Tutorial 2: Using Wireless 9DOF IMU for Real-Time Orientation/Rotation Tracking

Introduction

In this second tutorial, we will describe how to use the wireless 9DOF IMU sensor to monitor the orientation or rotation of a target object, in 3-Dimensional space and real time. This small, light, and cost-effective sensing unit (~$75) can be used in areas such as robotics, drones, autonomous vehicles, biomechanics, wearable technologies, gaming, VR/AR, motion monitoring, structural monitoring and many other situations. The wireless feature offers further advantages by eliminating the cables. Starting from Tutorial 3, we will demonstrate a variety of applications using this sensor. Here in this one, we will focus on the sensing technology itself. The step-by-step instruction will be covered in the following order:

- Set up the hardware
- Using the Firmware
- Use MovSens Orientation/Rotation Tracker
- Calibrate the sensor

Set Up the Hardware

What you need

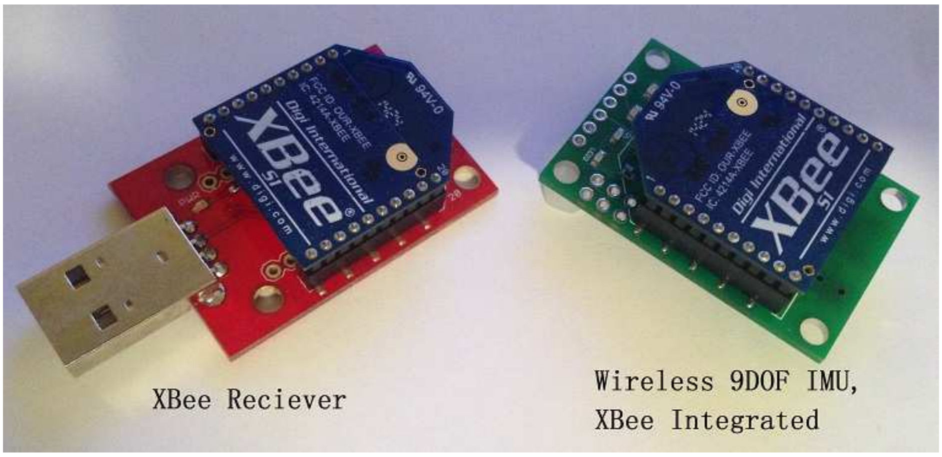
In order to use the wireless sensor, you need the following parts. If you do not have them already, a link to the place where you can get them is provided.

- MovSens Wireless 9DOF IMU (Get it here)
- A pair of XBee Serial 1 modules (Get it here)
- XBee Explorer USB Dongle (Get it here)
- A liPo Battery 1000mAh (Get it here)

It is also highly recommended that you have an FTDI basic breakout (Get it here) and a miniUSB cable, for downloading Arduino program to IMU board through UART.

Put them together

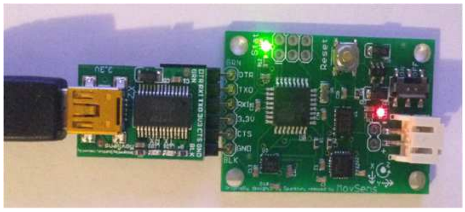
As introduced in Tutorial 1, the IMU board is pre-loaded with an orientation/rotation tracking firmware, and you can directly use it. All you need to do is to setup the 2 XBee modules in the same network, and insert them into IMU board and XBee explorer dongle respectively. You can then communicate with the IMU board, receiving data it measures and sending commends to it wirelessly through XBee transceivers.



In future, you probably want to write your own firmware, or revise the firmware for your specific applications, so here we want to introduce how to put hardware pieces together to program the IMU board using Arduino software.

1. Connect the FTDI basic breakout to the IMU board via the 6-pin connector. Match the ports correctly, DO NOT reverse the direction;

```
FTDI BREAKOUT <--> IMU board
GND           <--> GND
CTS           <--> CTS
3.3V          <--> 3.3V
TXO           <--> RXI
RXI           <--> TXO
DTR           <--> DTR
```



2. Connect the IMU board with your PC using a mini USB cable;

At this moment, the hardware is ready for programming. The following section describes the software setup and firmware downloading process.

