Design and Analysis of Quadcopter

By

Min Thiha Soe

Munyaradzi Antony

23rd April 2024

Class Paper submitted for MAE 6245

School of Engineering & Applied Science

The George Washington University

THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

Contents

THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

Abstract

The main purpose of this paper is to investigate the kinematic and dynamic of a quadcopter with a type of mobile robot. With this kinematic and dynamic model, classical methods like PD controller and optimal techniques for a quadcopter will be investigated. The performance of each control scheme will be simulated in MATLAB.

Introduction

Drone designs, particularly multi-copters, started from the Da Vinci era. Throughout the years, many attempts were made to improve the control and dynamic of the multi-copter system, most notable attempts were made by contributors like Breguet brothers in 1907 and Oehmishen and Bothezat in 1920s. With each successive attempt, further understanding of the multi-copter dynamic and control were gained, which led to the commercialization of the drone technology in the 2010s (Mendoza-Mendoza, Gonzalez-Villela, Aguilar-Ibañez, & Fonseca-Ruiz, 2021).

Regulation and Standardizations

The standard to link between the theory and the sensor components was established by ISO 1151-2:1985. (Mendoza-Mendoza, Gonzalez-Villela, Aguilar-Ibañez, & Fonseca-Ruiz, 2021)

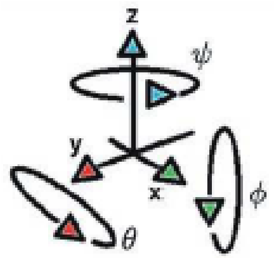


Figure 1 Drone motion and frame orientation established by ISO 1151-2:1985

The quadcopters were mainly developed photography, research in artificial vision or artificial intelligence algorithms, research in automatic control algorithms for aircraft flight, and agricultural purposes.

Quadcopter

Design

As the name suggests, the quadcopters have four motors which are oriented 90 degrees apart from each other. Brushless DC motors are commonly used in the quadcopter application due to their fast response time. The speed of the brushless motor can be controlled via RC-PWM as brushless motor is usually a multiphase motor. Aside from the motors, propellers and the base frame, the quadcopters can have various measuring components, such us IMU, GPS, gyroscopes, accelerometers, LIDARs and cameras to aid the control of the quadcopter. Quadcopters may also be equipped with telemetry and RC modules for communication and manual control and some type of microcontroller for autopilot functionality. Power components like batteries, power distributor, power module and battery indicator are also important components to consider during the designing phase of the quadcopter.

Kinematic and Dynamics modelling

In this section, a common configuration of the quadcopter was investigated. To simplify the modelling, the aerodynamics of the propellers are ignored, and the quadcopter operates in open space; far away from walls and objects. Figure 2 shows how the forces are acting on a quadcopter.

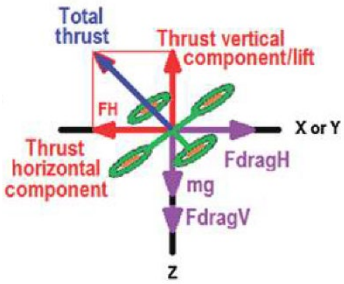


Figure 2 Forces acting on a quadcopter

Modelling Frames

There are three frames of modelling associated with the quadcopter, namely the world frame, the vehicle body frame, and the motor frame. World frame is the fixed frame where the vehicle moves with respect to this fixed reference frame. The body frame is attached to the quadcopter, usually placed at the center of gravity and mass. The translation and rotations of the quadcopter are measured from the body frame with respect to the world frame. Motor frames are placed on each of the motors to model the force and torque distribution of the quadcopter with respect to the vehicle reference frame.

Translational kinematics

The orientation of the vehicle with respect to the global frame can be related with rotation matrices. The rotation matrix associated with roll is $R_{z\psi}$ (where z is the roll axis) and it can be expressed as:

$$R_{z\psi} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The rotation matrix associated with pitch is $R_{x\phi}$ (where x is the pitch axis) and it can be expressed as:

$$R_{x\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}$$

The rotation matrix associated with yaw is $R_{y\theta}$ (where y is the yaw axis) and it can be expressed as:

$$R_{y\theta} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

Applying the Euler angle sequence of roll, pitch and yaw, the orientation of the drone in the vehicle frame can be expressed in terms of world frame as follows.

$$R_{Euler} = R_{z\psi} R_{x\phi} R_{y\theta}$$

$$R_{Euler} = \begin{bmatrix} c\psi c\theta - s\phi s\psi s\theta & -c\phi s\psi & c\psi s\theta + s\phi s\psi c\theta \\ s\psi c\theta + s\phi c\psi s\theta & c\phi c\psi & s\psi s\theta - s\phi c\psi c\theta \\ -c\phi s\theta & s\phi & c\phi c\theta \end{bmatrix}$$

The speed of the translation motion of the quadcopter in term of the work frame can be expressed as:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = R_{Euler} \begin{bmatrix} \dot{x}_b \\ \dot{y}_b \\ \dot{z}_b \end{bmatrix}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} c\psi c\theta - s\phi s\psi s\theta & -c\phi s\psi & c\psi s\theta + s\phi s\psi c\theta \\ s\psi c\theta + s\phi c\psi s\theta & c\phi c\psi & s\psi s\theta - s\phi c\psi c\theta \\ -c\phi s\theta & s\phi & c\phi c\theta \end{bmatrix} \begin{bmatrix} \dot{x}_b \\ \dot{y}_b \\ \dot{z}_b \end{bmatrix}$$

Rotational kinematics

The properties of skew-symmetric matrix associated with cross-product can be summarized as follows. If $a = [a_x \quad a_y \quad a_z]$ and $b = [b_x \quad b_y \quad b_z]$, cross product of a and b is

$$a \times b = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} = S(a)b$$

The angular velocity in the z-axis of the body frame is

$$\omega_z = \dot{\psi}$$

As skew matrix can be express as

$$S = \dot{R}R^T$$

$$S(\omega_z) = \begin{bmatrix} -\dot{\psi}\sin\psi & -\dot{\psi}\cos\psi & 0 \\ \dot{\psi}\cos\psi & -\dot{\psi}\sin\psi & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -\dot{\psi} & 0 \\ \dot{\psi} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Thus, all the angular velocities $(\omega_x, \omega_y, \omega_z)$ can be express in the skew symmetric matrix as:

$$S(\omega) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

$$\dot{R} = \frac{dR}{d\theta}\dot{\theta} + \frac{dR}{d\phi}\dot{\phi} + \frac{dR}{d\psi}\dot{\psi}$$

$$S(\omega) = R^T\dot{R}$$

$$S(\omega) = \begin{bmatrix} 0 & -\dot{\phi}\sin(\theta) - \dot{\psi}\cos(\phi)\cos(\theta) & \dot{\psi}\sin(\phi) + \dot{\theta} \\ \dot{\phi}\sin(\theta) + \dot{\psi}\cos(\phi)\cos(\theta) & 0 & -\dot{\phi}\cos(\theta) + \dot{\psi}\sin(\theta)\cos(\phi) \\ -\dot{\psi}\sin(\phi) - \dot{\theta} & \dot{\phi}\cos(\theta) - \dot{\psi}\sin(\theta)\cos(\phi) & 0 \end{bmatrix}$$

So,

$$\omega_x = \dot{\phi}\cos(\theta) - \dot{\psi}\sin(\theta)\cos(\phi)$$

$$\omega_y = \dot{\psi}\sin(\phi) + \dot{\theta}$$

$$\omega_z = \dot{\phi}\sin(\theta) + \dot{\psi}\cos(\phi)\cos(\theta)$$

$$\begin{bmatrix} {}^0\omega_x \\ {}^0\omega_y \\ {}^0\omega_z \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta)\cos(\phi) \\ 0 & 1 & \sin(\phi) \\ \sin(\theta) & 0 & \cos(\phi)\cos(\theta) \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

Dynamic Force Analysis of a drone

Force analysis

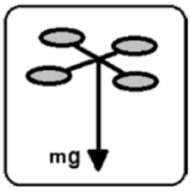


Figure 3 Gravity force acting on the quadcopter

Gravity is the most prominent force the drone needed to overcome for the flight to occur. For the simplicity of our model, the gravitational force is placed at the center of the quadcopter.

