Analysis of the Control Algorithm for Quadcopter Robots in Package Delivery Applications

Dr. Basit Memom
Assistant Professor, ECE
Habib University
Karachi, Pakistan
basit.memon@sse.habib.edu.pk

Ali Asghar Kerai
Electrical Engineering
Habib University
Karachi, Pakistan
ak06857@st.habib.edu.pk

Lakshman Maheshwari
Electrical Engineering
Habib University
Karachi, Pakistan
lm06435@st.habib.edu.pk

Muhammad Jazzel Mehmood
Computer Engineering
Habib University
Karachi, Pakistan
mm06886@st.habib.edu.pk

Hamad Abdul Razzaq
Computer Engineering
Habib University
Karachi, Pakistan
hr06899@st.habib.edu.pk

I. Abstract

In this paper the significance of package delivery robots will be discussed in the context of rapidly growing and the lack of resources to facilitate those. The theoretical modelling [1] of the Quadcopter beginning from defining its reference frames to defining the translational and the rotational dynamics helps us determine the closed form equations for all the motor speeds. After obtaining the model for the drone it was cross-referenced with the MATLAB example simulation that was used for this research [2]. Verification of the modelling from the literature and the dynamics equations used in the simulation files was done in order to implement the control algorithms [3] [1] for the path planning of the robot. Random objects were used to test the robustness and the responsiveness path planning algorithm, and the path following algorithm already implemented in the files were also tested with different parameters and discrete time stamp values to check whether they work under a different setup. The paper concludes with an analysis on the performance of the quadcopter model under different simulated environments and its ability to drop the package safely under those conditions.

II. Introduction

Drones that transport packages have been a game-changer in the logistics industry, providing faster and more efficient delivery times. Particularly in crowded metropolitan regions, these aerial vehicles are proven to be crucial in overcoming the conventional limitations connected with land transportation. Businesses may drastically cut delivery times, save operating expenses, and improve supply chain efficiency by using drones. Forward-thinking businesses like Amazon [4], UPS [5], and DHL [6] have made significant investments in drone technology in order to fully realise the potential of autonomous aerial delivery. These businesses see a day when drones are smoothly incorporated into regular logistical tasks, allowing for quick and accurate delivery, particularly in last-mile situations. Drone use helps cut down on carbon emissions from traditional delivery trucks while also streamlining the delivery process. Even though the sector is still developing, continued developments in drone technology and legal frameworks point to a bright future for package delivery drones, revolutionising the way items are carried and moved. This research is inspired by this massive potential for quadcopters to replace the traditional means of logistical deliveries. The aim of this research is to understand the mechanics of the a Quadcopter in depth to implement different control schemes and analyse what works better in a simulated environment where different obstacles are placed in the drone's way to simulate the real world scenarios. However due to the limited time available, this paper will focus on studying the theoretical model of the quadcopter, study its implementation in an already existing project, test it on some path, and to do any modifications if applicable to improve the performance of package delivery in any aspect.

III. Theoretical Model of the QuadCopter

In order to implement control algorithms for the Quadcopter in simulations, it is necessary to understand the dynamic model for such robots. As shown in Fig. 1, the quadcopter has 4 motors with propellers mounted on them for thrust situated at the end of each of the 4 arms. As per the convention in the figure, motors 1 and 3 rotate in the counter-clockwise direction while motors 2 and 4 rotate in the clockwise direction. The Dynamic model of the quadcopter can be based on two reference frames which are the Earth inertial frame(O) and the body fixed frame (B).

