Magnetic Calibration (magnetic.c)

Technical Note

Document TBD

Version: Draft

Authors: Mark Pedley and Michael Stanley

Date: September 2014

Table of Contents

2   **Technical Note**

freescale
semiconductor

Glossary

$\boldsymbol{A}$	Ellipsoid fit matrix
B	Geomagnetic field strength
$\boldsymbol{B}_c$	Calibrated magnetometer measurement vector
$\boldsymbol{B}_r$	Geomagnetic field in the reference (global) frame
$\boldsymbol{B}_p$	Uncalibrated magnetometer measurement vector
E	Fit error function
M	Number of measurements
$r[i]$	Error residual for *i-th* measurement
$\boldsymbol{r}$	Vector of error residuals
$\boldsymbol{R}$	Orientation matrix
$\boldsymbol{V}$	Hard iron vector
$\boldsymbol{W}$	Soft iron matrix
$X_j[i]$	Independent variable *j* for *i-th* measurement
$\boldsymbol{X}$	Matrix of independent variables $X_j[i]$
$Y[i]$	Dependent variable for *i-th* measurement
$\boldsymbol{Y}$	Vector of dependent variables $Y[i]$
β_i	Element i of the solution vector $\boldsymbol{\beta}$
$\boldsymbol{\beta}$	Solution vector
ε	Normalized fit error
λ	Lagrange Multiplier and eigenvalue

freescale ™
semiconductor

1 Introduction

1.1 Summary

This Application Note documents the mathematics underlying the functions in the file magnetic.c which determine the calibration of the magnetometer sensor and the hard and soft iron magnetic interference from the circuit board.

1.2 Functions

void fInitMagCalibration(struct MagCalibration *pthisMagCal, struct MagneticBuffer *pthisMagBuffer);
void iUpdateMagnetometerBuffer(struct MagneticBuffer *pthisMagBuffer, struct AccelSensor *pthisAccel, struct MagSensor *pthisMag, int32 loopcounter);
void fInvertMagCal(struct MagSensor *pthisMag, struct MagCalibration *pthisMagCal);   Computes the calibrated magnetometer measurements from the uncalibrated measurements using the computed magnetic calibration.   See section 3.
void fUpdateCalibration4INV(struct MagCalibration *pthisMagCal, struct MagneticBuffer *pthisMagBuffer, struct MagSensor *pthisMag);   See section 4.
void fUpdateCalibration7EIG(struct MagCalibration *pthisMagCal, struct MagneticBuffer *pthisMagBuffer, struct MagSensor *pthisMag);   See section 5.
void fUpdateCalibration10EIG(struct MagCalibration *pthisMagCal, struct MagneticBuffer *pthisMagBuffer, struct MagSensor *pthisMag);   See section 6.

Technical Note

freescale™
semiconductor

2 Least Squares Optimization

2.1 Linear Measurement Model

The general linear model relating independent variables $X_j[i]$ to dependent variable $Y[i]$ at measurement i via fitted model parameters β_j is:

$$Y[i] = \beta_0 X_0[i] + \beta_1 X_1[i] + \cdots + \beta_{N-1} X_{N-1}[i]$$

Eq 2.1.1

The fit to the model will, in general, not be perfectly accurate and will result in an error term $r[i]$ defined as:

$$r[i] = Y[i] - \beta_0 X_0[i] - \beta_1 X_1[i] - \cdots - \beta_{N-1} X_{N-1}[i]$$

Eq 2.1.2

For a series of M measurements, equation 2.1.2 can be written in the form:

$$\begin{pmatrix} r[0] \\ r[1] \\ \dots \\ r[M-1] \end{pmatrix} = \begin{pmatrix} Y[0] \\ Y[1] \\ \dots \\ Y[M-1] \end{pmatrix} - \begin{pmatrix} X_0[0] & X_1[0] & \dots & X_{N-1}[0] \\ X_0[1] & X_1[1] & \dots & X_{N-1}[1] \\ \dots & \dots & \dots & \dots \\ X_0[M-1] & X_1[M-1] & \dots & X_{N-1}[M-1] \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \dots \\ \beta_{N-1} \end{pmatrix}$$

Eq 2.1.3

With the definitions that $\boldsymbol{r}$ is the column vector of residuals:

$$\boldsymbol{r} = \begin{pmatrix} r[0] \\ r[1] \\ \dots \\ r[M-1] \end{pmatrix}$$

Eq 2.1.4

$\boldsymbol{Y}$ is the M by 1 column vector of M measurements on the dependent variable.

$$\boldsymbol{Y} = \begin{pmatrix} Y[0] \\ Y[1] \\ \dots \\ Y[M-1] \end{pmatrix}$$

Eq 2.1.5

$\boldsymbol{X}$ is the M by N matrix of M measurements of the independent variable:

$$\boldsymbol{X} = \begin{pmatrix} X_0[0] & X_1[0] & \dots & X_{N-1}[0] \\ X_0[1] & X_1[1] & \dots & X_{N-1}[1] \\ \dots & \dots & \dots & \dots \\ X_0[M-1] & X_1[M-1] & \dots & X_{N-1}[M-1] \end{pmatrix}$$

Eq 2.1.6

$\boldsymbol{\beta}$ is the N by 1 column vector of unknown model coefficients β_0 to β_{N-1} to be determined:

$$\boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \dots \\ \beta_{N-1} \end{pmatrix}$$

Eq 2.1.7

Then equation 2.1.3 can be written in the form:

$$\boldsymbol{r} = \boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}$$

Eq 2.1.8

If there are more measurements M than there are unknowns N then the equations are typically solved in a least squares sense by minimizing an error function E defined as the modulus squared of the error vector $\boldsymbol{r}$ defined in equation 2.1.5:

$$E = \sum_{i=0}^{M-1} r[i]^2 = \boldsymbol{r}^T \boldsymbol{r} = \|\boldsymbol{r}\|^2 = (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})^T (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}) = \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}\|^2$$

Eq 2.1.9

E is i) proportional to the number of measurements M and ii) has dimensions of the square of the elements $r[]$. A more useful normalized error measure ε is:

$$\varepsilon = \frac{1}{2}\sqrt{\frac{E}{M}}$$

<div align="right">Eq 2.1.10</div>

2.2   Normal Equations Solution in Non-Homogeneous Case

If the measurement vector on the dependent vector Y is not zero, then the equations are termed non-homogeneous. The error function E will be a minimum when it is stationary with respect to any perturbation $\delta\boldsymbol{\beta}$ about the optimal least squares solution $\boldsymbol{\beta}$:

$$lim_{\delta\boldsymbol{\beta}\to 0}\{E(\boldsymbol{\beta}+\delta\boldsymbol{\beta})\} - E(\boldsymbol{\beta}) = 0 \; for \; all \; \delta\boldsymbol{\beta} \qquad \text{Eq 2.2.1}$$

Substituting equation 2.1.8 into 2.2.1 and ignoring second order terms gives:

$$\left(Y - X(\boldsymbol{\beta}+\delta\boldsymbol{\beta})\right)^T\left(Y - X(\boldsymbol{\beta}+\delta\boldsymbol{\beta})\right) - (Y - X\boldsymbol{\beta})^T(Y - X\boldsymbol{\beta}) = 0 \qquad \text{Eq 2.2.2}$$

$$\Rightarrow -Y^T X\delta\boldsymbol{\beta} + (X\boldsymbol{\beta})^T X\delta\boldsymbol{\beta} - (X\delta\boldsymbol{\beta})^T(Y - X\boldsymbol{\beta}) = 0 \qquad \text{Eq 2.2.3}$$

$$\Rightarrow -Y^T X\delta\boldsymbol{\beta} + (X\boldsymbol{\beta})^T X\delta\boldsymbol{\beta} - \delta\boldsymbol{\beta}^T X^T Y + \delta\boldsymbol{\beta}^T X^T X\boldsymbol{\beta} = 0 \qquad \text{Eq 2.2.4}$$

Since $\delta\boldsymbol{\beta}^T X^T Y$ and $\delta\boldsymbol{\beta}^T X^T X\boldsymbol{\beta}$ are scalars, their values are unchanged by the transpose operation and equation 2.2.4 can be re-written as:

$$(-Y^T X + \boldsymbol{\beta}^T X^T X)\delta\boldsymbol{\beta} = 0 \; for \; all \; \delta\boldsymbol{\beta} \qquad \text{Eq 2.2.5}$$

$$\Rightarrow \boldsymbol{\beta}^T X^T X = Y^T X \Rightarrow X^T X\boldsymbol{\beta} = X^T Y \qquad \text{Eq 2.2.6}$$

$$\Rightarrow \boldsymbol{\beta} = (X^T X)^{-1} X^T Y \qquad \text{Eq 2.2.7}$$

Equation 2.2.7 is termed the Normal Equations solution for $\boldsymbol{\beta}$ in the non-homogeneous case.

The error function E at the optimum solution equals:

$$E = r^T r = \{Y - X\boldsymbol{\beta}\}^T\{Y - X\boldsymbol{\beta}\} = (Y^T - \boldsymbol{\beta}^T X^T)(Y - X\boldsymbol{\beta}) = Y^T Y - Y^T X\boldsymbol{\beta} - \boldsymbol{\beta}^T X^T Y + \boldsymbol{\beta}^T X^T X\boldsymbol{\beta} \qquad \text{Eq 2.2.8}$$

$$= Y^T Y - \boldsymbol{\beta}^T X^T Y - (\boldsymbol{\beta}^T X^T Y)^T + \boldsymbol{\beta}^T X^T X\boldsymbol{\beta} \qquad \text{Eq 2.2.9}$$

Since each term of equation 2.2.9 is a scalar and equal to its transpose, the error function can be written as:

$$E = Y^T Y - 2\boldsymbol{\beta}^T(X^T Y) + \boldsymbol{\beta}^T(X^T X)\boldsymbol{\beta} \qquad \text{Eq 2.2.10}$$

Substituting the expression for the solution vector $\boldsymbol{\beta}$ into 2.2.10 gives:

$$E = Y^T Y - 2((X^T X)^{-1} X^T Y)^T(X^T Y) + ((X^T X)^{-1} X^T Y)^T(X^T X)(X^T X)^{-1} X^T Y \qquad \text{Eq 2.2.11}$$

$$= Y^T Y - Y^T X\{(X^T X)^{-1}\}^T X^T Y \qquad \text{Eq 2.2.12}$$

For the special case where the number of measurements equals the number of model parameters to be fitted, the matrix X is square and the transpose and inversion operators commute:

$$(X^T X)^{-1} = X^{-1}(X^T)^{-1} \qquad \text{Eq 2.2.13}$$

In this special case:

$$E = Y^T Y - Y^T X\{X^{-1}(X^T)^{-1}\}^T X^T Y = Y^T Y - Y^T X X^{-1}(X^{-1})^T X^T Y = Y^T Y - Y^T(X^T)^{-1} X^T Y = 0 \qquad \text{Eq 2.2.14}$$

Equation 2.2.14 simply states the expected result that the error function is zero and the fit is perfect when the number of model parameters to be fitted equals the number of measurements.