If a simple propeller case has two symmetric blades with the same aerodynamic profile, the forces of the propeller can be decomposed into vertical and horizontal components. The lift force of the propeller is the sum of all the lift forces of each blade. According to Figure 4, the vertical lift force of a propeller is:

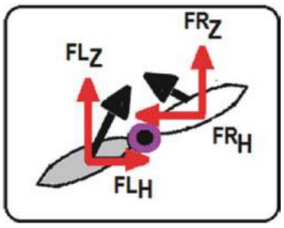


Figure 4 Vertical force component induce by blades of a propeller

$$\sum F_z = FL_z + FR_z = 2F_z$$

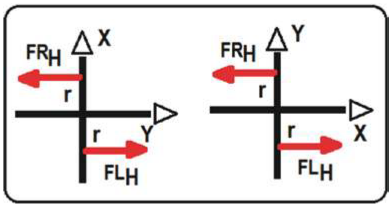


Figure 5 Horizontal forces acting on the y and x direction due to blades of the propeller

As the blades are modelled to be symmetric, the vertical forces induced by the blades can be assumed to cancel each other out resulting in net zero force in the horizontal direction.

$$\sum F_x = \sum F_y = FL_H + FR_H = 0$$

Torque analysis

The vertical torque caused by the blades can be expressed as follows.

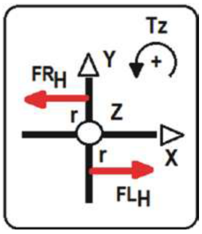


Figure 6 Vertical torque analysis

$$\sum \tau_z = r(FR_H + FL_H) = 2rF_H$$

Where F is the horizontal force generated by the blade and r is the distance from the propeller's center to the analysis point on each blade.

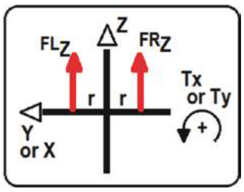


Figure 7 Horizontal torque analysis

On the other hand, the horizontal torque caused by the blades cancel each other due to the symmetric modelling assumption.

$$\sum \tau_x = \sum \tau_y = 0$$

To summarize, the well-balanced propellers can only generate lift force in the vertical direction and torque around the axis of rotation (z-axis).

Forces and Torques on a quadcopter

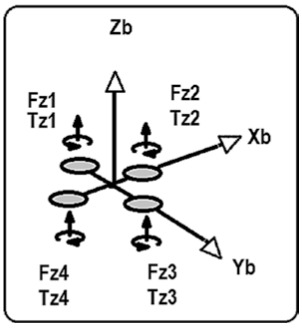


Figure 8 Forces and torques generated by each propeller of the quadcopter

Figure 8 shows the quadcopter's rotation setup, where two motors in the x axis rotate opposite to the two motors in the y axis. This setup prevents autorotation. The forces in the z-direction can be derived as follows:

$$\sum F_Z = F_{Z1} + F_{Z2} + F_{Z3} + F_{Z4}$$

If the quadcopter is assumed to be a well-balanced vehicle, the force in y and x axes will have zero net resultant force.

$$\sum F_x = \sum F_y = 0$$

This assumption may be erroneous for a large quadcopter system, but it was made to simplify the problem. As illustrated in Figure 8, the torque of the quadcopter around the z-axis can be expressed as:

$$\sum T_z = T_{z1} - T_{z2} + T_{z3} - T_{z4}$$

The torque of the quadcopter around the y-axis can be expressed as:

$$\sum T_y = rF_{Z2} - rF_{Z4}$$

Where r is the distance from the center of the quadcopter to the assumed location of the force in the vertical direction. Similarly, the torque of the quadcopter around the x-axis can be expressed as:

$$\sum T_x = rF_{Z1} - rF_{Z3}$$

In this paper, a simple drone with one thrust vector, three torques generated by the motors and the gravity acting at the center of the drone was investigated.

Dynamic modelling

Translational dynamics

A point mass assumption is made, and the geometric center of the drone coincides with the center of gravity. Newton's second law equation can be applied to the simple quadcopter model as follows:

$$\vec{F} = m\vec{a} = m\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}$$

As acceleration is just the second derivative of position, the equation becomes:

$$\vec{F} = m\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix}$$

Since there is gravity and net thrust force by the propellers in the direction of z, the equation can be expressed as combination of vertical force with respect to world frame and the gravity component:

$$\vec{F} = R\begin{bmatrix} 0 \\ 0 \\ F_z \end{bmatrix} + m\begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}$$

$$m\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} c\psi c\theta - s\phi s\psi s\theta & -c\phi s\psi & c\psi s\theta + s\phi s\psi c\theta \\ s\psi c\theta + s\phi c\psi s\theta & c\phi c\psi & s\psi s\theta - s\phi c\psi c\theta \\ -c\phi s\theta & s\phi & c\phi c\theta \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ F_z \end{bmatrix} + m\begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}$$

$$m\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} F_z(c\psi s\theta + s\phi s\psi c\theta) \\ F_z(s\psi s\theta - s\phi c\psi c\theta) \\ F_z(c\phi c\theta) \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ F_z \end{bmatrix} + m\begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}$$

Rotational dynamics

The general torque vector acting on the quadcopter due to angular acceleration can be expressed as:

$$\vec{\tau} = J\dot{\omega}_i$$

where J is the diagonal inertia matrix. However, in rotational dynamics of the quadcopter, centrifugal torque plays a significant role, and this effect can be integrated to the general expression of torque as:

$$\vec{\tau} = J\dot{\omega}_i + (\omega_i \times J\omega_i)$$

$$\vec{\tau} = \begin{bmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{bmatrix}\begin{bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix} + \left(\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \times \begin{bmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{bmatrix}\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}\right)$$

$$\begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} J_x\dot{\omega}_x \\ J_y\dot{\omega}_y \\ J_z\dot{\omega}_z \end{bmatrix} + \begin{bmatrix} \omega_y\omega_z J_z - \omega_y\omega_z J_y \\ \omega_x\omega_z J_x - \omega_x\omega_z J_z \\ \omega_x\omega_y J_y - \omega_x\omega_y J_x \end{bmatrix}$$

Next, the allocation matrix for frame configuration shown in figure 9 which relates to each individual propeller can be found as:

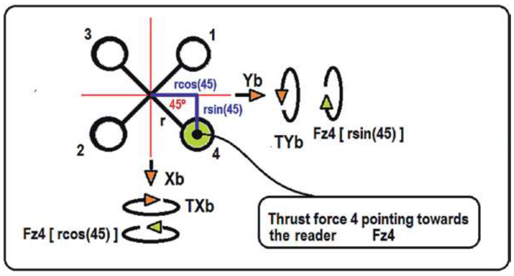


Figure 9 Torques generated at each propeller.

$$F_z = F_{Z1} + F_{Z2} + F_{Z3} + F_{Z4}$$

$$F_x = F_y = 0$$

To combat auto rotation:

$$\tau_z = \tau_{z1} + \tau_{z2} - \tau_{z3} - \tau_{z4}$$

If the configuration of the drone is shown as Figure 9, the torque generated at x and y axis can be expressed as:

$$\tau_x = r\cos(45)\,(-F_{Z1} + F_{Z2} + F_{Z3} - F_{Z4})$$

$$\tau_y = r\sin(45)\,(F_{Z1} - F_{Z2} + F_{Z3} - F_{Z4})$$

As forces and torques are generated by the speed of the motor, it can be express in terms of a constant times the speed of the motor and simplified allocating matrix can be expressed as:

$$\begin{bmatrix} F_x \\ F_y \\ F_z \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \end{bmatrix}\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix}$$

$$\begin{bmatrix} F_z \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \end{bmatrix}\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix}$$

The lift force and torque generated by the quadcopter can be controlled by controlling the speed of the motor according to the following simplified matrix:

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \end{bmatrix}^{-1}\begin{bmatrix} F_z \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix}$$

If the constant $\frac{1}{4}$ is taken out of the inverse allocation matrix, it can be simplified to:

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \\ 1 & -1 & -1 & -1 \end{bmatrix}\begin{bmatrix} F_z \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix}$$

$$\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\\ \omega_4\end{bmatrix} = \begin{bmatrix}F_z - \tau_x + \tau_y + \tau_z\\ F_z + \tau_x - \tau_y + \tau_z\\ F_z + \tau_x + \tau_y - \tau_z\\ F_z - \tau_x - \tau_y - \tau_z\end{bmatrix}$$

Linearizing the non-linear system

In a smooth flight roll ($\emptyset$) and pitch angles (θ) tend to be zero, while the yaw angle (ψ) can be set to desire value. For small yaw angles, $\sin(\psi) \approx \psi$ and roll and pitch angles, $\cos(\emptyset)\ or \cos(\theta)\ = 1$ and $\sin(\emptyset)\ or \sin(\theta)\ = 0.$ With these assumptions, the translational dynamic equation can be simplified as follows:

$$F = m\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix} = \begin{bmatrix}F_z(c\psi s\theta + s\phi s\psi c\theta)\\ F_z(s\psi s\theta - s\phi c\psi c\theta)\\ F_z(c\phi c\theta)\end{bmatrix} + m\begin{bmatrix}0\\ 0\\ -g\end{bmatrix}$$

$$\begin{bmatrix}F_x\\ F_y\\ F_z\end{bmatrix} = m\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix} = F_z\begin{bmatrix}\theta c\psi + \phi s\psi\\ \theta s\psi - \phi c\psi\\ 1\end{bmatrix} + m\begin{bmatrix}0\\ 0\\ -g\end{bmatrix}$$

$$\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix} = \frac{F_z}{m}\begin{bmatrix}\theta c\psi + \phi s\psi\\ \theta s\psi - \phi c\psi\\ 1\end{bmatrix} + \begin{bmatrix}0\\ 0\\ -g\end{bmatrix}$$

The rotational kinematic equation can be simplified to:

$$\begin{bmatrix}\omega_x\\ \omega_y\\ \omega_z\end{bmatrix} = \begin{bmatrix}\cos(\theta) & 0 & -\sin(\theta)\cos(\phi)\\ 0 & 1 & \sin(\phi)\\ \sin(\theta) & 0 & \cos(\phi)\cos(\theta)\end{bmatrix}\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix}$$

$$\begin{bmatrix}\omega_x\\ \omega_y\\ \omega_z\end{bmatrix} = \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} = \begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix}$$

$$\begin{bmatrix}\dot{\omega}_x\\ \dot{\omega}_y\\ \dot{\omega}_z\end{bmatrix} = \begin{bmatrix}\ddot{\phi}\\ \ddot{\theta}\\ \ddot{\psi}\end{bmatrix}$$

If the smooth flight, low angular velocity assumption was made, the centripetal affects are negligible, and the dynamic model can be linearized as follows:

$$\begin{bmatrix}\tau_x\\ \tau_y\\ \tau_z\end{bmatrix} = \begin{bmatrix}J_x\dot{\omega}_x\\ J_y\dot{\omega}_y\\ J_z\dot{\omega}_z\end{bmatrix} + \begin{bmatrix}\omega_y\omega_z J_z - \omega_y\omega_z J_y\\ \omega_x\omega_z J_x - \omega_x\omega_z J_z\\ \omega_x\omega_y J_y - \omega_x\omega_y J_x\end{bmatrix}$$

$$\begin{bmatrix}\tau_x\\ \tau_y\\ \tau_z\end{bmatrix} \approx \begin{bmatrix}J_x\dot{\omega}_x\\ J_y\dot{\omega}_y\\ J_z\dot{\omega}_z\end{bmatrix} \approx \begin{bmatrix}J_x\ddot{\phi}\\ J_y\ddot{\theta}\\ J_z\ddot{\psi}\end{bmatrix}$$

Controls

In this section, the control for the quadcopter will be implemented using the previously linearized and simplified kinematic and dynamic equations of the quadcopter. Zero by approximation method can drive the states of the system to zero and it can be achieved by implementing a decay function as follows:

$$y = Ae^{-Bt}$$

where B needs to be a positive value. By utilizing this type of decay function to the system, the required speed to drive the system to zero position can be express as follows:

$$\frac{dy}{dt} = -y$$

$$B\dot{y} = -Ky$$

Where B and k are positive gains. This equation is essentially known as PD controller and where B is the derivative gain and K is proportional gain.

For further analysis, the PD controller equation can be written as follow:

$$B\dot{y} + Ky = K_d\dot{E} + K_pE$$

The error value (E) is the difference between the desired value v_d and the actual value (v)

$$E = v_d - v$$

This implies that as the aim is for the actual value to be as close as possible to the desired value which implies that E becomes zero, the difference in desired value and actual value become zero and the actual value become desire value.

Another important system is the mass spring damper system or damped harmonic oscillator, effective when constants M, B, and K are positive.

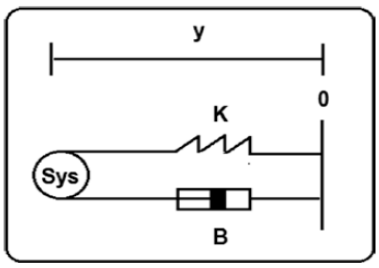


Figure 10 The mass spring damper.

$$M\ddot{y} + B\dot{y} + ky = 0$$

This can be implemented in the control theory as follows:

$$\frac{d(K_d\dot{E} + K_pE + K_i\int E)}{dt} = K_d\ddot{E} + K_p\dot{E} + K_iE$$

Linearized Control with No Yaw Variation

In this section, a basic control mode where the drone only follows a path without altering its orientation was investigated. Using the simplified and linearized systems developed, independent and dependent dynamics can be identified as shown in Figure 11.

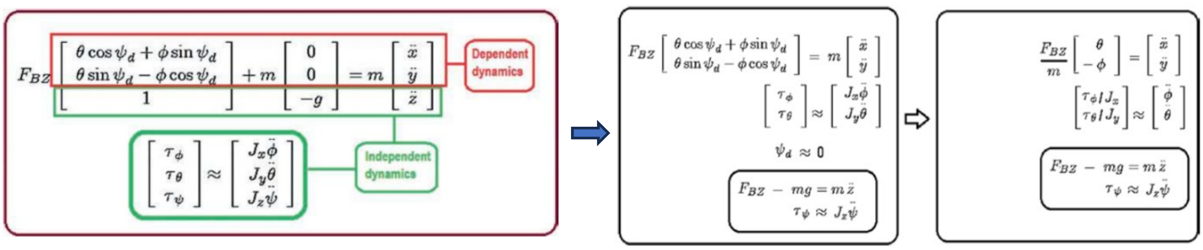


Figure 11 Dependent and Independent components from dynamic modelling.

Dynamics Control for Independent state

If the thrust in the z axis is controlled with PD control scheme, in other word adding mPD_Z term to the independent dynamic equation:

$$F_z = m(g + PD_Z + \ddot{z}_d) \tag{1}$$

Where the PD controller in the z-direction is implemented as:

$$PD_z = K_{PZ}(z_d - z) + K_{Dz}(\dot{z}_d - \dot{z})$$

Replacing this equation in the dynamic model results in:

$$m\ddot{z} + mg = m(g + PD_z + \ddot{z}_d)$$

$$m\left(PD_z + (\ddot{z}_d - \ddot{z})\right) = 0$$

$$K_{PZ}(z_d - z) + K_{Dz}(\dot{z}_d - \dot{z}) + (\ddot{z}_d - \ddot{z}) = 0$$

In other words, the above function would drive the system to the desired position, velocity and acceleration. The error in the z direction can be expressed as:

$$E_z = z_d - z$$

Therefore, the dynamic equation can be written in terms of the derivative of errors as shown below:

$$K_{PZ}(E_z) + K_{Dz}(\dot{E}_z) + (\ddot{E}) = 0$$

If error in z direction goes to zero, the desire position with desire velocity and desire acceleration will be achieved. From control theory, if the desired value is a constant or if it is a function with maximum absolute value, when the controller successfully drives the quadcopter to desire location, the speed and acceleration associated in given axis would go to zero, with the exception in z axis which need to counterbalance the gravitational force. In other words, when the quadcopter reaches the desire location, the force in the z-axis should equal to the gravitational force and the equation 1 becomes:

