

Unit-to-Unit Variation and On-Board Residual Hard Iron Error Correction

A project log for [MAX32660 Motion Co-Processor](#)

Super-small, ultra-low-power 96 MHz Cortex

M4F co-processor manages sensors and

processes data allowing the MCU host attend to other things.



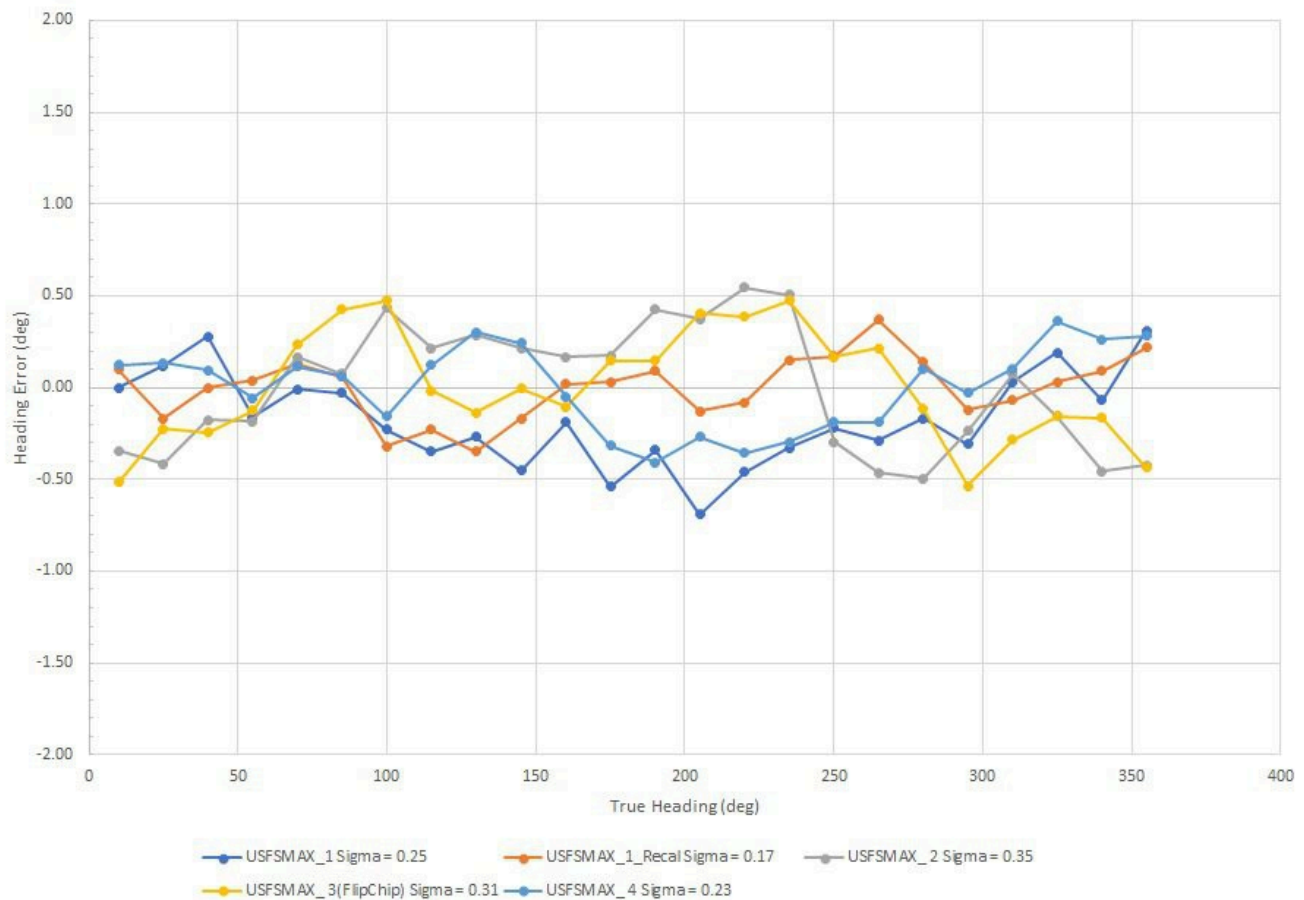
[Greg Tomasch](#) • 12/20/2019 at 00:44 • [1 Comment](#)

Unit-to-Unit Heading Accuracy Variation

In my last log entry I mentioned the need to characterize the unit-to-unit variation of heading accuracy among professionally manufactured MAX32660 motion co-processor boards. We are still waiting for delivery of our initial production runs, so in the mean time I decided to characterize the four prototype units I had on-hand. I used the same precision goniometer, calibration method and experimental procedures described in my previous log entry. The plot below shows heading error as a function of heading.

