Project Title: Smart Tennis Racket

Authors

Arthur Salamin <atsal@kth.se>

Bruno Da Costa <brunoadc@kth.se>

Raul Aznar Alvarez <raulaa@kth.se>

Valeria Bladinieres Justo <vabj@kth.se>

Course

Sensor Based Systems

Supervisor

Mark T. Smith

Version

Final Technical Report

16th of March 2020

Table of Contents

Description of the System

The main goal of this project is to create a system capable of detecting where a tennis ball hits on a tennis racket and with its respective force and acceleration.

The system cannot interfere the user to play in a normal match or perform a training session. Thus, the power requirements need to be achieved as well as ‘wireless’ communication to transmit the data. In order to obtain the data of the location and force, 4 piezoresistive sensor have been located in the strings of the racket. They will sense the force in each part of the racket, and using an algorithm predict where the ball hit, and with which force. This will be displayed to the user in an understandable way, so that the player knows how to improve his swing.

Application of the System

This system can be very useful the tennis players. For the beginners it could be used to improve the ball placement on the racket for each hit. For more advanced players it could be used to analyse their hit efficiency, reliability and repeatability.

Architecture of the System

This sensor-based system is divided between hardware and software. The first three stages are in the hardware side and the next three are software. In Figure 1, it can be observed a block diagram of the system. First is the hardware, where the sensors, the printed circuit board and the arduino nano are found. Then the software process are, the data transmission from arduino to the computer, the machine learning algorithm and mapping of the processed data.

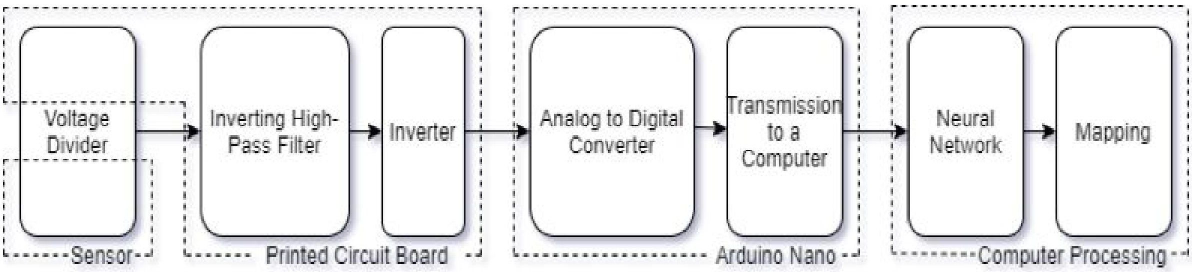


Figure 1: Block diagram sensor-based system.

Hardware

<u>Sensors</u>

The sensors used for this project were piezoresistive sensors. We used FSR 402 short tail sensors because they were already available and we could start the project immediately. We decided to position the four sensors at four different points on the racket to be able to triangulate more accurately the hit position, as it can be observed in Figure 2. These sensor are non-invasive since they could be mounted and removed from the racket without causing any damage.

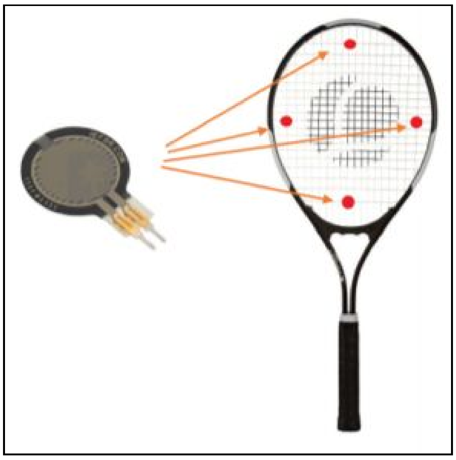


Figure 2: Position of Sensors

In order to measure the applied force on the sensor we used its conductance. These sensors were not made to measure the pressure, but were optimized to be used as "touch" sensors for user interface solution, because of this the force applied for a same response can vary quite a lot, apparently more than 20%. This is why we decide to use the conductance of the sensor to get an idea of the force of the hit and not derive the force directly from the resistance variation of the sensor. In Figure 3, it can be seen the relation between force, conductance and resistance.

