

# Autonomy in Motion (<https://charlestytler.com/>)

A blog on autonomy and dynamics which tends to diverge on tangents

[HOME \(HTTP://CHARLESTYTLER.COM\)](http://CHARLESTYTLER.COM)

[QUADCOPTER \(HTTPS://CHARLESTYTLER.COM/CATEGORY/QUADCOPTER/\)](https://CHARLESTYTLER.COM/CATEGORY/QUADCOPTER/)

[VIRTUAL REALITY \(HTTPS://CHARLESTYTLER.COM/CATEGORY/VIRTUAL-REALITY/\)](https://CHARLESTYTLER.COM/CATEGORY/VIRTUAL-REALITY/)

[ABOUT ME \(HTTPS://CHARLESTYTLER.COM/ABOUT-ME/\)](https://CHARLESTYTLER.COM/ABOUT-ME/)

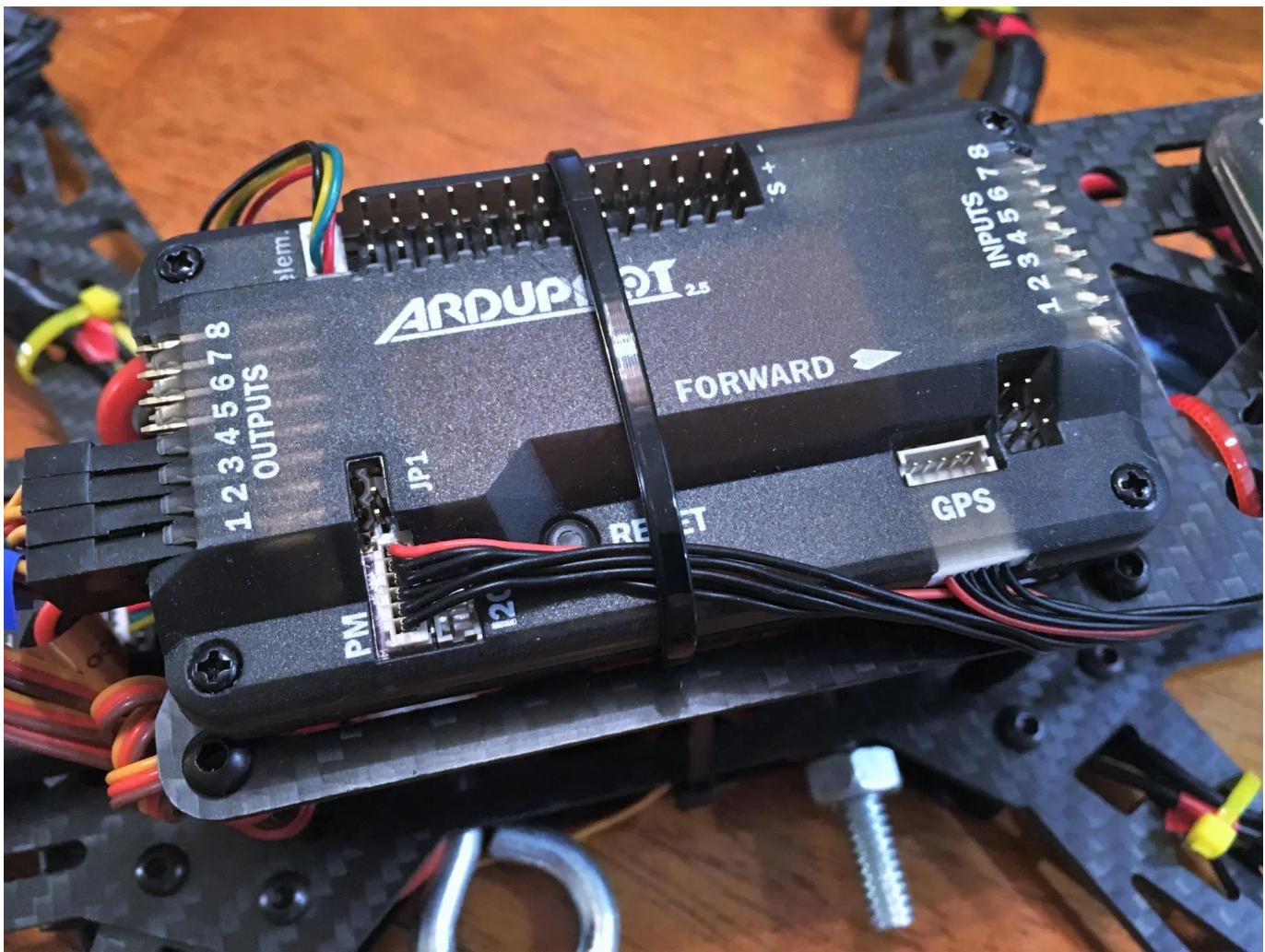
## Measuring Quadcopter Attitude with Sensors

