TENNIS SWING ANALYZER

October 16, 2018

ECE 445 Team 37

Yihong Liu (yliu287)

Guo-Cheng Lo (glo2)

Heting Gao (hgao17)

TA: Amr Martini

Contents

Introduction

Objective

When playing tennis, tennis beginner often resorts to their wrist or forearm to perform a strong forehand or stroke. Such nonstandard performed swing does not necessarily result in a weak stroke and is hard for beginners to realize. However, if the players keep performing bad swings, in the long run, these non-standard swings could result in diseases such as wrist pain or wrist injury [8]. Therefore, it is important not only to hit the ball properly but also to hit the ball with a well-performed swing.

We want to build a tennis racket swinging analyzer that can analyze and determine if a swing is standard or not for tennis beginners. It measures the player's arm motion using two sets of inertia sensors. Their data are retrieved and calibrated by micro-controllers and then transmitted to a personal computer via Wi-Fi modules. We plan to take several standard swings and non-standard swings, collect their acceleration data and label them standard or non-standard respectively. A classifier would be trained on these data to determine whether the swing is a good swing or not and a speaker will directly report the result to the user in real time.

Background

There have been several well-developed product on the market. Most of them focus on the recording characteristics such as intensity, speed, and trajectory of each swing. One example is Zepp Tennis Swing Analyzer, which is a small sensor integration attached to the bottom of the racket and transmits the collected data to a mobile device for ad-hoc analysis. Another set of products focus on analyzing the player's performance in a full match. Pivot, a product from TuringSense is a system consisting of a total of 14 sensors that aims to analyze people's 360-degree motion [1]. While this product provides professional insight into the player's performance, it is of high price and is hardly affordable for amateur tennis beginners.

Wrist injury is a very common disease among tennis players. It may occur from direct trauma, but most injuries occur due to chronic overuse [10]. Such overuse can result from longtime practice, in case of professional players, but can also result from nonstandard wrist movement in case of amateur players. Although there have been wide debates on the best tennis swing form regarding arm movement, and there might not exist a single best answer, tennis experts typically agree on a set of criteria on the wrist that are necessary for good swings. Most of the modern tennis coaches would agree that it is through spinning the wrist that the racket head achieves the highest speed, yet such spinning must be strictly constrained; before hitting the tennis ball the wrist should hold in a laid back position [8] and during the racket-ball contact the wrist stays laid back and could spin only in the direction perpendicular to the arm in order to roll over or whip over the ball. Such requirement not only stabilizes one's strokes but also greatly reduces unnecessary wrist motion, which, if being excessively carried out, is the source of the wrist injury and other related diseases [9].

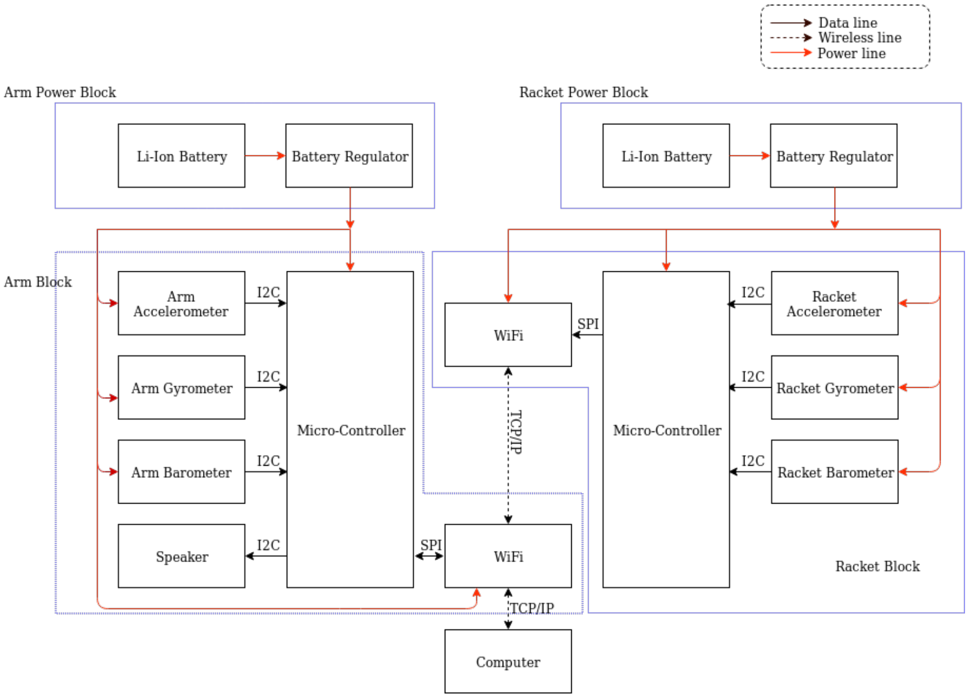
We, therefore, focus our design mainly on wrist part, using two sets of sensors to measure the accelerations of the wrist and the forearm and providing insight into how the player moves his or her wrist during the swing movement. We hope with such an affordable design we could make the tennis beginners notice their improper wrist usage so that wrist injuries can be reduced in the first place.

