**[Question] Can I set an initial orientation? #183**

FusionAhrsInitialise must be called the first function called for any instance of FusionAhrs. This is equivalent to an class constructor.

If you want to set the initial orientation then you must skip initialisation. The simplest way to do this is to perform a single update with a deltaTime value greater than INITIALISATION\_PERIOD. You can then call FusionAhrsSetQuaternion to set the orientation.

FusionAhrs ahrs;

FusionAhrsInitialise(&ahrs);

FusionAhrsUpdate(&ahrs, FUSION\_VECTOR\_ZERO, FUSION\_VECTOR\_ZERO, FUSION\_VECTOR\_ZERO, 4.0f); // 4 seconds is greater than INITIALISATION\_PERIOD

FusionAhrsSetQuaternion(&ahrs, insQuaternion);

**Proper way of calling FusionAhrsUpdate with accel, gyro and mag all having different update rates #170**

Apologies if this is obvious, but i get implausible results if i do this naively, and i can't find authoritative documentation on the matter.

i have an accelerometer sampling at 100Hz, a gyroscope at 400Hz and a magnetometer at about 150Hz without a common clock base, i.e. they have each their own free running oscilattors, but the samples come in with accurate timestamps.

Answer: You should call ***FusionAhrsUpdate*** for each new gyroscope measurement and use the most recent accelerometer and magnetometer measurements at that time. The ***deltaTime*** value should correspond to the gyroscope. This is the correct way to combine asynchronously sampled sensors. The errors associated with the phasing will be negligible.

**Integration of GPS data to estimate position and velocity #169**

Hello, me and some colleagues of mine are working on a Sensor Fusion project. Our setup includes an accelerometer, a gyroscope, a magnetometer and a GPS. We were able to use the first three sensors with the Madgwick AHRS library in C, and were starting to study a method to fuse also the GPS to the other sensors to have a better estimation of position and velocity, rather than using raw data from the GPS.

Searching online we found different resources saying that the Madgwick filter could be modified according to our use case, but none of it included meaningful papers and/or code. We were wondering if you could please pinpoint us to some useful resources, be it papers or libraries, even better if based on this very same repository that we found to be very clean and well documented, thank you very much.

Answer:  
I am not able to direct you to a complete solution. Fusion provides you with a measurement of earth acceleration. This is the second derivative of the latitude, longitude, and altitude expressed as an XYZ position. The main challenges of fusing GPS and IMU data are:

1. Utilising GPS heading reliably given that it is derived from the direction of travel and not the orientation of the subject.
2. Reliably exploiting the relationship between acceleration, velocity, and position given the exponential errors introduced by integration drift.

I suggest that you use IMU-Simulator to develop and test solutions. This simulator can generate complete 6 DOF kinematics from XYZ position and Euler angle CSV files. It also includes tools for plotting data and generating animations. For example, here is an animation of a simple simulation relevant to your application generated by 3d\_animation.py.

<https://private-user-images.githubusercontent.com/1748389/339275889-73d4f9fe-2aa3-4495-ae5f-1da1454370eb.gif?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..2Ij_9ef3O5DxC12w27CwxZXVcZX4hmFQq65uksV5-HU>

You may also be interested in the double\_integration.py example which uses Fusion with actual drone data to convert IMU measurements into 3D position. Note that tracking position in this way only works when simulating perfect IMU data and that this simple solution will never work in practise.