2.3   Eigenvector Solution in Homogeneous Case

If the dependent measurement vector Y is zero, then the equations are termed homogeneous. The model being fitted in a least squares sense is now:

freescale
semiconductor

$$X\beta = 0 \qquad \text{Eq 2.3.1}$$

The error function E to be minimised simplifies to:

$$E = \|r\|^2 = \|X\beta\|^2 = (X\beta)^T X\beta = \beta^T X^T X\beta \qquad \text{Eq 2.3.2}$$

Unfortunately, using the Normal Equations solution given by equation 2.2.7 for the non-homogeneous case simply gives the zero vector solution for β when Y is the null vector.

$$\beta = (X^T X)^{-1} X^T Y = 0 \qquad \text{Eq 2.3.3}$$

This is a valid solution but not terribly useful. A solution method is required which minimizes the error function E in equation 2.3.2 subject to the constraint that β has non-zero magnitude. Since equation 2.3.1 is linear, we can scale the solution vector β to have unit magnitude:

$$1 - \beta^T \beta = 0 \qquad \text{Eq 2.3.4}$$

The error function is unaffected by adding multiples of equation 2.3.4 and can be re-written as:

$$E(\beta) = (X\beta)^T X\beta + \lambda(1 - \beta^T \beta) = \beta^T X^T X\beta + \lambda(1 - \beta^T \beta) \qquad \text{Eq 2.3.5}$$

This is equivalent to method of Lagrange Multipliers for constrained optimization. Applying the stationary constraint that $E(\beta + \delta\beta) = E(\beta)$ to equation 2.3.5 gives:

$$(\beta + \delta\beta)^T X^T X(\beta + \delta\beta) + \lambda\big(1 - (\beta + \delta\beta)^T(\beta + \delta\beta)\big) = \beta^T X^T X\beta + \lambda(1 - \beta^T \beta) \qquad \text{Eq 2.3.6}$$

Ignoring second order terms gives:

$$\delta\beta^T X^T X\beta + \beta^T X^T X\delta\beta - \lambda(\beta^T \delta\beta + \delta\beta^T \beta) = 0 \qquad \text{Eq 2.3.7}$$

Since each term in equation 2.3.7 is a scalar and equal to its transpose, the solution for the optimum β which constrains the performance function is:

$$\delta\beta^T (X^T X\beta - \lambda\beta) = 0 \qquad \text{Eq 2.3.8}$$

$$\Rightarrow X^T X\beta = \lambda\beta \qquad \text{Eq 2.3.9}$$

Equation 2.3.9 states that the required solution vector β is an eigenvector of the product matrix $X^T X$ associated with eigenvalue λ.

Substituting equation 2.3.9 into equation 2.3.2 for the error function E_i associated with eigenvalue λ_i and eigenvector β_i gives:

$$E_i = \beta_i^T X^T X\beta_i = \lambda_i \beta_i^T \beta_i = \lambda_i \qquad \text{Eq 2.3.10}$$

The error function E_i for the *i-th* eigenvector solution therefore equals the eigenvalue λ_i of $X^T X$ and the minimum error function is equal to the smallest eigenvalue λ_{min}. The required solution β is then the eigenvector associated with the smallest eigenvalue λ_{min}.

2.4 Eigenvectors and Eigenvalues of Symmetric Matrices

The measurement matrix $X^T X$ in equation 2.3.9 is obviously symmetric since:

$$(X^T X)^T = X^T X \qquad \text{Eq 2.4.1}$$

The eigenvectors of equation 2.4.1 satisfy:

$$(\beta_j)^T X^T X \beta_k = \lambda_k (\beta_j)^T \beta_k \qquad \text{Eq 2.4.2}$$

Transposing equation 2.4.2 gives:

$$\left\{(\beta_j)^T X^T X \beta_k\right\}^T = \left\{\lambda_k (\beta_j)^T \beta_k\right\}^T \qquad \text{Eq 2.4.3}$$

$$\Rightarrow (\beta_k)^T X^T X\beta_j = \lambda_k (\beta_k)^T \beta_j \qquad \text{Eq 2.4.4}$$

$$\Rightarrow \lambda_j (\beta_k)^T \beta_j = \lambda_k (\beta_k)^T \beta_j \qquad \text{Eq 2.4.5}$$

freescale™
semiconductor

$$\Rightarrow \left(\lambda_j - \lambda_k\right)(\boldsymbol{\beta}_k)^T \boldsymbol{\beta}_j = 0 \qquad \text{Eq 2.4.6}$$

$$\Rightarrow (\boldsymbol{\beta}_k)^T \boldsymbol{\beta}_j = 0 \; if \; \lambda_j \neq \lambda_k \qquad \text{Eq 2.4.7}$$

The eigenvectors of a symmetric matrix are therefore orthogonal if the eigenvalues are distinct. Symmetric matrices occur in two places in Freescale's sensor fusion software: i) as the measurement matrix $X^T X$ and ii) as the transposed product of the inverse soft iron matrix $\{W^{-1}\}^T W^{-1}$. In both cases the matrices are a function of noisy magnetometer measurements and the conditions that can lead to repeated eigenvalues (exactly zero soft iron interference for example) will simply not occur and therefore the eigenvectors will be orthogonal.

The definition of a positive semi-definite matrix A is that it satisfies for all non-zero vectors v_j:

$$\left(v_j\right)^T A v_j \geq 0 \qquad \text{Eq 2.4.8}$$

Setting $j = k$ in equation 2.4.2 gives:

$$\left(\boldsymbol{\beta}_j\right)^T X^T X \, \boldsymbol{\beta}_j = \lambda_j \left(\boldsymbol{\beta}_j\right)^T \boldsymbol{\beta}_j \qquad \text{Eq 2.4.9}$$

$$\Rightarrow \left| X \, \boldsymbol{\beta}_j \right|^2 = \lambda_j \left| \boldsymbol{\beta}_j \right|^2 \qquad \text{Eq 2.4.10}$$

The left hand side of equation 2.4.10 is obviously non-negative. For non-zero $\boldsymbol{\beta}_j$ it therefore follows that i) symmetric matrices are positive semi-definite and ii) have non-negative eigenvalues if the associated eigenvector is non-zero.

freescale
semiconductor

3　Hard and Soft Iron Magnetic Model

3.1　General Linear Model

The calibrated magnetometer reading B_c (where 'calibrated' means that hard and soft iron distortions have been removed) is simply the local geomagnetic field B_r rotated by the orientation matrix R describing the orientation of the magnetometer. The geomagnetic vector B_r is a fixed vector in the global reference frame and the multiplication by the circuit board orientation matrix R is an example of a vector transformation from the global coordinate frame to the sensor coordinate frame:

$$B_c = RB_r \hspace{6cm} \text{Eq 3.1.1}$$

The most general linear model for the distortion of B_c into the measurement magnetometer reading B_p by hard and soft iron distortion is:

$$B_p = WB_c + V = WRB_r + V \hspace{4cm} \text{Eq 3.1.2}$$

V is a 3x1 vector and W is a 3x3 matrix. The vector V is termed the hard iron offset and the matrix W is termed the soft iron matrix. Simplifying a complicated subject somewhat, the hard iron offset models the sensor's intrinsic zero field offset plus the effects of permanently magnetized components on the circuit board and the soft iron matrix models the directional effect of induced magnetic fields and differing sensitivities in the three axes of the magnetometer sensor.

Equation 3.1.2 proves to be an excellent model for the magnetometer measurements but obviously becomes less accurate when the linearity assumption starts to break down. The most common reason for deviations from equation 3.1.2 is the presence of magnetic hysteresis which is, by definition, a non-linear path dependent magnetic distortion.

The calibration algorithms derived in this document estimate the hard iron offset V and the soft iron matrix W from magnetometer measurements stored in the magnetometer buffer and then invert equation 3.1.2 to give the calibrated magnetometer measurement as:

$$B_c = W^{-1}(B_p - V) \hspace{5cm} \text{Eq 3.1.3}$$

3.2　Measurement Locus

Under arbitrary rotation of the phone, the locus of the calibrated magnetometer reading $B_c = RB_r$ lies on the surface of a sphere with radius equal to B the local geomagnetic field strength:

$$(B_c)^T B_c = (RB_r)^T RB_r = B_r{}^T R^T RB_r = B^2 \hspace{3cm} \text{Eq 3.2.1}$$

In the presence of hard and soft iron effects, the locus of uncalibrated magnetometer reading B_p lies on the surface defined using equation 3.1.3 to be:

$$\{W^{-1}(B_p - V)\}^T \{W^{-1}(B_p - V)\} = (RB_r)^T RB_r = B_r{}^T R^T RB_r = B^2 \hspace{1cm} \text{Eq 3.2.2}$$

$$\Rightarrow (B_p - V)^T (W^{-1})^T W^{-1}(B_p - V) = B^2 \hspace{3cm} \text{Eq 3.2.3}$$

The general expression for the locus of a vector u lying on the surface of an ellipsoid with center at u_0 is:

$$(u - u_0)^T A(u - u_0) = const \hspace{4cm} \text{Eq 3.2.4}$$

where A is a symmetric matrix defining the shape of the ellipsoid.

Equations 3.2.3 and 3.2.4 are of the same form since it can be easily proved that the matrix $A = \{W^{-1}\}^T W^{-1}$ is a symmetric matrix:

$$A^T = \{\{W^{-1}\}^T W^{-1}\}^T = \{W^{-1}\}^T \{\{W^{-1}\}^T\}^T = \{W^{-1}\}^T W^{-1} = A \hspace{1cm} \text{Eq 3.2.5}$$

The hard and soft iron distortion therefore force the geomagnetic vector, as measured in the magnetometer reference frame, to lie on the surface on an ellipsoid centered at the hard iron offset V with shape determined by the transposed product of the inverse soft iron matrix with itself $\{W^{-1}\}^T W^{-1}$. The precise point on the ellipsoid surface where any measurement falls is determined by the orientation of the magnetometer.

freescale™
semiconductor

Once the hard iron offset $\boldsymbol{V}$ and the soft iron matrix $\boldsymbol{W}$ have been determined by the calibration algorithms, the mapping of the raw distorted magnetometer measurement $\boldsymbol{B}_p$ onto the calibrated measurement $\boldsymbol{B}_c$ using equation 3.1.3 equates to mapping measurements from the surface of the ellipsoid to the surface of a sphere with radius B.

