MAX32660 Motion Coprocessor - Initial Results

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 • 11/27/2019 at 00:29 • 5 Comments

Sensor Calibration and Performance Testing

Calibration

Many people seem to think that good orientation estimation results are primarily due to some "Special sauce" contained in the sensor fusion algorithm. In fact, few consider there to be any significant differences between the various MEMS sensor products offered by different manufacturers. The reality of it is that there is no such thing as an algorithm that can correct poor quality accelerometer, gyroscope and magnetometer data to give excellent orientation estimation results. We have shown that once the non-ideal behavior of the accelerometers and magnetometers have been neutralized *through effective sensor calibration*, any effect of the particular fusion algorithm on orientation estimation accuracy is secondary. Consequently, in the development of the MAX32660 motion co-processor I focused on making firmware infrastructure that can:

- Perform sophisticated sensor calibrations using embedded routines and store the results in the co-processor
- Reduce run-time implementation of the various sensor calibrations to simple, calculationally efficient forms

The need to embed complicated calibration functions and perform a significant volume of additional floating point calculations figured prominently into the selection of the MAX32660 for an advanced motion co-processor; it has a lot of memory and floating point processor power in a small package.

A short discussion of what constitutes effective accelerometer and magnetometer calibration would be helpful at this point. We will consider the magnetometers first. For a perfectly calibrated three-axis magnetometer, the Mx, My, Mz response surface is a perfect sphere centered at the origin. "Hard iron" errors occur when there is a static magnetic field in the *magnetometer's reference frame* that displaces the response surface from the origin by a constant vector. "Soft iron" errors result from magnetic flux divergence and are manifested as the response surface being distorted from a sphere to an ellipsoid. Both types of error can be corrected by fitting an ellipsoidal surface to uncorrected 3D magnetometer data and then transforming back to a sphere with no offset vector. Freescale Semiconductor (now NXP) published a good C library to fit the uncorrected magnetometer data and generate both soft and hard iron calibration corrections. This library is included in the "MotionCal" application distributed by Paul Stoffregen of PJRC. I started with the code in Paul Stoffregen's GitHub repository and re-cast Freescale's solver for embedding into the motion co-processor. This works quite well and consistently results in residual fit errors of <= 1.5%.

Effective accelerometer calibration is also essential to achieve accurate estimates for heading, as well as pitch and roll. There are numerous methods for correcting accelerometer imperfections including:

- Bias (zero-point offset)
- Scale error
- Non-orthonormality of the x, y, z sense axes

Of all the potential accelerometer calibration techniques in the literature, the "Tumble" method seemed the most suitable for the purposes at hand. By collecting raw ax, ay, az data in a number of orthogonal orientations, the accelerometer errors listed above can be quantified and calibration corrections accurately estimated.

The final category of sensor non-ideality that needs to be addressed is relative rotation between magnetometer and the accelerometer (inertial) sense axes. This turns out to be important because the pitch and roll angle estimates are derived from the 3D accelerometer data. Pitch and roll are used to resolve the 3D magnetometer components from the sensor reference (body) frame into the horizontal plane of the world reference frame for heading estimation. This is true for quaternion-based fusion filters as well as explicit direction cosine rotation-matrix-based projections of the magnetic field vector. Fundamentally, relative rotation of the inertial and magnetometer sense axes corrupts the "Tilt compensation" of the magnetometer, increasing the heading-dependent heading error.

I did a fairly extensive review of the literature and determined that there was a simple version of the accelerometer tumble calibration method presented in this reference. Specifically, the accelerometer calibration is performed by enclosing the sensor board in a fixture with six orthonormal surfaces and collecting data with the normal vector of each face parallel to gravity. The essential outcome of this method is that the correction offsets and 3x3 correction matrix effectively align the inertial axes parallel to the face normal vectors of the calibration fixture. By extending the technique to 24 positions, it is possible to simultaneously align the magnetometer

sense axes to the calibration fixture face normal vectors as well... *Effectively aligning the magnetometer and inertial sense axes to each other.*

