Least Squares Accelerometer Calibration in Precision Measurement Equipment

Paige Thielen, ME535 Spring 2018

Abstract – Various methods of accelerometer calibration can be used to increase the precision of acceleration measurements. The methods tested are two 12-parameter linear least squares optimizations, one using four calibration orientations, one using eight orientations, and two 15-parameter least squares optimizations using eight and 19 calibration orientations. Based on the data gathered, while it is not necessary to change the calibration method currently in use, good results could be obtained from applying a 12-parameter, 8-orientation least squares calibration without significant increase in time required for calibration.

I. Introduction

The system being analyzed results from a project that I worked on for almost two years at my previous job. I was tasked with investigating inconsistencies encountered during calibration which would cause two subsequent test sequences to yield different results. I left the position but I have a lot of working knowledge on the subject so I will use measurements taken during that investigation and the accelerometer manufacturer’s guidelines as a basis for this study. I will compare various least squares calibrations to determine which method would be sufficient for the level of accuracy desired in the equipment. The best possible method is one that requires the least number of calibration positions to compute a model that will provide the desired level of accuracy. The company does not currently use a least squares method for calibrating the equipment and my previous analysis proved that the method of calibration currently in use is sufficiently accurate to provide a measurement resolution within 0.1% pitch with a relatively low calibration time. I will do my best to not reveal any proprietary company information during this report.
The device under test (DUT) is a specialized type of level sensor designed to measure the roll and pitch of an underground drill. To determine the orientation of the drill, each device contains two accelerometers which measure the position of three axes: $x$, $y$, and $z$. During assembly, these accelerometers are inserted into the body of the transmitter at an unknown orientation. The $x$-axis is approximately aligned with the axis of the (cylindrical) measurement device, whereas $y$- and $z$-axes can be rotated at any angle with respect to the 12 o’clock position.

According to the accelerometer manufacturer, “the original factory accelerometer calibration will...be adequate for the vast majority of consumer applications. Manufacturers of premium products looking to obtain improved accuracy from a consumer accelerometer may, however, wish to perform their own calibration either by repeating the calibration performed by the accelerometer manufacturer or by using a more sophisticated calibration model.” The particular suggested calibration models will be discussed in §II.

In order to achieve the advertised level of accuracy of the locating systems, the accelerometers must be finely calibrated after assembly. Before calibration, it is impossible for the transmitter to determine its roll and pitch. After calibration, the transmitter uses a correlation equation determined by the raw accelerometer readings at six calibration locations to calculate roll and pitch.

II. Model

The system is modeled using corrected accelerometer outputs $x_c$, $y_c$, and $z_c$, which are related to the level angle of the sensor using the equation

\[
pitch = \sin^{-1} x_c
\]
\[
roll = \tan^{-1} \frac{y_c}{z_c}
\]

where roll is always positive and is rounded to 24 clock positions. The accelerometer outputs $x_c$, $y_c$, and $z_c$ are corrected using an internal calibration method, which will not be discussed in this report.

III. Data
Measurements for calculating the parameter estimation were collected separately at 21 positions, defined in Table 2. Data for verifying this model was obtained by iterating through one-degree measurement platform positions between -90° and 90°. The measurement device also provided an output of its own raw and corrected accelerometer values at each position for both of these datasets. The “true” roll and pitch measurements for comparison are obtained by the encoders within each motor and are assumed to be accurate to within 0.01 degrees pitch and 1.8 degrees roll. Since the transmitter outputs a roll position in 12 or 24 clock increments, 1.8 degrees is sufficiently precise for this purpose. The roll calculations do not need to be as accurate as pitch since the bore path is subject only to changes in pitch and yaw. Roll and yaw will not be discussed in this model.

