Final Project

by Matteo Pulcini

Submission date: 08-Dec-2020 08:17AM (UTC-0500)
Submission ID: 1468596678
File name: 76736_Matteo_Pulcini_Final_Project_168255_786536891.pdf (422.93K)
Word count: 1714
Character count: 8137

INTRODUCTION

“We are what we repeatedly do. Excellence, then, is not an act, but a habit.” -Aristotle. My goal was to develop a technology that can transform my future by altering my habits. I planned on doing this using the material and technical knowledge I have gained this year, inside and outside the classroom to fix my posture while sitting in my office chair.

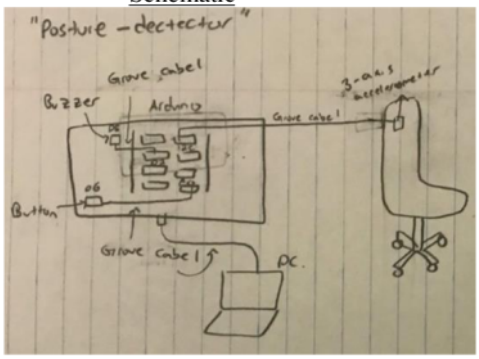
CONTEXT:

I wanted to create something that I can bring with me outside the classroom. Something that I would affect my well-being and improve my study habits. I noticed that my back would be sore after slouching in my chair all day from doing desk work. Throughout the day, I realized that I always need long breaks, even when I wasn't finished with my work, a problem stemming from my poor posture. I wanted to try and increase my productivity by teaching myself better-sitting habits. So, I created a device to frequently monitor my posture and notify me if I start indicating signs of slouching or leaning. I believe if I can be alerted when my posture dips below a standard that I set, then I can change my habit of slouch while doing desk work. Poor posture is bad for your health and more research is being uncovered to back up this claim, for example from google scholar, “Repetitive stress injuries (RSI) and musculoskeletal disorders in the United States and worldwide are increasing at an alarming rate due to the advent of ubiquitous computer usage. [1]” This information is not unknown, and I believe this technology can benefit my peers and me beyond this course to keeping good posture and to negate the consequences that come with poor posture such as, shoulder and back pain, headaches, more fatigue, and a negative affect on a person's mood, according to, Ruben Castaneda at U.S News [2].

TECHNICAL REQUIREMENTS / SPECIFICATIONS

- Accelerometer records acceleration of y and z axis and compares them against a thresehold value that I have tested for. (indicated in my code by, (avg_z > **9.15** || ~(avg_y < **2** && avg_y > -**2**)))
- The Arduino beeps if the measured acceleration is past threshold for more than five seconds
- The Arduino beeps in intervals until the measured y and z components are brought back between their thresholds.
- The program ends if the user presses the button on Arduino Board

Schematic



Figure[1]: Schematic Diagram

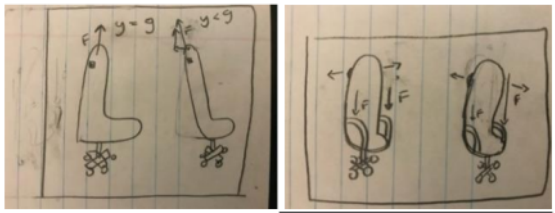


Figure [2]: Movement in z and y demonstaring how components change with posture shift.

COMPONENTS LIST:

- MPU-9250 accelerometer (I2C)
- Grove Beginner Kit
- Button (D6)
- Buzzer (D5)
- Three grove connector cables
- USB-C to USB adapter

Figure [3]: Final Set Up

PROCEDURE:

I knew I could accomplish my goal with a three-axis accelerometer. So, I connected the accelerometer (complex output & complex input) to the Arduino board and used the output values to control my buzzer (simple output), and allowed my button (simple input) to terminate the whole program.