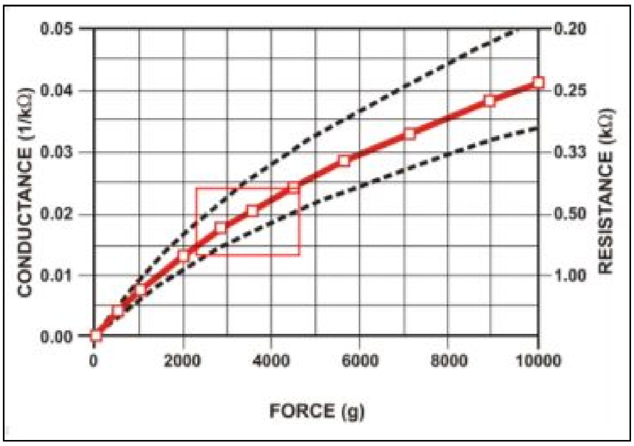


Figure 3: Relation between Force, Conductance and Resistance.

Circuit Board

The printed circuit board was made to process the data from the sensors, and amplified it by doing the hardware processing of the signal before sending it to the analog-digital converter. The circuit in Figure 4 was repeated four times, one for each sensor. The goal was to get rid of the offset introduced by the potential divider since we were only interested in the variation of the voltage created by the change of resistance of the sensor. To get rid of the offset we used a high-pass filter with gain of 366 to have the signal go from 0V to 5V on the output and cut-off frequency of 50Hz just to get rid of the offset.

Then we used a simple inverter because the arduino cards can only read positive voltage. We tried to do a non-inverting high-pass filter but for some reasons it didn't work as well so we kept this system with two Op Amps.

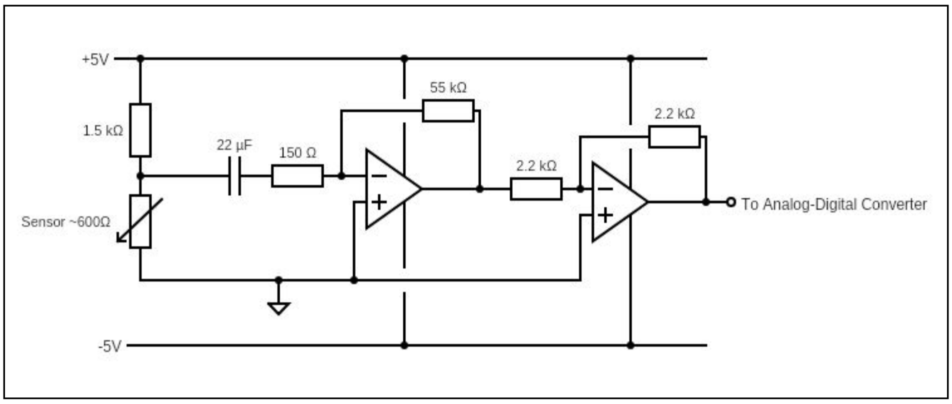


Figure 4: Hardware Schematic.

Power Supply

An important feature of our sensor based system, is a portable power supply, in order to guarantee the user to be able to play without any inconvenience. As the design of our system required a +5V and -5V supply. We thought about the usage of power bank batteries to keep the cost lower, if we go for a production model, the usage of 3.7V batteries with step-up boosters could be a solution to tackle this problem.

A problem faced for the prototype was that the power bank batteries have a minimal drawing current of around 70mA, therefore, it was spontaneously shutting down as it was not detecting a device connected to it as the current was really small. This we solved for the prototype connecting another powerbank to it, in order to have it with current circulation the whole time. However, that was a solution done for the prototype in order to not spend more money or time in that area and focus in the characterization.