$$F_z = mg$$

Where m is the mass of the quadcopter. If the differential and proportional gains are big enough the following assumptions can be made; Small acceleration can be neglected in a smooth flight and if a mass is constant or bounded by a constant it can be absorbed by the gains thus the thrust in the z-axis can be written simply as

$$F_Z = mg + PD_z$$

For the previously simplified and linearized rotational dynamic, the torque in z axis is

$$\tau_z = J_z\ddot{\psi}$$

The PD controller can be implemented to control the yaw angle (ψ) to desire angle ψ_d with desire velocity and acceleration as follow:

$$\tau_z = J_z\ddot{\psi}_d + J_z PD_\psi$$

$$\tau_z = J_z(\ddot{\psi}_d + PD_\psi) \tag{2}$$

$$J_z\ddot{\psi} = J_z(\ddot{\psi}_d + PD_\psi)$$

$$0 = J_z(\ddot{\psi}_d - \ddot{\psi} + PD_\psi)$$

The PD_ψ controller in the ψ direction is implemented as

$$PD_\psi = K_{P\psi}(\psi_d - \psi) + K_{D\psi}(\dot{\psi}_d - \dot{\psi})$$

So, the rotational dynamic in the ψ direction with the PD controller can be expressed in term of error (E_ψ) as:

$$0 = J_z(\ddot{\psi}_d - \ddot{\psi} + K_{P\psi}(\psi_d - \psi) + K_{D\psi}(\dot{\psi}_d - \dot{\psi}))$$

$$K_{P\psi}(\psi_d - \psi) + k_{D\psi}(\dot{\psi}_d - \dot{\psi}) + (\ddot{\psi}_d - \ddot{\psi}) = 0$$

$$K_{P\psi}(E_\psi) + k_{D\psi}(\dot{E}_\psi) + (\ddot{E}_\psi) = 0$$

Assuming the smooth flight assumption, the desire angular acceleration is very small and approximated as zero. Therefore, the torque in the z axis can be approximated as follows, starting from equation (2):

$$\tau_z = J_z(\ddot{\psi}_d + PD_\psi)$$

$$\tau_z \approx PD_\psi$$

$$\tau_z \approx K_{P\psi}(\psi_d - \psi) + K_{D\psi}(\dot{\psi}_d - \dot{\psi})$$

Dependent Dynamics Control

Similar to the implementation of torque in the z axis, the torques in the x and y (pitch and roll) can be simplified as follows, given that the flight is assumed to be smooth the desire angular acceleration for pitch and roll is close to zero.

$$\tau_x = J_x(PD_\phi + \ddot{\phi}_d) \approx PD_\emptyset$$

$$\tau_y = J_y(PD_\theta + \ddot{\theta}_d) \approx PD_\theta$$

The linearized acceleration in x and y direction was previously derived as:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \frac{F_z}{m}\begin{bmatrix} \theta c\psi + \phi s\psi \\ \theta s\psi - \phi c\psi \end{bmatrix}$$

Assuming the yaw angle (ψ) reaches the desired yaw angle, ψ becomes zero and if the acceleration in the z axis reaches the desired height, F_z is the lift component to counter act the gravity (mg). So, the above equation can be further linearized as:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = g\begin{bmatrix} \theta \\ -\phi \end{bmatrix} \approx g\begin{bmatrix} \theta_d \\ -\phi_d \end{bmatrix}$$

The above simplified equation shows the relationship between the acceleration in the x and y axis and the roll (θ) and pitch (ϕ) angles. By changing the desired roll and pitch angles, the quadcopter can be manipulated to move in the x and y plane. PD controller can be implemented to control the translational dynamic of the quadcopter as follows:

$$\ddot{x}_d + PD_x = g\,\theta_d$$

$$\theta_d = \frac{1}{g}(PD_x + \ddot{x}_d)$$

For the desired pitch angle ϕ_d, it can be expressed as:

$$\phi_d = -\frac{1}{g}(PD_y + \ddot{y}_d)$$

Using the smooth flight assumption and the gains of the PD controller absorb the constant values like gravity, $\frac{1}{g}$ for this case, the desired roll angle θ_d and the desired pitch angle can be. d

$$\theta_d = PD_x$$

$$\phi_d = -PD_y$$

In summary, implementing the PD controller to the dynamic equations of the quadcopter can be simplified as:

$$F_Z = mg + PD_z$$

$$\theta_d = PD_x$$

$$\phi_d = -PD_y$$

$$\tau_x = PD_\emptyset$$

$$\tau_y = PD_\theta$$

$$\tau_z = PD_\psi$$

So, the expended form of quadcopter dynamic equations with PD controller are:

$$\theta_d = K_{Px}(x_d - x) + K_{Dx}(\dot{x}_d - \dot{x})$$

$$\phi_d = -[K_{Py}(y_d - y) + K_{Dy}(\dot{y}_d - \dot{y})]$$

$$F_Z = mg + K_{PZ}(z_d - z) + K_{DZ}(\dot{z}_d - \dot{z})$$

$$\tau_x = K_{P\emptyset}(\emptyset_d - \emptyset) + K_{D\emptyset}(\dot{\emptyset}_d - \dot{\emptyset})$$

$$\tau_y = K_{P\theta}(\theta_d - \theta) + K_{D\theta}(\dot{\theta}_d - \dot{\theta})$$

$$\tau_z = J_z[K_{P\psi}(\psi_d - \psi) + K_{D\psi}(\dot{\psi}_d - \dot{\psi})]$$

With the above PD controller implemented dynamic equations, a simple quadcopter simulation, which can move the drone from one point in space to another point in space without changing the orientation of the drone could be achieved.

Quadcopter control setup

The inner loop and outer loop control method is also investigated for the quadcopter. For the quadcopter control, the outer loop controls the translation motion of the quadcopter and computes the roll and pitch angle. The inner loop controls the altitude and attitude of the quadcopter which controls the dynamic of the quadcopter by adjusting the speed of the four motors. This control scheme allows the quadcopter to relax the small angle flight assumption and improve the overall response speed and robustness of the system.

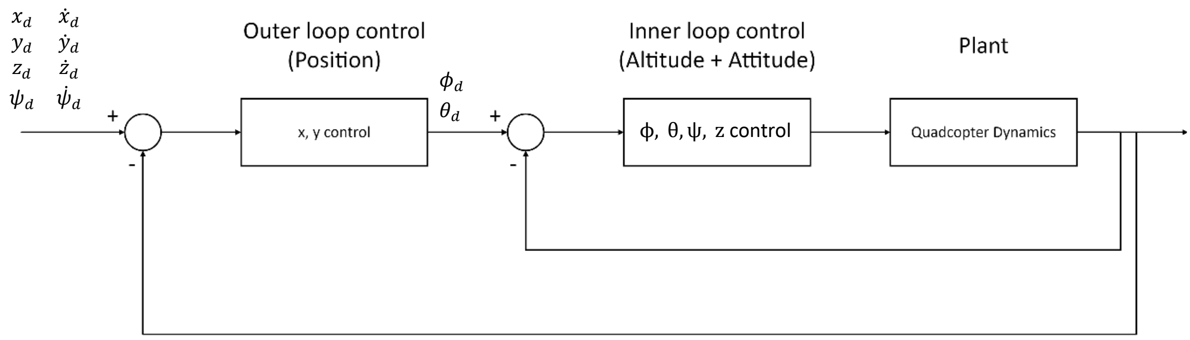


Figure 12 Typical inner and outer loop implementation of the quadcopter

Figure 12 shows the inner and outer loop setup which will be implemented in the first part of the Simulations section in this paper.

