INTERNATIONAL UNIVERSITY LIAISON INDONESIA (IULI)

Numerical Study of Baseline Design for Agricultural Multirotor Drone

Presented to the Faculty of
Engineering

In Partial Fulfilment
Of the Requirements for the Degree
Bachelor of Sciences
In
Aviation Engineering

By
Aloysius GUNTUR Pratama

July 17, 2019

"Nothing takes place in the world whose meaning is not that of some maximum or minimum."

Leonhard Euler

INTERNATIONAL UNIVERSITY LIAISON INDONESIA (IULI)

Abstract

Faculty of Engineering
Department of Aviation Engineering

Bachelor of Engineering

Numerical Study of Baseline Design for Agricultural Multirotor Drone

by Aloysius GUNTUR Pratama

A quad-copter is a multi-rotor copter developed as Unmanned Aerial Vehicle (UAV) which is lifted and propelled by 4 motors. At small size, a quad-copter is cheaper and durable. This make a quad-copter is easy to build, due to the mechanical simplicity, and can be developed for many purposes The 4 propellers attached on vertical motors in quad-copter make it possible to maneuver freely during the flight.

The disturbances from air pressure difference, swinging load, and load distribution difference on the motors often experienced by quad-copter. Therefore, finding a functional configuration that can be used as a baseline design for the agricultural drone is required to fulfil the needs of modern agriculture.

To approach the objective, the numerical simulation about the motion of quad-copter under the variation of mass is performed by using Python. The variables for the numerical simulation is determined after rescaling the design of quad-copter by trial and error. The data about the drone performance which had been designed is obtained after seizing the numerical method simulation of the quad-copter.

By this research, an assessment of the design performance of an agricultural drone can be compared with the baseline design, which has a numerical comparison as a guidance to determine the proper configuration for an efficient prototyping.

Statement by The Author

I hereby declare that this submission is my own work and to the best of my knowledge, it contains no material previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any other degree or diploma at any educational institution, except where due acknowledgement is made in the thesis.

Aloysius Guntur Pratama	
Student	Date

Committee Approval

Numerical Study of Baseline Design for Agricultural Multirotor Drone

Aloysius GUNTUR Pratama

July 17, 2019

Faculty of Engineering

Triwanto Simanjuntak, PhD
Advisor I, Department of Aviation Engineering — Date

Neno Ruseno S.T., M.Sc.
Advisor II, Department of Aviation Engineering — Date

Dr. Ir. Prianggada Indra Tanaya, M.M.E.
Dean of Engineering — Date

Acknowledgements

First of all, I would like to thank God for His kindness during this thesis work. I would like to thank and express my deep sincere gratitude to my advisor Mr. Triwanto for always support me and my bachelor study and research, for his patience, motivation, enthusiasm, and immense knowledge. His speech and guidance are very helpful for me during the time of the research and writing of this thesis. I could not have imagined having a better advisor and mentor.

Besides my advisor, I would like to thank all the people who help me during this work especially my friend and juniors in department of aviation engineering, Prana Rao for having me the CAD software design of quad-copter for this research. Without his mastery skill in CAD design, I wouldn't know what will happen. And also Giovanni Henggar Setyantoro, my best friend and very first friend since I attend the university, who keeps motivating me and help me with the coding for the simulation.

I thank all of my classmates in aviation engineering for the stimulating discussions, for the sleepless nights we were working together before deadlines, and for all the fun we have had in the last four years. Also I thank my best friends : Ignatius Yoel Ferdinand, Laurentius Adityo (Petot), Alexander Glenn, Florensia Witty, Farras Chaniago (Noh Man I), Nadim AR (Noh Man II), Aaron(Noh Man III), I Gede S (Mad Man), Abraham Enga (Abam), V. Kidung Larasati, Gabriella Kirana, Klara, Mega Hutagaol, Saras, Kak Adisti, and also Bernadette Dwita who are consciously and willingly prepared to be disturbed by me during this thesis work.

Last but not the least, I would like to thank my family: my parents Rudianto Winarto and Sesilia Trimiyati, for always not giving up on me until this time.

Contents

List of Figures

List of Tables

List of Abbreviations

UAV	**U**nmanned **A**erial **V**ehicle
COG	**C**enter **O**f **G**ravity
CAD	**C**omputer **A**ided **D**esign
RPM	**R**otation **P**er **M**inute
ESC	**E**lectronic **S**peed **C**ontrol

Dedicated to my parents

Chapter 1

Introduction

1.1 Introduction

Research Background

The technology development has bring human to the new era of industry. Especially in farming, which is often considered as a traditional way, now it has combined elements of technology in every stage of agricultural process. With technological advancements, the new farming method has been implemented to improve the quality of food production. Driven by the basic need for food and feeding an ever growing population, an era of using powered machineries has been opened to replace the work formerly performed by human and animals.

All of these machines have greatly increased farm output and changed the way people work and produce food worldwide. Utilization of tractor is an example of agricultural machinery. Mechanized farming also involves the use of aircraft and helicopters. Recently, utilization of drones are to begin to enter other aspects which are not previously imagined. This include the using of the technology modern agriculture[1]. An agricultural drones let the farmers see their whole field from the sky to inspect many issues about irrigation, fungal and pest infestations, and also soil variations. The drone is not only used for inspection but it can also be utilized for spraying the field with water, fertilizer, and pest control. All of these features can assist the farmers to improve their crop growth and production.

FIGURE 1.1: Agriculture Quad-copter Sprayer X-25 [2]

In particular, a multi-rotor drone has been designed to fulfil the need of modern agricultural problem. Typical modern day spraying drones have tank capacity of over ten litres of liquid pesticide with discharge rate of over a litre a minute, allowing them to cover a hectare in ten minutes [1]. For example, the multi-rotor drone has been designed for agricultural purpose has capability for watering and spraying the crop field. It carries a fluid in a water tank as payload. This fluid is certainly affects performance of the drone especially while the fluid carried by drone is discharging during the flight. In other words, when the drone performs a spraying procedure on crop field, the liquid carried by the drone will be release from the water tank periodically and continuously. The drone will experience mass change continuously as the liquid discharged. The effect of this whole system gives an impact in terms of stability and control of the drone. From this case, the need of designing the agricultural drone efficiently is very important.

Significance of Study

An efficient design of a multi-rotor copter for agricultural purpose is necessary. Because the modern agricultural has combined many technology to improve the production and quality of the crop field. Designing an efficient multi-rotor drone for this purpose would be challenging. Therefore the study of this research will be useful to improve the design of a multi-rotor drone for agricultural. The result from this research will provide the numerical comparison for the development of the baseline design of optimal prototype which can be useful for an agricultural multi-rotor drone.

1.2 Problem Statement

A quad-copter is a multi-rotor copter developed as Unmanned Aerial Vehicle (UAV) which is lifted and propelled by 4 motors. A quad-copter can be controlled from distance by human or fly autonomously. The 4 propellers attached on vertical motors in quad-copter make it possible to maneuver freely during the flight. In designing a multi-rotor the most important part is the propulsion system. This propulsion system should be able to give the multi-rotor enough thrust to fly.

Therefore in designing a multi-rotor there are several consideration to determine appropriate and suitable propulsion systems (motor and propeller):

- The thrust produced by the motor is influenced by the motor's and propeller's properties (size, mass, power, etc).
- Bigger thrust needs bigger motor but bigger motor gives extra weight to the vehicle hence it could require bigger thrust.
- Therefore, it's necessary to make a reliable approach to determine the propulsion system (motor efficiently).

Based on those problems, designing the multi-rotor copter for agricultural purpose that can fulfil the needs of watering and spraying the crop field is challenging.

Therefore, finding a functional configuration that can be used as a baseline design configuration for agricultural multi-rotor drone becomes the problem statement of this research.

1.3 Research Purpose

The primary purpose of this thesis are:

- To develop numerical based approach in determining the propulsion system.
- To develop a baseline design for agricultural drone.
- To test numerically the baseline design for predetermined sequence of flight or thrust command.

1.4 Research Scope

In order to keep the discussion stay focused on the problem, limitations and assumptions are given as follows:

1. Multi-rotor considered here is of a quad-copter type.
2. The quad-copter and the propeller are assumed as rigid bodies.
3. Due to symmetry of quad-copter, the product of inertia (I_{xy}, I_{yz}, I_{xz}) are all zero
4. Effect of mass discharged is ignored.
5. Feedforward command control only.
6. Gyroscopic Force is neglected.

1.5 Approach

The numerical approach is implemented for this research. Python scientific programming language is being used to simulate the quad-copter motion by numerical method as an approach to achieve the research objectives.

