

DEVELOPMENT OF TRAJECTORY TRACKING OF QUADROTOR UAV

NUR AININA BINTI YAZID

UNIVERSITI TEKNOLOGI MALAYSIA

TABLE OF CONTENTS:

Table of Contents

1.1 The Unmanned Aerial Vehicles (UAV).....	3
1.2 Path Planning And Documents Layout	8
2.0 Literature Review.....	9
2.1 Mathematical Modelling.....	9
2.1.1 Euler-Newton equations.....	14
2.1.2 The Flight Operation.....	16
2.2 Linear Controller.....	19
2.2.1 PID Controller.....	19
2.2.2 LQR Controller.....	21
2.3 Comparison between PID Controller and LQR Controller.....	22
2.4 Trajectory Development.....	23
3.0 Methodology.....	25
3.1 Choice of Trajectory	25
3.2 Simulink.....	29
3.2.1 Dynamic Model Simulation.....	29
3.3 Expected Results.....	31
3.4 Reference:	33
4.0 Appendix.....	34

1. Introduction

1.1 The Unmanned Aerial Vehicles (UAV)

Unmanned Aerial Vehicle (UAV) is an invention of transportation with no pilot on board. Unlike other aircrafts, the aviator usually found remotely controlling the vehicle at a ground station placed nearby. It also can be pre-programmed to fly autonomously according to its flight paths via complex dynamic system.

The early attempt of UAV development was done by Louis Braguet who designed a four rotor helicopter. This was the primary rotational wing air ship to lift itself off the ground, albeit just a few feet of elevation. In 1908, it was accounted for as having flown 'a few times' despite the fact of interest are unclear. This trial was followed by Etienne Oehmichen which explored different avenues regarding rotorcraft outlines in the 1920s. Among the six plans he attempted, his helicopter No.2 had four rotors and eight propellers, all determined by a solitary motor. The Oehmichen No.2 utilized a steel-tube outline, with two-bladed rotors at the end of the four arms. The point of these sharp edges could be shifted by twisting. Five of the propellers, turning in the flat plane, settled the machine laterally. Another propeller was mounted at the nose of steering. The remaining pair of propellers worked as its forward drive. The air ship showed an impressive level of solidness and expansion in control-precision, and made over a thousand experimental about the year of 1920s. By 1923 it could stay airborne for a few minutes on end, and on April 14, 1924 it built up the first-ever FAI separation record for helicopters of 360 m (390 yd). It exhibited the capacity to finish a circular motion and later, it finished the initial 1 kilometer (0.62 mi) closed-circuit flight by a rotorcraft (Wikipedia,2016)

However, for the past few decades, many modifications and advanced experiments to test the abilities of quadcopters have been done. Small-scale UAV, like Walkera Scout X4 showed promising manoeuvrability and ability to perform autonomous missions that are currently not possible with other vehicles. There are a few other drones that also becoming the focus of research like Parrot AR.Drone which is controlled by small radio with cameras attached to it and AeroQuad which is an open-source hardware and software project based on Arduino.

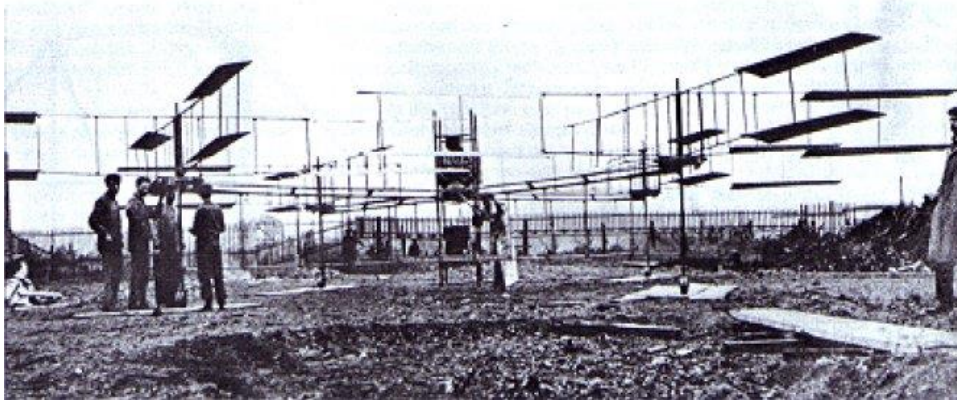


Figure 1.1: Breguet-Richet Gyroplane (1907)

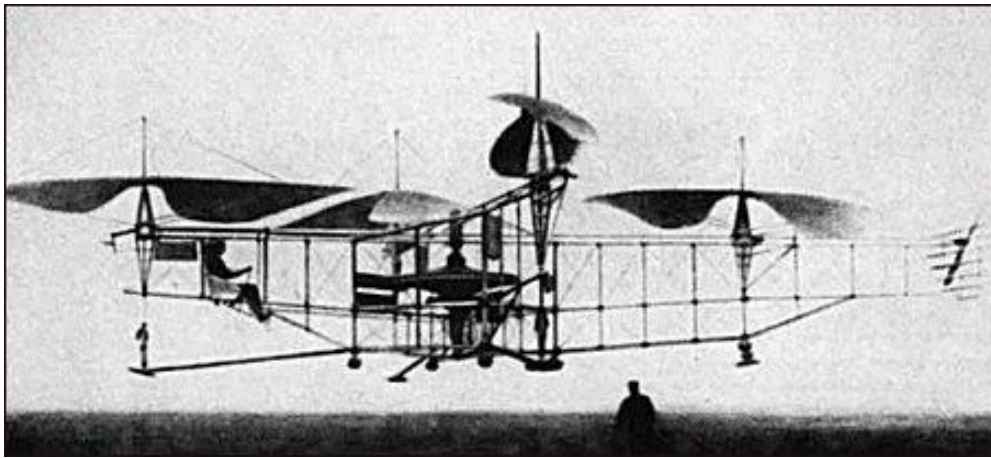


Figure 1.2: Oehmichen No. 2 (1920)



Figure 1.3: Parrot A.R Drone



Figure 1.4: AeroQuad



Figure 1.5: Walkera Scout X4

As mentioned earlier, the development of small scale UAV has been exaggerated. One of the most infamous UAV that has been used widely is the quadrotor.

Quadrotor is a helicopter with four rotors located at equal distance from the centre of the body. (Teppo Luukkonen,2011). This vehicle, just like shown in Figure 1.3 to 1.5, used the mechanism of angular velocities to control the movements; thrust by electric motors. Now, it is widely used all over the world due to its variety of sizes and capabilities depending on the mission obligated on the rotorcraft. For instance, this technology has been opted out a lot by the law enforcement and military for surveillance and reconnaissance, search and rescue and monitoring suspicious targets in big events.

Compared to a fixed wing UAV, a quadrotor has a lot of advantages that make it the most preferred aerial vehicle. These include the vertical take-off and landing ability (VTOL), hovering ability, low level of noise and its easy-to-control flight mechanism.

As the application of quadcopter has been accepted by the society, researchers start to take a serious interest towards this complex dynamic system; from the very basic avionic components to the trajectory tracking controllers for the quadrotor. Due to that phenomena, more complex aerodynamic properties have been introduced (G.M.Hoffman,2007). This include the mathematical modelling and the linear controller options to acquire the accurate information on the position and the attitude measurement performed by the gyroscope planted in the aerial vehicle (P.Martin, 2010).