Setting up the Software

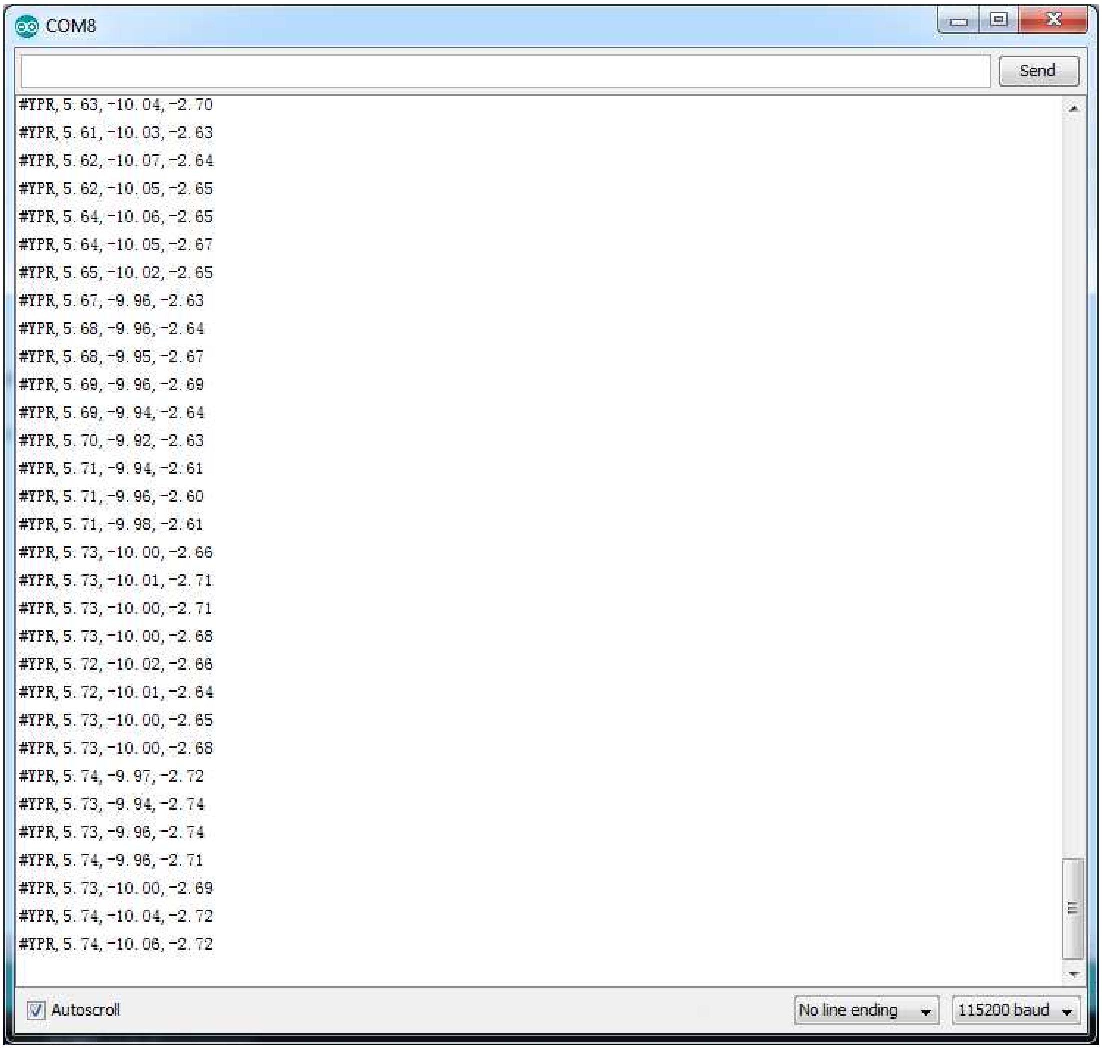
- Download the latest *MovSens_Orientation Firmware* from GitHub
- Download and install the latest Arduino Software. We will use it to upload the firmware. Any Arduino versions 1.x should work fine, tested with versions upto 1.6.7

Uploading the firmware If the XBee modules are set up correctly, you can upload the firmware wirelessly using the XBee modules. You can also use the FTDI adapter to upload the firmware. My experience is that using FTDI to upload the firmware is more convenient.