1.6 Thesis Outline

The outline of this thesis is given as follows:

- **Chapter I**: This chapter generally discusses about the introduction of the research, a background that underlies why the author wants to do the research about the quad-copter. The statement about the research problem which the author want to analyse from the study perspective. This introduction is divided into several sections including the research background, problem statement, research purpose, research scope, approach, and thesis outline. This chapter gives a brief outline of this research project so the readers know the scope for the further discussion. By the limitations within the scope of this study, it is expected that the discussion of this project will be more guided.

- **Chapter II**: This chapter focuses on the discussion about the theories of quad-copter, its motion, moment, and forces considered on a quad-copter during the flight. In this chapter the author describes about the theories applied for the research. The overview of the theories related to the research project are presented in this chapter. The brief explanation about the quad-copter is also given.

- **Chapter III**: This chapter discusses the research methodology which contains the design of quad-copter used for the numerical simulation, how to do the simulation with Python programming language, and analyse the data retrieved from the simulation.

- **Chapter IV**: This chapter contains about the results of the quad-copter motion simulation, the data processing and tests the suitability of theories with the simulation model that has been made. The discussion about the research is also included in this chapter.

- **Chapter V**: This chapter gives the final conclusion of the result from the whole research project and suggestions that contain recommendations from the author, for the further development of quad-copter as an object of the research.

Chapter 2

Literature Review

2.1 Overview of Quad-copter

2.1.1 Quad-copter

A quad-copter is a multi-rotor, a type of an Unmanned Aerial Vehicle (UAV), which is lifted and propelled by 4 motors mounted at each lever arm. In producing the thrust, usually the two set propellers rotate in opposite directions to eliminate the gyroscopic effect. The propellers are attached to each motor creating an airflow that produce down pressure for the lift force [3]. Quad-copters are popular now because they have simple mechanical and as an unmanned vehicle, quad-copter provide a good flight.

FIGURE 2.1: Most Common Configuration of Quad-copter [4]

A quad-copter has capability to do vertical take off and has maneuverability in a narrow space. The flight control of quad-copter is achieved by controlling the speed of each of the propellers, thus by controlling the speed of the motor independently[5, 6]. Consequently, by varying the speed of the motor the quad-copter can produce motion like a typical flying body such as rolling, pitching, and yawing.

2.1.2 Quad-copter's Propulsion System

Unlike other flying bodies with aerodynamic shapes, for a quad-copter, the propulsion system is the most important system that determines its performances[3]; the

FIGURE 2.2: DJI Mavic 2 [7]

important performances of a quad-copter are: speed, maximum take-off weight (MTOW) and flying endurance. In order to work properly, all of the components of the propulsion system should be compatible with each other. The propulsion system of a quad-copter typically consists of propellers, motors, electronic speed control (ESC), and battery.

2.2 Equation of Motions

The motions of a quad-copter are based on three translational subsystems (x, y, z) and three rotational subsystems (ϕ, θ, ψ). In the other words, a quad-copter can achieve 6 degrees of freedom around the axes. A mathematical model which relates to the quad-copter motions and responses to analyse the system dynamics and design the proper controller is needed. In order to model the quad-copter dynamics and stability, the equation of motions of a quad-copter has to be defined [8].

For the proper analysis the body of quad-copter is assumed as a rigid body as shown in figure 2.3. In physics a rigid body is a magnification of a solid body where the deformation in the system is neglected. Therefore the distance between any two given points of a rigid body remains constant in time regardless of external forces exerted on it. The idea of assuming the quad-copter as a rigid body means that any points on the vehicle has fixed location relatives to the center of gravity [9]. The equations of motion of a system of rigid bodies can be found by utilizing the Newton-Euler formulation [10] [11].

The free body diagram above shows the normal condition of a quad-copter body frame (x_b, y_b, z_b) with respect to the inertial frame. The x_b axis is pointed the front and positive flight direction of the quad-copter. While the z_b axis has a positive value when points down according to its normal condition as can be seen in the figure 2.3.

2.2.1 Coordinate Frame

Right Hand Rule

The right hand rule is used to show the positive direction of axes in 3D space. In order to define the coordinate frame, the right hand rule has to be introduced. The

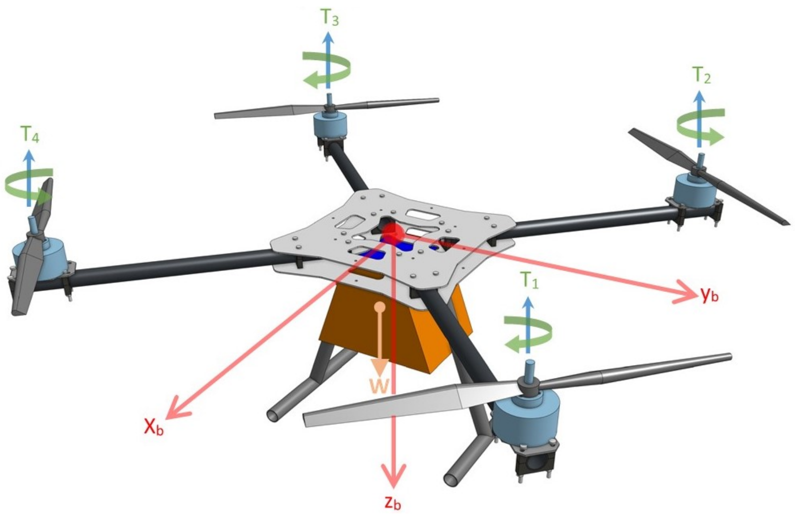

FIGURE 2.3: Quad-copter Free Body Diagram

thumb, index-finger, and middle-finger respectively pointing the direction and also the positive rotation of x, y, z as coordinate axes. [3]

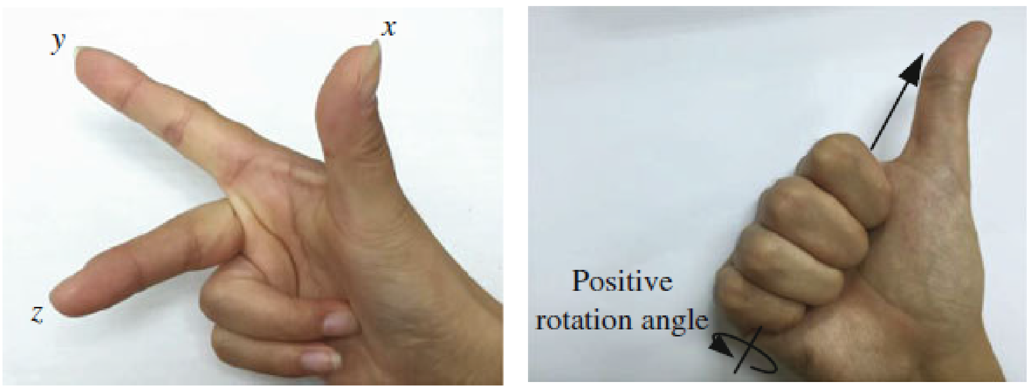

FIGURE 2.4: Right Hand Rule and The Positive Direction [3]

Earth Fixed Coordinate Frame and Body Coordinate Frame

Studying multi-rotor's dynamic with earth fixed coordinate frame (EFCF) is used to determine three dimensional position relatives to the earth's surface [12]. In this case the earth curvature is neglected and the earth's surface is assumed to be flat. It is necessary to place the initial position of a quad-copter as the coordinate origin O_e. The O_eX_e axis points to a certain direction in horizontal plan, O_eZ_e axis points perpendicularly to the ground, and the O_eY_e axis is determined according to right hand rule.

The body coordinate frame is fixed to the multi-rotor. The coordinate origin O_b of this frame is located at the Centre of Gravity (COG) of the multi-rotor. The O_bX_b

axis points to the nose. The O_bZ_b axis points downward. And the O_bY_b is determined according to the right hand rule.

Quad-copter's Motion in The Body Frame System

In deriving the equation of motion of the quad-copter, we assume the motion is of a flying rigid body. As Newton's law of motion is only directly applicable to inertial frame, but for practicality – vehicle's sensors are attached to the body – it's much more useful to use the body frame, hence in investigating the motion we must perform transformation from the inertial system[13] (e.g. the ground) to the body frame system [14].

2.2.2 Attitude Representation

Euler Angle

To represent the orientation by the set of 3 sequential rotations about specific axes in a reference frame, Euler angles are being used. The Euler Angles is based on Euler's theorem, the rotation of a rigid body around one fixed point can be regarded as the composition of several finite rotations around the fixed point.