In this paper, the focus of the study is centred on the trajectory tracking for a quadrotor UAV. As to that matter, a list of objectives must be performed in order to obtain the desired results. The aim of this project is to develop an algorithm to enable “coverage control” in which the UAV will be able to scan an area at an optimal time and conditions. So as to achieve that, a dynamic control model of quadrotor will be developed and a trajectory pathway will be designed to fulfil the mission. This so-called dynamic controlled will be focusing on a chosen linearized control system that will be justified in the paper and this controller also will be developed and implemented on MATLAB’s Simulink and then programmed to track any given trajectory (Fan Jiankun, 2014).

1.2 Path Planning And Documents Layout

This thesis can be divided into two parts which are:

- 1) Obtaining the dynamic equation and mathematical modelling of the subject which includes the derivation of the equation, design a modelling system based on the equation and simulate the proposed aftermath using MATLAB Simulink.
- 2) Designing the trajectory tracking controller path for the quadrotor. This matter at hand can be solved by covering an area and performed a coverage control over the entire area.

In order to cover the whole plan, this paper will be separated into three chapters. The first chapter will give an introduction to the Unmanned Aerial Vehicle (UAV) and the subject at hand which is the quadrotor. It also will cover the problem statement and objectives of the study.

The second chapter will discuss about the literature review on trajectory tracking of a quadcopter and the linear controller choices to perform the task will be listed and explained. This chapter will also cover the derivation of the equations which is expanded from both Newton-Euler equations and Euler-Langrange equations. Chapter 2 also will cover about the UAV trajectory development over the years and the expected results of the study will also be written.

Chapter 3 will be focusing on the methodology on how the objectives of the study will be achieved. In this section, the linear controller to run the system will be chosen and justified. The coding and the simulation of the dynamic equation will also be attached and presented by MATLAB. The mathematical model also will be tested by simulating the quadcopter equations with suitable control inputs. And the last section will be ended by a conclusion.

2.0 Literature Review.

There are a lot of surveys and research papers on the optimal control of a flight. However, UAVs are harder to govern since it belongs in non-linearized system. Despite of the development in mathematical equation, physicists are still working hard to create an easy way to control the quadcopter. Hence, the existence of decoupling equations of translation and rotational inputs to linearize the dynamic equation. Not only that, UAVs are known to be able to perform autonomous flight with pre-programmed memory. This include following a trajectory path to track any targets or to hover a coverage area. Usually, quadcopters are embedded with transmission system to track the trajectory and the Electronic Speed Control (ESC) which act as the brain of the vehicle will be programmed to follow the path. Many quadrotors controllers operate near hover on small angle assumptions for roll and pitch (Mellinger,Warren,2012). However, some are great enough to actually play around these parameters and perform aggressive manoeuvring. This chapter will discuss about the dynamic equation used to execute such task autonomously. Hence, the derivation of equation of motions and the mathematical modelling of the program.

2.1 Mathematical Modelling

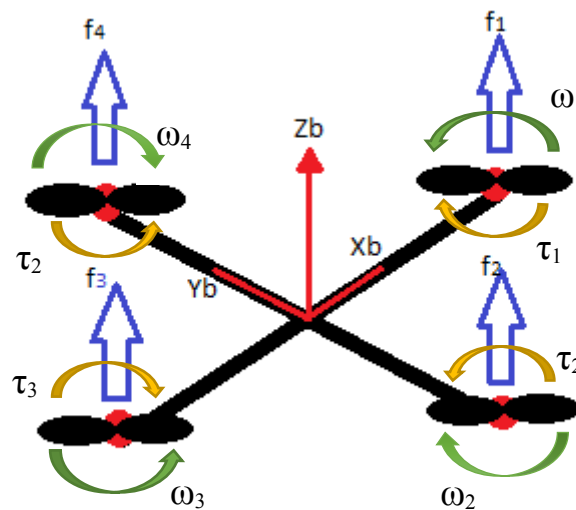


Figure 2.1: Body axes system

Figure 2.1 shows the schematic body axes system used widely in quadrotor modelling. Notice that the axes of y and x represents the arms of the robot while z pointing towards gravity (vertical direction). The centre of the body is the centre of gravity.

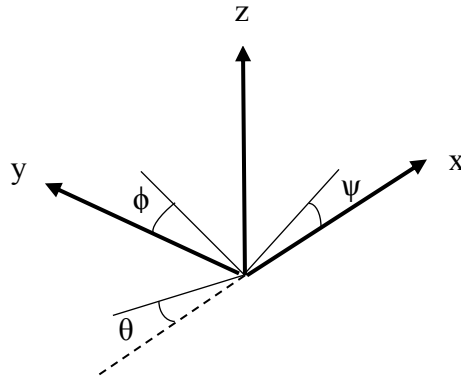


Figure 2.2: The inertial frames of a quadcopter.

Note that the inertial frame of the quadcopter shows the Euler angles of yaw, pitch and roll angles. The θ represents pitch angle which determines the rotation motion, ϕ is the roll angle determine the rotation around x-axis while ψ , which is the yaw angle determines the yaw angle in z-axis.

Quadcopter usually defined by six degrees of freedom which are the translational position of x, y, z . As mentioned earlier in the paper, the dynamic equation usually represented by either Euler-Newton's equation or Euler-Lagrangian's equation. However, the first mathematics seems more comprehensible than the latter. Hence, the inputs:

- The total thrust: $T = \tau_1 + \tau_2 + \tau_3 + \tau_4$
- The rolling moment: $M_\phi = l(\tau_3 - \tau_4)$
- The pitching moment: $M_\theta = l(\tau_1 - \tau_2)$
- The yawing moment: $M_\psi = v_1 + v_2 - v_3 - v_4$

These inputs are based on Figure 2.1 with v_i and τ_i represent normalized thrust from i th rotor respectively (C Balas, 2007, without permission).

The Euler angles are used to model the rotation of the quadrotor in the body frame; producing the first equation.

$$\text{This } R = \begin{bmatrix} c\psi c\theta & c\psi s\theta s\phi - s\psi c\phi & c\psi s\theta c\phi + s\psi s\phi \\ s\psi c\theta & s\psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - c\psi s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \quad (1)$$

'c' and 's' in the matrix represents cos and sin respectively. Note that the R is orthogonal and thus producing an equal value of R^{-1} and R^T which is the rotation matrix that relate inertial frame and the body frame.

The rotational matrix needed to be presented and understood before proceeding into detailed equations of motion. (Schwegel, M. 2015). This matrix actually involves Cardan angles but loosely followed the Euler angles notation. Euler angles rotation known to play around two axes which is ZYZ-scheme while Cardan angles involve all three axes (i.e XYZ-scheme).

The rotation for example from A to B can be one by a matrix of A_{AB} which constitute of 3 by 3 orthogonal matrices with determinant and eigenvalue of 1. The rotational matrix, R can be found by expressing vector B^a (body) into base I (inertial) by depicting its unit vectors in terms of the target coordinate system.

Say,

$$I^a = [Ie_x^B \ Ie_y^B \ Ie_z^B] \cdot B^a \quad (2)$$

Where the values in the column are represented by A_{IB} .

