Summer Projects, SnT Council
IIT Kanpur

Skin Track

Final Report

July 11, 2018

Contents

1 Acknowledgements

We are grateful to the the Electronics club and SnT council IIT Kanpur for providing us with the resources and motivation for this summer project. We thank our mentors and co-ordinaters of Electronics club: Ashok kumar choudhary and Suraj Hanchinal to guide us for our project.

We are thankful to the Yang Zhang, Junhan Zhao, Gierad Laput, Chris Harrison from Stanford university who are the actual people behind this idea that we have implemented in this project.We are also thankful to Element 14 for making electronics devices that we have made use of in this project.

2 Introduction

Skin track is one of the major part of long term project ”Human Computer Interface” of the Electronics Club IIT Kanpur. It is basically a wearable device that enables continuous touch tracking on the skin by movements of fingers. Basic principle of the project is that our body acts as a good conductor for AC currents and these signals develop a phase difference between them on travelling different paths. We have used a ring to generate AC signals and a band as a carrier with electrodes, which processes and computes the phase differences between AC signals and sends the data to our desktop and helps us in resolving the 2D location of touch by the finger tip.

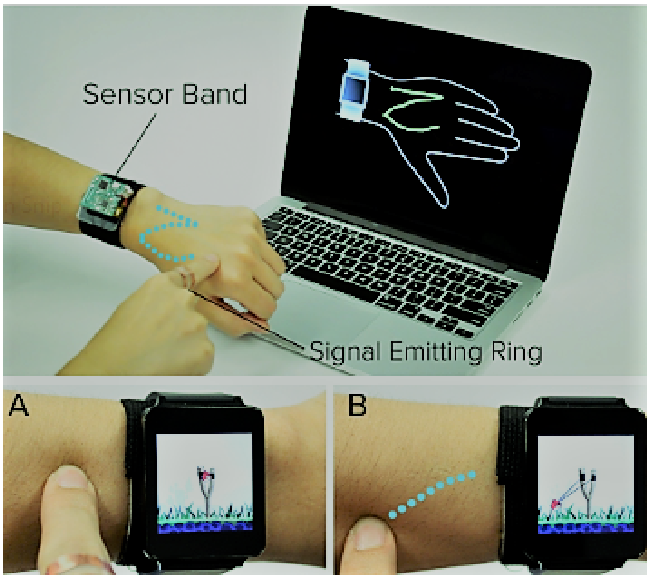


Figure 1: Model of skin track[1]

3 Our Approach in brief

Our project consists of a ring attached to a band which is worn on the user's one hand and generates AC signals. When the finger touches the skin of the other arm the signals travel using the user's skin as a waveguide and reach the strategically placed electrodes at the wrist band which are then fed into the IC's (also present on the wrist band).The IC's return the phase difference and the gain ratio which is then sent to our laptop through a wifi enabled device and a microcontroller. A machine learning algorithm is established that enables us to identify and locate the position of the touch on the arm and then consequently track its position.

[1]The image shown here is taken from actual research paper. Our project is a slightly modified version of this.

4 Why this Approach?

Other technologies that offer such functions generally use cameras over the shoulders or skin overlays on arms which appear socially obtrusive. The biggest advantage of this skin tracking principle is that it requires no direct instrumentation of touch area (i.e. skin overlay) or sensor line of sight (i.e. camera), hence making it more convenient for daily life usage. This particular method of skin track is quite rare and uncommon. This method can be used to increase the working area of small screen gadgets after complete development.