Linearized Control with Yaw Variation

The controller can be modified to allow variation in the yaw angle as follows. The torque control is similar to the previously derived equations:

$$F_Z = mg + PD_z$$

$$\tau_x = PD_\emptyset$$

$$\tau_y = PD_\theta$$

$$\tau_z = PD_\psi$$

If the yaw angle is assumed to be non-zero, the dynamic system equation is:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \frac{F_z}{m} \begin{bmatrix} \theta c\psi_d + \phi s\psi_d \\ \theta s\psi_d - \phi c\psi_d \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}$$

If F_z counter-balance gravitational force (mg):

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = g \begin{bmatrix} \theta c\psi_d + \phi s\psi_d \\ \theta s\psi_d - \phi c\psi_d \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}$$

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = g \begin{bmatrix} c\psi_d & s\psi_d \\ s\psi_d & -c\psi_d \end{bmatrix} \begin{bmatrix} \theta \\ \phi \end{bmatrix}$$

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = g \begin{bmatrix} c\psi_d & s\psi_d \\ s\psi_d & -c\psi_d \end{bmatrix} \begin{bmatrix} \theta_d \\ \phi_d \end{bmatrix}$$

Implementing the outer loop control to compute desired roll and pitch angle can be implemented as follow:

$$\begin{bmatrix} \theta_d \\ \phi_d \end{bmatrix} = \frac{1}{g} \begin{bmatrix} c\psi_d & s\psi_d \\ s\psi_d & -c\psi_d \end{bmatrix}^{-1} \begin{bmatrix} PD_x + \ddot{x}_d \\ PD_y + \ddot{y}_d \end{bmatrix}$$

$$\begin{bmatrix} \theta_d \\ \phi_d \end{bmatrix} \approx \begin{bmatrix} c\psi_d & s\psi_d \\ s\psi_d & -c\psi_d \end{bmatrix} \begin{bmatrix} PD_x \\ PD_y \end{bmatrix}$$

$$\begin{bmatrix}\theta_d \\ \phi_d\end{bmatrix} \approx \begin{bmatrix}PD_x c\psi_d + PD_y s\psi_d \\ PD_x s\psi_d - PD_y c\psi_d\end{bmatrix}$$

With this the desired yaw angle (ψ_d) can be incorporated to compute the desired roll and pitch angles. In Simulations, two setups, one where no yaw variation and another with yaw variation were investigated.

PD control gains

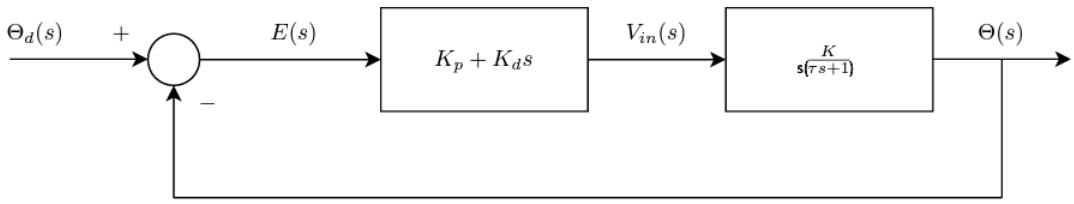


Figure 13 Typical PD controller setup

The PD controller can be described by following transfer function as:

$$\mathrm{K}(s) = k_p + k_d s$$

If our control system is modelled as

$$\frac{\Theta(s)}{\Theta_d(s)} = \frac{K_d K s + K_p K}{\tau s^2 + (K_d K + 1)s + K_p K}$$

As error of position in Figure 13 can be expressed as:

$$E(s) = \Theta_d(s) - \Theta(s)$$

Also, the final value theorem state that steady state error exists if $\lim_{s\to 0} sF(s) \neq 0$. So, if the limit of the close loop transfer function of the PD controller approaches zero:

$$\lim_{s\to 0}(sF(s)) = \lim_{s\to 0}\left(\frac{K_d K s^2 + K_p K s}{\tau s^2 + (K_d K + 1)s + K_p K}\right)$$

$$= \lim_{s\to 0}\left(\frac{0}{K_p K}\right)$$

Therefore, PD controller is usually implemented in the quadcopter control because it can mostly eliminate steady-state error. The procedure for calculating the PD controller gains can be derived as follows. For the simulation, the settling time (T_s) and the damping ratio (ζ) are arbitrarily chosen as 5 seconds and 0.8 respectively. The relationship between the settling time, the damping ratio and the natural frequency can be related as:

$$T_s = \frac{4}{\zeta \omega_n}$$

Substituting in the desired constants, the natural frequency associated with modelled system be computed as:

$$\omega_n = \frac{1}{\zeta\tau} = \frac{4}{\zeta T_s} = \frac{4}{0.8 \times 5} = 1\,rad/s$$

The second order system can be express as:

$$s^2 + 2\zeta\omega_n s + {\omega_n}^2$$

$$s^2 + \frac{(KK_d + 1)}{\tau}s + \frac{KK_p}{\tau}$$

Matching the equivalent terms, the proportional gain can be calculated as:

$${\omega_n}^2 = \frac{KK_p}{\tau}$$

$$K_p = \frac{{\omega_n}^2 \tau}{K}$$

$$K_p = \frac{1^2 \times 1.25}{K}$$

Assuming the quadcopter motor constant k_v is around 2000, the proportional gain can be calculated as:

$$K_p = \frac{1^2 \times 1.25}{2000} = 0.000625$$

For the differential gain K_d:

$$K_d = \frac{2\zeta\omega_n\tau - 1}{K}$$

$$K_d = \frac{(2 \times 0.8 \times 1 \times 1.25) - 1}{2000} = 0.0005$$

With these initial gains as starting point, the gains of the controller were fine tuned to achieve an over-damped system.

State estimators.

State Space system modelling of the quadcopter was also investigated to implement state estimator. The most general state-space representation of a linear system can be expressed as shown:

$$\dot{x}(t) = Ax(t) + Bu(t)$$

$$y(t) = Cx(t) + Du(t)$$

$x(t)$ is state of the system, $y(t)$ is output and $u(t)$ is the input to control the quadcopter (Tahir, Tahir, & Liaqat). According to the Tahir et al., the 6-dof rotational dynamic of the quadcopter can be express as following matrix:

$$\dot{x} = Ax + Bu$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -g & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & g & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ \phi \\ \theta \\ \psi \\ \dot{\phi} \\ \dot{\theta} \\ \theta \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{1}{J_x} & 0 & 0 \\ 0 & 0 & \frac{1}{J_y} & 0 \\ 0 & 0 & 0 & \frac{1}{J_z} \end{bmatrix} \begin{bmatrix} F_z \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix}$$

$$y = Cx + Du$$

$$\begin{bmatrix}\phi\\ \theta\\ \psi\end{bmatrix}=\begin{bmatrix}1&0&0&0&0&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0&0&0&0&0\\0&0&1&0&0&0&0&0&0&0&0&0\end{bmatrix}\begin{bmatrix}\phi\\ \theta\\ \psi\\ \dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix}+\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}\begin{bmatrix}F_z\\ \tau_\phi\\ \tau_\theta\\ \tau_\psi\end{bmatrix}$$

The state estimator can be implemented by predicting the state of the step as follows:

$$\tilde{X}_{k+1\,|k}=A\,\tilde{X}_{k|k}+Bu_k$$

$$\underbrace{m}_{M}\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix}=\underbrace{\begin{bmatrix}F_z(c\psi s\theta+s\phi s\psi c\theta)\\ F_z(s\psi s\theta-s\phi c\psi c\theta)\\ F_z(c\phi c\theta)\end{bmatrix}\begin{bmatrix}0\\0\\F_z\end{bmatrix}+m\begin{bmatrix}0\\0\\-g\end{bmatrix}}_{b}$$

Next, Optimal control using LQR (Ahmed, Basil, & Raafat, 2019) method was investigated, which has the following quadratic cost function.

$$J=\int(X^TQX+u^TRu)dt$$

The Q matrix is a positive diagonal matrix with the weight specified for each state and increasing the values of Q matrix penalize the state error of the corresponding state. The R matrix is the weight matrix related to the control effort of the controller.