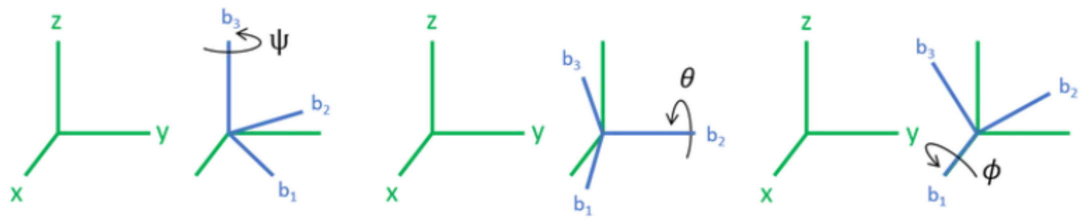

FIGURE 2.5: Euler Angles Projection [15]

From the figure above, the Euler Angles can be defined, from the left to the right, as follows:

- Yaw angle (ψ) is an angle between the projection of b_1 axis about the X reference axes and the projection of b_2 axis about the reference line of Y. It shows positive value when the b_3 as a pivot axes rotates to the left.
- Pitch angle (θ) is an angle between the projection of b_1 axis about the X reference axes and the projection of b_3 axis about the reference line of Z. It shows positive value when the b_2 as a pivot axes rotates to the left.
- Roll angle (ϕ) is an angle between the projection of b_2 axis about the Y reference axes and the projection of b_3 axis about the reference line of Z. It shows positive value when the b_1 as a pivot axes rotates to the left.

2.2.3 The Chain Rule

As previously stated, in studying the motion of the quad-copter, two frame systems are used, the body frame and the inertial system. Velocities of quad-copter are assumed to be measured by sensors implanted in the body frame so obviously the measured velocities and accelerations would be in body frame not in inertial frame. Since applying the Newtonian mechanics must be in inertial frame and the body frame is a rotating frame, so in transforming the measured velocity and accelerations to inertial system, the effect of Coriolis acceleration must be considered [16].

So the equation in body frame is expressed as,

$$\mathbf{V}^b = \begin{bmatrix} u \\ v \\ w \end{bmatrix}^b = u\hat{b}_1 + v\hat{b}_2 + w\hat{b}_3 \tag{2.1}$$

By including the Coriolis effect then the inertial velocity is written as,

$$(\frac{d\mathbf{v}}{dt})_{\text{inertial}} = (\frac{du}{dt}\hat{b}_1 + \frac{dv}{dt}\hat{b}_2 + \frac{dw}{dt}\hat{b}_3) + (u\frac{d\hat{b}_1}{dt} + v\frac{d\hat{b}_2}{dt} + w\frac{d\hat{b}_3}{dt}) \tag{2.2}$$

From observation, it can be seen that the first terms of the equation are acceleration of the quad-copter in body frame, while the second terms are effects of the fact that the body frame is a rotating or Coriolis acceleration. The equation above can be simplified as,

$$\dot{\mathbf{v}}_{\text{inertial}} = \dot{\mathbf{v}}^b + \boldsymbol{\omega}_n^b \times \mathbf{v}^b \tag{2.3}$$

2.2.4 Newton Second Law

Newton's second law of motion states that the acceleration of an object is dependent upon two variables: the total force acting on the object and the mass of the object. Newton's second law of motion related to the behaviour of objects for which all existing forces are imbalanced. When the force exerted by an object is increased, its acceleration is also increased. It is different with the mass of an object while its increased, the acceleration of the object is decreased.

In flight dynamics a more accurate statement is needed to accomplish the objectives. First, the acceleration should be an acceleration that respect to the inertial frame. Second, this law need to be stated in vector form [13].

$$\mathbf{F} = \frac{d\mathbf{p}}{dt} = m\frac{d\mathbf{v}}{dt} = m\mathbf{a} \tag{2.4}$$

The equation above governs the translational motion, while for rotational motion, it can directly obtained by taking the cross product of the equation above with vector position from some origin in inertial frame to get the torque is equal with rate-of-change of angular momentum.

2.2.5 Force, Mass, and Acceleration

Translational

The force acting on the translational axis is defined as:

$$F = m\frac{d\mathbf{v}}{dt} \tag{2.5}$$

The velocity in body frame can be written in the form

$$\mathbf{v}^b = [u, v, w]^T \tag{2.6}$$

By employing the transport theorem (here it's simply only Coriolis part), we can obtain that inertial velocity and then substituted to Eq. 2.5 to get,

$$\mathbf{F} = m(\dot{\mathbf{v}}^b + \boldsymbol{\omega}_n^b \times \mathbf{v}^b) \tag{2.7}$$

Rotational

The force acting on the rotational axis is defined as:

$$M = \frac{d\mathbf{H}}{dt} \tag{2.8}$$

2.2.6 Forces and Moments from the Propulsion

The eternal forces and moments that act on the quad-copter body should be considered for the analysis purpose. So in this case, the propeller thrust and moment as well as the gravity has to be noticed.

When the motor is turned-on the propellers rotates, which under fixed setting angle, and the thrust is produced perpendicularly to the center of gravity of the quad-copter.

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -F_1 - F_2 - F_3 - F_4 \end{bmatrix} \tag{2.9}$$

The torques represented as L, M, N are about b_1, b_2, b_3 axes in body frame. These torques are produced by the thrust with respect to center of gravity. By knowing the distances $(d_{1y}, d_{2y}, d_{3y}, d_{4y})$ of the thrusts to c.g, the torques for roll, pitch, and yaw, can be computed as,

$$L = F_1 d_{1y} - F_2 d_{2y} - F_3 d_{3y} + F_4 d_{4y} \tag{2.10}$$

$$M = -F_1 d_{1x} + F_2 d_{2x} - F_3 d_{3x} + F_4 d_{4x} \tag{2.11}$$

$$N = -T(F_1, d_{1x}, d_{1y}) - T(F_2, d_{2x}, d_{2y}) + T(F_3, d_{3x}, d_{3y}) + T(F_4, d_{4x}, d_{4y}) \tag{2.12}$$

2.2.7 Gravity Force in Body Frame

In addition with thrust forces, the gravity force is also must be transformed into body frame. So the gravity force which typically written in inertial frame as

$$F^n_{grav} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \tag{2.13}$$

Where g is a constant and the value of $g = 9.81m/s^2$. This force can be transformed into body coordinate system using, the rotational matrix from inertial coordinate system into body coordinate system, $\mathbf{C}^b_n$, to obtain:

$$\mathbf{F}^b_{grav} = \mathbf{C}^b_n \mathbf{F}^n_{grav} = \begin{bmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ -c_\theta s_\psi + s_\phi s_\theta c_\psi & c_\phi c_\psi + s_\phi s_\theta s_\psi & s_\phi c_\theta \\ s_\phi s_\psi + c_\phi s_\theta c_\psi & -s_\phi c_\psi + c_\phi s_\theta s_\psi & c_\phi c_\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \tag{2.14}$$

Which can then be written in the form:

$$\mathbf{F}^b_{grav} = \begin{bmatrix} -mg\sin(\theta) \\ mg\sin(\phi)\cos(\theta) \\ mg\cos(\phi)\cos(\theta) \end{bmatrix} \tag{2.15}$$

2.2.8 Mass & Moment of Inertia

For a rigid body, in the shape of point mass, the moment of inertia can be written as:

$$I = mr^2 \tag{2.16}$$

Moment of Inertia Matrix (I)

For more complex shape, the equation 2.16 is not applicable and the more general, the integral form, can be used instead,

$$\mathbf{I} = \int_{body} \mathbf{r}^2 dm \tag{2.17}$$

This equation can be rewritten in vector form to get,

$$\mathbf{I} = \int_{body} \mathbf{r}^2 dm = \int_{body} [\mathbf{r} \times \mathbf{r} \times dm] \tag{2.18}$$

which then by rewriting the component of the vectors, we can have

$$\mathbf{I} = \int_{body} \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} dm \tag{2.19}$$

$$\mathbf{I} = \begin{bmatrix} \int (y^2+z^2)dm & -\int (xy)dm & -\int (xz)dm \\ -\int (xy)dm & \int (x^2+z^2)dm & -\int (yz)dm \\ -\int (xz)dm & -\int (yz)dm & \int (x^2+y^2)dm \end{bmatrix} \tag{2.20}$$

$$\mathbf{I} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} \tag{2.21}$$

In this thesis, the computation of the mass and moment inertia are done by the CAD software after the quad-copter model is built and suitable material is selected.

2.2.9 Accelerations

Now, we can to use newton's law of motion in rotating frame to derive the equation of motion. As the motion of the quad-copter consists of translational and rotational motion, the derivation can be conducted as follows.