High-Level Requirements

1. The classifier correctly classifies a good swing and a bad swing and achieves an accuracy of 85% in the training stage and a false positive rate of 10% in worst case.
2. The sensors could all satisfy the precision requirement defined in Function Overview part.
3. The arm micro-controller could receive the swing data via WiFi within 0.5s from racket micro-controller.
4. The classifier can successfully perform feature extraction and classification on the arm micro-controller within 1s so the player can get immediate feedback on his or her swing.
5. The arm micro-controller could report the result by signaling the speaker to produce a detectable sound.

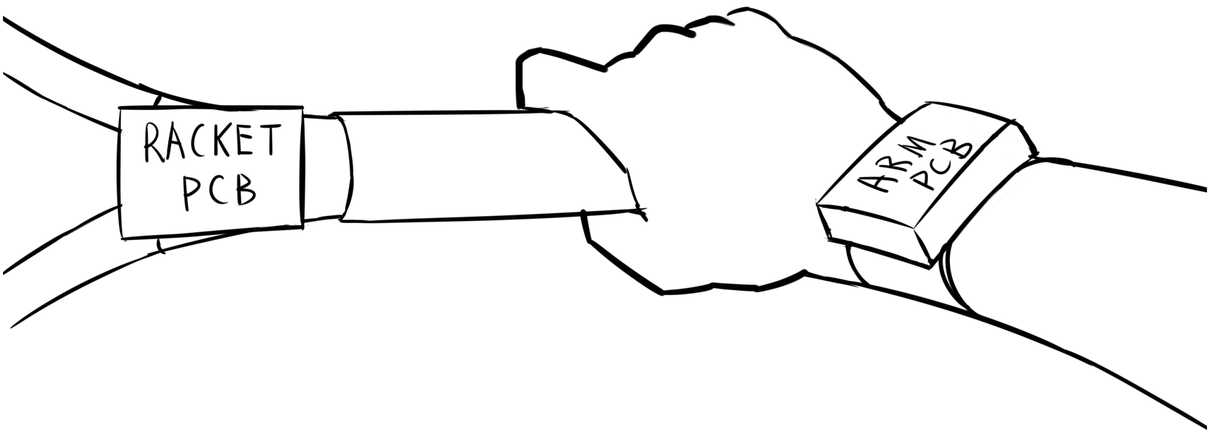
Design

Block Diagram



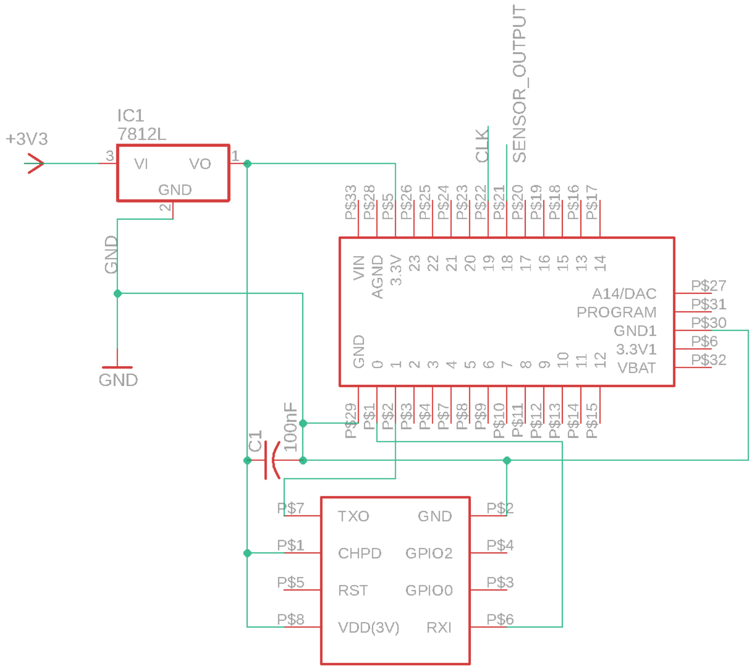
Mechanical Design

We plan to have two set of systems on the racket and on on the player's wrist. The racket PCB would collect, pre-process and send data to arm PCB for classification task in application phase.

