MANE 2960/ECSE 2960 MECHATRONICS HARDWARE AND SOFTWARE

Magnetometer: Making a Compass

Part 1: Obtaining Magnetometer Data

Objective:

Obtain data from the HMC5883L magnetometer and use this to create a simple compass.

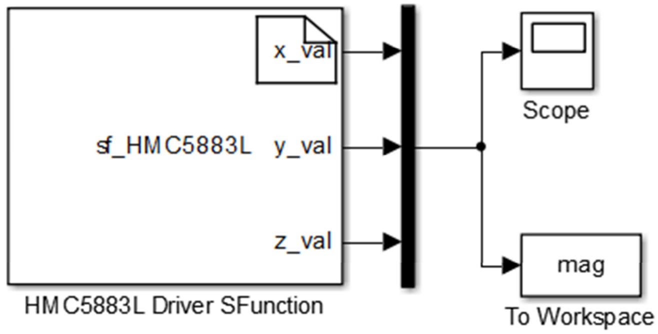
Background Information:

The HMC5883L magnetometer can measure the strength and direction of a magnetic field in 3 directions. The earth acts like a simple bar magnet with magnetic field lines going North to South. By measuring the earth's magnetic field a simple compass can be made.

Simulink Model:

- You will need to have the MinSegLibrary installed or
- The following files will be needed for the Simulink SFunction Driver:
  - HMC5883L_SFunction.slx
  - sf_MPU6050_Driver.mexw32
  - sf_MPU6050_Driver.mexw64
  - sf_MPU6050_Driver.tlc
  - sf_MPU6050_Driver_wrapper.cpp
  - HMC58X3.cpp (HMC5883L library)
  - HMC58X3.h
  - Wire.cpp (Arduino wire library - support for I2C communication)
  - Wire.h
  - twi.c (Arduino wire library - support for I2C communication)
  - twi.h

- Build the following Simulink diagram:



    - The separate x, y and z values are combined with a “mux” block to form a vector
    - The vector is displayed on the scope and written to the workspace in a variable named “mag”
- Run this on the target in external mode and ensure that you see data in the scope as you move sensor around. (you may have to click the ‘autoscale’ button on the scope to see the data)
- Disconnect from the target and verify you can plot the stored data from matlab with
    - plot(mag)

Part 2: Experimental Data and Analysis

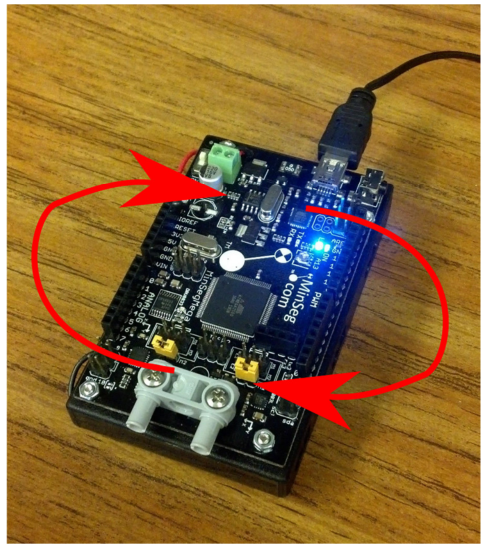
Objectives:

- Obtain data when the sensor is rotated 360 degrees.
- Analyze this data in Matlab:
    - plot the results
    - shift and scale the data to ‘calibrate’ the data
    - use the data to calculate heading

Experimental Procedure:

Collect data as the sensor is rotated 360 degrees:

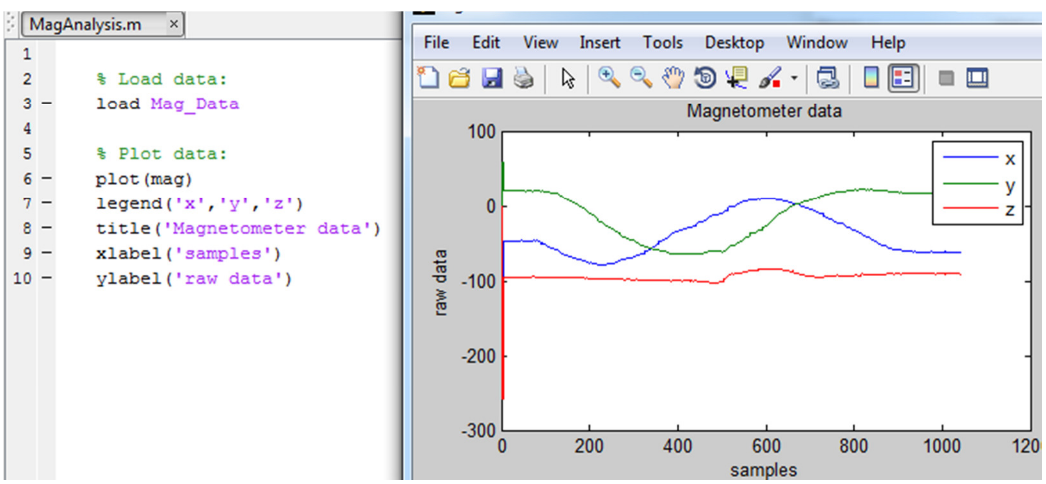
- Place the sensor system on flat surface away from any large ferrous (iron/metal) objects
    - If you system contains batteries that are close to the sensor remove them since they could significantly affect the magnetic field
- Connect and run the Simulink model in external mode to start logging data
- Slowly rotate the sensor around in a complete circle:



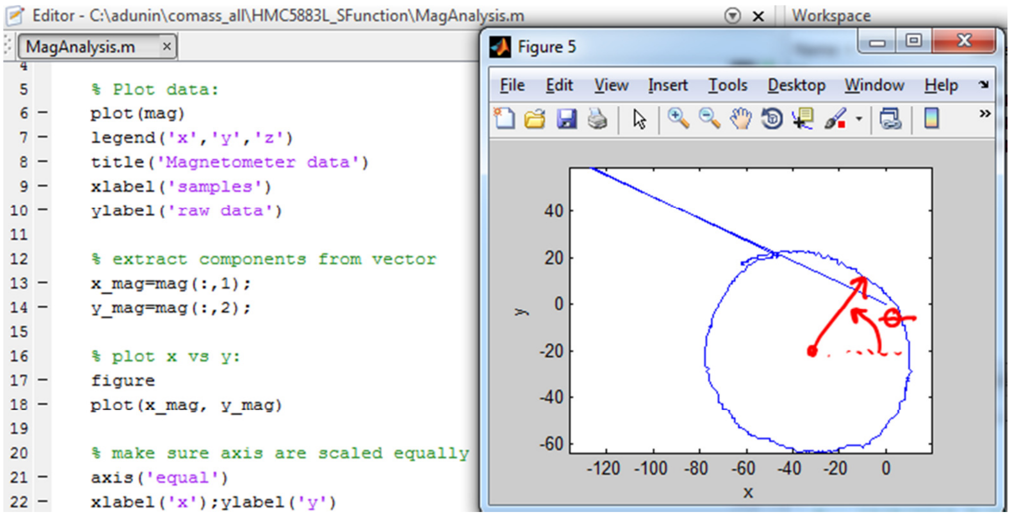
Analysis with Matlab:

The data from this experiment is now stored in a variable called ‘mag’ on the workspace. This section will developing a simple compass algorithm in Matlab from this data which will later be implemented in Simulink.