The total power consumption of the system is around 60-70 mA. The PCB with the Op-Amps consume roughly 10mA, the Arduino Nano is also using the 5V supply and needs a current around 20 mA to operate, last the bluetooth module needs approximately 30 mA. The need for autonomy in our system was conditioning the power supply design. However, this has not been a problem for our system, a cheap Li-Po battery which can provide 600 mAh, would be more than enough to cover the need for around 3 hours. We take care of the conversion of the battery capacity.

$$5V\ Capacity = 3.7 * \frac{3.7V\ Capacity}{5} = 3.7 * \frac{600mAh}{5} = 444 \text{ mAh}$$

That would be the value without considering the current that the step up booster needs to take. Which should be between 10-30 mA, as well.

We could say that with a 600mAh - 3.7V battery and a step-up booster, we could use the product for 4 hours.

To provide the -5V battery we used a power bank battery and get the red cable(normally 5v to ground) creating then a dif voltage -5V in the black cable(usually ground).

Connectivity

To Transmit the data gotten from the sensor through the arduino we decided to use a bluetooth module. The model of the module used is a HC-06, which is bluetooth module that acts only as a slave since we did not need a master for our system. The module was connected to the microcontroller(arduino) and to the power supply.

Software

Connectivity

Python is the language where all the information was processed. To connect the bluetooth module to the software the libraries Serial, Matplotlib and Numpy were needed. First, the information was written in the serial port and read from it. Then, the bits were transformed to our reference of voltage and then sent to the machine learning algorithm.

Machine Learning Algorithm

For the Machine Learning Algorithm we decided to use a neural network since it was the system that we understood better, and knew that it would be capable of handling such a process. The Neural Network was build in python using the Keras library with the Tensorflow library as backend.The architecture of the neural network was a 4-dimensions input, 3 hidden layer, and a 13-dimensions output.

The hidden layer consisted of a first layer of 64 nodes with a 20% dropout to avoid overfitting, then a 32 nodes layer followed by an other 32 node layer. Since none of us had an extensive knowledge about neural networks we builded it by trying different combination of parameters, changing the architecture of the model, the epoch or the batch size to get the best result we could.

The first model was a continuous 2-dimensions output of X and Y values but this didn’t work as planned, when testing the hit point of the ball was placed mostly outside of the racket. Because of this we decided to use a discrete output model with

13 different areas on the racket, so the model would just have to guess in which area the ball hitted the racket.

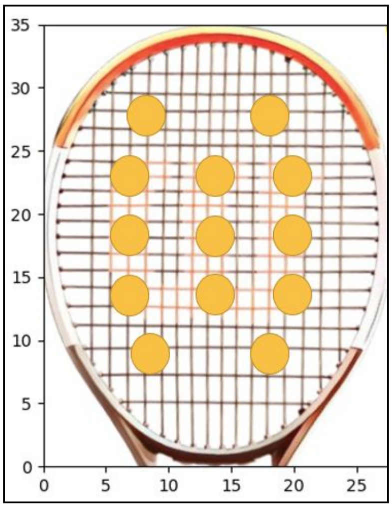
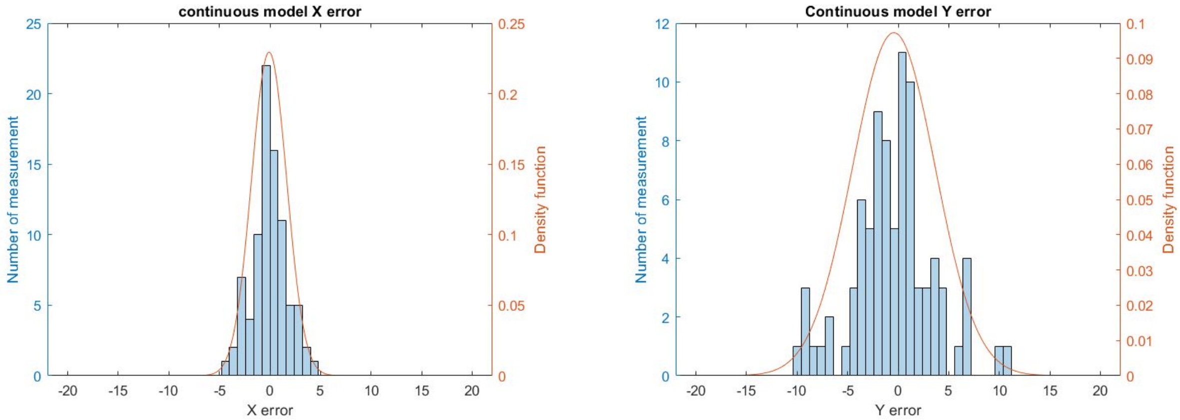