By using Algebraic Riccati Equation:

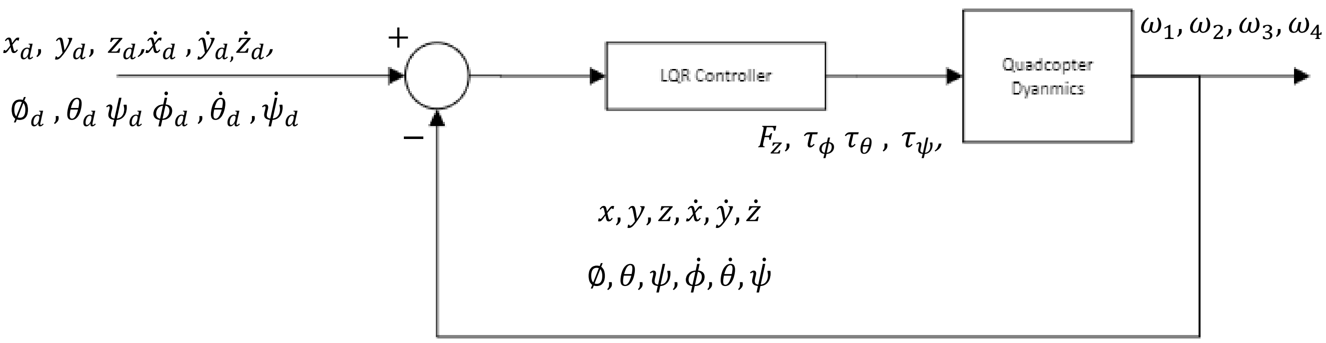
$$A^TXA-X-A-A^TXB(B^TXB+R)^{-1}B^TXA+Q=0$$

And the gain (G) can be calculated as:

$$G=(B^TXB+R)^{-1}B^TXA$$

The control law can then be implemented simply as:

$$u(t)=G(desired\ state\ -\ current\ state)+u_{past}$$



Simulations

In this section, the quadcopter simulation with and without yaw variation using a PD controller were investigated. The gains in the outer loop are significantly smaller compare to the gains in the inner loop, as smooth flight assumption and small change in pitch and roll angles. The Simulink simulation was set up according to the Figure 12, with an additional component which check the location of the drone and set a new waypoint if the desired waypoint was achieved for set time. The 2 points and the origin of the axes were arbitrary chosen as waypoints and the drone was set to move between those three points.

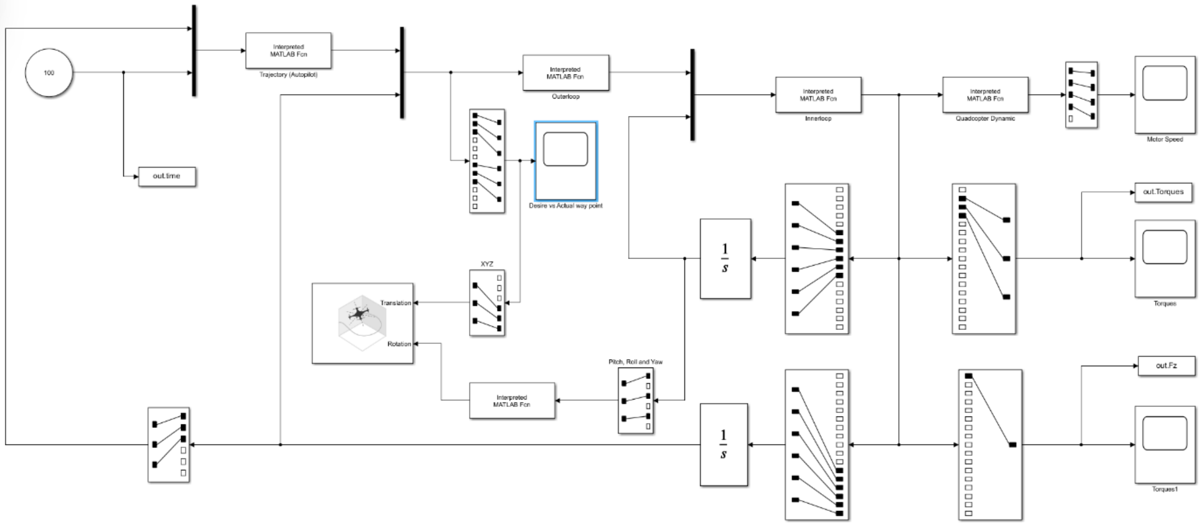


Figure 14 Quadcopter simulation setup in Simulink

Initially, all the proportional and differential gains were set to 0.000625 and 0.0005 respectively, with the exception of the yaw angle gain ($K_{p\psi}$ and $K_{d\psi}$). It was found that the controller was not able to control the quadcopter and drive the system to desired position.

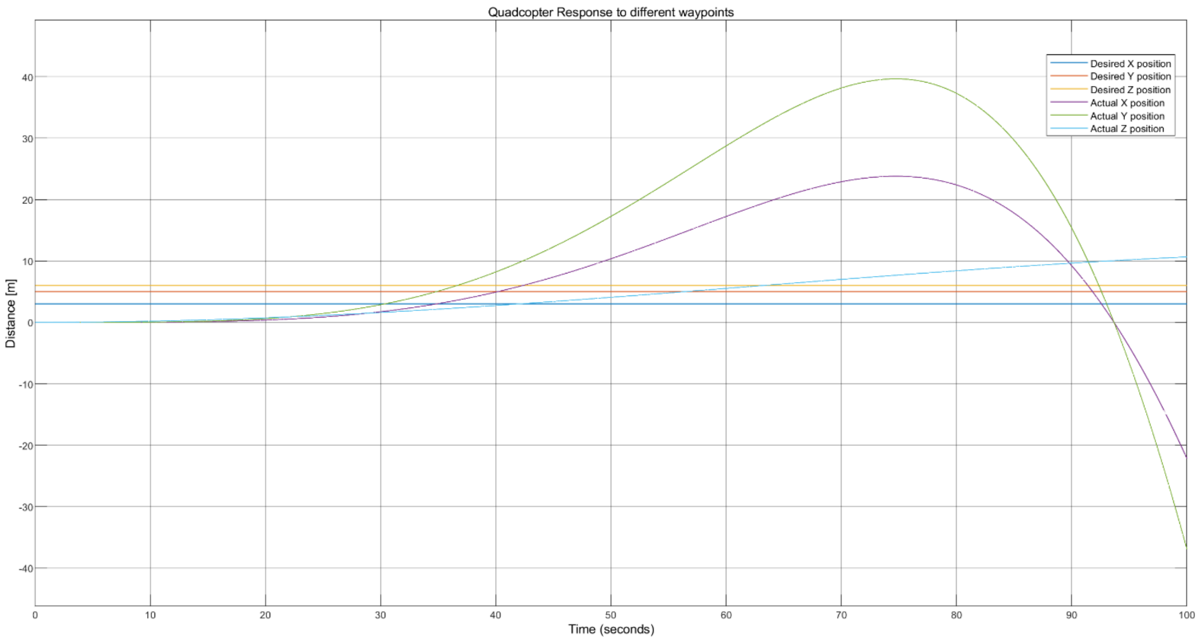


Figure 15 The quadcopter response when kp and kd are 0.000625 and 0.0005 respectively.

Therefore, the gains used by (Mendoza-Mendoza, Gonzalez-Villela, Aguilar-Ibañez, & Fonseca-Ruiz, 2021) were investigated. The gains for [K_{px}, K_{dx}, K_{py}, K_{dy}, K_{pz}, and K_{dz}] were set to [0.1 0.09 0.1 0.09 1 0.7] and the gains for angles [$K_{p\emptyset}$, $K_{d\emptyset}$, $K_{p\theta}$, $K_{d\theta}$, $K_{p\psi}$, and $K_{d\psi}$] were set to [3 1 3 1 3 0]. In order to have no rotation in yaw, the $K_{d\psi}$ was set to zero.