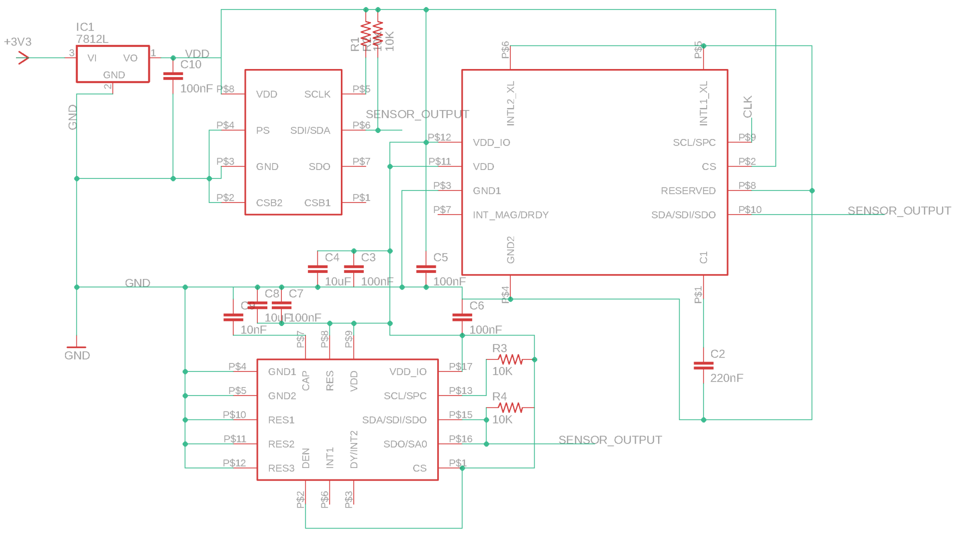


Schematic

Teensy Controller Schematic



Sensor System Schematic



Functional Overview

Sensor Subsystem

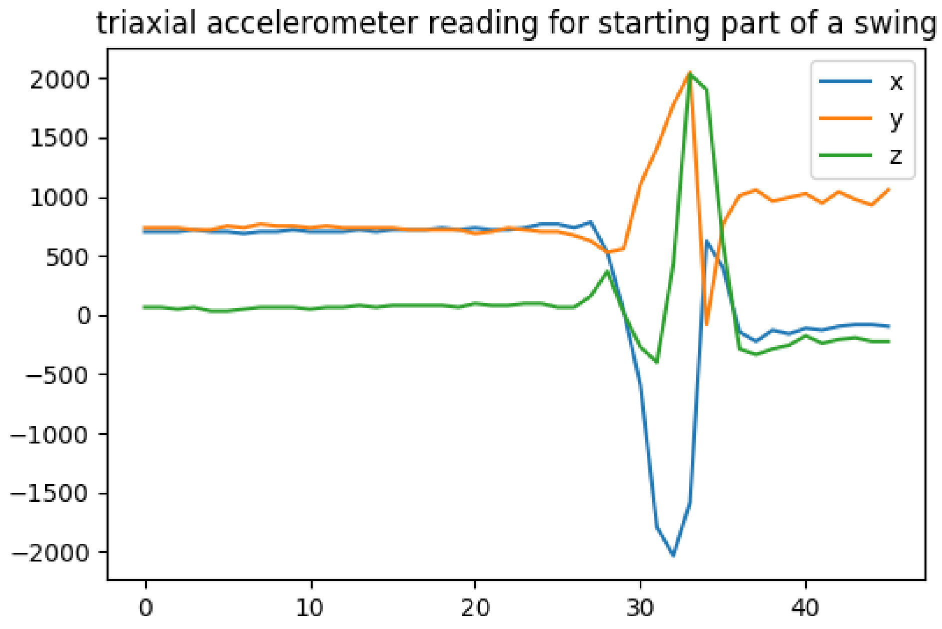
We plan to have two sensor sets, each set consists of an accelerometer, a gyroscope and a barometer, with one set on the forearm and the other on the racket. Each subsystem is powered separately. With the data collected by the accelerometer and the gyroscop combined, we can compute the quaternion of rotation and from the quaternion we can compute transform the acceleration to an absolute coordinate for each time frame, using Madgwick's sensor fusion [5].

Let **a**, **g**, **q** denote accelerometer vector, gyroscope vector, and quaternion respectively. Let $Rotate$ denote the rotation operation parametrized by **q** [6]. We have [7]

$$\begin{aligned} \mathbf{q} &= Madgwick(\mathbf{a}, \mathbf{g}) \\ \mathbf{a}_{absolute} &= Rotate(\mathbf{a}|\mathbf{q}) \end{aligned}$$

The barometer is used to measure the relative height between the racket and wrist and serves as an additional feature of a swing.