3.3 Example Calibration Surfaces

Figure 3-1 shows measurements taken from a simple sensor demonstration board with uncalibrated measurements in red and calibrated measurements in blue. The soft iron matrix $\boldsymbol{W}$ is close to the identity matrix and the hard iron vector is dominated by a 100uT offset in the z axis. This type of PCB could be calibrated for hard iron offset only using the algorithm described in section 5 although the more sophisticated soft iron algorithms of sections 6 and 7 could also be used although with little, if any, additional performance improvement.

In this particular example, the calibration mapping is approximately the translation:

$$\boldsymbol{B}_c \approx \boldsymbol{B}_p - V_z\widehat{\boldsymbol{k}} \approx \boldsymbol{B}_p - 100uT\widehat{\boldsymbol{k}} \qquad\qquad\qquad \text{Eq 3.3.1}$$
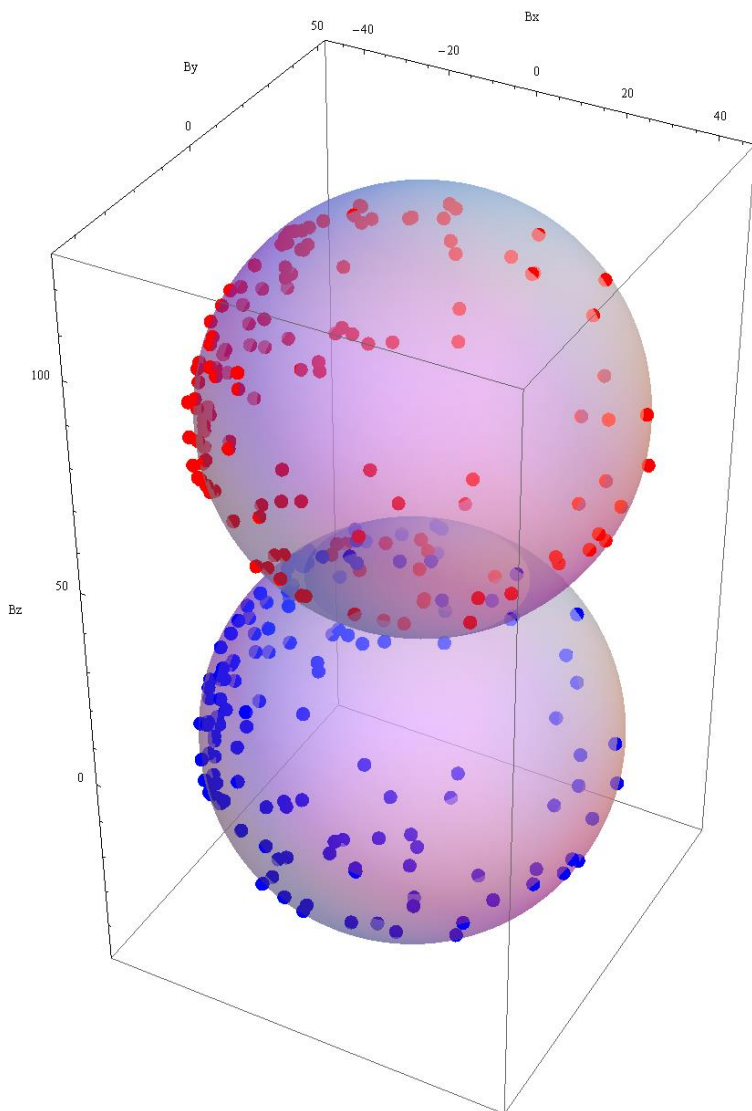

Figure 3-1: Uncalibrated (Red) and Calibrated (Blue) Measurements from a Simple Hard Iron Environment

Technical Note

Figure 3-2 shows measurements taken from an Android tablet with strong hard and soft iron distortions resulting from ferromagnetic components on the PCB. This type of distortion must be calibrated using the more sophisticated hard and soft iron algorithms described in sections 6 and 7.
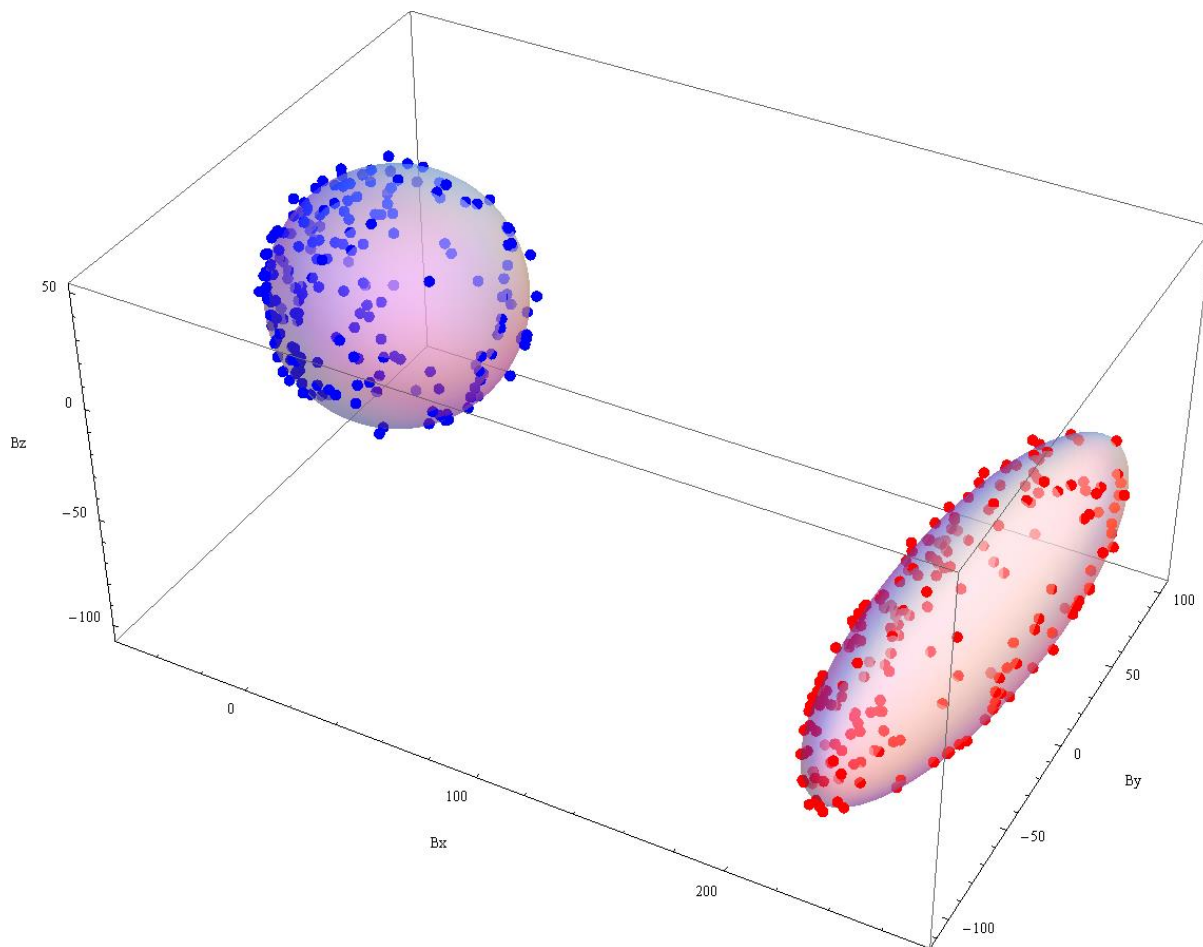

Figure 3-2: Uncalibrated (Red) and Calibrated (Blue) Measurements Taken in a Strong Hard and Soft Iron Environment

4 Magnetic Buffer

4.1 Introduction

The magnetic calibration algorithms described in sections 5 through 7 calculate the parameters of the hard and soft iron magnetic distortion by fitting the magnetometer measurements to the surface of an offset sphere or ellipsoid. The accuracy of the results depends on the availability of measurements whose distribution covers a reasonable proportion of the surface. Obviously 100 measurements all located in one corner of the surface will be more sensitive to magnetometer noise and give a poorer quality calibration than 100 measurements distributed throughout the surface. It would be fair to say that the careful selection of magnetometer measurements is of equal importance to the magnetic calibration algorithms in producing a high quality magnetic calibration.

4.2 Contents of the Magnetic Buffer

The default size of the magnetic buffer is 288 sets of *x*, *y*, *z* vector magnetometer measurements iBpFast organized as a two dimensional MAGBUFFSIZEX=12 by MAGBUFFSIZEY=24 element array. The name iBpFast denotes that the measurements are unaveraged and uncalibrated. It is not sensible to average the magnetometer measurements in an attempt to reduce noise since the averaged measurements will smear the distribution ellipsoid into a sphere and ultimate into a point giving a meaningless calibration.

Each element is dated by a 32 bit index counter storing the iteration when the measurement was taken. Older measurements are deemed less valuable than more recent measurements and are eventually aged out of the buffer.

The term iMagBufferCount contains the number of measurements in the buffer.

tanarray is a pre-computed lookup table of trigonometric tangents used to map the magnetometer measurements into a specific bin.

Only the most recent MAXMEASUREMENTS (default value 240) are used for the magnetic calibration. If MAXMEASUREMENTS were set to the full buffer size of 288, then a spurious measurement, perhaps caused by magnetic interference from a nearby magnet, would stay in the buffer affecting calibrations until the PCB is held at the same orientation and over-writes the spurious measurement with a new one. Setting MAXMEASUREMENTS to a value slightly less than the buffer size provides space for old measurements to be de-activated through age rather than being explicitly over-written.

```
// magnetometer measurement buffer

struct MagneticBuffer
{
   int16 iBpFast[3][MAGBUFFSIZEX][MAGBUFFSIZEY];   // uncalibrated magnetometer readings
   int32 index[MAGBUFFSIZEX][MAGBUFFSIZEY];        // array of time indices
   int16 tanarray[MAGBUFFSIZEX - 1];               // array of tangents of (100 * angle)
   int16 iMagBufferCount;                          // number of magnetometer readings
};
```

4.3 Update Algorithm

The accelerometer is used to index the magnetic measurements as a function of the two 'pseudo angles' with tangents equal to the ratio of y to x and z to x accelerometer measurements. The accelerometer measurements lie of the 1g sphere and the magnetic measurements lie of the magnetic ellipsoid. The indexing by accelerometer angles therefore provides a convenient binning of the magnetic ellipsoid by PCB orientation.