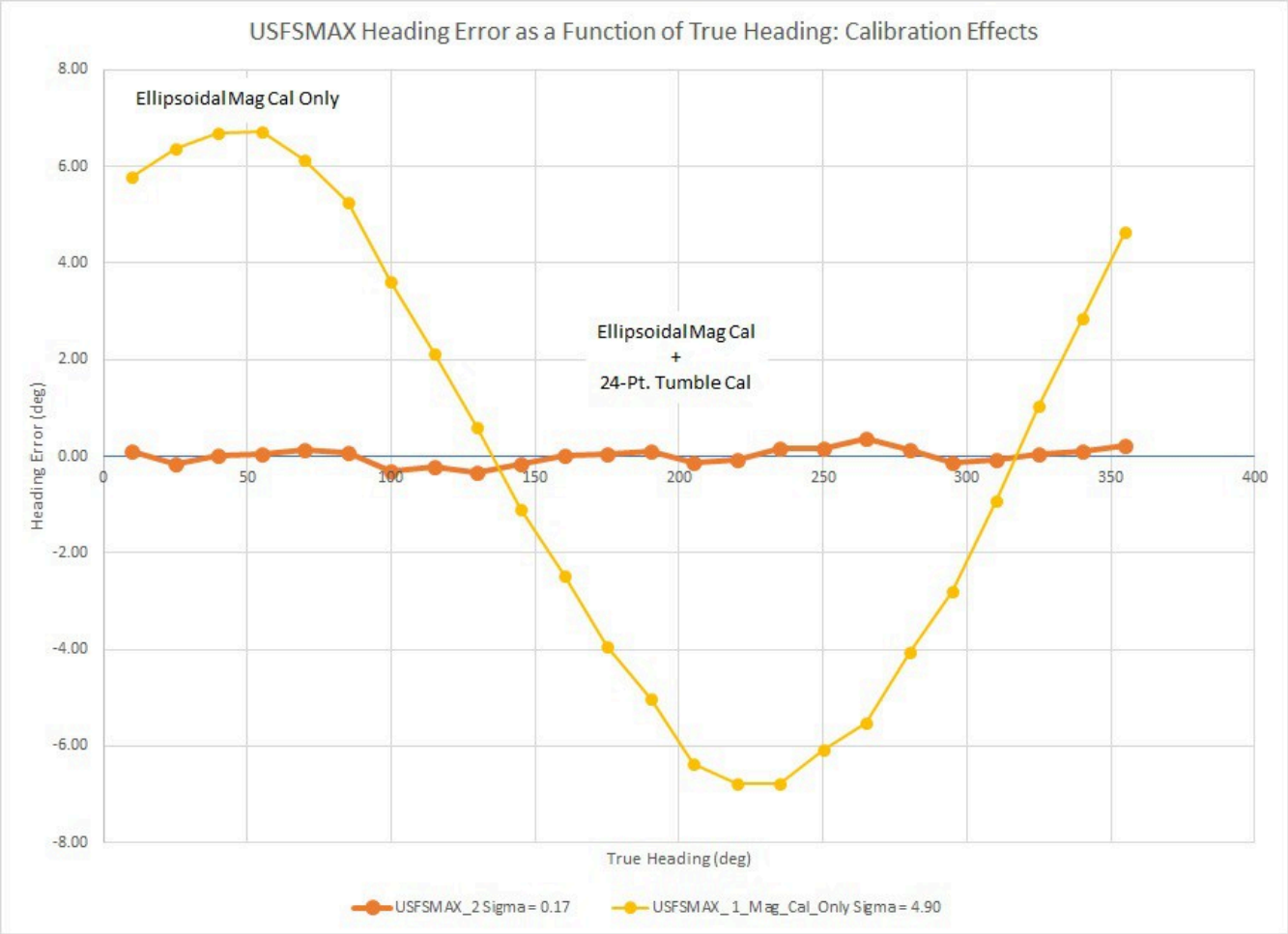
Weighing all of these factors carefully, the sensor calibration strategy I selected to program into the MAX32660 motion co-processor includes:

- Gyroscope bias estimation (startup, board at rest in any orientation)
    - Subtract biases from each data point at run time
- Embedded Freescale ellipsoidal magnetometer calibration (manually rotate board randomly in 3-space)
    - Collect raw magnetometer data during calibration
    - Subtract biases from and apply 3x3 correction matrix to each magnetometer 3D data point at run time
- Enhanced 24-point "Tumble" calibration (align board using orthonormal calibration fixture)
    - Apply ellipsoidal calibration corrections to 3D magnetometer data during calibration data collection
    - Collect raw accelerometer data during calibration
    - Apply magnetometer offsets and 3x3 correction matrix "On top of" the ellipsoidal calibration corrections at run time
    - Subtract biases from and apply 3x3 correction matrix to each accelerometer data point at run time