Translational Acceleration

The acceleration in translational mode as has been obtained previously, can be written as follows. The major difference here compared with the inertial frame is the second terms that represents the Coriolis acceleration.

$$\dot{\mathbf{v}}_{\text{inertial}} = \dot{\mathbf{v}}^b + \boldsymbol{\omega}_n^b \times \mathbf{v}^b \tag{2.22}$$

$$\dot{\mathbf{v}}_{\text{inertial}} = \dot{\mathbf{v}}^b + \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \mathbf{v} \tag{2.23}$$

By inserting the velocity in the body frame $\mathbf{v}^b$, where the rotational velocity is given by $\boldsymbol{\omega}^b$, we can get,

$$\dot{\mathbf{v}}_{\text{inertial}} = \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix}^b + \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}^b = \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix} \tag{2.24}$$

Translational Motion

Since,

$$\Sigma \mathbf{F}_b = m\mathbf{a}_b \tag{2.25}$$

and considering all the forces, thrusts from the propellers, and the weight, are already properly transformed into body frame, and the velocity is also received the same transformation, we can get,

$$\mathbf{F}_{prop} + \mathbf{F}_{grav} = m\dot{\mathbf{v}}^b_{inertial} \tag{2.26}$$

In matrix form we can rewrite it as,

$$\begin{bmatrix} -mg\sin(\theta) \\ mg\sin(\phi)\cos(\theta) \\ -F_1 - F_2 - F_3 - F_4 + mg\cos(\phi)\cos(\theta) \end{bmatrix} = m \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix} \tag{2.27}$$

Angular Acceleration and Rotational Motion Likewise, in rotational motion, the same process must be carried. Since we'll be using the body frame, then all of the torques, and angular acceleration must be transformed into body frame whenever necessary. Since in this thesis the gyroscopic torques aren't considered, and the gravity force obviously does not incur any torque, the torques are only came from the propellers. Since the propellers are fixed, and the quad-copter is assumed to be rigid body, then for torques, transformation becomes unnecessary. We only need to consider the effect of Coriolis in angular acceleration as shown below,

$$\mathbf{M} = \mathbf{I}^b\dot{\boldsymbol{\omega}}^b_n + \boldsymbol{\omega}^b_n \times \mathbf{I}^b\boldsymbol{\omega}^b_n \tag{2.28}$$

The matrix form of the equation above is,

$$\begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \tag{2.29}$$

$$\begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2.30}$$

Due to symmetric shape of the quad-copter with respect to x and y of body axes, then the products of inertia are negligible, the above equation can be simplified as,

$$\begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2.31}$$

which then can be further simplified as,

$$\begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} I_{xx}\dot{p} \\ I_{yy}\dot{q} \\ I_{zz}\dot{r} \end{bmatrix} + \begin{bmatrix} -I_{yy}qr + I_{zz}qr \\ I_{xx}pr - I_{zz}pr \\ -I_{xx}pq + I_{yy}pq \end{bmatrix} \tag{2.32}$$

2.2.10 Additional Coordinate Frame Transformation

The following formula states the relationship between the Euler angle rates with the angle rate measured directly by gyro sensors of the quad-copter.

$$p(\hat{b}_1) + q(\hat{b}_2) + r(\hat{b}_3) = \dot{\psi}(\hat{b}_3'') + \dot{\theta}(\hat{b}_2') + \dot{\phi}(\hat{b}_1) \tag{2.33}$$

Which in the form of matrix, it can be rewritten in the component of roll, pitch, and yaw rate, are given by

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} -\dot{\psi} s_\theta \\ \dot{\psi} c_\theta s_\phi \\ \dot{\psi} c_\phi c_\theta \end{bmatrix} + \begin{bmatrix} 0 \\ \dot{\theta} c_\phi \\ -\dot{\theta} s_\phi \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} \tag{2.34}$$

So the set of equations can be defined as:

$$\mathbf{p} = \dot{\phi} - \dot{\psi} \sin\theta \tag{2.35}$$

$$\mathbf{q} = \dot{\theta} \cos\phi + \dot{\psi} \cos\theta \sin\phi \tag{2.36}$$

$$\mathbf{r} = \dot{\psi} \cos\phi \cos\theta - \dot{\theta} \sin\phi \tag{2.37}$$

Rewritten the equations for the Euler angle derivatives:

$$\dot{\phi} = p + (q \sin\phi + r \cos\phi) \tan\theta \tag{2.38}$$

$$\dot{\theta} = q \cos\phi - r \sin\phi \tag{2.39}$$

$$\dot{\psi} = (q \sin\phi + r \cos\phi) \sec\theta \tag{2.40}$$

In order to track the displacement of the quad-copter, we need to define the inertial frame or the navigation coordinate system. One way to implement this is to simply transform the velocity in the body frame into inertial frame. In doing so, we can get,

$$\begin{bmatrix} \dot{x}^E \\ \dot{y}^E \\ -\dot{h}^E \end{bmatrix} = \mathbf{C}_b^n \begin{bmatrix} u \\ v \\ w \end{bmatrix}^b = \begin{bmatrix} c_\theta c_\psi & -c_\theta s_\psi + s_\phi s_\theta c_\psi & s_\phi s_\psi + c_\phi s_\theta c_\psi \\ c_\theta s_\psi & c_\phi c_\psi + s_\phi s_\theta s_\psi & -s_\phi c_\psi + c_\phi s_\theta s_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}^b \tag{2.41}$$

To be more descriptive, the z-axis of the inertial frame is taken as $-h^E$.

2.2.11 The Quad-copter Equations of Motion

Finally, after all the necessary equation has been derived, we will get the set of equations of motion that govern the dynamic of a quad-copter.

$$\dot{u} = -g\sin(\theta) + rv - qw \tag{2.42}$$

$$\dot{v} = g\sin(\phi)\cos(\theta) - ru + pw \tag{2.43}$$

$$\dot{w} = \frac{1}{m}(-F_z) + g\cos(\phi)\cos(\theta) + qu - pv \tag{2.44}$$

$$\dot{p} = \frac{1}{I_{xx}}(L + (I_{yy} - I_{zz})qr) \tag{2.45}$$

$$\dot{q} = \frac{1}{I_{yy}}(M + (I_{zz} - I_{xx})pr) \tag{2.46}$$

$$\dot{r} = \frac{1}{I_{zz}}(N + (I_{xx} - I_{yy})pq) \tag{2.47}$$

$$\dot{\phi} = p + (q\sin\phi + r\cos\phi)\tan\theta \tag{2.48}$$

$$\dot{\theta} = q\cos\phi - r\sin\phi \tag{2.49}$$

$$\dot{\psi} = (q\sin\phi + r\cos\phi)\sec\theta \tag{2.50}$$

$$\dot{x}^E = c_\theta c_\psi u^b + (-c_\phi s_\psi + s_\phi s_\theta c_\psi)v^b + (s_\phi s_\psi + c_\phi s_\theta c_\psi)w^b \tag{2.51}$$

$$\dot{y}^E = c_\theta s_\psi u^b + (c_\phi c_\psi + s_\phi s_\theta s_\psi)v^b + (-s_\phi c_\psi + c_\phi s_\theta s_\psi)w^b \tag{2.52}$$

$$\dot{h}^E = -1 * (-s_\theta u^b + s_\phi c_\theta v^b + c_\phi c_\theta w^b) \tag{2.53}$$

where

$$F_z = F_1 + F_2 + F_3 + F_4 \tag{2.54}$$

$$L = F_1 d_{1y} - F_2 d_{2y} - F_3 d_{3y} + F_4 d_{4y} \tag{2.55}$$

$$M = F_1 d_{1x} - F_2 d_{2x} + F_3 d_{3x} - F_4 d_{4x} \tag{2.56}$$

$$N = -T(F_1, d_{1x}, d_{1y}) - T(F_2, d_{2x}, d_{2y}) + T(F_3, d_{3x}, d_{3y}) + T(F_4, d_{4x}, d_{4y}) \tag{2.57}$$

and h^E is positive in the up direction.

Chapter 3

Research Methodology

This thesis is mainly aimed to propose a baseline design and provide numerical analysis of its basic performances that can be used for future prototyping for an agricultural quad-copter. At first a scaled down CAD model is established and then based on the model, numerical simulation was performed to determine suitable propulsion system. Selecting suitable propulsion system is usually conducted in a trial and error process. The numerical simulation will provide guidance so the trial-and-error process can be performed faster and efficiently.

3.1 Designing Quad-copter Model in CAD

The design of quad-copter structure should be modelled for this experiment. This quad-copter model is developed by using CAD drawing. The model design for the quad-copter can be seen in figure 3.1.