- From downloaded MovSens_Orientation Firmware package, open the file */MovSens_Orientation.ino* using Arduino
- In Arduino:
  - Go to "Tools" → "Board" and select "Arduino Pro or Pro Mini";
  - Go to "Tools" → "Processor" and select "ATmega328 (3.3v, 8mhz) ";
  - Go to "Tools" → "Serial Port" and select the port used with the *MovSens 9DOF IMU*.
  - Go to "Sketch" → "Upload". After a short while, at the bottom of the Arduino code window it should say "Done uploading".

Testing the firmware

Without changing the hardware connection, bring up the Serial Monitor of Arduino under "Tools" → "Serial Monitor". Set it to 115200 baud and you should get output like this:



COM8

Send

```
#YPR, 5.63, -10.04, -2.70
#YPR, 5.61, -10.03, -2.63
#YPR, 5.62, -10.07, -2.64
#YPR, 5.62, -10.05, -2.65
#YPR, 5.64, -10.06, -2.65
#YPR, 5.64, -10.05, -2.67
#YPR, 5.65, -10.02, -2.65
#YPR, 5.67, -9.96, -2.63
#YPR, 5.68, -9.96, -2.64
#YPR, 5.68, -9.95, -2.67
#YPR, 5.69, -9.96, -2.69
#YPR, 5.69, -9.94, -2.64
#YPR, 5.70, -9.92, -2.63
#YPR, 5.71, -9.94, -2.61
#YPR, 5.71, -9.96, -2.60
#YPR, 5.71, -9.98, -2.61
#YPR, 5.73, -10.00, -2.66
#YPR, 5.73, -10.01, -2.71
#YPR, 5.73, -10.00, -2.71
#YPR, 5.73, -10.00, -2.68
#YPR, 5.72, -10.02, -2.66
#YPR, 5.72, -10.01, -2.64
#YPR, 5.73, -10.00, -2.65
#YPR, 5.73, -10.00, -2.68
#YPR, 5.74, -9.97, -2.72
#YPR, 5.73, -9.94, -2.74
#YPR, 5.73, -9.96, -2.74
#YPR, 5.74, -9.96, -2.71
#YPR, 5.73, -10.00, -2.69
#YPR, 5.74, -10.04, -2.72
#YPR, 5.74, -10.06, -2.72
```

☑ Autoscroll

No line ending

115200 baud

You can rotate the IMU board now and see the numbers streaming out on the serial monitor change. This suggests the firmware is working fine.

Testing the Wireless Orientation/Rotation Tracking

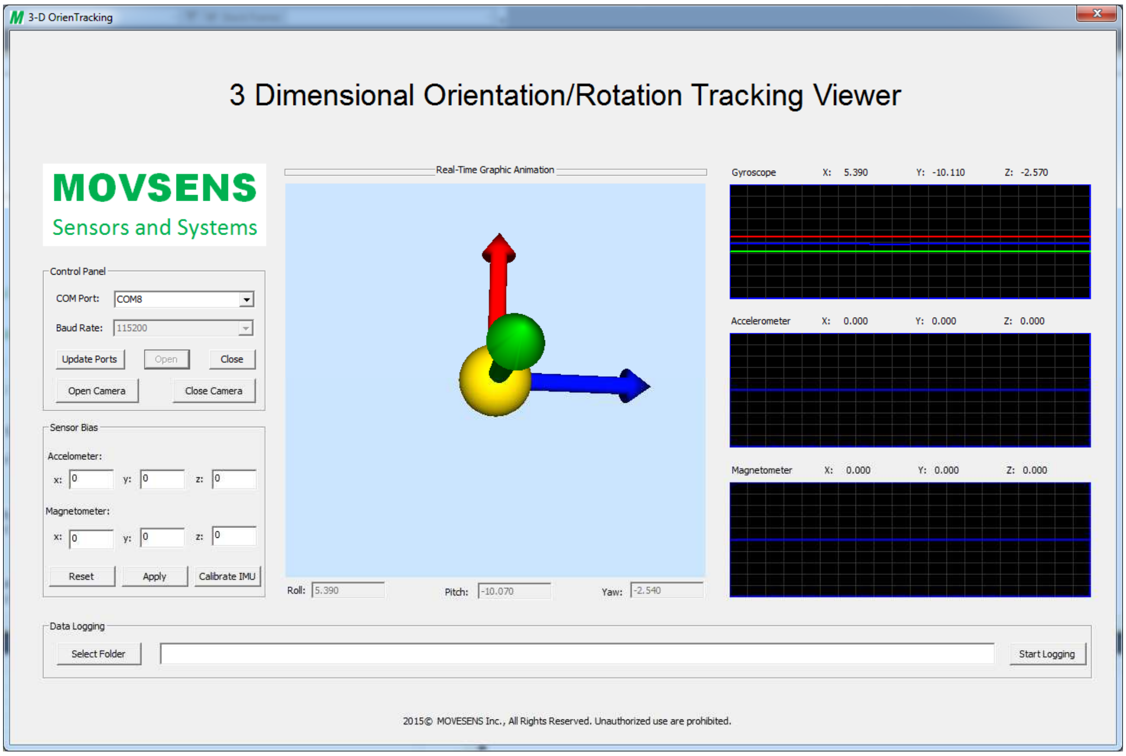
Now your IMU board is loaded with the latest orientation/rotation tracking firmware, and you can set up the XBee modules and test the wireless tracking.