First, I created an array to store my values…

```
%Create array (x,y,z) y=[0,0,0]; z=[0,0,0]; % Initialize parameters i = 1;
```

Then, I decided I needed to create an infinite loop to continuously track my posture. Inside my while loop, my code took data from the accelerometer and filled the (y) and (z) arrays that were initialized previously.

```
%Take acceleration from MPU-9250 aR= readAcceleration(imu);
% Initialize ith index of each axis array y(i) = aR(1); z(i) = aR(2); i=i+1;
%Enter loop ever 3 values every 3 values if(i == 4)
```

This ran on a loop storing my real-time values in my array, and once i==4, the average of y and z values were taken.

```
%Take average and set them in each array
avg_y =  (y(1) + y(2) + y(3))/3; avg_z =  (z(1) + z(2) + z(3))/3;
```

To avoid random errors I added a timer that if I indicated ‘poor posture’, my code would receive positive input from the accelerometer and wait a period of time before checking the average y and z values again and giving me a reminder through a beep. This increased the accuracy and allowed me to move around in

my chair (fidgeting) for short periods of time without setting up my program. The next sequence demonstrates this…

```
%Track time
if (( (avg_z > 9.15 || ~(avg_y < 2 && avg_y > -2))))
 %start time start_time=tic; end_time = 5;
 %Timer
while (toc(start_time)<end_time)end
for c=1:3
%Take acceleration from MPU-9250 aR= readAcceleration(imu);
% Initialize ith index of each axis arrayy(c) = aR(1); z(c) = aR(2);
end
avg_y =  (y(1) + y(2) + y(3))/3; avg_z = (z(1) + z(2) + z(3))/3;
```

Then, the sound will go if the condition is still met because after the while loop an average is taken again…

```
% Initialize conditions if x-axis/x-axis passes threshold for a certain period of time
    while (fix == 0)
          if ((avg_z < 9.15 && (avg_y < 2 && avg_y > -2)))
            fix = 1; break;
          else
%Beep 3 timess
writePWMDutyCycle(a,'D5',0.25); pause(0.2); writeDigitalPin(a,'D5',0);pause(0.5);
writePWMDutyCycle(a,'D5',0.25); pause(0.2); writeDigitalPin(a,'D5',0);pause(0.5);
writePWMDutyCycle(a,'D5',0.25); pause(0.2); writeDigitalPin(a,'D5',0);pause(0.5);
              for c=1:3
                %Take acceleration from MPU-9250
                aR= readAcceleration(imu);
                % Initialize ith index of each axis array
                y(c) = aR(1); z(c) = aR(2);
              end
```

There is always error with any solution and I can only be as accurate as my accelerormeter is which means I am limited to a properly functioning acceleromter. If the acceleromter stops working, the whole program would be useless and provide meaning less data.

***This is missing simple steps such as imu = mpu9250(a); a=arduino('/dev/cu.usbserial-0001','uno');**
Refer to video for full code, THIS IS JUST GENERAL WALKTHROUGH, NOT FULL CODE.

TEST:

Test 1

I created a graph using the code in **Figure [5].** The graph in **Figure [4]** was the result of testing the code and moving one axis if another was held constant at 0. When I moved the z-axis, I kept the y-axis at 0 to determine what direction I would need to move the accelerometer to track the z and y values respectively. This test proved the accelerometer was functioning concerning each component independently, this test was important because I ordered this accelerometer from Digi key, an external provider.

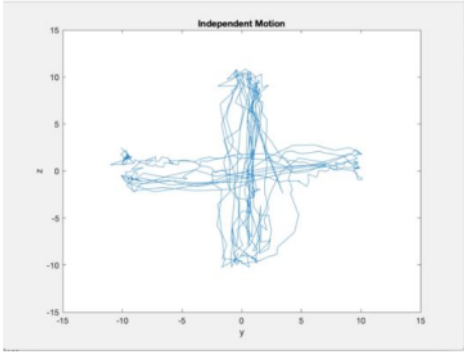


Figure [4]: The accelerometers y and z outputs were Independent of eachother

```
function graph = my_graph(y,z)
  i=1;
  %Connect to Arduino
  a=arduino('/dev/cu.usbserial-0001','uno');

  %Connect to Accelerometer
  imu = mpu9250(a);
  while (true)
    %Take acceleration from MPU-9250
    aR= readAcceleration(imu);
    %Initilize ith index of each axis array
    y(i) = aR(1);
    z(i) = aR(2);

    %Set parameters
    i = i+1;

      %Take average and set them in each array
      plot(y,z); ylabel('z'); xlabel('y'); title('Independent Motion');

  end
end
```