- First save the data by typing in the command line: `save Mag_Data mag`
    - This will save the variable ‘mag’ to a file called Mag_Data.bin
- Create an m-file to perform your analysis MagAnalysis.m. Run the following script and observe the results

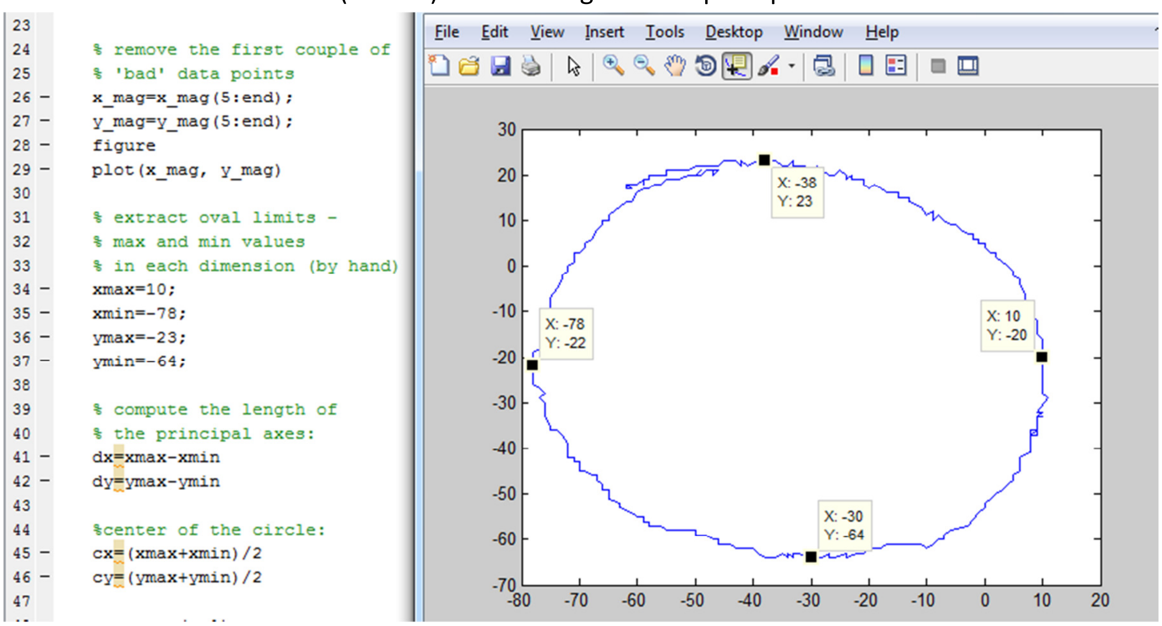


The y axis lags the x axis, and both of these change much more than the z axis. It would appear that the x and y data could be useful to determine heading. A more useful representation would be plotting x versus y instead of plotting them both with time:



Aside from what looks like a few 'bad' data points a circular shape results. In general it may be an oval instead of a nice circle. The heading could be found from the x and y components of the data if the center of the circle is known:

- Find the center of the circle (or oval) and the length of the principal axes:



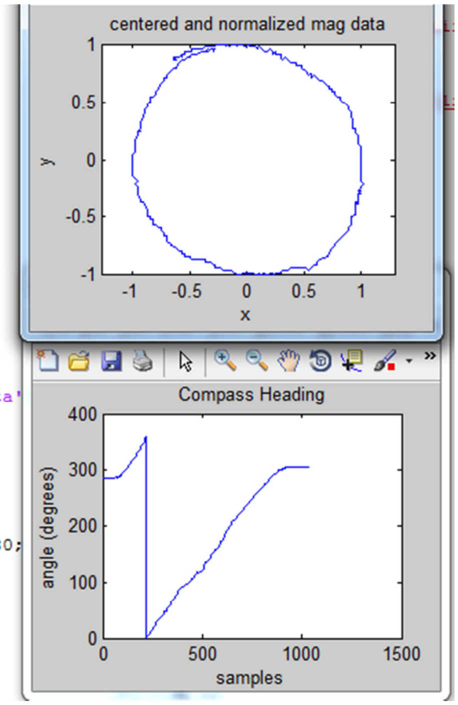
- Shift the data - so the circle center is at the origin
- Scale the data – so if the shape was oval it will be circular
- Use the arctangent to determine the angle

```
%center of the circle:
cx=(xmax+xmin)/2
cy=(ymax+ymin)/2

% Normalize and center the circle:
% The raw data is uint16 so
% before division convert to a double!
x_mag_n=double(x_mag-cx)/dx*2;
y_mag_n=double(y_mag-cy)/dy*2;

% plot normalized and shifted circle:
figure
plot(x_mag_n,y_mag_n)
axis('equal')
title('centered and normalized mag data'
xlabel('x')
ylabel('y')

% now extract the angle:
% convert to degrees
angle=180/pi*atan2(y_mag_n,x_mag_n)+180;
figure
plot(angle)
title('Compass Heading')
xlabel('samples')
ylabel('angle (degrees)')
```