We record the accelerometer reading for the starting part of a swing and find that there exists a peak in acceleration when the player starts his swing.



The horizontal axis is the number of samplers and since sample rate is $10Hz$, one can easily convert it to seconds. The vertical axis is in unit of $\frac{1}{1000}g = 9.8 \times 10^{-3} m/s^2$. At the beginning of the swing, we can use the accelerometer to detect the start peak and begin collecting data from sensors. Each micro-controller will also save the time when they start collecting and these time stamps would be used to align the acceleration series.

Device	Requirement	Verification
Barometer	The device will measure the relative change in height with an accuracy of +/- 5 centimeters. Able to send out data to micro controller through I2C interface.	1. Set accelerometer and micro-controller on the breadboard. 2. Fix the breadboard on ground read the initial data. 3. Place the breadboard up 1 meter up from origin position, make sure output value is within 5 centimeters.
Accelerometer	The device will measure the relative change in height with an accuracy of +/- 8mg Able to send out data to micro controller through I2C interface.	1. Set accelerometer and micro-controller on the breadboard. 2. Set the breadboard on a horizontal table or ground 3. Read acceleration data from micro-controller directly and make sure one axis has value 9.8+/-8mg, and the other two axes have 0+/-8mg.
Gyroscope	The device will measure the relative change in height with an accuracy of +/- 300 mdps Able to send out data to micro controller through I2C interface.	1. Set gyroscope and micro-controller on the breadboard. 2. Stick the breadboard with iPhone 3. Compare the value from gyroscope to the value on iPhone gyroscope and make sure the difference is within 300 mdps.

Arm Micro-controller Subsystem

The micro-controller will calibrate the sensor and set up the WiFi at the beginning. After that, we will ensure the communication between the two micro controller subsystem is correctly setup and then synchronized the clock. Then it takes sampled data from the three sensors at a frequency of 100Hz, updating and shifting data in a larger output buffer than buffer in racket MC.Racket micro controller will send a data set with time stamp to arm MC after detecting swing. Arm micro controller will find the beginning of swing according to time stamp.The large data buffer will allow it to wait for input data from racket without losing any arm data. Then it performs sensor fusion techniques to get a combined estimation of accelerations of the racket.

In the training phase, the data will be pre-processed and then send to WiFi modules to a computer using TCP/IP socket. In the application phase, after pre-processing the data, it performs classification directly using trained classifier transported from the computer and signal the speaker to report the result.

We plan to use Teensy 3.2 micro-controller. It is a micro-controller with 32 bit ARM cortex and it contains 64KB RAM and 34 digital pins as well as support for real-time FFT computation. It has an input voltage of 3.3V and operating current of 250mA It can be ordered from Sparkfun[1].

We plan to take data from 3 sensors, with the accelerometer and the gyroscope providing three 32-bit floating point measurement on 3 different axes and the barometer one 32-bit. Since our sampling rate is $100Hz$ and we measure a period of 5s movement from the starting point, we would have a total of 14KB of data collected from sensors. After performing sensor fusion, we should get a set of calibrated triaxial acceleration data and the data size is therefore reduced to 8KB. There will be another 8KB transmitted from Racket controller, so the controller would have to store in a total of 16KB of data. Then data will be transmitted to the computer during the training phase and

[1] https://www.sparkfun.com/products/13736

will be processed on itself during the application phase.

$$
\begin{aligned}
& \#sensor \times \#axis \times data\_size \times sample\_rate \times duration \\
= \quad & (2 \times 3 + 1 \times 1) \times \frac{32bit}{axis \cdot sample} \times 100sample/s \times 5s \\
= \quad & 112000bit \\
\approx \quad & 112Kbit \\
= \quad & 14KByte
\end{aligned}
$$

Requirement	Verification
Support sending out data through Wifi module using TCP socket.	1. Randomly generate data bits from micro-controller. 2.Set up Wifi module, connecting laptop hot spot. 3. Transmit data to a laptop through TCP socket and print them
Support reading data from three different sensors	1.Set up all sensors and using micro-controller getting data 2.Test Wifi module before testing sensors. 3. Ask micro-controller require a certain amount of samples from sensors and send them to the laptop using Wifi module.
Micro-controller can require data from sensors in a frequency of 300 samples per second (3 sensors per controller, 100 samples per sensor).	1. Set up power and WiFi modules. 2. Let the micro-controller sample each sensor at a frequency of 100Hz, recording the start and ending time using an internal real-time clock. 3. See if the number of samples from each sensor reached 100 per second.
After getting trained classifier from the computer, the main micro-controller can come out result in 1 second after collecting sensors data.	1. After training classifier on the computer, transmit it to the micro-controller using WiFi module. 2. Perform a swing and collect data from sensor. 3.Use real-time clock inside micro-controller to record starting and ending time of running classifiers. Check if the output time is within our limit by subtraction.