Figure [5]: Code created to plot graph in figure 4

Test 2

The code in **Figure[6]** was used to test to see input parameters…

```
function result=test
        %Connect to Accelerometer
        a=arduino('/dev/cu.usbserial-0001','uno');
        imu = mpu9250(a);
        while true
        aR= readAcceleration(imu);  disp(aR);
        end
end
```

Figure[6]: Function to display components in real time

This code displayed the values of the x and y-axis in real-time (this is one screenshot). I could see how they changed as I shifted in my chair, I leaned back for several seconds and took an average of the z values, then I leaned forward and took another measurement to determine my threshold. I did the same for the y values, but they were leaning to the left and right of the chair. This test was essential for testing what values I needed to use when determining my threshold in real-time.

Figure[6a]: Sitting in chair (respect to z)

Command Window

```
-0.1192    8.6340
-0.1018    8.6328
-0.1371    8.6017
-0.1760    8.6065
-0.2036    8.6586
-0.2096    8.6598
```

Figure[6b]: Leaning forward respect to z)

Command Window

```
-0.2317    8.6801
 0.0970    8.9693
 0.1066    8.8616
-0.0772    8.8951
-0.0641    8.8490
-0.1629    8.7616
-0.1144    8.9023
-0.0796    8.9957
-0.2868    8.8741
```

Figure [6]: The output of accelerometer in order y and z respectively to each column (column with negative values is the y-axis & the z-axis is the column that is similar to the acceleration of gravity). This is the example of me leaning in my chair to test how the accelerometer measured the z component.

CONTINGENCY

The idea I planned on doing was the one I was able to create. Along the way, I had to change certain aspects of the design to achieve my idea. Next time, I would focus on one step at a time and not get devote as much of my time to adding extra features. I realized that I wanted to add features before making sure my code was efficient and this wasted a lot of time. Looking forward, in ENG 4000, I a going to try to create an effective solution before talking about adding additional cool features.

ADDITIONAL MATERIAL

I believe this is a candidate for exceeding expectations because I meet all criteria while demonstrating a social contextualization, and provided medical reports and studies outlining the issue my device tries to fix. The problem that I choose to solve is important and it can change the habit of thousands of students and desk works if they allowed a device like this to remind them of their poor posture. Regarding soft computing skills, I used functions to avoid repeating myself and at the same time increased the efficiency of the code, all why using good coding practice and leaving comments for future readers. I used graphs to prove my technology was working efficiently, as stated previously. I averaged out values to minimize random error, as well as testing at two different times. I did this by looping through values of different inputs, averaging them out, and taking the 3 most recent inputs, and creating one output (less error & ambiguity). On top of this, when 'poor posture' was detected, my machine waited for several sections, then tested again to make sure it was not some mistake or a small movement I made unrelated to poor posture. The real use and application of this device can save people a lot of time which they can be spending doing things they are more interested in, if the scientific research is correct then a device that can monitor posture can only do good things, especially for students. Further benefiting this school and work experience, through altering one simple bad habit.

CONCLUSION

The goal of this project was to create a device to change my future habits, thankfully I was successful in developing the technology, but technology is only good as the person who uses it. I wrote this program to better myself, and that is what I plan on doing, bettering my posture while sitting down, and to do this, I started at the root of a bad habit.

SOURCES

[1] https://content.iospress.com/articles/work/wor01287

[2]https://health.usnews.com/wellness/slideshows/10-ways-poor-posture-can-harm-your-health?slide=11

Final Project

GRADEMARK REPORT

FINAL GRADE

/1

GENERAL COMMENTS

Instructor

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5