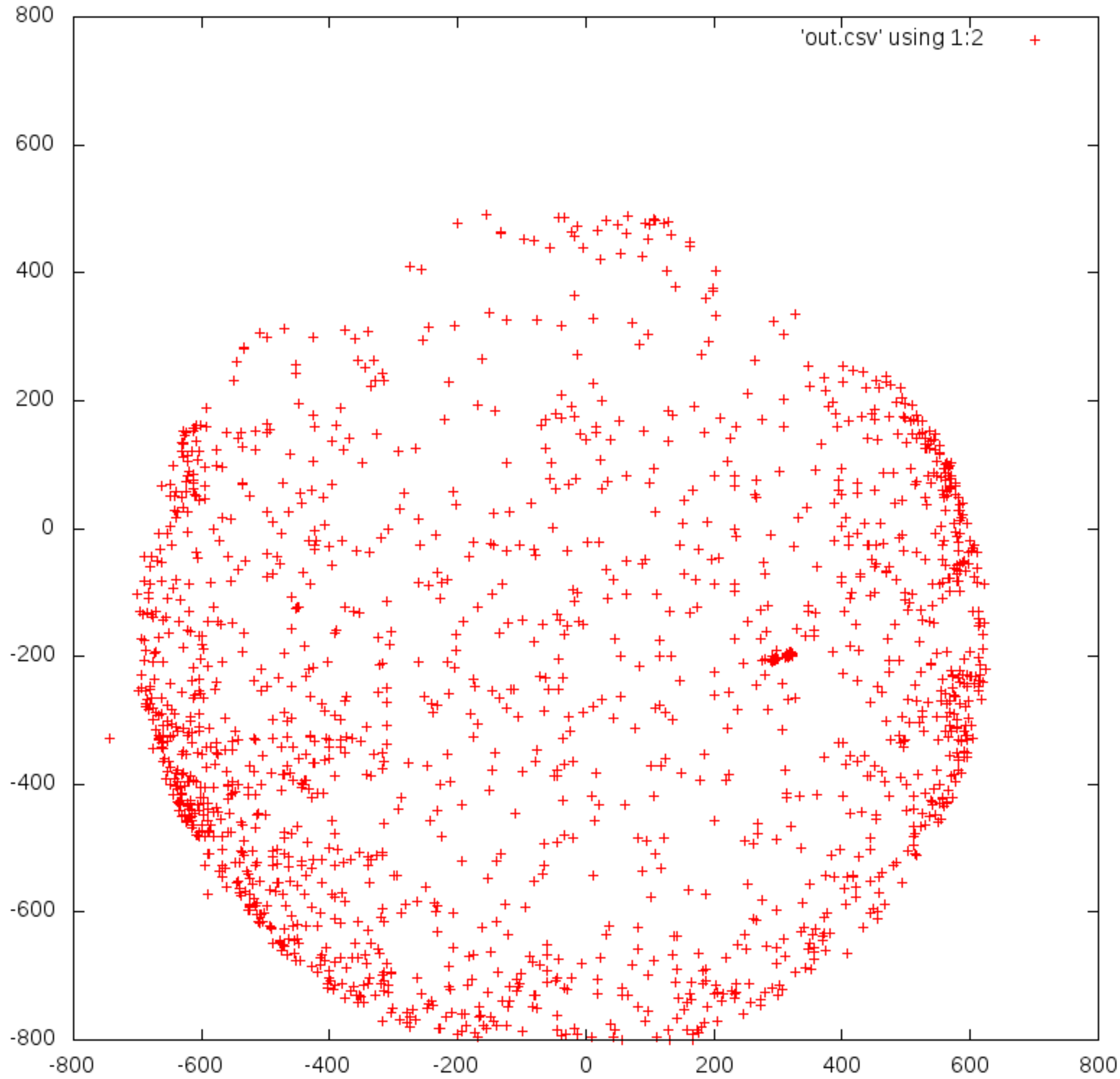
(http://www.camelsoftware.com/firetail/blog/uavs/3-axis-magnetometer-calibration-a-simple-technique-for-hard-soft-errors/)