FIGURE 3.1: Quad-copter frame design using CAD software

In purpose for this research, Onshape is used to design the physical model of quad-copter. Onshape is an open source CAD software, delivered over the internet via a Server as a Service(SAAS) model. The design of the quad-copter is needed to give a better picture of the quad-copter dimension, size, and its specific component that affecting the motion of quad-copter. With the design, the software can performs a calculation for the initial mass properties of the quad-copter (quad-copter mass,

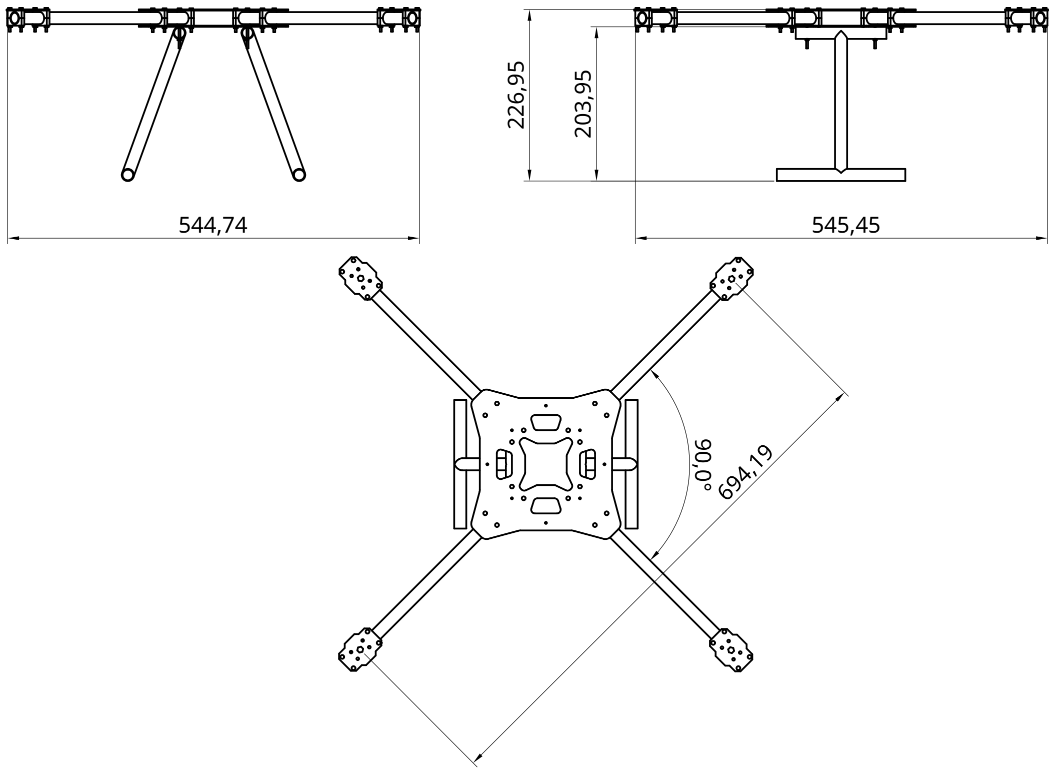

FIGURE 3.2: Quad-copter Design in 3 View with Scale 1:16

moment of inertia, volume, center of mass). The result of the calculation performed by the software give an advantage of information to predict and evaluate the quad-copter design for the later simulation.

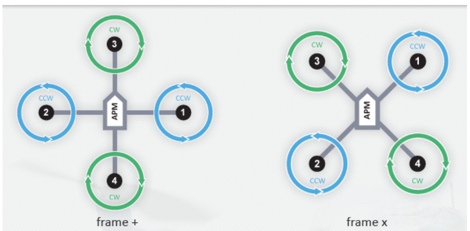

FIGURE 3.3: Quad-copter Propeller Configuration [3]

The design of the quad-copter use the standard X propeller configuration instead of + configuration because of beneficial value compared with + configuration, X configuration have higher maneuverability. The water tank has been attached below the body of the frame, between the landing skid, and 4 propellers are attached to each motors at the edge of its arms. The quad-copter is designed to carry the maximum payload of 1 kg liquid in the water tank.

3.2 Selecting The Propulsion System

3.2.1 The Motors

The propulsion system of the quad-copter design should be modelled after the market available brush-less motors. Since one of the objective of this research is to determine the proper propulsion system as a baseline configuration, the availability of the motors in the market has to be researched and gathered. These are several motors that proper for the design in the table given:

No	Motor Name	Mass (kg)	RPM	Max Power Consumption (Watt)
1	MT-4006-13 740KV	0,092	740	500
2	Sunnysky X2212-13 980KV	0,056	980	150
4	AX-2810Q-750KV	0,07	750	444
5	EMAX RS1108	0,0082	4500	211,2
7	F60 PRO III Motor - 1750KV	0,0351	1750	950

These motors will be modelled after CAD and assembled with the quad-copter design. The purpose to model the motors in CAD design is to calculate the mass properties of the whole quad-copter design. Since we considered the motors have different in size and mass then the moment of inertia of the quad-copter will also be different for each motor.

FIGURE 3.4: A Motor Sample of MT-4006-13 740kv [17]

3.2.2 The Propeller

The propellers used in each motor have been designed with CAD software as shown in the figure 3.5. In order to gain the minimum required thrust the propeller is designed to have a radius of 0.102 m and pitch angle of 0.1016 m. The chord length of propeller is also important. Usually, the chord is located at the 2/3 of the propeller radius length. The design uses carbon fibre as its material. The performance of propeller is shown by the amount of thrust it can produced for a certain motor's speed.

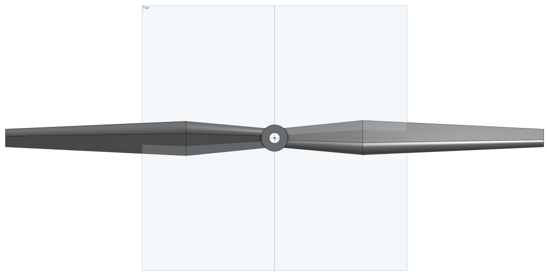

FIGURE 3.5: The Propeller Design From CAD

3.3 Calculate The Mass Properties

After the quad-copter and the propulsion system have been modelled, we can calculate the mass properties of the whole design. Onshape CAD software can perform the calculation automatically based on the design we've been built and assembled. As a consideration, the mass properties is generated for each different payload.

The payload difference gives an impact for the quad-copter mass and performance. The variation of the mass change will be distinguished in a percentage count from full capacity payload to 0 capacity due to the liquid in the water tank. The amount of liquid carried by the quad-copter is assumed to decreased per 25% of its maximum capacity.

3.4 Generate The Motion Numerical Simulation Code

The motion numerical simulation will be generated by using Python programming code. The code represents the mathematical model of quad-copter motion, moment, and force acting on it in 3-Dimensional space. Most of the equations are the derivative from Newton second law in translational and rotational coordinates. All the variables and parameters are defined orderly to model the dynamic model of the quad-copter.

Python is a high level programming language used to approach the problem solution by logical code. From quad-copter mathematical model, the codes are developed for the numerical simulation of the quad-copter motion, moments, and force. The numerical simulation is used as an approach for proper analysis of the problem occur in quad-copter when the continuous mass discharged applied during the flight condition.

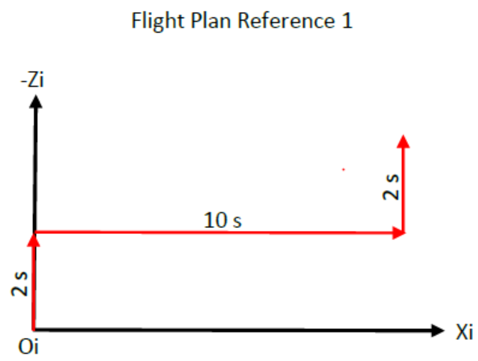

FIGURE 3.6: Desired Flight Profile

This simulation will also generate the flight profile of the quad-copter and this flight profile expected to have a pattern similar with the baseline for the flight profile given in the figure above, figure 3.6, as a typical agricultural drone would function in rice paddy watering. Since the control input for the simulation use feed forward control method, the expected flight plan is measured based on the input command control in time.

3.5 Control Input

The quad-copter is given an input command control to project its movement in the simulation. The code created in the simulation is adopting a feed forward control input, which means there is no feedback for the control. The control command given in simulation to the quad-copter is only 1 way through without correction of error. Later in the simulation, the command control input of the quad-copter is generated as a control profile plot.

3.6 Analyse The Result of The Numerical Simulation

The numerical simulation analysis will based on the performance assessment of the quad-copter design. But the design of the quad-copter and its propulsion system should be able to follow the desired flight plan in fig 3.6

3.7 Switch the Motor

If the quad-copter cannot follow the desired flight plan, then we have to switch the propulsion system (motor) for the quad-copter design. The change of the propulsion performed in sequence after the simulation running and show the result.

Chapter 4

Results and Discussions

This chapter will provide the result from the Python numerical simulation of the research, the findings of the research and also discussion of the research. The result of the research found from the plot and data simulation performed due to mass variation. The findings of this research found from the observation and analysis of the data and plot of the quad-copter motion. The discussion of this research is based on the obtained results pertained theories used in this research. All of those will be presented on the following.

4.1 Results

This research use a numerical simulation by using python. The codes in numerical simulation are generated under the circumstance of the quad-copter CAD design. There are certain variables use, based on the quad-copter CAD design.