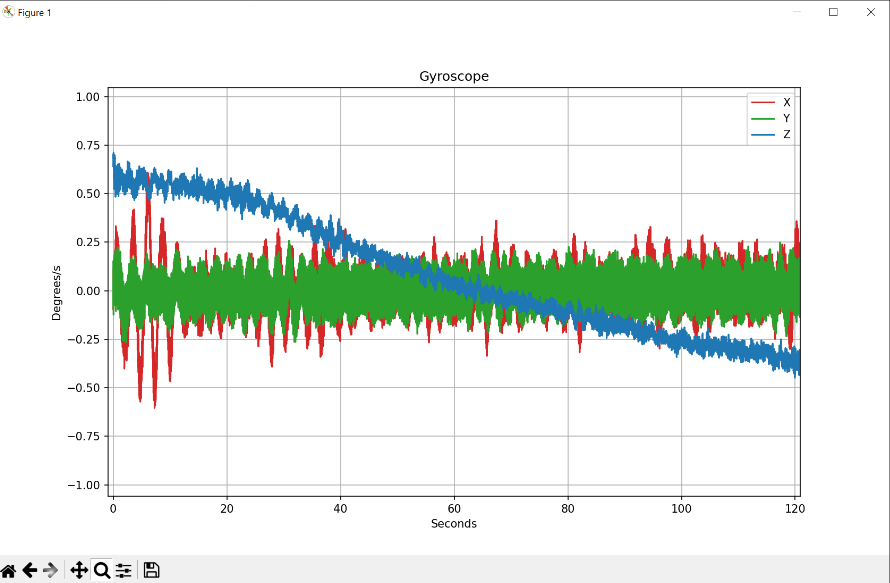
<https://private-user-images.githubusercontent.com/1748389/339288820-cd7f63dd-c719-43d1-9f57-2e8cd950dd78.gif?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..C9TbvdVwOtjimq6k-VG9swaIRAe1ksESM-EcNGs3iAw>

**Recommended sampling rate? #160**

An appropriate sample rate will depend on the application. If you are unsure then I suggest you start with 100 Hz. Less than 50 Hz will often cause problems, and more than 400 Hz will offer diminishing returns. The algorithm only utilises first-order feedback processes and so cannot become unstable.

**Gyroscope Offset Algorithm #155**

Thank you for sharing the data and describing the application. I have plotted the first 2 minutes of gyroscope measurements below.



The gyroscope measurements represent exceptionally smooth motion and so would not be compatible with the offset correction algorithm. This is not an issue and is not unexpected. Any x-io product that includes the offset correction algorithm also includes the option to disable it for precisely these situations. The AHRS algorithm already works well without the offset correction algorithm, as demonstrated by your original plot.

You are using magnetic\_rejection: float = 10. This enables the magnetic rejection feature with an error threshold of 10°. In the case of update\_external\_heading, this will mean that spurious GPS heading errors (>10°) will be rejected unless they persist for the recovery\_trigger\_period. If the GPS heading and AHRS heading disagree for the this period then the AHRS algorithm will immediately update to match the GPS heading. You may which to monitor the magnetic\_recovery flag for an application-level indication of this event.

It is not clear from your post what the numerical value of recovery\_trigger\_period is. The units of this parameter is samples. If the sample rate is 200 Hz and the intended period is 5 seconds, then the value of recovery\_trigger\_period must be 1000 = 200 Hz \* 5 seconds.

**How to use the magnetometer data with different sampling frequency from the inertial sensors? #151**

You should call update for each new gyroscope measurement. For many IMUs, including the x-IMU3, the magnetometer measurements will be out of phase or at a lower sample rate relative to the gyroscope. This is not a problem. For real-time systems, you should use the most recent magnetometer measurement for each update. For post processing, it is convenient to interpolate the magnetometer measurements so that all measurements vectors are of the same length. For example, this can be done in Python using scipy.interpolate.interp1d with kind="previous". More complex interpolation techniques (linear, quadratic, etc.) are unlikely to provide any practical benefit because the angular error associated with the measurement phrase error is typically extremely small.

**TEST data #143**

The Python examples in this repository include example data. There is also the Gait Tracking repository that uses Fusion, and IMU Simulator for synthetic data.

**Roll and pitch relative to device, yaw relative to earth #138**

Fusion uses the [ZYX Euler angles](https://github.com/xioTechnologies/Fusion/blob/main/Fusion/FusionMath.h#L466) sequence with [an axes convention](https://github.com/xioTechnologies/Fusion/blob/main/Fusion/FusionConvention.h#L20) selected in the settings, either NWD, ENU, or NED.

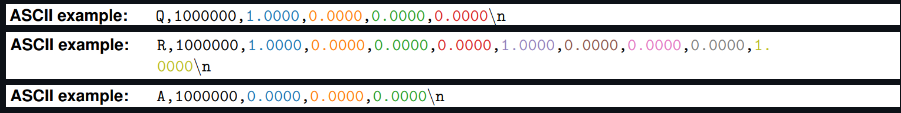
It sounds like your enquiry is not about Fusion and you just need clarification on what Euler angles actually are. I suggest you search online. There are countless [YouTube videos](https://www.youtube.com/results?search_query=euler+angles) that do a far better job of explaining Euler angles with animations than I could do in written comments.

