

# Modelling, and Design of a Quadcopter With MATLAB Implementation

Omar Horan<sup>1</sup>, Amarendra Reddy<sup>2</sup>

<sup>1</sup>M.Tech Student in control and Automation, Department of Electrical Engineering, AU Engineering college, Andhra University, Visakhaptnam, Andhra Pradesh, India

<sup>2</sup>Associate Professor, Department of Electrical Engineering, AU Engineering college, Andhra University, Visakhaptnam, Andhra Pradesh, India

\*\*\*

**Abstract** -: Quadcopter is an unmanned aerial vehicle, which can be implemented in different applications. In paper it will be represented a development of a quadcopter system. In The paper we will demonstrate how to model a basic physical quadcopter, then increase the complexity of the plant model, and finally demonstrate how the model can be implemented in MATLAB/Simulink. The advantages of utilizing MATLAB/Simulink to create and implement these models come from the tool's processing power as well as the simplicity with which Simulink models may be applied to hardware.

**Key Words:** model, state variables, quadcopter, plant, sensor

## 1.INTRODUCTION

Quadcopters are an emerging technology gaining in popularity. Most users purchase quadcopters fully assembled with everything that's needed to fly. One of the most interesting aspects of quadcopters that may make flying them even more amazing is understanding how they work. Believe it or not, users have the ability to install their own control system in order to have the quadcopter fly exactly how they want, depending on hardware of course. Many users are limited by the controller that comes preinstalled on quadcopters. A custom control system, depending on the complexity of the system, can allow the quadcopter to have automatic functions. These functions could include flips, or inverted flight. However, to do this, a plant model of the quadcopter is needed to simulate flight to prove the control system works prior to installing on the quadcopter. Creating a plant model and simulating it provides a prediction of what the real-world quadcopter will do. Quadcopter plant models can increase in accuracy depending on how involved the modelling process is. The idea of controlling and creating plant model of a quadcopter is rather complex but can be broken down significantly so it can be easily understood. Control systems and plant models can be seen almost everywhere without knowing it. The three pieces required to create a model are the control system, a plant model and sensors. Since the control system and plant model of a quadcopter can be complex, the paper will demonstrate how to model the basic physical quadcopter, then increase the plant model's complexity and then demonstrate how

the model can be implemented in MATLAB\Simulink. It also demonstrates how to build and tune controllers that re able to control a quadcopter's roll, pitch, yaw, altitude and motor dynamic. The benefits of using MATLAB\Simulink to create and implement these models come from the computing power of the software tool, along with ease of implementation of Simulink models to hardware. document is template. We ask that authors follow some simple guidelines. In essence, we ask you to make your paper look exactly like this document. The easiest way to do this is simply to download the template, and replace(copy-paste) the content with your own material. Number the reference items consecutively in square brackets (e.g. [1]). However, the authors name can be used along with the reference number in the running text. The order of reference in the running text should match with the list of references at the end of the paper.

### 1.1 Building the Plant Model

With the use of quadcopter in outdoor applications, researchers are interested in evaluating the performance of modelling and controlling of the quadcopter. To understand the quadcopter system modelling the block seen in Figure 1 can be used.

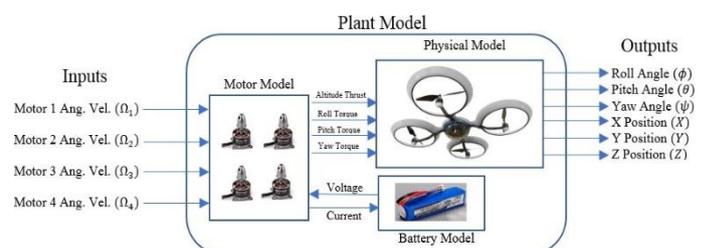


Figure 1-Plant model realization

The quadcopter model is a system of equations that generate outputs based upon inputs, parameters, and system states. Within the broad plant model of the quadcopter there are separate plant models such as the body dynamics, motor, and battery models. There is somewhat of a flow of variables within the plant model. The system starts with the inputs, which in this case are motor angular velocity commands. With these commands, states within the plant model angular velocity, and axial velocity. Parameters are constants that effect the states,

such as the friction coefficient of the motors. The outputs of the plant are based on the previous states, parameters, and inputs. Lastly, it is important to note that the states dependent on the frame of reference, or where the zero point is the ideal plant model would simply be the physical model, assuming instant thrust, no disturbances, and unlimited state values. A real-world model would include motor and battery dynamics along with state limitation.