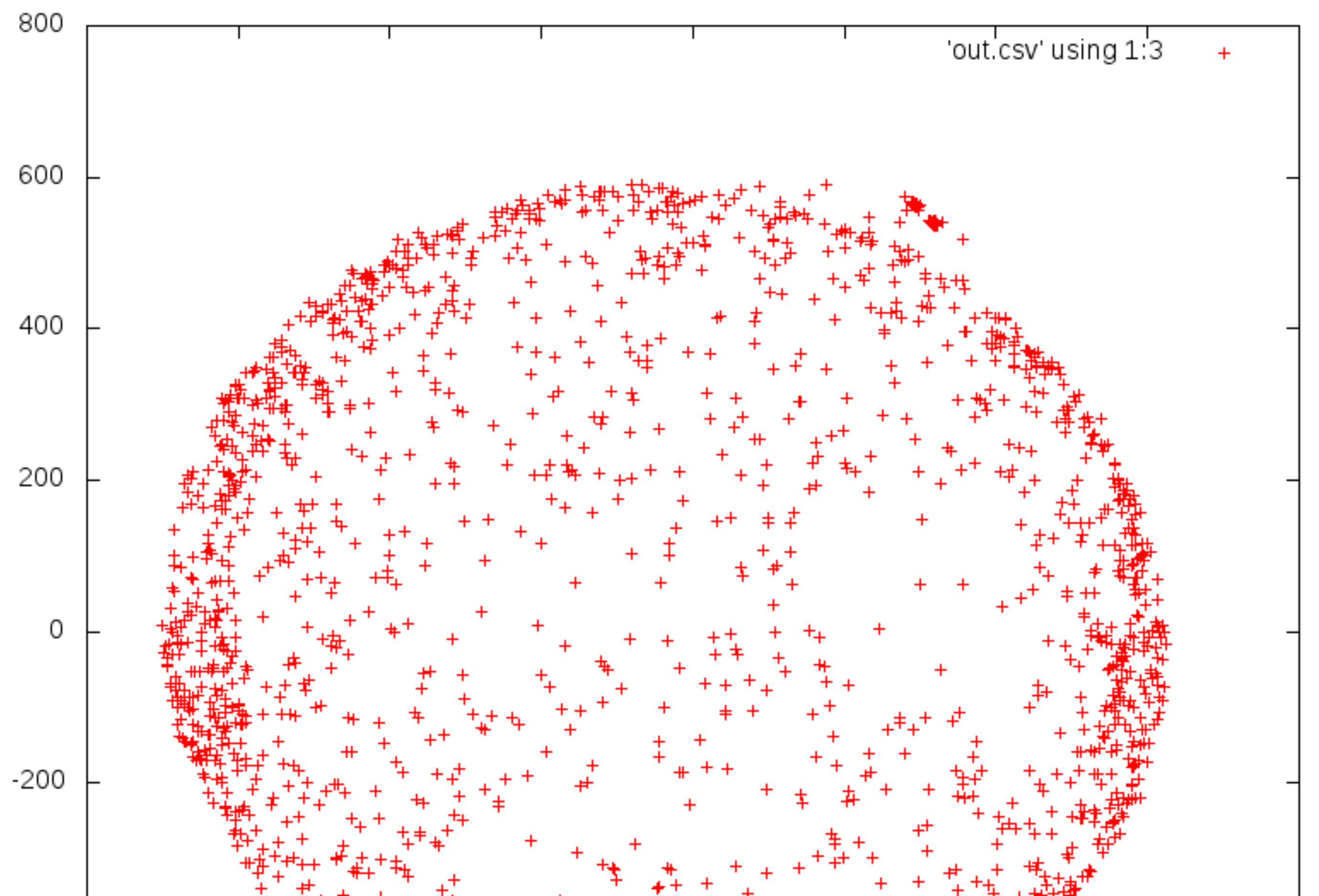
3-AXIS MAGNETOMETER CALIBRATION – A SIMPLE TECHNIQUE FOR HARD & SOFT ERRORS

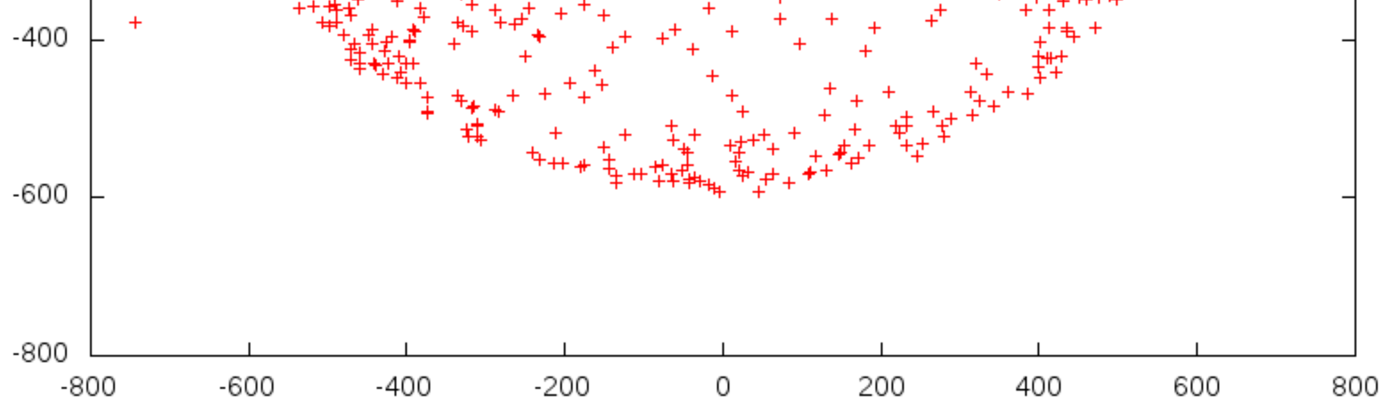
BY CAMEL   11 COMMENTS

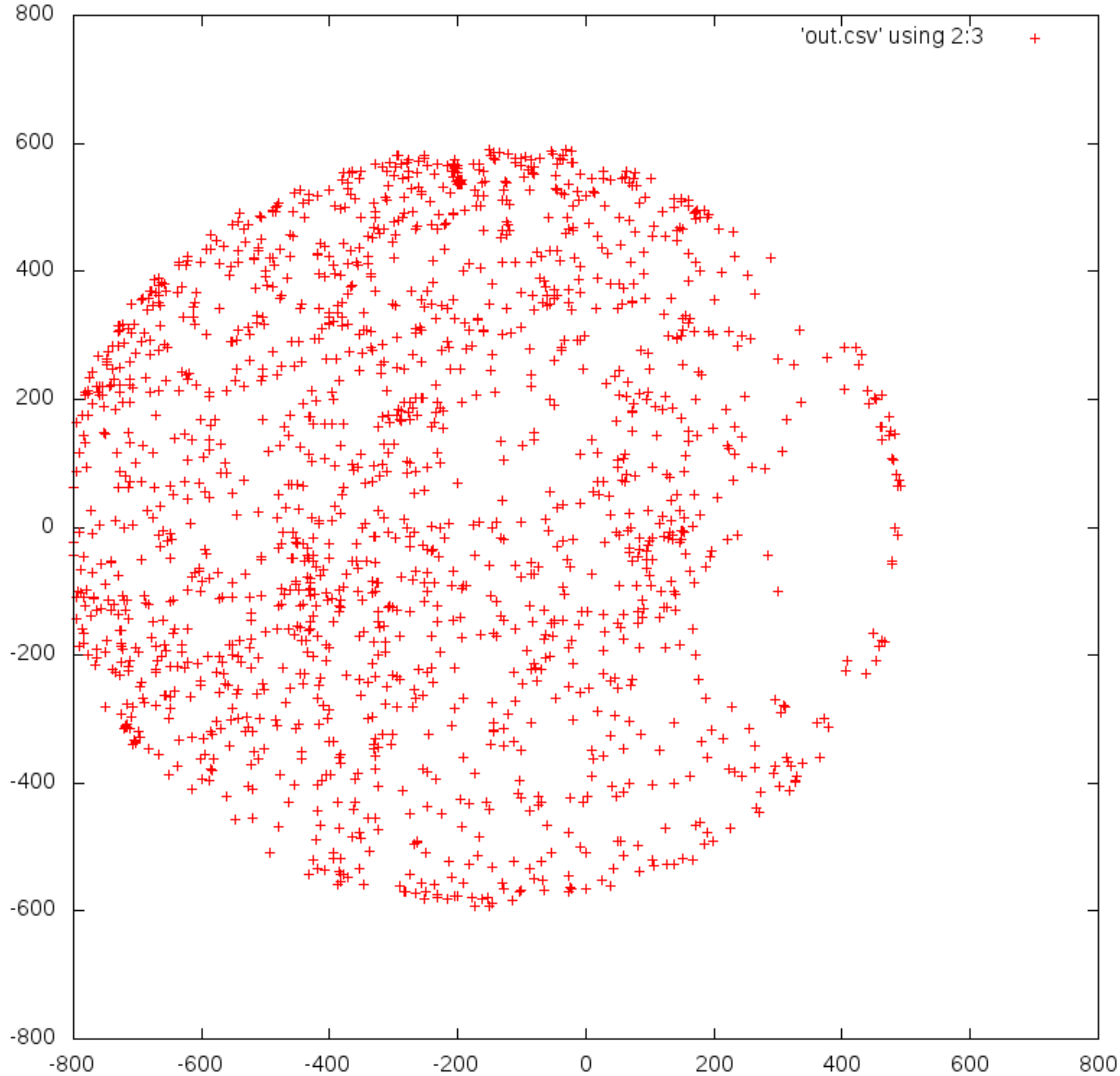
Calibrating magnetometers is critically important. Without