

Automatic Calibration of Magnetometers

Sean D'Epagnier

March 26, 2009

Table of Contents

1	Introduction.....	2
2	Filtering.....	3
3	StillPoints.....	4
4	Spherical Calibration.....	5
5	Rotated Ellipsoid Calibration.....	6
6	Non-linear Rotated Ellipsoid Calibration.....	7
7	Quaternion Alignment Calibration.....	8
8	Rotated Aligned Ellipsoid Calibration.....	9
9	Box Alignment Calibration.....	10
10	Laser Alignment Calibration.....	11

1 Introduction

I came to the idea of auto-calibration when I realized the magnitude of my 3-axis accelerometers is constant when the device is not moving. This fact alone essentially allows me to auto-calibrate the sensors and recalibrate them when the device is in use.

2 Filtering

Without filtering, the data from the sensors is raw. This would be ok if the raw data were good enough to achieve the level of calibration desired. With filtering I find a 10-20x improvement in calibration accuracy. This is because many samples can be averaged over a longer period of time to give a higher resolution result, however this is only useful when the device is still.

There are a lot of ways to filter data. A typical lowpass filter simply computes the weighted average of the newest sample with the running average, if y is filtered data, and $lowpass$ is the lowpass constant:

$$y_n = (1 - lowpass)y_{n-1} + (lowpass)x_n$$

The trouble with this filter is that if the lowpass is too close to zero, it will react too slowly to large changes, and if it is too close to one, it will not effectively filter the data when the sensor is stable (device is not moving)

My calibration algorithms largely rely having the device perfectly still. This allows me to average a lot of samples to get a very good reading, but I also need to react quickly if the device is moving. For this I increase the lowpass factor toward 1 when the noise level is higher, and reduce it toward 0 when the noise is lower.

There are minimums and maximum values, and some other logic to keep things in order. Also this operation is now performed using only bit shifts and adds for performance reasons.

3 StillPoints

A stillpoint is a raw measurement taken when the device is not moving. This is determined by looking at the noise levels on the sensors. When the device is still, there are 3 important assumptions:

- The accelerometers should only be measuring gravity.
- The magnetic sensors get a good low-noise reading for just the earth's magnetic field.
- The angle between the accelerometer and magnetometer readings is the inclination angle and is constant for a given geographic location.

The first assumption is used to calibrate the accelerometers, the second for magnetometers (note: the device measures just the magnetic field if it is moving or not, the unfiltered data is used for the magfast calibration which works even if the device is moving) The last assumption is used to calculate the misalignment between the accelerometers and magnetometers. The last assumption can also compute calibration for the magnetometer.

4 Spherical Calibration

Spherical calibration only computes biases for each of the 3 sensors, as well as a single scale-factor for all 3 axis. This is used for the accelerometer and magnetometer (fast calibration) See (fit.c) for the actual implementation.

The equation used is:

$$(x - a)^2 + (y - b)^2 + (z - c)^2 = d^2$$

The unknowns a, b, and c are the biases for the x, y, and z sensors. The unknown d is the overall scalefactor.

To estimate these unknowns, recursive least squares is used. This is a simplification of a kalman filter. The following equations are used:

$$K = P_{k-1}H^T(H P_{k-1}H^T + R)^{-1}$$

$$X = X_{k-1} + KZ$$

$$P = (I - KH)P(I - KH)^T + KRK^T$$

- X – The current state, so for the case of 4 unknowns, it is a 4-vector.
- P – The covariance, a 4x4 matrix.
- I – The identity matrix.
- R – The confidence factor of the new measurement.
- H – The partial derivatives of the above equation with respect to each unknown. For example, for the spherical truth equation: $H = [-2(x - a), -2(y - b), -2(z - c), -2d]$
- Z – The residule of the measurement. For the spherical truth equation it is $d^2 - (x - a)^2 - (y - b)^2 - (z - c)^2$

Whenever there is a new measurement, the above recursive least squares equations are applied.

5 Rotated Ellipsoid Calibration

The magnetometer differs from the accelerometer in that it suffers from soft-iron distortion which increases crosscoupling (axes not orthogonal), as well large differences in scale factors between axis (typically 10%). The soft-iron calibration strategy taken is simplified as it essentially treats the sensors as soft-iron, not the metals around it. This means the soft-iron calibration is not complete, and the overall accuracy is reduced with additional iron. Soft iron distortions show up as cross-coupling errors, so these are both handled by estimating cross-coupling coefficients. I believe that it is possible to do a more complete model of the soft-iron distortions using tensors, however I would need a faster microprocessor, and gyros to attempt this calibration. The current calibration is significant; for example, with a given amount of iron, the heading errors are as much as 10 degrees with basic calibration (only bias and overall scalefactor) where with the calibration errors are reduced to 1 degree with scalefactor ratios, cross-coupling, misalignment factors.

The equations used:

$$b = \langle x_x - X_0, x_y - X_1, x_z - X_2 \rangle$$

$$m_x = X_3(b_x)$$

$$m_y = X_3(X_4b_y + X_6b_x)$$

$$m_z = X_3(X_5b_z + X_7b_x + X_8b_y)$$

$$m_x^2 + m_y^2 + m_z^2 = 1$$

By plugging in for m_x, m_y, m_z into the first equation, the overall equation is calculated. The same technique of recursive least squares is used, except there are 9 unknowns instead of only 4.