The measurement platform is initially set to level as determined by an externally calibrated tilt sensor, and the pitch motor’s encoder resolution is within 0.05°. Each pitch location is measured relative to this zero position. The x, y, and z channels of one DUT were read at the zero position one hundred times to classify the measurement noise in each accelerometer channel. The measurement noise in each channel is assumed to be zero-mean Gaussian white noise based on this analysis. Figure 1 shows the results of this classification. While it appears that the mean of each channel is very slightly biased, these measurements were taken in terms of accelerometer counts, with a resolution of 16384 bits/g. The mean and standard deviation of the measurement noise in each channel is shown in Table 1.
Table 1 Measurement Noise Characterization

<table>
<thead>
<tr>
<th></th>
<th>x channel</th>
<th>y channel</th>
<th>z channel</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean (g)</td>
<td>-1.1482e-05</td>
<td>1.0273e-05</td>
<td>3.1424e-05</td>
</tr>
<tr>
<td>standard deviation (g)</td>
<td>0.0053</td>
<td>0.0061</td>
<td>0.0667</td>
</tr>
</tbody>
</table>

IV. Methods

In the calibration manual\(^1\) for this particular model of accelerometer, Freescale Semiconductor suggests several optimal calibration orientations that can be used to maximize the minimum separation between gravitational field vectors under the condition that the magnitude of the measurement acceleration is equal to g, which is the case in this study. Suggested orientations as defined by the Freescale accelerometer manual are shown in Table 2.

Table 2 Suggested Optimum Calibration Orientations

<table>
<thead>
<tr>
<th>Optimum pitch angles for 8-orientation calibration</th>
<th>Optimum roll angles for 8-orientation calibration</th>
</tr>
</thead>
<tbody>
<tr>
<td>-35°</td>
<td>-45°</td>
</tr>
<tr>
<td>-73°</td>
<td>161°</td>
</tr>
<tr>
<td>5°</td>
<td>17°</td>
</tr>
<tr>
<td>-16°</td>
<td>84°</td>
</tr>
<tr>
<td>16°</td>
<td>-96°</td>
</tr>
<tr>
<td>-5°</td>
<td>-163°</td>
</tr>
<tr>
<td>73°</td>
<td>-18°</td>
</tr>
<tr>
<td>35°</td>
<td>135°</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Optimum pitch angles for 6-orientation calibration</th>
<th>Optimum roll angles for 6-orientation calibration</th>
</tr>
</thead>
<tbody>
<tr>
<td>6°</td>
<td>-55°</td>
</tr>
<tr>
<td>-6°</td>
<td>125°</td>
</tr>
<tr>
<td>20°</td>
<td>-147°</td>
</tr>
<tr>
<td>-20°</td>
<td>33°</td>
</tr>
<tr>
<td>-69°</td>
<td>-128°</td>
</tr>
<tr>
<td>69°</td>
<td>52°</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Optimum pitch angles for 4-orientation calibration</th>
<th>Optimum roll angles for 4-orientation calibration</th>
</tr>
</thead>
<tbody>
<tr>
<td>39°</td>
<td>-158°</td>
</tr>
<tr>
<td>-66°</td>
<td>164°</td>
</tr>
<tr>
<td>18°</td>
<td>66°</td>
</tr>
<tr>
<td>-1°</td>
<td>-44°</td>
</tr>
</tbody>
</table>

\(^1\) http://cache.freescale.com/files/sensors/doc/app_note/AN4399.pdf
<table>
<thead>
<tr>
<th>Optimum pitch angles for 3-orientation calibration</th>
<th>Optimum roll angles for 3-orientation calibration</th>
</tr>
</thead>
<tbody>
<tr>
<td>0°</td>
<td>45°</td>
</tr>
<tr>
<td>-45°</td>
<td>180°</td>
</tr>
<tr>
<td>45°</td>
<td>-90°</td>
</tr>
</tbody>
</table>