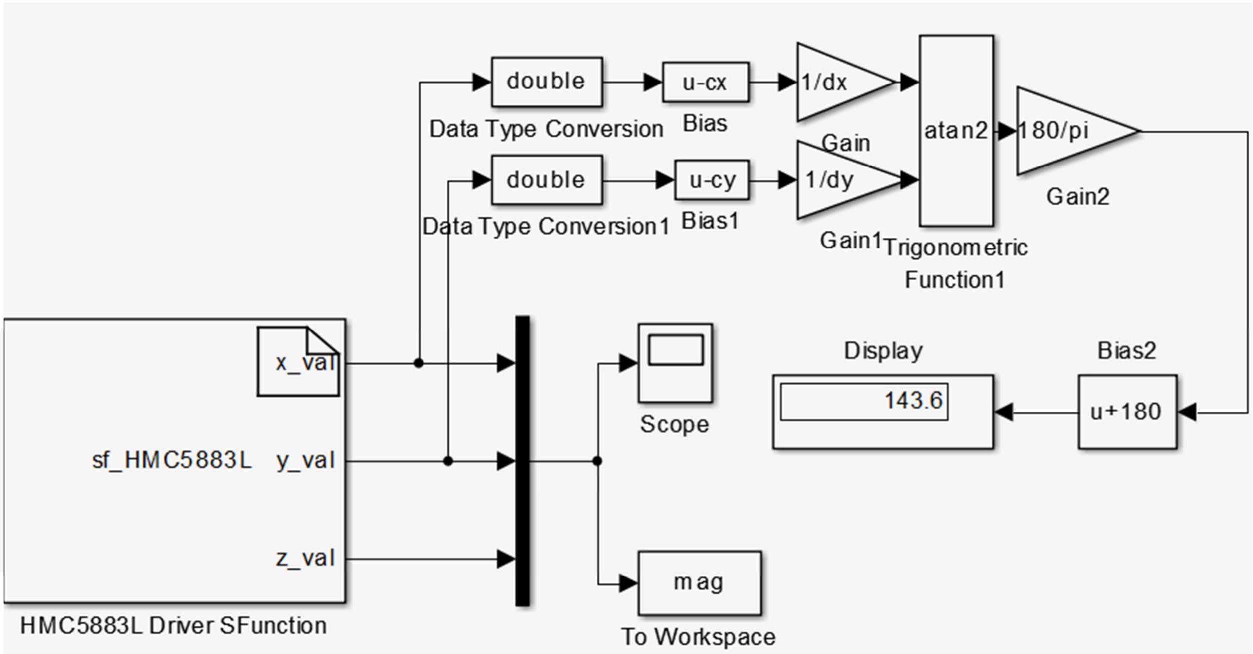


These are the general steps needed to calibrate the compass. In general this type of calibration routine would be automated to shift and scale the data appropriately. The heading changes from 0 to 360 degrees as expected!

Part 3: Implementation In Simulink

Objectives:

Now that the algorithm has been developed in Matlab it can be implemented in real time on the hardware:


double
u-cx
1/dx
atan2
180/pi
Data Type Conversion
Bias
Gain
double
u-cy
1/dy
Gain2
Data Type Conversion1
Bias1
Gain1
Trigonometric Function1
Display
Bias2
143.6
u+180
x_val
Scope
sf_HMC5883L
y_val
z_val
mag
HMC5883L Driver SFunction
To Workspace