LAB NOTEBOOK - PRIOSCALE

25/01/2017

This project aims of compare biological sensors and electronic sensors. We chose to study human hand sensors and a scale. We observed their reactions to pressure variation.

Human sensors characteristics:
According to our biological book Pearson, we have special mechanoreceptor in our skin. This mechanoreceptor (Merkel cells) send to the brain information through axons. This information is interpreted by the brain as a pressure.

Electronical sensors characteristics:
Tanguy Chotel lend a lead scale to us. This sensor is used for mass indication. A pressure will change the strength of the resistor. This resistor change the voltage of the circuit. Connected to an arduino, we can read results which is in gram unit.

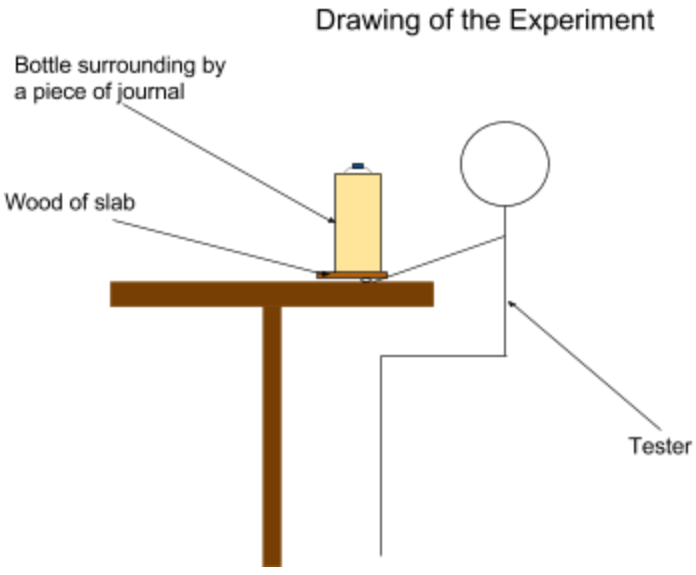
Preliminary experiment : to test our protocol in real condition and improve it.

hod

We prepared 7 bottles. We fill them with a certain amount of water inside corresponding to a known mass. Thus, we chose : 0g, 300g, 500g, 800g, 1000g, 1300g, and 1500g. Then, we used a wood of slab to balance and flatten the hand of the volunteer. To avoid bias because of vision, we blindfolded the volunteer. The bottle were covered with paper to reduce the noise of water when we moved them.

First, we put nothing on the wood of slab to know if the volunteer can feel the difference between no-bottle and bottle. Thus, when he answered about what he felt, we put 1500g on the slab. We asked him, if he felt something. This part is about testing the presence of active sensors.

Secondly, we put in a random order the different bottles on the slab. Between each weight, we calibrated the volunteer with a reference bottle. This bottle is filled with 1000g. Thus, we let him know the mass of this bottle to avoid that with time he lost the notion of weight.



We chose 7 identical bottles. To fill them, we chose three others bottles. This particular bottles were filled with 50 mL pipette. We put 300 mL, 500mL and 1000 mL in them. Then we draw a line where the water end. This was our reference to fill the other bottles.

With this test, we can know the time required to make the experiment. It is around 4 minutes. Moreover, comments from our tester was crucial : he informed us of the effort required to maintain the bottle in balance. In fact, he used his muscles to maintain the bottle and according to the force he required to do it, he estimated the weight. Thus, he based his judgement on the difficulty of maintain the bottle in balance and not on the pressure he felt on his hand. To correct this point, we put a piece of piece of tissue below his hand. In this way, his hand was stable enough to support the bottle without dealing with the balance of the bottle.

Concerning the arduino device, we struggled with the load cell arduino that gave us Tanguy. Here is a picture of the load cell :