- Set up the XBee modules like in Tutorial 1, and make sure the baud rates for both XBee modules are set to 115200.
- In Arduino software, select the port corresponding to your XBee Explorer and bring up the Serial Monitor of Arduino under "Tools" → "Serial Monitor".
- Turn on your IMU board. After the initialization process (~2 seconds), the wireless communication starts, and you can see the data steaming on the Serial Monitor.
- Without the constraint of a cable, you can move the IMU board freely and remotely from the PC, and you can see the 3D Yaw-Pitch-Roll angles are continuously monitored.

Use MovSens Orientation/Rotation Tracker

The wireless 9 Degrees-of-Freedom IMU, together with the "3D Orientation/Rotation Tracking" software, forms a complete inertial tracking system. This system could be directly used for your orientation tracking and data logging projects, or it could serve as your development kit, where you can write your own firmware and visualize the tracking with MovSens software interface.

- Download the latest version of "3D Orientation/Rotation Tracking" (current version 1.0) from www.movsens.com/downloads, and install it onto your computer;
- Double click to open the software, and the following graphic interface pops up;
- On the left side, select the right COM port and baud rate. Then click "Open" button;
- Rotate your IMU board, and you can see the rotation is continuously tracked by the graphic 3-axis. The numeric values of yaw, pitch and roll, are shown in the text box in real time;
- If you want to save the data your IMU transmits, you can use the file directory in the bottom to create a file and the data will be continuously saved in this file.



3 Dimensional Orientation/Rotation Tracking Viewer

COM Port: COM8

Baud Rate: 115200

Sensor Bias

Accelometer: x: 0 y: 0 z: 0

Magnetometer: x: 0 y: 0 z: 0

Roll: 5.390 Pitch: -10.070 Yaw: -2.540

Gyroscope X: 5.390 Y: -10.110 Z: -2.570

Accelerometer X: 0.000 Y: 0.000 Z: 0.000

Magnetometer X: 0.000 Y: 0.000 Z: 0.000

2015© MOVESENS Inc., All Rights Reserved. Unauthorized use are prohibited.

Now you have all the infrastructure to develop your own IMU tracking projects. The next section of this tutorial will describe how to calibrate the sensors and make the orientation tracking more accurate.

Calibrate the Sensors [1]

Accuracy of *Orientation/Rotation Tracking* can be improved a lot by calibrating the sensors. If not calibrated you may get effects like

- drifts in yaw when you apply roll to the board.

Standard calibration

It might be good to power up the MovSens 9DOF IMU a few minutes before calibration, so the sensors can warm up. Calibrating the sensor the first time can be a little tricky, but let's go:

- Open MovSens_Orientation.ino using Arduino and find the section "USER SETUP AREA" / "SENSOR CALIBRATION". This is where you put the calibration values later.
- Connect the MovSens 9DOF IMU to your computer, set the correct serial port in Arduino and open the Serial Monitor.
- If you didn't change the firmware defaults, you should see lots of output like this: *#YPR, -155.73,-76.48,-129.51*
- Set the firmware output mode to calibration by sending the string #oc. You should now see output like this: *accel x,y,z (min/max) = -5.00/-1.00 25.00/29.00 225.00/232.00*