### 1.2 Quadcopter Plant model Parameters

Quadcopters inherently have six degrees of freedom. This means the device can travel in the X, Y and Z direction, as well as rotate in roll, pitch and yaw directions. These terms are generally related to most aircraft, subs, and robotics as not many other platforms are able to control all 6 degrees. The roll and pitch are what allow the quadcopter to move in the X and Y directions. Roll can be defined as the rotation around the X axis, pitch can be defined as the rotation around the Y axis and yaw can be defined as the rotation around Z axis. With this, the axis of rotation for the quadcopter and state variables can be defined. State variables are the physical dynamics of the quadcopter that include angles and positions. Many of the states are dependent on other states and user input. The reference frame of the quadcopter will be local NED. Which is why the Z axis is pointed down.

Table -1: Quadcopter states

Quadcopter states			
Symbol	Description	Unit	Observability
$\theta$	Pitch Euler Angle	rad	Stereo Vision
$\dot{\theta}$	Pitch Euler Angular Velocity	rad/s	Gyroscope
$\ddot{\theta}$	Pitch Euler Angular Accel.	rad /s <sup>2</sup>	-
$\phi$	Roll Euler Angle	rad	Stereo Vision
$\dot{\phi}$	Roll Euler Angular Velocity	rad/s	Gyroscope
$\ddot{\phi}$	Roll Euler Angular Accel.	rad /s <sup>2</sup>	-
$\psi$	Yaw Euler Angle	rad	Stereo Vision
$\dot{\psi}$	Yaw Euler Angular Velocity	rad/s	Gyroscope

$\ddot{\psi}$	Yaw Euler Angular Accel.	rad /s <sup>2</sup>	-
X	Position in X	m	Stereo Vision
$\dot{X}$	Velocity in X	m/s	-
$\ddot{X}$	Accel. in X	m /s <sup>2</sup>	Accelerometer
Y	Position in Y	m	Stereo Vision
$\dot{Y}$	Velocity in Y	m/s	-
$\ddot{Y}$	Accel. in Y	m /s <sup>2</sup>	Accelerometer
Z	Position in Z	m	Stereo Vision
$\dot{Z}$	Velocity in Z	m/s	-
$\ddot{Z}$	Accel. in Z	m /s <sup>2</sup>	Accelerometer
$\Omega_1$	Motor 1 Angular Velocity	rad/s	Voltage/Current/Optical
$\Omega_2$	Motor 2 Angular Velocity	rad/s	Voltage/Current/Optical
$\Omega_3$	Motor 3 Angular Velocity	rad/s	Voltage/Current/Optical
$\Omega_4$	Motor 4 Angular Velocity	rad/s	Voltage/Current/Optical

$\theta$ -Theta (/they-tuh/),  $\phi$ -Phi (/fi/),  $\psi$ -Psi (/sai/),  $\Omega$ -Omega (/oh-mega/)

Before generating the plant model equations of motion, Parameter need to be defined. It is important to know the difference between states and parameters. States are dynamics of the quadcopter, where parameters are constants associated with the quadcopter. The first parameters to be defined are the moment of inertia in each axis. The moment of inertia is the amount of torque needed for a desired angular acceleration around a rotation axis. These relies heavily on the physical design and components of the quadcopter. These are generated through various methods which mostly include estimates from lengths/diameters and masses of the quadcopter. The rotor inertia, which is the total rotational moment of inertia around the propeller axis is another parameter used in the equations of motion. The distance from the rotor axis to the center of the quadcopter also can be defined, along with the thrust and drag coefficients. These can be seen in Table 2 below.

**Table -2:** Quadcopter parameters

Symbol	Description	Unit
$I_{xx}, I_{yy}, I_{zz}$	Inertia moments	$Kg\ m^2$
$J_r$	Rotor inertia	$Kg\ m^2$
$l$	Rotor axis to copter center distance	$m$
$b$	Thrust coefficient	$N/s^2$
$d$	Drag coefficient	$Nm/s^2$

The next set of parameters have to do with the quadcopter and environment. In the real world, quadcopters experience aerodynamical effects, namely from air resistance and wind. Air resistance can be thought of as a drag on the quadcopter, opposing the motion of the quadcopter. This is what makes quadcopters slow down when no actuator force is acting on them. To envision air resistance, one can simply drop a piece of paper, and a pencil from the same height. Since air resistance effects the quadcopter in all 3 axial directions, a coefficient is needed for each. Measuring the air resistance is extremely complex, so in most cases estimates suffice. The values used for the plant model were 0.1 kg/s for the resistance in each axis. The parameters can be seen in Table 3 below.