Noting that the orthogonality property of matrix 3 by 3 can be used for transposition and thus $A^T = A^{-1}$, coordinate transform from I to B becomes:

$$A_{BI} = A_{IB}^{-1} = R_{BI}^T = R_{BI}^{-1} \quad (3)$$

By using (2), the rotation around the z axis by the positive angle of ψ produces:

$$A_{10} = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4)$$

Which is the yaw angle in vehicle dynamics. Following (4) is the elementary rotation about the x-axis represented by the ϕ angle and about y axis represented by θ angle. The matrices correspond to the angles are:

$$A_{21} = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \quad (5)$$

$$A_{32} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \quad (6)$$

These representations of a vector can now be found by applying the three rotation consecutively as:

$$3^c = A_{32} A_{21} A_{10} \cdot 0^c \quad (7)$$

And;

$$[A_{32} A_{21} A_{10}] = A_{30} \quad (8)$$

Considering the 0-system as Earth fixed and as the inertial system, and using $A_{30} = A_{BI} = A_{IB}^T$, the vector in body axes system can be expressed as R.

Other than rotation matrix about Euler angles, another parameter that might come in handy are the absolute linear position. In quadcopter, it is defined in the inertial frame as x,y and z while the angular position is represented by the Euler angles. In the body frame, the linear velocities defined by V_x, V_y, V_z and angular velocities by p,q and r. Matrices below showed the subscripts of each parameters for an easier references in this paper.

$$\xi = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad \eta = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \quad (9)$$

$$V = \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} \quad v = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (10)$$

Moving on, the transformation matrix for angular velocities that relating the inertial frame to the body frame is $W\eta$ while the inverse of the symbol represents the transformation

matrix of body frame to the inertial frame.

$$\dot{\eta} = W\eta^{-1} v; \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & s_{\phi}t_{\theta} & c_{\phi}t_{\theta} \\ 0 & c_{\phi} & -s_{\phi} \\ 0 & s_{\phi}/c_{\theta} & c_{\phi}/c_{\theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (11)$$

And thus,

$$v = W\eta \dot{\eta}; \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -s_{\theta} \\ 0 & c_{\phi} & c_{\theta}s_{\phi} \\ 0 & -s_{\phi} & c_{\theta}c_{\phi} \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (12)$$

c,s and t represent cos, sin and tan respectively.

As in Figure 2.1, the quadcopter is assumed to be symmetric with four equal arms aligned with the body x and y axes. Thus, the inertia matrix I is defined by:

$$I = \begin{bmatrix} I_{xx} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I_{yy} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_{zz} \end{bmatrix} \quad (13)$$

Also in the same figure, there are angular velocities rotating around the rotors, denoted by ω . Let's take a look at the fourth rotor, ω_4 create f_4 force in the direction of rotor axis. It then establishes an according amount of torque, τ_{M4} . This notation applied to each rotor and by the equation (14) below, the value of f and τ_M can be found.

$$f = k\omega^2 ; \tau_M = b\omega^2 + I_M\dot{\omega} \quad (14)$$

k in the equation represents the lift constant while b is the drag constant. I_M is known to be the inertia moment of the rotor. $\dot{\omega}$ is negligible since the value is way too small.

These forces combination then formed thrust, T in the vertical direction of z axis. Equation (15) and (16) below explain the movement of the quadcopter.

$$T = \sum_{i=1}^4 f_i = k \sum_{i=1}^4 \omega_i^2, T = \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} \quad (15)$$

$$\tau = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} l k (-\omega_2^2 + \omega_4^2) \\ l k (-\omega_1^2 + \omega_3^2) \\ \sum_{i=1}^4 \tau_{Mi} \end{bmatrix} \quad (16)$$

l in the equation is the distance between the centre mass of the body of the quadrotor to the rotor. In equation (16), the torque or the roll movement can be achieved by reducing the angular velocity of rotor 2 and increase the velocity of rotor 4, the pitch movement on the other hand can be obtained by lowering the velocity of rotor 1 and rising the velocity of rotor 3. Yawing action can be acquired by increasing the angular velocities of rotor 1 and 3 and decreasing the velocities of rotor 2 and 4. Note that these pairs are located oppositely from each other.

2.1.1 Euler-Newton equations.

As mentioned earlier, this paper will focus on Newton-Euler equations which is easier to be understood compared to Euler-Langragian equations. Equation (17) shows describe the dynamics of the quadcopter which is assumed to be rigid body.

$$m\dot{V} + v \times (m\dot{V}) = R^T G + T \quad (17)$$

Know that:

$v \times (m\dot{V})$: centrifugal force

T : total thrust of the quadcopter body frame

However, in the initial frame, the force is nullified. So, only gravitational force and the magnitude and direction of the thrust are the contributor that accelerates the vehicle.

In other words, equation (17) become:

$$m\ddot{\xi} = G + RT \quad (18)$$

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ -g \\ 1 \end{bmatrix} + \frac{T}{m} \begin{bmatrix} c_\psi s_\theta c_\phi + s_\psi s_\phi \\ s_\psi s_\theta c_\phi - c_\psi s_\phi \\ c_\theta c_\phi \end{bmatrix} \quad (19)$$

Meanwhile in the body frame,

$$I\dot{v} + v \times (Iv) + \Gamma = \tau$$

$$\dot{v} = I^{-1} \left(- \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} I_{xx}p \\ I_{yy}q \\ I_{zz}r \end{bmatrix} - I_r \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega_\Gamma + \tau \right) \quad (20)$$

Recall that; $v = \begin{bmatrix} p \\ q \\ r \end{bmatrix}$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} (I_{yy} - I_{zz})qr/I_{xx} \\ (I_{zz} - I_{xx})pr/I_{yy} \\ (I_{zz} - I_{yy})pq/I_{zz} \end{bmatrix} - \begin{bmatrix} q/I_{xx} \\ -p/I_{yy} \\ 0 \end{bmatrix} \omega_\Gamma + \begin{bmatrix} \tau_\phi/I_{xx} \\ \tau_\theta/I_{yy} \\ \tau_\psi/I_{zz} \end{bmatrix} \quad (21)$$

ω_Γ in this context is $\omega_\Gamma = \omega_1 - \omega_2 + \omega_3 - \omega_4$.

The angular accelerations in the inertial frames are then attracted to the body frame accelerations with the transformation matrix W_η^{-1} and its time derivative:

$$\begin{aligned} \ddot{\eta} &= \frac{d}{dt}(W_\eta^{-1}v) = \frac{d}{dt}(W_\eta^{-1})v + W_\eta^{-1}\dot{v} \\ &= \begin{bmatrix} 0 & \dot{\phi}c_\phi t_\theta + \dot{\theta}s_\phi/c_\theta^2 & -\dot{\phi}s_\phi c_\theta + \dot{\theta}c_\phi/c_\theta^2 \\ 0 & -\dot{\phi}s_\phi & -\dot{\phi}c_\phi \\ 0 & \dot{\phi}c_\phi/c_\theta + \dot{\phi}s_\phi t_\theta/c_\theta & -\dot{\phi}s_\phi/c_\theta + \dot{\theta}c_\phi t_\theta/c_\theta \end{bmatrix} v + W_\eta^{-1}\dot{v} \quad (22) \end{aligned}$$

The Euler-Newton equation above is also a representation of the equation of motion that involves rotational and translational variables.

2.1.2 The Flight Operation.

Figure 2.3 to Figure 2.6 explained about the flight operation of the quadcopter and how the rotation about the Euler angles affects the movement of the UAV.