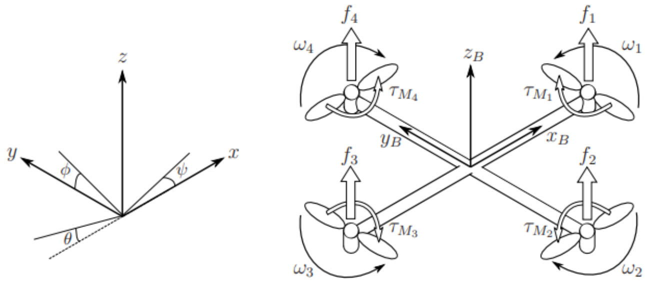

Figure 1: Model for a Quadcopter [7]

The absolute position of the center of mass of the quadcopter can be expressed in the earth inertial frame as;

$$\xi = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

The attitude/ angular position is defined in the inertial frame as;

$$\eta = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}$$

Where ϕ, θ & ψ are the roll, pitch and yaw angles respectively. The linear velocities velocities (V_B) and the angular velocities (v) are defined in the body frames as,

$$V_B = \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} \qquad v = \begin{bmatrix} P \\ Q \\ R \end{bmatrix}$$

The relation between the two frames can be expressed by the following rotation matrix from frame O to frame (B).

$$R = \begin{bmatrix} C_\psi C_\theta & C_\psi S_\theta S_\phi - S_\psi C_\phi & C_\psi S_\theta C_\phi + S_\psi C_\phi \\ S_\psi C_\theta & S_\psi S_\theta S_\phi + C_\psi C_\phi & S_\psi S_\theta C_\phi - S_\psi C_\phi \\ -S_\theta & C_\theta S_\phi & C_\theta C_\phi \end{bmatrix}$$

The inverse transform R^{-1} from frame B to frame O is just the transpose of R as the matrix R is orthogonal.

For a single motor based propeller system, the thrust acting on it is given by,

$$T = C_D \rho A r^2 \omega^2$$

Where C_D is the thrust coefficient, *pho* is the air density, A is the cross sectional area of the propeller's rotation, r is the radius of the propeller rotor. This can be simplified to,

$$T = K\omega^2$$

The net thrust acting on the robot in the body frame is given by,

$$T_T = K \sum_i \omega_i^2$$

The net thrust on the quadcopter is given as,

$$F_T = \begin{bmatrix} 0 \\ 0 \\ T_T \end{bmatrix}$$

The resisting drag force acting on the robot is given as,

$$F_D = \begin{bmatrix} A_x & 0 & 0 \\ 0 & A_y & 0 \\ 0 & 0 & A_z \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}$$

Where A_x, A_y & A_z are drag coefficients. The pitching motion of the quadcopter happens due to the difference in the thrust produced by the motors 2 and 4 and it is expressed as,

$$M_\phi = L(T_4 - T_2)$$

The rolling motion on the other hand occurs due to the difference between the thrust produced by the motors 1 and 3. It is expressed as,

$$M_\theta = L(T_3 - T_1)$$

L in the motion equations above is the distance between the center of the motor's rotor and the center of the quadcopter. The yawing motion for the quadcopter occurs due to the unbalanced drag forces acting on all the propellers, opposing their motion. It is given as,

$$M_\psi = B(-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2)$$

Where B is the torque constant. The combined rotational moment acting on the quadcopter is expressed as,

$$M_B = \begin{bmatrix} M_\phi \\ M_\theta \\ M_\psi \end{bmatrix}$$

The resistive torque on the quadcopter is due to the rotational drag and this is proportional to the parameter initially given for the given angular velocity. It is given as,

$$M_R = \begin{bmatrix} A_r P \\ A_r M \\ A_r R \end{bmatrix}$$

Using the Newton-Euler formulation of the dynamic model for this robot is done. The inertia matrix for the quadcopter is is diagonal and time-invariant due to the assumption that it has a symmetrical structure. The inertia matrix is given as,

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}$$

For convenience, the translational dynamics for the quadcopter are represented with respect to the inertial frame O and the rotational dynamics are represented with respect to the body frame B. The force equilibrium in the inertial frame can be expressed as,

$$m\dot{V}_B + v \times (mV_B) = R^T G + F_B - R^T F_D$$

Where $v \times (mV_B)$ is the centrifugal force, $R^T G$ is the total gravitational force, F_B is the total external thrust, $R^T F_D$ is the aerodynamic drag force and $m\dot{V}_B$ is the force causing the acceleration. In the inertial frame the centrifugal effects are negligible as only the translational motion is being referred to in this frame. Hence, the equilibrium equation can be stated as,

$$\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix} = \begin{bmatrix}0\\0\\-g\end{bmatrix} + R\frac{F_B}{m} - \frac{F_D}{m}$$

