GAIT ANALYSIS

YASH JAIN
under the guidance of
Prof. Pranav Nerurkar
of Veermata Jijabai Technical Institute

A dissertation submitted to the Veermata Jijabai Technical Institute
for the degree of Doctor of Philosophy

Abstract

Gait analysis is the systematic study of animal locomotion, more specifically the study of human motion, using the eye and the brain of observers, augmented by instrumentation for measuring body movements, body mechanics, and the activity of the muscles. Modern gait analysis offers a broad variety of biomechanical parameters through which to quantify gait. This technique is widely used in sports industries to analyse the fitness of the athletes. It can also be used for analyse the motion of the person before and after any surgery to check the recovery of the patientâĂŹs movements.

Declaration

This dissertation is the result of my own work, except where explicit reference is made to the work of others, and has not been submitted for another qualification to this or any other university. This dissertation does not exceed the word limit for the respective Degree Committee.

Yash S Jain

Acknowledgements

Of the many people who deserve thanks, some are particularly prominent, such as my supervisor…

Preface

This thesis describes my research on various aspects of GAIT Analysis using MPU 8266 and Python.

Contents

Chapter 1.

MPU 6050

The MPU-6050 parts are the worldâĂŹs first MotionTracking devices designed for the low power, low cost, and high-performance requirements of smartphones, tablets and wearable sensors.

SThe MPU-6050 incorporates InvenSenseâĂŹs MotionFusion and run-time calibration firmware that enables manufacturers to eliminate the costly and complex selection, qualification, and system level integration of discrete devices in motion-enabled products, guaranteeing that sensor fusion algorithms and calibration procedures deliver optimal performance for consumers.

The MPU-6050 devices combine a 3-axis gyroscope and a 3-axis accelerometer on the same silicon die, together with an onboard Digital Motion Processor (DMP), which processes complex 6-axis MotionFusion algorithms. The device can access external magnetometers or other sensors through an auxiliary master I2C bus, allowing the devices to gather a full set of sensor data without intervention from the system processor. The devices are offered in a 4 mm x 4 mm x 0.9 mm QFN package.

Chapter 2.

IMPLEMENTATION (using MPU-6050)

2.1. Implementation

MPU-6050 was connected to the Wemos board as shown in the above pin diagram. SCL and SDA pins are used for sending data via I2C communication protocol.

Using Ardunio IDE we programmed Wemos Board to get accelerometer and gyroscope data.

Accelerometer data had gravity component in all three axes which is undesirable and so we filtered that out using a low pass filter by applying the following transformation on all three axes. g = 0.9 * g + 0.1 * v Where g is a global variable initialized to 0 and v is accelerometer reading of any axis.

7

Using v = v âĂŞ g you can remove the gravity factor in that axis. Time interval between two readings was calculated using the difference in the times in milliseconds returned by the millis() function in the Ardunio library.

We created a local server using XAMPP and using ESP8266 we uploaded the MPU data using PHP to MySQL database that was hosted on the local server.

Once the data was uploaded using the server then the data from the local server was fetched using python library MySQLdb. We then applied complementary filters on the raw data to remove noise.

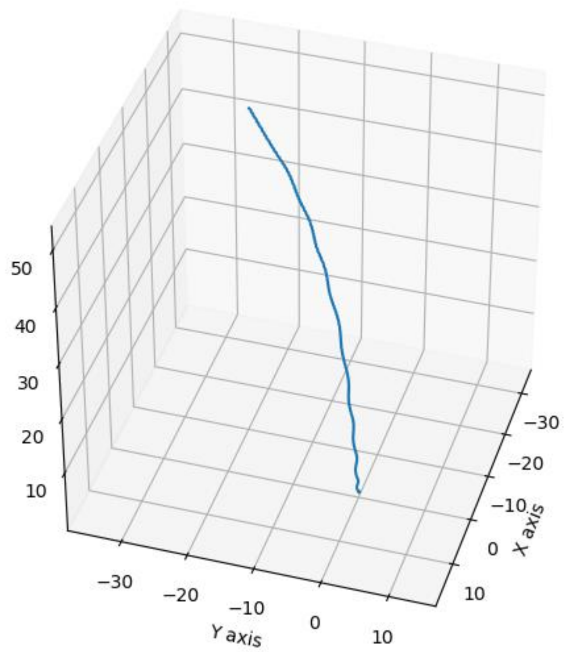
We initially tried to interpolate the accelerometer readings using the interpolate function in SciPy module. But the results of the plot and time taken to plot was not at all satisfactory so we then used the SimpsonâĂŹs Method of integration using the simps() function in the scipy.integrate module. Using this we integrated the acceleration data twice with respect to time to get displacement.

Finally after receiving the displacements in all 3 directions, we used matplotlib.pyplot to get the 3D graph of the motion.

2.2. Problems Faced

As mentioned above, the MPU-6050 is a cheap module and thus generates a lot of noise in the data. While, for a lot of applications this noise can be filtered effectively and the MPU-6050 can be used satisfactorily, but in our case even the slightest of noise got magnified over and over again as we integrated over large time intervals.

8 As a result we always got a increasing or decreasing displacement irrespective of the motion performed because the errors kept on adding up.



Appendix A.

Conclusion

Thus, from the above plots we can conclude that the MPU-6050 is very inferior compared to Mobile Phone Sensors and hence, MPU-6050 cannot be used for reliable position tracking.

Also, while the plots using Mobile Phone Sensors are quite good for the above datasets, we need to take into account the fact that these datasets are for motions performed in small time intervals only. For larger time intervals, even the Mobile Sensors was not able to give us satisfactory results.

Thus the Conclusion of this project is that pure IMU based position tracking is infeasible and for accurate results we always require optical sensors and cameras. But for smaller datasets and a cheaper alternative, IMU based position tracking can prove to be useful.

Bibliography

[1] https://matplotlib.org/1.4.2/users/pyplot$_t$*utorial.html*

[2] https://docs.scipy.org/doc/scipy/reference/tutorial/integrate.html

[3] https://docs.scipy.org/doc/scipy/reference/interpolate.html

[4] https://pykalman.github.io/

[5] http://mysql-python.sourceforge.net/MySQLdb.html

[6] http://www.starlino.com/imu$_g$*uide.html*