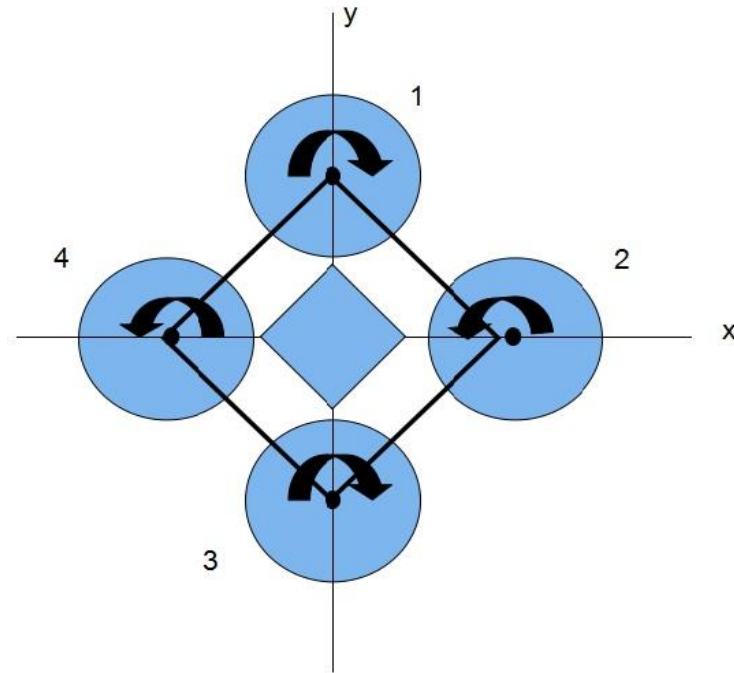


Figure 2.3: Schematic Diagram of Rotors Reaction

Each rotor produces a thrust and torque about its centre of rotation and works as pair of two identical fixed pitched propellers. Based on the diagram, rotor 1 and 3 produces a clockwise angular velocity (ω) while rotor 2 and 4 produces a counter clockwise angular velocity. As complied by a fixed wing aircraft, quadcopters also produce drag force opposite to its direction of flight. As for the reaction produces by the rotors, it then generates an equilibrium aerodynamic torque and thus cancel out the angular acceleration about the yaw axis. (ψ).

Just like any conventional helicopters, quadcopters also capable of adjusting its altitude and position about the yaw, pitch and roll axis. What so interesting about flying a quadcopter is its properties which includes the weight supported by each rotor. For quadcopter with four rotors, the weight is distributed equally among each rotors which means each motor only lifts quarter of the weight of the quadcopter.

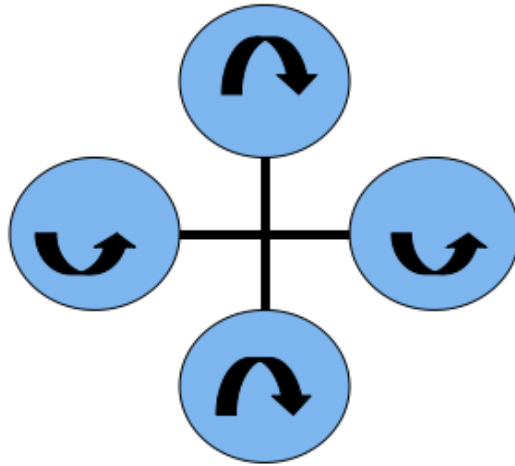


Figure 2.4: Hover:

By applying an equilibrium torque at each rotor, the quadrotor can hover and adjust its altitude.

The two pairs of rotors that rotates at different torque direction allow the quadcopter to stay facing the same way without rolling around. From now, control becomes the matter of fact on which the position of the quadcopter

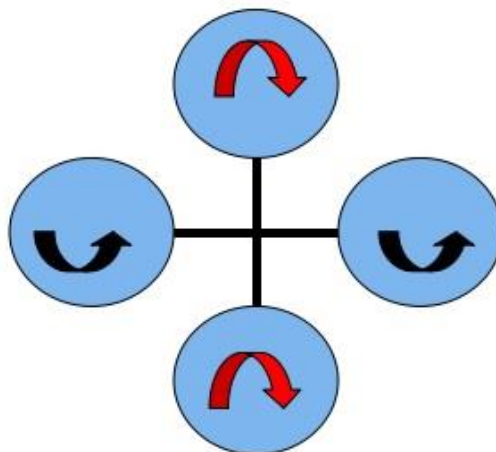


Figure 2.5: Yaw:

Yawing can be achieved by adding more forces at a pair of rotors that rotates in one direction.

Note that to achieve this movement successfully, the pilot needs to determine the

front part of the quadcopter to avoid out of control flight. To perform this action, control becomes the focus of matter in which pair of rotors received more power or torque. Usually, the clockwise rotors are determined as the front.

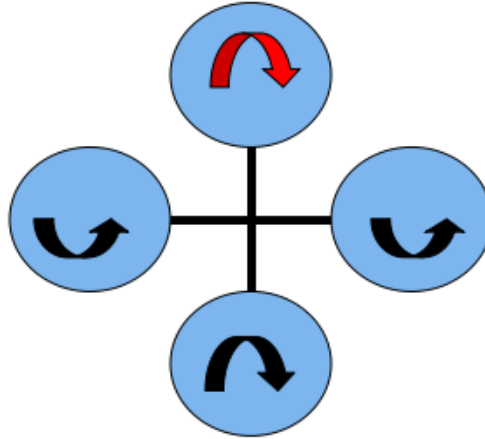


Figure 2.6: Rolling and pitching:

Can be attained by putting more speed at one rotor only. Rolling and pitching usually determined by its front.

Rolling can be achieved by putting more power to one motor while decreasing the power on the opposite one. Pitch on the other hand uses a different pair of sets. Whilst this action may be confusing, the pilot needs to understand the movement made by the quadcopter. Pitch action can be performed just like nodding, yaw when turning to left and right and roll by performing a circular formation while still facing the same direction.

2.2 Linear Controller

2.2.1 PID Controller.

Also known as Proportional Integral Derivative Controller. This mechanism is common in industrial control system due to its simple and straightforward manner. This controller has amazingly robust performance and hence, its popularity in engineering devices.

As the name suggests, PID controller exploits the three basic coefficients; proportional (present error), integral (accumulation of past error) and derivative (prediction of future error) to get an optimum response. This controller is mainly used in closed-loop system and the inputs variables can be adjusted accordingly to get the desired results. Quadcopter uses PID controller to achieve stability.

By estimating the error, control over the quadcopter can be achieved. This function can only be attained by measuring the quadcopter output sensor; for instance, what is the desired angle to make the drone yaw. By applying these three control algorithms to the error, the output values finally reach the motors to correct the error made. Each of these controllers are unique and could determine the flight characteristic of the rotorcraft. (Oscar Liang,2013).

To design this controller, the model has been linearized around the hover condition. Hence, no gyroscopic effect will be discussed further in this paper. Generally, PID controller can be utilised by having the following equation (K.J. Astrom et al,2006)

$$e(t) = x_d(t) - x(t)$$
$$u(t) = K_p \cdot e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt} \quad (23)$$

$u(t)$ in the equation is the control input, $e(t)$ is the difference between desired state $x_d(t)$ and present state $x(t)$. K_p , K_i and K_d is the parameters for the proportional, integral and derivatives for the controller respectively.