4.1.1 Simulation Result

Several rules have been considered in creating numerical simulation code about the quad-copter motion. For some of these rules are really affecting the performance of the quad-copter. The simulation will be very useful for the later analysis.

- The simulation varies the volume of the water attached on the quad-copter. The variation scheme that put into the numerical simulation code begins when the water tank has a full capacity of water, then it is reduced for each quarter from the maximum capacity of the water tank. So the mass input to the code is the total drone mass plus the percentage of volume of the water tank.

- The propeller size and dimension consider to be the fix variables for the code. Its size is fixed, since this part is the main component that will produce the thrust to overcome the weight of the quad-copter. The performance quality of the propulsion system depends on its propeller.

- The feasibility of the configuration is assessed based on the ability of the motor to provide the amount of thrust to follow the flight plan within the acceptable angular velocity of the motor. With a certain control input to command the quad-copter movement, the result of the simulation creates a flight profile as shown in the figure 4.4.

4.1.2 Motor Properties

The motor properties are gained by the Onshape CAD software. The motor attached on the design will define the moment of inertia in the whole design due to the mass variant consideration. These data of mass properties will be used to set the quad-copter initial condition in the simulation. There are 5 suitable motors choose for the quad-copter design.

MT-4006-13 740KV							
Water Capacity	**Drone Mass (kg)**	**Centre of Mass (mm)**			**Moment of Inertia (kgmm^2)**		
		x	**y**	**z**	**Ixx**	**Iyy**	**Izz**
100%	2.15758987	-0.00850	0.00398	-34.32524	4,59E+09	4,59E+09	7,87E+09
75%	1.89348554	-0.00742	0.00592	-33.66847	4,56E+09	4,57E+09	7,84E+09
50%	1.64362485	-0.00855	0.00682	-27.87867	4,51E+09	4,51E+09	7,80E+09
25%	1.39345537	-0.01009	0.00804	-15.95613	4,36E+09	4,35E+09	7,75E+09
0%	1.19781007	-0,01173	0.00935	-0.12266	4,13E+09	4,11E+09	7,71E+09

Sunnysky X2212-13 980KV							
Water Capacity	**Drone Mass (kg)**	**Centre of Mass (mm)**			**Moment of Inertia (kgmm^2)**		
		x	**y**	**z**	**Ixx**	**Iyy**	**Izz**
100%	2.01691751	-0.00726	-0.00277	-37.66269	3,71E+09	3,71E+09	6,18E+09
75%	1.75281317	-0.00592	-0.00170	-37.45608	3,68E+09	3,69E+09	6,14E+09
50%	1.50295248	-0.00690	-0.00198	-31.75405	3,63E+09	3,64E+09	6,10E+09
25%	1.25278300	-0.00828	-0.00237	-19.26663	3,49E+09	3,49E+09	6,06E+09
0%	1.05713770	-0.00981	-0.00281	-1.93889	3,27E+09	3,26E+09	6,01E+09

AX-2810							
Water Capacity	**Drone Mass (kg)**	**Centre of Mass (mm)**			**Moment of Inertia (kgmm^2)**		
		x	**y**	**z**	**Ixx**	**Iyy**	**Izz**
100%	2.06966964	-0.00774	-0.00013	-36.35798	4,04E+09	4,04E+09	6,81E+09
75%	1.80556531	-0.00651	0.00130	-35.96656	4,01E+09	4,02E+09	6,78E+09
50%	1.55570462	-0.00755	0.00151	-30.21865	3,96E+09	3,97E+09	6,74E+09
25%	1.30553514	-0.00900	0.00180	-17.94159	3,82E+09	3,82E+09	6,69E+09
0%	1.10988984	-0.01059	0.00211	-1.20386	3,59E+09	3,58E+09	6,65E+09

EMAX RS1108							
Water Capacity	**Drone Mass (kg)**	**Centre of Mass (mm)**			**Moment of Inertia (kgmm^2)**		
		x	**y**	**z**	**Ixx**	**Iyy**	**Izz**
100%	2.11362976	-0.00813	0.00197	-35.32047	4,32E+09	4,31E+09	7,34E+09
75%	1.84952543	-0.00698	0.00366	-34.79020	4,29E+09	4,29E+09	7,31E+09
50%	1.59966473	-0.00807	0.00423	-29.01651	4,23E+09	4,24E+09	7,27E+09
25%	1.34949526	-0.00956	0.00502	-16.91652	4,09E+09	4,08E+09	7,22E+09
0%	1.15384996	-0.01118	0.00587	-0.64266	3,86E+09	3,84E+09	7,18E+09

F60 PRO III Motor - 1750kv							
Water Capacity	**Drone Mass (kg)**	**Centre of Mass (mm)**			**Moment of Inertia (kgmm^2)**		
		x	**y**	**z**	**Ixx**	**Iyy**	**Izz**
100%	1.92899728	-0.00640	-0.00749	-39.99578	3,15E+09	3,16E+09	5,12E+09
75%	1.66489295	-0.00485	-0.00711	-40.14836	3,12E+09	3,14E+09	5,09E+09
50%	1.41503225	-0.00570	-0.00836	-34.56744	3,08E+09	3,09E+09	5,05E+09
25%	1.16486277	-0.00693	-0.01016	-21.74172	2,95E+09	2,96E+09	5,00E+09
0%	0.96921747 kg	-0.00833	-0.01221	-3.34177	2,74E+09	2,73E+09	4,95E+09

The result from the simulation will determine the most proper motor for the baseline design of quad-copter design in this research.

4.1.3 The Time History and Control Input

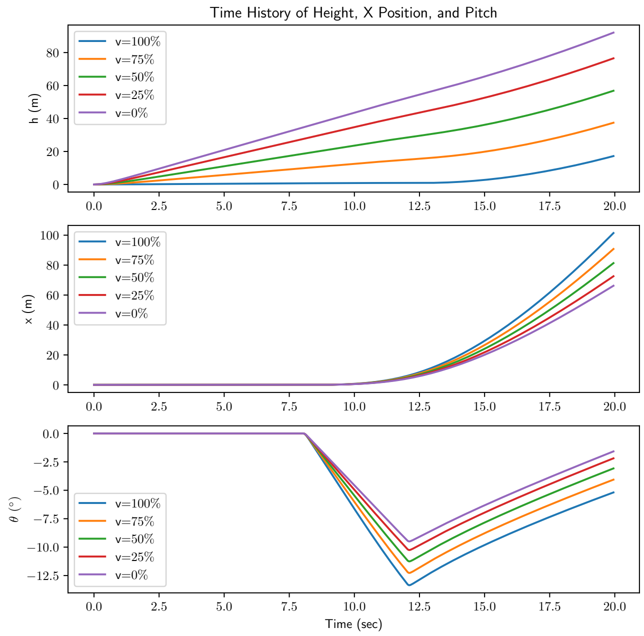

FIGURE 4.1: The Time History

Figure 4.1 shows the plot of the relationship between time, position and orientation of the quad-copter. The plot is generated from the Python numerical simulation that had been built for this research.

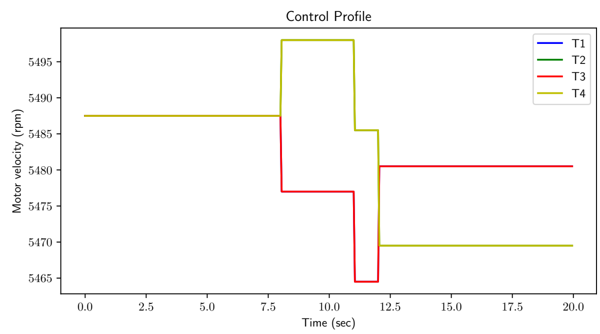

FIGURE 4.2: The Control Profile

Figure 4.2 shows the control profile of control command input for the 4 motors of the quad-copter in the simulation. The RPM of the motors is trimmed to be 4600 RPM for this simulation. To gain a desired position

4.1.4 Simulation Result for Different Volume of Payload

The numerical simulation takes 5 different payloads for the input. These variation of mass consequently gives impact to the performance of the quad-copter thrust as can be shown in the figure 4.3.