Substituting $\begin{bmatrix}\dot{x} & \dot{y} & \dot{z}\end{bmatrix}^T with \begin{bmatrix}w & y & z\end{bmatrix}^T$

$$\dot{u} = (S_\phi S_\psi + C_\phi S_\theta C_\psi)\frac{T_T}{m} - \frac{A_x}{m}u$$
$$\dot{v} = (-S_\phi C_\psi + C_\phi S_\theta S_\psi)\frac{T_T}{m} - \frac{A_y}{m}u$$
$$\dot{w} = -g + (C_\phi C_\theta)\frac{T_T}{m} - \frac{A_z}{m}w$$

With the close understanding of the rotational forces on the robot properly. The following equation describes the rotational equilibrium was established,

$$I\dot{v} + v \times (Iv) + \tau = M_B - M_D$$

In the component form the equation written above can be stated as,

$$\dot{P} = \left(\frac{I_{xx} - I_{yy}}{I_{zz}}\right)QR - \frac{I_R}{I_{xx}}Q\Omega + \frac{M_\phi}{I_{xx}} - \frac{A_r}{I_{xx}}P$$
$$\dot{Q} = \left(\frac{I_{zz} - I_{xx}}{I_{yy}}\right)PR - \frac{I_R}{I_{yy}}P\Omega + \frac{M_\theta}{I_{yy}} - \frac{A_r}{I_{yy}}Q$$
$$\dot{R} = \left(\frac{I_{xx} - I_{yy}}{I_{zz}}\right)PQ + \frac{M_\psi}{I_{zz}} - \frac{A_r}{I_{zz}}R$$

Where $\Omega = -\omega_1 + \omega_2 - \omega_3 + \omega_4$ and I_R is the rotational inertia of each of the motors. Using the equations from the moments acting at the quadcopter and the net thrust summation the following matrix equation can be established to find the individual rotor speeds.

$$\begin{bmatrix}T\\M_\phi\\M_\theta\\M_\psi\end{bmatrix} = \begin{bmatrix}K & K & K & K\\0 & KL & 0 & -KL\\-KL & 0 & KL & 0\\-B & B & -B & B\end{bmatrix}\begin{bmatrix}\omega_1^2\\\omega_2^2\\\omega_3^2\\\omega_4^2\end{bmatrix}$$

Inverting the matrix to make the rotor speeds the subject, gives the following motor speeds,

$$\omega_1^2 = \frac{T}{4k} - \frac{m_\theta}{2KL} - \frac{m_\psi}{4b}$$
$$\omega_2^2 = \frac{T}{4k} - \frac{m_\phi}{2KL} + \frac{m_\psi}{4b}$$
$$\omega_3^2 = \frac{T}{4k} + \frac{m_\theta}{2KL} - \frac{m_\psi}{4b}$$
$$\omega_4^2 = \frac{T}{4k} + \frac{m_\phi}{2KL} + \frac{m_\psi}{4b}$$

IV. Attitude Controller

There are 6 output parameters for the Quadcopter, x, y, z, θ, ϕ and ψ. However, this 6 desired outputs are controlled by only 4 inputs which is the angular speed ω of each motor. The most common scheme used is a dual stage controller, in which in first stage the attitude variables are controlled and in next stage position variables are controlled. Due to independent behaviour of the rotational variables, it is controlled first then followed by the translational variables.

To implement this controller, sensors are used which can measure vehicle's orientation, using which controller can generate a control signal which is then used by actuators.