Further in this paper, the closed loop model of PID will be simulated on MATLAB's Simulink. The parameters will be decided and adjusted to suit the model through real life data sheets.

Recall that quadcopter has six degrees of freedom, (positions ξ and angles η), but they only have four control inputs which is the angular velocities of the rotors. From Euler-Newton's equations, we see the development of relation between the angular velocities up to the thrust produced by the vehicle to stay in the air. Know that thrust affects the acceleration in z-axis while the torque accompanied by the Euler angles affect the accelerations of the quadcopter in respective axis accordingly. It is easier to write as:

$$\begin{aligned}
 T &= (g + K_{z,D}(\dot{z}_d - \dot{z}) + K_{z,P}(z_d - z)) \frac{m}{c_\phi c_\theta} \\
 \tau_\phi &= (K_{\phi,D}(\dot{\phi}_d - \dot{\phi}) + K_{\phi,P}(\phi_d - \phi)) I_{xx} \\
 \tau_\theta &= (K_{\theta,D}(\dot{\theta}_d - \dot{\theta}) + K_{\theta,P}(\theta_d - \theta)) I_{yy} \\
 \tau_\psi &= (K_{\psi,D}(\dot{\psi}_d - \dot{\psi}) + K_{\psi,P}(\psi_d - \psi)) I_{zz}
 \end{aligned} \tag{24}$$

In the equation, g is gravity, m is mass and I is the inertia of the quadcopter.

2.2.2 LQR Controller.

LQR stands for Linear Quadratic Regulator which reduces the amount of work done by the control systems engineer to optimize the controller. This controller's theory is concerned with operating a dynamic system at minimum cost. LQR algorithm is essentially an automated way to find an appropriate state-feedback controller. In the past, there are a lot of study that have been performed to find the best way to stabilize the quadcopter by using LQR controller. A lot of researchers used classic LQR technique in closed-loop. However, it is found that at low thrust level, the control was acceptable but once it is thrust to a higher level, the performance was exhausted due to vibrations. With this findings, many attempts has been done to make the quadrotor hover correctly. Castillo in his book, *Stabilization of a Mini Rotorcraft with Four Rotors* (2005), is one of the researcher that implemented iteratively from simulation results LQR controller and applying the feedback obtained to y and ϕ .

Now, there are a lot of LQR types that has been modified that uses modern techniques to design the aircraft's system. One of which is LQR with output feedback. By regulating certain states of the crafts to zero, the desired closed loop characteristics can be obtained. Though this method involves the problem of stabilizing the aircraft by placing the closed-loop poles at desirable location.

Once the performance criterion has been selected, the control gains are explicitly computed by matrix design equation and closed loop stability will generally be guaranteed (Brian L.Stevens et al,200). In 2004, Bouabdallah implemented LQR controller in the closed loop to stabilize the UAV.

2.3 Comparison between PID Controller and LQR Controller

In any ways, the controller is actually better depending on the desired output of the system. LQR is a controller that gives the optimal state-feedback law that minimizes certain quadratic objective function. It is quite an assurance to say that LQR is a better controller compared to PID (Amin Mohammadbagheri et al,2011). However, it has drawback that assumes all the states of the system are measurable. If not, an observer that estimates the states by observing the measurable output is required. LQR requires an analytical model of the system its controlling with or without observer. Additionally, if the system model is not linearized, it cannot proceed through. So, as to this, LQR is more complex than PID.

In contrast, PID is way simpler and can be tuned without having any analytical modelling. It is also more applicable for any commercial application and research due to its favourable aspects which provide simple tuning procedures for those who just wants to learn about controller. Of course, by referencing to Demet Canpolat Tosun et al, 2015, LQR controller deals with balances between low control effort and has a faster response while PID provide only small stability in its system. However, note that PID controller is more widely used in quadcopters and the parameters required are easily obtainable compared to the LQR controller.

2.4 Trajectory Development

Over the past few years, the development of path according to the pre-programmed algorithm for unmanned aerial vehicles has been on the uprising. Interested researchers from many fields had gather around and create a trajectory track and tested it upon specific requirements of the missions. Several concepts of trajectory have been introduced since the UAV debut. The findings on sensor coverage of a specified area has allocated many interests researchers in this topic. Mostly they are focusing on advancement that can be done to upgrade the whole system of a UAV. Nevertheless, the challenges are getting bigger as the refinement continued to develop. One of which is to control the motion of the drones to drive the sensor to carry out specific task such as exhaustive coverage of an area (Jiankun Fan,2014) . Noted that these enhancements comes with navigation solution from the geometric planning techniques which is the autonomous system. There has been a lot of work focused on developing planning strategies recently. This includes Potvin's research on delivery systems, autonomous UAV path planning , and mobile sensor network .

As for the track for drones, L.E. Dubins was the first person who did the research on the time-optimal paths of a simple airplane. Thus the existence of Dubins curve theory. Dubins curve always list as: LSL, RSR, RSL, LSR, RLR, and LRL with;

S means straight line segment

L is a circular arc to the left

R stands for the circular arc on the right side.

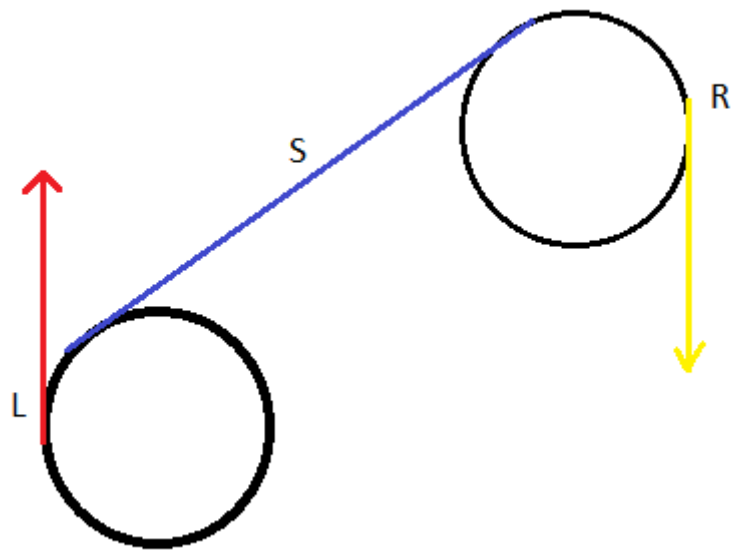


Figure 2.7: The Dubin's Curve

He designed the shortest path by giving a characterization of time-optimal trajectories for a car with a bounded turning radius. Following that, a research has been conducted by Reeds and Shepp where they try to give a path with a minimum turning radius on a designated model (Reed, J.A et al, 1990). Only afterwards, Boissonnat, C'er'ezo, and Leblond solved the same problem by using theory named "Minimum principle of Pontryagin" which enable them to get the shortest path between two oriented points.

3.0 Methodology

3.1 Choice of Trajectory

In this paper, the trajectory path that will be design will fulfil the mission constrains of motion of the quadcopter. These paths are circular, sinusoidal, and random coordinates with obstacle. It is also important to obtain the optimize time taken for the quadcopter to finish the coverage task.