a good mag cal, you'll get rubbish out of your magnetometers. Fortunately it isn't that hard to do a basic calibration.

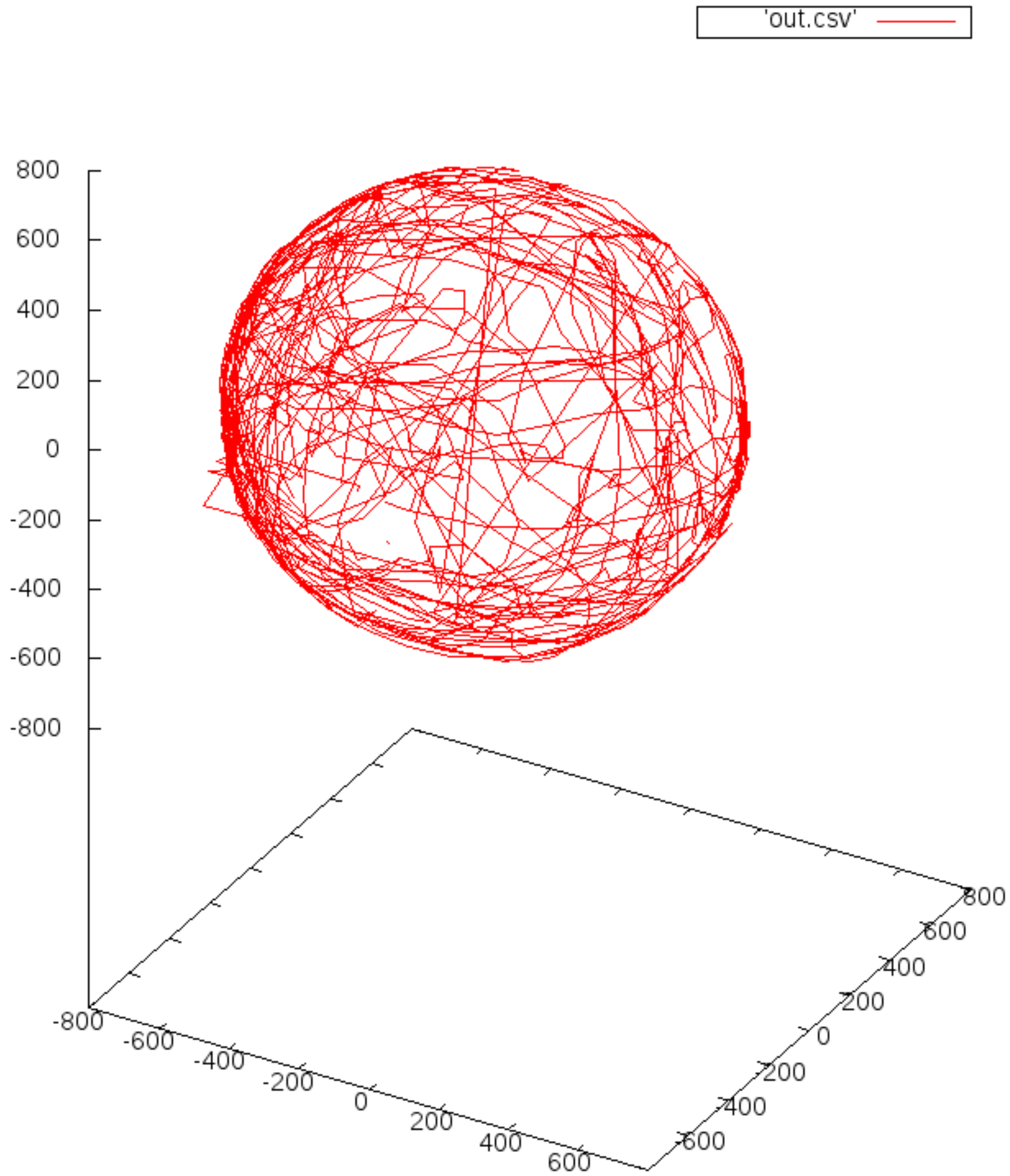
First of all, here are some scatter plots of raw magnetometer data. The sensor I used to create these was an LSM303.