**Ways to debug Fusion inputs/outputs? #123**

Fusion assigns roll as a rotation around the x-axis, and pitch as a rotation around the y-axis. Other libraries may switch these. If you Google image search [pitch roll yaw](https://www.google.com/search?q=pitch+roll+yaw) then the results suggest a 50/50 split with perhaps a slight favour for the convention used by Fusion.

The best way of validating your measurement of a 3D orientation is to view it in 3D, in real-time. That way you can move the physical object and confirm the on-screen image moves as expected.

I suggest you use the [x-IMU3 GUI](https://x-io.co.uk/x-imu3/) and modify your hardware to send either **Q**uaternion, **R**otation matrix, or Euler **A**ngles messages as shown below. These examples are taken from the user manual.

  
  
You only need to stream one of these messages continuously by either USB, serial, TCP, or UDP and you can view your measurements in real-time using the x-IMU3 GUI as shown below. No handshaking or other message types required. You can also load your own custom 3D model.



**what is the default IMU coordinate look like ? #117**

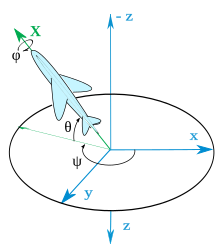
You have described your XYZ directions as: forwards, left, up. These directions follow the right-hand convention. If you are not using a magnetometer then axes conventions ENU and NWU will be functionality identical. The measurement of heading (yaw) will be relative to the direction the device was facing when it was switched on, and will slowly drift over time and with motion.

In your code you are passing sample\_rate\_ for the argument deltaTime. This is an error and you should instead pass 1.0f / sample\_rate\_ . I also suggest you add the gyroscope offset correction algorithm to your code as demonstrated by lines 24 and 52.

**earth acceleration direction convention\_enu #107**

ENU means East = X, North = Y, and Up = Z. The three letters correspond to XYZ, in order. The heading will determine how accelerometer measurements map onto the Earth's XY plane.

From Wikipedia, the free encyclopedia

[](https://en.wikipedia.org/wiki/File:Plane.svg)Heading, elevation and bank angles (Z-Y’-X’’) for an aircraft. The aircraft's pitch and yaw axes Y and Z are not shown, and its fixed reference frame xyz has been shifted backwards from its center of gravity (preserving angles) for clarity. Axes named according to the air norm [DIN](https://en.wikipedia.org/wiki/Deutsches_Institut_f%C3%BCr_Normung) 9300

In [ballistics](https://en.wikipedia.org/wiki/Ballistics) and [flight dynamics](https://en.wikipedia.org/wiki/Flight_dynamics), **axes conventions** are standardized ways of establishing the location and orientation of [coordinate axes](https://en.wikipedia.org/wiki/Coordinate_axes) for use as a [frame of reference](https://en.wikipedia.org/wiki/Frame_of_reference). Mobile objects are normally tracked from an external frame considered fixed. Other frames can be defined on those mobile objects to deal with relative positions for other objects. Finally, attitudes or [orientations](https://en.wikipedia.org/wiki/Orientation_(geometry)) can be described by a relationship between the external frame and the one defined over the mobile object.

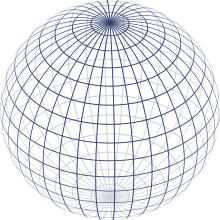
The orientation of a vehicle is normally referred to as *attitude*. It is described normally by the [orientation](https://en.wikipedia.org/wiki/Orientation_(geometry)) of a frame fixed in the body relative to a fixed reference frame. The attitude is described by *attitude coordinates*, and consists of at least three coordinates.[[1]](https://en.wikipedia.org/wiki/Axes_conventions#cite_note-Schaub-1)

While from a geometrical point of view the different methods to describe orientations are defined using only some reference frames, in engineering applications it is important also to describe how these frames are attached to the lab and the body in motion.

Due to the special importance of international conventions in air vehicles, several organizations have published standards to be followed. For example, German DIN has published the [DIN](https://en.wikipedia.org/wiki/Deutsches_Institut_f%C3%BCr_Normung) 9300 norm for aircraft[[2]](https://en.wikipedia.org/wiki/Axes_conventions" \l "cite_note-2) (adopted by ISO as [ISO 1151–2](https://en.wikipedia.org/wiki/List_of_ISO_standards_1%E2%80%931999#ISO_1000_%E2%80%93_ISO_1499):1985).