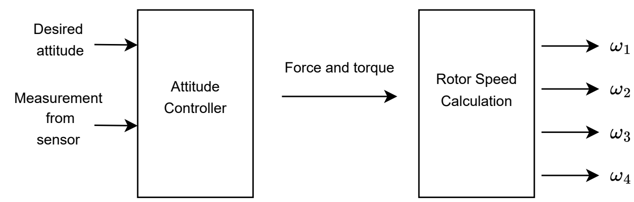

Figure 2: Block diagram of Attitude Controller

There are many types of control methods used for optimal performance and attitude stabilization few of them are discussed below:

A. PID controller

It is one of the easiest and most popular technique as it can be applied on non-linear model directly without linearizing it and can control each variable independently too, thus leading to better stabilization of Quadcopter. General expression for PID controller is:

$$u(t) = K_P e(t) + K_D\frac{d}{dt}e(t) + K_I\int e(\tau)d\tau$$

where the error $e(t)$ is the difference between desired state and the current state. K_P,K_I and K_D are the control parameters for the proportional, integral and derivative elements. For Quadcopter we have 3 different controllers to control the value of θ, ϕ and ψ respectively. The ϕ controller stabilizes the angle ϕ along the x-axis of the Quadcopter and helps in its translational movement too. Similarly θ and ϕ controller controls the angles θ and ψ along y and z-axis respectively. Using PID control scheme following equations for torque are generated.

$$m_\phi = I_{xx}(K_{P,\phi}e(t) + K_{D,\phi}\dot{e(t)} + K_{I,\phi}\int e(\tau)d\tau)$$
$$m_\theta = I_{yy}(K_{P,\theta}e(t) + K_{D,\theta}\dot{e(t)} + K_{I,\theta}\int e(\tau)d\tau)$$
$$m_\psi = I_{xx}(K_{P,\psi}e(t) + K_{D,\psi}\dot{e(t)} + K_{I,\psi}\int e(\tau)d\tau)$$

The Z-controller is used to control the altitude of the QuadCopter, PID scheme is also applied here however thrust is used in calculations here. Since all PID terms are in inertial frame while thrust is in body frame, therefore a rotational matrix is applied given by C_θ, C_ϕ. Final expression is given as:

$$T = mC_\theta C_\phi(g + K_{P,z}e(t) + K_{D,z}\dot{e(t)} + K_{I,z}\int e(\tau)d\tau)$$

V. Setup & Simulation

For the simulation of the Quadcopter model, the simulink project given in [2] was studied in detail for the implementation. The high-level System Architecture of the QuadCopter in [2] is shown in figure 3. It contains the following sub-blocks:

- Ground
- Package Delivery QuadCopter
- Trajectory Generation & Control
- Some other submodules for Physical modelling of the entire environment.

The physical modelling of the ground and the quadcopter is already done in this project, and hence it is out of the scope of our project. The Trajectory Generation & Control Module is of our primary interest, along with the Package Delivery QuadCopter sub-module.

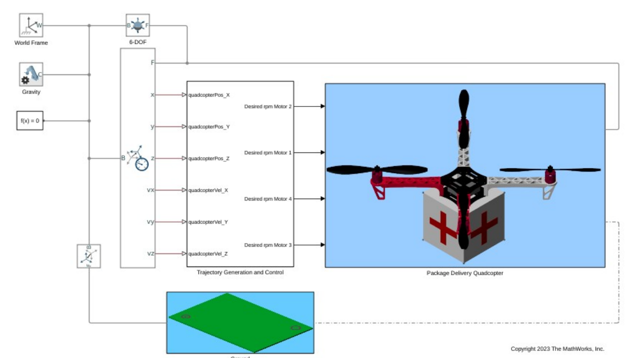

Figure 3: System Architecture [2]

A. Package Delivery Quadcopter

There are two main components of this sub-module, the Physical Modelling of the Quadcopter, and the package release functionality.

1) Physical Modelling of Quadcopter: In the Package Delivery Quadcopter sub-module, the physical modelling of the quadcopter is done. It is responsible for two jobs. Firstly, it models the physics of all the four rotors of the quadcopter. This is shown in figure 4. From this already established physical model of quadcopter in accordance with the equations discussed in the theoretical model, we get as an output, the rotor speeds of all four of these rotors, which will then be used to control and plan the drone trajectory for efficient package delivery.