The 12-parameter least squares optimization takes the form

\[
G_{12} = \begin{bmatrix} G_{12x} \\ G_{12y} \\ G_{12z} \end{bmatrix} = Wf + V = \begin{bmatrix} W_{xx} & W_{xy} & W_{xz} \\ W_{yx} & W_{yy} & W_{yz} \\ W_{zx} & W_{zy} & W_{zz} \end{bmatrix} \begin{bmatrix} G_{fx} \\ G_{fy} \\ G_{fz} \end{bmatrix} + \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix}
\]

where \(W\) and \(V\) are the parameters to be determined from the least squares optimization, \(\begin{bmatrix} G_{fx} \\ G_{fy} \end{bmatrix}\) is a vector of raw accelerometer readings in \(x\), \(y\), and \(z\), normalized to \(g\), and \(\begin{bmatrix} G_{12x} \\ G_{12y} \\ G_{12z} \end{bmatrix}\) are the corrected accelerometer values after the calibration is applied. Here, \(\begin{bmatrix} G_{12x} \\ G_{12y} \\ G_{12z} \end{bmatrix} \approx \begin{bmatrix} -\sin\theta \\ \cos\theta \sin\phi \\ \cos\theta \cos\phi \end{bmatrix}\), where \(\theta\) is the pitch angle and \(\phi\) is the roll angle. The elements of \(G_{12}\) represent the true \(x\), \(y\), and \(z\) components of the applied gravitational field at each measurement orientation.

The matrix \(X\) of measurements of the independent variables is defined as

\[
X = \begin{bmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_m & y_m & z_m & 1 \end{bmatrix},
\]

where the subscript \(m\) represents the measurement index. The vector \(\beta_x = \begin{bmatrix} W_{xx} \\ W_{xy} \\ W_{xz} \end{bmatrix}\) and its \(y\) and \(z\) counterparts, are calculated using least squares. The complete equation is

\[
\beta_x = (X^TX)^{-1}X^TY_x \\
\beta_y = (X^TX)^{-1}X^TY_y \\
\beta_z = (X^TX)^{-1}X^TY_z
\]
where the $X$ matrix is the same in each instance and the $Y$ vectors are the $x$, $y$, and $z$ components of the measured values.

The elements of $G_{12}$ will be used to calculate roll and pitch to verify the accuracy of the 12-parameter linear least squares optimizations.

In addition, two sets of 15-parameter least squares optimizations will be performed using the 8 optimal orientations, and once again using a combination of all 21 orientations listed in Table 2 to determine if there is any benefit to the time lost by taking measurements at more calibration positions.

The 15-parameter least squares optimization includes a cubic nonlinearity term and takes the form

$$G_{15} = \begin{bmatrix} G_{15x} \\ G_{15y} \\ G_{15z} \end{bmatrix} = W^T f + V + \Gamma f^3 = \begin{bmatrix} W_{xx} & W_{xy} & W_{xz} \\ W_{yx} & W_{yy} & W_{yz} \\ W_{zx} & W_{zy} & W_{zz} \end{bmatrix} \begin{bmatrix} G_{fx} \\ G_{fy} \\ G_{fz} \end{bmatrix} + \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} + \begin{bmatrix} \Gamma_{xx} & 0 & 0 \\ 0 & \Gamma_{yy} & 0 \\ 0 & 0 & \Gamma_{zz} \end{bmatrix} \begin{bmatrix} G_{f1}^3 \\ G_{f2}^3 \end{bmatrix}$$

where $W$, $V$, and $\Gamma$ are the parameters to be determined from the least squares optimization, $\begin{bmatrix} G_{fx} \\ G_{fy} \\ G_{fz} \end{bmatrix}$ is a vector of raw accelerometer readings in $x$, $y$, and $z$, normalized to $g$, and $\begin{bmatrix} G_{15x} \\ G_{15y} \\ G_{15z} \end{bmatrix}$ are the corrected accelerometer values after the calibration is applied. Here again, $\begin{bmatrix} G_{15x} \\ G_{15y} \\ G_{15z} \end{bmatrix} \approx \begin{bmatrix} -\sin\theta \\ \cos\theta\sin\phi \\ \cos\theta\cos\phi \end{bmatrix}$, where $\theta$ is the pitch angle and $\phi$ is the roll angle. The elements of $G_{15}$ represent the true $x$, $y$, and $z$ components of the applied gravitational field at each measurement orientation.