**Table -3:** environmental parameters.

Symbol	Description	Unit
$A_x, A_y, A_z$	Air resistance in each axis	$kg/s$

### 1.3 Plant Model Mathematical Equations

Before the quadcopter is modeled in MATLAB/Simulink the mathematical equations need to be generated. To do this a few assumptions are made [1].

- The drone is a rigid structure
- The air frame and components are symmetric
- The center of gravity and air frame origin coincide
- Thrust and drag are proportional to the square of propeller speed
- There are no external disturbances to the quadcopter such as wind, temperature, etc.

Next, The physical effects on the quadcopter need to be considered. These include aerodynamic effects from the propellers, inertial counter-torques from changes in propeller speed, gravity and gyroscopic effects from changes in body orientation. These all play a role in the equations of motion of the quadcopter. The internal

plant model equations of the system can now be described. There are four main thrust equations, vertical thrust, roll thrust, pitch thrust and yaw thrust with the “+” coordinate which can be seen in table four.

**Table -4:** input Equations in the “+” orientation

Input	Thrust Equation	Description
$U1_+$	$b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2)$	General Thrust
$U2_+$	$b(-\Omega_2^2 + \Omega_4^2)$	Roll Thrust
$U3_+$	$b(\Omega_1^2 - \Omega_3^2)$	Pitch Thrust
$U4_+$	$d(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2)$	Yaw Thrust

The quadcopter inputs can also be defined in the “X” orientation [2]. These input equations can be seen in table 5 below.

**Table -5:** input Equations in the “x” orientation

Input	Thrust Equation	Description
$U1_+$	$b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2)$	General Thrust
$U2_+$	$b\sin(\frac{\pi}{4}) (\Omega_1^2 - \Omega_2^2 - \Omega_3^2 + \Omega_4^2)$	Roll Thrust
$U3_+$	$b\sin(\frac{\pi}{4}) (\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2)$	Pitch Thrust
$U4_+$	$d(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2)$	Yaw Thrust

The overall angular velocity of the all propellers can also be defined. This will be used in angular acceleration equations. The positives and negatives in equation (9) are used because of the direction the propeller is spinning. In most cases the opposite side motors are spinning in the same direction, and 2 and 4 in the same direction).

$$\Omega_r = \Omega_1 - \Omega_2 + \Omega_3 - \Omega_4 \tag{9}$$

Now equations of motion can be defined

**Table -6:** Plant model equations

Rolling Torque	$I_{xx}\ddot{\phi} = \dot{\theta}\psi(I_{yy} - I_{zz}) + J_r\dot{\theta}\Omega_r + l(U2)$
Roll Ang. Accel	$\ddot{\phi} = \frac{\dot{\theta}\psi(I_{yy} - I_{zz}) + J_r\dot{\theta}\Omega_r + l(U2)}{I_{xx}}$
Pitching Torque	$I_{yy}\ddot{\theta} = \dot{\phi}\psi(I_{zz} - I_{xx}) - J_r\dot{\phi}\Omega_r + l(U3)$
Pitch Ang. Accl.	$\ddot{\theta} = \frac{\dot{\phi}\psi(I_{zz} - I_{xx}) - J_r\dot{\phi}\Omega_r + l(U3)}{I_{yy}}$
Yawing Torque	$I_{zz}\ddot{\psi} = \dot{\theta}\dot{\phi}(I_{xx} - I_{yy}) + (U4)$