5 Parts and Electronic components

5.1 Ring: signal emitter

Following are the electrical components used to make our ring:

- Coil of copper wire as ring.
- Two 3.7 volt 500 mAh Li-ion Nokia batteries.
- An 8 pin crystal oscillator.
- A switch and diodes

5.2 Sensor band: signal processor

Following are the electrical components we are using to make our wrist band:

- 4 Copper electrodes
- Power bank to supply a constant voltage of 5 Volts.
- 2 Phase difference and Voltage ratio detector integrated chips: AD8302
- A micro-controller chip: ATMega 328
- A WiFi enabled chip : ESP8266-01

6 Implementing Electronics of the Ring

6.1 Circuits in ring

For making oscillator-ring circuit, some facts for the nature of alternating currents that will be produced by the ring are considered. One such fact is the frequency. If the frequency is too low, the phase difference will be too small to accurately measure and if the frequency is too high, the wave could complete a full cycle during propagation producing ambiguous positions due to wraparound. Human body has an average permittivity of 17 at 80 MHz which means an 80 MHz electromagnetic wave propagates at 7.3×10^7 m/s. This gives a peak to peak wave length of 91 cm which results in a phase difference of $\sim 4°$per cm traveled that makes 80 MHz an ideal frequency. Also high frequency will mean more power consumption and more risk of signal leakage and might cause interference and multiple path issues. So these facts were tested using 12.5 MHz and 80 MHz oscillators. Only the latter one could be detected by electrodes. Following is the circuit that we are using in our oscillator.

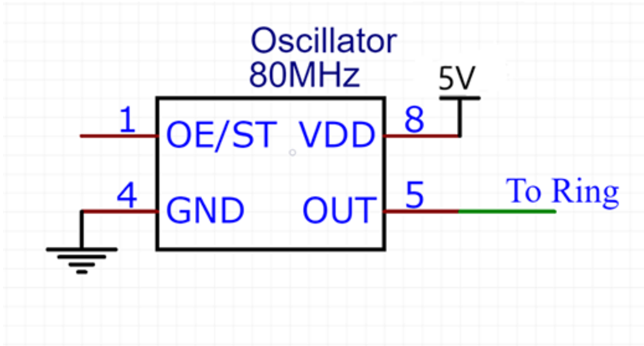
new.png 

Figure 2: Oscillator circuit

Two copper rings are used, one connected to the ground pin and other to the output pin of the oscillator. The oscillator contains self-built integrated circuit of the crystal. These rings are to be worn in the finger. The two rings of copper have a current flow longitudinal to the finger, an arrangement that generates optimal SNR with minimal energy radiating into free air. A thin insulator layer is stuck in the inner portions of the ring to prevent any DC being passed. It also prevents any direct contact to be made between rings due to sweat.

6.2 Power management for ring

Oscillator :- Two 3.7 Volt Li-ion batteries connected in series are used to power the oscillator.
Sensor Band :-This is powered by a power bank which gives a constant voltage of 5 Volt.
For Recharging the batteries: A 6 volt lead-acid battery and potentiometer circuit is used to recharge the batteries.

6.3 Band attached to ring

The band is made of elastic cloth which act as a holder for the battery and the oscillator thus providing a stable connection within them.

7 Signal Receiver

7.1 Sensor Band

The band is made of elastic cloth which contains 4 structured electrodes made of copper plates strategically and symmetrically placed so that after wearing the band, two of the electrodes are on the top of the hand and rest two at the bottom making diagonal pairs. Two diagonal pairs(e.g, left top horizontal and bottom right horizontal) are further connected to 2 AD-8302 devices.The band also has a ground plane made of Aluminum to improve SNR ratio.

