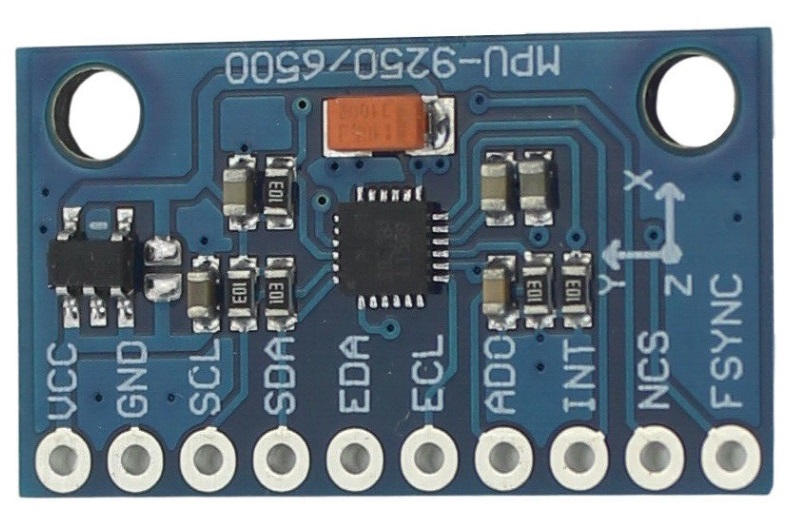
**10/26/2022 – Wednesday - Quaternions to Euler Angles Algorithm used for a Quadcopter 9DOF IMU Sensor**

MPU-9250 Module Shown below:



The object of this algorithm is to provide the three Euler angles θ, φ, and ψ, computed from the raw data coming from a low-cost 9 Degree-of-Freedom (DOF) microelectromechanical (MEMs) sensor.

Many low-cost 9 Degree-of-Freedom (DOF) microelectromechanical (MEMs) sensor such as Inertial Measurement Units (IMUs) can provide nine (9) sensor parameters. They are: 1) pitch rate, 2) roll rate, 3) yaw rate, 4) longitudinal acceleration, 5) lateral acceleration, 6) vertical acceleration, 7) X-earth magnetic field, 8) Y-earth magnetic field, and 9) Z-earth magnetic field.

Using the raw sensor data from the IMU, this paper presents an algorithm used compute the Euler angles, “theta”, “phi”, and “psi” by using Quaternions. Using Quaternions avoids issues with singularities should multiple rotations of the vehicle occur.

Rather than going through the theory of Quaternions using Clifford Algebra or Geometric Algebra, this paper goes into a simple practical implementation of using Quaternions to provide this algorithm. The understanding of Quaternions is very abstract and meaningless. There have been many other very completed methods to achieve this algorithm. However, I believe this is the simplest method.

Public domain references are used to derive this algorithm as included in this paper.

Complementary filters are used in this method. Quaternions are used to get short term “theta”, “phi”, and “psi” where the accelerometer is used to compute the long term “theta” & “phi”. The compass is used to compute the long term “psi”.

This is the standard axes convention normally used for aircraft.

The subscript “V” is for the vehicle axis and the subscript “B” is for the body axis.