Yaw Ang. Accl.	$\ddot{\psi} = \frac{\theta \dot{\phi} (I_{xx} - I_{yy}) + (U_4)}{I_{zz}}$
X force	$m\ddot{X} = (\sin \psi \sin \phi - \cos \psi \sin \theta \cos \phi)U_1 - A_x\dot{X}$
X Accl.	$\ddot{X} = \frac{(\sin \psi \sin \phi - \cos \psi \sin \theta \cos \phi)U_1 - A_x\dot{X}}{m}$
Y force	$m\ddot{Y} = (\cos \psi \sin \phi + \sin \psi \sin \theta \cos \phi)U_1 - A_y\dot{Y}$
Y Accl.	$\ddot{Y} = \frac{(\cos \psi \sin \phi + \sin \psi \sin \theta \cos \phi)U_1 - A_y\dot{Y}}{m}$
Z force	$m\ddot{Z} = mg - (\cos \theta \cos \phi)U_1 - A_z\dot{Z}$
Z Accl.	$\ddot{Z} = \frac{mg - (\cos \theta \cos \phi)U_1 - A_z\dot{Z}}{m}$

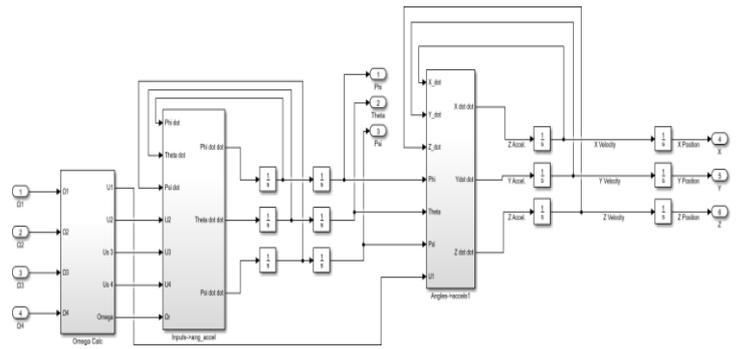


Figure 2 Quadcopter Plant feedback system

### 2.1 Basic Simulink Plant model

Now that the basic equations of motion have been generated, the Simulink plant model can now be formed. First, a basic script that defines the inertia moments, rotor inertia moments, rotor inertia, length, mass, thrust, and drag coefficients need to be created. Then the Simulink model is created. Starting a fresh Simulink model block diagram, First define the solver options. On the top bar of Simulink hit Simulation. Then Model configuration parameters. Under solver option, make sure the type is variable step, and the solver is on Auto. A new subsystem can now be defined. The subsystem has four inputs ( $\Omega_1, \Omega_2, \Omega_3, \Omega_4$ ) and six outputs (X, Y, Z, roll, pitch, yaw). At this point the model should be looking like the following figure below

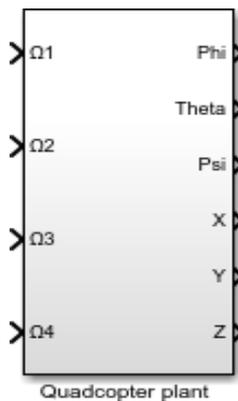


Figure 1 Quadcopter Plant

After implementing the previous mathematical equations, we get the following quadcopter plant feedback system