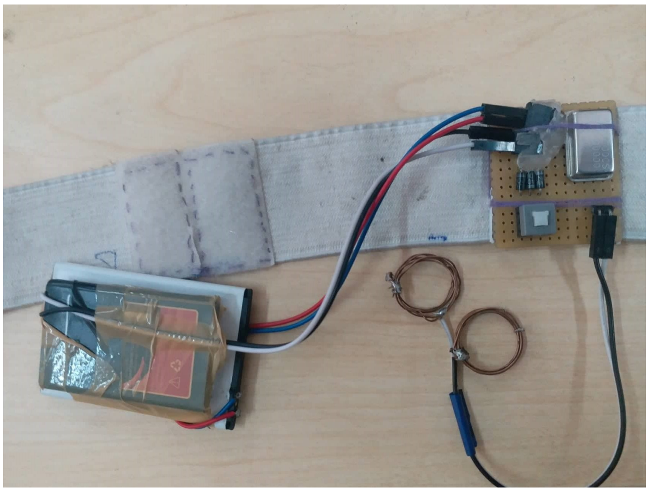


Figure 3: Band prototype

7.2 AD-8302

Analog Devices' AD8302 contains RF/IF gain and phase comparator chips. Each AD8302 IC takes two signals as input, and computes the gain ratio and phase difference. It provides an accurate measurement of gain ratio over $a \pm 30$dB range scaled to 30 mV/dB, and of phase over a 0°–180°range scaled to 10 mV/°. To improve SNR, we replaced the input termination resistors from the recommended 52.3 Ω to 2 MΩ.

Further, Capacitors are placed between the resistors and electrodes to form a high pass filter, dampening low frequency environmental EM noise (e.g. 50 Hz power line). This setup captures a receiving signal of around 23 mVpp when the signal-emitting finger is touching the skin. The prototype sensing board produces four analog values — one phase difference and one gain ratio for each of the two comparison channels.