Racket Micro-controller Subsystem

The micro-controller subsystem on the racket is similar to the Arm Micro controller because it also needs to calibrate and fuse the sensor data on the racket and set up the WiFi connections as well as synchronize its clock with other micro-controller. However, it does not need to perform classification task; it only transmits the data via WiFi to Arm Micro-controller for further processing in both training phase and application phase. There is a internal buffer inside racket micro controller to keep updating and shifting data. In order to detect whether swing has initiated, the racket micro-controller needs detected a peak value of at least 4g for 0.5 seconds. When detected, the micro-controller will truncate a window out of buffer according to the start of the high-G reading.It will send whole data set attached with a timestamp labeling beginning of swing to Arm micro controller.

Since the micro-controller on the racket does not require any additional features from Teensy 3.2. It will be easier to reuse the same micro-controller as the racket micro-controller subsystem. The controller would collect 14KB data (calculated in the previous subsection), fuse them to 8KB and transmit them to arm controller.

Requirement	Verification
Support sending out data through Wifi module using TCP socket.	1. Randomly generate data bits from micro-controller. 2. Set up Wifi module, connecting laptop hot spot. 3. Transmit data to a laptop through TCP socket and print them
Support reading data from three different sensors	1. Set up all sensors and using micro-controller getting data 2. Test Wifi module before testing sensors. 3. Ask micro-controller require a certain amount of samples from sensors and send them to a laptop using Wifi module.
Micro-controller can require data from sensors in a frequency of 300 samples per second (3 sensors per controller, 100 samples per sensor).	1. Set up power and WiFi modules. 2. Let the micro-controller sample each sensor at a frequency of 100Hz, recording the start and ending time using an internal real-time clock. 3. See if the number of samples from each sensor reached 100 per second.

WiFi Subsystems

There are two connections in the design. One is MC to MC WiFi communication and the other is MC to PC communication. MC-MC communication transmits data from Racket Micro-controller to Arm Micro-controller. MC-PC communication transmits data to the personal computer in the training phase and does nothing in the application phase. Since WiFi communication uses TCP/IP protocol and the data size is small, the transmission should be very reliable. We consider placing our computer 5 meters away from the player, which is roughly half of the width of the tennis court and will not affect the player's swing.

Requirement	Verification
Micro-controller can transmit data to a laptop in a 5-meter distance without 100% success.	1. Generate sequential data bits from the micro-controller with a certain order and place laptop 5 meters far from sensors. 2. Set up WiFi module, connecting laptop hot spot. 3. Transmit data to the laptop through TCP socket and print them. 4. Code a small program to verify the correctness of received bit sequence according to a predefined order.

Computer

The computer would receive data from WiFi. In the training phase, the computer would be collecting a set of standard and non-standard swing acceleration and barometer data. The acceleration data is calculated on the micro-controller by applying sensor fusion on the gyroscope and accelerometer data. The resulting accelerations will be in an absolute 3-axis-coordinate. We labels the data manually. We will then implement a set of classifiers starting from hand-crafted decision tree [12] to a more advanced model. One advanced approach would be extracting statistical features like mean, variance time peaks and bin distribution and use SVM for classification, following the steps in [10, 11]. Or we may divide the time series to frames and use Fourier transform for frequency domain features. Another approach may use the frames directly and use RNN [13] or HMM [14] to model time series. Among the two approaches, we will pick the one with higher accuracy.

We would expect the baseline accuracy to be at least 80%, because papers conducting activity recognition experiments on PAMAP2 [15] usually report an accuracy is about 75% to distinguish two similar activities such as upstairs and downstairs movements. The commonly reported accuracy is roughly 95% to distinguish two dissimilar activities such as walking and lying.

In the application phase, the sensor fusion task will be distributed on each micro-controllers and the trained classifier will be stored on Arm Micro-controller to build an embedded system.

