# Gyroscope Calibration Guide

The following guide shows how one calibrates the gyroscopes for the AR Drone 2.0. This guide assumes

1. You’ve installed the support package
2. Tested out external mode and everything works fine

**What are we calibrating?**



In the above plots, we have raw gyroscope values of the AR Drone at rest and it’s integrated signal in the second plot. Integration of the rate gyro will tell us how much angular displacement has taken place. Since there is an offset in the sensor reading even with the drone at rest, this results in the attitude to drift to 4000 degrees despite the drone staying absolutely stationary. This is obviously a problem that needs to be addressed.

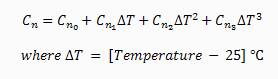
The goal of this calibration is to remove bias from the rate gyro such that instead of reading a value of 60 such as in this example it should be close to zero leading to lower drift.

The biggest challenge here is that the gyroscope bias Is not constant and will change based on numerous factors. From literature, the bias is mainly dependent on temperature, since the sensor is MEMs and electrical in nature, it will suffer from thermal noise.

Luckily, the AR Drone navigation data bus provides a temperature sensor reading. With sufficient data, one can correlate the bias with change in temperature.

One can map this relation with a simple polynomial, as discussed here in literature

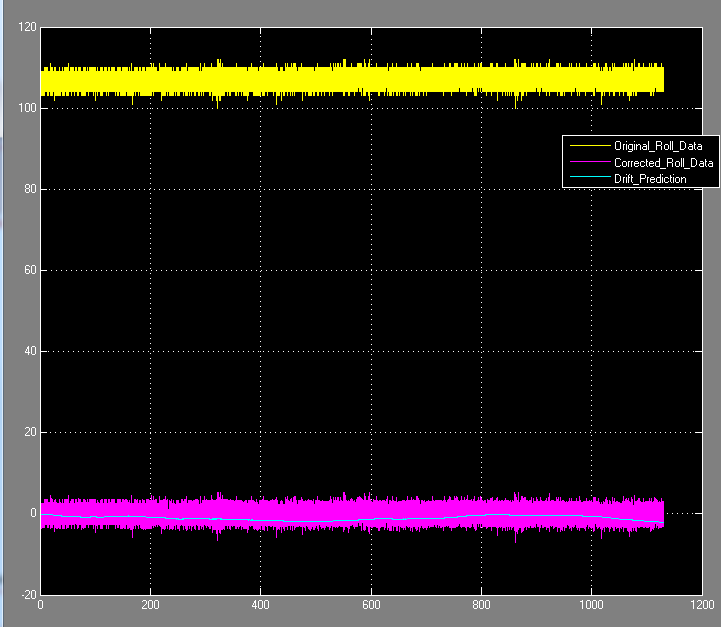
<http://www.vectornav.com/support/library/gyroscope>



For simplicity, we’re only going to consider a linear y = mx + b calibration although the model has been setup for a quadratic term.

**Steps to calibrate:**

1. Run the model TempMeasure.slx – this will collect data from the gyroscope and temperature measurements. Ensure the drone is completely stationary at this time, record for about 1000 seconds. The model should be configured for this already.
2. Run the TempCorrect\_OfflineTest\_Automated.slx model to examine the data offline and apply filtering on the data to average out the response. The filtering is necessary to ensure good curve fitting which is done in the next step
3. Temperature\_Testing\_Automated\_Script.m can be run cell-by-cell now, which will automatically curve fit the change in temperature vs change in drift. The script will save data to a MAT file which can then be retrieved later
4. Run the simulation again and notice how the purple line (corrected version) now hovers about 0 and the integration drift is reduced dramaticlly



1. Update the data dictionary as needed with the proper gain and offset values around line 44 in the AR\_DRONE\_SCRIPT.m file for the main hover model

Mag\_Calib\_Offset\_Vector.Gain\_X = -23;

Mag\_Calib\_Offset\_Vector.Gain\_Y = 4.3;

Mag\_Calib\_Offset\_Vector.Gain\_Z = 1.44;

Mag\_Calib\_Gain\_Vector.Gain\_X = 187;

Mag\_Calib\_Gain\_Vector.Gain\_Y = 163;

Mag\_Calib\_Gain\_Vector.Gain\_Z = 139;

% %Parameters for polynomial fit of gyro drift versus temperature

% P1 = quadratic gain, P2 = multiply gain, P3 = offset

Gyro\_Correction\_Roll.P1 = -0.004462 ;

Gyro\_Correction\_Roll.P2 = 1.144 ;

Gyro\_Correction\_Roll.Offset = -67.5100;

Gyro\_Correction\_Pitch.P1 = 0.009828;

Gyro\_Correction\_Pitch.P2 = 0.4486 ;

Gyro\_Correction\_Pitch.Offset = 4.961;

%yaw gains taken from recent experiment, see file

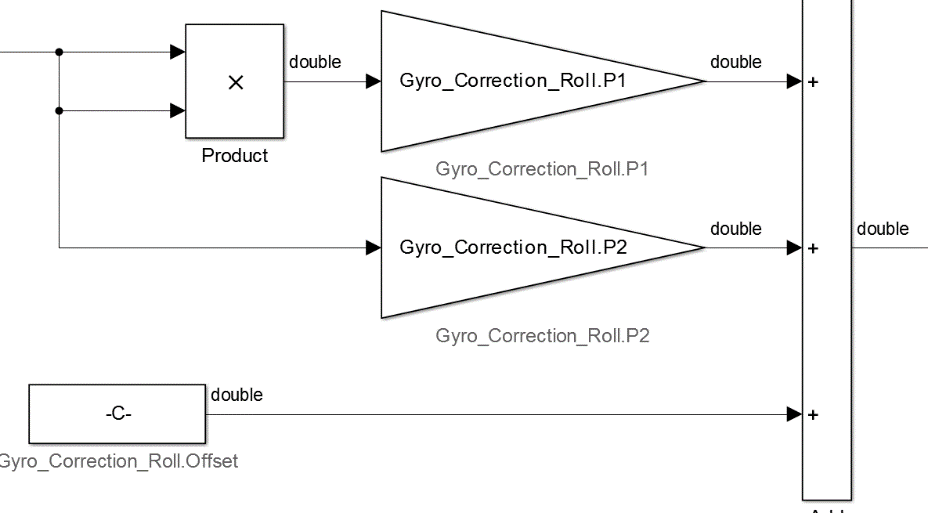
%'GyroDrift\_OffsetCorrections\_II.mat'

Gyro\_Correction\_Yaw.P1 = -0.0070;

Gyro\_Correction\_Yaw.P2 = 0.9867;

Gyro\_Correction\_Yaw.Offset = -35.2100;

Note that the automated procedure will do a linear y=xm+b fit and not have any quadratic terms. To cover a wider temperature range you can try fitting to a polynomial. The calibration bus object has been configured for this to include this term. This is shown here inside the offset subsystem



By default, P1 (quadratic term) is set to zero, but feel free to change this if you have done any curve fits for the quadratic term