# Quadcopter Control: Design and Simulation

By: Prahlad Chaudhary (17085054), Sarthak Choudhary (17085066), Shashank Kumar Singh (17085068)

***Abstract*— This paper presents an approach for the design, implementation, and simulation of a PID controller for a Quadcopter. Equations of classical mechanics are used in the mathematical modelling of the quadcopter kinematics. A Proportional, Integral and Derivative (PID) based control system is designed and implemented in the Simulink module of MATLAB. The designed controller is able to control the altitude of the vehicle and the roll, pitch, and yaw motion.**

***Keywords*—Quadcopter, Euler angles, altitude, roll, pitch, yaw, PID controller**

## I. Introduction

THE quadcopter, also known as a quadrotor, is a helicopter with four rotors. It consists of rotors placed in a square or cross formation at equal distances from the centre of mass of the quadcopter. The control of the quadcopter is done by adjusting the angular velocities of the rotors via motors. It has found use in multiple areas: surveillance, inspections, and search and rescue operations.

The quadcopter has a simple mechanical structure, but the control theory is complex due to the non-linear dynamics of the vehicle. It has 6 degrees of freedom (DOF) but only four motors, which makes it challenging to study. The basic tasks of a quadcopter (highlighted in this paper) are to achieve stable motion in all four directions and to achieve altitude control. The study of the quadcopter control problem stalled until recently because the development of the necessary control system was almost impossible without the advent of low cost and high-performance electronic devices such as microprocessors and microcontrollers, making quadcopters available for a large scale of applications. This paper provides a brief introduction to quadcopter kinematics and PID control. Simulation using the Simulink module of MATLAB and the tuning of PID controllers is also demonstrated.

## II. MATHEMATICAL MODELLING

The position of the quadcopter can be described by x, y and z coordinates, providing the respective distances from an origin fixed to the earth known as the inertial frame of reference. To represent the orientation or the attitude of the vehicle $\varphi$, $\theta$ and $\psi$ angles are used with respect to the body frame of reference which has its origin at the body's center of gravity. Here, we have assumed that the earth is flat and the force of gravity is constant. Additionally, the structure of the quadcopter is assumed to be rigid and its COM and COG are coincident.

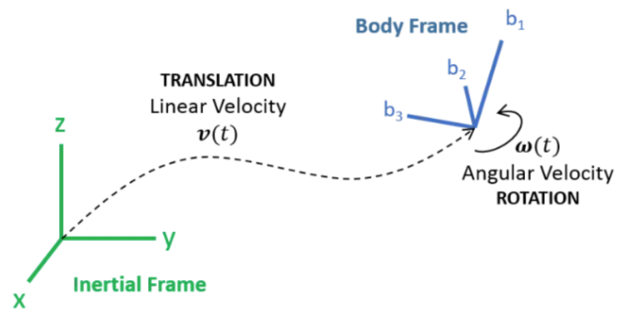


*Figure 1*

### A. Variables Used

| Symbol | Meaning |
|---|---|
| U1, $F_z$ | General Thrust |
| U2, $L_1$ | Roll Thrust |
| U3, $M_1$ | Pitch Thrust |
| U4, $N_1$ | Yaw Thrust |
| $J_r$ | Rotor Inertia |
| l | Rotor axis to center distance |
| $A_x$, $A_y$, $A_z$ | Coefficients of Air Resistance |
| $\phi$ | Roll |
| $\theta$ | Pitch |
| $\psi$ | Yaw |
| m | Total mass of the quadcopter |
| d | Coefficient of drag |
| b | Coefficient of thrust |
| $\omega_1$, $\omega_2$, $\omega_3$, $\omega_4$ | Angular velocities of motors |
| $I_{xx}$, $I_{yy}$, $I_{zz}$ | Moments of Inertia along the respective axis |

### B. Euler Angles

Using Euler angles we can find the final orientation of the vehicle with-respect to the body frame by using an inertial to body frame transformation matrix. It is convenient to use matrix multiplication for vector transformation. Let us say (x, y and z) is an inertial frame of reference and (b1, b2 and b3) is a body frame of reference. Rotation around one of the body frame axis results in the displacement of the other body frame axes. At the same time the rotation axis remains parallel to the corresponding inertial axis. Rotated body frame axes can be represented as inertial trigonometric function equations which can be later expressed in matrix forms. Similarly, we can get two other matrices by rotating other two body frame axes. Matrix equations (1) and (2) are used to convert the inertial frame to body frame and vice-versa.