Requirement	Verification
The accuracy should be at least 80% and a false positive rate should be less than 20%	split the data into training and testing data. Should get 80% overall accuracy or above on testing data

Power System

A Lithium button battery will be used to power the micro-controller subsystem and the sensor subsystem. There will be 2 sets of identical power subsystem to power the subsystem on the racket. 3.7 Volts is chosen due to the voltage limit on the micro-controller and sensors. Also, a voltage regulator is used to maintain input voltage at a certain level in order for sensors to work properly

Requirement	Verification
Provide stable $3.7V$ voltage for micro-controller subsystems and sensor subsystems. Provide a maximum current of at least $300mA$ for micro-controller subsystems and sensor subsystems	Full charge the battery Measure if the output for battery's output voltage and current meet our requirements, recording the working time.

Cost and Schedule

Cost

Assuming the hourly rate for each person is $\$45$ and 15 hours per week. Also assuming 16 weeks for designing this project. The total labor cost for this project would be

$$3 \times \$45/hr \times 15hr/week * 16week = \$32,400$$

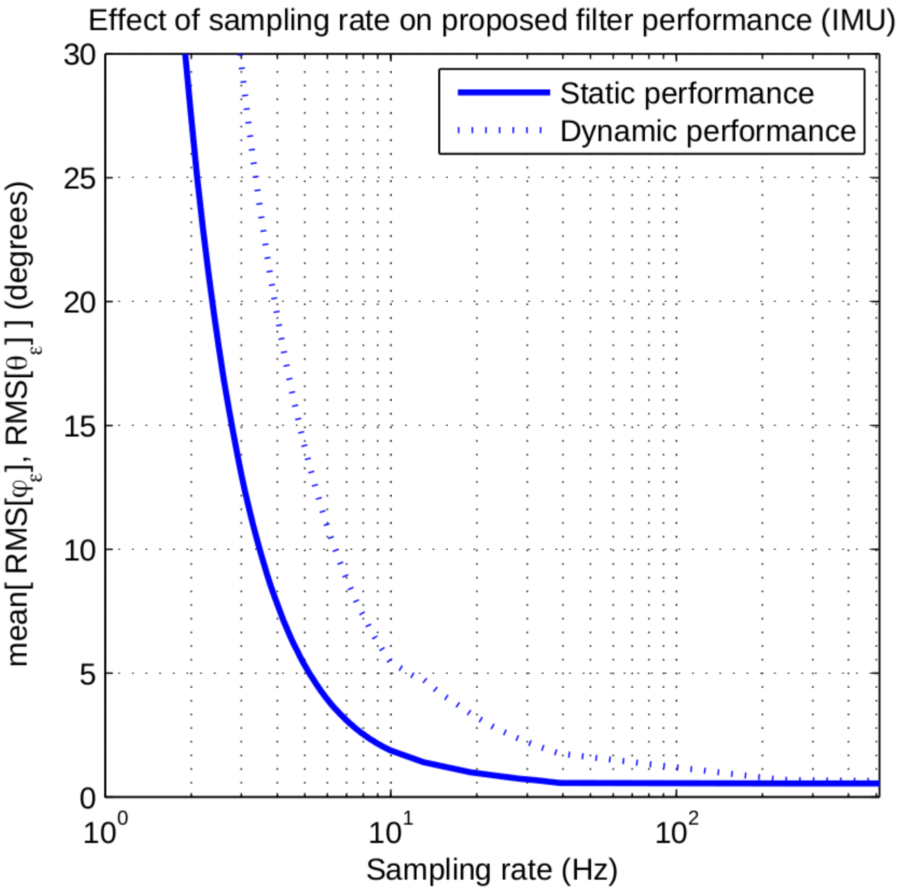
Description	Parts	Quantity	Manufacturer	Cost
Microcontroller	Teensy 3.2	2	Sparkfun	$39.6
Barometer	MS561101BA0350	2	DIgiKey	$24.96
Gyroscope	L3GD20HTR	2	Mouser	$6.64
Accelerometer	511-LSM303AHTR	2	Mouser	$5.84
Wifi Module	ESP8266	2	Sparkfun	$13.9
Speaker	433-1104-ND	1	Digikey	$1.64
3.3 Voltage Regulator	LD1117D33CTR	2	Mouser	$1.06
Polymer Li-ion Rechargeable Battery 3.7V/850mAh	DTP603443	2	Sparkfun	$19.9
total cost				**$113.54**

Schedule

Week	Samuel	Heting	Yihong
10/8	Purchase micro-controller, battery, voltage regulator	Purchase barometer, gyroscope, speaker	Purchase accelerometer, WiFi module
10/15	Assemble different components	Analyze data from sensors	Collect samples
10/22	Order first PCB	Code and test classifier	Debugging wifi module
10/29	PCB soldering	Reorder parts that are needed	Refine circuit design and test circuits
11/5	Make final PCB modifications	Integrate everything together	Unit test sensor subsystem
11/12	Second round PCB soldering	Unit test classifier	Unit test micro-controller subsystem
11/19	Work on classifier code	Work on classifier code	Work on classifier code
11/26	Integration testing on PCB	Integration testing on PCB	Integration testing on PCB
12/3	Testing on final product	Make final modifications	Fine tune classifier
12/10	Work on Final paper	Work on Final paper	Work on Final paper