MAY

30

 CHARLIE ([HTTPS://CHARLESTYTLER.COM/AUTHOR/ADMIN/](https://CHARLESTYTLER.COM/AUTHOR/ADMIN/))





In modeling the quadcopter for simulation we have full knowledge of the vehicle attitude, however for the real hardware we have limited observability with our sensors, and must deal with the errors that come with those sensors. In this post I will go over how we can estimate the quadcopter's orientation in Euler angles from the available sensors. None of the sensors directly measure an Euler angle, so instead we must calculate estimates for the angles based on the information the sensors provide us. I will show how we can derive orientation information from the gyroscopes, magnetometer, and accelerometers; all of which are included on the APM board. This post will be limited to the individual measurements the sensors provide; how we can combine them through sensor fusion will be covered in another post.

## Gyroscopes

The first sensor is the primary sensor for attitude determination. Gyroscopes measure angular rates which must be integrated over time to estimate angular orientation. Because it must be integrated means that we need some initial condition which we can then add on to.

### Initial Condition

The initial condition can either be assumed based on operating procedure (for example, vehicle is only started up on level ground) or from other sensors which provide absolute orientation measurements (as we'll see the magnetometer and accelerometers can provide).

## Transforming Measured Angular Rates to Euler Angle Rates

Once we have an initial condition of vehicle attitude identified, we can integrate the measured angular rates to propagate the orientation over time. One caveat though, is that the measured rates from the gyros represent the angular velocity about the body-fixed frame, which is not the same as the rate of change of the Euler angles (<http://charlestytler.com/modeling-vehicle-dynamics-euler-angles/>).