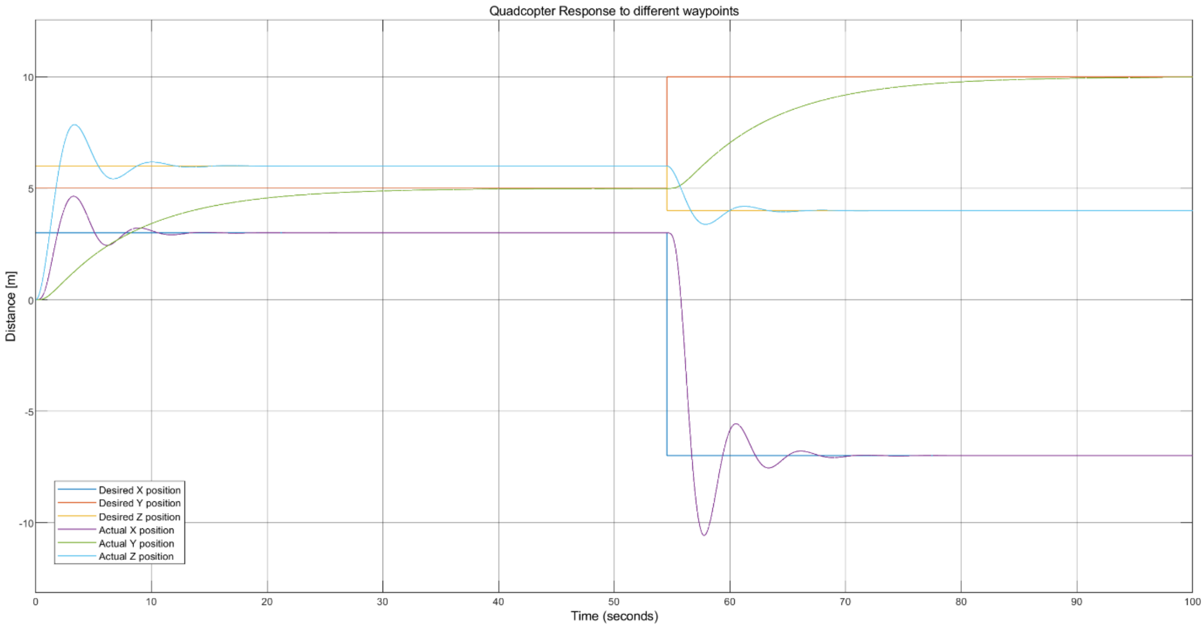


Figure 16 Response of the drone with the reference gains from (Mendoza-Mendoza, Gonzalez-Villela, Aguilar-Ibañez, & Fonseca-Ruiz, 2021)

Figure 16 shows that the drone's movement in the x and z axis are underdamped while, the movement in y axis is over-damped. Also, this control movement in the z axis might not be desirable as when the drone return back to origin, the drone might crash to the ground before arriving the desired x and y location. For the drone to become a fully damped system and land safely at the origin, the following control gains for [K_{px}, K_{dx}, K_{py}, K_{dy}, K_{pz}, and K_{dz}] were set to [0.3 0.09 0.3 0.09 3 3.5] and the gains for angles [$K_{p\emptyset}$, $K_{d\emptyset}$, $K_{p\theta}$, $K_{d\theta}$, $K_{p\psi}$, and $K_{d\psi}$] were set to [5 2 5 2 5 0].

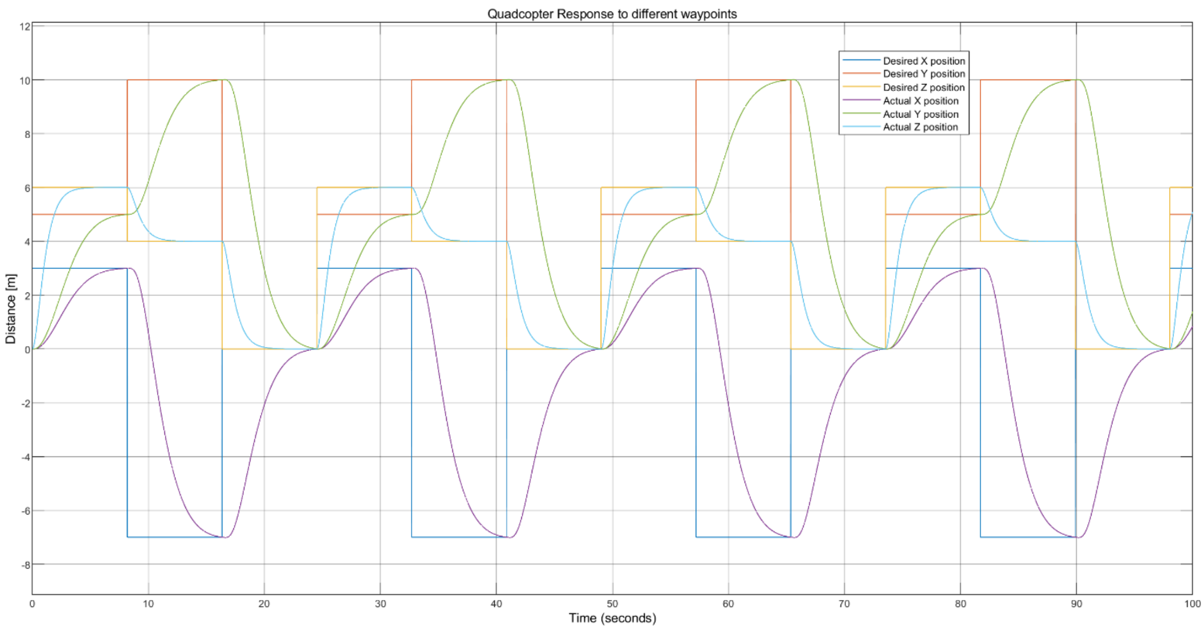


Figure 17 The motor speed response in the quadcopter

Figure 17 shows the fully damped quadcopter response. As the PD controller can fully damp the quadcopter position response and the decrease in the transient response, from more than 100 seconds to less than 10 seconds, no further adjustments were made.

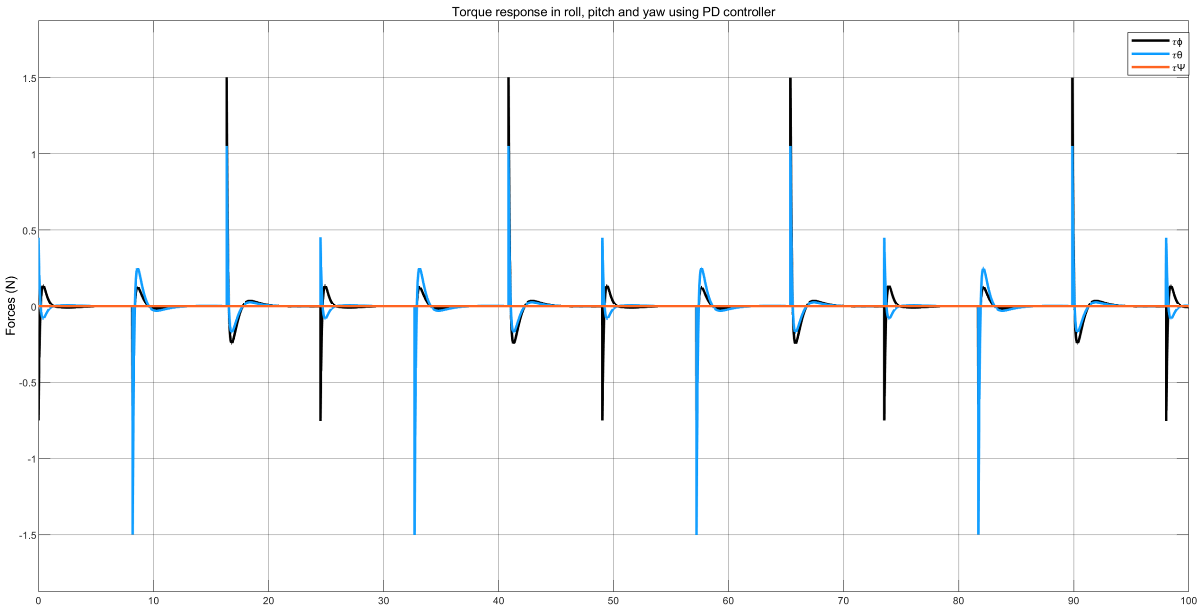


Figure 19 Torque response of the quadcopter

The simulation can be explained as follows, the gains K_{px}, K_{dx}, K_{py}, K_{dy} are controlling the desired pitch and roll angles need to dive the quadcopter in x and y plane while K_{pz}, K_{dz}, $K_{p\theta}$, $K_{d\theta}$, $K_{p\phi}$, $K_{d\phi}$, $K_{p\psi}$, and $K_{d\psi}$ are used to compute the forces and torques need to drive the quadcopter to desired position in x, y and z axis. The proportional gain K_p is affecting the system response to the error (difference between the desired and actual position). Using several tests, it can be shown that a higher K_p will make the system respond more quickly but as time it gives rise to overshoots. The derivative gain K_d affects how the system responds to the rate of change of error. It is the one that causes damping of the system to avoid over-shoot and improve stability.