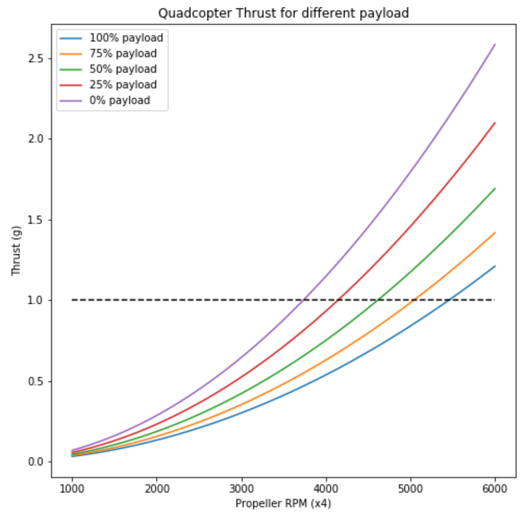

FIGURE 4.3: The Quad-copter Thrust

The payload variation cause the quad-copter to exert the different amount of force to overcome the weight and produce lift. The payload variation impacts the flight profile in simulation. The plot result is given as follow:

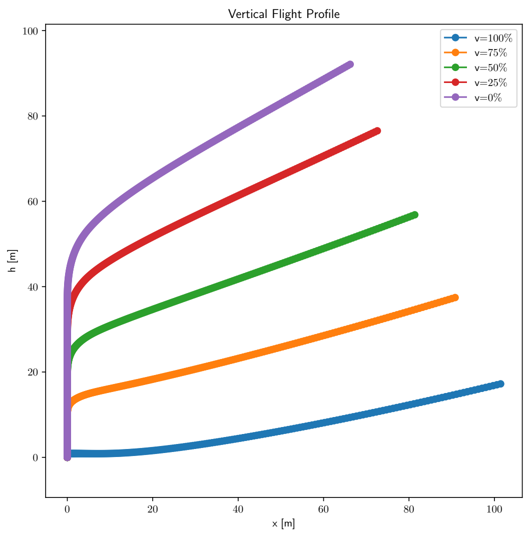

FIGURE 4.4: The Quad-copter Vertical Profile

4.2 Discussions

Based on the result of the simulation, the difference in mass obviously governs the work of propulsion system. When the quad-copter carries bigger payload, the motor only required to produce less thrust in the same RPM as the quad-copter carries less payload; this can be seen in figure4.3.

In order to follow the flight plan given, the quad-copter propulsion system has to be able to provide the required thrust. The control command input creates in the simulation gives the motors order to reach certain angular velocity/RPM to generate thrust. In order to do so the RPM in control command input is trimmed in 4600 RPM. This number is the initial value that required by the quad-copter to do a movement.

The most proper motor to be set as a baseline design configuration of the quad-copter design in this research is the EMAX RS1108. The choice of this motor is based on the motors table in 3.

Chapter 5

Conclusions and Future Works

5.1 Conclusions

Based on analysis and discussions given in Chapter 4, the author concluded that the design of the quad-copter used in the simulation can meet the objectives required as the baseline design. All of the results from the test and simulation are summarized as follows:

- Numerical based approach in determining the propulsion system has been developed.
- The quad-copter baseline design has also been designed where it has mass of 1.8 kg, with minimum RPM trim is 4600 (EMAX RS1108 is the most possible option), and it should has a minimal propeller radius of 0.102 m and pitch angle of 0.1016 m.
- The baseline design was also successfully tested numerically.
- The numerical simulation could be use as a comparison of the baseline design for the performance assessment for prototyping.

5.2 Recommendation for Future Works

The author suggests for the further improvement in creating a numerical comparison for the baseline design for the prototype, especially for the need of agricultural work.

This configuration can be used for building a prototype of quad-copter that need a consideration of varying mass system. The numerical comparison and the baseline design also can be used to design an optimal and robust of control of the drone.

The recommendation is summarized in these points:

- Comparison with flight test of the baseline design should be conducted.
- This influence of varying mass system while discharging should be studied.
- To design an optimal and robust of control of the drone.