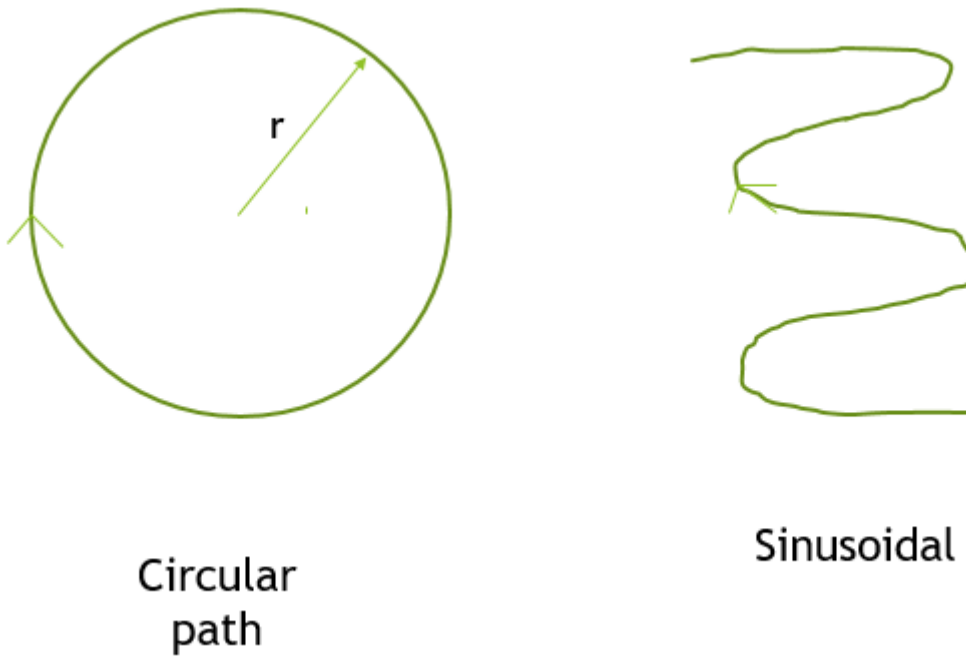


Figure 3.1: The desired trajectory design

However, in order to perform the task, the trajectory equation is first need to be found. The path supposedly will be designed in 3D and thus required an understanding in space. For circular, the parameterization of the circle of radius r around the axis, centred at (c_1, c_2, c_3) is given by:

$$\begin{aligned}x(\theta) &= c_1 + r \cos(\theta) a_1 + r \sin(\theta) b_1 \\y(\theta) &= c_2 + r \cos(\theta) a_2 + r \sin(\theta) b_2 \\z(\theta) &= c_3 + r \cos(\theta) a_3 + r \sin(\theta) b_3\end{aligned}\tag{25}$$

The unknown (centre position, radius and theta angle) is yet to be found to obtained the desired and actual trajectory of the quadrotor following the circular track.

Following that, a sinusoidal trajectory path is chosen to be one of the tested design. The trajectory equation used in MATLAB for the simulation is

$$z = \cos(k_1x) \times \cos k_2y \quad (26)$$

The value of k_1 and k_2 also yet will be determined.

The equation already been tested with undecided values and able to form the desired trajectory.

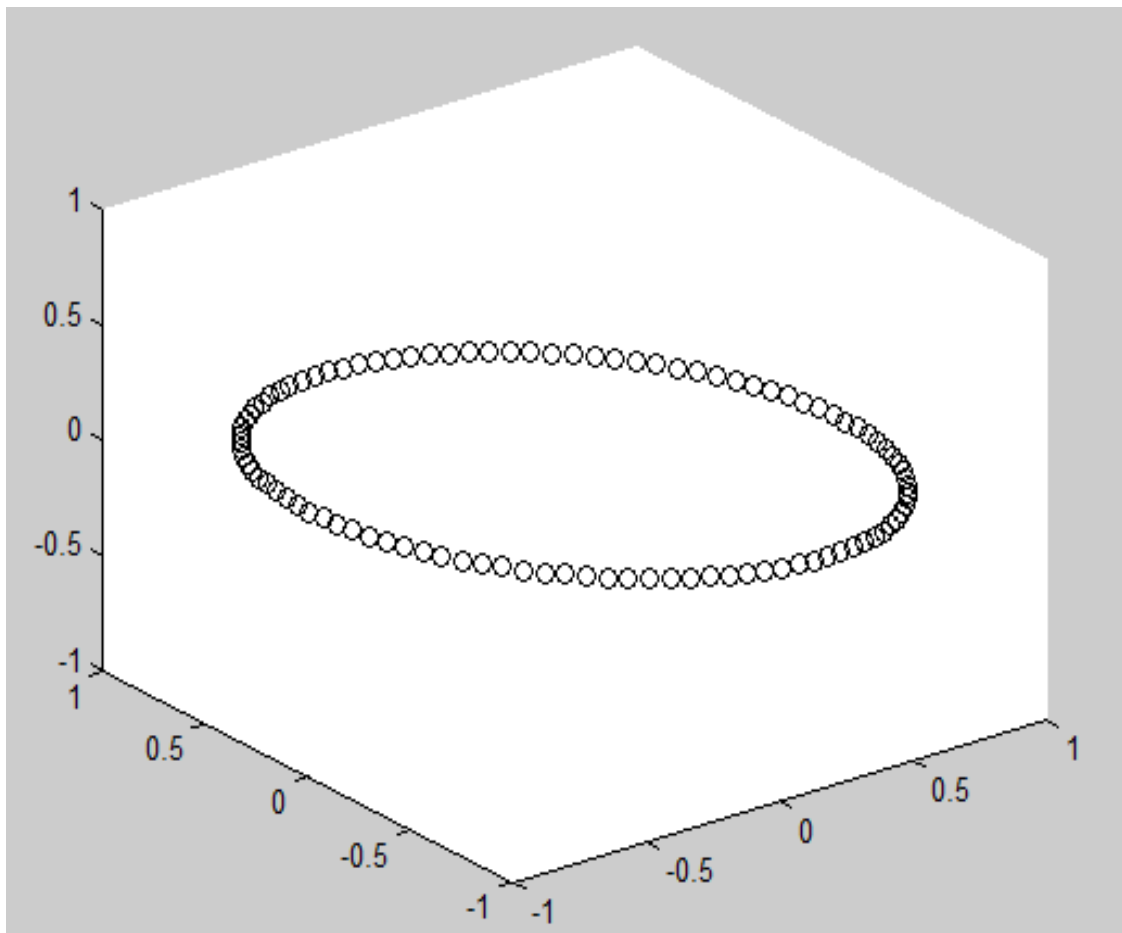


Figure 3.2: The circular path in 3D.