1. The first rotation is in azimuth “psi”.



1. The second rotation is in pitch “theta”.



1. The third rotation is in roll “phi”.

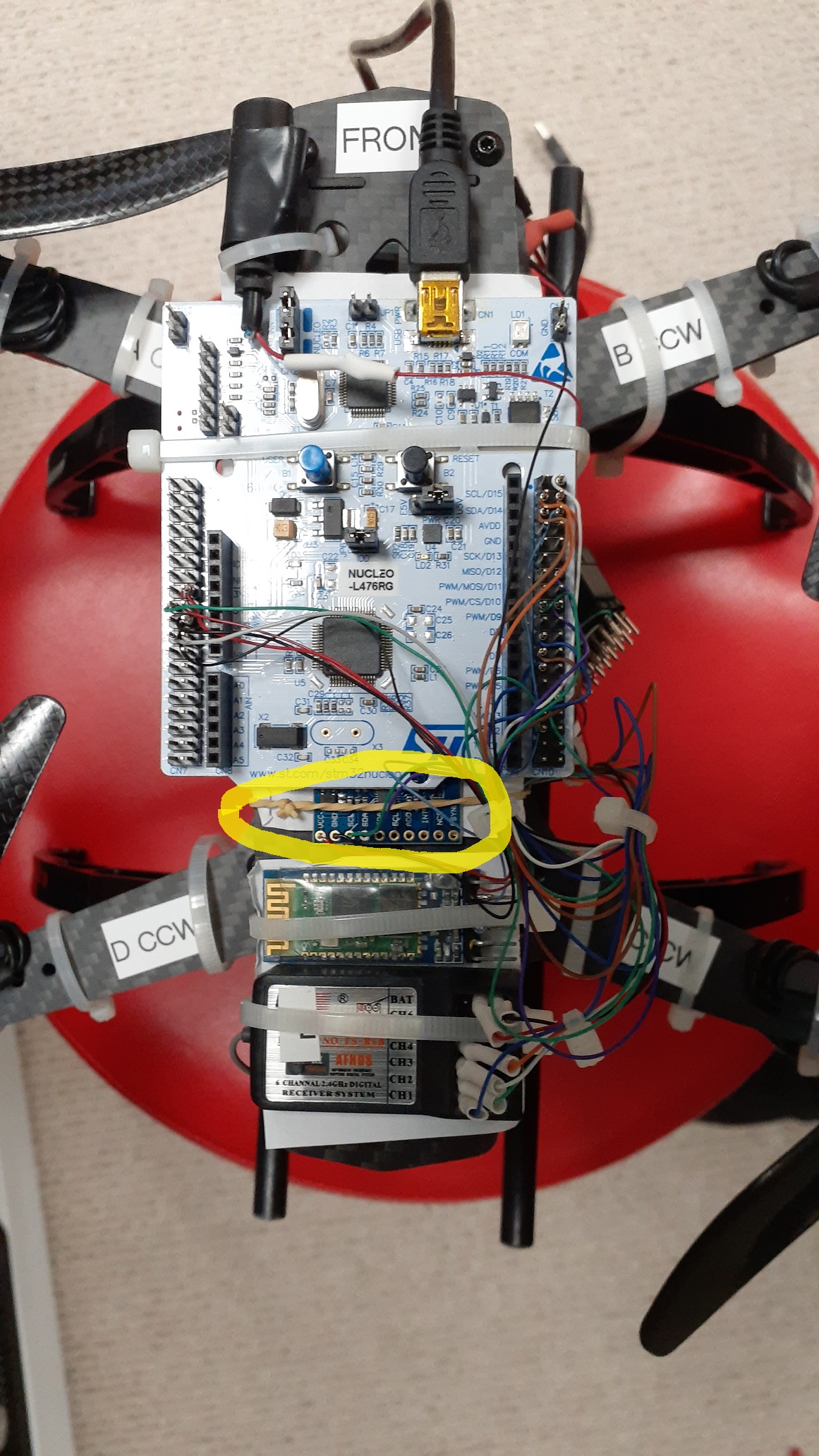


After multiplying the three rotation together, the following transformation is obtained:

Body Axis Angular Rates to Euler Angle Rates can be solved using the following equation:

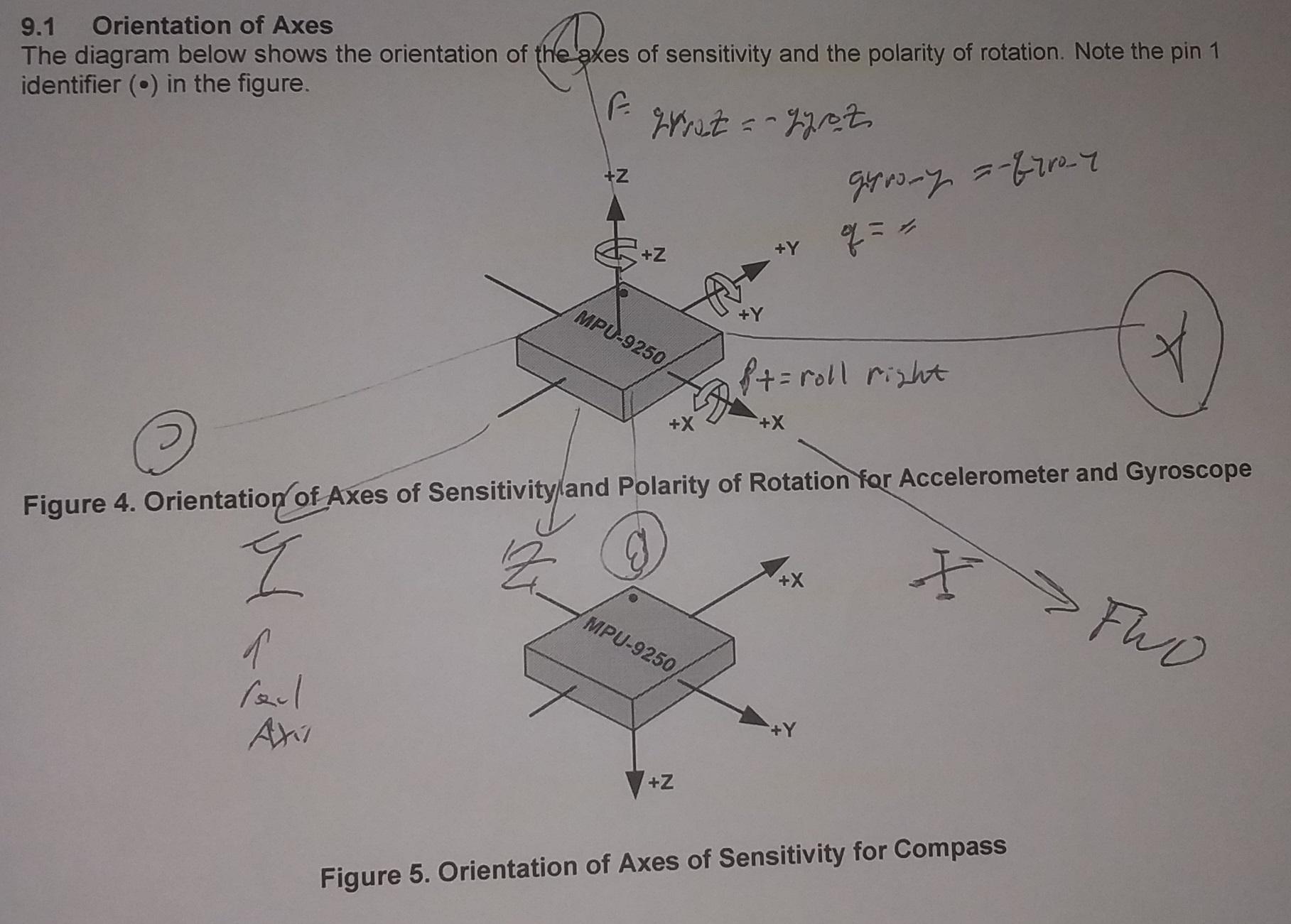
However, with the trigonometric functions in the matrix such as tangent and secant there are singularities. Therefore, this method is limited as there are singularities. Therefore a method using Quaternions is used to avoid these singularities. Quaternions will allow multiple rotations to avoid the issue with the singularities as shown in the above equation.