Case 1:

Technical Note

freescale™
semiconductor

If the magnetic buffer is fully populated (iMagBufferCount == MAXMEASUREMENTS), and the current entry selected by the accelerometer pseudo-angles is populated, then the buffer entry is simply over-written with the new measurement.

Case 2:

If the magnetic buffer is fully populated and the current entry is empty, then the buffer entry is over-written and the oldest measurement is retired.

Case 3:

If the magnetic buffer is not yet full (iMagBufferCount < MAXMEASUREMENTS) and the current entry is empty, then the current measurement is simply stored.

Case 4:

If the magnetic buffer is not yet full (iMagBufferCount < MAXMEASUREMENTS) and the current entry is not empty then, in order to populate the buffer as quickly as possible after power on, the measurement is stored in an empty bin provided it differs significantly from all existing measurements.

freescale
semiconductor

5 Four Parameter Magnetic Calibration Model

5.1 Construction of Four Parameter Linear Model

This section documents the simplest magnetic calibration algorithm implemented in function fUpdateCalibration4INV which calculates the 4 parameters comprising the hard iron offset vector $\boldsymbol{V}$ and geomagnetic field strength B. The soft iron matrix $\boldsymbol{W}$ is assumed to be the identity matrix. This model provides fairly high performance with high simplicity on simple circuit boards which tend to have limited soft iron distortion.

Equation 3.1.2 for the case of a PCB subject to hard iron interference only simplifies to:

$$\left(\boldsymbol{B}_p - \boldsymbol{V}\right)^T \left(\boldsymbol{B}_p - \boldsymbol{V}\right) = B^2 \qquad \text{Eq 5.1.1}$$

$$\Rightarrow \boldsymbol{B}_p{}^T \boldsymbol{B}_p - 2\boldsymbol{B}_p{}^T \boldsymbol{V} + \boldsymbol{V}^T \boldsymbol{V} - B^2 = 0 \qquad \text{Eq 5.1.2}$$

Equations 5.1.1 and 5.1.2 model the locus of the magnetometer measurements $\boldsymbol{B}_p$ as lying on the surface of a sphere with radius B offset from the origin by $\boldsymbol{V}$.

The residual error $r[i]$ for the *i-th* observation is:

$$r[i] = B_{px}[i]^2 + B_{py}[i]^2 + B_{pz}[i]^2 - 2B_{px}[i]V_x - 2B_{py}[i]V_y - 2B_{pz}[i]V_z + V_x{}^2 + V_y{}^2 + V_z{}^2 - B^2 \qquad \text{Eq 5.1.3}$$

Simplifying and returning to matrix format gives:

$$r[i] = \left(B_{px}[i]^2 + B_{py}[i]^2 + B_{pz}[i]^2\right) - \begin{pmatrix} B_{px}[i] \\ B_{py}[i] \\ B_{pz}[i] \\ 1 \end{pmatrix}^T \begin{pmatrix} 2V_x \\ 2V_y \\ 2V_z \\ B^2 - V_x{}^2 - V_y{}^2 - V_z{}^2 \end{pmatrix} \qquad \text{Eq 5.1.4}$$

The dependent measurement variable $y[i]$ can be defined to be:

$$y[i] = B_{px}[i]^2 + B_{py}[i]^2 + B_{pz}[i]^2 \qquad \text{Eq 5.1.5}$$

and the solution vector $\boldsymbol{\beta}$ as:

$$\boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix} = \begin{pmatrix} 2V_x \\ 2V_y \\ 2V_z \\ B^2 - V_x{}^2 - V_y{}^2 - V_z{}^2 \end{pmatrix} \qquad \text{Eq 5.1.6}$$

The error residual $r[i]$ is now given by:

$$r[i] = B_{px}[i]^2 + B_{py}[i]^2 + B_{pz}[i]^2 - \begin{pmatrix} B_{px}[i] \\ B_{py}[i] \\ B_{pz}[i] \\ 1 \end{pmatrix}^T \begin{pmatrix} 2V_x \\ 2V_y \\ 2V_z \\ B^2 - V_x{}^2 - V_y{}^2 - V_z{}^2 \end{pmatrix} \qquad \text{Eq 5.1.7}$$

$$= B_{px}[i]^2 + B_{py}[i]^2 + B_{pz}[i]^2 - \begin{pmatrix} B_{px}[i] \\ B_{py}[i] \\ B_{pz}[i] \\ 1 \end{pmatrix}^T \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix} \qquad \text{Eq 5.1.8}$$

Equation 5.1.8 can be expanded to represent M measurements as:

freescale semiconductor

$$\begin{pmatrix} r[0] \\ r[1] \\ ... \\ r[M-1] \end{pmatrix} = \begin{pmatrix} B_{px}[0]^2 + B_{py}[0]^2 + B_{pz}[0]^2 \\ B_{px}[1]^2 + B_{py}[1]^2 + B_{pz}[1]^2 \\ ... \\ B_{px}[M-1]^2 + B_{py}[M-1]^2 + B_{pz}[M-1]^2 \end{pmatrix} -$$

$$\begin{pmatrix} B_{px}[0] & B_{py}[0] & B_{pz}[0] & 1 \\ B_{px}[1] & B_{py}[1] & B_{pz}[1] & 1 \\ ... & ... & ... & 1 \\ B_{px}[M-1] & B_{py}[M-1] & B_{pz}[M-1] & 1 \end{pmatrix} \begin{pmatrix} 2V_x \\ 2V_y \\ 2V_z \\ B^2 - V_x^2 - V_y^2 - V_z^2 \end{pmatrix}$$
Eq 5.1.9

With the definitions of the error residual vector r as:

$$r = \begin{pmatrix} r[0] \\ r[1] \\ ... \\ r[M-1] \end{pmatrix}$$
Eq 5.1.10

and Y the vector of dependent variables:

$$Y = \begin{pmatrix} B_{px}[0]^2 + B_{py}[0]^2 + B_{pz}[0]^2 \\ B_{px}[1]^2 + B_{py}[1]^2 + B_{pz}[1]^2 \\ ... \\ B_{px}[M-1]^2 + B_{py}[M-1]^2 + B_{pz}[M-1]^2 \end{pmatrix}$$
Eq 5.1.11

and X the M by 4 measurement matrix:

$$X = \begin{pmatrix} B_{px}[0] & B_{py}[0] & B_{pz}[0] & 1 \\ B_{px}[1] & B_{py}[1] & B_{pz}[1] & 1 \\ ... & ... & ... & 1 \\ B_{px}[M-1] & B_{py}[M-1] & B_{pz}[M-1] & 1 \end{pmatrix}$$
Eq 5.1.12

then equation 5.1.9 can be written as:

$$r = Y - X\beta$$
Eq 5.1.13

The model being fitted has the non-homogeneous form $r = Y - X\beta$ and can be solved using the Normal Equations method documented in section 2.

The matrices $X^T X$, $X^T Y$ and $Y^T Y$ have values:

$$X^T X = \begin{pmatrix} B_{px}[0] & B_{py}[0] & B_{pz}[0] & 1 \\ B_{px}[1] & B_{py}[1] & B_{pz}[1] & 1 \\ ... & ... & ... & 1 \\ B_{px}[M-1] & B_{py}[M-1] & B_{pz}[M-1] & 1 \end{pmatrix}^T \begin{pmatrix} B_{px}[0] & B_{py}[0] & B_{pz}[0] & 1 \\ B_{px}[1] & B_{py}[1] & B_{pz}[1] & 1 \\ ... & ... & ... & 1 \\ B_{px}[M-1] & B_{py}[M-1] & B_{pz}[M-1] & 1 \end{pmatrix}$$
Eq 5.1.14

$$= \sum_{i=0}^{M-1} \begin{pmatrix} B_{px}[i]^2 & B_{px}[i]B_{py}[i] & B_{px}[i]B_{pz}[i] & B_{px}[i] \\ B_{px}[i]B_{py}[i] & B_{py}[i]^2 & B_{py}[i]B_{pz}[i] & B_{py}[i] \\ B_{px}[i]B_{pz}[i] & B_{py}[i]B_{pz}[i] & B_{pz}[i]^2 & B_{pz}[i] \\ B_{px}[i] & B_{py}[i] & B_{pz}[i] & 1 \end{pmatrix}$$

Eq 5.1.17

freescale™
semiconductor

$$\mathbf{X}^T\mathbf{Y} = \begin{pmatrix} B_{px}[0] & B_{py}[0] & B_{pz}[0] & 1 \\ B_{px}[1] & B_{py}[1] & B_{pz}[1] & 1 \\ ... & ... & ... & 1 \\ B_{px}[M-1] & B_{py}[M-1] & B_{pz}[M-1] & 1 \end{pmatrix}^T \begin{pmatrix} B_{px}[0]^2 + B_{py}[0]^2 + B_{pz}[0]^2 \\ B_{px}[1]^2 + B_{py}[1]^2 + B_{pz}[1]^2 \\ ... \\ B_{px}[M-1]^2 + B_{py}[M-1]^2 + B_{pz}[M-1]^2 \end{pmatrix} \qquad \text{Eq 5.1.18}$$

$$= \sum_{i=0}^{M-1} \begin{pmatrix} B_{px}[i]\left(B_{px}[i]^2 + B_{py}[i]^2 + B_{pz}[i]^2\right) \\ B_{py}[i]\left(B_{px}[i]^2 + B_{py}[i]^2 + B_{pz}[i]^2\right) \\ B_{pz}[i]\left(B_{px}[i]^2 + B_{py}[i]^2 + B_{pz}[i]^2\right) \\ B_{px}[i]^2 + B_{py}[i]^2 + B_{pz}[i]^2 \end{pmatrix}$$