The matrix $X_x$ of measurements of the independent variables is defined as

$$X_x = \begin{bmatrix} x_1 & y_1 & z_1 & 1 & x_1^3 \\ x_2 & y_2 & z_2 & 1 & x_2^3 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_m & y_m & z_m & 1 & x_m^3 \end{bmatrix}$$
where the subscript \( m \) represents the measurement index. \( X_y \) and \( X_z \) are defined similarly. The vector \( \beta_x = \begin{bmatrix} W_{xx} \\ W_{xy} \\ W_{xz} \\ V_x \\ \Gamma_{xx} \end{bmatrix} \)
is again calculated using least squares. However, while \( Y \) is defined as before, the measurement matrix \( X \) is now different for each \( \beta \). The complete equation is

\[
\beta_x = (X_x^T X_x)^{-1} X_x^T Y_x \\
\beta_y = (X_y^T X_y)^{-1} X_y^T Y_y \\
\beta_z = (X_z^T X_z)^{-1} X_z^T Y_z.
\]

V. Results and Conclusions

The least squares estimates all lie on approximately the same line, but originally there was a noticeable offset between this line and the lines that depict the measured pitch position and the pitch position calculated internally to the transmitter using the current calibration method. It was determined after the data for this study was collected and the study was performed that when the measurement device outputs its calibrated position, there are two calibrations being applied to the final values that, understandably, are not applied to the truly “raw” accelerometer outputs. There is an initial temperature correction that is applied to the raw values before being corrected with the position correction discussed above.

In the absence of these temperature calibration parameters, and without the ability to go find them, it is impossible to determine an accurate comparison for the validity of the least squares estimates. For the purpose of this report, to try and determine the results under more ideal conditions, I have made an attempt to correct the outputs for the least squares estimate to see if applying those same temperature correction values to the least squares estimates would result in better matches of actual position versus current calibration method versus the different least squares methods.
From my experience in designing this and other calibration equipment at this company, I recall that the largest effect of this temperature calibration is the application of a linear offset. Thus, I have estimated this offset as the difference between the “zero” reading of the actual calibrated value and the mean of the zero readings of the least squares estimated values. Then I applied this offset to the calculated values in the least squares estimates to attempt to simulate the results of applying the temperature correction.

The RMS errors resulting from the various corrected methods are shown in Table 3. The least squares estimates were all very close to each other and were able to somewhat accurately represent the actual trajectory of the pitch platform.

<table>
<thead>
<tr>
<th>Current Calibration Method</th>
<th>12-parameter least squares, 8 orientations</th>
<th>12-parameter least squares, 4 orientations</th>
<th>15-parameter least squares, 8 orientations</th>
<th>15-parameter least squares, 21 orientations</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0074</td>
<td>0.0133</td>
<td>0.0112</td>
<td>0.0140</td>
<td>0.0159</td>
</tr>
</tbody>
</table>

It was suggested in the Freescale manual that a 4-position calibration would be sufficient for a 12-parameter least squares optimization since the number of parameters to be calculated exactly equals the number of measurements available (three channels in four positions). This is somewhat evident from the results, as the 12-parameter least squares optimization at four positions performed the best in terms of RMS error (other than the current method). However, as seen in Figure 2, 15 parameters at eight orientations and 12 parameters at eight orientations most closely matched the results of the currently used calibration method.
Using 12 parameters at four orientations outperformed the 15-parameter calibration at 21 orientations, which obviously would have been much too time-consuming to reasonably pursue. The maximum reasonable number of measurements that can be taken during each calibration cycle is eight, since the company produces and calibrates many transmitters each day. The current calibration and verification cycle (which uses six positions) takes just under four minutes and should not be increased unless absolutely necessary. Based on these results, there is no compelling reason to change the calibration model. However, if time could be saved by using only four orientations without significant negative effect in the measurements, it might be worth considering, if only for lower-end models. I would suggest that the company repeat this test on more than a single DUT with the availability of the temperature corrections to verify this. In addition, the twelve-parameter model would require the least amount of change to the internal code on the measurement device itself, which has limited space allocated to storing calibration coefficients. The current model also uses twelve parameters, although not calculated by the same method.
Figure 3 shows the comparison of the five different methods in 181 pitch locations at a fixed roll position. The graph is zoomed in to zero degrees for easier visualization of the results in the most important part of the trajectory.
% Paige Thielen
% ME535 Final Project
% The following is the code used during this analysis.