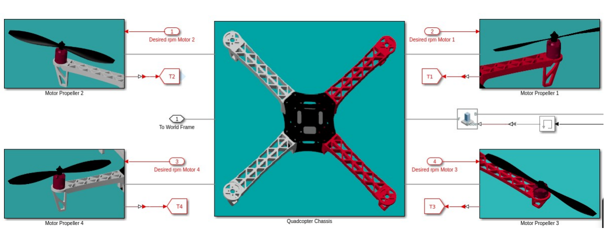

Figure 4: Physical Modelling of Quadcopter - Rotors

2) Package Release Functionality: The other part of the Package Delivery Quadcopter sub-module us to decide when to release the package. This part of the sub-module is shown in figure 5. In the existing code given in `package_release_trigger.m`, the logic was slightly incorrect which led to the package being dropped at the drop off location with a higher altitude. This can lead to the package being getting damaged, especially if its contents are fragile. The revised scheme for dropping the package but with a lower enough altitude such that the contents of the package are not damaged was implemented and the updated code can be found in the accompanied GitHub repository linked at the end of the paper.

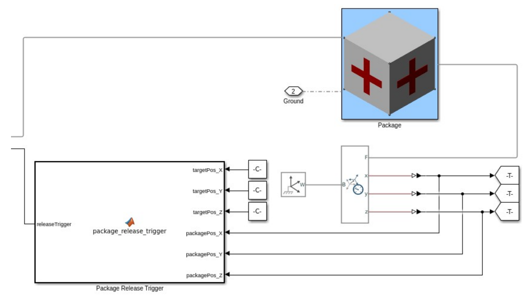

Figure 5: Physical Modelling of Quadcopter - Rotors

B. Trajectory Generation and Control

This module is the second main part of this project. All the theoretical calculations, control strategy for altitude, etc. that were discussed in the theoretical modelling section is implemented with in this block. As a blackbox, this block takes in as input, the quadcopter's current position P, and velocity V, i.e.:

$$P = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix}$$

$$V = \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix}$$

Taking in the above input, this block outputs the rotor speeds corresponding to each of the four rotors of the quadcopter, i.e.:

$$\omega = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix}$$

The internal sub-systems of this block is discussed next. There are four internal blocks in this system, which are:

- Trajectory Generator
- Position & Attitude Controller
- Altitude Controller
- Motor mixer

1) Trajectory Generator: This block is responsible for generating the trajectory of the drone from the start point to the destination point of the delivery. The block takes in as input, the desired path (i.e. waypoints) and their corresponding times (timespot), the initial position of the robot (homePose), the current position of the drone (pos), and the current time (t) as shown in figure 6. It outputs the deisred trajectory based on the provided inputs, and the velocity and yaw values that can help achieve the desired trajectory. Recall that, trajectory has the involvement of time and hence the velocity of the drone must also be determined and is crucial to achieve the desired trajectory.

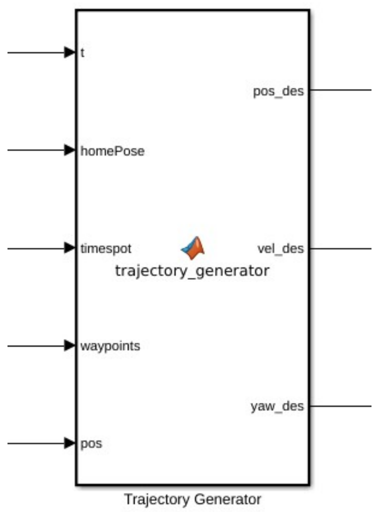

Figure 6: Trajectory Control Module

C. Position & Attitude Controller

As the name suggests, this block is control block that controls the speed, and the attitude of the quadcopter. It takes as an input, the desired position, velocity and yaw from the trajectory generation block, it also takes the current position and velocity as input and outputs the control commands for pitch, yaw and roll of the quadcopter. Within this block, lies two more sub-blocks, a position controller and an attitude controller.

1) Position Controller: This block takes in as input, all the parameters defined above. Additionally, it also takes in the current attitude (pitch, yaw and roll) of the quadcopter, the K_p and K_d constants, and outputs the desired pitch, yaw and roll of the quadcopter. It's block diagram is shown in figure 7. Note that, this block only outputs the desired attitude by controllin the position. It does not control the attitude.