Appendix A: Python Code

```
#!/usr/bin/env python

import numpy as np
from scipy.optimize import fsolve


# Quachopter's Frame Properties
m = 0.1                  # kg
Ixx = 0.00062             # kg-m^2
Iyy = 0.00113             # kg-m^2
Izz = 0.9 * (Ixx + Iyy)  # kg-m^2 (Assume nearly flat object, z=0)
dx = 0.114      # m
dy = 0.0825     # m
g = 9.81        # m/s/s


# Quachopter's Propeller Properties
R = 0.0762   # propeller length/ disk radius (m)
A = np.pi * R ** 2
rho = 1.225  #kg/m^3  at MSL
a = 5.7      # Lift curve slope used in example in Stevens & Lewis
b = 2        # number of blades
c = 0.0274   # mean chord length (m)
eta = 1      # propeller efficiency

# Manufacturer propeller length x pitch specification:
p_diameter = 6  #inches
p_pitch = 3   #inches
theta0 = 2 * np.arctan2(p_pitch, (2 * np.pi * 3/4 * p_diameter/2))
theta1 = -4 / 3 * np.arctan2(p_pitch, 2 * np.pi * 3/4 * p_diameter/2)



def control_command(t):
    # Inputs: Current state x[k], time t
```

```
# Returns: Control inputs u[k]

#### Placeholder Function ####

# Trim RPM for all 4 propellers to provide thrust for a level hover
trim = 3200

pitch_cmd = 0
roll_cmd = 0
climb_cmd = 0
yaw_cmd = 0

# Example open loop control inputs to test dynamics:
#  Climb
if t < 11.0:
    climb_cmd = 500

#  Pitch Forward
if t > 8.0:
    pitch_cmd = -10
if t > 9.0:
    pitch_cmd = 10
if t > 10.0:
    pitch_cmd = 0

#  Pitch Backward
if t > 12.0:
    pitch_cmd = 15
if t > 13.0:
    pitch_cmd = -15
if t > 14.0:
    pitch_cmd = 0

#  Increase lift
if t > 16.0:
    climb_cmd = 150


# RPM command based on pitch, roll, climb, yaw commands
u = np.zeros(4)
u[0] = trim + ( pitch_cmd + roll_cmd + climb_cmd - yaw_cmd) / 4
u[1] = trim + (-pitch_cmd - roll_cmd + climb_cmd - yaw_cmd) / 4
```

```
    u[2] = trim + ( pitch_cmd - roll_cmd + climb_cmd + yaw_cmd) / 4
    u[3] = trim + (-pitch_cmd + roll_cmd + climb_cmd + yaw_cmd) / 4


    return u


def control_profile(control_command, ts):
    """
    Returns control profile for quadchopter.
    Keyword Arguments:
    control_command -- Function, feedforward control based control
    ts              -- Time span
    """
    controls = np.array([])
    for t in ts:
        control = control_command(t)
        controls = np.append(controls, control)

    controls = controls.reshape((-1, 4), order='C')

    return controls


# Propeller Thrust equations as a function of propeller induced velocity, vi
def v_induced_residu(vi, *prop_params):

    # Unpack parameters
    R, A, rho, a, b, c, eta, theta0, theta1, U, V, W, Omega = prop_params

    # Calculate local airflow velocity at propeller with vi, V'
    Vprime = np.sqrt(U**2 + V**2 + (W - vi)**2)

    # Calculate Thrust averaged over one revolution of propeller using vi
    Thrust = 1/4 * rho * a * b * c * R * \
        ( (W - vi) * Omega * R + 2/3 * (Omega * R)**2 * (theta0 + 3/4 * theta1) + \
          (U**2 + V**2) * (theta0 + 1/2 * theta1) )

    # Calculate residual for equation: Thrust = mass flow rate * delta Velocity
    residual = eta * 2 * vi * rho * A * Vprime - Thrust

    return residual
```

```python
def Fthrust(x, u, dx, dy):
    """
    Returns the magnitude of thrust of a propeller.
    Keyword Arguments:
    x  -- state space [u, v, w, p, q, r, phi, theta, psi, xe, ye, he]
    u  -- angular velocity of the propeller (rpm)
    dx -- distance of the propeller to c.g. in x axis
    dy -- distance of the propeller to c.g. in y axis
    """

    # Local velocity at propeller from vehicle state information
    ub, vb, wb = x[0], x[1], x[2]
    p, q, r = x[3], x[4], x[5]

    # Transofrm velocity to local propeller location:
    #      [U,V,W] = [ub,vb,wb] + [p,q,r] x [dx,dy,0]
    U = ub - r * dy
    V = vb + r * dx
    W = wb - q * dx + p * dy

    # Convert commanded RPM to rad/s
    Omega = 2 * np.pi / 60 * u

    #Collect propeller config, state, and input parameters
    prop_params = (R, A, rho, a, b, c, eta, theta0, theta1, U, V, W, Omega)

    # Numerically solve for propeller induced velocity, vi
    # using nonlinear root finder, fsolve, and prop_params
    vi0 = 0.1     # initial guess for vi
    vi = fsolve(v_induced_residu, vi0, args=prop_params)

    # Plug vi back into Thrust equation to solve for T
    Vprime = np.sqrt(U**2 + V**2 + (W - vi)**2)
    Thrust = eta * 2 * vi * rho * A * Vprime

    return Thrust


def gyro_torque(T, dx, dy):
    """
    Returns gyro torque experience by the quadchopter
```

```
    Note: currently is not implemented

    Keyword Arguments:
    T  -- thrust produced by the motor
    dx -- geometric arm of the thrust in x-axis
    dy -- geometric arm of the thrust in x-axis
    """

    return 0


# Nonlinear Dynamics Equations of Motion
def state_gradient(x, u):
    # Inputs: state vector (x), input vector (u)
    # Returns: time derivative of state vector (xdot)

    #  State Vector Reference:
    #idx  0, 1, 2, 3, 4, 5,  6,   7,   8,   9, 10, 11
    #x = [u, v, w, p, q, r, phi, the, psi, xE, yE, hE]

    # Store state variables in a readable format
    ub = x[0]
    vb = x[1]
    wb = x[2]
    p = x[3]
    q = x[4]
    r = x[5]
    phi = x[6]
    theta = x[7]
    psi = x[8]
    xE = x[9]
    yE = x[10]
    hE = x[11]

    # Calculate forces from propeller inputs (u)
    F1 = Fthrust(x, u[0],  dx,  dy)
    F2 = Fthrust(x, u[1], -dx, -dy)
    F3 = Fthrust(x, u[2],  dx, -dy)
    F4 = Fthrust(x, u[3], -dx,  dy)
    Fz = F1 + F2 + F3 + F4
    L = (F2 + F3) * dy - (F1 + F4) * dy
    M = (F1 + F3) * dx - (F2 + F4) * dx
```

```python
    N = -gyro_torque(F1,dx,dy) - gyro_torque(F2,dx,dy) +\
        gyro_torque(F3,dx,dy) + gyro_torque(F4,dx,dy)

    # Pre-calculate trig values
    cphi = np.cos(phi);   sphi = np.sin(phi)
    cthe = np.cos(theta); sthe = np.sin(theta)
    cpsi = np.cos(psi);   spsi = np.sin(psi)

    # Calculate the derivative of the state matrix using EOM
    xdot = np.zeros(12)

    xdot[0] = -g * sthe + r * vb - q * wb  # = udot
    xdot[1] = g * sphi*cthe - r * ub + p * wb # = vdot
    xdot[2] = 1/m * (-Fz) + g*cphi*cthe + q * ub - p * vb # = wdot
    xdot[3] = 1/Ixx * (L + (Iyy - Izz) * q * r)  # = pdot
    xdot[4] = 1/Iyy * (M + (Izz - Ixx) * p * r)  # = qdot
    xdot[5] = 1/Izz * (N + (Ixx - Iyy) * p * q)  # = rdot
    xdot[6] = p + (q*sphi + r*cphi) * sthe / cthe  # = phidot
    xdot[7] = q * cphi - r * sphi  # = thetadot
    xdot[8] = (q * sphi + r * cphi) / cthe  # = psidot

    xdot[9] = cthe*cpsi*ub + (-cphi*spsi + sphi*sthe*cpsi) * vb + \
        (sphi*spsi+cphi*sthe*cpsi) * wb  # = xEdot

    xdot[10] = cthe*spsi * ub + (cphi*cpsi+sphi*sthe*spsi) * vb + \
        (-sphi*cpsi+cphi*sthe*spsi) * wb # = yEdot

    xdot[11] = -1*(-sthe * ub + sphi*cthe * vb + cphi*cthe * wb) # = hEdot

    return xdot


def rk4(x_c, u, dt):
    """
    Implement numerical integration of 4th order of Runge Kutta
    for qudchopter dynamics under acceleration.

    Keyword Arguments:
    x  -- current state vector of quadchopter
    u  -- input vector
    dt -- time step of integration
    """
```

```
    # Gradient estimation
    k1 = state_gradient(x_c, u)
    k2 = state_gradient(x_c + k1 * dt / 2, u)
    k3 = state_gradient(x_c + k2 * dt / 2, u)
    k4 = state_gradient(x_c + k3 * dt, u)

    # Prediction of state
    x_n = x_c + 1/6 * (k1 + 2*k2 + 2*k3 + k4) * dt

    return x_n
```

```
#!/usr/bin/env python

import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint, solve_ivp

import quadchopter as qc


# Simulation time and model parameters
dt = 0.01                                  # Sampling time (sec)
simulation_time = 30                       # Length of time to run simulation (sec)
ts = np.arange(0, simulation_time, dt)     # time array


# Initialize State Conditions
x = np.zeros((12, np.size(ts)))  # time history of state vectors

# Initial height
x[11,0] = 0.0

# Initialize inputs
u = np.zeros((4, np.size(ts)))  # time history of input vectors

# Initial control inputs
u[:,0] = np.zeros(4)


controls = qc.control_profile(qc.control_command, ts)

# Trajectory integration
```

```python
for k in range(0, np.size(ts) - 1):

    # Determine control inputs based on current state
    u[:, k] = qc.control_command(ts[k])

    # Predict state after one time step
    x[:, k+1] = qc.rk4(x[:,k], u[:,k], dt)


states = np.transpose(x)

plt.figure(1, figsize=(8,8))
plt.subplot(311)
plt.plot(ts, states[:, 11],'b',label='h')
plt.ylabel('h (m)')
#plt.xlabel('Time (sec)')
#plt.legend(loc='best')
plt.title('Time History of Height, X Position, and Pitch')


plt.subplot(312)
plt.plot(ts, states[:, 9],'b', label='x')
plt.ylabel('x (m)')
# plt.xlabel('Time (sec)')

plt.subplot(313)
plt.plot(ts, np.degrees(states[:, 7]), 'b', label='theta')
plt.ylabel('Theta (deg)')
plt.xlabel('Time (sec)')


plt.figure(3, figsize=(8,4))
plt.plot(ts, controls[:, 0], 'b', label='T1')
plt.plot(ts, controls[:, 1], 'g', label='T2')
plt.plot(ts, controls[:, 2], 'r', label='T3')
plt.plot(ts, controls[:, 3], 'y', label='T4')
plt.xlabel('Time (sec)')
plt.ylabel('Motor velocity (rpm)')
plt.legend(loc='best')
plt.title('Control Profile')

plt.show()
```

```
# states = odeint(stateDerivative2, x0, ts)

# xout = x0

# for n in np.arange(ts.shape[0] - 1):
#     un = controls[n]
#     xn = odeint(stateDerivative2, x0, ts[n: n+2], args=(un, ))
#     x0 = xn[1]
#     xout = np.append(xout, x0)

# states = xout.reshape(-1, 12, order='C')

# 4th Order Runge Kutta Calculation
```

Bibliography

[1] Rambaldi Giacomo and Guerin David. *Unmanned aerial systems (UAS) in agriculture: regulations and good practices*. 2017.

[2] *Agriculture Quadcopter Sprayer X-25*. URL: http://www.unbten.com/product/Agriculture-Quadcopter-Sprayer-X-25.html.

[3] Quan Quan. *Introduction to Multicopter Design and Control*. 2017.

[4] *Propeller_Rotation*. URL: https://www.dronezon.com/wp-content/uploads/2017/11/How-A-Quadcopter-Flies-Using-Propeller-And-Motor-Direction-300x293.jpg.

[5] Qikai Huang and Bruce Wingo. "Mathematical Modeling of Quadcopter Dynamics". In: *Rose-Hulman Undergraduate Research Publications* (2016), p. 14.

[6] W Kuntjoro, A H M Saleh, and R E M Nasir. "Flight Thrust Performance of Quadcopter". In: 5.1 (2018).

[7] *DJI Mavic 2*. URL: https://pisces.bbystatic.com/image2/BestBuy{_}US/images/products/6262/6262620{_}sd.jpg.

[8] Andrea L'Afflitto. *A Mathematical Perspective on Flight Dynamics and Control*. 2017, pp. 35–64.

[9] Brian L. Stevens, Frank L. Lewis, and Eric N. Johnson. *Aircraft Control and Simulation, Dynamics, Controls Design, and Autonomous Systems*. Third. Wileys, 2001.

[10] Eswarmurthi Gopalakrishnan. "QUADCOPTER FLIGHT MECHANICS MODEL AND CONTROL ALGORITHMS". PhD thesis. Czech Technical University, 2014, p. 67.

[11] Vicente Martinez Martinez Z. "Modelling of the Flight Dynamics of a Quadrotor Helicopter". PhD thesis. Cranfield University, 2007, p. 289.

[12] Michael V. Cook. *Flight Dynamics Principles*. Elsevier Ltd.

[13] David K Schmidt. *Modern flight dynamics*. Vol. 271. 5. 2003, p. 433.

[14] Randal Beard. "Quadrotor Dynamics and Control Rev 0.1". In: *All Faculty Publications* 1325 (2008), p. 48.

[15] *Euler_Angles_3*. URL: https://charlestytler.com/modeling-vehicle-dynamics-euler-angles/.

[16] Charles Tytler. *Modeling Vehicle Dynamics*. 2017. URL: https://charlestytler.com/.

[17] *T_Motor*. URL: https://www.getfpv.com/tiger-motor-mt-4006-13-740kv-1.html.