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} \cos(\theta)\cos(\Psi) & \cos(\theta)\sin(\Psi) & -\sin(\theta) \\ -\cos(\phi)\sin(\Psi)+\cos(\Psi)\sin(\theta)\sin(\phi) & \cos(\Psi)\cos(\phi)+\sin(\theta)\sin(\phi)\sin(\Psi) & \cos(\theta)\sin(\phi) \\ \sin(\Psi)\sin(\phi)+\cos(\Psi)\cos(\phi)\sin(\theta) & -\sin(\phi)\cos(\Psi) \mid +\cos(\phi)\sin(\theta)\sin(\Psi) & \cos(\theta)\cos(\phi) \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad \textbf{(1)}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos(\theta)\cos(\Psi) & -\cos(\phi)\sin(\Psi)+\cos(\Psi)\sin(\theta)\sin(\phi) & \sin(\phi)\sin(\Psi)+\cos(\Psi)\sin(\theta)\cos(\phi) \\ \cos(\theta)\sin(\Psi) & \cos(\Psi)\cos(\phi)+\sin(\theta)\sin(\phi)\sin(\Psi) & -\sin(\phi)\cos(\Psi)+\cos(\phi)\sin(\theta)\sin(\psi) \\ -\sin(\theta) & \cos(\theta)\sin(\phi) & \cos(\theta)\cos(\phi) \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad \textbf{(2)}$$

### *C. Thrust*

Thrust produced by the propellers of the quadcopter acts perpendicular to the vehicle and moment acts at the centre of gravity of the vehicle.

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -F_1 - F_2 - F_3 - F_4 \end{bmatrix} \quad \textbf{(3)}$$

### *D. Gravity in Body Frame*

The force of gravity acting on the vehicle can be represented in the body frame using Euler angles. Transformation matrices can be used for inertial to body frame conversion and vice-versa. Below equation shows the force of gravity in an inertial frame of reference.

$$F_g^b = \begin{bmatrix} -mg\sin(\theta) \\ mg\sin(\phi)\cos(\theta) \\ mg\cos(mg\cos(\phi)\cos(\theta)) \end{bmatrix} \quad \textbf{(2)}$$

### *E. Moments of Inertia*

Moment of inertia gives the amount of moment needed to rotate a still object and moment needed to stop a rotating object. We need moments around all three axes of the vehicle. Moment of inertia is the square of distance from the centre of mass of the body.

$$I = \begin{bmatrix} I_{xx} & -I_{yx} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} \quad \textbf{(2)}$$

For a symmetrical body the moment of inertia on opposite sides of the vehicle cancel each other.

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad \textbf{(4)}$$

### *F. Mathematical Equations*

The system consists of desired values of z, roll, pitch, yaw as inputs and their actual values as outputs. The motor speeds depend on the desired values of the above parameters and are inputted to the Plant Model block. The four PID controllers take the difference between the desired and actual values for each parameter and reduce the error or the difference between the two. The control system of the entire model is as follows:

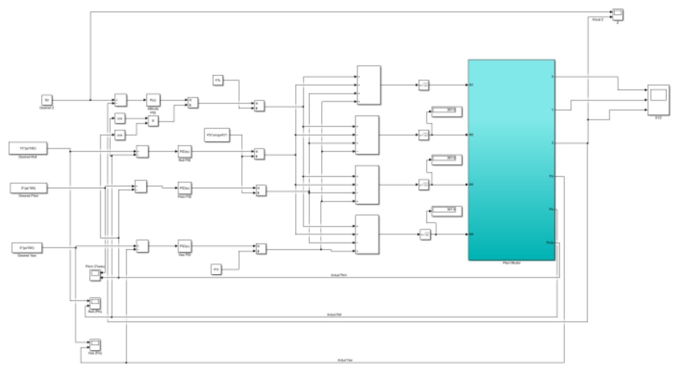

***Figure 2***

The equations for the PID controllers are as follows (as per Figure 1):

$$I = \frac{Output \quad of \quad PID(Z)}{cos\phi cos\theta} * \frac{1}{4b} \quad \textbf{(5)}$$