## Earth bounded axes conventions

[[edit](https://en.wikipedia.org/w/index.php?title=Axes_conventions&action=edit&section=1)]

[](https://en.wikipedia.org/wiki/File:Sphere_wireframe.svg)Representation of the Earth with parallels and meridians

### World reference frames: ENU and NED

[[edit](https://en.wikipedia.org/w/index.php?title=Axes_conventions&action=edit&section=2)]

*Main article:*[*Local tangent plane*](https://en.wikipedia.org/wiki/Local_tangent_plane)

Basically, as lab frame or reference frame, there are two kinds of conventions for the frames:

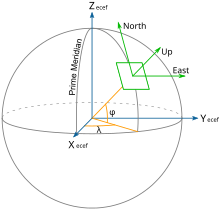
* East, North, Up (ENU), used in geography
* North, East, Down (NED), used specially in aerospace

This frame referenced w.r.t. Global Reference frames like Earth Center Earth Fixed (ECEF) non-inertial system.

#### World reference frames for attitude description

[[edit](https://en.wikipedia.org/w/index.php?title=Axes_conventions&action=edit&section=3)]

To establish a standard convention to describe attitudes, it is required to establish at least the axes of the reference system and the axes of the rigid body or vehicle. When an ambiguous notation system is used (such as [Euler angles](https://en.wikipedia.org/wiki/Euler_angles)) the convention used should also be stated. Nevertheless, most used notations (matrices and quaternions) are unambiguous.

[](https://en.wikipedia.org/wiki/File:ECEF_ENU_Longitude_Latitude_relationships.svg)Earth Centered Earth Fixed and East, North, Up coordinates.

[Tait–Bryan angles](https://en.wikipedia.org/wiki/Tait%E2%80%93Bryan_angles) are often used to describe a vehicle's attitude with respect to a chosen reference frame, though any other notation can be used. The positive *x*-axis in vehicles points always in the direction of movement. For positive *y*- and *z*-axis, we have to face two different conventions:

* In case of land vehicles like cars, tanks etc., which use the ENU-system (East-North-Up) as external reference (*World frame*), the vehicle's (body's) positive *y*- or pitch axis always points to its left, and the positive *z*- or yaw axis always points up. World frame's origin is fixed at the center of gravity of the vehicle.[[3]](https://en.wikipedia.org/wiki/Axes_conventions#cite_note-3)
* By contrast, in case of air and sea vehicles like submarines, ships, airplanes etc., which use the NED-system (North-East-Down) as external reference (*World frame*), the vehicle's (body's) positive *y*- or pitch axis always points to its right, and its positive *z*- or yaw axis always points down. World frame's origin is fixed at the center of gravity of the vehicle.
* Finally, in case of space vehicles like the [Space Shuttle](https://en.wikipedia.org/wiki/Space_Shuttle) etc., a modification of the latter convention is used, where the vehicle's (body's) positive *y*- or pitch axis again always points to its right, and its positive z- or yaw axis always points down, but “down” now may have two different meanings: If a so-called *local frame* is used as external reference, its positive z-axis points “down” to the center of the Earth as it does in case of the earlier mentioned NED-system,[[4]](https://en.wikipedia.org/wiki/Axes_conventions#cite_note-4) but if the *inertial frame* is used as reference, its positive z-axis will point now to the [north celestial pole](https://en.wikipedia.org/wiki/North_celestial_pole), and its positive x-axis to the Vernal [Equinox](https://en.wikipedia.org/wiki/Equinox)[[5]](https://en.wikipedia.org/wiki/Axes_conventions#cite_note-5) or some other reference meridian.

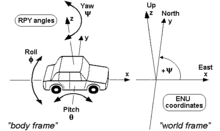
### Frames mounted on vehicles

[[edit](https://en.wikipedia.org/w/index.php?title=Axes_conventions&action=edit&section=4)]

Specially for aircraft, these frames do not need to agree with the earth-bound frames in the up-down line. It must be agreed what ENU and NED mean in this context.