clear all, close all, clc

% Accelerometer resolution (bits/g)
N = 16384;

%% Characterize noise
% Take 100 samples at zero position
noiseclassdata = csvread('100points.csv');
t = -3:0.001:3;

xdata = noiseclassdata(:,3);
xnoise = xdata - median(xdata);
bx = mean(xnoise);
varx = var(xnoise);
ax = 1/(sqrt(varx*2*pi));
cx = sqrt(varx);
gaussx = ax*exp(-(t - bx).^2/(2*cx^2));

ydata = noiseclassdata(:,4);
ynoise = ydata - median(ydata);
by = mean(ynoise);
vary = var(ynoise);
ay = 1/(sqrt(vary*2*pi));
cy = sqrt(vary);
gaussy = ay*exp(-(t - by).^2/(2*cy^2));

zdata = noiseclassdata(:,5);
znoise = zdata - median(zdata);
bz = mean(znoise);
varz = var(znoise);
az = 1/(sqrt(varz*2*pi));
cz = sqrt(varz);
gaussz = az*exp(-(t - bz).^2/(2*cz^2));

figure(1)
plot(t,gaussx,t,gaussy,t,gaussz)
legend('x channel noise','y channel noise','z channel noise','Location','Best')
title('Accelerometer x, y, and z noise characterization')

% Normalize mean of error
bx = bx/N;
by = by/N;
bz = bz/N;

stddevx = sqrt(varx/N);
stddevy = sqrt(vary/N);
stddevz = sqrt(varz/N);
%%
% Import all raw data
alldata = csvread('rollpitchdata.csv',1);

pitchset = alldata(:,1);
rollset = alldata(:,2);

pitchread = alldata(:,3);
rollread = alldata(:,4);

xread = alldata(:,5);
xreadg = xread/N;
yread = alldata(:,6);
yreadg = yread/N;
zread = alldata(:,7);
zreadg = zread/N;

res_sq = (pitchread - pitchset).^2;
rms = sqrt(sum(res_sq)/181);

%%
% Import data from 8-orientation calibration
orient8 = csvread('8point.csv');
xread8 = orient8(2:9,3:5);
roll8 = (orient8(2:9,1))*pi/180;
pitch8 = orient8(2:9,2)*pi/180;

% Import data from 6-orientation calibration
orient6 = csvread('6point.csv');
xread6 = orient6(2:7,3:5);
roll6 = (orient6(2:7,1))*pi/180;
pitch6 = orient6(2:7,2)*pi/180;

% Import data from 4-orientation calibration
orient4 = csvread('4point.csv');
xread4 = orient4(2:5,3:5);
roll4 = (orient4(2:5,1))*pi/180;
pitch4 = orient4(2:5,2)*pi/180;

% Import data from 3-orientation calibration
orient3 = csvread('3point.csv');
xread3 = orient3(2:4,3:5);
roll3 = (orient3(2:4,1))*pi/180;
pitch3 = orient3(2:4,2)*pi/180;