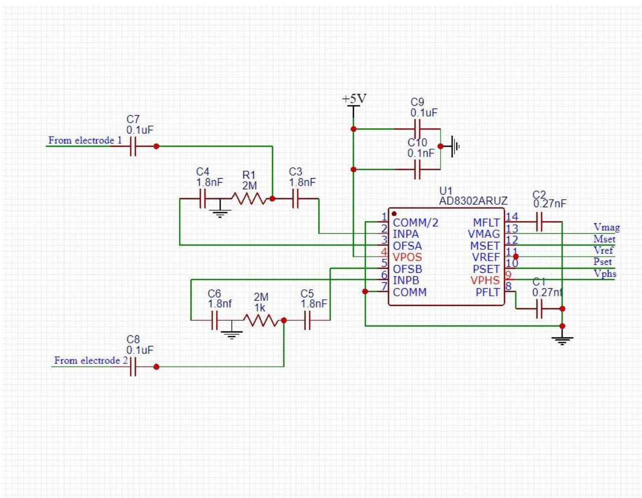


Figure 4: AD8302 circuit

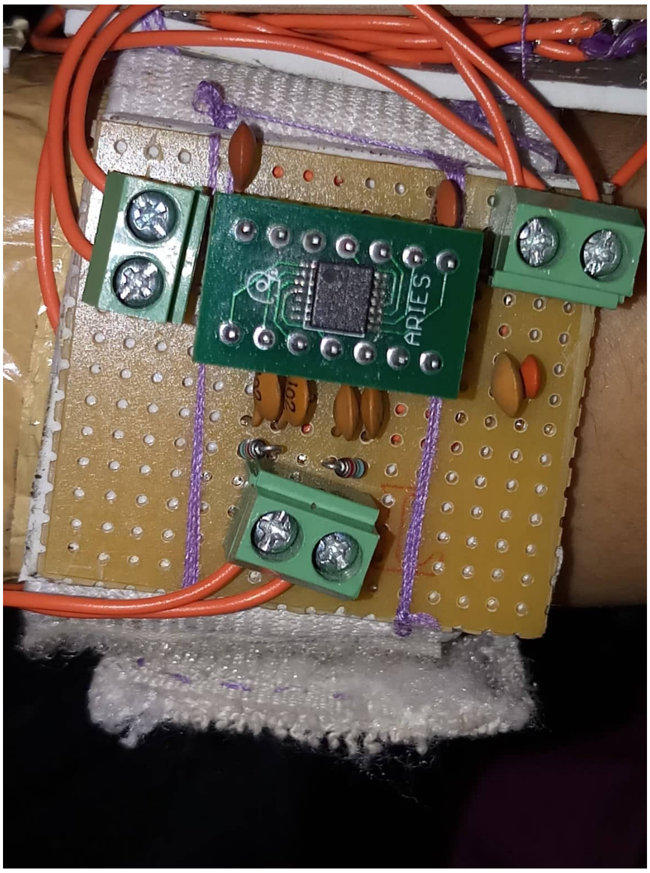


Figure 5: AD8302 circuit made by our team

7.3 ATMega328

7.3.1 Purpose of ATMega328 in our project

ATMega is a microcontroller used for receiving phase difference and gain ratios from the IC's and sends them to the Laptop wirelessly using ESP.
The outputs (phase differences and gain ratios) are sampled using an ATMega328 microcontroller running at 8 MHz, which has eight 10-bit analog-digital converter (ADC) channels. The ESP transmits these values at a suitable rate to the Laptop.

Figure 6: ATMega328 microchip

7.4 ESP8266-01

The main purpose of ESP is to transmit the output, in the form of phase differences and gain ratios, it receives from the ATMega , to the Laptop at a suitable rate.
Insulator covering is used to decrease the interference between 80MHz signal and ESP module while ensuring proper connection between mobile hotspot and ESP.

Figure 7: Esp8266

7.4.1 Working of ESP8266-01

ESP8266-01 is used in ATP mode (connects with a hotspot as it does not behave as a hotspot) with UDP transport protocol. UDP is a simple high speed low functionality "wrapper" that interface applications to the network layer.

UDP sends 8 analog inputs in the form of strings to an interactive device connected to the hotspot(same as that of the esp) which is then stored in an excel sheet upon which data manipulation and machine learning can be applied.

7.4.2 ESP8266 connections

The following connections are made and then a blank code is uploaded.
TX - TX
RX - RX
Ground (ESP8266) - Ground (Arduino)
VCC ESP8266 - Arduino(5V)
GPIO0 - Ground(It means ESP8266 in programming mode)
Arduino reset (Ground directly send the code to ESP8266)
ESP reset (not connected)
CHPD - 5V (Same power given to VCC)

GPI02- (not connected)(It is used further for sending signals in hardware circuits, not used now.)

Then upload the main code and then change the connections as follows:
TX - RX
RX - TX
Ground (ESP8266) - Ground (Arduino)
VCC - Arduino (5V)
GPI00- (Not connected)
Arduino reset (Not connected)
ESP reset (Not connected)
CHPD - 5V (Same as VCC)
GPI02 - (Not connected)

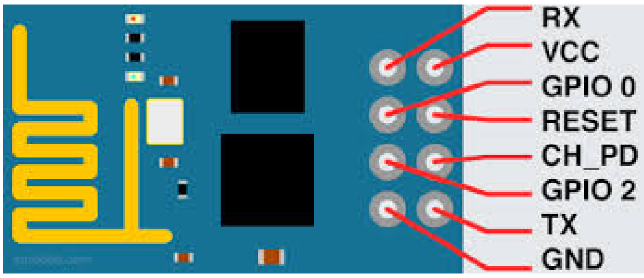


Figure 8: Esp8266 connection

8 Modification of the data obtained

The data obtained was first smoothed using the smoothing algorithm. Then to eliminate the redundant noise, noise filter algorithms were applied.This modification helps us eliminate noise,ensuring that the data sent to the machine learning algorithm is within the correct range. This will help us in getting desired output after applying machine learning algorithm on the data.
The filters applied were:

- Low-pass butterworth filter
- Median filter
- savgol_filter

These filters are described in detail below:

8.1 Low-pass butterworth Filter

The Butterworth filter is a type of signal processing filter designed to have a frequency response as flat as possible in the passband. It is also referred to as a maximally flat magnitude filter.

