TAMPERE UNIVERSITY
OF APPLIED SCIENCES

IMPLEMENTING MATLAB WITH ANDROID SENSORS

Mukul Nain

Bachelor’s thesis
May 2018
ICT Engineering

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
ICT Engineering
Exchange Student
MUKUL NAIN:
Bachelor's thesis 21 pages, appendices 9 pages
May 2018

This thesis works on giving information on various mobile sensors and in turn how to use them using MATLAB Mobile for various applications. MATLAB (matrix laboratory) is a multi-paradigm numerical computing environment. A proprietary programming language developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, C#, Java, Fortran and Python.

As of 2017, MATLAB has roughly 1 million users across industry and academia. MATLAB users come from various backgrounds of engineering, science, and economics.

Various mobile sensors such as acceleration sensor, magnetic field sensor, position sensor and gyroscope are also discussed. In the conclusion, we see the various applications which can be completed using acceleration sensor and also magnetic field sensor, i.e. step counter and a 3D compass respectively.

INDEX

1. Introduction	6
2. Sensor Basics	8
2.1 About Sensors	8
2.2 Hardware and Software sensors	9
2.3 Sensor types and availability	9
2.4 Handling different sensor configurations	11
3. Motion and Position Sensors	12
3.1 Coordinate Systems	12
3.2 Device coordinates	12
3.3 Earth’s coordinates	14
3.4 Motion sensors	15
3.4.1 Acceleration sensor	15
3.4.2 Gyroscope	16
3.4.3 Magnetic Field Sensor	17
3.5 Position Sensors	18
4. Conclusion	20
5. References	21
6. Appendices	22

ABBREVIATIONS

ENIAC	Electronic Numerical Integrator And Computer
GSM	Global System for Mobile
LTE	Long Term Evolution
ICT	Information and Communication Technologies
GSMA	Groupe Speciale Mobile Association
ITU	International Telecommunication Union
GPS	Global Positioning System
API	Application Program Interface
DoD	Department of Defence in the United States of America

STRUCTURE OF THESIS

This thesis discusses about basics of the android smartphone sensors in the beginning. Then briefly about the hardware and software sensors by listing the currently available sensors as well. Next the focus shifts on handling of different sensor configurations.

Then we discuss descriptively about the motion and position sensors by first discussing about coordinate system in general and then device coordinates and earth's coordinates. Lastly we discuss about the motion sensors such as acceleration sensor, magnetic field sensor and gyroscope sensor. Later we discuss about the position sensor.

Then finally concluding the thesis, in the appendices, application of magnetic field sensor can be seen with the screenshots. Also another application of acceleration sensor, the step counter can also be seen with the screenshots and codes for both the applications.

1. INTRODUCTION

As Moore’s law states that technology doubles its efficiency every two years, we have seen that in the past few years technology has developed at a really fast rate and is playing an important role in social well being of mankind. Human race has come a long way since the invention of the ENIAC by J. Presper Eckert and John Mauchly in 1946. Now we are all surrounded by computers, especially the one in our pockets.

The Mobile Phone has come a long way as well since its invention in 1973 by Martin Cooper of Motorola. Currently, affordable mobile phones and broad coverage of GSM and LTE are opening huge possibilities of providing ICT based services which will in turn bring about development, especially in developing nations.

The emergence of new Information and communication technologies (ICT), the Web and Internet in particular, in late 80s, has changed the World, offering a new paradigm in communication, exchange and commerce. However, while the new Information Society is still developing today, a new gap has also appeared with those without regular, effective access and ability to use these digital technologies. This is known as the Digital Divide, which is particularly affecting Developing Countries.

On another hand, ICTs are also a great opportunity for the Developing World. Providing minimal services (Health, Education, Business, Government...) to rural communities and underprivileged populations is of major importance to improve people lives, and to sustain development. Using ICTs would be the easiest and possibly only way to develop and deploy those services. It is therefore critical to work towards bridging this Digital Divide.

At the end of last year, according to the GSMA and ITU, the total number of people having accessing to a mobile phone was around 2.7 billions, and 80% of the World population was currently covered by a GSM network [1], [2]. These numbers illustrate the potential of the mobile platform to be the right solution to deploy services now, compared to other options, which are still in development phase.

2. SENSOR BASICS

Most Android-powered devices have built-in sensors that measure motion, orientation, and various environmental conditions. These sensors are capable of providing raw data with high precision and accuracy, and are useful if you want to monitor three-dimensional device movement or positioning, or you want to monitor changes in the ambient environment near a device.

For example, a game might track readings from a device's gravity sensor to infer complex user gestures and motions, such as tilt, shake, rotation, or swing. Likewise, a compass app might use the geomagnetic field sensor and accelerometer to report a compass bearing [2].

2.1 About Sensors

The Android platform supports three major categories of sensors:

- Motion sensors, to measure device motion. These sensors include accelerometers, gravity sensors, gyroscopes, and rotational vector sensors.
- Environmental sensors, to measure various environmental conditions, such as ambient air temperature and pressure, illumination, and humidity. These sensors include barometers, photometers (light sensors), and thermometers.
- Position sensors, to measure the physical position of a device. This category includes magnetometers (geomagnetic field sensors) and proximity sensors.

The device camera, fingerprint sensor, microphone, and GPS (location) sensor all have their own APIs and are not considered "sensors" for the purposes of the Android sensor framework.

2.2 Hardware and software sensors

Sensors can be hardware or software-based. Hardware-based sensors are physical components built into a handset or tablet device. Hardware sensors derive their data by directly measuring specific environmental properties, such as acceleration, geomagnetic field strength, or angular change[2].