Tolerance Analysis

We will be using Madgwick's sensor fusion to get an estimate of absolute-coordinate accelerations, which does not require highly accurate sensor but can get a relatively accurate estimate. At a sampling rate of 10Hz, the RMS (root mean square error) of the estimated angular rate will be lower than 2°. Since we use a sample rate of 100Hz, the error will be further reduced to less than $1° = \frac{\pi}{180} rad$ as reported in Madgwick's report [5].



We will then use the quaternion to rotate acceleration data. Let $\mathbf{R} \in \mathbb{R}^{3\times3}$ denote the rotation matrix and $\mathbf{a}$ denote the acceleration data. The rotated acceleration $\mathbf{a}_r = \mathbf{Ra}$. Since rotation is a convex operation and the accuracy of acceleroeter is $8mg$ for each axis, the RMS will still be around $\Delta\mathbf{a}_r = \frac{\pi}{180}\Delta\mathbf{a} \le \frac{\pi}{180}3 \times 8mg = \frac{24\pi}{180}mg$. If we assume the error is distributed as a zero mean Gaussian, $e_a \sim \mathcal{N}(\mu = 0, \sigma = \frac{24\pi}{180}mg)$. For a acceleration series of length T, the integrated distance is d and the error of integrated distance is e_d

$$\begin{aligned}
d &= \int_0^T \int_0^t a(\tau) d\tau dt \\
&= \sum_0^T \Big(\sum_0^t a(\tau) \Delta\tau \Big) \Delta t \\
&= (\Delta t)^2 \sum_{t=0}^{T} \sum_{\tau=0}^{t} a(\tau) \\
&= (\Delta t)^2 \sum_{t=0}^{T} \sum_{\tau=0}^{t} (a(\tau) + e(\tau)) \\
e_d &= (\Delta t)^2 \sum_{t=0}^{T} \sum_{\tau=0}^{t} (e(\tau)) \\
&\sim N\Big(\mu = 0, \sigma = \frac{24\pi}{180} mg \Big(\frac{(T+1)T}{2}\Big) \Delta t^2\Big)
\end{aligned}$$

We plan to take $\Delta t = 0.01s$ $T = \frac{2s}{\Delta t} = 200$. RMS or σ.

$$\begin{aligned}
\sigma &= \frac{24\pi}{180} mg \Big(\frac{(T+1)T}{2}\Big) \Delta t^2\Big) \\
&= \frac{2\pi}{180} \times 10^{-3} \times 9.8 \times \frac{1}{2} \times 200(200+1) \times \Big(\frac{1}{100}\Big)^2 \\
&= 8.1mm
\end{aligned}$$

Therefore, with our acceleromter and gyroscope accuracy, we would expect a standard deviation of 8.1mm in the integrated distance, which should be enough for classification.

Risk Analysis

Since our project is mainly depended on the three sensors, accelerometer, barometer, and gyroscope, the data received by these sensors need to be precise in order to allow classifier to classify correctly. In other words, the measurement from these data needs to have as less bias as possible. Also, another main requirement for this project would be the classifier to be able to classify in high accuracy. If the classifier fails to classify correctly due to incorrect estimation of the position or biased measurements, then the project will not succeed.

On the topic of inaccurate classification, the classifier may produce false positive and false negative that potentially can worsen a person's swinging posture. If the classifier classifies a correct swinging posture as incorrect (false negative), then the person's swinging posture may deviate from the standard correct posture. On the other hand, if the classifier classifies an incorrect swinging posture as correct (false positive), then the person will not be able to improve his/her swinging because the classifier reports incorrect results. These two are potentially cases where the user may suffer from unable to converge to correct swinging posture. Since the aim of this product is to remind the tennis beginner of their nonstandard swings, the false positive case should receive more attention when optimizing our classifier.

Also, consistent result from classification is also important. If the result is inconsistent, users may feel frustrated or confused toward the feedback. Therefore, having an inconsistent result may worsen a person's swinging posture or even make the person's swinging posture converge to a bad swinging posture. Therefore, the classification must be consistent and accurate in order to reduce the potential risks associated with this project.

Safety and Ethics

There are several possible safety hazards in our design. Lithium battery can explode if it is overcharged or heated. We will monitor its temperature and make it open circuit whenever it reach dangerous temperature.