'out.csv' using 1:2
800
600
400
200
0
-200
-400
-600
-800
-800
-600
-400
-200
0
200
400
600
800


'out.csv' using 1:3
800
600
400
200
0
-200


-400
-600
-800
-800
-600
-400
-200
0
200
400
600
800


800
600
400
200
0
-200
-400
-600
-800
'out.csv' using 2:3
-800
-600
-400
-200
0
200
400
600
800



These scatter plots show both hard iron (or bias) and soft iron errors. An ideal plot is circular and is centred on the chart. Hard iron errors cause the measurements to be off centre. Soft iron errors cause the shape to be elliptical rather than circular.

To remove these errors, you need data. Lots of data. So get your magnetometer and move it around until you've got a few thousand samples in as many different orientations as possible.

Hard iron errors are the simplest to remove. Find the maximum and minimum measurements for each axis and average them. This gives the amount of offset for each axis. Every time you take a reading from your magnetometers, you should subtract this offset from each axis.

```
raw_values[i].x() -= (min_x + max_x)/2.0;
raw_values[i].y() -= (min_y + max_y)/2.0;
raw_values[i].z() -= (min_z + max_z)/2.0;
```

Soft iron errors are slightly harder to remove. There are some more advanced techniques that involve calculating rotation matrices and bla bla, but it's computationally expensive (for my brain) and simply scaling each axis to remove the elliptical shape works quite well too.

First the hard iron errors are removed from the maximums and minimum magnetometer vectors. These minimum and maximum vectors are the same as the ones being used to correct for hard iron errors.


```
imu::Vector<3> vmax;
vmax.x() = max_x - ((min_x + max_x)/2.0);
vmax.y() = max_y - ((min_y + max_y)/2.0);
vmax.z() = max_z - ((min_z + max_z)/2.0);

imu::Vector<3> vmin;
vmin.x() = min_x - ((min_x + max_x)/2.0);
vmin.y() = min_y - ((min_y + max_y)/2.0);
vmin.z() = min_z - ((min_z + max_z)/2.0);
```

The average distance from the centre is now calculated. We want to know how far from the centre, so the negative values are inverted.

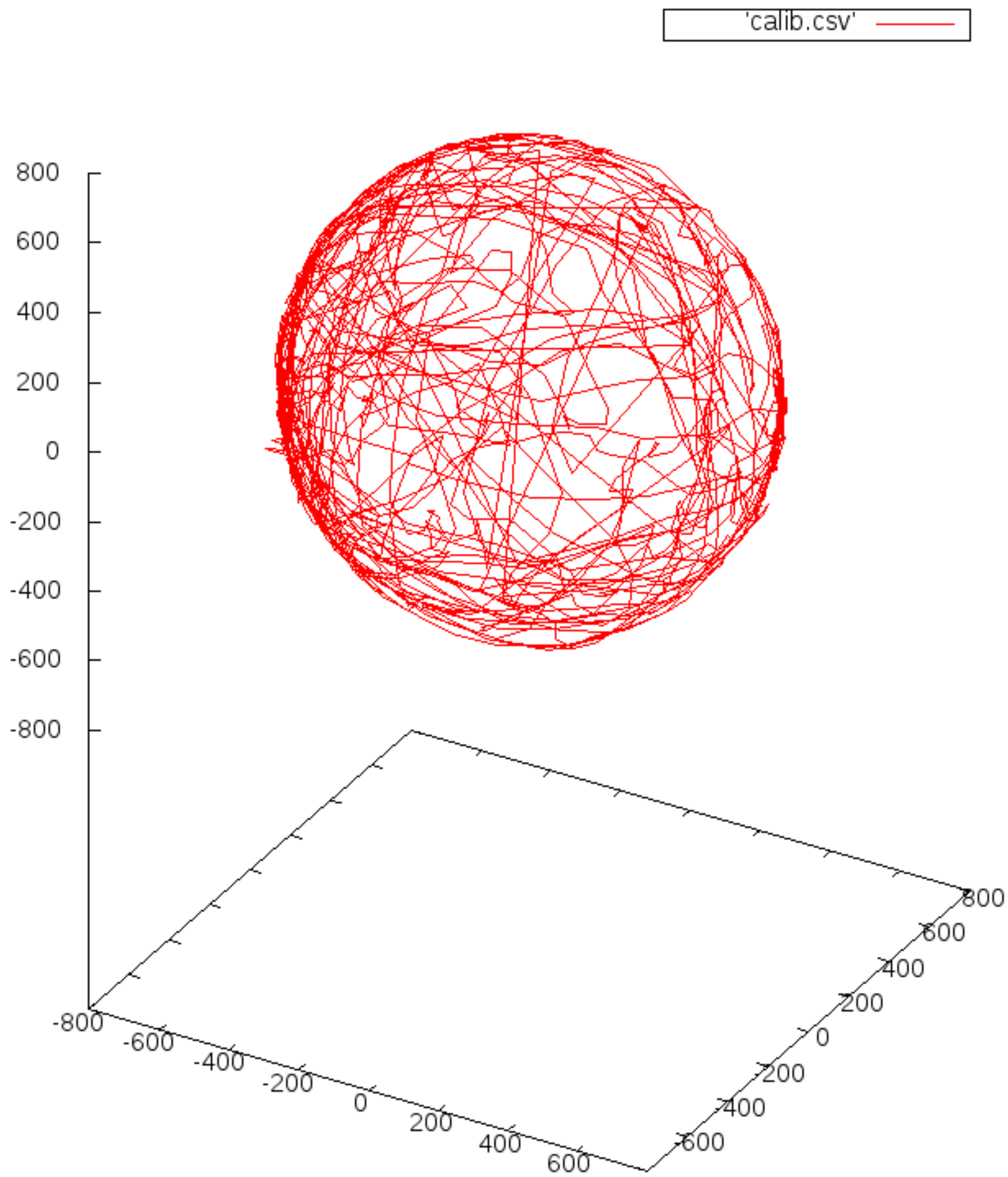
```
imu::Vector<3> avgs;
avgs = vmax + (vmin*-1); //multiply by -1 to make negative values positive
avgs = avgs / 2.0;

The components are now averaged out
float avg_rad = avgs.x() + avgs.y() + avgs.z();
avg_rad /= 3.0;

Finally calculate the scale factor by dividing average radius by average value for that axis.
float x_scale = (avg_rad/avgs.x());
float y_scale = (avg_rad/avgs.y());
float z_scale = (avg_rad/avgs.z());
```