Software-based sensors are not physical devices, although they mimic hardware-based sensors. Software-based sensors derive their data from one or more of the hardware-based sensors and are sometimes called virtual sensors or composite sensors. The proximity sensor and step counter sensors are examples of software-based sensors.

2.3 Sensor Types and Availability

Few Android-powered devices have every type of sensor. For example, most handset devices and tablets have an accelerometer, GPS and a magnetometer, but many fewer devices have barometers or gyroscope. Also, a device can have more than one sensor of a given type. For example, a device can have two gravity sensors, each one having a different range.

Sensor availability can also vary between Android versions, because the Android sensors have been introduced over the course of several platform releases.

The following are the sensors that the Android platform supports-

- Accelerometer- Used for motion detection (shake, tilt, and so on).
- Ambient Temperature- Monitoring air temperature.
- Gravity- Motion detection (shake, tilt, and so on).
- Gyroscope- Rotation detection (spin, turn, and so on).
- Light- Controlling screen brightness.
- Linear Acceleration- Monitoring acceleration along a single axis.
- Magnetic Field- Creating a compass.
- Orientation- Determining device position.
- Pressure- Monitoring air pressure changes.
- Proximity- Phone position during a call.
- Relative Humidity- Monitoring ambient humidity (relative and absolute), and dew point.
- Rotation Vector- Motion and rotation detection.
- Temperature- Monitoring temperatures [4].

2.4 Handling different sensor configurations

Android does not specify a standard sensor configuration for devices, which means device manufacturers can incorporate any sensor configuration that they want into their Android-powered devices. As a result, devices include a variety of sensors in a wide range of configurations. If your application relies on a specific type of sensor, you have to ensure that the sensor is present on a device so your app can run successfully.

You have two options for ensuring that a given sensor is present on a device:

- Detect sensors at runtime, then turn on or turn off app features as appropriate.
- Use Google Play filters to target devices with specific sensor configurations.

3. MOTION AND POSITION SENSORS

We can use motion and position sensors to monitor a device's movement or its position in space.Use the classes and methods from the Android sensor framework to gain access to the motion and position sensors and to handle changes to sensor data.All the motion sensors return multi-dimensional arrays of sensor values for each event. For example, during a single sensor event the accelerometer returns acceleration force data for the three coordinate axes (*x* , *y* , *z*) relative to the device.

3.1 Coordinate Systems

The motion and position sensors in Android typically use two different coordinate systems: a device coordinate system relative to the device, and a coordinate system relative to the surface of the Earth. Both systems use a standard 3-axis system (*x* , *y* , *z*). In addition, some sensors and methods in the Android sensor framework provide their data as angles around the three axes.

3.2 Device Coordinates

Most Android sensors, including the accelerometer and gyroscope, use a standard 3-axis coordinate system defined relative to the device's screen when the device is held in its default orientation (portrait orientation, for a phone). In the standard 3-axis coordinate system, the *x* -axis is horizontal and points to the right. The *y* -axis is vertical and points up, and the *z* -axis points toward the outside of the screen face. In this system, coordinates behind the screen have negative *z* values[3].

The most important point to understand about this coordinate system is that, unlike activity rotation, the axes are not swapped when the device's screen orientation changes—that is, the sensor's coordinate system never changes as the device rotates.

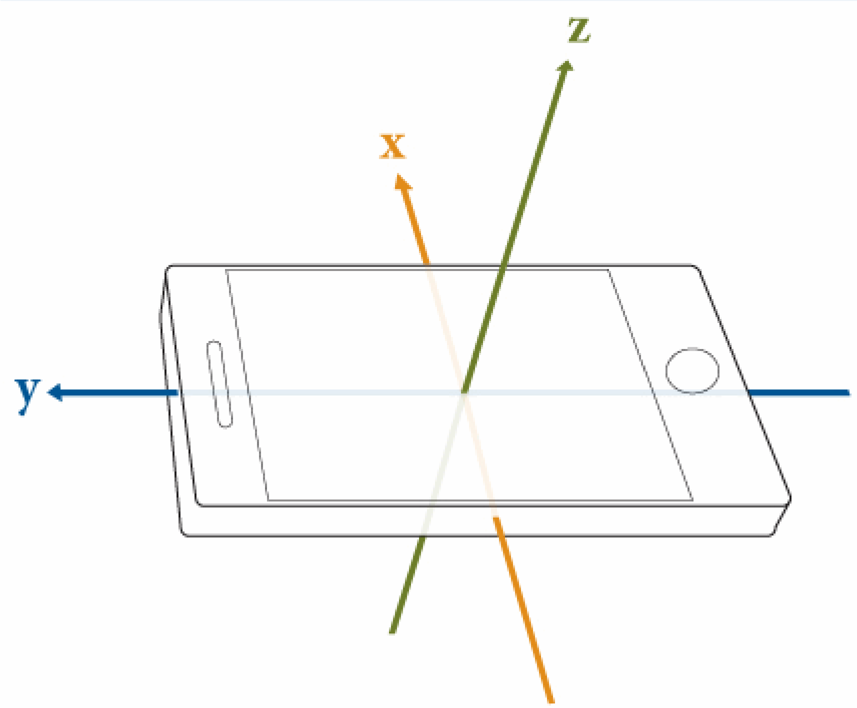

Figure 1- Coordinates of the given device; Source-

https://se.mathworks.com/help/supportpkg/mobilesensor/ug/phoneorv2.gif

If the program uses sensor data to position views or other elements on the screen, you need to transform the incoming sensor data to match the rotation of the device. The program must not assume that a device's natural (default) orientation is portrait. The natural orientation for many tablet devices is landscape. And the sensor coordinate system is always based on the natural orientation of a device.