$$II = \frac{Output \quad of \quad PID(\phi)}{4bsin(\pi/4)l} \quad \textbf{(6)}$$

$$III = \frac{Output \quad of \quad PID(\theta)}{4bsin(\pi/4)l} \quad \textbf{(7)}$$

$$IV = \frac{Output \quad of \quad PID(\psi)}{4d} \quad \textbf{(8)}$$

The above equations are then translated into motor speed through the following equations

$$M_1 = \sqrt{I + II + III + IV} \quad \textbf{(9)}$$

$$M_2 = \sqrt{I - II + III - IV} \quad \textbf{(10)}$$

$$M_3 = \sqrt{I - II - III + IV} \quad \textbf{(11)}$$

$$M_4 = \sqrt{I + II - III + IV} \quad \textbf{(12)}$$

The motor speeds are then converted into the general thrust, roll thrust, pitch thrust, and the yaw thrust as per equations (13) to (16).

$$U_1 = b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \quad \textbf{(13)}$$

$$U_2 = b(\omega_1^2 - \omega_2^2 - \omega_3^2 + \omega_4^2)sin(\pi/4) \quad \textbf{(14)}$$

$$U_3 = b(\omega_1^2 + \omega_2^2 - \omega_3^2 - \omega_4^2)sin(\pi/4) \quad \textbf{(15)}$$

$$U_4 = d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \quad \textbf{(16)}$$

Using the parameters that have been calculated, the three angular accelerations are found out using equations (17) to (19).

$$\frac{\dot{\theta}\dot{\psi}(I_{yy} - I_{zz}) + lU_2 + J_rU_5\dot{\theta}}{I_{xx}} = \ddot{\phi} \quad \textbf{(17)}$$

$$\frac{\dot{\phi}\dot{\psi}(I_{zz} - I_{xx}) + lU_3 + J_rU_5\dot{\phi}}{I_{yy}} = \ddot{\theta} \quad \textbf{(18)}$$

$$\frac{\dot{\theta}\dot{\phi}(I_{xx} - I_{yy}) + U_4}{I_{zz}} = \ddot{\psi} \quad \textbf{(19)}$$

The actual values of roll, pitch and yaw are found out by twice integrating the above equations. This is easily achieved through the integrator block in Simulink.

The three linear accelerations are found out using equations (20) to (22).

$$(sin\phi sin\psi - cos\psi cos\phi sin\theta)\frac{U_1}{m} - \frac{\dot{X}A_x}{m} = \ddot{X} \quad \textbf{(20)}$$

$$(-sin\phi cos\psi + sin\psi cos\phi sin\theta)\frac{U_1}{m} - \frac{\dot{Y}A_y}{m} = \ddot{Y} \quad \textbf{(21)}$$

$$(cos\theta cos\psi\frac{U_1}{m} - g) - \frac{\dot{Z}A_z}{m} = \ddot{Z} \quad \textbf{(22)}$$

## III. Design Method

The simulation is performed in the Simulink module of MATLAB. The entire model is broken down into its constituent models to better visualize the working. The parent model is the Plant Model which takes the desired values of altitude (z) and the Euler angles and converts them into motor speeds (using equations (9) to (12)) and outputs the actual values of z and the Euler angles. This model corrects any error between the actual and the desired values using PID controllers (four in total, one for each parameter).

The plant model consists of three sub-models: Orientation and Motors model converts the motor speeds to the general thrust, roll thrust, pitch thrust, and the yaw thrust. Similarly, the Linear Acceleration and the Angular Acceleration Models output the respective accelerations based on the required equations.

These models are further divided into models for x, y, z accelerations and roll, pitch, yaw accelerations respectively.

The breakdown of the plant model follows:

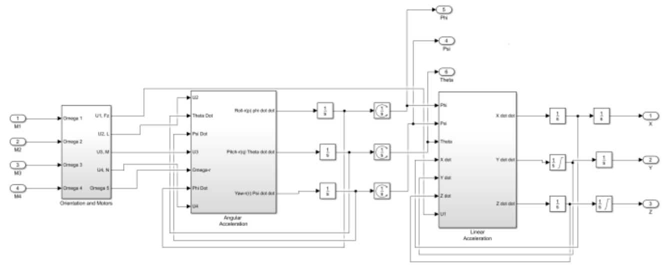

*Figure 3*

### A. Orientation and Motors

This block takes various motor speeds as inputs and outputs the various thrusts. Since the motor speed cannot be negative (we’ve considered the CW and ACW directions using positive and negative signs in the equations themselves), a saturation block is used which adds a lower limit of 0 to the motor speeds. The equations (13) to (16) are used to calculate the various thrusts.