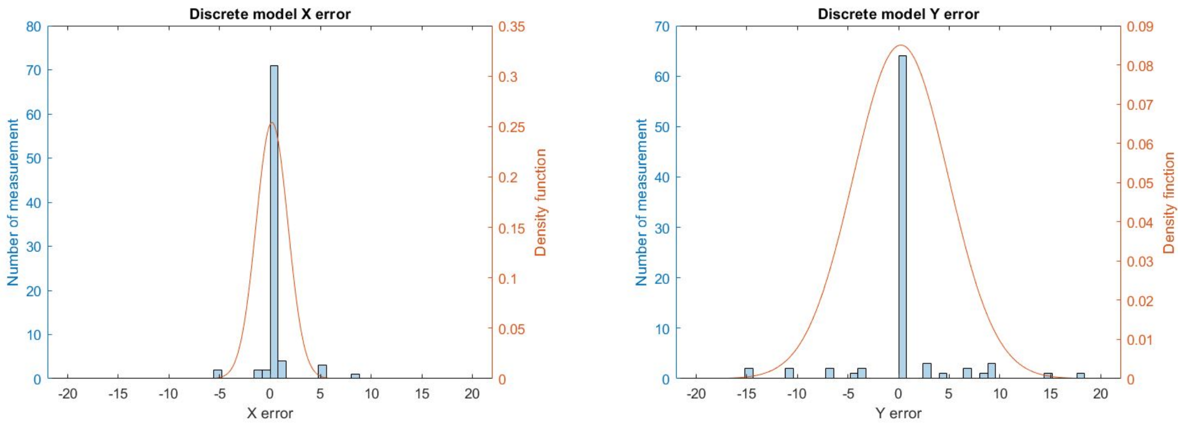
Figure 5: Discrete Positions of Racket.

When we look at the training results in Figures 6-10 of the neural network the density function for X and Y for the continuous model and the discrete model are quite similar. But the result when testing with the racket where very different. This is because the training result are made by splitting the data available in training data and testing data. Because of this the training data and testing data are very similar, and the testing gives really good results. But when used in reality the data coming from the sensor had a lot more variance that the one in testing. Because of a lot of different factors (sensors not optimized for this, the cables used were flimsy, the connectors where taped together and could still move a bit, etc.) the data coming from the sensor from two very similar hit on the same position could look very different. And the discrete model was a lot more resilient to these variation than the continuous model.

To get the data to train the model we just threw a ball at the racket and recorded the values given by the sensors and the position of the hit. The final training was made with 850 different data points taken over 2 different days.



Figures 6-7: Error in Continuous Model



Figures 8-9: Error in Discrete Model

Force Measurement

To measure the force we used the datasheet from the FSR-402 sensor. Using Figure 3 as reference of the conductance-force relation, we determined the formula F(N)=(C(μ℧)-1000)/30.

However, to measure the force on the impact we need to know first the force which the strings already had. And then in every hit determine the resistance of each FSR for that impact, allowing to measure the force of the impact by subtracting one to the other.

First step to obtain the data of the force and acceleration would be to insert the resistance of the sensor already allocated in the racket.

From these resistances we will obtain the conductances in micromhos. And from that conductance calculate the Force in Newtons with the formula previously mentioned.

$$C = \frac{10^6}{R}$$

$$F(N) = \frac{C - 1000}{30}$$

By doing that the force in which the sensor are in the racket while resting is obtained. Once the ball hits the racket, the Arduino receives from the ADC values from 0 to 1023, which are mapped from 0 to 5000 mV. In other words multiplying the value obtained from the Arduino by 4.887 will give us the value in mV.