3.3 Earth's Coordinates

Some sensors and methods use a coordinate system that represents device motion or position relative to the Earth. In this coordinate system:

- *y* points to magnetic north along the surface of the Earth.
- *x* is 90 degrees from *y* , pointing approximately east.
- *z* extends up into space. Negative *z* extends down into the ground.

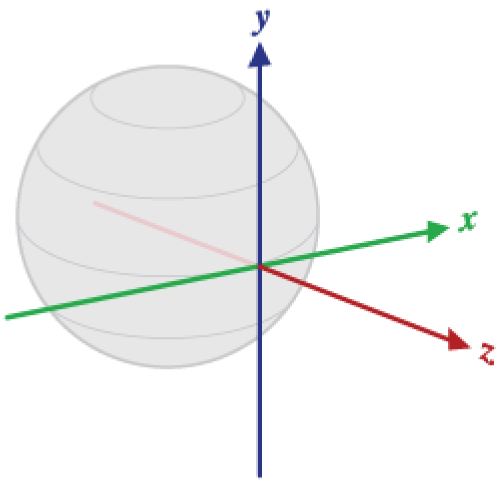

Figure 2- Earth's coordinates; Source- https://google-developer-training.gitbooks.io/android-developer-advanced-course-concepts/images/3-2-c-motion-and-position-sensors/world-coordinates.png

3.4 Motion sensors

The Android platform provides several sensors that let you monitor device motion such as tilt, shake, rotation, or swing. The movement is usually a reflection of direct user input, for example a user steering a car in a game, or a user controlling a ball in a game. Movement can also be a reflection of the device's physical environment, for example the device moving with you while you drive your car[2].

- When movement is a reflection of direct user input, you are monitoring motion relative to the device's frame of reference, or your app's frame of reference.
- When movement is a reflection of the device's physical environment, you are monitoring motion relative to the Earth.

Motion sensors by themselves are not typically used to monitor device position, but they can be used with other sensors, such as the geomagnetic field sensor, to determine a device's position relative to the Earth's frame of reference[25].

This section describes many of the most common Android motion sensors.

3.4.1 Accelerometer

The accelerometer measures the acceleration applied to the device along the three device axes (*x* , *y* , *z*), including the force of gravity. The inclusion of the force of gravity means that when the device lies flat on a table, the acceleration value along the *z* -axis is +9.81. This corresponds to the acceleration of the device (0 m/s^2) minus the force of gravity (-9.81 m/s^2). A device in freefall has a*z* value of 0, because the force of gravity is not in effect[4].

Nearly every Android-powered handset and tablet has an accelerometer in hardware. The accelerometer uses many times less power than other motion sensors. However, the raw accelerometer data is quite noisy and there is a need to implement filters to eliminate gravitational forces and reduce noise[4].

3.4.2 Gyroscope

The term “gyroscope”, conventionally referred to the mechanical class of gyroscopes, derives from the Ancient Greek language, being the Physics of the “precession motion”, a phenomenon also observed in ancient Greek society [6].

Gyroscopes are devices mounted on a frame and able to sense an angular velocity if the frame is rotating [5]. Many classes of gyroscopes exist, depending on the operating physical principle and the involved technology. Gyroscopes can be used alone or included in more complex systems, such as Gyrocompass [7], Inertial Measurement Unit [8], Inertial Navigation System [9] and Attitude Heading Reference System [10].

In this thesis, we will talk about Gyroscope as an Android sensor.The gyroscope sensor measures the rate of rotation around the three device axes (x ,y , z). All values are in radians/second. Although you can use the gyroscope to determine the orientation of the device, it is generally better practice to use the accelerometer with other sensors such as the magnetometer. Many current devices have a gyroscope sensor, although the gyroscope may be unavailable in older or lower-end devices [4].

3.4.3 Magnetic Field Sensor

The magnetic field sensor is similar to the rotation vector sensor, but it uses a magnetometer instead of a gyroscope. The accuracy of this sensor is lower than the normal rotation vector sensor, but the power consumption is reduced[11].

The system computes the orientation angles by using a device's geomagnetic field sensor in combination with the device's accelerometer. Using these two hardware sensors, the system provides data for the following three orientation angles:

- **Azimuth (degrees of rotation about the -z axis).** This is the angle between the device's current compass direction and magnetic north[15]. If the top edge of the device faces magnetic north, the azimuth is 0 degrees; if the top edge faces south, the azimuth is 180 degrees. Similarly, if the top edge faces east, the azimuth is 90 degrees, and if the top edge faces west, the azimuth is 270 degrees[17].
- **Pitch (degrees of rotation about the x axis).** This is the angle between a plane parallel to the device's screen and a plane parallel to the ground. If you hold the device parallel to the ground with the bottom edge closest to you and tilt the top edge of the device toward the ground, the pitch angle becomes positive. Tilting in the opposite direction— moving the top edge of the device away from the ground—causes the pitch angle to become negative[16]. The range of values is -180 degrees to 180 degrees[17].
- **Roll (degrees of rotation about the y axis).** This is the angle between a plane perpendicular to the device's screen and a plane perpendicular to the ground. If you hold the device parallel to the ground with the bottom edge closest to you and tilt the left edge of the device toward the ground, the roll angle becomes positive[14]. Tilting in the opposite direction—moving the right edge of the device toward the ground— causes the roll angle to become negative. The range of values is -90 degrees to 90 degrees[17].