With these scale values we can correct for soft iron errors by multiplying them with the relevant magnetometer axis reading. Here's an example

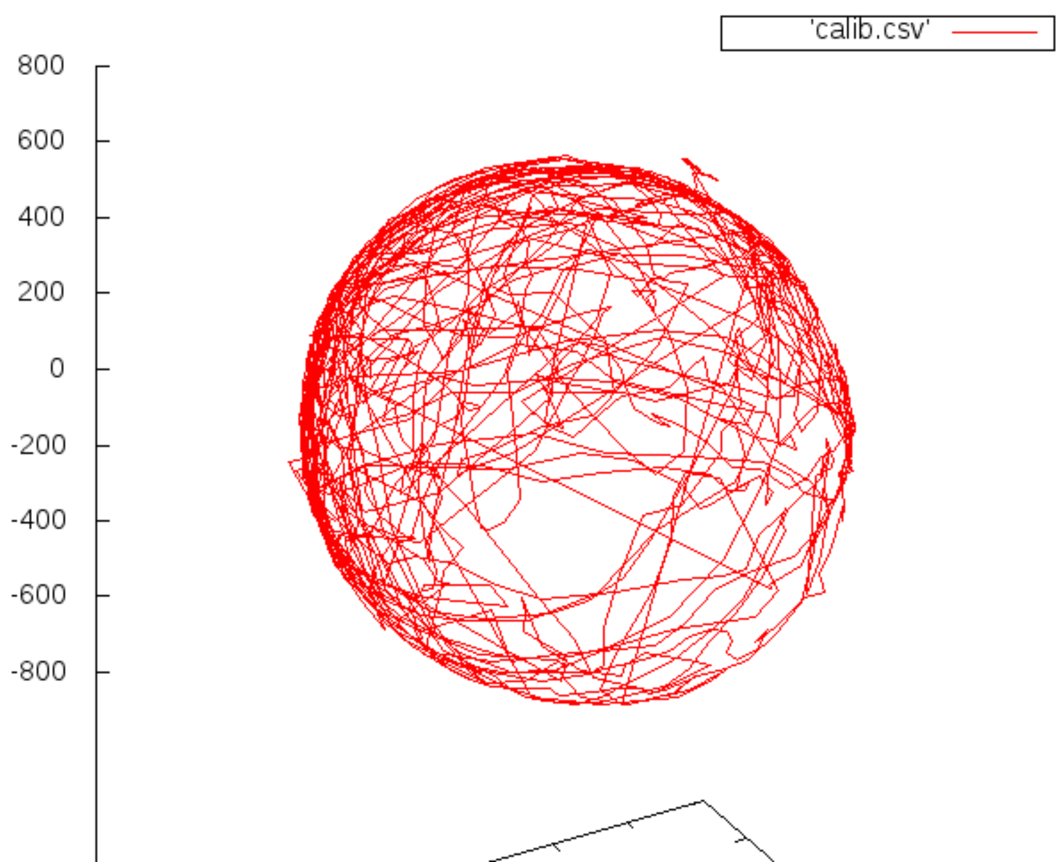
```
imu::Vector<3> mag_reading = magnetometer.read();

mag_reading.x() -= (min_x + max_x)/2.0;
mag_reading.y() -= (min_y + max_y)/2.0;
mag_reading.z() -= (min_z + max_z)/2.0;
```

```
mag_reading.x() *= x_scale;
mag_reading.y() *= y_scale;
mag_reading.z() *= z_scale;
```

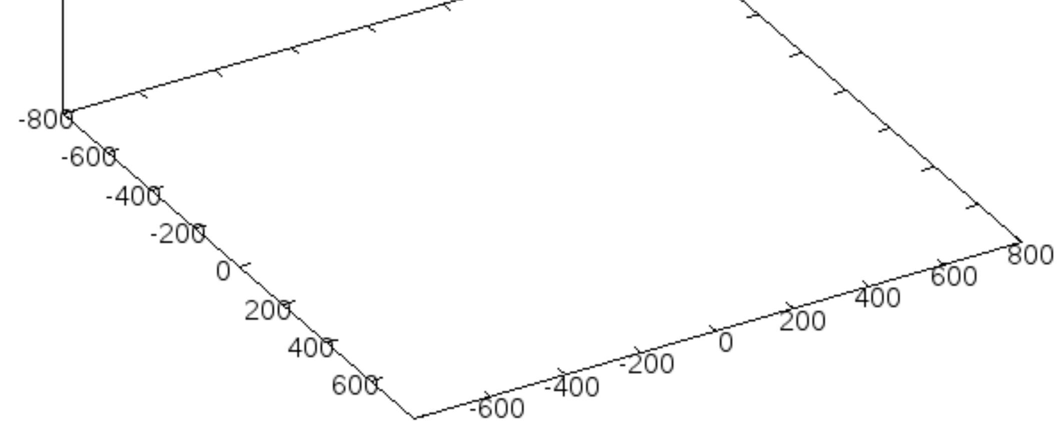
Here are the results using this technique.