<u>Calibrating the accelerometer:</u>

- We'll try to find the minimum and maximum output values for the earth gravitation on each axis. When you move the board, **move it real slowly**, so the acceleration you apply to it is as small as possible. We only want pure gravity!
- Take the board and point straight down with the x-axis (remember: x-axis = towards the short edge with the connector holes). While you do that, you can see the x-maximum (the second value) getting bigger.
- Hold the board very still and reset the measurement by sending #oc again.
- Now carefully tilt the board a little in every direction until the value does not get bigger any more and write down the x-maximum value.
- Do the same thing for the opposite side (x-axis pointing up) to get the x-minimum: bring into position, send #oc to reset measurement, find x-minimum value and write it down.
- Do the same thing for the z-axis (down and up) and the y-axis (right and left).
  - If you think you messed up the measurement by shaking or moving the board too fast, you can always reset by sending #oc.
- You should have all min/max values. Put them into MovSens_Orientation.ino.
- NOTE: You have to be really careful when doing this! Even slightly tapping the board with the finger messes up the measurement (try it!) and leads to wrong calibration.

<u>Calibrating the gyroscope:</u>

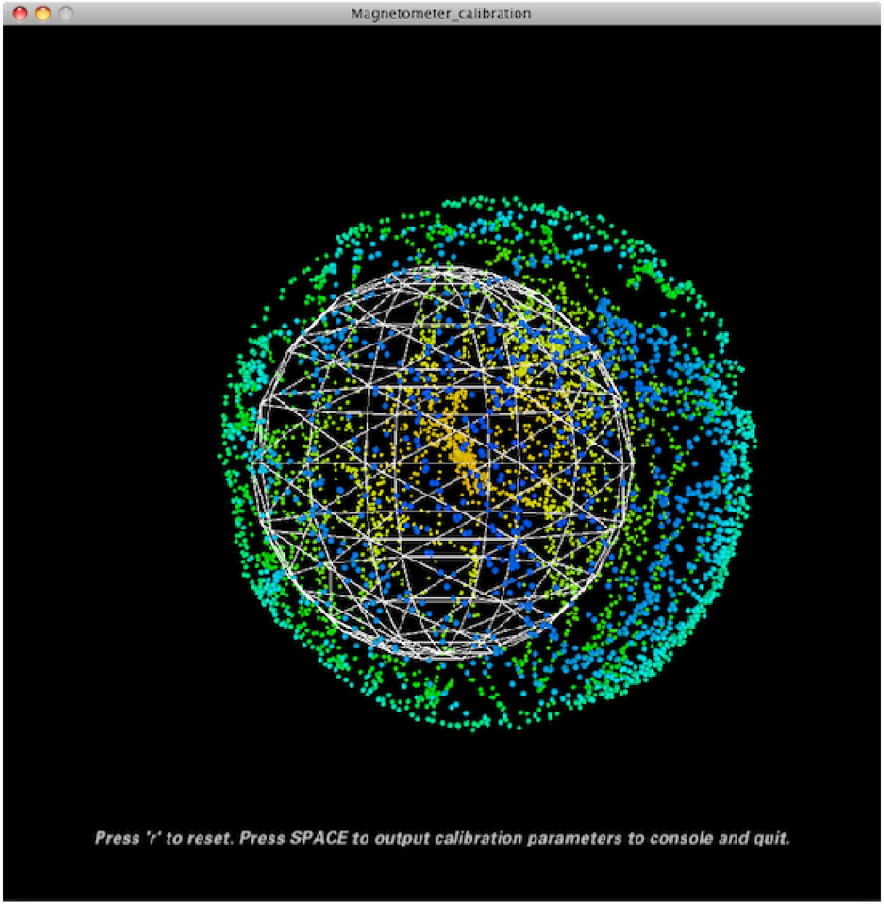
- Lay the IMU still on the table.
- We're still in calibration mode for the magnetometer. Send #on, which will move calibration to the next sensor, which is the gyroscope.
- Wait for 10 seconds, and do not move the IMU. It will collect and average the noise of the gyroscope on all three axes.
- You should now have output that looks like this:
  gyro x,y,z (current/average) = -29.00/-27.98 102.00/100.51 -5.00/-5.85
  - If you think you messed up the measurement by shaking or moving the board, you can reset by sending #oc.

- Take the second values of each pair and put them into MovSens_Orientation.ino.