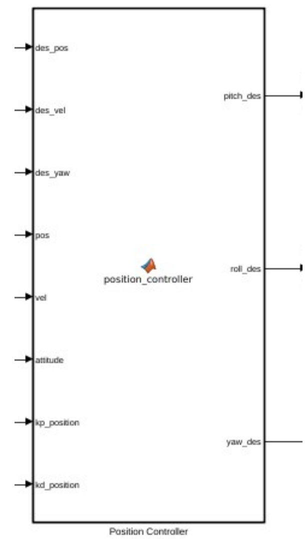

Figure 7: Position Controller

2) Attitude Controller: The block diagram for this sub-block is shown in figure 8. From the figure, it can be seen that there are separate PID controllers used for desired pitch, yaw and roll as given by the position controller and the control commands for pitch, yaw and roll are sent on the output.

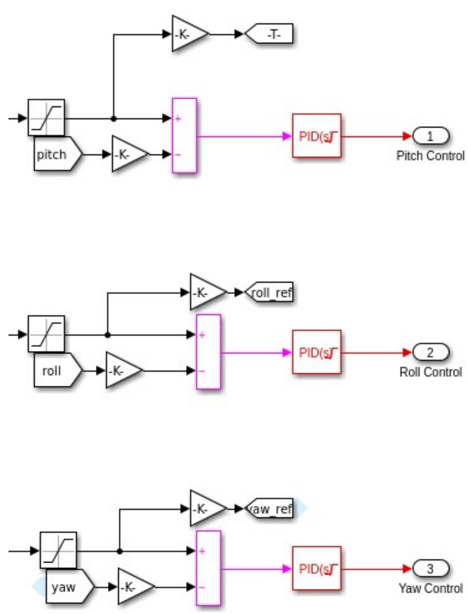

Figure 8: Attitude Control

D. Altitude Controller

The third of the four sub-modules in trajectory generation & control is the Altitude controller. As the name suggests, it takes as an input, the current altitude and the desired altitude of the quadcopter and generates a control command (throttle). Just like the control strategy for attitude, the control strategy used for controlling the altitude is PID control. It's block diagram can be seen in figure 9.

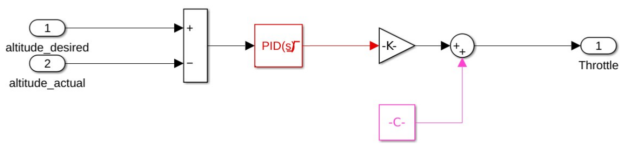

Figure 9: Altitude Control

E. Motor Mixer (Inverse Kinematics)

The last sub-module of the Trajectory Generation and control module is the motor mixer module. It acts as an inverse kinematics module and takes in as input, the pitch, roll, yaw and throttle control commands generated from their respective control blocks, and outputs the motor speeds of all the four rotors of the quadcopter. To do so, it uses the equations as discussed in the theoretical modelling section. Its block diagram is given in figure 10.

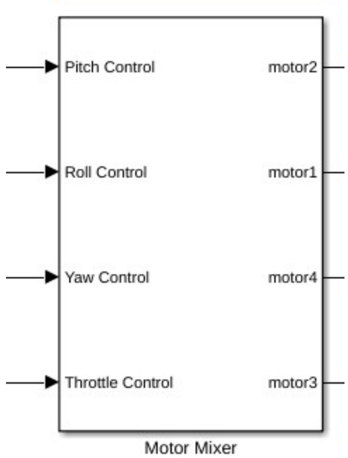

Figure 10: Motor Mixer (Inverse Kinematics)

F. Putting it all together

Finally all the four sub-blocks are connected together to obtain the Trajectory Generation and Control Module. Figure 11 shows the integrated four sub-blocks to create the top-level module.