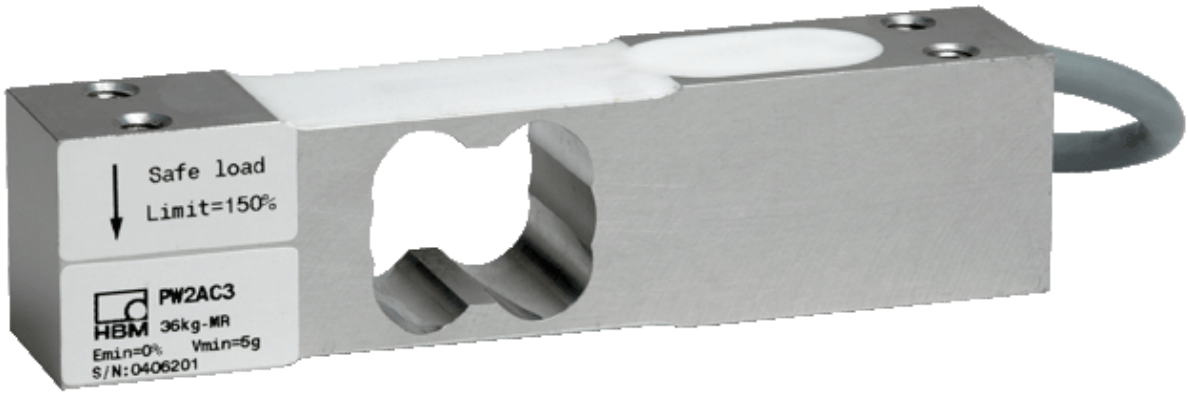
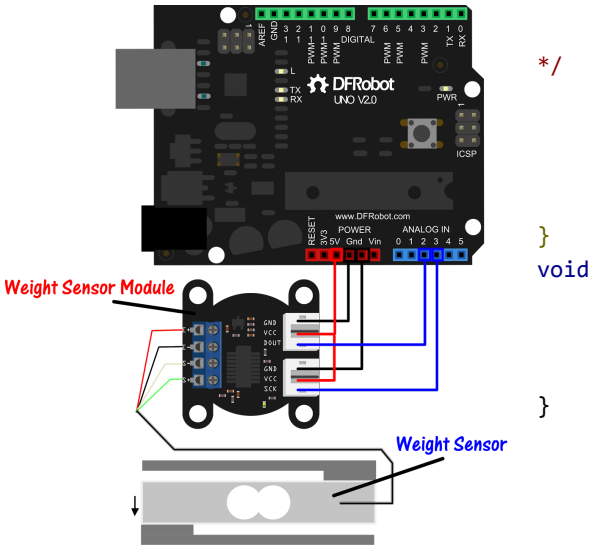


Figure : picture of a load cell (from wikipedia.org)

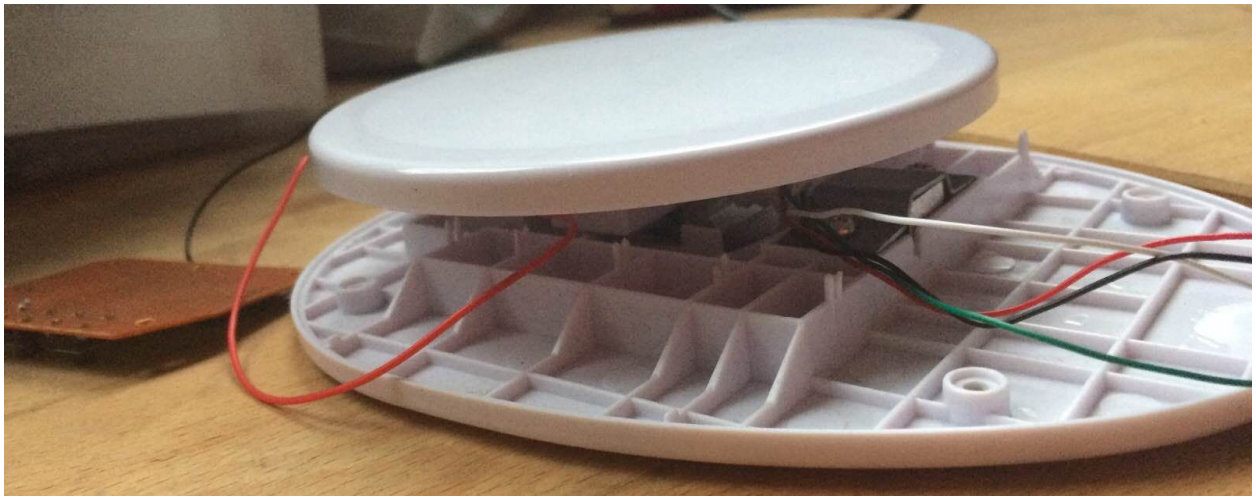
Here is the code and the device we found on the internet :
From : https://www.dfrobot.com/wiki/index.php/Weight_Sensor_Module_V1

```
/*
 * description:
 * The sample code for digital weight scale of hx711. The weight will be printed in the
serial.
 * library host on
https://github.com/aguegu/ardulibs/tree/3cdb78f3727d9682f7fd22156604fc1e4edd75d1/hx711
 * connection:
Hx711.DOUT - pin #A2
Hx711.SCK  - pin #A3                                                              */

#include <Hx711.h>
Hx711 scale(A2, A3);
void setup() {
Serial.begin(9600);                                                               }

loop() {                                                                          void
Serial.print(scale.getGram(), 1);
Serial.println(" g");
delay(200);                                                                       }
```



However the output doesn't work. We checked several times the device but it prints only "-8314 g" and sometimes "0 g".
Because of the time we decided to change the arduino device and we decided to take another scale lend by Tanguy.

Here is a picture of the new scale :



The device is exactly the same than with the load cell.
But the code still didn't work, but with a little of improve ! There is sometimes variation when we put an object on the top, but it still very weird value such as "-514" to "40".