Results from a total of four units are included in this plot with two data sets from "Unit 1" representing two different trials of the calibration method. **The RMS heading error ranged between 0.17 and 0.35 degrees.** If anything we would expect professionally manufactured units to have more precise location/alignment of the MEMS sensor chips on the circuit board than these hand-built prototypes. So if placement of the accel/gyro and magnetometer chips effects post-calibration heading accuracy, we might potentially see less unit-to-unit heading accuracy variation on production units than what I have measured here. However, the nature of the 24-point tumble calibration [described earlier](#) should robustly correct accel/gyro/mag misalignment. So I would expect these results to be a representative sample of the MAX3260 motion co-processor's heading accuracy capability. I will perform similar measurements on a sample from our first run of production units when they arrive early in 2020.

USFSMAX Heading Error as a Function of True Heading: Unit-to-Unit Comparison (Prototype)



On-Board Real-Time Hard Iron Error Correction

Throughout the slate of heading error characterizations I have performed, I noticed that heading error performance degrades slightly when the magnetic environment differs from the exact calibration conditions. This effect is not gross but is more like degrading from ~ 0.26 to ~ 1 degree RMS heading accuracy. This illustrates a typical complaint regarding IMU calibration: The stellar results generated during calibration on the test bench are seldom translated to the real performance in the field. Analysis of the heading error plots showed that the additional error is hard-iron-like in that it has a **period of 360 degrees**. This means the magnetometer response surface is still spherical but is displaced from the origin.

The heading error results shown above indicate that the suite of MEMS sensors we are using (**LSM6DSM** and **LIS2MDL**) are stable enough to deliver ~ 0.26 degree RMS heading accuracy... So degrading to ~ 1 degree by incidental changes in the IMU's local environment should be correctable, especially since the additional error takes the form of a hard iron offset (the easiest to estimate and correct). The **MotionCal (Freescale) magnetic calibration algorithm** embedded in the MAX32660 is quite capable of dealing with this additional incidental hard iron error but it is too computationally intensive to run in parallel with the main AHRS fusion filter in real time.

Instead of just saying, "Oh well, such is life..." I decided to investigate the possibility of a simple, computationally efficient method of fitting a sphere to the locus of run-time (M_x , M_y , M_z) data points and extract estimates of the residual hard iron offsets. I found [this reference](#) which presents a practical embodiment of a method for geometrically fitting the "best sphere" through a 3D ensemble of data points. Upon examination, I particularly liked this method because 1) It is analytic and does not need to be solved iteratively and 2) Adding data to the point ensemble to be fitted only results in using more memory *without increasing the computational load on co-processor MCU*. The center point and radius of the best-fit sphere are calculated from a set of running sums that are cubic in X and Y . Executing a spherical fit after adding a new (M_x , M_y , M_z) data point consists of:

- Calculating the necessary cubic and quadratic terms from the M_x , M_y , M_z data
- Adding new terms to the running cubic, quadratic and linear term sums
- Using the running sums to calculate the new center point and radius estimates by simple algebraic manipulations

So the total computational load is fairly light and is the same for the first and N th data points.

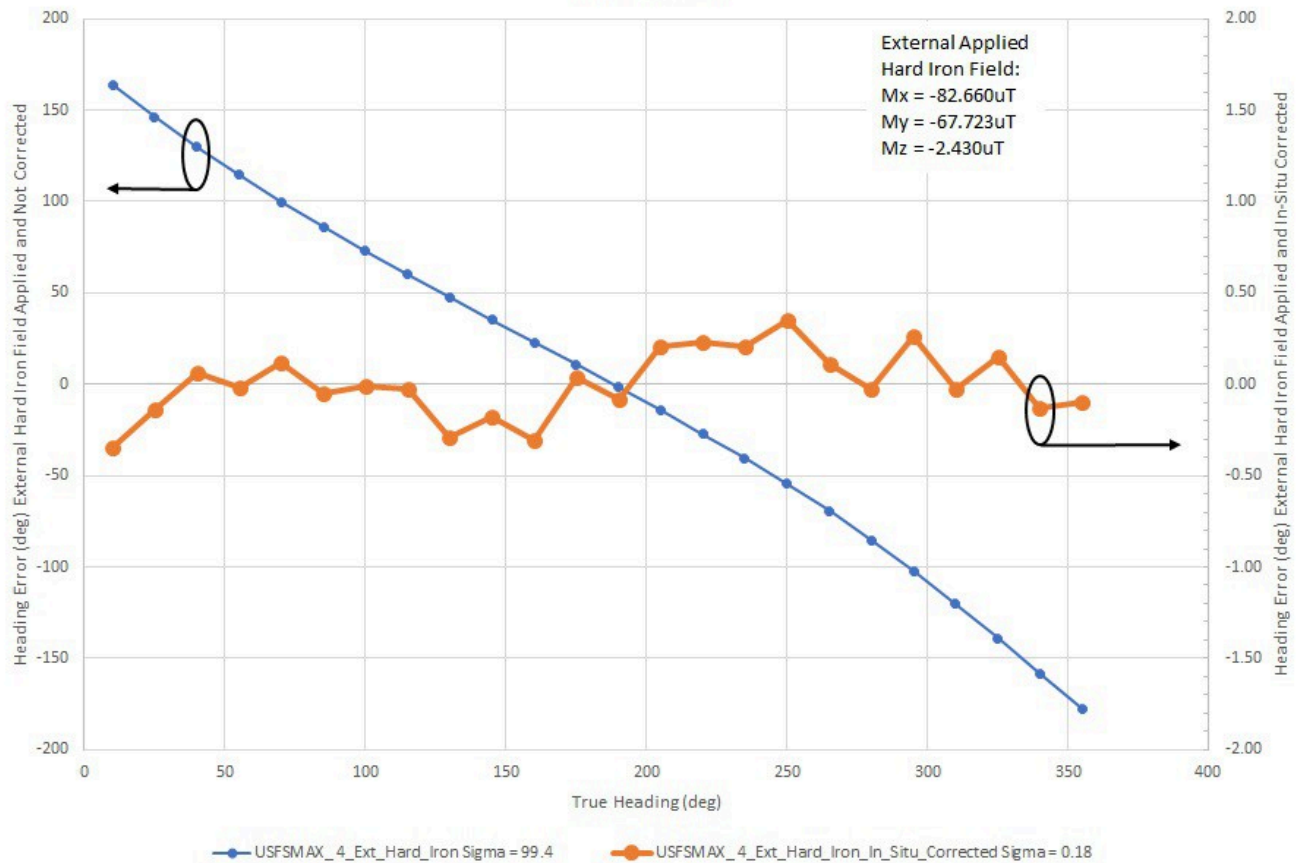
I tested the efficacy of this approach by:

- Programming the analytic sphere-fitting method on the host MCU
- Subtracting the sphere-fit center point coordinates from each (M_x , M_y , M_z) data point
- Programming a duplicate AHRS fusion filter on the host MCU to reproduce the Euler angles *with the hard-iron-corrected (M_x , M_y , M_z) data*

Using this method I can generate the USFS max heading solution under the influence of a stray hard iron field and simultaneously simulate the sphere-fit-corrected version. Initial tests showed that the real-time sphere fit method is quite capable of neutralizing incidental hard iron effects. To prove the point, I purposely attached a small rare-earth magnet to the USFSMAX calibration/test fixture to impose a gross hard iron offset and then:

- Tumbled the test fixture briefly in three-space to populate sphere fit arrays with valid data and generate the hard iron corrections *in-situ*
- Characterized the heading-dependent heading error curves for the corrected and uncorrected cases simultaneously

USFSMAX Heading Error as a Function of True Heading: External Hard Iron Field Applied and In-Situ Corrected



The applied hard iron offsets estimated from the spherical fit were $M_x = 82.660\text{uT}$, $M_y = -67.723\text{uT}$ and $M_z = -2.430\text{uT}$. These components are actually quite large, as the geomagnetic field is nominally $\sim 50\text{uT}$. The blue curve (Left-hand axis) shows the heading error of the uncorrected USFSMAX motion co-processor AHRS solution under the influence of the applied hard iron offsets. The hard iron interference is so bad that the indicated heading is nearly constant, only ranging between 141.5 and 167.1 degrees in a full 360 degree rotation. **The corresponding RMS heading error was 99.4 degrees.** The bold orange curve (Right-hand axis) shows the simulated *corrected* AHRS solution generated simultaneously on the host MCU. **The RMS heading accuracy recovered to 0.18 degrees**, essentially identical to the immediate post-calibration value.

I believe this is an important result. This real-time, *in-situ* method of fitting a sphere to the (M_x , M_y , M_z) data ensemble allows us to neutralize the degradation in heading accuracy performance that often occurs between calibration on the test bench and deployment out in the field. My next step will be to program this analytic sphere fitting method and associated infrastructure into the USFSMAX standard firmware. There will be a significant amount of experimentation required to learn how to best manage/filter which points are included into the sphere-fitting (M_x , M_y , M_z) data ensemble... And when to clear the data buffers and start over. But these initial results show there is a straightforward path to follow.

[Previous Log](#)

MAX32660 Motion Coprocessor - Initial Results

11/27/2019 at 00:29 • 5 comments

[Next Log](#)

MAX32660 Motion Coprocessor - MMC5983MA Low Noise Magnetometer Results

08/11/2020 at 23:51 • 4 comments

DISCUSSIONS

Log In or become a member to leave your comment

[Log In/Sign up to comment](#)



Alfredo Muniz wrote 02/05/2020 at 17:39

Hi Kris and Greg, nice work on the initial results. The project looks like it will turn out great!

I noticed that you haven't mentioned anything about dealing with temperature. I was wondering if in your testing you found it to be small enough to be ignored or if you need to "warm-up" the sensor before using and calibrating to get consistent readings?

↑ Going up?

[About Us](#)

[Contact Hackaday.io](#)

[Give Feedback](#)

[Terms of Use](#)

[Privacy Policy](#)

[Hackaday API](#)

© 2024 Hackaday