3.5 Position Sensors

The system has only been available for public use for a couple decades ago. This was due to demand for the need of locating objects for various reasons and also the need of reducing cost which is the restricted limitation of applications and usage of the GPS. The host of the GPS is the Department of Defence in the United States of America. Due to public demands, the DoD has researched and innovated the GPS system to be ready for public use. One of the aims of the system was to develop a single unified application. A single unified application that allowed more civilian use and user friendly. This was very attractive to real time users such as businesses and the public[25]. Local government authorities can also benefit from it such as the police and fire departments for life and death situations. To an extent, it will also have value in terms of security of the property of a private owner. Initially, it was only affordable to large corporations, largely because of the need for insurance of a valuable item. It was a gradual process where the GPS was available for use by the general public.

Due to other technological advances such as more efficient communication systems, geographic databases and innovations and breakthroughs in the microchip industry and the availability of Internet access, the GPS became more affordable; hence, it is now widely used.

Although built for military purposes, due to social needs, the GPS has been innovated for the use of the public. Such uses can be seen in the field of transportation, geographic research and weather prediction. The information of position can be invaluable in these areas if given in relation to the intended path, showing points of interest and potential hazards: an aircraft position in relation to a destination; a car on a moving street map; or a boat in relation to islands and obstacles. Combined with communication technology such as the cellular phone, the knowledge of position can be life saving, reducing search and rescue mission.[25]

4. CONCLUSION

This thesis designs a step counter based on MATLAB and the Android smartphone sensor, acceleration sensor. It also designs a 3D compass using the magnetic field sensor in the given Android smartphone.

It also discusses in brief about the available smartphone sensors which can be interfaced with MATLAB using MATLAB Mobile such as gyroscope and GPS. Firstly the research is tending towards hardware and software sensors, also listing the sensors which are currently available in an android device. Then a brief discussion about different types of sensors and their availability. Then the research focuses on handling different sensors and their configurations in detail. In the next section, focus is shifted to motion and position sensors, which are used in designing of 3D compass and step counter. Firstly, there is a brief discussion about device coordinates and earth’s coordinates which are explained pictorially as well also with respect to each other. Then a detailed research on motion sensors including acceleration sensor, gyroscope, magnetic field sensor and finally the position sensor or GPS is presented.

The scope is very high as more and more sensors could be added keeping in mind the configurations of the given device.This is just a small step towards digitalization, which the present world is running towards.

REFERENCES

[1] Design of the Pedometer Based on MATLAB and Android Smartphone Sensor- Na Yang, Yanan Hu, December 2017, 978-94-6252-430-9
[2] Experimental Evaluation of Mobile Phone Sensors; Zhizhong Ma, Yuansong Qiao, Doi: 10.1049/Ic.2013.0047
[3] Android, "Sensor Coordinate System"; https://developer.android.com/guide/topics/sens ors/sensors_overview.html /
[4] Android, "SensorManager"; https://developer.android.com/reference/androi d/hardware/SensorManager.html/
[5] Gyroscope Technology and Applications: A Review in the Industrial Perspective; Antonello Cuccovillo, Lorenzo Vaiani, Edoardo Campanella, 7 October 2017.
[6] NASA. Available online: https://solarsystem.nasa.gov/news/2005/03/14/brief-history-of-gyroscopes
[7] Elliott-Laboratories. The Anschutz Gyro-Compass and Gyroscope Engineering; Wexford College Press: Kiel, Germany, 2003; pp. 7–24. ISBN 978-1-929148-12-7.
[8] Honeywell. Available online: https://aerospace.honeywell.com/en/products/navigation-and-sensors/ inertial-measurement-units
[9] King, A.D. Inertial Navigation—Forty Years of Evolution. GEC Rev. 1998, 13, 140–149
[10] Inertial Labs. Available online: https://inertiallabs.com/ahrs.html
[11] Measurement of the magnetic field of small magnets with a smartphone: a very economical laboratory practice for introductory physics courses by Enrique Arribas; 19 August 2015.
[12] Cole M 1990 Cognitive development and formal schooling: the evidence from cross-cultural research Vygotski and Education: Instructional Implications and Applications of Sociohistorical Psychology ed L C Moll (Cambridge: Cambridge University Press).
[13] Bodrova E and Leong D J 2004 Herramientas de la mente: El aprendizaje en la infancia desde la perspectiva de Vygotsky (Juarez: Pearson Education).
[14] Monsoriu J A, Giménez M H, Riera J and Vidaurre A 2005 Eur. J. Phys. 26 1149
[15] Shamin S, Zia W and Anwar M S 2010 Am. J. Phys. 78 433
[16] Chung H C, Liang J, Kushiyama S and Shinozuka M 2004 Int. J. Non-Linear Mech. 39 691
[17] Greczylo T and Debowska E 2002 Eur. J. Phys. 23 441
[18] Sirisathitkul C, Glawtanong P, Eadkong T and Sirisathitkul Y 2013 Rev. Bras. Ensino Fis. 35 1504
[19] Romulo O O and Franklin K N 1997 Am. J. Phys. 65 1115 [9] Ng T W and Ang K T 2005 Am. J. Phys. 73 793
[20] Tomarken S L et al 2012 Am. J. Phys. 80 351
[21] Vannoni M and Straulino S 2007 Eur. J. Phys. 28 781
[22] Kawam A and Kouh M 2011 Phys. Teach. 49 508
[23] Ochoa R, Rooney F G and Somers W J 2011 Phys. Teach. 49 16
[24] Breitbart J and Khanna G 2009 Numer. Anal. Appl. Math. 1168 1065
[25] Nicholas D.Lane et al., "A Survey of Mobile Phone Sensing". IEEE Communications Magazine, pp. 140-150, 2010

APPENDICES

Requirements

- Android device (4.0 and above)
- MATLAB (2015a and above)

Usage

- Interface the android device with MATLAB on your computer using MATLAB mobile.
- Install MATLAB Hardware support package for Android sensors
- Install MATLAB Mobile Application on the given Android device.
- Start acquiring data by enabling the sensors using the device or by coding.