**Quadcopter that has a MPU-9250 IMU installed:**



**The IMU MPU-9250 is used (here is some polarity information from its data sheet):**

We first need to take the raw data from the sensor and adjust the polarities to get them to comply with the standard aircraft axes. Note the dot on the sheet on the device. In our aircraft as shown in Figure 4 below, the nose of the aircraft is aligned with the X axis, and the right wing of the aircraft is aligned with the -Y axis. The Z axis is shown in Figure 4 is opposite to that on the standard aircraft axes.



Relative to Figure 4, we want the conventional standard axes polarities where:

Forward – Nose = +X

Right Wing = +Y

Downward = +Z

Roll Rate Right Wing Up = +p

Pitch Rate Nose Up = +q

Yaw Rate Nose Right = +r

It has been found through experimentation that the accels are opposite to the polarities shown in the diagram in Figure 4.

In the Figure 4 above it shows:

Accel\_X has the reverse polarity.

Accel\_Y has the correct polarity.

Accel\_Z has the correct polarity.

Gyro\_X has the correct polarity for p.

Gyro\_Y has the reverse polarity for q.

Gyro\_Z has the reverse polarity for r.

So:

Relative to the gyro axes shown in Figure 4.

Where gyro\_x, gyro\_y & gyro\_z are the raw data taken from the sensor.

p = gyro\_x

q = -gyro\_y

r = -gyro\_z

Relative to the compass axes shown in Figure 5.

Where compass\_x, compass\_y & compass\_z are the raw data taken from the sensor.