% Perform 12-parameter least squares optimization with 8 measurement
% orientations (M = 8) and then with 4 measurements
% Normalize measurements
% X is the matrix of M accelerometer measurements
% X = [x(1) y(1) z(1) 1;  
%      x(2) y(2) z(2) 1;  
%      ...  
%      x(M) y(M) z(M) 1];
X12LS8 = [xread8/N ones(8,1)];
X12LS4 = [xread4/N ones(4,1)];
% Y values
% Yx is the vector of the true x components (in g) of the applied gravitational
% field at the M measurement orientations
YxLS8 = -sin(pitch8);
YxLS4 = -sin(pitch4);
% Yy is the vector of the true y components (in g) of the applied gravitational
% field at the M measurement orientations
YyLS8 = cos(pitch8).*sin(roll8);
YyLS4 = cos(pitch4).*sin(roll4);
% Yz is the vector of the true z components (in g) of the applied gravitational
% field at the M measurement orientations
YzLS8 = cos(pitch8).*cos(roll8);
YzLS4 = cos(pitch4).*cos(roll4);

YLS8 = [YxLS8';
        YyLS8';
        YzLS8'];

YLS4 = [YxLS4';
        YyLS4';
        YzLS4'];

% For the combined orientation model (19 points):
YxLSAll = [-sin(pitch8);
            -sin(pitch6);
            -sin(pitch4);
            -sin(pitch3)];

YyLSAll = [cos(pitch8).*sin(roll8);
            cos(pitch6).*sin(roll6);
            cos(pitch4).*sin(roll4);
            cos(pitch3).*sin(roll3)];

YzLSAll = [cos(pitch8).*cos(roll8);
            cos(pitch6).*cos(roll6);
            cos(pitch4).*cos(roll4);
            cos(pitch3).*cos(roll3)];

YLSAll = [YxLSAll';
          YyLSAll';
          YzLSAll'];

% 12 parameter least squares optimization (linear) with 8
% calibration orientations
Bx12LS8 = inv(X12LS8'*X12LS8)*X12LS8'*YxLS8;
By12LS8 = inv(X12LS8'*X12LS8)*X12LS8'*YyLS8;
Bz12LS8 = inv(X12LS8'*X12LS8)*X12LS8'*YzLS8;
B12LS8 = [Bx12LS8  By12LS8  Bz12LS8];

W12LS8 = B12LS8(1:3,:);
V12LS8 = B12LS8(4:,:);''

% 12 parameter least squares optimization (linear) with 4
% calibration orientations
Bx12LS4 = inv(X12LS4'*X12LS4)*X12LS4'*YxLS4;
By12LS4 = inv(X12LS4'*X12LS4)*X12LS4'*YyLS4;
Bz12LS4 = inv(X12LS4'*X12LS4)*X12LS4'*YzLS4;
B12LS4 = [Bx12LS4  By12LS4  Bz12LS4];

W12LS4 = B12LS4(1:3,:);
V12LS4 = B12LS4(4,:);'

%%
% % Perform 15-parameter linear least squares optimization with 8
% measurement orientations
Xx15LS8 = [xread8/N ones(8,1) (xread8(:,1)/N).^3];
Xy15LS8 = [xread8/N ones(8,1) (xread8(:,2)/N).^3];
Xz15LS8 = [xread8/N ones(8,1) (xread8(:,3)/N).^3];

% Can the model be improved much by combining the data from all unique
% calibration orientations?
Xx15LSAll = [xread8/N ones(8,1) (xread8(:,1)/N).^3;
xread6/N ones(6,1) (xread6(:,1)/N).^3;
xread4/N ones(4,1) (xread4(:,1)/N).^3;
xread3/N ones(3,1) (xread3(:,1)/N).^3];

Xy15LSAll = [xread8/N ones(8,1) (xread8(:,2)/N).^3;
xread6/N ones(6,1) (xread6(:,2)/N).^3;
xread4/N ones(4,1) (xread4(:,2)/N).^3;
xread3/N ones(3,1) (xread3(:,2)/N).^3];

Xz15LSAll = [xread8/N ones(8,1) (xread8(:,3)/N).^3;
xread6/N ones(6,1) (xread6(:,3)/N).^3;
xread4/N ones(4,1) (xread4(:,3)/N).^3;
xread3/N ones(3,1) (xread3(:,3)/N).^3];