CODES

1. 3D Compass using magnetic field sensor

```
                % Compass

        % Plots Magnetic Field Data as a vector


        connector on


        m = mobiledev;
        m.MagneticSensorEnabled=1;
        m.Logging=1;




        for k = 1:200
                                pause(1)
                                [mf, t] = magfieldlog(m);
                                mfa = mean(mf);
                                if ~isempty(mf)
                quiver3(0,0,0,mfa(1), mfa(2), mfa(3));

                % Set plot display
                xlim([-90 90]);
                ylim([-90 90]);
                zlim([-90 90]);

       discardlogs(m)
       end
end


        % Stop Acquiring Data & Disable Sensor
        m.Logging=0;
        m.MagneticSensorEnabled=0;


clear m
connector off
```

2. Step Counter

```
% Steps_acceleration
% Counts Number of Steps from Acceleration Data
connector on


m = mobiledev;
m.AccelerationSensorEnabled=1;
m.Logging=1;
```

```
% Walk around.
% Changes in Acceleration Sensors will indicate steps
disp('Walk Around')
pause(10)


% Stop Acquiring Data & Disable Sensor
m.Logging=0;
m.AccelerationSensorEnabled=0;


% Read Log & Process Data
[a, t] = accellog(m);


% Change vectors to scalars
% Removes dependency on Orientation
x = a(:,1);
y = a(:,2);
z = a(:,3);
mag = sqrt(sum(x.^2 + y.^2 + z.^2, 2));


% Plot magnitude
subplot(3,1,1);
stem(t, mag);
xlabel('Time (s)');
ylabel('Acceleration (m/s^2)');
title('Raw Magnitude')


% Remove effects of gravitity
magNoGrav = mag - mean(mag);


subplot(3,1,2);
stem(t, magNoGrav);
xlabel('Time (s)');
ylabel('Acceleration (m/s^2)');
title('No Gravity')


% Find Peaks
amag = abs(magNoGrav);
subplot(3,1,3);
stem(t, amag);
title('Absolute Magnitude')
xlabel('Time (s)');
ylabel('Acceleration Magnitude, No Gravity (m/s^2)');


THR = 2;
n = 1;
peaks = [];
peaksi = [];
minMag = std(amag);
for k = 2:length(amag)-1
    if (amag(k) > minMag) && ...
            (amag(k) > THR*amag(k-1)) && ...
            (amag(k) > THR*amag(k+1))
```

```
        peaks(n) = amag(k);
        peaksi(n) = t(k);
        n = n + 1;
    end
end


if isempty(peaks)
    disp('No Steps')
    return
end


nSteps = length(peaks);
disp('Number of Steps:')
disp(nSteps)


% Plot markers at peaks
hold on;
plot(peaksi, peaks, 'r', 'Marker', 'v', 'LineStyle', 'none');
hold off;



% Clean up
clear m
connector off
```