Extended magnetometer calibration

The standard magnetometer calibration only compensates for *hard iron* errors, whereas the extended calibration compensates for *hard* and *soft iron* errors. Still, in both cases the source of distortion has to be fixed in the sensor coordinate system, i.e. moving and rotating with the sensor.

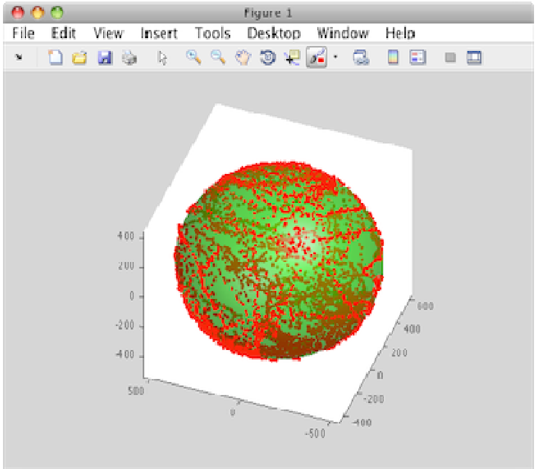
- To start calibrating, **put the sensor in the magnetic environment where it will be used later -** e.g. in the exact spot on your headphones (headphones have strong magnets, the less you move the sensor after calibrating, the better your results will be). If you intend to use the IMU with XBee, best also use that setup when calibrating now.
- Download and install Processing. We will use it to compile and run the test program. Any *Processing* versions 2.x should work fine, tested with version 3.2.3.
- Quit all applications that read from the sensor (e.g. *Serial Monitor*, *Processing* test sketch, …) and run the *Processing* magnetometer calibration sketch located in Processing/Magnetometer_calibration. In fact, you have to install the EJML library first, otherwise the sketch won't run. How to do that? Have a look at the *NOTE* at the top of Magnetometer_calibration.pde. Below that note you'll also find an explanation of how to set the right serial port so that the sketch can communicate with the IMU. If you're using the IMU with XBee and have paired and connected your computer with it, you'll also find the serial XBee port here.
- Try to rotate the sensor in a way so that you cover all orientations so you produce dots that more or less evenly cover the sphere.
- In a mostly undistorted environment this could look something like this:

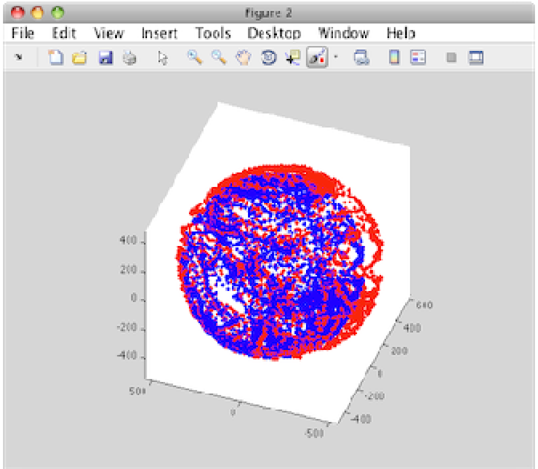


Press 'r' to reset. Press SPACE to output calibration parameters to console and quit.

- Hit SPACE and watch the *Processing* console - **you'll find some lines of code that you have to put into the firmware under "`USER SETUP AREA`" / "`SENSOR CALIBRATION`"** and you're done.