Performance Testing

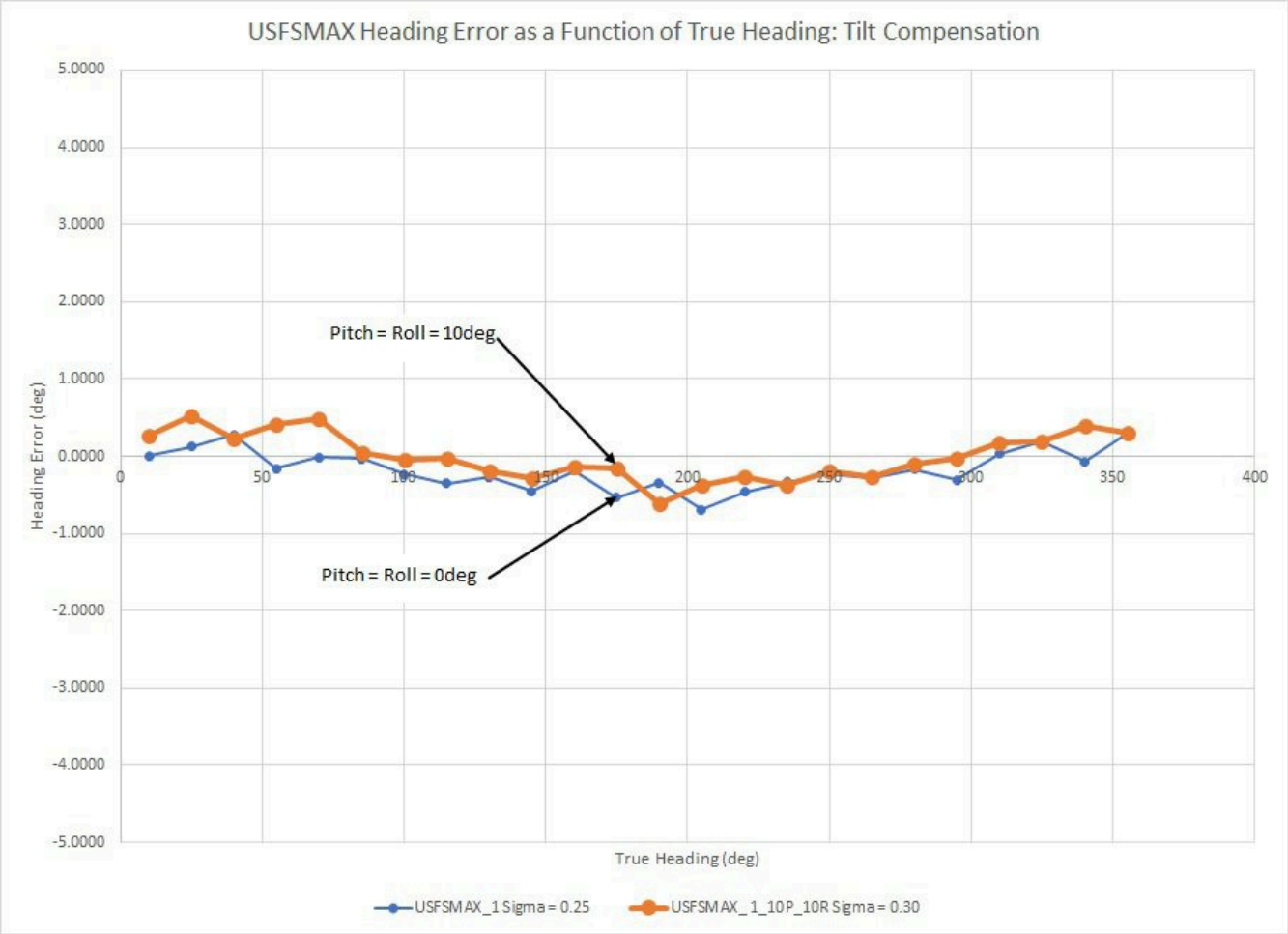
I measured heading accuracy using a precision three-axis goniometer to accurately align the motion co-processor board and provide the "Ground truth" orientation. The specific instruments and fixtures I built for the 24-point tumble calibration and motion co-processor characterization will be the subject of an upcoming Hackaday project.

