# Embedded Challenge Lab Report

Yichao Qin

- SPI Communication

The first thing in this project is to sample data and read the data out from our gyroscope. To do this, I used the SPI bus and set the configuration as it was taught in class. The gyroscope itself could have a maximum ~800hz sampling rate but since we are limited to a 0.5s sampling rate so the difference between different sampling rate is not significant.

- Device set-up

The device is anchored on the side of the thigh with the screen facing outward and the USB port pointing upwards. With this configuration, legs movement will be represented as rotation in the z-axis. When legs move forward, rotation over the z-axis is negative and when legs move backward, the rotation over the z-axis is positive.

Although gaiting has multiple stages, our implementation only considers only forward (negative rotation) leg movement as the contributing stage for forward moving.

- Data interpretation

I take the raw data in our out registers and multiply them by sensitivity to make them real rotation velocity in degree per second.

The converted data represents the actual rotation velocity of the legs by the time of sampling.

- Data transformation

1) To calculate linear distance and linear velocity, I perform three procedures.

First of all, I perform a discrete Fourier transformation on the real DPS data. The motivation to do this transformation is that I find out that when the participant is walking slowly, in other word, the legs angles are changing slower than the sampling rate, there would be a problem that two or more samples are recording a single leg swing. For instance, at t = 0.5s I record a -10 DPS, and at t = 1.0s I record a -20 DPS. During this time interval, these two data belong to one gait cycle. After the Fourier transformation, they will be on the same falling edges, so the algorithm won't repeatedly count them. Another benefit of doing a Fourier transformation is that I can determine the maximum velocity of a swing. When by the time of sampling I am in the middle of the swing, the data I record does not represent the maximum velocity. If the next sample being recorded is positive, then I would completely lose the actual maximum velocity of that swing.

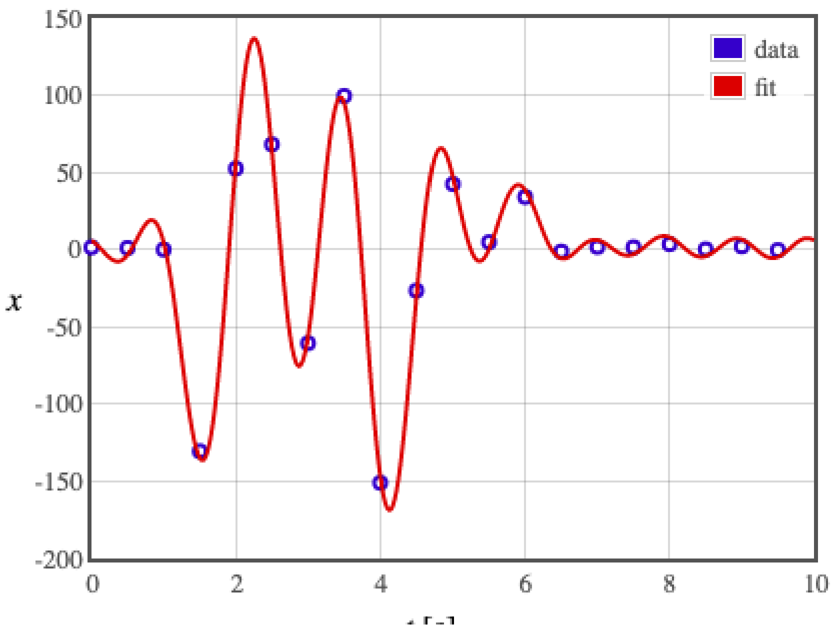
I implement the Discrete Fourier transformation by taking our real DPS as input data and calculating all coefficients for the Fourier series. The new data can be obtained by plugging in the time t.

Fourier series formula:

$$x(t) = \sum_{n=0}^{n<N/2} \left[ a_n \cos\left(\frac{2\pi nt}{N\Delta t}\right) + b_n \sin\left(\frac{2\pi nt}{N\Delta t}\right) \right].$$

where NΔt is the total sampling time

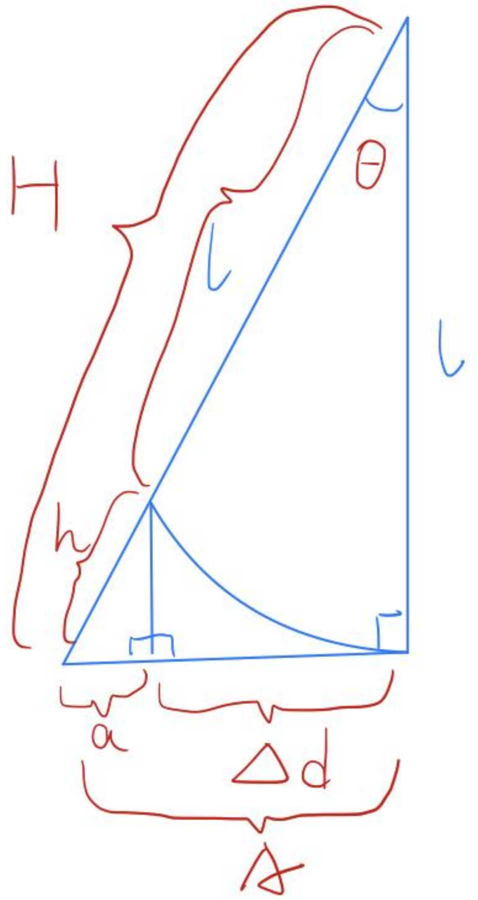
An example of data after the transformation(sampling over 10 sec):




At t = 6.5 the participant stops fully. There are three ditches on the graph meaning we detect 3 steps (one legs).

2) Secondly, we need to compute the change in degree of the leg. To do so, I perform integral over the Fourier series. Since I am only taking negative DPS into account, there is no easy way to deduce an integral formula. I decide to estimate the integral by splitting each second into small segments. In my code, I split each second into ten pieces. And then I calculate the area above each ditch by multiplying the depth at t and length of the t. By doing so, I am able to get the change in degree. In my code, I split each second into 10 smaller pieces. Each piece is 0.1s in length.

3) Finally, I need to convert the angular distance into linear distance.




Let Δd be the linear distance the participant has moved, Θ be the degree of the leg, l be the leg length (the height from device to the ground in cm). I use 77cm as leg length, this may vary based on the person and device set up.

$H = l / \cos(\Theta)$, $h = H - l$
$A = l * \tan(\Theta)$, $a = h * \cos(90 - \Theta)$
$\Delta d = A - a$

- Results

I print my results (in cm) through serial output. I tested on a treadmill, and my modeling works well with all walking paces including alternating paces (walk slowly- >stop->walking quickly..etc). If the device is calibrated so that at stationary rotation readings are all 0s, I could get even higher accuracy. On one of my devices, at stationary the readings are ideal and on the other one, the zero level outputs are significant. In all, I get around 97% accuracy on the later one. I believe the 0.5s sampling rate is limiting my accuracy. And accuracy drop drastically when walking frequency is much faster than sampling rate(2Hz).