As an electrical device used in tennis courts, moisture and temperature should be taken into consideration of protection consideration. Water can lead short circuit and direct sunlight may heat up the PCB to a dangerous temperature. We want to cover our design with material that can isolate board from water but not accumulate heat inside, making our design work in appropriate temperature and moisture. More specifically, the cover material around design will need to protect micro-controller and sensors from external factors like water and heat in order to ensure safety. The cover design would be similar to a box which will be attached to racket and to arm.

Our design will analyze player's swing and compare the captured data with standard swing models stored in system, then classifying those swings to be good or not by machine learning algorithm. This implies IEEE Code of Ethics, # 3:"To be honest and realistic in stating claims or estimates based on available data. "[3] The swing analyzer will only calculate reflect on recorded player's swing data and won't fabricate output.

We believe our design can help tennis beginners maintain a proper swing posture and prevent them from harming their health. It shows an implementation of IEEE Code of Ethics, # 5 and #7 : "To improve the understanding by individuals and society of the capabilities and societal implications of conventional and emerging technologies, including intelligent systems.""To seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others." [3]

Unfortunately, the analyzer will classify players' swings as "good" or "bad". The result of classification can be used to appropriately to attack tennis beginners, which may violate IEEE Code of Ethics,# 9 "to avoid injuring others, their property, reputation, or employment by false or malicious action".[3] Apparently, we do not have good solutions to that and can only advice player's to view the analyzing result as relevant reference instead of unquestionable answer.

References

[1] Conor Elison, 'Best tennis wearables and trackers for a better technique', 2018. [Online]. Available: https://www.wareable.com/sport/best-tennis-trackers-sensors-wearables. [Accessed:18-Sep-2018].

[2] OptimumTennis, 'THE TENNIS FOREHAND AND THE ROLE OF THE WRIST', 2015.[Online]. Available: https://www.optimumtennis.net/tennis-forehand-wrist.htm. [Accessed:18-Sep-2018].

[3] Ieee.org, 'IEEE Codes of Ethics', 2018. [Online]. Available: https://www.ieee.org/about/corp-orate/governance/p7-8.html. [Accessed:18-Sep-2018]

[4] Erina Ferro and Francesco Potorti, 'Bluetooth and Wifi Wireless Protocols: A Survey and A Comparison'. 2005. [Online]. Available: http://www.ece.stonybrook.edu/~xwang/public/readings/01404569.pdf. [Accessed:30-Sep-2018]

[5] S Madgwick. An efficient orientation filter for inertial and inertial/magnetic sensor arrays. Report x-io and University of Bristol (UK), 2010.

[6] Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Quaternions_and_spatial_rotation#Using_quaternion_rotations. [Accessed:30-Sep-2018]

[7] Ali, Stack Overflow, 2013. [Online]. Available: https://stackoverflow.com/questions/18252692/gravity-compensation-in-accelerometer-data [Accessed:30-Sep-2018]

[8] Tomaz Mencinger, Tennis Forehand Technique – 8 Steps To A Modern, 2017. [Online]. Available: https://www.feeltennis.net/modern-forehand-technique/ [Accessed:3-Oct-2018]

[9] Dr. David Dines and Dr. Joshua Dine, Common Wrist Injuries in Tennis Players, 2012 [Online]. Available: https://www.hss.edu/playbook/common-wrist-injuries-in-tennis-players/ [Accessed:3-Oct-2018]

[10] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine, in Ambient Assisted Living and Home Care, Springer, 2012, pp. 216-223

[11] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, Activity recognition using cell phone accelerometers, ACM SigKDD Explorations Newsletter, vol. 12, no. 2, pp.74-82, 2011.

[12]D. Karantonis, M. Narayanan, M. Mathie, N. Lovell, and B. Celler, "Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring," IEEE Trans. Inf. Technol. Biomed., vol. 10, no. 1, pp. 156–167, Jan. 2006

[13] Hammerla NY, Halloran S, Ploetz T. Deep, convolutional, and recurrent models

for human activity recognition using wearables. arXiv preprint arXiv:1604.08880
[14] F. I. Bashir, A. A. Khokhar, and D. Schonfeld, “Object trajectory-based activity classification and recognition using hidden markov models,”IEEE Trans. Image Process. , vol. 16, no. 7, pp. 1912-1919, Jul. 2007
[15] UCI, PAMAP2 Physical Activity Monitoring Data Set, 2012. [Online]. Available: http://archive.ics.uci.edu/ml/datasets/pamap2+physical+activity+monitoring. [Accessed:04-Oct-2018]