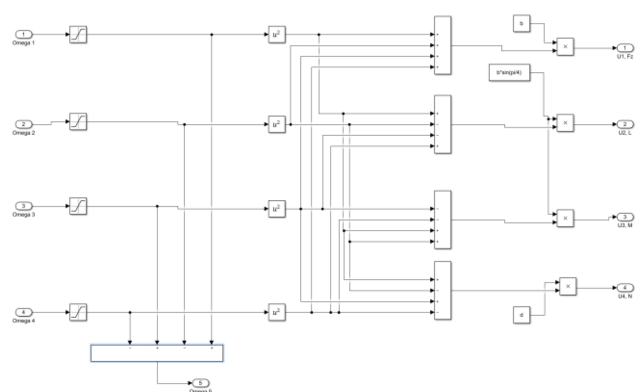

*Figure 4*

### B. Angular Acceleration

This block uses equations (17) to (19) to calculate the various angular accelerations. Since the input also requires the angular velocity (or the integration of the angular acceleration), the output is fed into an integrator block which then acts as the input.

The double integration of angular acceleration is angular displacement. This value can be more than 2π. However, we know that it repeats after 2π, so it’s better to limit its range to values between 0 to 2π. To achieve this, the second integrator block uses the wrap-state function and limits the output to the range (–π, π).

The equations (17) to (19) are used to calculate the angular accelerations.

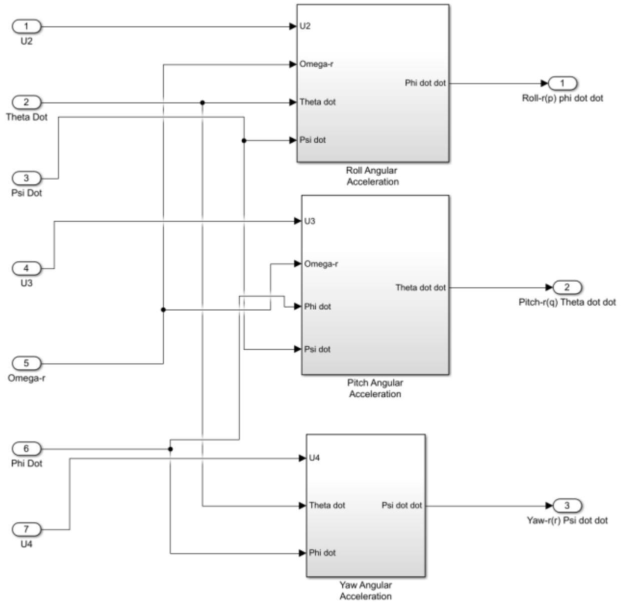


***Figure 5***

## C. Linear Acceleration

The inputs to this block are the three Euler angles, the three linear velocities and the general thrust $U_1$. The outputs are the three linear accelerations which are calculated using equations (20) to (22).

The second integrator block for Z-acceleration is a limited integrator, i.e. it defines a lower limit of 0 to Z since the quadcopter cannot have negative height.

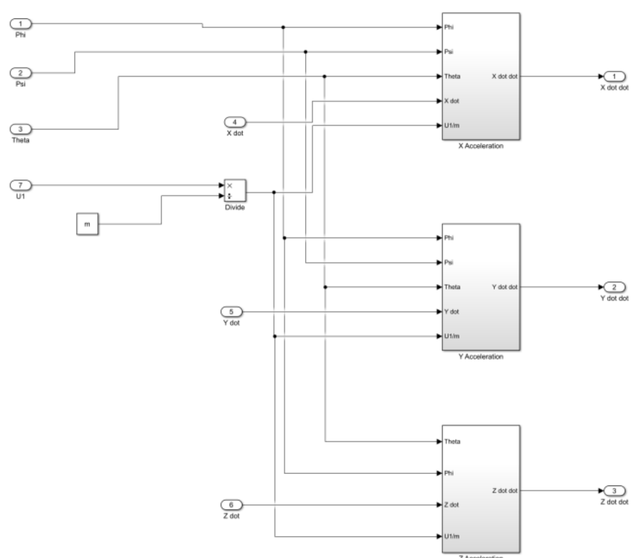


***Figure 6***

## IV. Simulation

The display block was added at various stages of the model to verify the output and correct any errors in the equations. The scope block is used to plot the graphs.