This is the voltage in the output of the PCB, following we have to calculate inversely ΩΩthe gain, using the conversion with the resistance which are affecting the system.

$$V1.FSR = V1.Out/(56000/(((1500 * FSR)/(1500 + FSR)) + 150))$$

As we know the system has a supply of 5V, which goes to the resistance of 1.5 kΩ.

$$FSR.R1 = V1.FSR/((5000.0/1500) * (FSR/(FSR + 150)))$$

We calculate the current that goes into the system, and to calculate the resistance we calculate which part of the current that enters go to the FSR sensor and which one goes to the resistance at the Op.Amp. Dividing the Voltage in mV in the FSR, and the current that goes through it, using Ohm's Law we can calculate the resistance.

Once change in resistance is calculated, we subtract this value from the value in the resistance at the beginning. Thus we have the resistance in the system when the hit was applied.

$$FSR.R1 = FSR\ (Beginning) - FSR.R1\ (Change\ of\ resistance)$$

Then the first two formulas are applied again calculating the conductance and the force.

$$C' = \frac{10^6}{R}$$

$$F'(N) = \frac{C - 1000}{30}$$

This force is the Force which was applied at the Hit, but also contains the force which the sensor is already having from the racket strings. Subtracting the first force from the second we obtain the force from the impact.

$$F impact(N) = F'(N) - F original(N)$$

And in order to know the acceleration, assuming that is hit by a tennis ball which has a mass of 0.058 kg, we just need to divide the force between the mass.

$$Acceleration = \frac{Fim`pact(N)}{0.058}$$

Design Performance

Position

The mapping of the result was done by displaying an image of a tennis racket and displaying where the ball hitted with the guessed values of the machine learning algorithm. In Figure 10, the representation of the position of the ball in computer and real life.

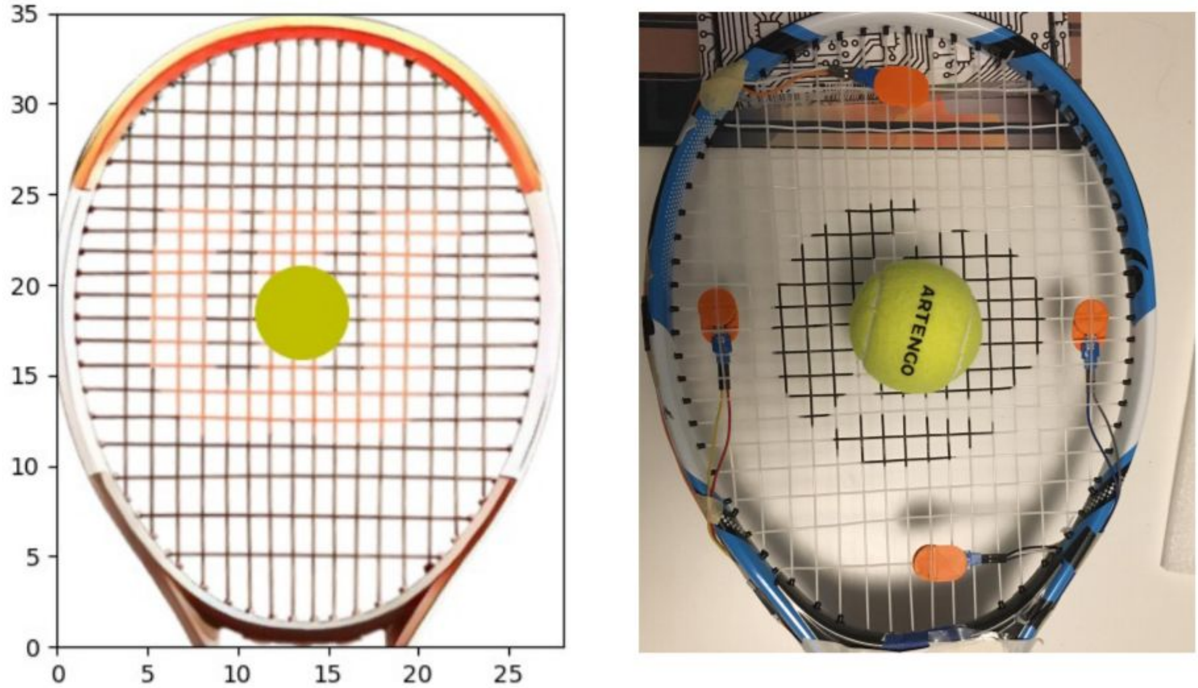
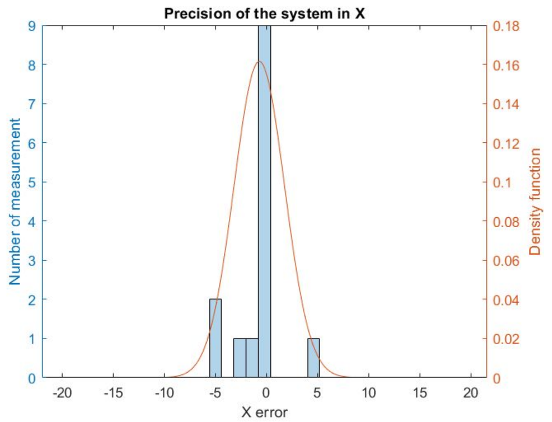
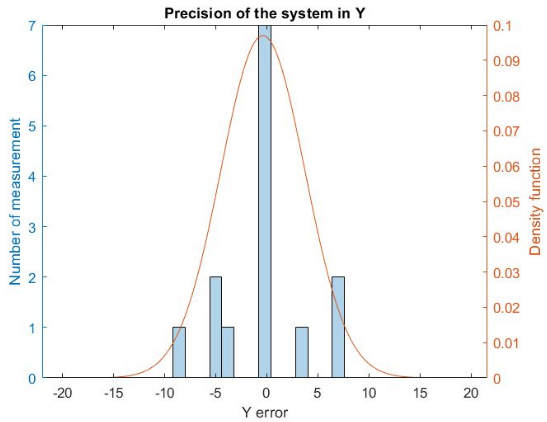