#### Conventions for land vehicles

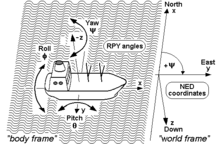
[[edit](https://en.wikipedia.org/w/index.php?title=Axes_conventions&action=edit&section=5)]

[](https://en.wikipedia.org/wiki/File:RPY_angles_of_cars.png)RPY angles of cars and other land vehicles

For land vehicles it is rare to describe their complete orientation, except when speaking about [electronic stability control](https://en.wikipedia.org/wiki/Electronic_stability_control) or [satellite navigation](https://en.wikipedia.org/wiki/Satellite_navigation). In this case, the convention is normally the one of the adjacent drawing, where RPY stands for [roll-pitch-yaw](https://en.wikipedia.org/wiki/Roll-pitch-yaw).

#### Conventions for sea vehicles

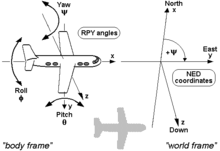
[[edit](https://en.wikipedia.org/w/index.php?title=Axes_conventions&action=edit&section=6)]

[](https://en.wikipedia.org/wiki/File:RPY_angles_of_ships.png)RPY angles of ships and other sea vehicles

As well as aircraft, the same terminology is used for the [motion of ships and boats](https://en.wikipedia.org/wiki/Ship_motions). Some words commonly used were introduced in maritime navigation. For example, the *yaw* angle or heading, has a [nautical](https://en.wikipedia.org/wiki/Navigation) origin, with the meaning of "bending out of the course". Etymologically, it is related with the verb 'to go'.[[6]](https://en.wikipedia.org/wiki/Axes_conventions#cite_note-6) It is related to the concept of [bearing](https://en.wikipedia.org/wiki/Bearing_(navigation)). It is typically assigned the shorthand notation *ψ*.[[7]](https://en.wikipedia.org/wiki/Axes_conventions#cite_note-7)

#### Conventions for aircraft local reference frames

[[edit](https://en.wikipedia.org/w/index.php?title=Axes_conventions&action=edit&section=7)]

[](https://en.wikipedia.org/wiki/File:RPY_angles_of_airplanes.png)RPY angles of airplanes and other air vehicles[](https://en.wikipedia.org/wiki/File:Roll_pitch_yaw_mnemonic.svg)Mnemonics to remember angle names

*Main article:*[*aircraft principal axes*](https://en.wikipedia.org/wiki/Aircraft_principal_axes)

Coordinates to describe an aircraft attitude (Heading, Elevation and Bank) are normally given relative to a reference control frame located in a control tower, and therefore ENU, relative to the position of the control tower on the earth surface.

Coordinates to describe observations made from an aircraft are normally given relative to its intrinsic axes, but normally using as positive the coordinate pointing downwards, where the interesting points are located. Therefore, they are normally NED.

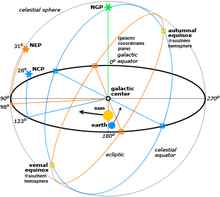
These axes are normally taken so that X axis is the longitudinal axis pointing ahead, Z axis is the vertical axis pointing downwards, and the Y axis is the lateral one, pointing in such a way that the frame is [right-handed](https://en.wikipedia.org/wiki/Right-hand_rule).

The *motion* of an aircraft is often described in terms of rotation about these axes, so rotation about the *X*-axis is called rolling, rotation about the *Y*-axis is called pitching, and rotation about the *Z*-axis is called yawing.

## Frames for space navigation

[[edit](https://en.wikipedia.org/w/index.php?title=Axes_conventions&action=edit&section=8)]

*Main article:*[*Celestial coordinate system*](https://en.wikipedia.org/wiki/Celestial_coordinate_system)

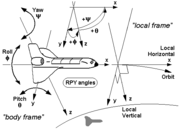
[](https://en.wikipedia.org/wiki/File:Celestial.png)Different reference systems for coordinates in space

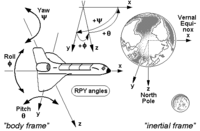
For satellites orbiting the Earth it is normal to use the [Equatorial coordinate system](https://en.wikipedia.org/wiki/Equatorial_coordinate_system). The projection of the Earth's equator onto the celestial sphere is called the [celestial equator](https://en.wikipedia.org/wiki/Celestial_equator). Similarly, the projections of the Earth's north and south geographic poles become the north and south [celestial poles](https://en.wikipedia.org/wiki/Celestial_pole), respectively.