Tomorrow we will try another sensor called FSR (Force sensitive resistor). It looks like this :

From Wikipedia.org

26/01/2017

Today, we decided to do a repetition of our experiment. We passed **two time each bottle** which is equal to one repetition.
In this way, we did another test. We measured the time required to practice the experiment. It is around 10 minutes. Moreover, we prepared a script to say to each volunteer before the beginning of the experiment.

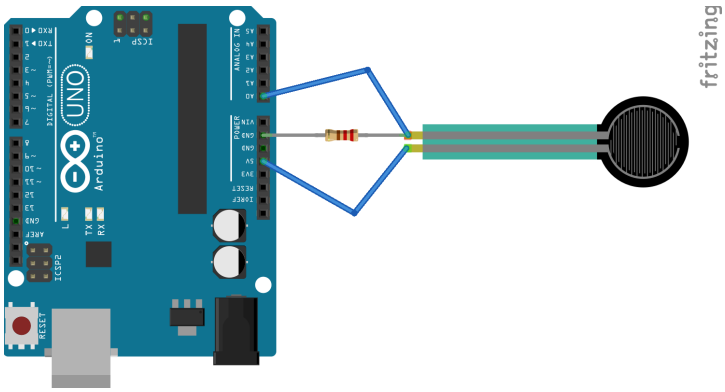
Then, we tested our different roles in the experiment to be efficient with the volunteer. One is focused on write the data which is the response of the volunteer. An other one is focused on measuring the response time which is the time required to obtain the estimation of the volunteer. Moreover, he said the script. Finally, the last one take care of put the bottle on the slab.

We began our experiment in the afternoon (we got three participants).
We had some difficulties with the empty bottle, sometimes the participant ask if there were something on the wood and we had to say yes.
however, our teacher ask us to stop it by e-mail. He shouldn't begin the experiment before tomorrow afternoon.

We didn't begin the third arduino device.

27/01/2017

In the morning, we finished the arduino device.



The code was simple to find on the internet :

```
//From the article: http://bildr.org/2012/11/force-sensitive-resistor-arduino

int FSR_Pin = A0; //analog pin 0

void setup(){
  Serial.begin(9600);
}

void loop(){
  int FSRReading = analogRead(FSR_Pin);
```

```
  Serial.println(FSRReading);
  delay(100); //just here to slow down the output for easier reading
}
```

We have to start the experiments at the same time than the other group. We also have to reconsider the number of replicate and repetition.

Today, we talk about repetitions and replicats. Our experiment, which has just one repetition, was not sufficient to produce viable data. Biological noise was too high when we work with human. The state of mind, light or other random bias could interfere with our result. We need another replicate to obtain a better idea of human hand sensors sensibility to pressure variation. Thus, we tried to find some volunteers that could come the next day, to repeat the experiment. In this way, we could have a replicate of our experiment. However, we afraid to have some people that can't repeat the experiment. Moreover, to reduce the time required to finished the experiment, we removed 2 weights :

We already thought about how we will plot our data. We think to make three different graphs :

- Accuracy

Abs : absolute weight
Ordonnée : Given weight

- Precision

Abs : absolute weight
Ordo : variance

- Time response

Abs : absolute weight
Ordo : second.

02/02/2017

Today we went to montparnasse to fulfill our experiment in the afternoon. We choose the "teacher room" at montparnasse. It was quiet but quite hot (we didn't measure the temperature unfortunately).
Next time, we should try to make more bibliographical research on "what affect human mechanosensor ?"
We had the time to ask 7 differents humans to make our experiment.
Daphné is saying the script and record the time.
Léonie is putting the weights on the hand and announcing the tare.

Clément is recording the estimation.
We always kept this configuration for the day.

For the 4th participant we didn't put well the bottle on the hand, so it felt. So we put again the tare and the bottle to estimate.

Something else that could make a difference : the time we let the tare was to the appreciation of the participant. We wait to his or her signal to put the new weight.

Concerning the arduino device, we started to finished the python code.
There is two different scripts :
One to get the information for the arduino and put it in a .txt file.
One to interprete the .txt file to get the .csv file with the mass, the output and the time response.