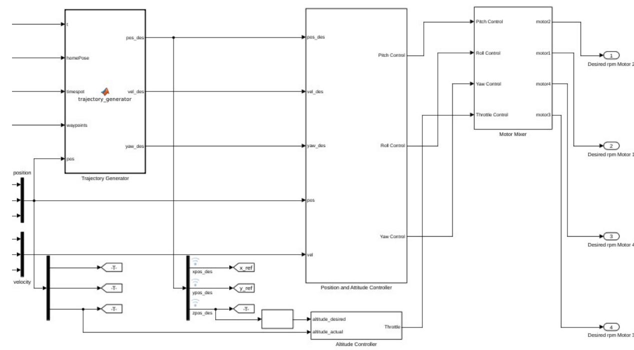

Figure 11: Trajectory Generation and Control Module

VI. Results

A. Testing Environment

The environment in which the drone is to be tested is shown in figure 12. The Drone has to deliver the package from the circle marked H to the other circle in the environment.

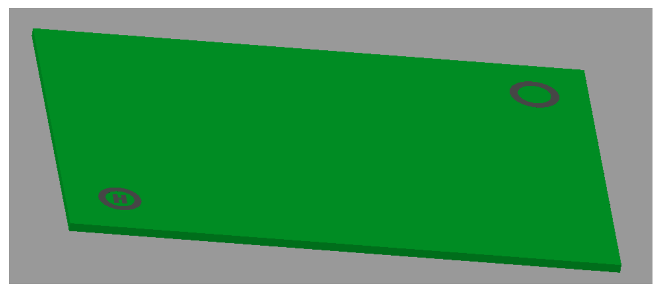

Figure 12: Quadcopter Environment

The quadcopter operating in this environment is shown in figure 13 carrying the package that it has to deliver, currently at the start position.

Figure 13: Simulated Quadcopter

Our Simulation was tested on a single path and the results were drawn from it. These results are discussed next. THe demo video can be found in the GitHub repository, in the media folder.

B. Actual & Desired Position

Figure 14 shows the comparison between the obtained and the desired positions along each axes separately. It can be seen from the result that our Position control block works fairly well enough to obtained the desired trajectory.

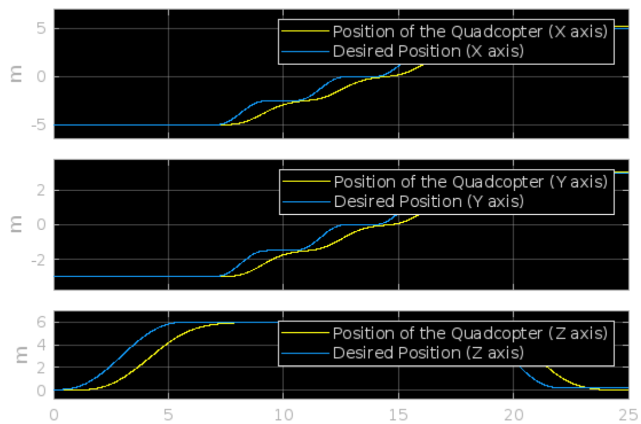

Figure 14: Plot for Obtained & Desired Position

C. Desired & Obtained Attitude

Figure 15 shows the comparison between the obtained and the desired attitude (pitch, yaw and roll). It can be seen from the result that our Attitude control block works fairly well enough to obtained the desired attitude for the drone to maintain its aerial balance.

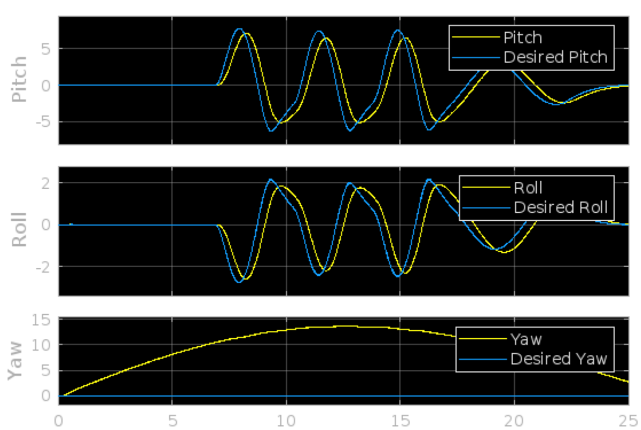

Figure 15: Plot for Obtained & Desired Attitude

D. Propeller Speeds & Thrusts

The figures 16 and 17 shows the propeller speeds and thrusts respectively. The thrust for all the four rotors seems to be equal. The speeds for a pair of the propellers seems equal.