The frequency response of the Butterworth Filter approximation function is also often referred to as “maximally flat” (no ripples) response because the pass band is designed to have a frequency response which is as flat as mathematically possible

Higher frequencies beyond the cut-off point rolls-off down to zero

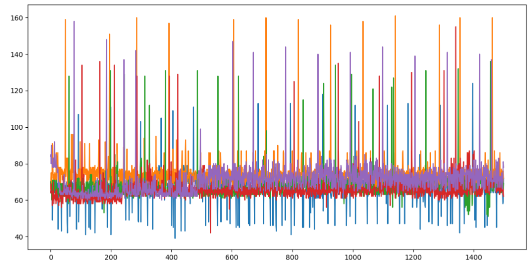


Figure 9: Raw Data

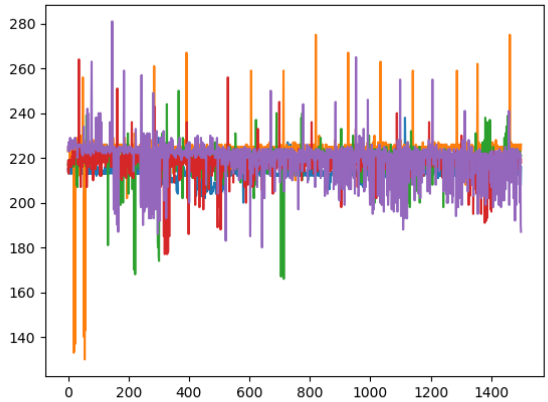


Figure 10: After using Low-pass Butterworth filter

8.2 Median Filter

The main idea of the median filter is to run through the signal entry by entry, replacing each entry with the median of neighboring entries. The pattern of neighbors is called the "window", which slides, entry by entry, over the entire signal. For 1D signals, the most obvious window is just the first few preceding and following entries, whereas for 2D (or higher-dimensional) signals such as images, more complex window patterns are possible (such as “box” or “cross” patterns). Note that if the window has an odd number of entries, then the median is simple to define: it is just the middle value after all the entries in the window are sorted numerically. For an even number of entries, there is more than one possible median.

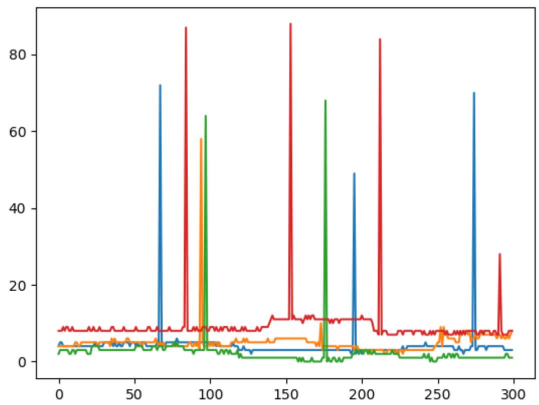


Figure 11: Before applying Median filter

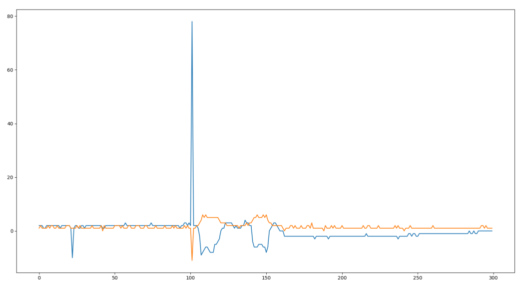


Figure 12: After applying Median filter

8.3 Savgol Filter

Savgol_filter is an acronym for SavitzkyGolay filter.A SavitzkyGolay filter is a digital filter that can be applied to a set of digital data points for the purpose of smoothing the data, that is, to increase the signal-to-noise ratio without greatly distorting the signal. This is achieved, in a process known as convolution, by fitting successive sub-sets of adjacent data points with a low-degree polynomial by the method of linear least squares. When the data points are equally spaced, an analytical solution to the least-squares equations can be found, in the form of a single set of "convolution coefficients" that can be applied to all data sub-sets, to give estimates of the smoothed signal, (or derivatives of the smoothed signal) at the central point of each sub-set.