Initially, I aligned the motion co-processor board level to the goniometer stage using a precision spirit level and measured heading accuracy at pitch = roll = 0deg (level attitude). The figure below shows the individual effect of the ellipsoidal magnetometer calibration on heading error and the overall effect when followed by the 24-point tumble calibration. The deviation of the indicated heading from the actual heading (imposed by the goniometer) is plotted as a function of the actual heading. The heading error curve for the "Ellipsoidal Mag Cal Only" case is strikingly sinusoidal with a period of 360deg. This is characteristic of uncompensated hard iron error, magnetometer tilt correction error or both.



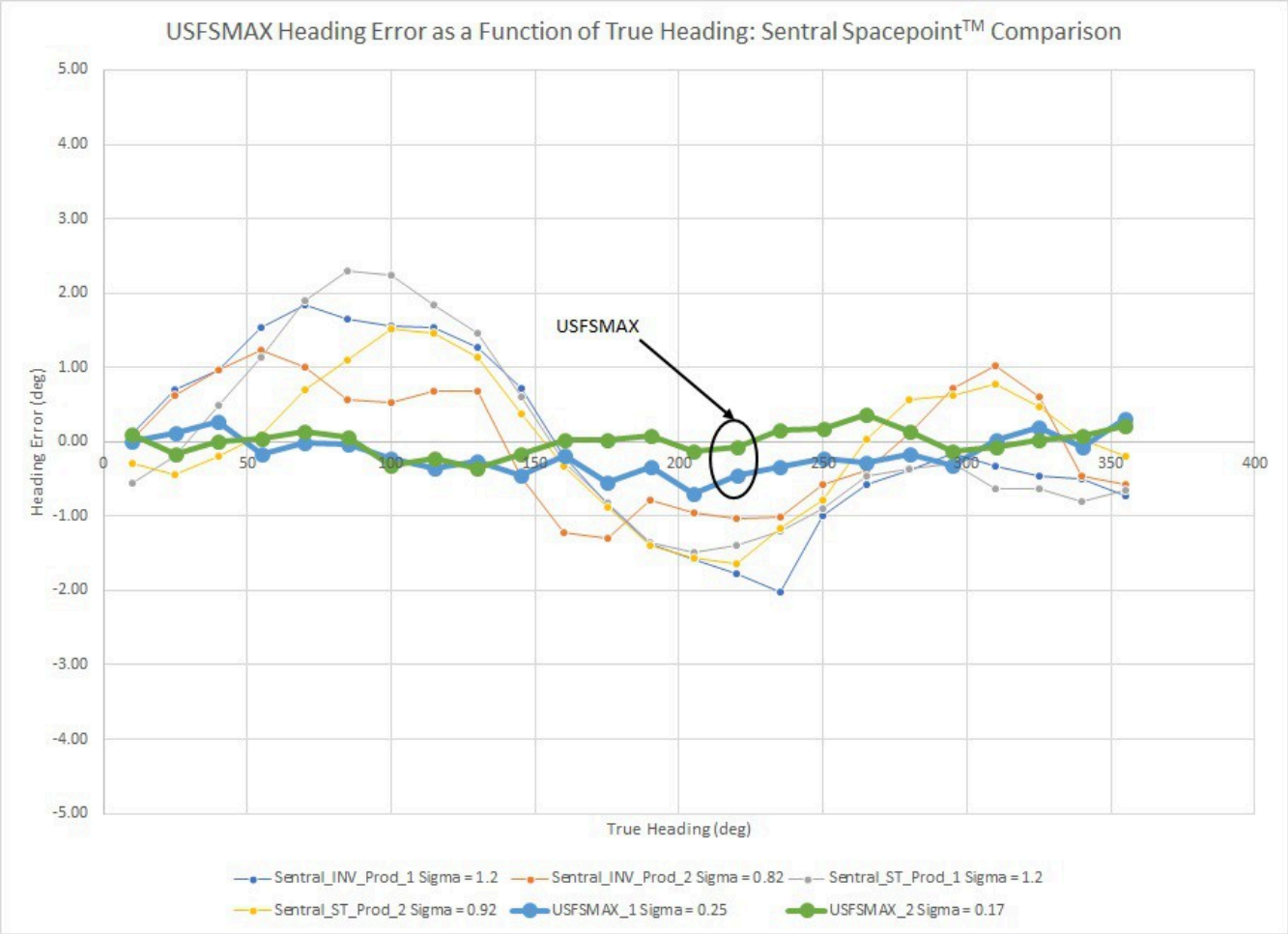
Since the ellipsoidal magnetometer calibration is effective at removing hard iron biases and uncalibrated accelerometers will certainly induce pitch and roll estimation errors, tilt correction error is much more likely. Application of the 24-point tumble calibration after the ellipsoidal calibration (bold curve) totally neutralizes the residual sinusoidal heading error. This strongly suggests that indeed the +/- ~6deg sinusoidal heading error is due to erroneous tilt correction from relative rotation between the magnetic sense and inertial axes. The 24-point tumble calibration effectively aligns the magnetometer and accelerometer responses and the 360deg-period heading error goes away.