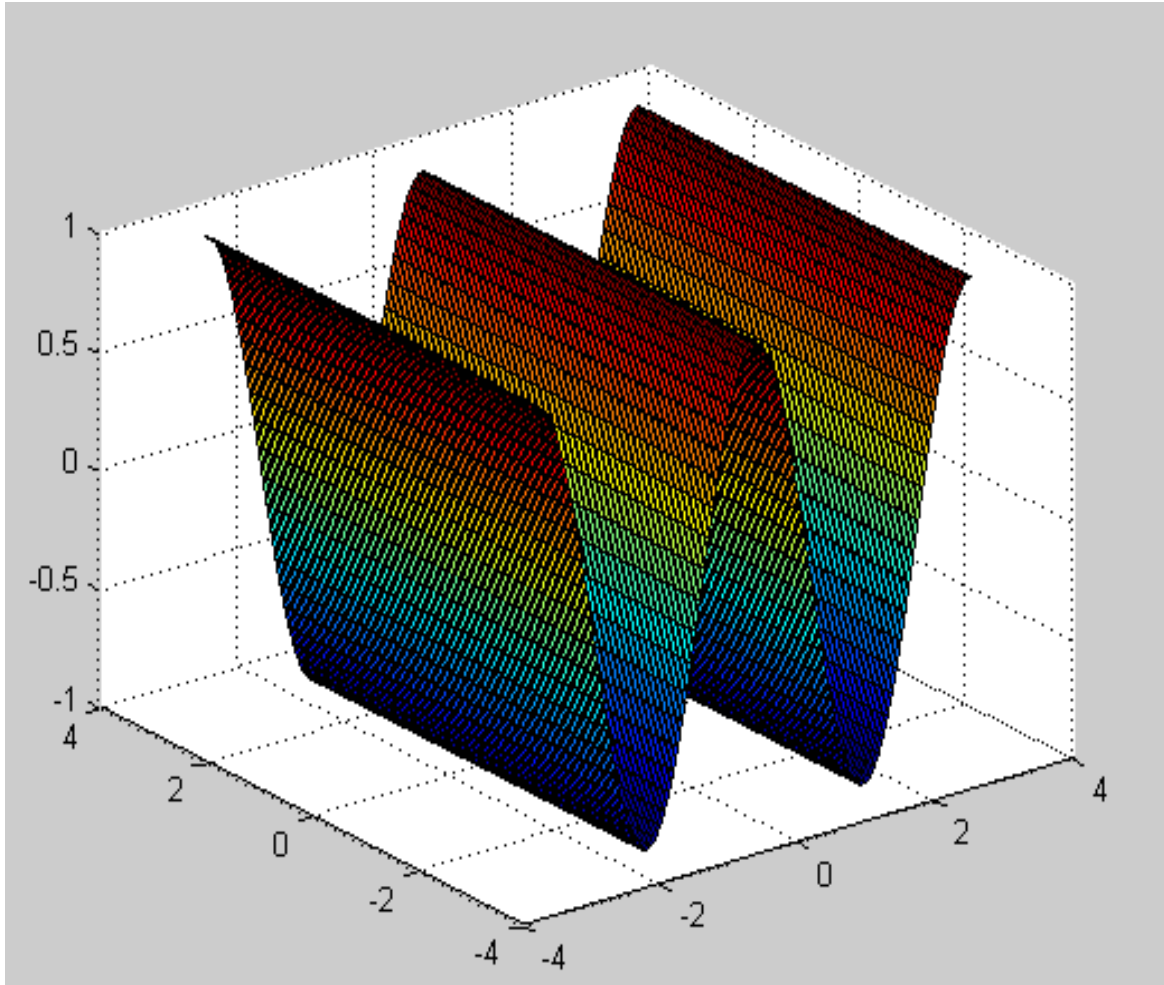


Figure 3.3: The sinusoidal path in 3D

The corresponding MATLAB coding is attached in Appendix.

If these trajectories are successful in tracking the designed path, the random coordinate with obstacle will be applied and the time optimization will be found for the drone to complete the trajectory.

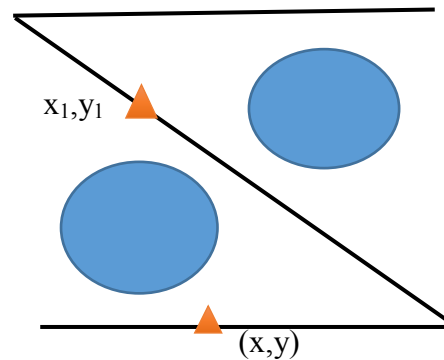


Figure 3.4: The random coordinates with obstacle.

The obstacle in this course will be represented as building or trees and the drone will be commanded to track the path according to the formation that yet will be decided in the next semester.

In developing the trajectories, a controller needed to be determined. As opted out in the previous section, PID controller is the preferable option compared to LQR Controller. The decision is made due to the referenced quadcopter which is Walkera Scout X4 that uses PID and its controller and of course it is so much simpler to find its required parameters to find the gain needed to run the algorithm.

3.2 Simulink

3.2.1 Dynamic Model Simulation.

First on hand, there is a need to understand the rotor dynamics modelling of the project. Figure 3.2.1 below shows the diagram that simplified the considerations that needed to be done in order to perform the objective.

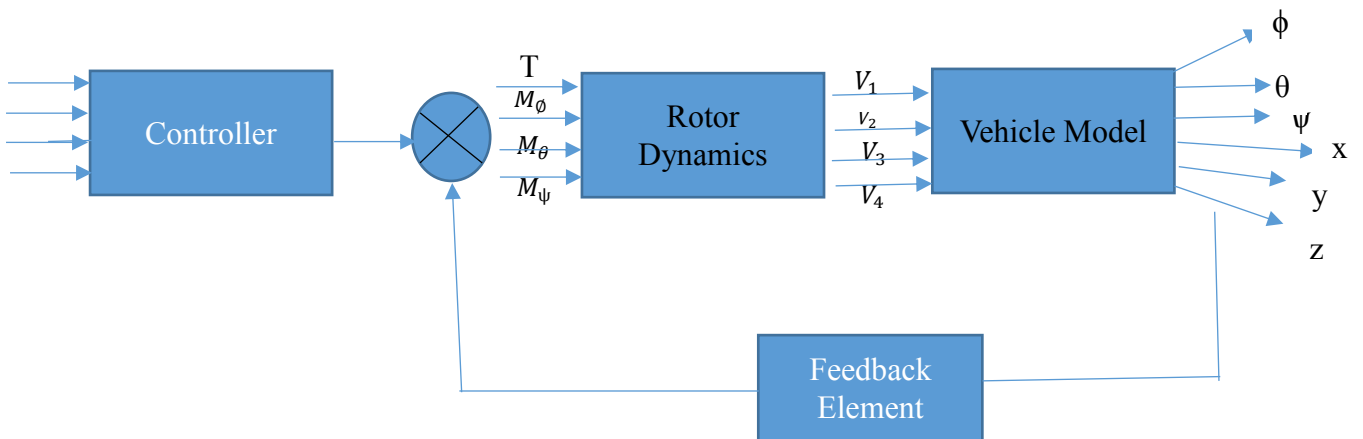


Figure 3.5: The nested rotor dynamic for the system

In this section, numerical simulation will be provided through MATLAB Simulink. The data from previous parts are used to simulate the dynamic model. The quadcopter model chosen for this simulation is Walkera Scout X4 which provided the following parameters:

Table 1: Parameters for dynamic model

Parameters	Value
m	1.77 kg
g	9.81 m/s ²
l	0.2 m
I _{xx}	0.1416
I _{yy}	0.1416
I _{zz}	0.2832

The inputs for the controller is also yet to be decided. However, as in for now, the understanding that instead of having six degrees of freedom, the quadcopter only able to accommodate four significant inputs. In the previous published paper done by C Ballas in his work, Modelling and Linear Control of A Quadcopter, he used a lot of parameters and made comparisons for the desired simulation. For instance, in his first rotor dynamic modelling, the input is x,y,z and the second order of ψ . The decoupling of the parameters is done to obtain the necessary output that he wanted and those outputs became the input for his rotor dynamic. By using voltage difference method, he managed to obtain the vehicle dynamic for the quadcopter and finally, the desired output was obtained to run the quadcopter according to the trajectory designed.