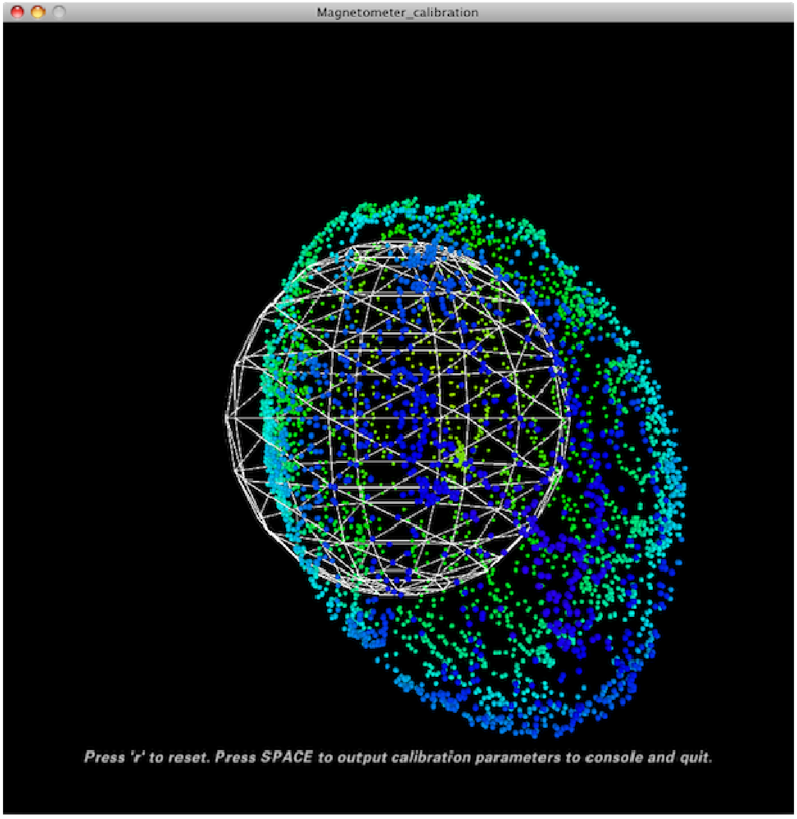
The collected data (the dots) are also written to a file `magnetom.float` in the sketch folder. Now in case you own Matlab, under `Matlab/magnetometer_calibration` you'll find a script called `magnetometer_calibration.m` that uses this file and produces some plots for you, so you can visually check the calibration. Ellipsoid fit and corrected values:





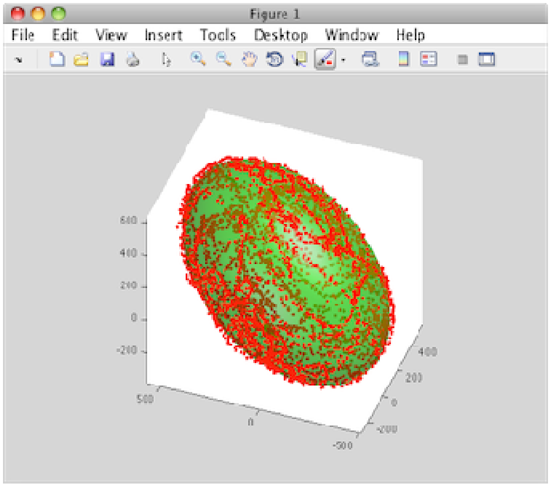
Another calibration example: Soft iron gives a sphere scaled and distorted into an ellipsoid.

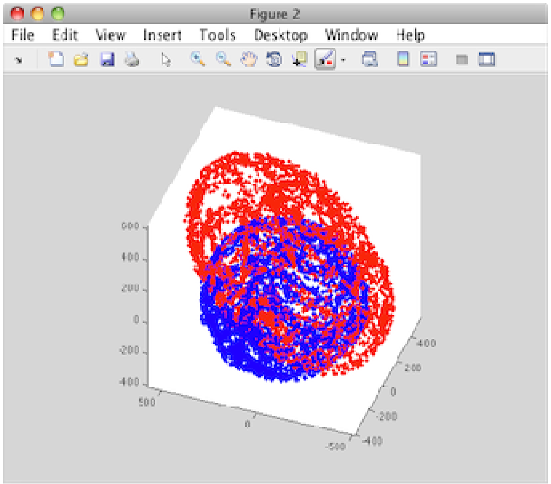
- Sampled raw magnetometer values:



Press 'r' to reset. Press SPACE to output calibration parameters to console and quit.