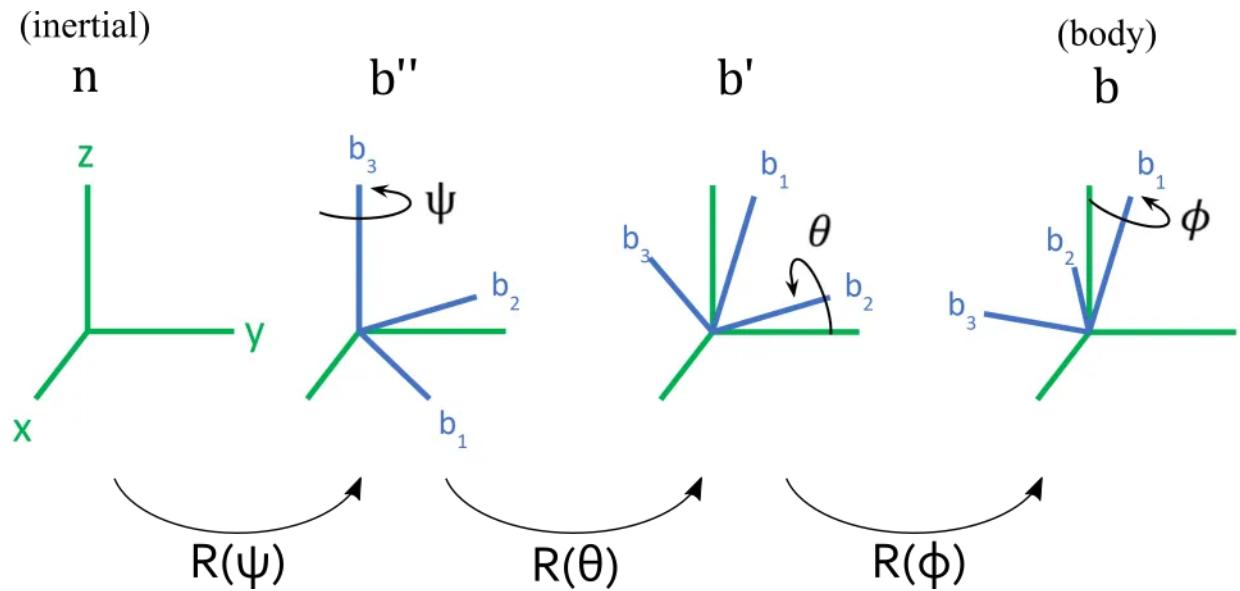
This is because the Euler angles represent a sequence of rotations each in their own intermediate frame of reference, whereas the angular velocity is the instantaneous angular rate about each of the body-fixed axes. We can use coordinate transformations to express the angular velocity as a function of the Euler angle derivatives by transforming them all into the body frame.

**Measured angular velocity**  $\omega^b = (p, q, r)$ , **Euler angle rates**  $\dot{\Phi} = (\dot{\phi}, \dot{\theta}, \dot{\psi})$

To help make this clear I've labeled the intermediate reference frames in the sequence of Euler rotations below as  $b''$  and  $b'$ . So if we want to know how  $\dot{\psi}$  would be represented in the body frame ( $b$ ), we must transform it from its intermediate frame. Here we see  $\dot{\psi}$  is the rotation rate about unit vector  $\hat{b}_3$  in the  $b''$  frame, so we will refer to it as  $\hat{b}_3''$ . However in the final body-fixed frame,  $b$ , that unit vector has been rotated by the second and third Euler angles in the sequence.

Therefore, the rotation rates measured in the body-fixed frame  $(p, q, r)$  due to the rate of change of the first Euler angle will equal the  $\dot{\psi}$  rotation transformed by the  $\theta$  and  $\phi$  rotations. (Since its rotation angle is also parallel with the inertial axis  $\hat{z}$ , you could get the same answer by multiplying it by all three Euler angle rotations with the  $C_n^b$  inertial-to-body coordinate transformation derived in Modeling Vehicle Dynamics – Euler Angles (<http://charlestytler.com/modeling-vehicle-dynamics-euler-angles/>)).

The second Euler rotation,  $\dot{\theta}$  is treated similarly, and  $\dot{\phi}$  needs no rotation as the unit vector it rotates around,  $\hat{b}_1' = \hat{b}_1$ , is a unit vector in the body frame,  $b$ .



Sequence of Euler Angle Rotations (Body 3-2-1) from inertial to body-fixed reference frames, with intermediate reference frames  $b''$  and  $b'$

The relationship between measured gyro rates and Euler angle rates can be expressed in scalar form multiplied by unit vectors as:

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

If we express it in matrix form however, we can take advantage of our rotation matrices to handle the disparate reference frames:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix}^b = \mathbf{R}(\phi)\mathbf{R}(\theta) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix}^{b''} + \mathbf{R}(\phi) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix}^{b'} + \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix}^b$$

To make things a little easier to read and fit on the page, I'm going to use a shorthand for the trigonometric functions:  $c_\theta = \cos(\theta)$  and  $s_\theta = \sin(\theta)$ . The above expression can be simplified by multiplying the matrices.