### A. Tuning the PID Controller

The PID controller can be tuned using the auto-tune function, however, this did not have the desired accuracy. Manual tuning was done instead.

1. Altitude PID Tuning

First, the I and D gains were kept at zero and then the P gain was tuned. The gain was kept low and increased until the actual value overshoots the desired value of 50m.

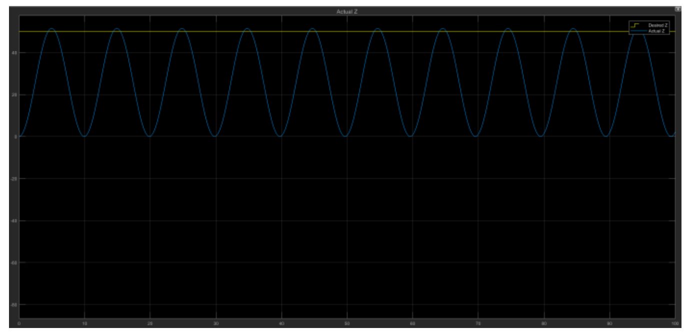

***Figure 7***

Then D gain is tuned to remove future errors. It led to a decrease in oscillations, and consequently, a smoother response.

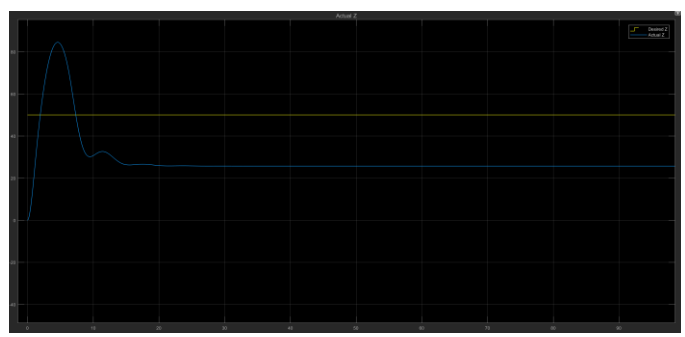

***Figure 8***

Now, the difference between the desired and actual values is due to the continuous error. This is reduced by tuning the I gain. Finally, the following output was obtained.

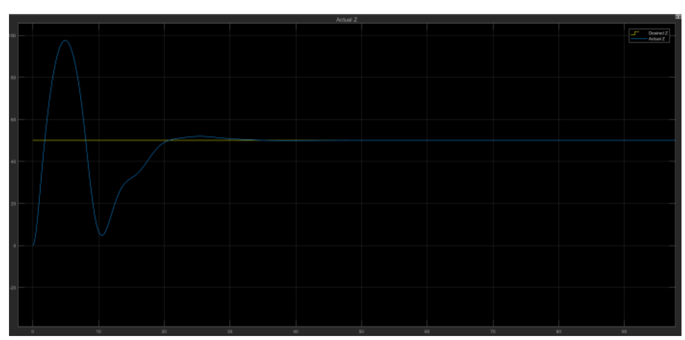

***Figure 9***

2. Roll and Pitch Tuning

The roll and pitch PIDs were tuned in the same fashion. A value of 5 degrees have been used as desired values. Figure 9 shows the Roll plot and Figure 10 shows the Pitch plot.