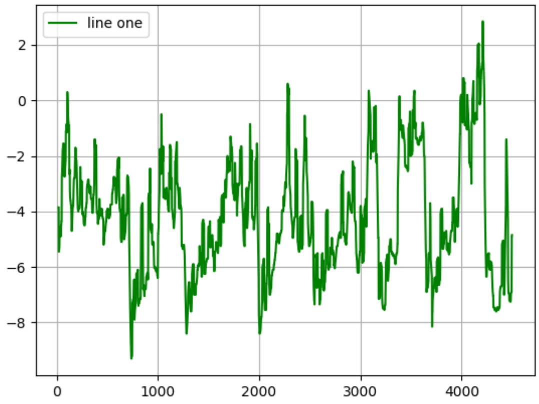


Figure 13: Before applying Savgol filter

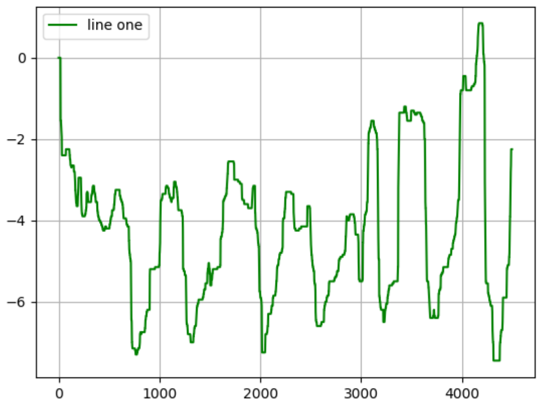


Figure 14: After applying Savgol filter

9 Machine learning

9.1 Why use Machine Learning here?

There are two main reasons to use Machine Learning :

- To be able to detect or rather classify touch and non touch conditions of the finger on the arm
- To be able to track the touch points using regression models.

The machine learning algorithm has been successfully trained to classify touch and nontouch states of the finger.The regression and classification models used are as follows:

- Support Vector Machine
- K-Nearest Neighbours

9.2 Classes of Dataset

- No touch
- Single tap
- Double tap

9.3 Features used for Classification

- Number of peaks
- Range
- Skew
- Kurtosis
- Kurtosis frequency domain
- Standard deviation
- Number of slope sign change
- Interquartile range
- Median standard deviation

9.4 Support Vector Machine (SVM)

Support vector machine (SVM) is a supervised machine learning algorithm in which we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the optimal hyperplane that maximizes the margin between two classes. The vectors that define the hyperplane are support vectors.

It maps data to high dimensional space where it is easier to classify with linear decision surfaces: reformulate problems so that data is mapped implicitly to this space.

9.5 K-Nearest Neighbours (KNN)

K-Nearest Neighbors is a simple and effective machine learning classification algorithm overall. It works on minimum distance from the query instance to the training samples to determine the K-nearest neighbors to be the prediction of the query instance.

KNN makes prediction using the training dataset directly. Predictions are made for a new instance (x) by searching through the entire training set for the K most similar instances (the neighbors) and summarizing the output variable for those K instances. For regression this might be the mean output variable, in classification this might be the mode (or most common) class value.

To determine which of the K instances in the training dataset are most similar to a new input a distance measure is used. For real-valued input variables, the most popular distance measure is Euclidean distance.

$$\textbf{Euclidean Distance(x,y)} = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_i - y_i)^2}$$
$$= \sqrt{\textstyle\sum_{i=1}^{n}(x_i - y_i)^2}$$

10 References

- Paper on Skin Track by Yang Zhang, Junhan Zhao, Gierad Laput, Chris Harrison (http://yang-zhang.me/research/SkinTrack/SkinTrack.pdf)
- Datasheet of AD8302 (http://www.analog.com/media/en/technical-documentation/data-sheets/AD8302.pdf)
- Datasheet of ESP8266-01 (https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf)
- Datasheet of AEL1200CS (http://www.quartz1.com/price/PIC/405N3306000.pdf]
- Code for ESP programming (https://github.com/eclub-iitk/skintrack)