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & s_\phi \\ 0 & -s_\phi & c_\phi \end{bmatrix} \begin{bmatrix} c_\theta & 0 & -s_\theta \\ 0 & 1 & 0 \\ s_\theta & 0 & c_\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & s_\phi \\ 0 & -s_\phi & c_\phi \end{bmatrix} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix}$$

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

So finally we arrive at the set of equations:

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

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

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

These equations can also be rewritten for the Euler angle derivatives:

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

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

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

## Estimating Euler Angles from Rate Measurements

These equations can be used to propagate the attitude of the vehicle for each successive sampling frame using numerical integration techniques. If your sampling time is small enough, or the rates are small, a first order approximation may be sufficient. The estimated orientation for new frame  $k + 1$  would be a function of the estimated orientation and measured angular velocity at time frame  $k$  where  $t(k + 1) - t(k) = \Delta t$ :

$$\Phi_{k+1} = \Phi_k + \dot{\Phi}(\Phi_k, \omega_k^b) \Delta t \text{ where } \Phi := [\phi, \theta, \psi]^T$$

An example of this in Python code would be:

```

1. # Derivative function
2. def xdot(x,u):
3.     x_dot = [0,0,0]
4.     # phidot = p + (q*sin(phi) + r*cos(phi))*tan(theta)
5.     x_dot[0] = u[0] + (u[1]*sin(x[0]) + u[2]*cos(x[0]))*tan(x[1])
6.     # thetadot = q*cos(phi) - r*sin(phi)
7.     x_dot[1] = u[1]*cos(x[0]) - u[2]*sin(x[0])
8.     # psidot = (q*sin(phi) + r*cos(phi))*sec(theta)
9.     x_dot[2] = (u[1]*sin(x[0]) + u[2]*cos(x[0]))*sec(x[1])
10.    return x_dot
11.
12. # Initial condition setup
13. eulerAngles = [0,0,0]           #Assume start up with phi = theta = psi = 0 radians
14. deltaT = 0.01                  #Sampling time (sec)
15.
16. # While loop
17. while True:
18.     omegaRead = gyros.getAngularVelocity()
19.     eulerAngles += xdot(eulerAngles,omegaRead) * deltaT

```

## Errors

Gyroscopes will have errors you may need to account for in your attitude estimation. These include measurement bias as well as scaling and cross-coupling errors. These errors can be accounted for through calibration during initial startup or estimated by a Kalman filter while operating.

The primary concern with any estimate achieved through integration is drifting orientation estimates due to errors being accumulated over time through the integration. A way to counteract this is to perform sensor fusion (through techniques such as a complimentary

filter or a Kalman filter) with sensors that measure absolute orientation.

# Magnetometer

With a magnetometer we can receive a 3-axis reading of the magnetic field of the Earth. However, while we read a 3-dimensional vector from the sensor, it is generally only used to gather an estimate of the heading angle of a vehicle. This is because the geometry of the magnetic field is predominantly parallel to the ground in most areas away from the poles, and thus is not conducive to estimating pitch or roll.

## Initial Condition

Magnetic north does not directly align with true north. If you wish to resolve the vehicle's orientation in a reference frame for navigation along cardinal directions or latitude/longitude, you will need to correct your heading measurements to true north. In order to do this, one needs the declination angle ( $\alpha$ ) for the vehicle's location about Earth. One way to do this is to simply store a lookup table that can be queried with a known starting position or measured/estimated location. The magnetic field in the inertial frame is equal to the following expression with declination angle, inclination angle ( $\gamma$ ), and magnetic field magnitude  $|B|$ :

$$\mathbf{b}^n = \begin{bmatrix} \cos \alpha \cos \gamma \\ \sin \alpha \cos \gamma \\ \sin \gamma \end{bmatrix} |B|$$

In some instances where navigation is not as important, the magnetic heading may be all that is needed to correct for gyro drift in attitude estimation. Either way, by measuring an absolute orientation, the magnetometer can be used to initialize the heading of the vehicle before propagating orientation through use of the gyros.