### 2.2 Basic Simulation Results

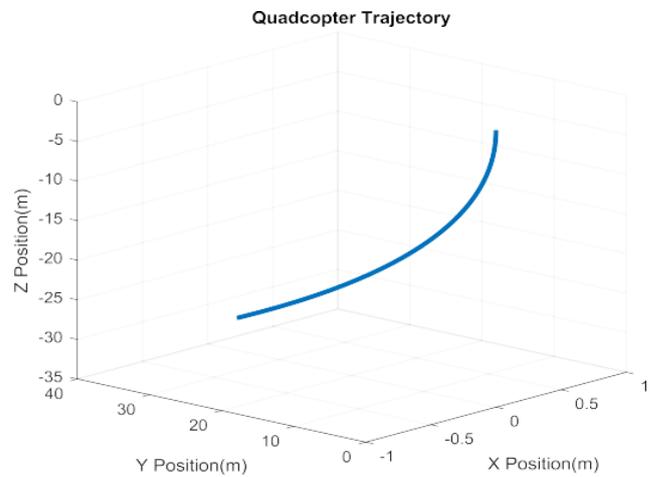


Figure 3 Roll Plant Simulation

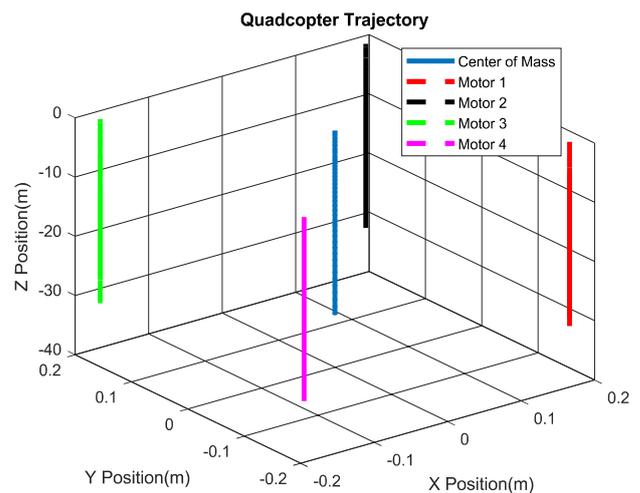


Figure 4 Thrust Plant Simulation

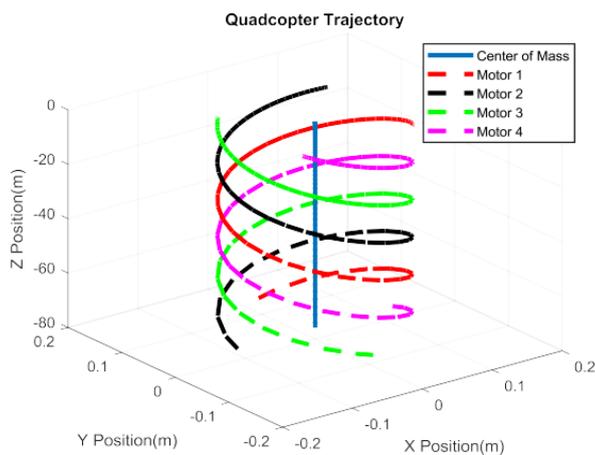


Figure 5 Quadcopter Trajectory Simulation

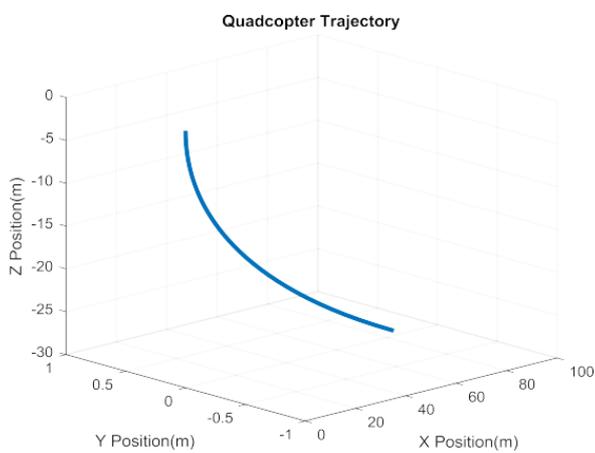


Figure 6 pitch Plant Simulation

[3] Carrillo, L. R., Lozano, R., & Pégard C. (2013). Modeling the Quad-Rotor Mini-Rotorcraft. *Advances in Industrial Control Quad Rotorcraft Control* 23-34 doi:10.1007/978-1- 4471-4399-4\_2.

[4] Horn, G. (2009, March 17). The Aerospace Euler Angles. Retrieved November06, 2017

[5] P.-J. Bristeau, P. Martin, E. Salaun, and N. Petit, “The role of propeller aerodynamics in the model of a quadrotor,” in *European Control Conference*, (Budapest, Hungary), 2009.

[6] W. Khan and M. Nahon, “A propeller model for general forward flight conditions,” *Inter J of Intelligent Systems*, 2015.

### 3. CONCLUSIONS

Quadcopters with accelerating industry are an interesting platform to perform different modelling strategies. The goal was to develop a simple modelling approach using MATLAB-Simulink. The vehicle’s Simulink model was made, based on derived differential Simulink model simulated perfectly depending on the equations of motion. This research shows how the original complex model of quadcopter is modeled taking into account wind effects.

### REFERENCES

[1] B. Fares, P. Apkarian, D. Noll, An Augmented Lagrangian Method for A class of LMI-Constraint Problems in Robust Control Theory. *Int. J. Control*. 2001.

[2] Carrillo, L. R., López, A. E., Lozano, R., & Pégard, C. (2013). Modeling the Quad-Rotor Mini-Rotorcraft. *Advances in Industrial Control Quad Rotorcraft Control*, 23-34. doi:10.1007/978-1- 4471-4399-4\_2