From the force in z direction and torques computed from the PD controller, the velocity need for the quadcopter is computed by Quadcopter dynamic module as shown in Figure 14. Figure 19 shows the overall velocity response of the all the motors in the quadcopter, and sharp change in speed was observed when the drone switch different waypoint.

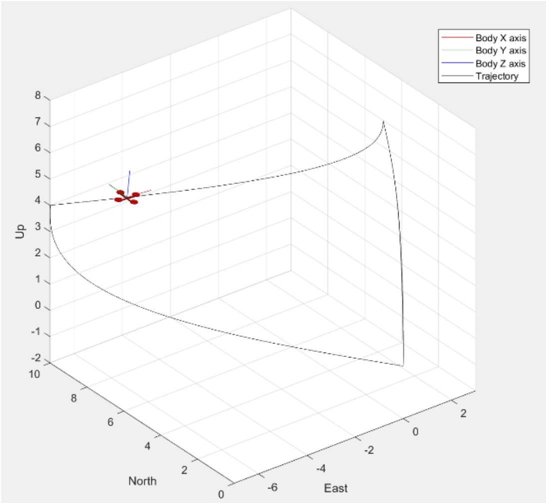


Figure 20 Quadcopter waypoint tracking simulation

To simulate the drone, the current pitch, roll and yaw angles were converted to quaternion vector using the following equations (Cariño, Abaunza, & Castillo, 2022).

$$q = \begin{bmatrix} c\left(\frac{\emptyset}{2}\right)c\left(\frac{\theta}{2}\right)c\left(\frac{\psi}{2}\right) + s\left(\frac{\emptyset}{2}\right)s\left(\frac{\theta}{2}\right)s\left(\frac{\psi}{2}\right) \\ s\left(\frac{\emptyset}{2}\right)c\left(\frac{\theta}{2}\right)c\left(\frac{\psi}{2}\right) - c\left(\frac{\emptyset}{2}\right)s\left(\frac{\theta}{2}\right)s\left(\frac{\psi}{2}\right) \\ c\left(\frac{\emptyset}{2}\right)s\left(\frac{\theta}{2}\right)c\left(\frac{\psi}{2}\right) + s\left(\frac{\emptyset}{2}\right)c\left(\frac{\theta}{2}\right)s\left(\frac{\psi}{2}\right) \\ c\left(\frac{\emptyset}{2}\right)c\left(\frac{\theta}{2}\right)s\left(\frac{\psi}{2}\right) - s\left(\frac{\emptyset}{2}\right)s\left(\frac{\theta}{2}\right)c\left(\frac{\psi}{2}\right) \end{bmatrix}$$

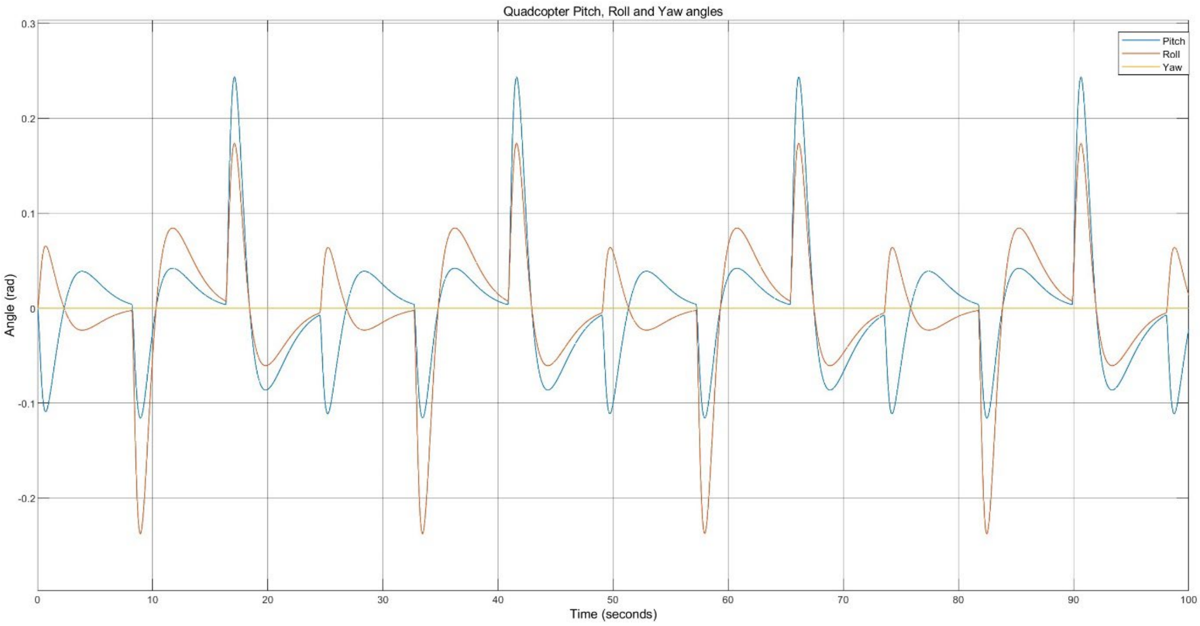


Figure 21 Pitch Roll and Yaw response of the quadcopter

In the simulations, it was found that without setting the differential gain in the yaw (PD_{ψ}) to zero was essential to prevent auto rotation. The optimal controller using LQR method for the quadcopter was also explored. The weighting matrix Q for states sensor was chosen as a diagonal matrix with [1 1 1 10 10 10 1 1 1 10 10 10] and the control effort matrix R was equally weighted. The function named LQR, shows how optimal method could be implemented. Although, the LQR method simulation was not included in this project, it is one of the optimal methods to control the quadcopter without tuning the PD gains.

Future work and considerations

The simulation work was completed; however, there are a lot of things that can be improved to make the model work in real life applications. The following may be considered as opportunities for improvement.

Introduction of autonomy and navigation is essential for the use of the drone in many cutting-edge technologies. This would involve adding cameras on the drone which will do Simultaneous Localization and Mapping (SLAM). Techniques like the Kalman filter can be employed for this algorithm to locate the position of the camera (drone) relative to the ground and the SLAM builds a map as the sensor distinguishes identifiable features from the environment.

Control Systems of the simulation can be advanced to a more robust one using methods like Sliding Mode Control (SMC). This will assist the drone to adjust to a changing environmental condition unlike the traditional control with fixed parameters. It will deal with non-linearities and ensure finite time convergence.

Having a flying drone is good but the best thing is for it to have specific design for specific applications like agriculture monitoring, search and rescue, environmental monitoring.

Conclusion

Comprehensive findings on kinematics, dynamics and control calculations for the quadcopter have been presented. Through the simulation we have validated modeling approach of system linearization which is realized in the stability and performance of the quadrotor. The Proportional and Derivative Controller PD has been utilized to control the quadrotor's attitude around its hover position. Simulink was used to generate graphs to show the behavior of the drone and an animation was also generated to represent the actual motion of the drone in 3D space.

The practical implications of this document are manifold, extending across many fields which are delivery, surveillance, and entertainment. For instance, enhanced control systems can enable drones to perform more reliably in delivering goods in urban areas, while advanced stabilization features allow for more precise surveillance capabilities, which can be particularly useful in law enforcement and media coverage.

The presented information will need further research to make much contribution to the field so that the drones can possess intelligence for autonomous flight systems, potentially leading to fully autonomous drones capable of complex decision-making in real-time.

References

Ahmed, A. A., Basil, H. J., & Raafat, S. M. (2019). Nonlinear Estimation of Quadcopter States Using Unscented Kalman Filter. *Periodicals of Engineering and Natural Sciences*, .1626-1637.

Cariño, J., Abaunza, H., & Castillo, P. (2022). A fully-actuated quadcopter representation using quaternions. *International Journal of Control*, 3132–3154.

Mendoza-Mendoza, J. A., Gonzalez-Villela, V. J., Aguilar-Ibañez, C. F., & Fonseca-Ruiz, L. (2021). *Drones to Go.* New York: Springer Science.

Tahir, Z., Tahir, W., & Liaqat, S. A. (n.d.). State Space System Modelling of a Quad Copter UAV.