SCREENSHOTS

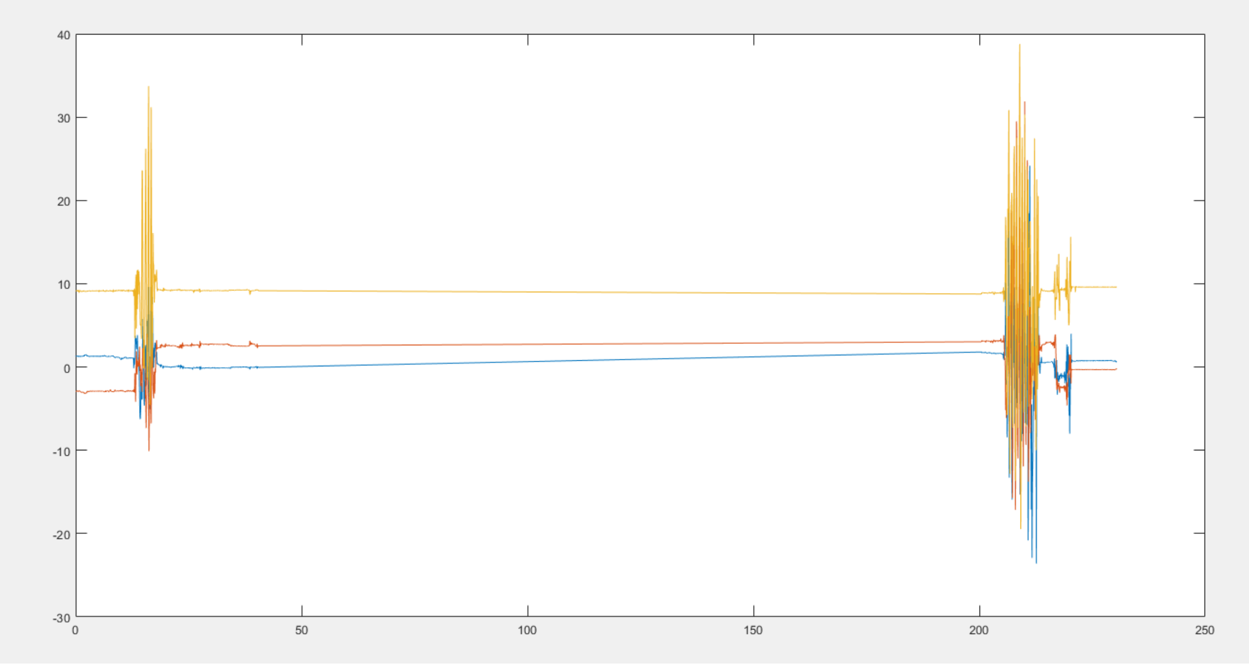

Figure 3- Acceleration sensor

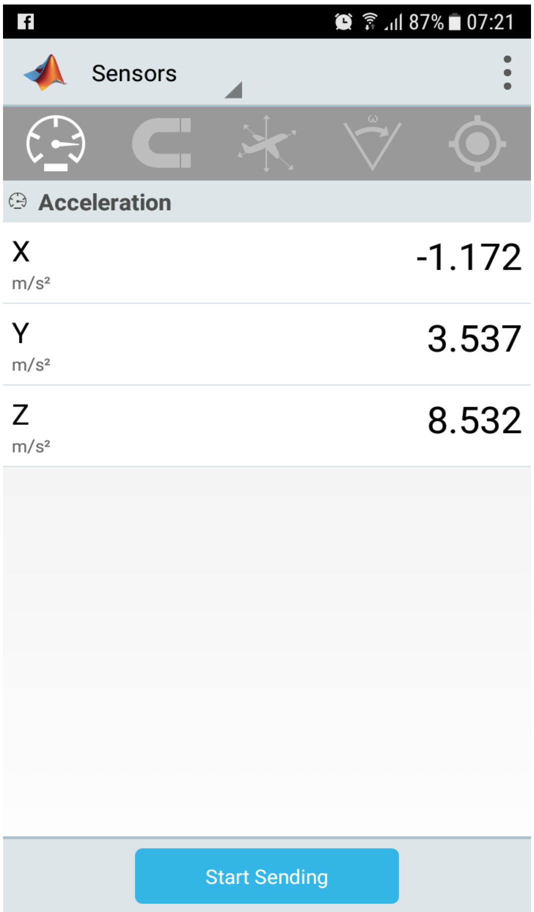

Figure 4- The mobile screenshot for Acceleration sensor.

