Report on Control system of Quadrocopter

# I-FUFO's Quadrocopter requirement

To be sure about the control method we will use in this project, we have to classify the requirements of this Quadrocopter:

- The Quadrocopter **must**:

+ Keep balance in the middle of the air.

- The Quadrocopter **should**:

+ Maintain a reasonable specified altitude.

- The Quadrocopter **do not need to**:

+ Hover from point A to point B in space automatically.

+ Adjust its position in the space (Similar to hover from point A to point B).

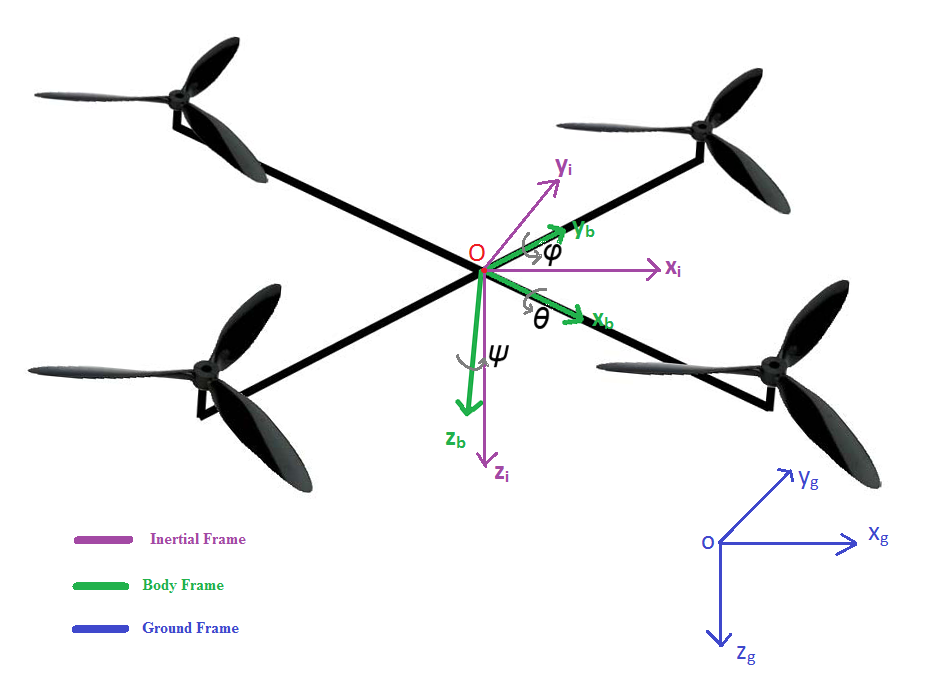
Those requirement is for simplifying the Control system of Quadrocopter, which gives advantages to the development time and hardware design since we only use Accelerometer and Gyroscope in this project.

# II-Analysis the requirement

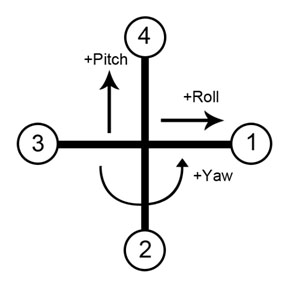
## a. The "must" requirement:

According to my previous report, to keep the Quadrocopter balance in the middle of the air, only three variables of the Quadrocopter dynamic need to be considered: 𝜓 (yaw), 𝜃 (roll), 𝜑(pitch). Each of these variables is an Euler angle which represent the difference between the Body frame and the Inertial Frame, given that the considered Inertial Frame has its Origin (O) at the same place as the Body frame's Origin, at the Mass Center of the Quadrocopter. This Inertial Frame has the same Oz's direction as the Ground Frame. (**Figure 1**)

That is the theory, in fact, we have the Origin of the Inertial Frame and Body Frame according to the Accelerometer sensor's Origin because it is the only measurable Origin in the system. The Measured variables in Accelerometer can be converted to 𝜓, 𝜃, 𝜑 via some equations which will be discussed in the next parts of this report.

**Figure 1**

After measuring all the needed angle, the control system will automatically calculate the needed PWM for each motor to rotate the Body Frame into the Inertial Frame. We will use PID control system to archive this. Figure 2 shows the Euler Angles according to Right Hand Rule.



**Figure 2**

Each yaw, pitch, roll angle will need a PID equation to calculate the needed PWM value. We can simply understand this by assume the Inertial Frame's angles are the "**desired angles**" and the Body Frame's angles are the "**actual angles**". Each yaw, pitch, roll angle will have the following equation in embedded environment:

**e = desired angle - actual angle**

**P = Kpe**

**I = Ki(I + e)\*dt**

**D = Kd(current\_e - past\_e)/dt**

**angle\_sum = P + I + D**

where "**dt**" is the time period between each read from sensors.

According to firgure 2 and the PID control system, PWM of each motor can be calculate as:

**PWM\_motor1 = – roll\_sum + yaw\_sum**

**PWM\_motor2 = pitch\_sum – yaw\_sum**

**PWM\_motor3 = roll\_sum + yaw\_sum**

**PWM\_motor4 = – pitch\_sum – yaw\_sum**

## b. The "should" requirement:

To maintain a specified altitude, the Quadrocopter must have a sensor which measure the distant between the ground and the vehicle. That sensor can be either an Infrared RangeFinder sensor or a Pressure Sensor. However, including another sensor may increase the development time, and lead to many risk.

After review many solution on this problem, we decided that the Quadrocopter will maintain a **blinded-altitude**. A blinded-altitude is an altitude measured by the forces of motor, not by the metric system.

We are assumed that there is a constant **k**, that:

**Motor's force = k\*PWM value.**

What we input into the motor controller is a PWM value, and then the motor will produce a corresponding force. That force will lift the Quadrocopter to an expected altitude which depends mostly on the characteristic of that environment (Temperature, air density, position on the earth, etc). To get to this expected altitude, the pilot need to observe the current altitude of the Quadrocopter and adjust the PWM value. The expected PWM value is called the "**Thrust Floor**". This Thrust Floor is only related to the altitude measurement of the Quadrocopter, therefore, we have to combine it with the PWM values of pitch, roll, yaw angles to make the Quadrocopter balance in a expected altitude. The PWM value of each motor is now calculated as:

**PWM\_motor1 = Thrust\_Floor – roll\_sum + yaw\_sum**

**PWM\_motor2 = Thrust\_Floor + pitch\_sum – yaw\_sum**

**PWM\_motor3 = Thrust\_Floor + roll\_sum + yaw\_sum**

**PWM\_motor4 = Thrust\_Floor – pitch\_sum – yaw\_sum**