To further verify this assertion, I imposed tilts of pitch = roll = 10deg and re-measured the heading error.



We typically use the root-mean-square (RMS) of the heading error as a figure of merit for comparing various attitude estimation solutions. In this case, the RMS heading error at level attitude is 0.25deg. After imposing a resolved tilt of ~14.4deg (bold curve), the RMS heading error was 0.3deg, basically unchanged. So not only does the 24-point tumble calibration address the large residual sinusoidal error at level attitude, the results hold up at a significant tilt. From this I conclude that the calibration regimen programmed into the MAX32660 motion co-processor effectively addresses hard iron error, soft iron error and magnetometer tilt compensation.

Finally, I performed the full calibration regimen (from "scratch") on a MAX32660 motion co-processor board and measured the heading error twice at level attitude. The results (bold curves) are plotted with those from four different production USFS motion co-processor boards using the EM7180 "Sentral" sensor fusion hub.



All measurements were conducted under identical conditions using the same three-axis goniometer instrument. The RMS heading error for the USFSMAX motion co-processor was 0.17 - 0.25deg while the Sentral ranged between 0.92 and 1.2deg. It is also interesting to note that the Sentral heading error is sinusoidal in nature with a period of 180deg. This is characteristic of uncompensated soft iron error. Based upon these results, I conclude that the Sentral's heading accuracy is limited by the on-board dynamic calibration algorithm's ability to compensate soft iron effects... And that my calibration methods addresses this weakness.

The results presented in this log are based upon hand-built prototype boards. Kris is in the process of having a small production run of the MAX32660 motion co-processor boards made at a professional manufacturer. When these are complete, my next step will be to randomly select a sample of units and repeat the same calibrate-and-measure procedure to estimate baseline unit-to-unit variation of the heading accuracy...

Previous Log Next Log

Final Hardware Design
11/19/2019 at 01:01 • 0 comments

Unit-to-Unit Variation and On-Board Residual Hard Iron Error Correction
12/20/2019 at 00:44 • 1 comment

DISCUSSIONS

Log In or become a member to leave your comment

Log In/Sign up to comment

Greg Tomasch wrote 12/01/2019 at 00:30

Simon,

Again, thanks for your kind words. If you read the last two paragraphs of the reference above, this is the fruition of the hunches and "next steps" I had at the end of the last paper. It just takes a while to work through all of the details! As I mentioned in my last log entry, I plan to open a project soon to document the instruments and methods I developed/used to generate these latest results...

Best,

Greg

Simon Merrett wrote 12/01/2019 at 12:56

I look forward to that!

Kris Winer wrote 11/30/2019 at 19:53

We have published some previous results in Hackaday's journal, see https://hackaday.com/wp-content/uploads/2019/03/hackaday_journal-gregorytomasch_kriswiner-heading_accuracy_using_mems_sensors.pdf. We will likely submit a follow on next year.

Simon Merrett wrote 12/01/2019 at 12:55

Great, thanks for your contributions to the community.

Simon Merrett wrote 11/30/2019 at 19:47

This is gold. Have you thought about submitting something to to the Hackaday Journal of What You Don't Know? You may not think these logs are special but they seem so to me.

Going up?