% Y values are defined above
% Perfom least squares optimization with cubic nonlinearity term
% Beta (B)
Bx15LS8 = inv(Xx15LS8'*Xx15LS8)*Xx15LS8'*yLS8;
By15LS8 = inv(Xy15LS8'*Xy15LS8)*Xy15LS8'*yLS8;
Bz15LS8 = inv(Xz15LS8'*Xz15LS8)*Xz15LS8'*yLS8;
B15LS8 = [Bx15LS8  By15LS8  Bz15LS8];

Bx15LSAll = inv(Xx15LSAll'*Xx15LSAll)*Xx15LSAll'*yLSAll;
By15LSAll = inv(Xy15LSAll'*Xy15LSAll)*Xy15LSAll'*yLSAll;
Bz15LSAll = inv(Xz15LSAll'*Xz15LSAll)*Xz15LSAll'*yLSAll;
B15LSAll = [Bx15LSAll  By15LSAll  Bz15LSAll];

W15LS8 = B15LS8(1:3,:);
W15LSAll = B15LSAll(1:3,:);

V15LS8 = B15LS8(4,:);
V15LSAll = B15LSAll(4,:);

% Gamma (G)
Gam15LS8 = diag(B15LS8(5,:));
Gam15LSAll = diag(B15LSAll(5,:));

%%
% Test all models on 181 measurement points at 12 o'clock roll (0) and
% various pitch angles (from -90 to 90)
% Compare 4- and 8- orientation calibrations. According to documentation,
% points should be sufficient for a 12-parameter calibration. Also compare
% 15-parameter calibration with 8 orientations and see if combining all
% calibration orientations with 15 parameters is enough of an improvement
% to justify the extra time.
Xall = [xreadg yreadg zreadg];
pitch12LS8 = zeros(181,1);
roll12LS8 = zeros(181,1);
pitch12LS4 = zeros(181,1);
roll12LS4 = zeros(181,1);
for i=1:181
    theta = pitchset(i);
    phi = rollset(i);
    Yact = [-sin(theta)';
        (cos(theta).*sin(phi))';
        (cos(theta).*cos(phi))'];
    Yall(:,i) = Yact;
    Xi = Xall(i,:');
    Yi12_8 = W12LS8*Xi + V12LS8;
    Yi4 = W12LS4*Xi + V12LS4;
    Yi15_8 = W15LS8*Xi + V15LS8' + Gam15LS8*Xi.^3;
    Yi15_All = W15LSAll*Xi + V15LSAll' + Gam15LSAll*Xi.^3;
    pitch12LS8(i) = real(asin(Yi12_8(1))); %pitch in radians
    pitch12LS4(i) = real(asin(Yi4(1))); %pitch in radians
    pitch15LS8(i) = real(asin(Yi15_8(1))); %pitch in radians
    pitch15LSAll(i) = real(asin(Yi15_All(1))); %pitch in radians
end

% Simulate temperature correction by applying an offset determined by the
% difference in the means of the LS estimates and the actual calibrated
% readings at zero position
zero_readings = [pitch12LS4(91);
pitch12LS8(91);
pitch15LS8(91);
pitch15LSAll(91)];
offset = pitchread(91) - mean(zero_readings);
pitch12LS4 = pitch12LS4 + offset;
pitch12LS8 = pitch12LS8 + offset;
pitch15LS8 = (pitch15LS8 + offset)';
pitch15LSAll = (pitch15LSAll + offset)';

% Residuals
res12LS8 = (pitchset - pitch12LS8).^2;
res12LS4 = (pitchset - pitch12LS4).^2;
res15LS8 = (pitchset - pitch15LS8).^2;
res15LSAll = (pitchset - pitch15LSAll).^2;

