

# Lab 1: Interfacing with the WiiMote

Sam Mansfield and Toan Vounh  
EECS149

September 11th, 2013

## Introdcution

Something goes here.

## Analysis

### 2 Pair a desktop computer with the WiiMote

#### a What is the MAC address of your WiiMote?

Something goes here.

### 3 Plot the uncalibrated accelerometer signal from the Wiimote

#### a What are the values of the x and y axes when the WiiMote is at rest on level ground?

Something goes here.

#### b What is the value of the z axis when the WiiMote is at rest on level ground? What quantity is begin measured?

Something goes here.

### 4 Control the WiiMote actuators

#### a What is the Report ID and Payload to turn WiiMote LEDs 2 and 4 on, and LEDs 1 and 3 off?

Something goes here.

#### b How can you modify the above sequence to also enable the rumble motor?

Something goes here.

## 5 Measure sensitivity and bias of the accelerometer

### a Where are these measurements stored?

Something goes here.

### b What bias did you record?

Something goes here.

### c What sensitivity did you record?

Something goes here.

### d Provide screenshots of changes to the block diagram code

Something goes here.

## 6 Calibrate the accelerometer

### a What equations did you use to calibrate the accelerometer and to calculate the magnitude of the acceleration it measures?

To calibrate the accelerometer we used a simple affine function and the assumptions that when the WiiMote is at rest, with the buttons facing upwards, the calibrated x and y will be 0 and the calibrated z will be 1:

$$calibratedX = \frac{x - bias}{sensitivity}$$

$$calibratedY = \frac{y - bias}{sensitivity}$$

$$calibratedZ = \frac{z - bias}{sensitivity}$$

And at rest:

$$calibratedX = \frac{x - bias}{sensitivity} = 0$$

$$calibratedY = \frac{y - bias}{sensitivity} = 0$$

$$calibratedZ = \frac{z - bias}{sensitivity} = 1$$

So solving for bias we get:

$$bias = x$$

$$bias = y$$

And since x and y aren't the same value at rest we averaged x and y for the bias:

$$bias = \frac{x + y}{2}$$

To calculate sensitivity we used the following formula:

$$sensitivity = z - bias$$

Substituting in bias we get:

$$sensitivity = z - \frac{x + y}{2}$$

To solve for magnitude we used the distance formula:

$$magnitude = \sqrt{x^2 + y^2 + z^2}$$

**b Based on your calibration parameters and the number of bits of resolution of the WiiMote ADC, and an affine model of the WiiMote accelerometer, what is your estimate of the maximum acceleration that the WiiMote can measure?**

The maximum acceleration is when x, y, and z are at their max. It appeared that the max uncalibrated value of x, y, and z was 256, which would be 8 bits. This gives a max calibrated x, y, and z of:

$$calibratedXMAX = \frac{256 - bias}{sensitivity} = 0$$

$$calibratedYMAX = \frac{256 - bias}{sensitivity} = 0$$

$$calibratedZMAX = \frac{256 - bias}{sensitivity} = 1$$

Substituting in bias and sensitivity:

$$calibratedXMAX = \frac{256 - 124}{26} \approx 5$$

$$calibratedYMAX = \frac{256 - 124}{26} \approx 5$$

$$calibratedZMAX = \frac{256 - 124}{26} \approx 5$$

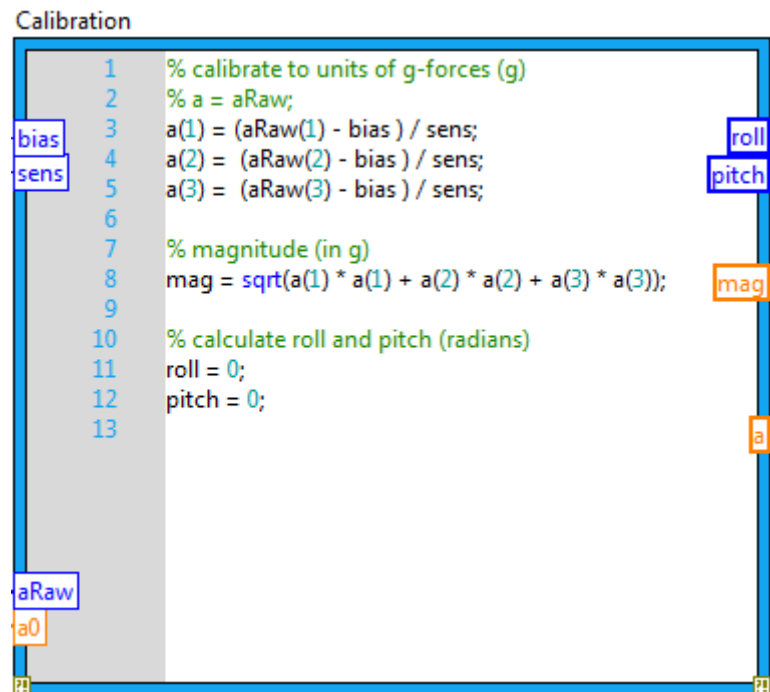
**c When the WiiMote is at rest, what should be the value of the magnitude indicator?**

At rest the only acceleration is gravity. Since we are measuring magnitude in g-forces, it should be 1g.