Standard least squares is used which gives better results. Whenever the device is still, the raw sensor measurements are stored. Once enough measurements are stored, the calibration state can be computed with:

$$X = X + (J^T J)^{-1} J^T R^T$$

- X – the current state
- J – the matrix with the partial derivatives for each point
- R – the vector of residues for each point

6 Non-linear Rotated Ellipsoid Calibration

To achieve maximum accuracy from the accelerometer sensors, their non-linearities must be corrected. In my tests this makes them roughly twice as accurate (average error 0.05 degrees instead of 0.1 degrees). The error can be reduced further using higher orders, but the returns are diminishing. Higher orders may appear to reduce the error more, but they require so many data points that they may fit to the specific data, not the sensors. 3rd order gives the best advantage while only increasing the number of states from 9 to 15.

The same method used for the magnetic sensors is employed however there are 6 unknowns added (2nd and 3rd order terms for each axis)

$$b = \langle x_x - X_0, x_y - X_1, x_z - X_2 \rangle$$

$$r = \langle b_x + X_9 b_x^2 + X_{10} b_x^3, b_y + X_{11} b_y^2 + X_{12} b_y^3, b_z + X_{13} b_z^2 + X_{14} b_z^3 \rangle$$

$$m_x = X_3(b_x)$$

$$m_y = X_3(X_4 b_y + X_6 b_x)$$

$$m_z = X_3(X_5 b_z + X_7 b_x + X_8 b_y)$$

$$m_x^2 + m_y^2 + m_z^2 = 1$$

This equation solves for many more unknowns, as a result requires more data points, but provides more accurate calibration.

7 Quaternion Alignment Calibration

The above chapters describe how to compute normalized orthogonal measurements for both magnetometer and accelerometer given enough good datapoints. Due to manufacturing imperfections the accelerometer is not in the same coordinate space as the magnetometer. They may be misaligned typically as much as 2-3 degrees. To correct for this error, we must calculate a rotation that needs to be applied to the magnetometer readings to put it into accelerometer (or sensor) coordinates.

A unit quaternion can be used to represent a rotation in 3-space. Because of this, we solve for a unit quaternion for our misalignment. Given a quaternion q , and vector m , we can apply the quaternion rotation to the vector with the quaternion triple product:

$$qnq^t$$

n is a quaternion which is $\langle 0, m_x, m_y, m_z \rangle$

The result is a quaternion, the last 3 parts is the rotated vector.

From this known fact, we can derive an interesting truth equation:

$$g.qmq^t = d$$

- g – the calibrated accelerometer measurement (gravity)
- m – the calibrated magnetometer measurement
- q – the unknown rotation to get from magnetometer to sensor coordinates
- d – the sin of the inclination (constant for a given magnetic latitude)

The unknowns are q and d . q is a quaternion with 4 parts, but we force it to be unit, so the first part is computed from the last 3. This leaves us with only 4 unknowns, and the recursive least squares method and regular least squares are used to compute the coefficients.

8 Rotated Aligned Ellipsoid Calibration

Rather than calibrate the magnetometer from magnitude, and align it to the accelerometers using dip angle, it is possible to perform both steps at the same time using both measurements. So far the results seem to be very similar, but not identical. I perform both methods for comparison purposes.

This method also can find the magnetometer biases, relative scale-factors, and cross-coupling terms using the accelerometer data which may allow it to converge with less data points. A minimum of only 7 measurements are required where the other method requires at least 9.

For this method there are 13 unknowns: a 3x3 transformation matrix, 3 biases, and the dip angle.

The equations used:

$$b = \langle x_x - X_0, x_y - X_1, x_z - X_2 \rangle$$

$$m_x = X_3(b_x + X_6b_y + X_9b_z)$$

$$m_y = X_3(X_4b_y + X_7b_x + X_{10}b_z)$$

$$m_z = X_3(X_5b_z + X_8b_x + X_{11}b_y)$$

$$m_x^2 + m_y^2 + m_z^2 = 1$$

$$g.m/|m| = X_{12}$$

The last two equations give two measurements to estimate the 13 states. Least squares can be performed, there are two rows added for each measurement, one for each truth equation.

9 Box Alignment Calibration

Box alignment calibration is used to align the sensors to whatever coordinate frame is being used. This is represented by a quaternion rotation. There are two methods used to compute this rotation:

- Rotation calculated when the sensors are told they are level and facing north. First the rotation to make the accelerometer point down is applied, then the rotation to rotate to make the x axis face north is measured and combined with the first rotation and this is the box alignment.
- The second method is if the direction of north is not known. In this case, the alignment to make the accels point down is applied, then we need data of the x axis pointed straight up. This allows us to compute the alignment for the yaw rotation using only the accels. These rotations are combined and this is the box alignment. Notice that the second method does not use the magnetometer, so it works even before the magnetometer is calibrated.

10 Laser Alignment Calibration

Laser alignment is a rotation to get from the sensor coordinates to align to an axis or vector. This means roll is ignored because we only want to align to the vector, the rotation around the laser is irrelevant. To compute laser alignment, we use the truth equation for multiple shots rotated around the laser axis:

$$a_x n_x + a_y n_y + a_z n_z = 1$$

- a – The Calibrated normalized accelerometer sensor reading for the shot.
- n – The rotation from this vector to $\langle 1, 0, 0 \rangle$ is the laser alignment calibration quaternion. In polar coordinates n is two angles plus a magnitude. The two angles are needed for the alignment rotation, the magnitude is $1/\cos(\text{incline})$. Incline is the incline angle that is used for the shots, and is arbitrary.

Notice that the magnetometers are not used for laser alignment calibration, so the laser can be aligned before the magnetometer is calibrated. The recursive least squares method can be used to find n .