c\_x = compass\_y

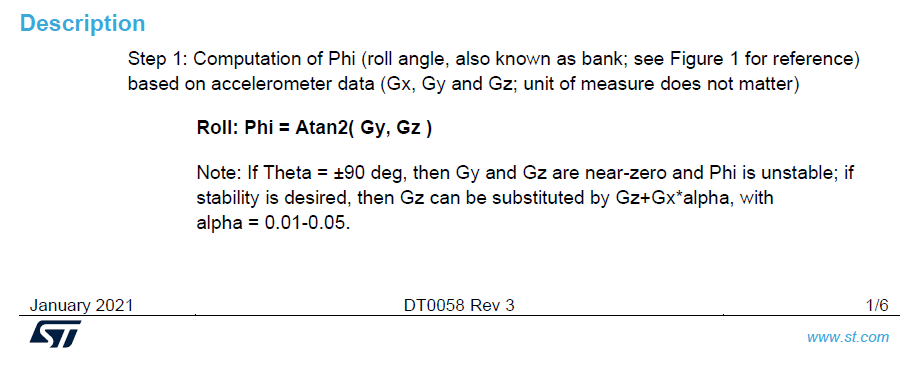
c\_y = -compass\_x

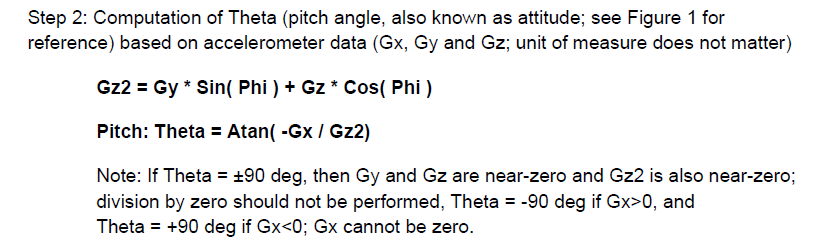
c\_z = compass\_z

**The Algorithm is broken into a few parts to make it easier to compute:**

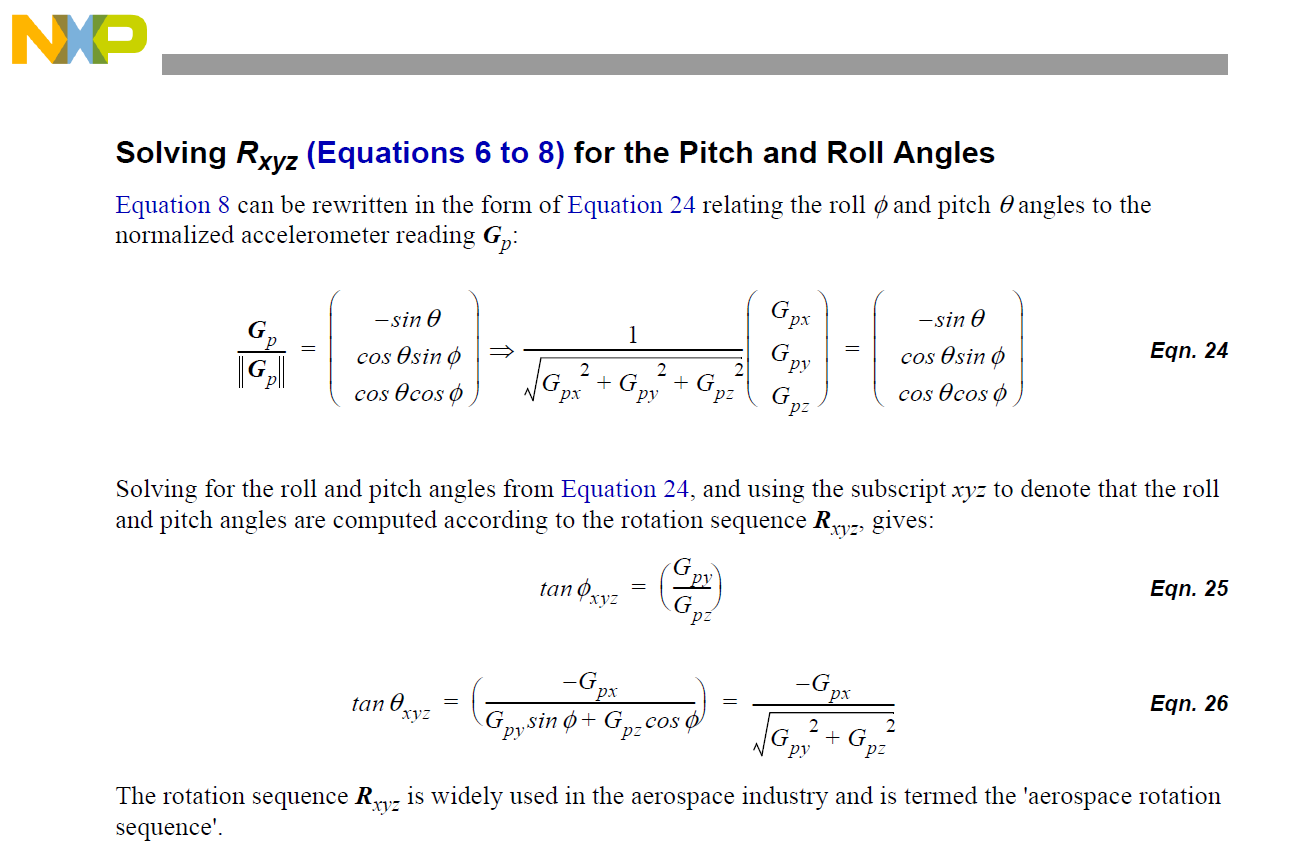
**Part A: - Compute Euler Angles Theta & Phi from the Accelerometers**

**Referencing STMicroelectronics Document DT0058:**





**Referencing Freescale Document Number AN3461:**



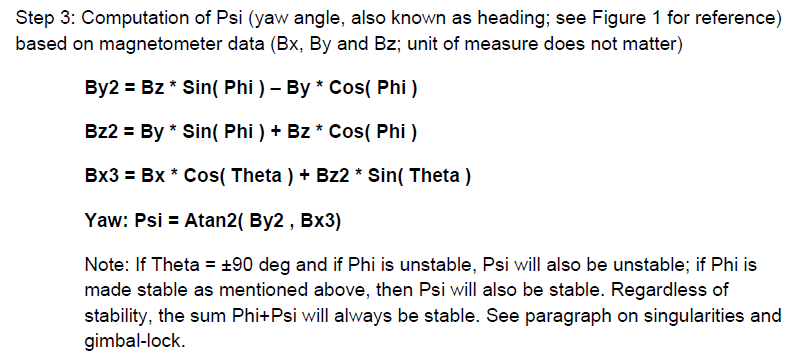
**Let’s first normalize the accelerometer reading to avoid issues with units.**

Note that: This is a good check.