<div align="right">Eq 5.1.19</div>

$$\mathbf{Y}^T\mathbf{Y} = \begin{pmatrix} B_{px}[0]^2 + B_{py}[0]^2 + B_{pz}[0]^2 \\ B_{px}[1]^2 + B_{py}[1]^2 + B_{pz}[1]^2 \\ ... \\ B_{px}[M-1]^2 + B_{py}[M-1]^2 + B_{pz}[M-1]^2 \end{pmatrix}^T \begin{pmatrix} B_{px}[0]^2 + B_{py}[0]^2 + B_{pz}[0]^2 \\ B_{px}[1]^2 + B_{py}[1]^2 + B_{pz}[1]^2 \\ ... \\ B_{px}[M-1]^2 + B_{py}[M-1]^2 + B_{pz}[M-1]^2 \end{pmatrix} \qquad \text{Eq 5.1.20}$$

$$= \sum_{i=0}^{M-1} \left(B_{px}[i]^2 + B_{py}[i]^2 + B_{pz}[i]^2\right)^2$$

<div align="right">Eq 5.1.21</div>

The solution vector $\boldsymbol{\beta}$ is then given by equation 2.2.7 as:

$$\begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix} = \left\{ \sum_{i=0}^{M-1} \begin{pmatrix} B_{px}[i]^2 & B_{px}[i]B_{py}[i] & B_{px}[i]B_{pz}[i] & B_{px}[i] \\ B_{px}[i]B_{py}[i] & B_{py}[i]^2 & B_{py}[i]B_{pz}[i] & B_{py}[i] \\ B_{px}[i]B_{pz}[i] & B_{py}[i]B_{pz}[i] & B_{pz}[i]^2 & B_{pz}[i] \\ B_{px}[i] & B_{py}[i] & B_{pz}[i] & 1 \end{pmatrix} \right\}^{-1} \sum_{i=0}^{M-1} \begin{pmatrix} B_{px}[i]\left(B_{px}[i]^2 + B_{py}[i]^2 + B_{pz}[i]^2\right) \\ B_{py}[i]\left(B_{px}[i]^2 + B_{py}[i]^2 + B_{pz}[i]^2\right) \\ B_{pz}[i]\left(B_{px}[i]^2 + B_{py}[i]^2 + B_{pz}[i]^2\right) \\ B_{px}[i]^2 + B_{py}[i]^2 + B_{pz}[i]^2 \end{pmatrix}$$

<div align="right">Eq 5.1.22</div>

5.2 Hard Iron Vector

The hard iron solution vector is given directly by equation 5.1.6 as:

$$\begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix} = \left(\frac{1}{2}\right) \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{pmatrix} \qquad\qquad \text{Eq 5.2.1}$$

5.3 Soft Iron Matrix

The soft iron matrix $\mathbf{W}$ is always the identity matrix in the 4 parameter magnetic calibration model.

5.4 Geomagnetic Field Strength

The geomagnetic field strength is computed from the last component of equation 5.1.6 as.

$$B^2 = \beta_3 + V_x^{\,2} + V_y^{\,2} + V_z^{\,2} \Rightarrow B = \sqrt{\beta_3 + V_x^{\,2} + V_y^{\,2} + V_z^{\,2}} \qquad\qquad \text{Eq 5.4.1}$$

5.5 Fit Error

Technical Note

freescale ™
semiconductor

The residuals $r[i]$ of equation 5.1.4 have dimensions of magnetic field strength squared. The error function E is proportional to the number of measurements M and has dimensions of the residuals squared or the fourth power of the geomagnetic field strength B. The dimensionless measure of fit error ε used is defined to be:

$$\varepsilon = \frac{1}{2B^2}\sqrt{\frac{E}{M}}$$

<div align="right">Eq 5.5.1</div>

freescale™
semiconductor

6 Seven Parameter Magnetic Calibration Model

6.1 Construction of Seven Parameter Linear Model

This section documents the magnetic calibration algorithm implemented in function fUpdateCalibration7EIG which extends the 4 parameters model of the previous section with the additional of three gain terms of the diagonal of the soft iron matrix giving a total of 7 magnetic calibration parameters. This model gives a significant improvement when either the magnetometer sensor has differing gains in its three channels or when the PCB has differing magnetic impedances along its three Cartesian axes. The diagonal form of W means that the magnetic distribution ellipsoid is modeled as having its principal axes aligned with the PCB's Cartesian axes.

The magnetometer measurement B_p in the presence of arbitrary orientation and hard and soft iron interference is modeled as:

$$B_p = WRB_r + V \qquad \text{Eq 6.1.1}$$

where for the 7 parameter magnetic calibration model the soft iron matrix W is diagonal.

The locus of the magnetometer measurements is:

$$\{W^{-1}(B_p - V)\}^T W^{-1}(B_p - V) = (B_p - V)^T (W^{-1})^T W^{-1}(B_p - V) = (B_p - V)^T A(B_p - V) = B^2 \qquad \text{Eq 6.1.2}$$

The manipulations which follow derive an expression for the error residual $r[i]$ from the i-th measurement defined as:

$$r[i] = \left|W^{-1}(B_p[i] - V)\right|^2 - B^2 = |B_c[i]|^2 - B^2 \qquad \text{Eq 6.1.3}$$

$r[i]$ is defined as the difference between the squared modulus of the calibrated magnetometer measurement $B_c[i]$ and the square of the radius of the geomagnetic sphere. $r[i]$ therefore has dimensions of B^2 as was also the case for the four element calibration model.

Expanding equation 6.1.3 gives:

$$r[i] = B_p^T A B_p - B_p^T A V - V^T A B_p + V^T A V - B^2 \qquad \text{Eq 6.1.4}$$

Since $B_p^T A V$ is a scalar:

$$\left(B_p^T A V\right)^T = V^T A B_p = B_p^T A V \qquad \text{Eq 6.1.5}$$

Substituting equation 6.1.5 into equation 6.1.4 and re-arranging gives:

$$r[i] = B_p^T A B_p - 2B_p^T A V + V^T A V - B^2 \qquad \text{Eq 6.1.6}$$

Expanding equation 6.1.6 for the *i-th* measurement gives:

$$r[i] = A_{xx}B_{px}[i]^2 + A_{yy}B_{py}[i]^2 + A_{zz}B_{pz}[i]^2 - 2B_{px}[i]A_{xx}V_x - 2B_{py}[i]A_{yy}V_y - 2B_{pz}[i]A_{zz}V_z$$

$$+ A_{xx}V_x{}^2 + A_{yy}V_y{}^2 + A_{zz}V_z{}^2 - B^2 \qquad \text{Eq 6.1.7}$$

Simplifying and returning to matrix format gives:

$$r[i] = \begin{pmatrix} B_{px}[i]^2 \\ B_{py}[i]^2 \\ B_{pz}[i]^2 \\ B_{px}[i] \\ B_{py}[i] \\ B_{pz}[i] \\ 1 \end{pmatrix}^T \begin{pmatrix} A_{xx} \\ A_{yy} \\ A_{zz} \\ -2A_{xx}V_x \\ -2A_{yy}V_y \\ -2A_{zz}V_z \\ A_{xx}V_x{}^2 + A_{yy}V_y{}^2 + A_{zz}V_z{}^2 - B^2 \end{pmatrix} \qquad \text{Eq 6.1.8}$$

Defining the right hand side of equation 6.1.8 to be the solution vector β gives:

freescale™
semiconductor

$$\boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \\ \beta_6 \end{pmatrix} = \begin{pmatrix} A_{xx} \\ A_{yy} \\ A_{zz} \\ -2A_{xx}V_x \\ -2A_{yy}V_y \\ -2A_{zz}V_z \\ A_{xx}V_x{}^2 + A_{yy}V_y{}^2 + A_{zz}V_z{}^2 - B^2 \end{pmatrix} \qquad \text{Eq 6.1.9}$$

Equation 6.1.7 for the error residual $r[i]$ whose squared sum is to be minimized is now:

$$r[i] = \begin{pmatrix} B_{px}[i]^2 \\ B_{py}[i]^2 \\ B_{pz}[i]^2 \\ B_{px}[i] \\ B_{py}[i] \\ B_{pz}[i] \\ 1 \end{pmatrix}^T \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \\ \beta_6 \end{pmatrix} \qquad \text{Eq 6.1.10}$$

With the definition of the error residual vector $\boldsymbol{r}$ from M measurements as:

$$\boldsymbol{r} = \begin{pmatrix} r[0] \\ r[1] \\ \dots \\ r[M-1] \end{pmatrix} \qquad \text{Eq 6.1.11}$$

and $\boldsymbol{X}$ defined as is the M by 7 measurement matrix:

$$\boldsymbol{X} = \begin{pmatrix} B_{px}[0]^2 & B_{py}[0]^2 & B_{pz}[0]^2 & B_{px}[0] & B_{py}[0] & B_{pz}[0] & 1 \\ B_{px}[1]^2 & B_{py}[1]^2 & B_{pz}[1]^2 & B_{px}[1] & B_{py}[1] & B_{pz}[1] & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots & 1 \\ B_{px}[M-1]^2 & B_{py}[M-1]^2 & B_{pz}[M-1]^2 & B_{px}[M-1] & B_{py}[M-1] & B_{pz}[M-1] & 1 \end{pmatrix} \qquad \text{Eq 6.1.12}$$

then equation 6.1.8 can be expanded to represent M measurements as:

$$\boldsymbol{r} = \boldsymbol{X}\boldsymbol{\beta} \qquad \text{Eq 6.1.13}$$

The model being fitted is the homogeneous model $\boldsymbol{X}\boldsymbol{\beta} = 0$ which can be solved for $\boldsymbol{\beta}$ using the eigen-decomposition approach described in section 2.