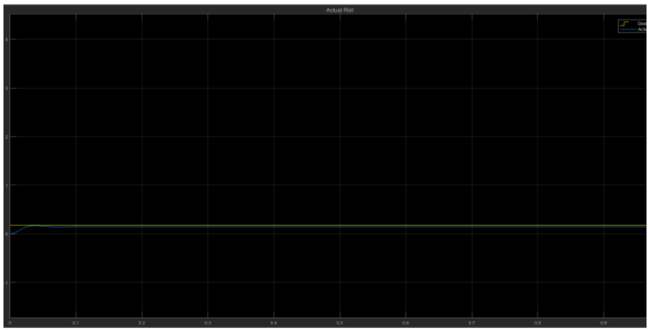

*Figure 10*

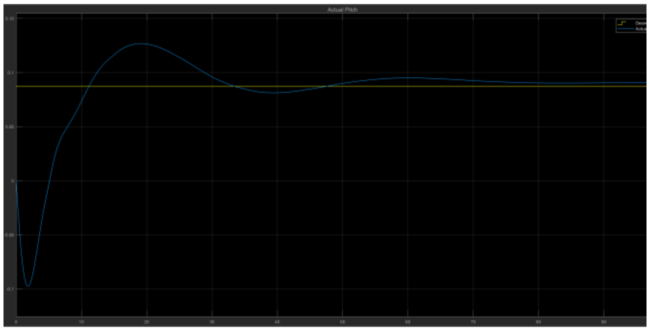

*Figure 11*

3. Yaw Tuning

The yaw depends on the torque produced by all four rotors, consequently, its tuning is challenging. The same procedure was used for tuning and a desired value of 5 degrees is used. The tuned response is as follows:

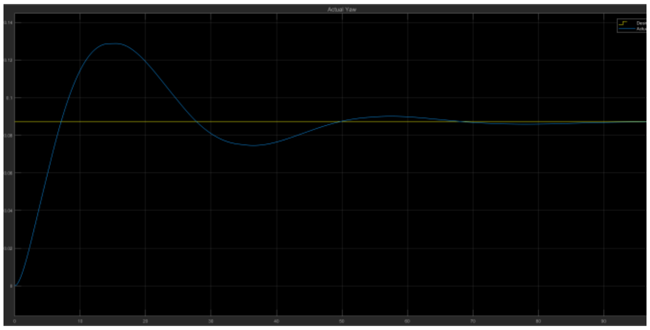

*Figure 12*

4. Response for Ramp Input

When ramp signal was used as input (slope=1) the following output from the Altitude PID controller was observed.

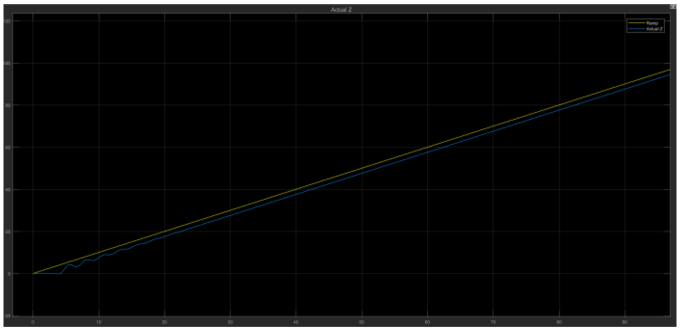

*Figure 13*

## APPENDICES

THE VALUES OF THE VARIOUS CONSTANTS USED ARE AS FOLLOWS:

| | |
|---|---|
| $J_R$ | 0.04439 $KGM^2$ |
| L | 0.24 M |
| $A_X$, $A_Y$, $A_Z$ | O.1 $KGS^{-1}$ |
| B | 3.59 X $10^{-5}$ KGM |
| D | 2.0810 X $10^{-6}$ $KGM^2S^{-2}$ |
| $I_{XX}$, $I_{YY}$ | 0.00963$KGM^2$ |
| $I_{ZZ}$ | 0.019 $KGM^2$ |

## REFERENCES

[1] Eswarmurthi Gopalakrishnan "Quadcopter Flight Mechanics and Control Algorithms". *Masters' Thesis, Czech Technical University, May 2016*

[2] Daniel Gheorghita, Ionut Vintu, Letita Mirea, Catalin Braescu "Quadcopter Control System: Modelling and Implementation" *19th International Conference on System Theory, Romania, 2015*

[3] Muhammad Usman "Quadcopter Modelling and Control with MATLAB/Simulnk Implementation". *Masters' Thesis, LAB University of Applied Sciences, Lappeenranta, 2020.*

[4] Ernesto Paiva, Juan Soto, Julio Salinas, William Ipanaque "Modelling, Simulation and Implementation of a modified PID Controller for stabilizing a quadcopter

[5] Muhammad Usman "Quadcopter Modelling and Control with MATLAB/Simulnk Implementation". *Masters' Thesis, LAB University of Applied Sciences, Lappeenranta, 2020. Page Number 10, Equations (2) and (3)*