In other findings, they make use of all the translational and rotational parameters as the inputs in second order system and obtained the output as the first order that works on the quadcopter. For this paper, the inputs will be understood and decided later.

In Figure 3.2.1, notice that there is a feedback element that needed to take care of. Not only that, the controller, as in section 2.4 is chosen to be the PID Controller. This controller need gain to achieve stability of the drone. To find the related gain, Flight Stability And Automatic Control by Nelson (1998) is used as the reference.

From the book, it is important to find the transfer function first before we can stabilize the error in the closed loop system. Next, the gain will be find by plotting root locus using the obtained transfer function. A trial coding is done to ensure that the parameters work with the gain used and the result is attached in the Appendix.

3.3 Expected Results

The final product of the study is of course to obtain the Matlab's Simulink and get the time optimization for the trajectory designed. To be able to control the quadcopter and perform the task according to its algorithm and tracks required a lot of interactions between the equations presented in the previous section. It does not necessarily only involve the equation of motion but more than that. The trajectories equation are also important to design the desired path to be tested by the quadcopter. So, that is main reason why the inputs are very important because it also determine the output of the system.

A lot of assumption are made upon completion of the objectives in which involve the neglectation of aerodynamical effects and gyroscopic effects. In other words, the quadcopter is in perfect condition and was made to be in hovering state.

However, after the presentation, the panels already suggested that noise should be added into the study. With noise, the usage of controller will make more sense since the controller is used to nullify the errors occur in the trajectory. The simulation, on the other hand, hopefully will provide good results and the drone can freely follows its trajectory track around the equilibrium state.

The future work as in for this paper is to provide the missing information and to obtain the desired result of the study. Flow chart below shows the flow of the work done up till now to the work that will be done in the next semester.

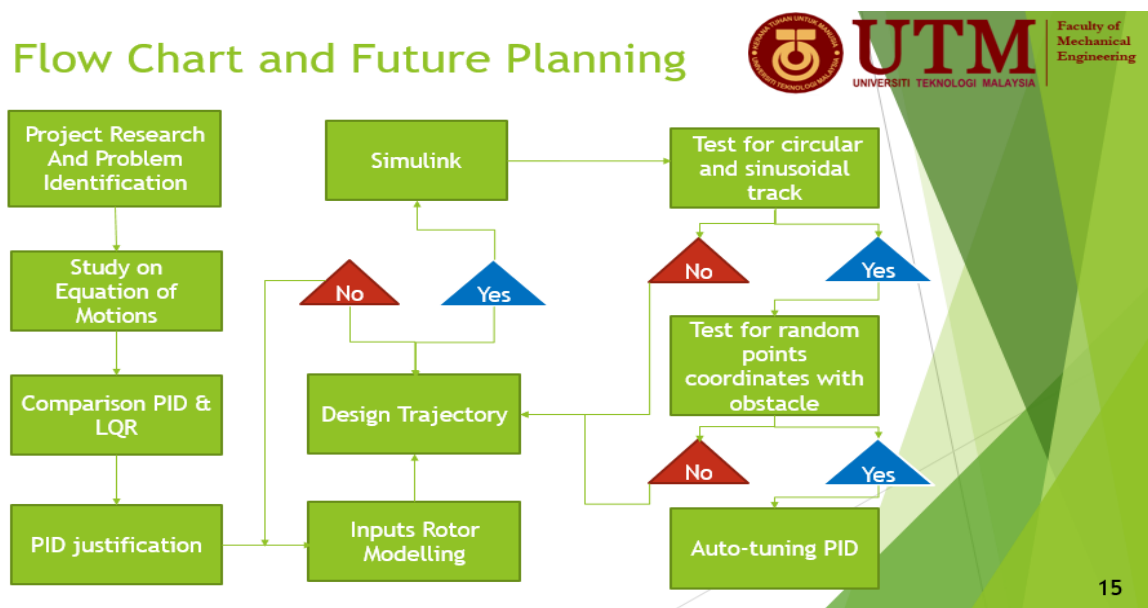


Figure 3.6: The Flow Chart and Future Planning of the Study

Up to this day, the inputs for rotor modelling is yet to be determined. However, some of the works has been done for the algorithm like finding the root locus for the PID Controller and finding the trajectory equation needed in the project.

In next semester, the design trajectory will be fully updated and Simulink may be able to develop. Upon that, the testing for the trajectories design will be done. If the test became successful, another part of the study will be explored which involve the Auto Tuning of the PID Controller.

3.4 Reference:

1. Teppo Luukkonen,2011. *Modelling and Control Quadcopter*. School of Science, Aalto University.
2. C Balas,2007. *Modelling And Linear Control Of A Quadrotor*. Cranfield University.
3. Schwegel M,2015. *Design and Control of a Quadcopter*.
4. J.A Reeds and L.A Shepp,1990. *Optimal paths for a car that goes forwards and backwards*. Pacific J Math, pp 367-393.
5. Hamidreza Chitsaz and Steven M LaValle, *Time-optimal Path for a Dubins Airplane*.
6. Jiankun Fan,2014. *Optimal path Planning And Control of A Quadrotor Unmanned Aerial Vehicle for Area Coverage*. University of Toledo.
7. Robert C. Nelson,1998. *Flight Stability And Automatic Control, 2nd edition*. McGraw-Hill International Edition.

4.0 Appendix

1. Coding for 3D Circular (undetermined values)

```
% Original points, original plane
t = linspace(0,2*pi);
x = cos(t);
y = sin(t);
z = 0*t;
pnts = [x;y;z];
% unit normal for original plane
n0 = [0;0;1];
n0 = n0/norm(n0);
% unit normal for plane to rotate into
% plane is orthogonal to n1... given by equation
% n1(1)*x + n1(2)*y + n1(3)*z = 0
n1 = [1;1;1];
n1 = n1/norm(n1);
% theta is the angle between normals
c = dot(n0,n1) / ( norm(n0)*norm(n1) ); % cos(theta)
s = sqrt(1-c*c); % sin(theta)
u = cross(n0,n1) / ( norm(n0)*norm(n1) ); % rotation axis...
u = u/norm(u); % ... as unit vector
C = 1-c;
% the rotation matrix
R = [u(1)^2*C+c, u(1)*u(2)*C-u(3)*s, u(1)*u(3)*C+u(2)*s
     u(2)*u(1)*C+u(3)*s, u(2)^2*C+c, u(2)*u(3)*C-u(1)*s
     u(3)*u(1)*C-u(2)*s, u(3)*u(2)*C+u(1)*s, u(3)^2*C+c];
% Rotated points
newPnts = R*pnts;
plot3(newPnts(1,:),newPnts(2,:),newPnts(3,:), 'ko')
```

2. Coding for sinusoidal formation 3D (undetermined value)

```
k1=2;
k2=0;

[X,Y]=meshgrid(linspace(-pi,pi));
Z=cos(k1*X).*cos(k2*Y)
surf(X,Y,Z)
```

3. Coding of trial and error for transfer function and root locus to find gain of PID Controller

```
s = tf('s');  
G = 3.8*(s + 0.1);  
H = (s + 10)*(s^2 + 0.039*s + 0.053);  
Oloop = G/H  
  
% determine gain Ka1, Ka2, Ka3 from root locus  
  
figure(1)  
rlocus(Oloop)  
grid on
```