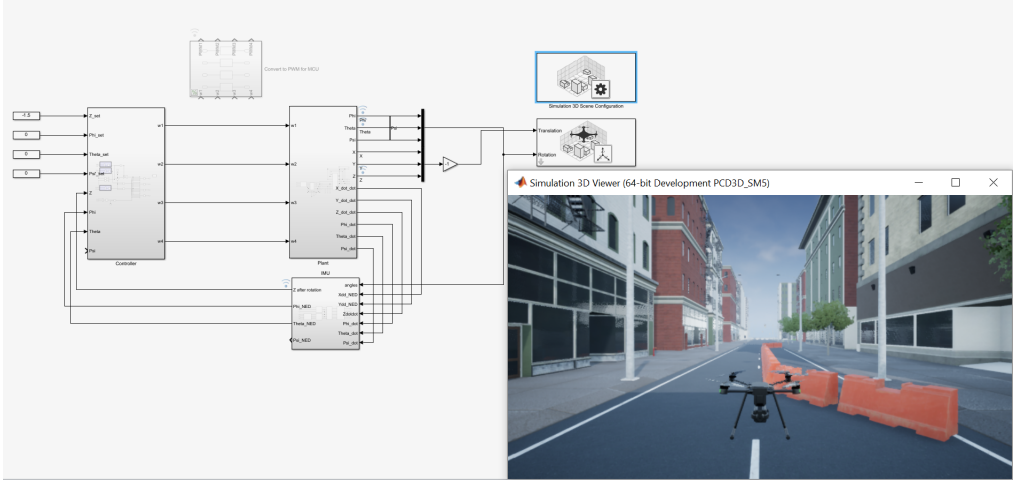


Figure 31: Ilustration of the workflow, with the utilization of the UAV Toolbox

7.5 The theory of PID

7.6 Principles of PID Control

PID (Proportional-Integral-Derivative) control is a widely used feedback control algorithm that plays a crucial role in various control systems.

PID control is based on three components: proportional, integral, and derivative. The proportional term provides control action based on the current error between the desired setpoint and the measured process variable. The integral term accumulates the past error over time to eliminate steady-state errors and enhance system stability. The derivative term calculates the rate of change of the error and helps dampen rapid changes, improving the response time and reducing overshoot.

7.7 PID Control Loop

The PID control loop consists of four main elements: the process or plant being controlled, in our case roll and pitch. A sensor or measurement device. The PID controller and the actuator. The IMU measures the process variable (angle), which is then compared to the desired setpoint. The PID controller calculates the control signal based on the error, and the actuator adjusts the system accordingly. This closed-loop feedback system continuously adjusts the control signal to maintain the process variable close to the setpoint.

7.8 Benefits of PID Control

PID control is a versatile and widely used control algorithm that provides effective control in a range of applications. There are other options, for example model predictive control, but those are less common and thus have smaller community.

Stability PID control helps maintain system stability by continuously adjusting the control signal based on the error feedback. The proportional, integral, and derivative terms work together to provide a balance between stability and responsiveness.

Robustness PID control is robust and effective in handling disturbances, noise, and external factors that may affect the system's performance. The integral term, in particular, helps eliminate steady-state errors caused by external disturbances.

Adaptability PID control can be tuned and adjusted to optimize performance based on specific system requirements. The controller gains can be modified to enhance stability, reduce overshoot, or improve response time.

Simplicity PID control is relatively easy to implement and understand. It offers a straightforward approach to control systems without requiring complex mathematical models or extensive computational resources.

7.9 Implementation

The performance of a PID controller depends on appropriate tuning to match the characteristics of the controlled system. Tuning involves adjusting the proportional, integral, and derivative gains to achieve the desired response. Various tuning methods, such as manual tuning, Ziegler-Nichols method, or model-based optimization, can be employed to find the optimal parameters for the specific system.

For the drone PID was used Ziegler-Nichols method, as it is compromise between complexity and time.

7.10 One-axis setup

Making PID for both roll and pitch at the same time is not ideal, as there are more variable introduced, therefore a rig to fix movement to one axis was constructed.

Ziegler-Nichols method for PID controller requires coefficient P value and period of an stable oscillation with purely proportional gain. From those are calculated coefficient with help of constants using the following formulas:

$$K_p = 0.6K_u$$

$$K_i = 1.2K_u/T_u$$

$$K_d = 0.075K_uT_u$$

Where 'u' denotes ultimate as in limit value before unstable oscillation.

Value for the drone were found out to be ...