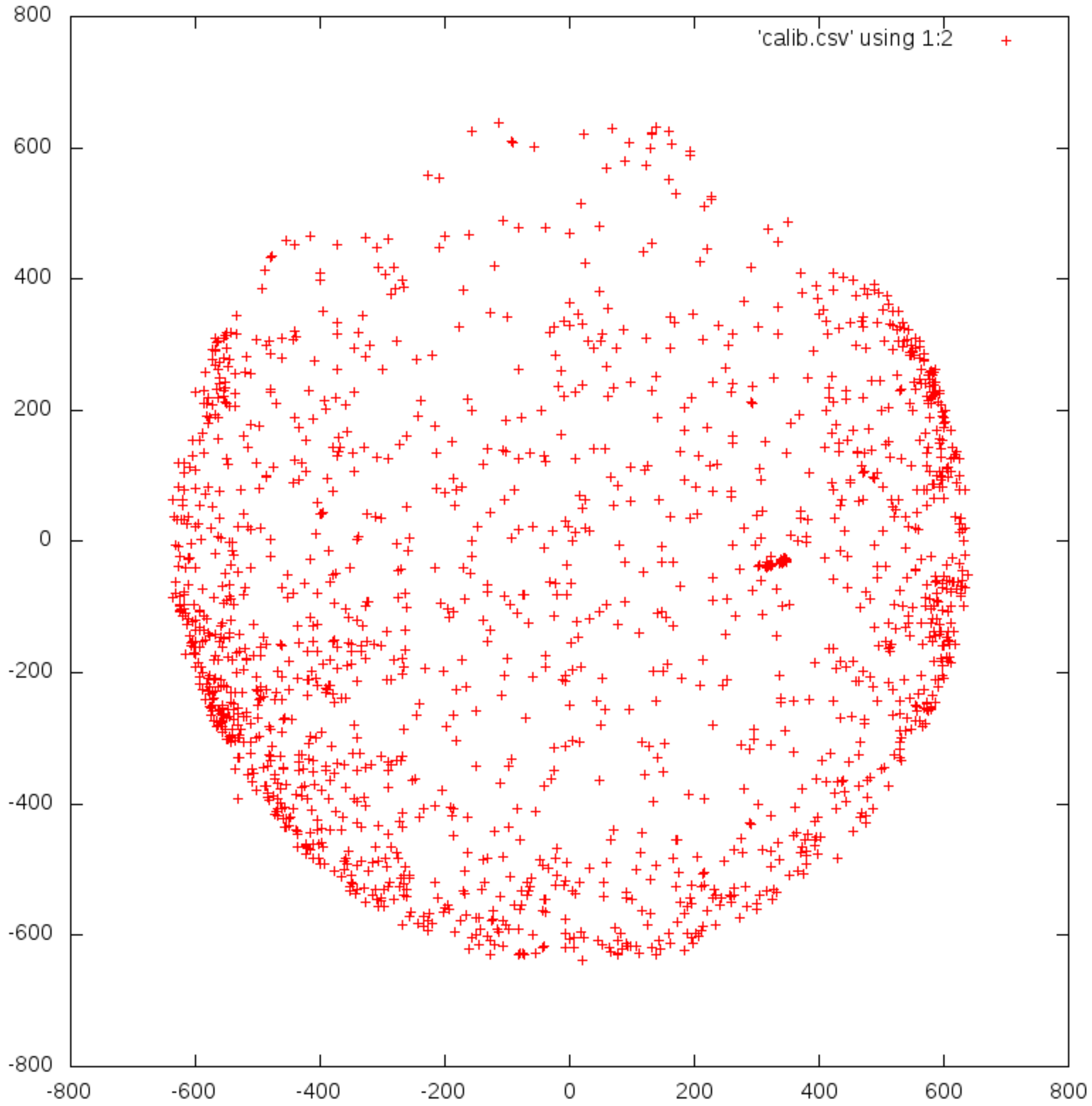
'calib.csv'
800
600
400
200
0
-200
-400
-600
-800
-800
-600
-400
-200
0
200
400
600
800
600
400
200
0
-200
-400
-600