% RMS errors
rms12LS8 = sqrt(sum(res12LS8)/181);
rms12LS4 = sqrt(sum(res12LS4)/181);
rms15LS8 = sqrt(sum(res15LS8)/181);
rms15LSAll = sqrt(sum(res15LSAll)/181);
% Plot normalized accelerometer readings
figure(2)
plot(pitchset,xreadg,pitchset,yreadg,pitchset,zreadg)
title('Raw Accelerometer Readings at 12 o'clock Roll, Normalized to 1g')
xlabel('Pitch (radians)')
ylabel('Accelerometer reading (g)')
legend('x','y','z','Location','Best')
axis([min(pitchset) max(pitchset) -1.1 1.1])

% Plot true pitch angle and measured pitch angle
figure(3)
pitchset,pitchset,...
pitchset,pitch12LS4,...
pitchset,pitch12LS8,...
pitchset,pitch15LS8,...
pitchset,pitch15LSAll);
title('Pitch Readings vs True Pitch Using Various Calibration Models')
legend('true pitch',...,'current pitch calibration',...,'12 parameter 4 orientation LS',...,'12 parameter 8 orientation LS',...,'15 parameter 8 orientation LS',...,'15 parameter all orientations LS',...,'Location','Best')
xlabel('pitch setting')
ylabel('pitch readout')
axis([min(pitchset) max(pitchset) min(pitchset) max(pitchset)])

figure(4)
pitchset,pitchread-pitchset,...
pitchset,pitch12LS4-pitchset,...
pitchset,pitch12LS8-pitchset,...
pitchset,pitch15LS8-pitchset,...
pitchset,pitch15LSAll-pitchset);
title('Error in Pitch Readings vs True Pitch Using Various Calibration Models')
legend('current pitch calculation',...,'12 parameter 4 orientation LS',...,'12 parameter 8 orientation LS',...,'15 parameter 8 orientation LS',...,'15 parameter all orientations LS',...,'Location','Best')
xlabel('pitch setting')
ylabel('pitch readout')
xlim([min(pitchset) max(pitchset)])

% zoom in to see the effects around zero (+/- 10 degrees)
zeropitch12LS4 = pitch12LS4(88:93);
zeropitch12LS8 = pitch12LS8(88:93);
zeropitch15LS8 = pitch15LS8(88:93);
zeropitch15LSAll = pitch15LSAll(88:93);
zeropitchset = pitchset(88:93);
zeropitchread = pitchread(88:93);

% Plot true pitch angle and measured pitch angle
figure(5)
plot(zeropitchset,zeropitchset,...
    zeropitchset,zeropitchread,...
    zeropitchset,zeropitch12LS4,...
    zeropitchset,zeropitch12LS8,...
    zeropitchset,zeropitch15LS8,...
    zeropitchset,zeropitch15LSAll);
title('Near-Zero Pitch Readings vs True Pitch Using Various Calibration Models')
legend(['true pitch',...
    'current pitch calibration',...
    '12 parameter 4 orientation LS',..., 
    '12 parameter 8 orientation LS',..., 
    '15 parameter 8 orientation LS',...,
    '15 parameter all orientations LS',...
    'Location','Best'])
xlabel('pitch setting')
ylabel('pitch readout')
axis([min(zeropitchset) max(zeropitchset) min(zeropitchset) max(zeropitchset)])

figure(6)
plot(zeropitchset,zeropitchread-zeropitchset,...
    zeropitchset,zeropitch12LS4-zeropitchset,...
    zeropitchset,zeropitch12LS8-zeropitchset,...
    zeropitchset,zeropitch15LS8-zeropitchset,...
    zeropitchset,zeropitch15LSAll-zeropitchset);
title('Error Near Zero in Pitch Readings vs True Pitch Using Various Calibration Models')
legend(['current pitch calculation',...
    '12 parameter 4 orientation LS',...
    '12 parameter 8 orientation LS',...
    '15 parameter 8 orientation LS',...
    '15 parameter all orientations LS',...
    'Location','Best'])
xlabel('pitch setting')
ylabel('pitch readout')
xlim([min(zeropitchset) max(zeropitchset)])