## Calculating Heading ( $\psi$ ) from Magnetometer Readings

The magnetometer is fixed to the vehicle frame, and thus will produce magnetic flux readings in the body frame. To find magnetic heading, we first need to reorient the magnetic reading into a frame level with the ground. To do this I will define a new reference frame called the "level frame" (L) which is simply only one Euler angle ( $\psi$ ) rotation from the inertial reference frame. This means that the Level frame is aligned with the heading of the vehicle, but is before the pitch and roll Euler angle rotations. We can transform our body frame magnetometer reading,  $\mathbf{m}^b = [m_x^b, m_y^b, m_z^b]^T$ , into the level frame with the rotation matrices:

$$\mathbf{m}^L = \begin{bmatrix} c_\theta & 0 & -s_\theta \\ 0 & 1 & 0 \\ s_\theta & 0 & c_\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & s_\phi \\ 0 & -s_\phi & c_\phi \end{bmatrix} \mathbf{m}^b = \begin{bmatrix} c_\theta & s_\phi s_\theta & c_\phi s_\theta \\ 0 & c_\phi & -s_\phi \\ -s_\theta & c_\theta s_\phi & c_\phi c_\theta \end{bmatrix} \begin{bmatrix} m_x^b \\ m_y^b \\ m_z^b \end{bmatrix}$$

This transformation provides us with the magnetic field strength in the x and y components of the level frame as a function of the body frame measurements:

$$m_x^L = m_x^b \cos \theta + m_y^b \sin \phi \sin \theta + m_z^b \cos \phi \sin \theta$$
$$m_y^L = m_y^b \cos \phi - m_z^b \sin \phi$$

To determine the magnetic heading, we take the arctangent of the x and y components in the level frame:

$$\psi_{mag} = \arctan2(-m_y^L, m_x^L)$$

And if we know the declination angle,  $\alpha$ , then the true heading is given by

$$\psi_{true} = \psi_{mag} + \alpha$$

An example Python code calculating magnetic heading,  $\psi$ , given magnetometer reading and estimated pitch and roll angles

```
1. def magHdg(magRead, phi, theta):
2.     # Takes magnetometer reading and estimated phi and theta Euler angles,
3.     # and calculates a measured psi Euler angle (magnetic heading)
4.     mx_b = magRead[0]
5.     my_b = magRead[1]
6.     mz_b = magRead[2]
7.     # Rotate by phi and theta estimates to Level (L) frame
8.     mx_L = mx_b*cos(theta) + my_b*sin(theta)*sin(phi) + mz_b*sin(theta)*cos(phi)
9.     my_L = my_b*cos(phi) - mz_b*sin(phi)
10.    #Calculate psi
11.    psi_meas = arctan2(-my_L, mx_L)
12.    return psi_meas
```

## Errors

The magnetometer is sensitive to sources of magnetism other than the Earth's magnetic field. Calibration is normally done to account for soft-iron and hard-iron magnetism from the vehicle itself. Other magnetic fields nearby the magnetometer may also influence its readings.

## Accelerometers

While accelerometers are generally used to measure accelerations to integrate into velocity and position, they can also be used to aid attitude estimation through accelerometer leveling. When the vehicle is stationary the 3-axis accelerometer will only detect the specific force due to gravity. Using the magnitude of the gravity measurement in each component, an estimate of pitch and roll orientation can be made.

## Initial Condition

For accelerometer leveling the only prior information needed is the magnitude of gravity,  $g$ , which should be stored as a constant. Accelerometer leveling may also be used to help define the initial  $\phi$  and  $\theta$  orientation of the vehicle during startup.