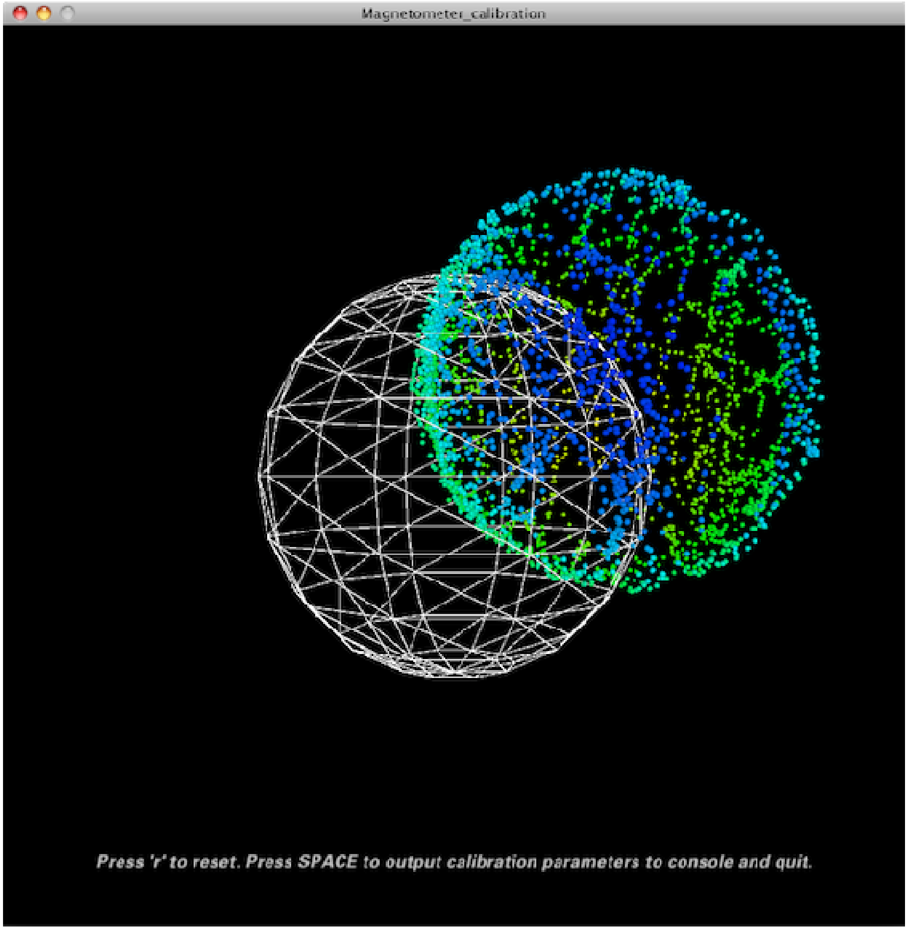
- Ellipsoid fit and corrected values:





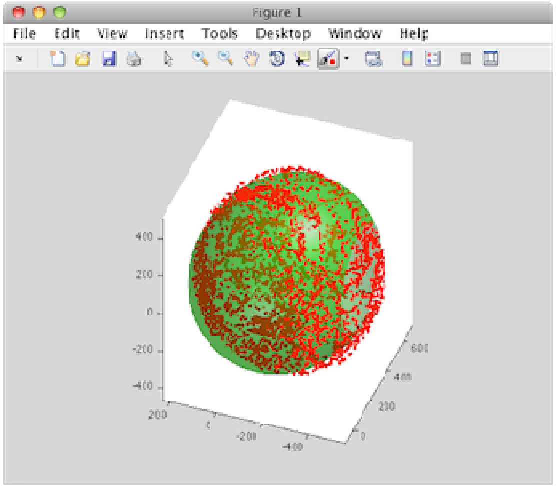
Another calibration example: Hard iron gives an offset sphere.

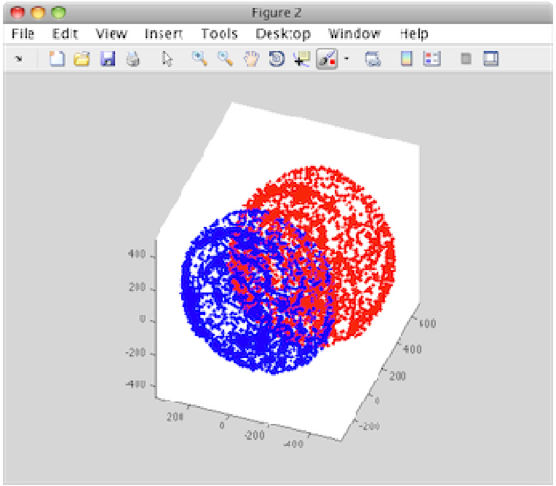
- Sampled raw magnetometer values:



Press 'r' to reset. Press SPACE to output calibration parameters to console and quit.

- Ellipsoid fit and corrected values:





Reference

[1] This section is adapted for MovSens Wireless 9DOF IMU from original document by Peter Bartz. Thanks, Peter!