Figure 10: Ball Position Computer and Real Life.

The result we got were good, but unreliable. As we can see in Figure 11-12 the precision in X and Y, the system was able to guess the hit position with a quite good precision and accuracy. The mean value of the error for X and Y was around 0 and the standard deviation for X was 0.5 and the standard deviation for Y was 4.3 . Since a tennis ball has a diameter of 6.6 cm this is clearly a good enough precision. The problem is that these results were very difficult to get. The racket had to be set up in exactly the same way each time for it to work, and if a cable was moving too much or if one of the sensor was a bit misplaced all the result of the system would be very wrong. This is because the machine learning algorithm is very dependant on the homogeneity of the data. The data has to be taken in exactly the same way as the data used to train the model, otherwise it won't work. Since we took the data with the racket supported with two foam pads on a table, if someone was to use it while playing the results would not be very precise.





Figures 11-12: Precision of System.

Force and Acceleration

The performance of the force and acceleration, have not been tested, as it was very difficult to characterize this, The formula has been used according to the datasheet, and the number do make sense. However, this could be tested with more equipment. A machine throwing balls at a certain force to the racket to test the values and calibrate it would be one solution.

Work Distribution

Project Activities	Remarks	People
Brainstorming	-Choice of captor -Division of the work -Discussions about the skills area of each member	All group
Circuit Design	-Designing the circuit (high-pass, etc)	Bruno & Arthur
Shopping	-Purchases of different components (wires, captors, Arduino, etc)	All group

Testing and validation of the circuit	-Test of the circuit in the mentorspace (choice of gain/resistances)	All group
Design of the PCB	-Design of the 4 circuits for each captor	Raul & Valeria
Milling PCB	-Milling in the mentorspace	Raul & Valeria
Soldering PCB	-Soldering in the mentorspace	All group
Assembly on the racket	-Choice of batteries -Assembly of the captors, wires, PCB and Arduino on the Racket	Raul & Valeria
Software	-Coding for the Arduino and Python program to process the data	Raul & Valeria
Data measure	-Creating a dataset for the impact of the ball	All group
Machine Learning	-Creation of a model to process the Data and give the position of impact	Bruno & Arthur
Force and Acceleration	-Creation of an algorithm to calculate the force of impact	Raul & Valeria