The 7x7 product matrix $\boldsymbol{X}^T\boldsymbol{X}$ whose eigenvectors and eigenvalues are to be determined evaluates to:

$$\boldsymbol{X}^T\boldsymbol{X} = \begin{pmatrix} B_{px}[0]^2 & B_{px}[1]^2 & \dots & B_{px}[M-1]^2 \\ B_{py}[0]^2 & B_{py}[1]^2 & \dots & B_{py}[M-1]^2 \\ B_{pz}[0]^2 & B_{pz}[1]^2 & \dots & B_{pz}[M-1]^2 \\ B_{px}[0] & B_{px}[1] & \dots & B_{px}[M-1] \\ B_{py}[0] & B_{py}[1] & \dots & B_{py}[M-1] \\ B_{pz}[0] & B_{pz}[1] & \dots & B_{pz}[M-1] \\ 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} B_{px}[0]^2 & B_{py}[0]^2 & B_{pz}[0]^2 & B_{px}[0] & B_{py}[0] & B_{pz}[0] & 1 \\ B_{px}[1]^2 & B_{py}[1]^2 & B_{pz}[1]^2 & B_{px}[1] & B_{py}[1] & B_{pz}[1] & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots & 1 \\ B_{px}[M-1]^2 & B_{py}[M-1]^2 & B_{pz}[M-1]^2 & B_{px}[M-1] & B_{py}[M-1] & B_{pz}[M-1] & 1 \end{pmatrix}$$

$$\text{Eq 6.1.14}$$

freescale
semiconductor

$$= \sum_{i=0}^{M-1} \begin{pmatrix} B_{px}[i]^4 & B_{px}[i]^2 B_{py}[i]^2 & B_{px}[i]^2 B_{pz}[i]^2 & B_{px}[i]^3 & B_{px}[i]^2 B_{py}[i] & B_{px}[i]^2 B_{pz}[i] & B_{px}[i]^2 \\ B_{py}[i]^2 B_{px}[i]^2 & B_{py}[i]^4 & B_{py}[i]^2 B_{pz}[i]^2 & B_{py}[i]^2 B_{px}[i] & B_{py}[i]^3 & B_{py}[i]^2 B_{pz}[i] & B_{py}[i]^2 \\ B_{pz}[i]^2 B_{px}[i]^2 & B_{pz}[i]^2 B_{py}[i]^2 & B_{pz}[i]^4 & B_{pz}[i]^2 B_{px}[i] & B_{pz}[i]^2 B_{py}[i] & B_{pz}[i]^3 & B_{pz}[i]^2 \\ B_{px}[i]^3 & B_{px}[i] B_{py}[i]^2 & B_{px}[i] B_{pz}[i]^2 & B_{px}[i]^2 & B_{px}[i] B_{py}[i] & B_{px}[i] B_{pz}[i] & B_{px}[i] \\ B_{py}[i] B_{px}[i]^2 & B_{py}[i]^3 & B_{py}[i] B_{pz}[i]^2 & B_{py}[i] B_{px}[i] & B_{py}[i]^2 & B_{py}[i] B_{pz}[i] & B_{py}[i] \\ B_{pz}[i] B_{px}[i]^2 & B_{pz}[i] B_{py}[i]^2 & B_{pz}[i]^3 & B_{pz}[i] B_{px}[i] & B_{pz}[i] B_{py}[i] & B_{pz}[i]^2 & B_{pz}[i] \\ B_{px}[i]^2 & B_{py}[i]^2 & B_{pz}[i]^2 & B_{px}[i] & B_{py}[i] & B_{pz}[i] & 1 \end{pmatrix}$$

<div align="right">Eq 6.1.15</div>

Since the eigenvalues of $X^T X$ are equal to the fit errors associated with the seven candidate eigenvector solutions, the required solution vector β is the eigenvector associated with the smallest eigenvalue λ_{min}.

6.2   Ellipsoid Fit Matrix

The ellipsoid fit matrix A is obtained directly from the first three rows of the solution vector β in equation 6.1.9:

$$A = \begin{pmatrix} A_{xx} & 0 & 0 \\ 0 & A_{yy} & 0 \\ 0 & 0 & A_{zz} \end{pmatrix} = \begin{pmatrix} \beta_0 & 0 & 0 \\ 0 & \beta_1 & 0 \\ 0 & 0 & \beta_2 \end{pmatrix}$$

<div align="right">Eq 6.2.1</div>

The solution eigenvector β is undefined within a multiplicative factor of +/-1 (assuming it's normalized to unit magnitude). A test must therefore be performed on the determinant of the ellipsoid matrix A defined in equation 7.2.1 and the entire solution vector β negated if the determinant is negative. Negating the solution vector β changes the sign of A and ensures a positive determinant.

The ellipsoid matrix A is then normalized to have unit determinant:

$$\begin{vmatrix} A_{xx} & 0 & 0 \\ 0 & A_{yy} & 0 \\ 0 & 0 & A_{zz} \end{vmatrix} = A_{xx} A_{yy} A_{zz} = 1$$

<div align="right">Eq 6.4.2</div>

The justification for the normalization in equation 6.4.2 is that it's impossible to separate out the geomagnetic field strength B from the soft iron magnetic matrix gain terms. A 25 uT geomagnetic field strength with no soft iron gain terms gives the same magnetometer measurement as a 50uT geomagnetic field strength attenuated 50% by magnetic shielding. The solution taken in the Freescale software sets the determinant of the soft iron matrices to 1 and assigns the magnitude of the calibrated measurement to the geomagnetic field strength B.

6.3   Hard Iron Vector

The hard iron model is given by equation 6.1.9 as:

$$\begin{pmatrix} -2A_{xx}V_x \\ -2A_{yy}V_y \\ -2A_{zz}V_z \end{pmatrix} = \begin{pmatrix} \beta_3 \\ \beta_4 \\ \beta_5 \end{pmatrix} \Rightarrow \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix} = \begin{pmatrix} \left(\dfrac{-\beta_3}{2A_{xx}} \right) \\ \left(\dfrac{-\beta_4}{2A_{yy}} \right) \\ \left(\dfrac{-\beta_5}{2A_{zz}} \right) \end{pmatrix} = \begin{pmatrix} \left(\dfrac{-\beta_3}{2\beta_0} \right) \\ \left(\dfrac{-\beta_4}{2\beta_1} \right) \\ \left(\dfrac{-\beta_5}{2\beta_2} \right) \end{pmatrix}$$

<div align="right">Eq 6.3.1</div>

6.4   Inverse Soft Iron Matrix

The inverse soft iron matrix by easily be found from the square root of the diagonal ellipsoid matrix as:

freescale™
semiconductor

$$W^{-1} = \begin{pmatrix} W_{xx} & 0 & 0 \\ 0 & W_{yy} & 0 \\ 0 & 0 & W_{zz} \end{pmatrix} = \sqrt{A} = \begin{pmatrix} \sqrt{\beta_0} & 0 & 0 \\ 0 & \sqrt{\beta_1} & 0 \\ 0 & 0 & \sqrt{\beta_2} \end{pmatrix}$$

Eq 6.4.1

6.5   Geomagnetic Field Strength

The geomagnetic field strength B is given by the last component of equation 6.1.9:

$$\beta_6 = A_{xx}V_x{}^2 + A_{yy}V_y{}^2 + A_{zz}V_z{}^2 - B^2$$

Eq 6.5.1

$$\Rightarrow B^2 = A_{xx}V_x{}^2 + A_{yy}V_y{}^2 + A_{zz}V_z{}^2 - \beta_6$$

Eq 6.5.2

Since the solution vector and ellipsoid matrix elements may be negated in software to force a positive determinant, it is necessary to compute the geomagnetic field from the absolute value of the right hand side of equation 6.5.2.

$$B = \sqrt{\left| A_{xx}V_x{}^2 + A_{yy}V_y{}^2 + A_{zz}V_z{}^2 - \beta_6 \right|}$$

Eq 6.5.3

6.6   Fit Error

The error function E equals the smallest eigenvalue λ_{min} of the product matrix $X^T X$ but is not normalized to the number of measurement points M nor is it normalized to the geomagnetic field strength B. Since $E = r^T r$ and r has M elements with each element having dimensions B^2, a suitable normalized calibration fit error measure ε is:

$$\varepsilon = \frac{1}{2B^2} \sqrt{\frac{\lambda_{min}}{M}}$$

Eq 6.6.1

freescale™
semiconductor

7   Ten Parameter Magnetic Calibration Model

7.1   Construction of the Ten Parameter Linear Model

This section documents the magnetic calibration algorithm implemented in function fUpdateCalibration10EIG which extends the 7 parameters model of the previous section with the additional of 3 off-diagonal soft iron matrix terms to give a total of 10 magnetic calibration parameters. This model gives an improvement over the 7 element model when the PCB's magnetic impedances steer the geomagnetic field in directions which are not aligned with the PCB's Cartesian axes giving a rotated magnetic ellipsoid.

The magnetometer measurement $\boldsymbol{B}_p$ in the presence of arbitrary orientation and hard and soft iron interference is modeled as:

$$\boldsymbol{B}_p = \boldsymbol{W}\boldsymbol{R}\boldsymbol{B}_r + \boldsymbol{V} \tag{Eq 7.1.1}$$

where for the 10 parameter magnetic calibration model the soft iron matrix $\boldsymbol{W}$ is symmetric.

The locus of the magnetometer measurements is:

$$\left\{ \boldsymbol{W}^{-1}(\boldsymbol{B}_p - \boldsymbol{V}) \right\}^T \boldsymbol{W}^{-1}(\boldsymbol{B}_p - \boldsymbol{V}) = (\boldsymbol{B}_p - \boldsymbol{V})^T (\boldsymbol{W}^{-1})^T \boldsymbol{W}^{-1}(\boldsymbol{B}_p - \boldsymbol{V}) = (\boldsymbol{B}_p - \boldsymbol{V})^T \boldsymbol{A}(\boldsymbol{B}_p - \boldsymbol{V}) = B^2 \tag{Eq 7.1.2}$$

Equation 7.1.2 model the locus of the magnetometer measurements $\boldsymbol{B}_p$ as lying on the surface of an ellipsoid with arbitrary dimensions and directions of its axes and offset from the origin by the hard iron vector $\boldsymbol{V}$.

The manipulations which follow derive an expression for the error residual $r[i]$ from the *i*-th measurement defined as:

$$r[i] = \left| \boldsymbol{W}^{-1}(\boldsymbol{B}_p[i] - \boldsymbol{V}) \right|^2 - B^2 = |\boldsymbol{B}_c[i]|^2 - B^2 \tag{Eq 7.1.3}$$

$r[i]$ is defined as the difference between the squared modulus of the calibrated magnetometer measurement $\boldsymbol{B}_c[i]$ and the square of the radius of the geomagnetic sphere. $r[i]$ therefore has dimensions of B^2.