**d Why is it better to perform calibration at runtime, rather than hard-coding values for the sensitivity and bias of the sensor?**

The problem with hardcoding values for bias and sensitivity is that you are assuming that the WiiMote will always produce the same uncalibrated values for x, y, and z, which is not a stable assumption. The uncalibrated values can be dependent on temperature, which WiiMote you are using, and age of the WiiMote.

e Provide the relevant MathScript code and screenshots of changes (if any) to the block diagram



## 7 Measure Pitch and Roll

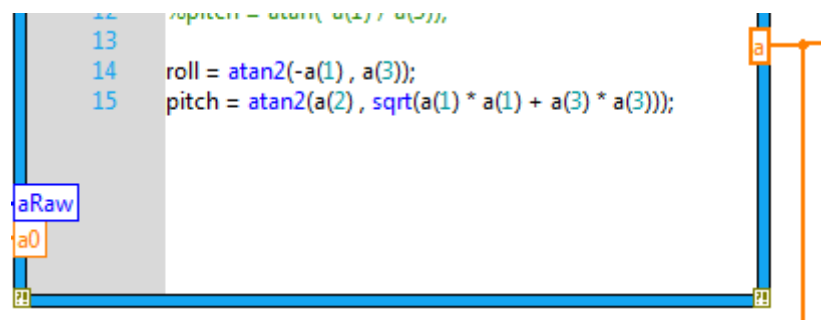
a What equations did you use to calculate pitch and roll?

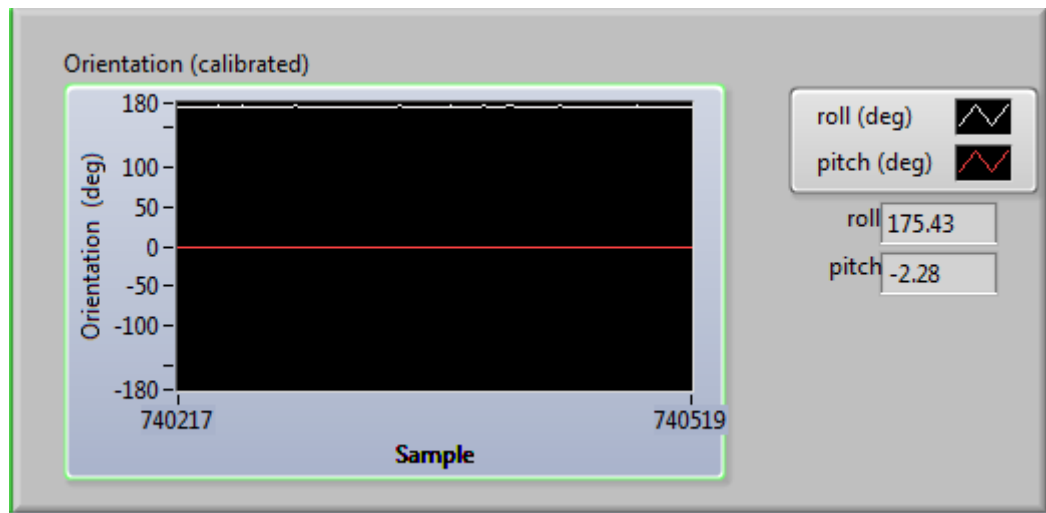
We used these two equations to calculate pitch and roll:

$$pitch = \arctan\left(\frac{y}{\sqrt{x^2 + z^3}}\right)$$

$$roll = \arctan\left(\frac{-x}{z}\right)$$

b Provide the relevant MathScript code and screenshots of changes (if any) to the block diagram.





## 8 Filter the accelerometer signal

a Describe your filter, and include an equation for its impulse response, frequency response, or its output as a fraction of input.

We used a averaging low pass filter of the form:

$$Filter(x) = (1 - q) * calibratedX(x) + q * calibratedX(x - 1)$$

$$Filter(y) = (1 - q) * calibratedY(y) + q * calibratedY(y - 1)$$

$$Filter(z) = (1 - q) * calibratedZ(z) + q * calibratedZ(z - 1)$$

We varied the q from 0.5 to 0.99 and decided that 0.9 provided the best response. The signal was less noisy and the delay was still low. This was a good compromise. The impulse response of our filter was:

$$h(n) = q^n * U(n)$$

And our frequency response was:

$$H(z) = \frac{q}{1 - (1 - q) * z^{-1}}$$

b Provide the relevant MathScript codd and screenshots of changes (if any) to the block diagram

```

1 % calibrate to units of g-forces (g)
2 % a = aRaw;
3 q = .9;
4 a(1) = (1-q)*((aRaw(1) - bias) / sens) + q*a0(1);
5 a(2) = (1-q)*((aRaw(2) - bias) / sens) + q*a0(2);
6 a(3) = (1-q)*((aRaw(3) - bias) / sens) + q*a0(3);
7

```

## 9 Share Your Feedback

This was an overall straightforward and helpful lab.

## Conclusion

This lab helped us to go through the process of receiving unfiltered data from a standard embedded system (WiiMote), and using that data to produce meaningful data, such as a calibrated accelerometer with magnitude. It was helpful to go through the process of figuring out how to calculate the bias and sensitivity without hardcoding the values, which is an extremely good way to think about calibration. It would have been nice to do a little more interfacing with the WiiMote, such as with the turning on and off different features and looking how the data is coming in byte wise and filtering it based on its significance.

Overall the lab covered a broad range of ideas from basic algebra (calculating bias and sensitivity), vector analysis (calculating magnitude, pitch, and roll), and signal analysis (filtering the accelerometer signal). All these skills were simple, but also complicated since we had to pull from different sections of our brain, which was a great exercise.