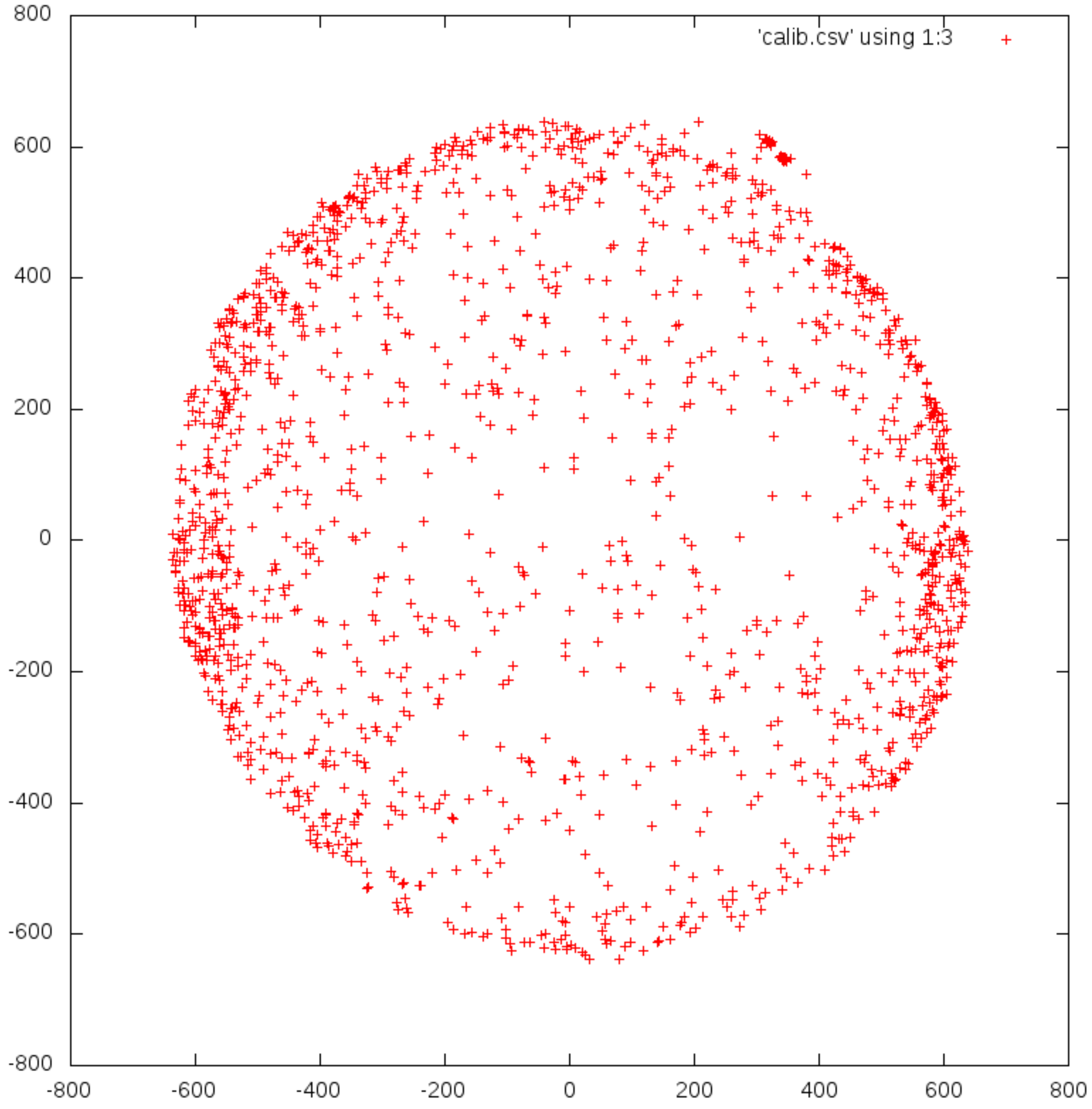
'calib.csv'
800
600
400
200
0
-200
-400
-600
-800

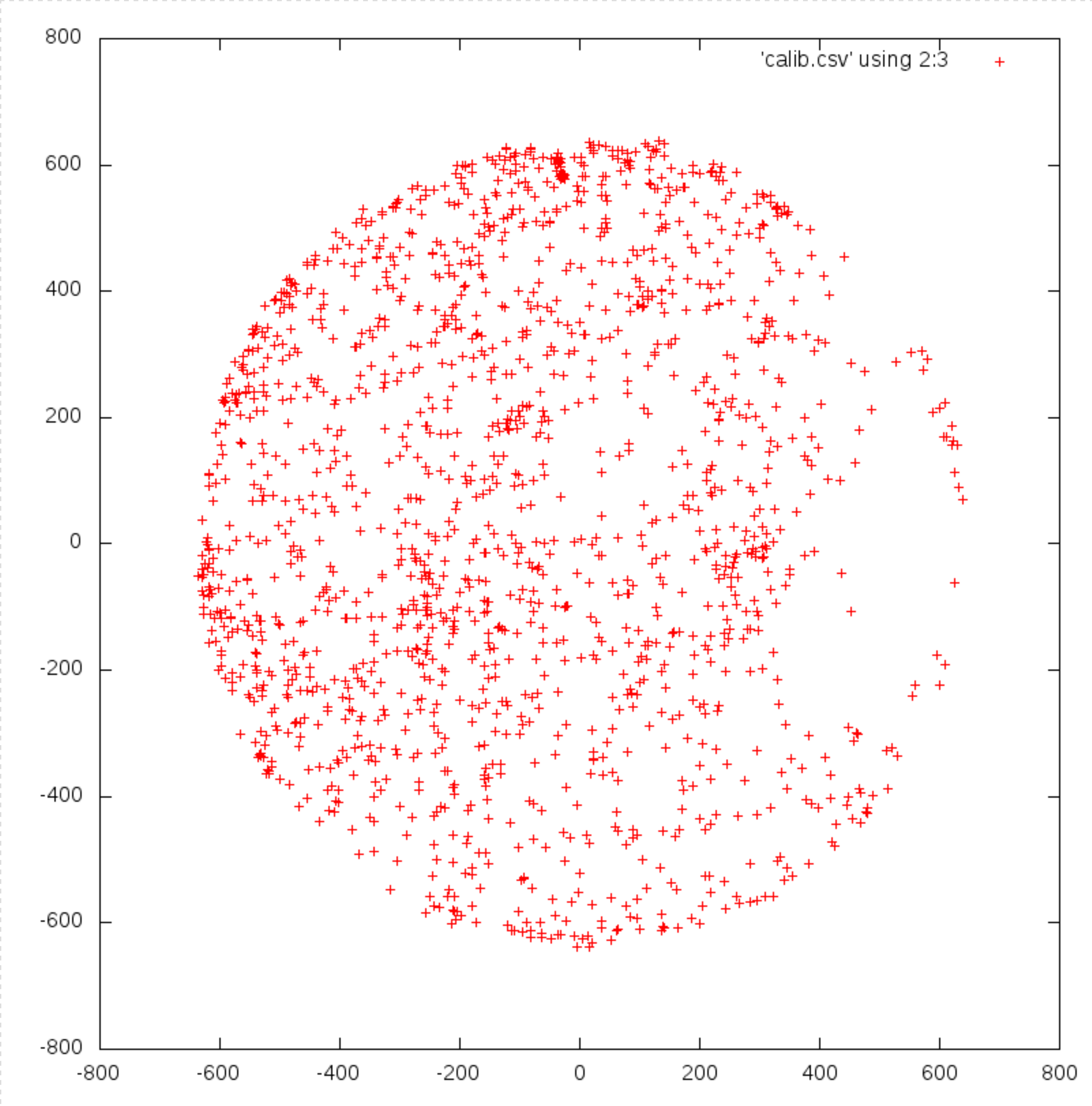


-800
-600
-400
-200
0
200
400
600
-600
-400
-200
0
200
400
600
800



'calib.csv' using 1:2
800
600
400
200
0
-200
-400
-600
-800
-800
-600
-400
-200
0
200
400
600
800