Expanding equation 7.1.3 gives:

$$r[i] = \boldsymbol{B}_p{}^T \boldsymbol{A}\boldsymbol{B}_p - \boldsymbol{B}_p{}^T \boldsymbol{A}\boldsymbol{V} - \boldsymbol{V}^T \boldsymbol{A}\boldsymbol{B}_p + \boldsymbol{V}^T \boldsymbol{A}\boldsymbol{V} - B^2 \tag{Eq 7.1.4}$$

Since $\boldsymbol{B}_p{}^T \boldsymbol{A}\boldsymbol{V}$ is a scalar:

$$\left(\boldsymbol{B}_p{}^T \boldsymbol{A}\boldsymbol{V} \right)^T = \boldsymbol{V}^T \boldsymbol{A}\boldsymbol{B}_p = \boldsymbol{B}_p{}^T \boldsymbol{A}\boldsymbol{V} \tag{Eq 7.1.5}$$

Substituting equation 7.1.5 into equation 7.1.4 and re-arranging gives:

$$r[i] = \boldsymbol{B}_p{}^T \boldsymbol{A}\boldsymbol{B}_p - 2\boldsymbol{B}_p{}^T \boldsymbol{A}\boldsymbol{V} + \boldsymbol{V}^T \boldsymbol{A}\boldsymbol{V} - B^2 \tag{Eq 7.1.6}$$

Expanding equation 7.1.6 for the *i-th* measurement gives:

$$r[i] = \begin{pmatrix} B_{px}[i] \\ B_{py}[i] \\ B_{pz}[i] \end{pmatrix}^T \begin{pmatrix} A_{xx} & A_{xy} & A_{xz} \\ A_{xy} & A_{yy} & A_{yz} \\ A_{xz} & A_{yz} & A_{zz} \end{pmatrix} \begin{pmatrix} B_{px}[i] \\ B_{py}[i] \\ B_{pz}[i] \end{pmatrix}$$

$$-2 \begin{pmatrix} B_{px}[i] \\ B_{py}[i] \\ B_{pz}[i] \end{pmatrix}^T \begin{pmatrix} A_{xx} & A_{xy} & A_{xz} \\ A_{xy} & A_{yy} & A_{yz} \\ A_{xz} & A_{yz} & A_{zz} \end{pmatrix} \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix}$$

$$+ \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix}^T \begin{pmatrix} A_{xx} & A_{xy} & A_{xz} \\ A_{xy} & A_{yy} & A_{yz} \\ A_{xz} & A_{yz} & A_{zz} \end{pmatrix} \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix} - B^2 \tag{Eq 7.1.7}$$

The first term in equation 7.1.7 expands to:

freescale™
semiconductor

$$\begin{pmatrix} B_{px}[i] \\ B_{py}[i] \\ B_{pz}[i] \end{pmatrix}^T \begin{pmatrix} A_{xx} & A_{xy} & A_{xz} \\ A_{xy} & A_{yy} & A_{yz} \\ A_{xz} & A_{yz} & A_{zz} \end{pmatrix} \begin{pmatrix} B_{px}[i] \\ B_{py}[i] \\ B_{pz}[i] \end{pmatrix} = \begin{pmatrix} B_{px}[i] \\ B_{py}[i] \\ B_{pz}[i] \end{pmatrix}^T \begin{pmatrix} A_{xx}B_{px}[i] + A_{xy}B_{py}[i] + A_{xz}B_{pz}[i] \\ A_{xy}B_{px}[i] + A_{yy}B_{py}[i] + A_{yz}B_{pz}[i] \\ A_{xz}B_{px}[i] + A_{yz}B_{py}[i] + A_{zz}B_{pz}[i] \end{pmatrix} \qquad \text{Eq 7.1.8}$$

$$= A_{xx}B_{px}[i]^2 + A_{yy}B_{py}[i]^2 + A_{zz}B_{pz}[i]^2 + 2A_{xy}B_{px}[i]B_{py}[i] + 2A_{xz}B_{px}[i]B_{pz}[i] + 2A_{yz}B_{py}[i]B_{pz}[i] \qquad \text{Eq 7.1.9}$$

The second term in equation 7.1.7 expands to:

$$-2 \begin{pmatrix} B_{px}[i] \\ B_{py}[i] \\ B_{pz}[i] \end{pmatrix}^T \begin{pmatrix} A_{xx} & A_{xy} & A_{xz} \\ A_{xy} & A_{yy} & A_{yz} \\ A_{xz} & A_{yz} & A_{zz} \end{pmatrix} \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix} = -2 \begin{pmatrix} B_{px}[i] \\ B_{py}[i] \\ B_{pz}[i] \end{pmatrix}^T \begin{pmatrix} A_{xx}V_x + A_{xy}V_y + A_{xz}V_z \\ A_{xy}V_x + A_{yy}V_y + A_{yz}V_z \\ A_{xz}V_x + A_{yz}V_y + A_{zz}V_z \end{pmatrix} \qquad \text{Eq 7.1.10}$$

$$= -2B_{px}[i]A_{xx}V_x - 2B_{px}[i]A_{xy}V_y - 2B_{px}[i]A_{xz}V_z$$

$$-2B_{py}[i]A_{xy}V_x - 2B_{py}[i]A_{yy}V_y - 2B_{py}[i]A_{yz}V_z$$

$$-2B_{pz}[i]A_{xz}V_x - 2B_{pz}[i]A_{yz}V_y - 2B_{pz}[i]A_{zz}V_z \qquad \text{Eq 7.1.11}$$

The third term in equation 7.1.7 expands to:

$$\begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix}^T \begin{pmatrix} A_{xx} & A_{xy} & A_{xz} \\ A_{xy} & A_{yy} & A_{yz} \\ A_{xz} & A_{yz} & A_{zz} \end{pmatrix} \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix} = \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix}^T \begin{pmatrix} A_{xx}V_x + A_{xy}V_y + A_{xz}V_z \\ A_{xy}V_x + A_{yy}V_y + A_{yz}V_z \\ A_{xz}V_x + A_{yz}V_y + A_{zz}V_z \end{pmatrix} \qquad \text{Eq 7.1.12}$$

$$= A_{xx}V_x{}^2 + A_{yy}V_y{}^2 + A_{zz}V_z{}^2 + 2A_{xy}V_xV_y + 2A_{xz}V_xV_z + 2A_{yz}V_yV_z \qquad \text{Eq 7.1.13}$$

The full equation for the residual error $r[i]$ from the *i-th* observation is then:

$$r[i] = A_{xx}B_{px}[i]^2 + A_{yy}B_{py}[i]^2 + A_{zz}B_{pz}[i]^2$$

$$+2A_{xy}B_{px}[i]B_{py}[i] + 2A_{xz}B_{px}[i]B_{pz}[i] + 2A_{yz}B_{py}[i]B_{pz}[i]$$

$$-2B_{px}[i]A_{xx}V_x - 2B_{px}[i]A_{xy}V_y - 2B_{px}[i]A_{xz}V_z$$

$$-2B_{py}[i]A_{xy}V_x - 2B_{py}[i]A_{yy}V_y - 2B_{py}[i]A_{yz}V_z$$

$$-2B_{pz}[i]A_{xz}V_x - 2B_{pz}[i]A_{yz}V_y - 2B_{pz}[i]A_{zz}V_z$$

$$+A_{xx}V_x{}^2 + A_{yy}V_y{}^2 + A_{zz}V_z{}^2 + 2A_{xy}V_xV_y + 2A_{xz}V_xV_z + 2A_{yz}V_yV_z - B^2 \qquad \text{Eq 7.1.14}$$

Simplifying and returning to matrix format gives:

$$r[i] = \begin{pmatrix} B_{px}[i]^2 \\ 2B_{px}[i]B_{py}[i] \\ 2B_{px}[i]B_{pz}[i] \\ B_{py}[i]^2 \\ 2B_{py}[i]B_{pz}[i] \\ B_{pz}[i]^2 \\ B_{px}[i] \\ B_{py}[i] \\ B_{pz}[i] \\ 1 \end{pmatrix}^T \begin{pmatrix} A_{xx} \\ A_{xy} \\ A_{xz} \\ A_{yy} \\ A_{yz} \\ A_{zz} \\ -2A_{xx}V_x - 2A_{xy}V_y - 2A_{xz}V_z \\ -2A_{xy}V_x - 2A_{yy}V_y - 2A_{yz}V_z \\ -2A_{xz}V_x - 2A_{yz}V_y - 2A_{zz}V_z \\ A_{xx}V_x{}^2 + 2A_{xy}V_xV_y + 2A_{xz}V_xV_z + A_{yy}V_y{}^2 + 2A_{yz}V_yV_z + A_{zz}V_z{}^2 - B^2 \end{pmatrix} \qquad \text{Eq 7.1.15}$$

Defining the right hand side of equation 7.1.15 to be the solution vector $\boldsymbol{\beta}$ gives:

freescale™
semiconductor

$$\boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \\ \beta_6 \\ \beta_7 \\ \beta_8 \\ \beta_9 \end{pmatrix} = \begin{pmatrix} A_{xx} \\ A_{xy} \\ A_{xz} \\ A_{yy} \\ A_{yz} \\ A_{zz} \\ -2A_{xx}V_x - 2A_{xy}V_y - 2A_{xz}V_z \\ -2A_{xy}V_x - 2A_{yy}V_y - 2A_{yz}V_z \\ -2A_{xz}V_x - 2A_{yz}V_y - 2A_{zz}V_z \\ A_{xx}V_x{}^2 + 2A_{xy}V_xV_y + 2A_{xz}V_xV_z + A_{yy}V_y{}^2 + 2A_{yz}V_yV_z + A_{zz}V_z{}^2 - B^2 \end{pmatrix}$$ Eq 7.1.16

$$= \begin{pmatrix} A_{xx} \\ A_{xy} \\ A_{xz} \\ A_{yy} \\ A_{yz} \\ A_{zz} \\ -2\beta_0 V_x - 2\beta_1 V_y - 2\beta_2 V_z \\ -2\beta_1 V_x - 2\beta_3 V_y - 2\beta_4 V_z \\ -2\beta_2 V_x - 2\beta_4 V_y - 2\beta_5 V_z \\ A_{xx}V_x{}^2 + 2A_{xy}V_xV_y + 2A_{xz}V_xV_z + A_{yy}V_y{}^2 + 2A_{yz}V_yV_z + A_{zz}V_z{}^2 - B^2 \end{pmatrix}$$ Eq 7.1.17