Deep space satellites use other [Celestial coordinate system](https://en.wikipedia.org/wiki/Celestial_coordinate_system), like the [Ecliptic coordinate system](https://en.wikipedia.org/wiki/Ecliptic_coordinate_system).

### Local conventions for space ships as satellites

[[edit](https://en.wikipedia.org/w/index.php?title=Axes_conventions&action=edit&section=9)]

[](https://en.wikipedia.org/wiki/File:RPY_angles_of_spaceships_(local_frame).png)

[](https://en.wikipedia.org/wiki/File:RPY_angles_of_spaceships_(inertial_frame).png)

RPY angles of the Space Shuttle and other space vehicles, first using a local frame as reference and second using an inertial frame as reference.

If the goal is to keep the shuttle during its orbits in a constant attitude with respect to the sky, e.g. in order to perform certain astronomical observations, the preferred reference is the *inertial frame*, and the RPY angle vector (0|0|0) describes an attitude then, where the shuttle's wings are kept permanently parallel to the Earth's equator, its nose points permanently to the vernal [equinox](https://en.wikipedia.org/wiki/Equinox), and its belly towards the northern [polar star](https://en.wikipedia.org/wiki/Polar_star) (see picture). (Note that rockets and missiles more commonly follow the conventions for aircraft where the RPY angle vector (0|0|0) points north, rather than toward the vernal equinox).

On the other hand, if the goal is to keep the shuttle during its orbits in a constant attitude with respect to the surface of the Earth, the preferred reference will be the *local frame*, with the RPY angle vector (0|0|0) describing an attitude where the shuttle's wings are parallel to the Earth's surface, its nose points to its heading, and its belly down towards the centre of the Earth (see picture).

## Frames used to describe attitudes

[[edit](https://en.wikipedia.org/w/index.php?title=Axes_conventions&action=edit&section=10)]

Normally the frames used to describe a vehicle's local observations are the same frames used to describe its attitude with respect to the ground tracking stations. i.e. if an ENU frame is used in a tracking station, also ENU frames are used onboard and these frames are also used to refer local observations.

An important case in which this does not apply is aircraft. Aircraft observations are performed downwards and therefore normally NED axes convention applies. Nevertheless, when attitudes with respect to ground stations are given, a relationship between the local earth-bound frame and the onboard ENU frame is used.

**Best delta time + calibration #58**

Ideally, deltaTime should be calculated as the difference between two timestamps where each timestamp is obtained using the gyroscope data ready interrupt. This would achieve timing calculation errors inline with your system clock error. For example, 0.002% (20 ppm). Alternatively, could use the nominal sample period of the sensor as a constant value for deltaTime. The [LSM6DSOX](https://www.st.com/en/mems-and-sensors/lsm6dsox.html) sample clock error appears to not be specified in the datasheet but it would likely be around 0.5%. This is significantly better than the 12.5% (80 ms ±10 ms) you are currently expecting.

It is not clear from your description if you system is guaranteed to process every gyroscope sample, or if some samples may be discarded due to the jitter in task scheduling. It is essential that you process every gyroscope sample.

If you are unable to provide a specific value for the misalignment argument of FusionCalibrationInertial then you should use FUSION\_IDENTITY\_MATRIX. If you are unable to provide a specific value for sensitivity then you should use FUSION\_VECTOR\_ONES. For example, if you only wanted to apply gyroscopeOffset then you would use the following expression.

gyroscope = FusionCalibrationInertial(gyroscope, FUSION\_IDENTITY\_MATRIX , FUSION\_VECTOR\_ONES, gyroscopeOffset);

Note that this expression is equivalent to gyroscope = FusionVectorSubtract(gyroscope, gyroscopeOffset); and would typically compile to the same machine code due to the use of the definitions FUSION\_IDENTITY\_MATRIX and FUSION\_VECTOR\_ONES.

FusionCalibrationInertial applies fixed calibration parameters, FusionOffsetUpdate is the gyroscope offset correction algorithm. The following is copied from the [README](https://github.com/xioTechnologies/Fusion/blob/main/README.md).

"The gyroscope offset correction algorithm provides run-time calibration of the gyroscope offset to compensate for variations in temperature and fine-tune existing offset calibration that may already be in place. This algorithm should be used in conjunction with the AHRS algorithm to achieve best performance."