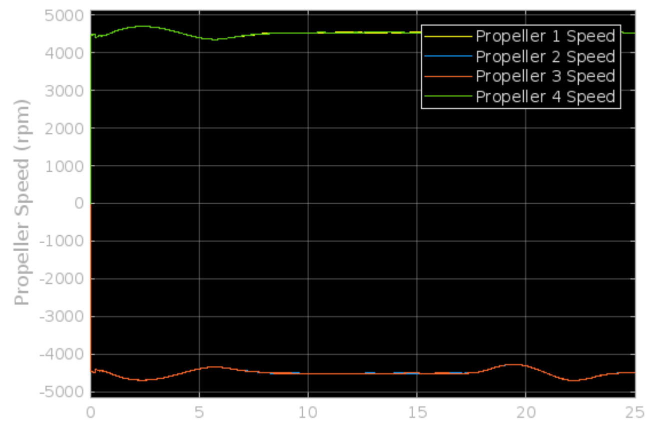

Figure 16: Propeller Speeds

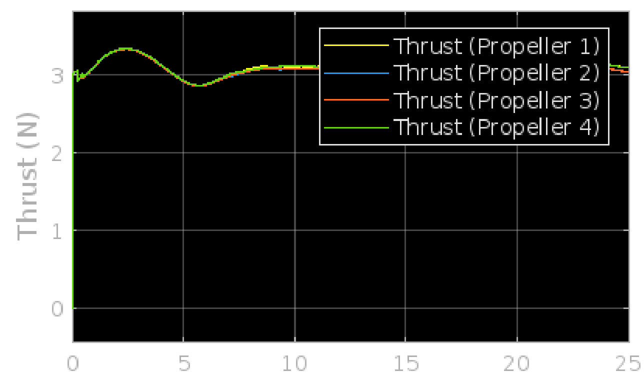

Figure 17: Propeller Thrusts

VII. Conclusion

In short, this paper first covers the theoretical aspect of the quadcopter, the equations related to its dynamics, kinematics and control of both the position (including altitude) and the attitude (pitch, yaw and roll). The paper then shifts its focus to actual test these equations on an already existing quadcopter project, for which it covers the entire details of the studied quadcopter model. Upon understanding the quadcopter simulation, a path is generated from a source to destination and some modifications are made in the already existing simulink file to improve the performance of the package delivery quadcopter. The simulation is performed and the results are extracted in the end, which upon analyzing seems to perform as desired with minimal altercations.

VIII. GitHub Repository

The GitHub Repository for all the files related to this project can be accessed by clicking here

IX. Acknowledgements

We would like to acknowledge the tribal chief, the head of the table, Dr. Basit Memon for his thorough help throughout this course and for accommodating us on a lot of occasions.

REFERENCES

[1] V. Praveen, S. Pillai *et al.*, "Modeling and simulation of quadcopter using pid controller," *International Journal of Control Theory and Applications*, vol. 9, no. 15, pp. 7151–7158, 2016.

[2] Mathworks. Package delivery quadcopter. [Online]. Available: https://www.mathworks.com/help/sm/ug/quadcopter.html

[3] H. Gao, C. Liu, D. Guo, and J. Liu, "Fuzzy adaptive pd control for quadrotor helicopter," in *2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*. IEEE, 2015, pp. 281–286.

[4] T. Video, "Ups tests residential drone delivery," Mar 2023. [Online]. Available: https://techcrunch.com/unified-video/ups-tests-residential-drone-delivery/

[5] A. Staff, "Amazon prime air prepares for drone deliveries," Jun 2022. [Online]. Available: https://www.aboutamazon.com/news/transportation/amazon-prime-air-prepares-for-drone-deliveries

[6] [Online]. Available: https://www.dhl.com/global-en/home/insights-and-innovation/thought-leadership/trend-reports/drones-logistics.html

[7] T. Luukkonen, "Modelling and control of quadcopter," *Independent research project in applied mathematics, Espoo*, vol. 22, no. 22, 2011.