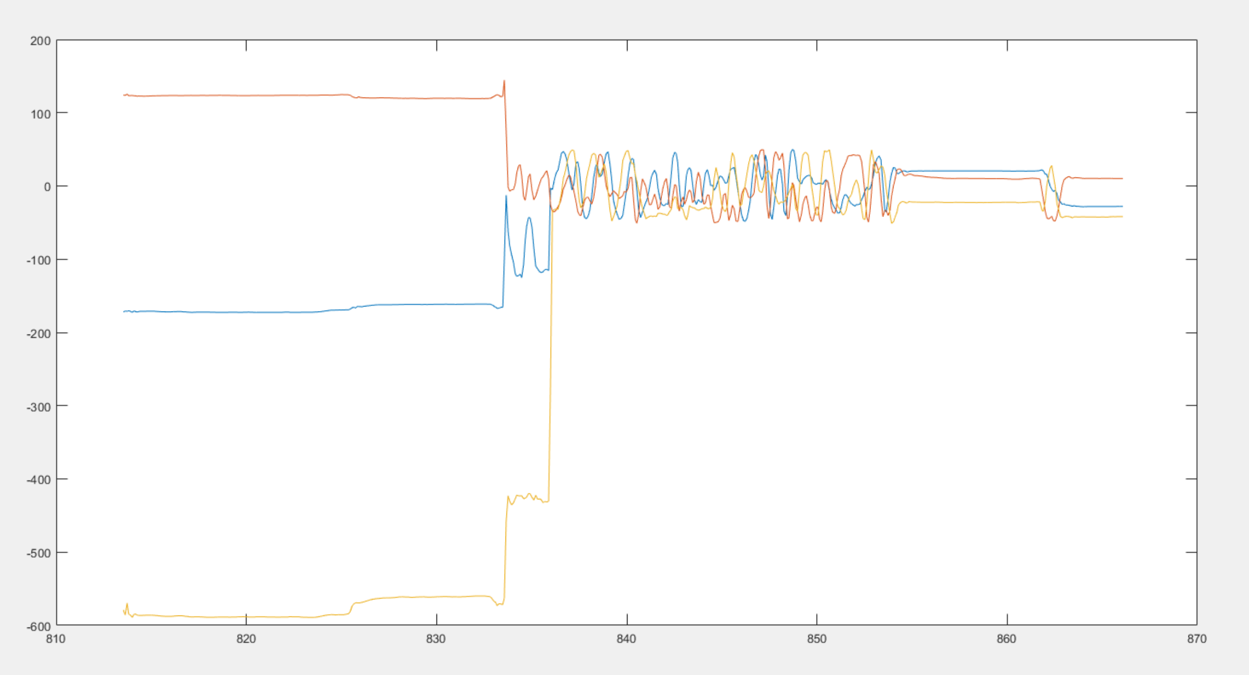

Figure 5- Magnetic Field Sensor

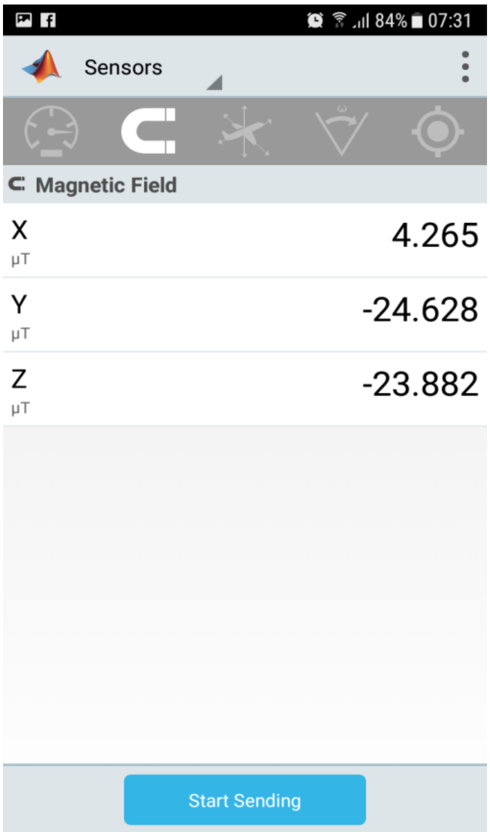

Figure 6- The mobile Screenshot for Magnetic field sensor.

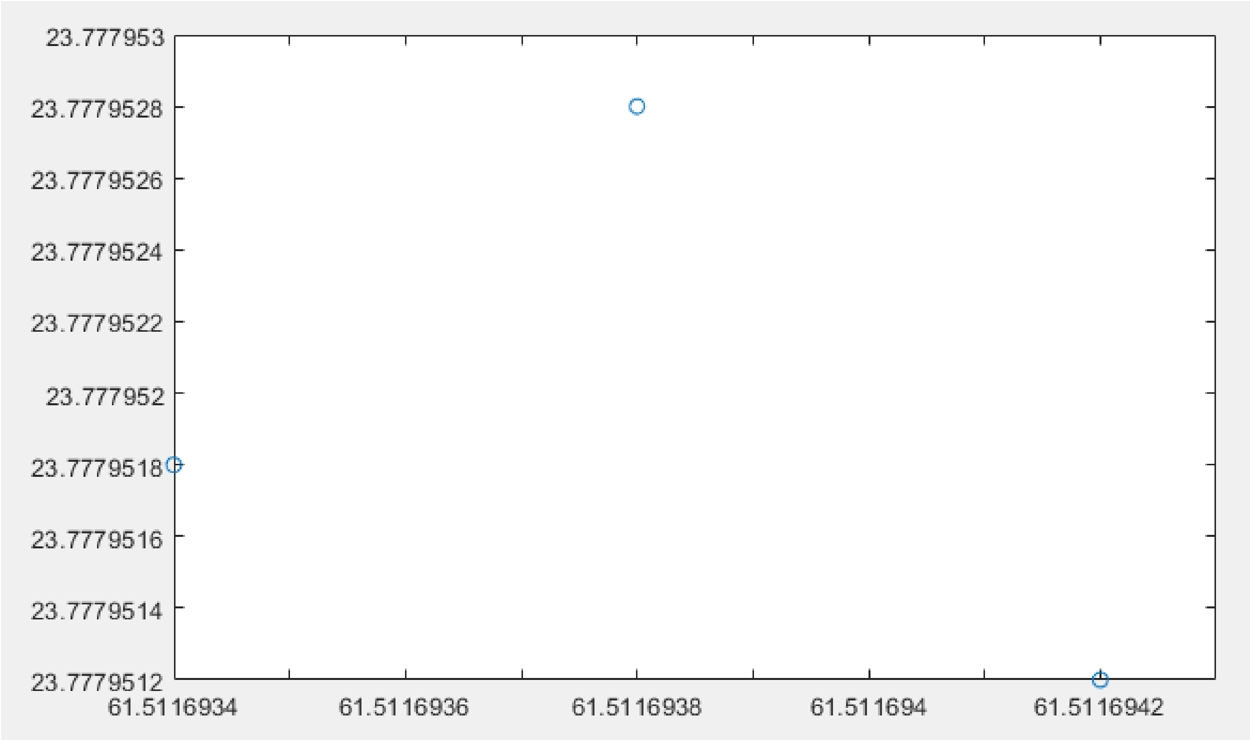

Figure 7- Position Sensor (GPS)

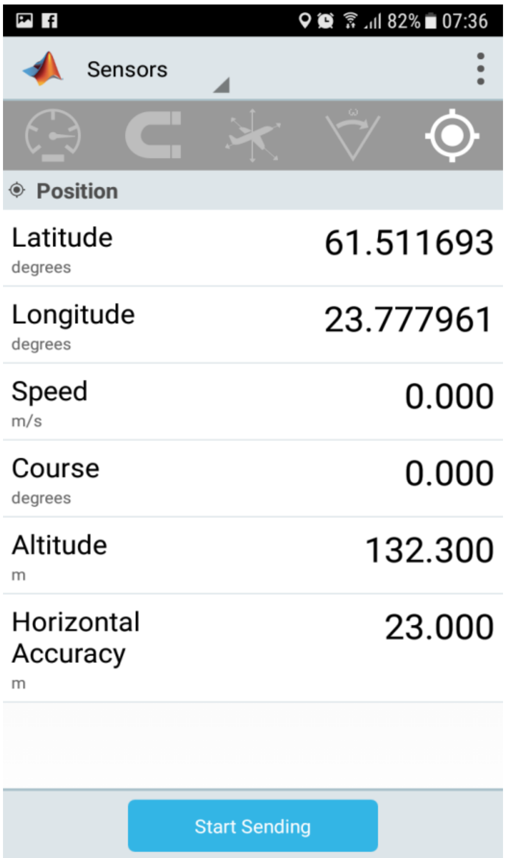

Figure 8- The mobile Screenshot for Position sensor.