Note in the computation for “theta” the formula differs from the equation in Freescale AN3461 because of the opposite polarity of “accel\_x” as provided by the MPU-9250 sensor.

**Part C: - Compute Euler Angle Psi from the Compass**

**Referencing STMicroelectronics Document DT0058:**



**Raw Data magnetic field readings from the MPU-9250:**

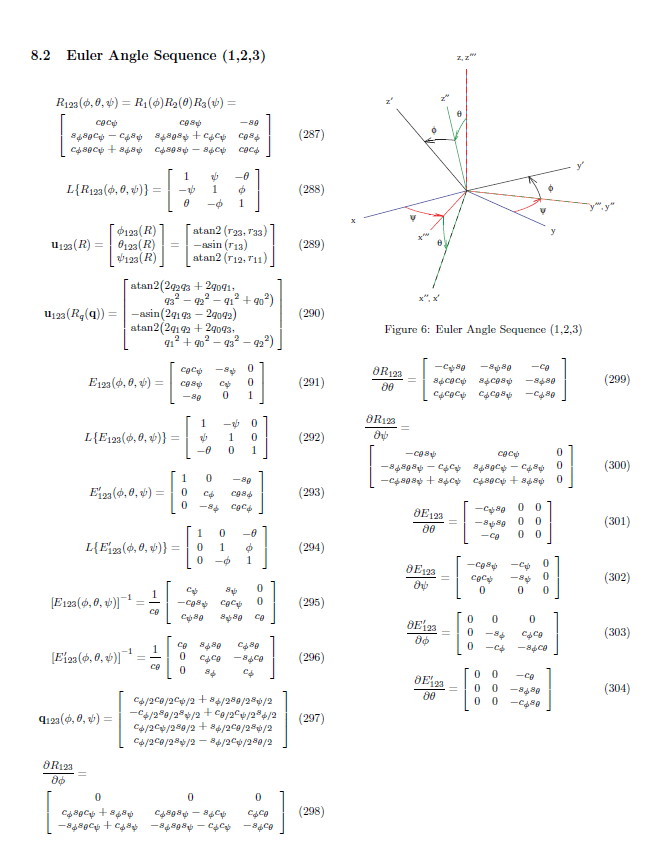
**Let’s first normalize the compass readings:**

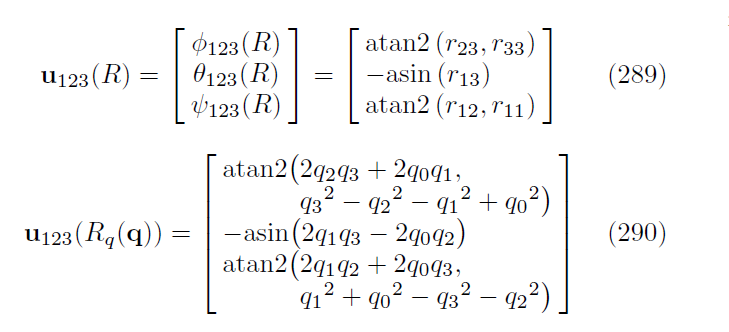
Note that: This is a good check.

Where θ and φ is the past final solution for θ and φ.