## Transforming Inertial Gravity Vector to Body Coordinates

Gravity is assumed to be a constant acceleration always oriented towards the center of the Earth. Since the accelerometers are attached to the vehicle frame, they will measure acceleration in the body fixed reference frame. We can use the coordinate transformation matrix,  $\mathbf{C}_n^b$ , from the definition of our Euler angles (<http://charlestytler.com/modeling-vehicle-dynamics-euler-angles/>) to transform the inertial gravity vector into body-fixed specific force measurements,  $\mathbf{f}^b$ .

$$\mathbf{f}^b = \mathbf{C}_n^b \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} = \begin{bmatrix} g \sin \theta \\ -g \cos \theta \sin \phi \\ -g \cos \phi \cos \theta \end{bmatrix}$$

## Calculating Pitch ( $\theta$ ) and Roll ( $\phi$ ) Attitude from Accelerometer Reading

By using the separate components of the accelerometer measurements we can see that

$$\frac{f_y^b}{f_z^b} = \frac{g \sin \phi}{g \cos \phi} = \tan \phi$$
$$\frac{f_x^b}{\sqrt{f_y^b + f_z^b}} = \frac{g \sin \theta}{\sqrt{g^2 \cos^2 \theta (\sin^2 \phi + \cos^2 \phi)}} = \frac{g \sin \theta}{g \cos \theta} = \tan \theta$$

And therefore can solve for the Euler angles by

$$\phi = \arctan2(-f_y^b, -f_z^b)$$

$$\theta = \arctan\left(\frac{f_x^b}{\sqrt{f_y^b + f_z^b}}\right)$$

A sample excerpt of Python code illustrates how this estimate can be calculated:

```
1. def accelLevel(accelRead):
2.     # Takes accelerometer specific force reading when stationary and estimates
3.     # phi and theta from the gravity vector
4.     fx_b = accelRead[0]
5.     fy_b = accelRead[1]
6.     fz_b = accelRead[2]
7.     phi_est = arctan2(-fy_b, -fz_b)
8.     theta_est = arctan2(fx_b, sqrt(fy_b*fy_b + fz_b*fz_b))
9.     return phi_est, theta_est
```

## Errors

Just as with the gyroscopes, accelerometers suffer from bias, scaling, and cross-coupling errors. These can be calibrated out during initial startup or estimated by a Kalman filter during operation.

Accelerometer leveling is only accurate when the vehicle is stationary. If there are accelerations then the specific force components are no longer measuring only gravity. One way to mitigate this is by using a threshold limit in acceleration readings to identify when an accelerometer leveling attitude estimate is acceptable. Additionally known states of rest could trigger an accelerometer leveling update.

## Next Steps

Now that we have identified how to extract Euler angle information from three independent sensor sources, we can next look at how to combine them all into a single, accurate estimate of orientation.

## Further Reading

- [1] Groves, Paul D. "Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems"

---

### Share this:

 (<https://charlestytler.com/measuring-quadcopter-attitude-sensors/?share=twitter&nb=1>)

 (<https://charlestytler.com/measuring-quadcopter-attitude-sensors/?share=facebook&nb=1>)

Modeling Vehicle Dynamics -  
Quadcopter Equations of Motion  
(<https://charlestytler.com/quadcopter-equations-motion/>)

Modeling Vehicle Dynamics -  
Euler Angles  
(<https://charlestytler.com/modeling-vehicle-dynamics-euler-angles/>)

Hardware Components and  
Build  
(<https://charlestytler.com/build-and-hello-world-scripts/>)

---

Posted in [Control](https://charlestytler.com/category/quadcopter/control/), [Quadcopter](https://charlestytler.com/category/quadcopter/)

Leave a comment or ask a question:

Enter your comment here...

[Home](https://charlestytler.com/) (<https://charlestytler.com/>)    [About Me](https://charlestytler.com/about-me/) (<https://charlestytler.com/about-me/>)

Theme Designed by [InkHive.com](http://inkhive.com/) (<http://inkhive.com/>). Unless otherwise stated, all content and images are created by Charles Tytler.