SCREENSHOTS FOR 3D COMPASS AND STEP COUNTER

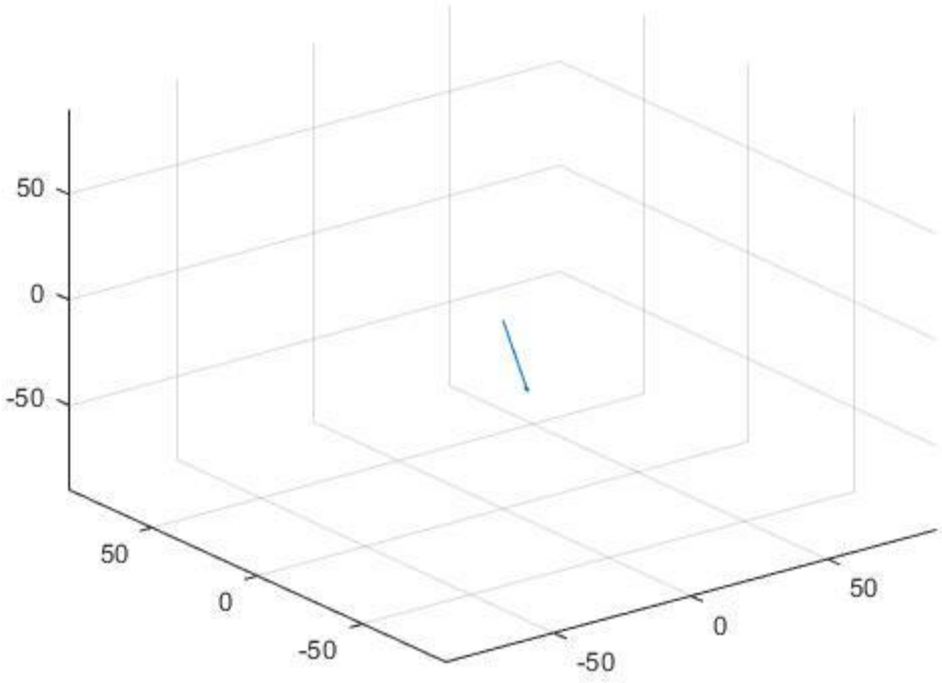

Figure 9- 3D Compass

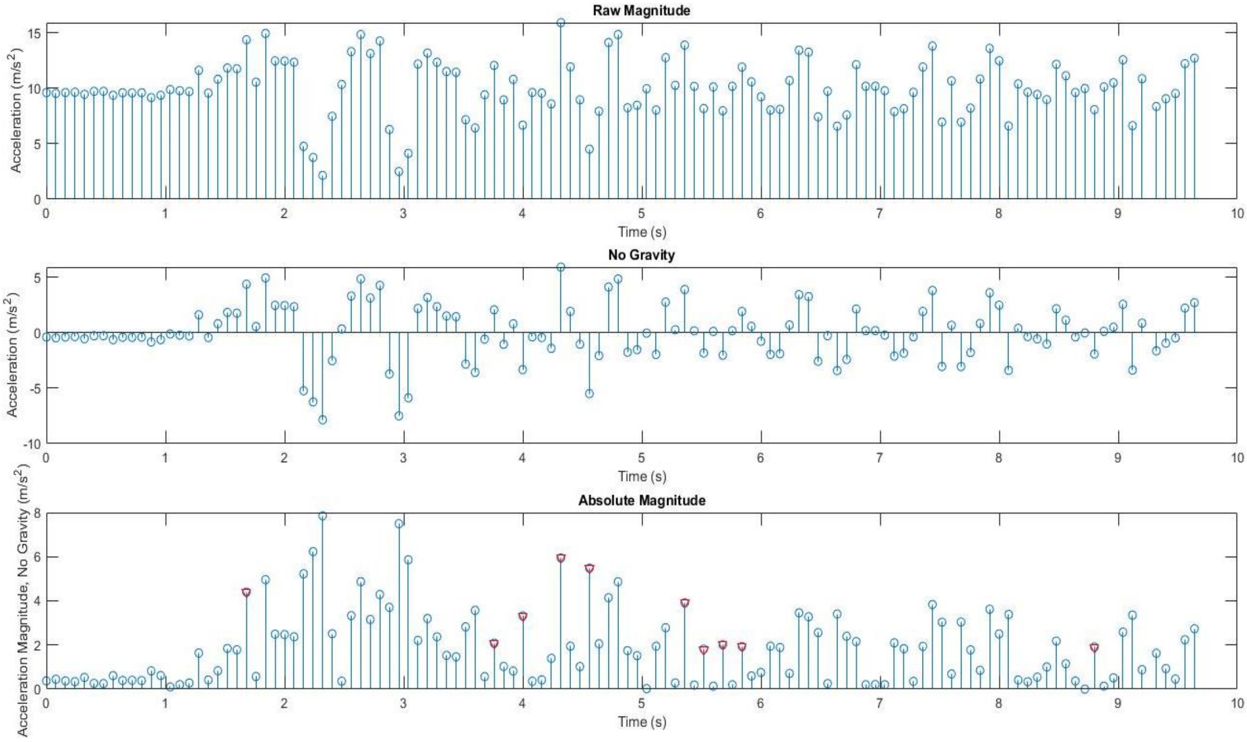

Figure 10- Step Counter