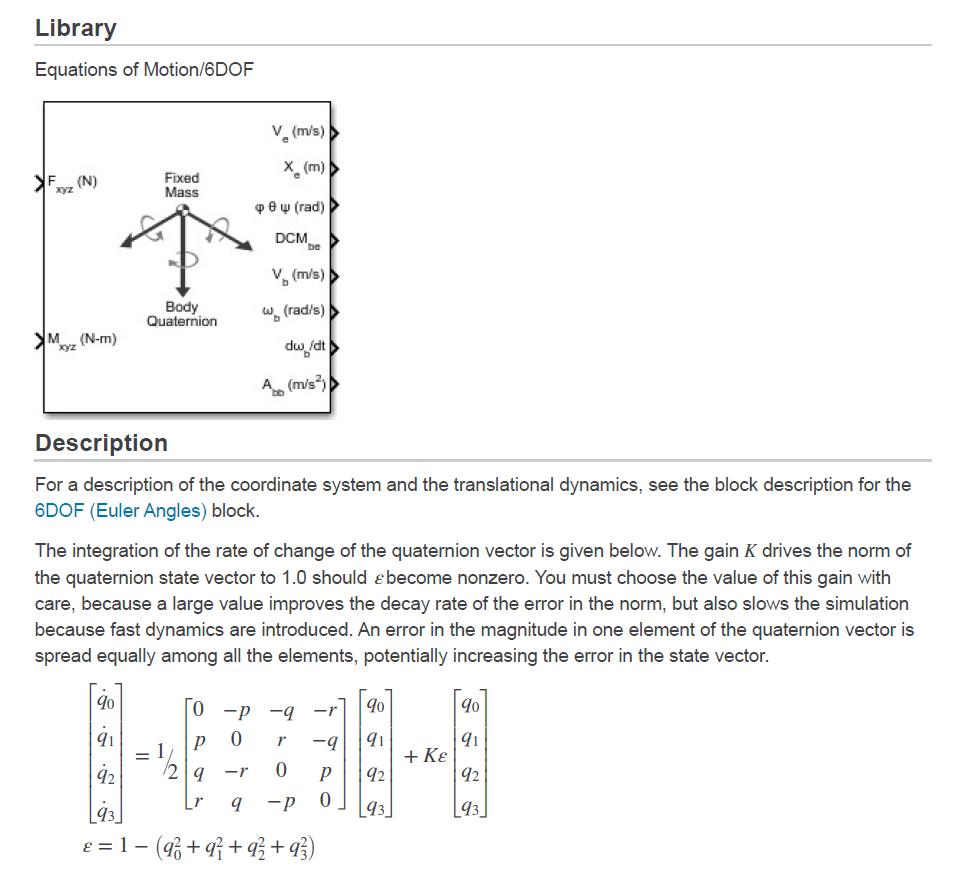
**Part Q: - Compute Euler Angles from Quaternions**

Referencing a paper from “Diebel” Dated 20 October 2006, “Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors” a relationship between the Euler Angles φ, θ, & ψ can be found in equation (290).





Referencing the Mathworks Documentation, a relationship between angular rates and Quaternions can be found.



With p, q, & r in radians/second taken from the raw data of the MPU-9250. It was found through testing that “ε” was always close to zero. Therefore, the feedback path Kε was not required.

Note that the subscripts are changed for convenience in the next set of equations where:

The raw sensor data for angular body rate polarity is adjusted for the MPU-9250 sensor:

The following Quaternion equations are computed as in the reference from “Mathworks” and the reference from “Diesel”:

Next, the short term Euler Angles are obtained from the Quaternion Equations:

Note that the ATAN2 function is used to avoid any singularities.

Next, Integrate the Quaternions with the ICs shown below:

ICs

is checked to make sure it is always close to “1”.

**Part F: - Use Complementary Filters to get a Final Solution for the Euler Angles.**

The short term solution for the Euler angles is computed from the Quaternions while the long term solution for the Euler angles is computed from the Accelerometers and Compass Sensors. So complementary filters are used to mix these two solutions into one final solution. This seems to work well for the low cost 9DOF MEMs sensors.