'calib.csv' using 1:3
800
600
400
200
0
-200
-400
-600
-800
-800
-600
-400
-200
0
200
400
600
800



It doesn't use any matrix maths and the magnetometers are now quite accurate. In fact, using a map I can't see any errors. Without any proper test equipment, I'd guess that it's good to +/- 2 degrees.

11 COMMENTS

1. DON HOWES

Posted July 29, 2014 at 10:16 pm | Permalink

I am using the code from MinIMU-9-Arduino-AHRS:

compass.readMag();

magnetom_x = SENSOR_SIGN[6] * compass.m.x;
magnetom_y = SENSOR_SIGN[7] * compass.m.y;
magnetom_z = SENSOR_SIGN[8] * compass.m.z;

// adjust for LSM303 compass axis offsets/sensitivity differences by scaling to +/-0.5 range
c_magnetom_x = (float)(magnetom_x – SENSOR_SIGN[6]*M_X_MIN) / (M_X_MAX – M_X_MIN) – SENSOR_SIGN[6]*0.5
c_magnetom_y = (float)(magnetom_y – SENSOR_SIGN[7]*M_Y_MIN) / (M_Y_MAX – M_Y_MIN) – SENSOR_SIGN[7]*0.5;
c_magnetom_z = (float)(magnetom_z – SENSOR_SIGN[8]*M_Z_MIN) / (M_Z_MAX – M_Z_MIN) – SENSOR_SIGN[8]*0.5;

Do I need to take into account the Sensor Signs? If so where in your equations would I use them.
Also they scale by *0.5 is this the same as the /2 in your method.
Thanks Don

Reply ↓

> CAMEL
>
> Posted July 30, 2014 at 9:49 am | Permalink
>
> You might need to take into account the sensor sign. It's not really clear to me what the purpose of that is. If it's just inverting the value, you might be able to simply hard code that if necessary.
>
> Yes, the *0.5 is doing the same thing as /2. It appears that the code you've posted is for correcting hard iron errors.
>
> Reply ↓

2. MAARTEN

Posted August 17, 2014 at 1:18 am | Permalink

Hi, Thanks for the calibartion code and info. I've got the code working so far. I got nice round circles now.

Do you know how I can use these calibrated mag_reading.x,
mag_reading.y, mag_reading.z() values to get the compass heading?

Thanks!

Reply ↓

CAMEL

Posted August 17, 2014 at 5:14 pm | Permalink

Hello,

There are a number of methods. The one I use is described here. It uses accelerometers and gyros as well to form ar attitude and heading reference system (AHRS).
http://www.camelsoftware.com/firetail/blog/c/imu-maths/

But there are simpler methods out there. Google for Pololus LSM303 library on Github. They've got a decent algorithr that works on just magnetometers + accelerometers.

Reply ↓

3. MAARTEN

Posted August 17, 2014 at 7:05 pm | Permalink

Hi Camel,

That's what I was looking for 🙂

Thanks again

Reply ↓

4. NELSON JENKINS

Posted August 30, 2014 at 9:09 pm | Permalink

Thanks for sharing the calibration tips and tricks. They are useful. I recently purshased a small form of triaxial fluxgate magnetometer from Gemsystems , for my project at College. Initially it was perfectly working since we got it calibrated the But later on, the meter lost its accuracy when we did a reset recently. This piece of information on calibration is of great he I wil get back after trying it people !

Reply ↓

5. COMPASS DEVELOPER

Posted September 8, 2014 at 12:06 pm | Permalink

Soft iron error correction alghoritm is good, it solved my compass problem. You can get good results with one time calibra (= rotate the device along all axes couple of times) that find magnetic min and max values.

Some development boards do not have virtual com port so you have to use device display to get max and min values.

Reply ↓

CAMEL

Posted September 8, 2014 at 12:59 pm | Permalink

Yes, you're right! I said in the post you need lots of samples but really the only important ones are the minimums an maximums of each axis.

Reply ↓

6. CABROUWERS

Posted January 24, 2015 at 3:22 pm | Permalink

Thanks for the code. The calculation of the scaling factor can be simplified. A little algebra or a drawing shows that

avgs_x = (max_x – min_x)/2

Reply ↓

7. JULIAN

Posted April 3, 2015 at 10:17 pm | Permalink

Hey, thank you very much for this easy approach.
In general it works very fine for me but yesterday I had some problems with my measuring that led to bad offsets and scalings.
The main part of the measuring was fine but at one point there was a concentrated package of measurement points. To ge an idea ou can see my (raw) measurements here:http://www.imgbox.de/show/img/1E3WHvBrJP.png Do you know a way o have an idea of how to erase these outliers?

Reply ↓

8. YACKETY

Posted June 11, 2015 at 12:36 am | Permalink

It works great with my sensor! Really appreciate your effort!! Thanks!

Reply ↓

LEAVE A REPLY

Your email address will not be published. Required fields are marked *

NAME *

EMAIL *

WEBSITE

COMMENT

You may use these HTML (HyperText Markup Language) tags and attributes: `<a href="" title=""> <abbr title=""> <acronym title=""> <b> <blockquote cite=""> <cite> <code> <del datetime=""> <em> <i> <q cite=""> <s> <stri <strong>`

Post Comment