Equation 7.1.15 for the error residual $r[i]$ whose squared sum is to be minimized is then:

$$r[i] = \begin{pmatrix} B_{px}[i]^2 \\ 2B_{px}[i]B_{py}[i] \\ 2B_{px}[i]B_{pz}[i] \\ B_{py}[i]^2 \\ 2B_{py}[i]B_{pz}[i] \\ B_{pz}[i]^2 \\ B_{px}[i] \\ B_{py}[i] \\ B_{pz}[i] \\ 1 \end{pmatrix}^T \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \\ \beta_6 \\ \beta_7 \\ \beta_8 \\ \beta_9 \end{pmatrix}$$ Eq 7.1.18

With the definition of the error residual vector $\boldsymbol{r}$ from M measurements as:

$$\boldsymbol{r} = \begin{pmatrix} r[0] \\ r[1] \\ ... \\ r[M-1] \end{pmatrix}$$ Eq 7.1.19

and $\boldsymbol{X}$ defined as the M by 10 measurement matrix:

$$\boldsymbol{X} = \begin{pmatrix} B_{px}[0]^2 & 2B_{px}[0]B_{py}[0] & ... & B_{py}[0] & B_{pz}[0] & 1 \\ B_{px}[1]^2 & 2B_{px}[1]B_{py}[1] & ... & B_{py}[1] & B_{pz}[1] & 1 \\ ... & ... & ... & ... & ... & 1 \\ B_{px}[M-1]^2 & 2B_{px}[M-1]B_{py}[M-1] & ... & B_{py}[M-1] & B_{pz}[M-1] & 1 \end{pmatrix}$$ Eq 7.1.20

then equation 7.1.18 can be expanded to represent M measurements as:

Technical Note

freescale™
semiconductor

$$r = X\beta \qquad\qquad\qquad\qquad \text{Eq 7.1.21}$$

The model being fitted is the homogeneous model $X\beta = 0$ which can be solved for β using the eigen-decomposition approach described in section 2.

The 10x10 product matrix $X^T X$ whose eigenvectors and eigenvalues are to be determined evaluates to:

$$X^T X = \sum_{i=0}^{M-1} \begin{pmatrix} B_{px}[i]^4 & 2B_{px}[i]^3 B_{py}[i] & 2B_{px}[i]^3 B_{pz}[i] & \dots & B_{px}[i]^2 B_{pz}[i] & B_{px}[i]^2 \\ 2B_{px}[i]^3 B_{py}[i] & 4B_{px}[i]^2 B_{py}[i]^2 & 4B_{px}[i]^2 B_{py}[i] B_{pz}[i] & \dots & 2B_{px}[i] B_{py}[i] B_{pz}[i] & 2B_{px}[i] B_{py}[i] \\ 2B_{px}[i]^3 B_{pz}[i] & 4B_{px}[i]^2 B_{py}[i] B_{pz}[i] & 4B_{px}[i]^2 B_{pz}[i]^2 & \dots & 2B_{px}[i] B_{pz}[i]^2 & 2B_{px}[i] B_{pz}[i] \\ B_{px}[i]^2 B_{py}[i]^2 & 2B_{px}[i] B_{py}[i]^3 & 2B_{px}[i] B_{py}[i]^2 B_{pz}[i] & \dots & B_{py}[i]^2 B_{pz}[i] & B_{py}[i]^2 \\ 2B_{px}[i]^2 B_{py}[i] B_{pz}[i] & 4B_{px}[i] B_{py}[i]^2 B_{pz}[i] & 4B_{px}[i] B_{py}[i] B_{pz}[i]^2 & \dots & 2B_{py}[i] B_{pz}[i]^2 & 2B_{py}[i] B_{pz}[i] \\ B_{px}[i]^2 B_{pz}[i]^2 & 2B_{px}[i] B_{py}[i] B_{pz}[i]^2 & 2B_{px}[i] B_{pz}[i]^3 & \dots & B_{pz}[i]^3 & B_{pz}[i]^2 \\ B_{px}[i]^3 & 2B_{px}[i]^2 B_{py}[i] & 2B_{px}[i]^2 B_{pz}[i] & \dots & B_{px}[i] B_{pz}[i] & B_{px}[i] \\ B_{px}[i]^2 B_{py}[i] & 2B_{px}[i] B_{py}[i]^2 & 2B_{px}[i] B_{py}[i] B_{pz}[i] & \dots & B_{py}[i] B_{pz}[i] & B_{py}[i] \\ B_{px}[i]^2 B_{pz}[i] & 2B_{px}[i] B_{py}[i] B_{pz}[i] & 2B_{px}[i] B_{pz}[i]^2 & \dots & B_{pz}[i]^3 & B_{pz}[i] \\ B_{px}[i]^2 & 2B_{px}[i] B_{py}[i] & 2B_{px}[i] B_{pz}[i] & \dots & B_{pz}[i] & 1 \end{pmatrix}$$

$$\text{Eq 7.1.22}$$

==Since the eigenvalues of $X^T X$ are equal to the fit errors associated with the 10 candidate eigenvector solutions, the required solution vector β is the eigenvector associated with the smallest eigenvalue λ_{min}.==

7.2   Ellipsoid Fit Matrix

The ellipsoid fit matrix A is obtained directly from the first six rows of the solution vector β:

$$A = \begin{pmatrix} A_{xx} & A_{xy} & A_{xz} \\ A_{xy} & A_{yy} & A_{yz} \\ A_{xz} & A_{yz} & A_{zz} \end{pmatrix} = \begin{pmatrix} \beta_0 & \beta_1 & \beta_2 \\ \beta_1 & \beta_3 & \beta_4 \\ \beta_2 & \beta_4 & \beta_5 \end{pmatrix} \qquad\qquad \text{Eq 7.2.1}$$

==The solution eigenvector β is undefined within a multiplicative factor of +/-1 (assuming it's normalized to unit magnitude). A test must therefore be performed on the determinant of the ellipsoid matrix A defined in equation 7.2.1 and the entire solution vector β negated if the determinant is negative. Negating the solution vector β changes the sign of A and ensures a positive determinant.==

For the same reasons as for the 7 element calibration software, the determinant of A is set to 1.0.

7.3   Hard Iron Vector

The last three rows of equation 7.1.16 can be written as:

$$\begin{pmatrix} \beta_6 \\ \beta_7 \\ \beta_8 \end{pmatrix} = -2 \begin{pmatrix} \beta_0 & \beta_1 & \beta_2 \\ \beta_1 & \beta_3 & \beta_4 \\ \beta_2 & \beta_4 & \beta_5 \end{pmatrix} \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix} = -2A \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix} \qquad\qquad \text{Eq 7.3.1}$$

The solution for the hard iron vector V is then:

$$V = \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix} = -\left(\frac{1}{2}\right) \begin{pmatrix} \beta_0 & \beta_1 & \beta_2 \\ \beta_1 & \beta_3 & \beta_4 \\ \beta_2 & \beta_4 & \beta_5 \end{pmatrix}^{-1} \begin{pmatrix} \beta_6 \\ \beta_7 \\ \beta_8 \end{pmatrix} = -\left(\frac{1}{2}\right) A^{-1} \begin{pmatrix} \beta_6 \\ \beta_7 \\ \beta_8 \end{pmatrix} \qquad\qquad \text{Eq 7.3.2}$$

The solution for the hard iron vector V is independent of any sign change or other scaling of the solution vector β as a consequence of the multiplication by the inverse soft iron matrix which cancels the scaling.

7.4   Inverse Soft Iron Matrix

The inverse soft iron matrix W^{-1} is computed from the square root of the symmetric matrix A whose components are provided in elements β_1 to β_6 of the solution vector β:

$$W^{-1} = \sqrt{A} = \begin{pmatrix} A_{xx} & A_{xy} & A_{xz} \\ A_{xy} & A_{yy} & A_{yz} \\ A_{xz} & A_{yz} & A_{zz} \end{pmatrix}^{\frac{1}{2}} = \begin{pmatrix} \beta_0 & \beta_1 & \beta_2 \\ \beta_1 & \beta_3 & \beta_4 \\ \beta_2 & \beta_4 & \beta_5 \end{pmatrix}^{\frac{1}{2}}$$

Eq 7.4.1

The matrix square root is calculated using a further eigen-decomposition but this time of the 3x3 ellipsoid matrix A. By definition, the 3x3 matrix Q of the eigenvectors and the 3x3 diagonal matrix Λ of the eigenvalues of A are related by:

$$AQ = Q\Lambda \Rightarrow A = Q\Lambda Q^{-1} \Rightarrow \Lambda = Q^{-1}AQ$$

Eq 7.4.2

The matrix $Q\sqrt{\Lambda}Q^{-1}$ can be shown to be the required square root of A by simple multiplication and using the standard result that the eigenvectors of a symmetric matrix are orthonormal:

$$\left(Q\sqrt{\Lambda}Q^{-1}\right)\left(Q\sqrt{\Lambda}Q^{-1}\right) = Q\sqrt{\Lambda}Q^{-1}Q\sqrt{\Lambda}Q^{-1} = Q\Lambda Q^{-1} = A$$

Eq 7.4.3

The required square root solution for the inverse soft iron matrix is then:

$$W^{-1} = \sqrt{A} = Q\sqrt{\Lambda}Q^{-1} = Q\sqrt{\Lambda}Q^{T}$$

Eq 7.4.4

7.5   Geomagnetic Field Strength

The geomagnetic field strength can be computed from the last component of equation 7.1.16:

$$B^2 = A_{xx}V_x{}^2 + 2A_{xy}V_xV_y + 2A_{xz}V_xV_z + A_{yy}V_y{}^2 + 2A_{yz}V_yV_z + A_{zz}V_z{}^2 - \beta_9$$

Eq 7.5.1

Since the right hand side of equation 6.5.1 may be negated to force a positive determinant for A, it is important that the geomagnetic field calculation is taken from the absolute value of the right hand side.

$$B = \sqrt{\left| A_{xx}V_x{}^2 + 2A_{xy}V_xV_y + 2A_{xz}V_xV_z + A_{yy}V_y{}^2 + 2A_{yz}V_yV_z + A_{zz}V_z{}^2 - \beta_9 \right|}$$

Eq 7.5.2

7.6   Fit Error

The error function E equals the smallest eigenvalue λ_{min} of the product matrix X^TX but is not normalized to the number of measurement points M nor is it normalized to the geomagnetic field strength B. Since $E = r^Tr$ and r has M elements with each element having dimensions B^2, a suitable normalized calibration fit error measure ε is:

$$\varepsilon = \frac{1}{2B^2}\sqrt{\frac{\lambda_{min}}{M}}$$

                                                                                                  Eq 7.6.1