Here are the codes :

1. Arduino to txt

```
#from
http://www.instructables.com/id/Using-an-Arduino-and-Python-to-plotsave-data/step3/Python-code
/
#http://stackoverflow.com/questions/24214643/python-to-automatically-select-serial-ports-for-a
rduino
import serial
import matplotlib.pyplot as plt
import numpy as np
import warnings
import serial.tools.list_ports

arduino_ports = [
        p.device
        for p in serial.tools.list_ports.comports()
        if 'Arduino' in p.description
]
if not arduino_ports:
        raise IOError("No Arduino found")
if len(arduino_ports) > 1:
        warnings.warn('Multiple Arduinos found - using the first')

ser = serial.Serial(arduino_ports[0])
#ser = serial.Serial(comPort, 9600) #sets up serial connection (make sure baud rate is correct
- matches Arduino)
connected = False
while not connected:
        serin = ser.read()
        connected = True
plt.ion()                       #sets plot to animation mode
length = 300                    #determines length of data taking session (in data points)
```

```
x = [0]*length               #create empty variable of length of test
#y = [0]*length
xline, = plt.plot(x)         #sets up future lines to be modified
#yline, = plt.plot(y)
plt.ylim(0,1500)      #sets the y axis limits

time = 0
for i in range(length):      #while you are taking data
       data = ser.readline() #reads until it gets a carriage return. MAKE SURE THERE IS A
CARRIAGE RETURN OR IT READS FOREVER
       sep = data.split()    #splits string into a list at the tabs
       print sep
       if len(sep) != 0:
        x.append(int(sep[0]))
        #y.append(int(time))
        time += 200
        del x[0]
        #del y[0]
        xline.set_xdata(np.arange(len(x))) #sets xdata to new list length
        #yline.set_xdata(np.arange(len(y)))
        xline.set_ydata(x)                 #sets ydata to new list
        #yline.set_ydata(y)
        plt.pause(0.001)                   #in seconds
        plt.draw()                         #draws new plot
np.savetxt("data25", x) # change the name each time.
#row_arr = np.array(rows)           #creates array from list
#np.savetxt("result", row_arr) #save data in file (load w/np.loadtxt())

ser.close() #closes serial connection (very important to do this! if you have an error partway
through the code, type this into the cmd line to close the connection)
```

2. Txt to csv

```
bottle = [i for i in range(1, 26)]
match = {4 : 300, 11: 300, 13: 300, 17: 300, 20: 300, 7 : 500, 14 : 500, 22: 500, 23: 500, 24:
500, 1 : 800, 12: 800, 16: 800, 18: 800, 19: 800, 2: 1000, 5: 1000, 9: 1000, 10: 1000, 15 :
1000, 3: 1300, 6 : 1300, 8 : 1300, 21 : 1300, 25 : 1300}

#arduino
def Extract(nom): # this function is able to read the txt and give the response time and mean.
    a = open(nom, "r") #open the txt file
    ResponseTime = 0
    maxi = 0
    courbe = []
    u = 0

    for i in a: #create a list with all the data
        courbe.append(float(i))
    output = []
    tic = 0
```

```
    toc = 0

    means = []
    ResponseTimes = []
    for i in range(len(courbe)-4):
        if courbe[i] > 30: # if the arduino felt a mass
                ResponseTime+=100
                if (-courbe[i] + courbe[i+1] + courbe[i+2] - courbe[i-1]) < 5: #check if the
value is stabilized.
                        tic += 1
                        if tic == 5:
                                ResponseTimes.append(ResponseTime - 500)
                        output.append(courbe[i])
        Else: #when it is a new individu
                if len(output[2:-2]) != 0:
                        u += 1
                        mean = sum(output[2:-2])/len(output[2:-2])     #process to give the
output (we take a mean of the variation.
                        means.append(mean)
                        output = []
                        toc = 1
                        print("------")
                if toc == 1:
                        ResponseTime = 0
                        tic = 0
                        toc = 0
    a = open(nom + ".csv", "w")
    for i in range(len(means)):
        #print(str(means[i]) + "," + str(ResponseTimes[i]))
        a.write(str(means[i]) + "," + str(ResponseTimes[i]) + "\n")
    print(nom)
    print(u)
    return means, ResponseTimes

a = open("arduino.csv", "w")
for bot in bottle:
    try:
        mean, ResponseTime = Extract("data" + str(bot))
    except:
        print("error")
    weights = [match[bot] for i in range(len(mean))]
    toplot = (mean[1] * 1000 )/ mean[0]
    #plt.plot(match[bot], toplot, "o", color="green")
    a.write(str(bot) + "," + str(ResponseTime[1]) + "," + str(toplot) + "\n")

a.close()
```

Advantage of the code : It is able to take in account the little fluctuation of the arduino device and is able to give an objective approximation of the weights felt by the arduino.

03/02/2017

This afternoon we began again the experiment.
We tried to make some replicates, so we try to find again the same people.
In total we make :
12 different people and 5 people came twice !

The kept the same configuration than yesterday :
Daphné is saying the script and record the time.
Léonie is putting the weights on the hand and announcing the tare.
Clément is recording the estimation.

Except one time with the participant 1.6 because Léonie wasn't here.
Daphné is saying the script and record the time and recording the estimation.
Clément was putting the weights on the hand and announcing the tare.

Once we take too long to find the order of the bottle to put on the hand, so the participant ask to feel a second time the tare. Otherwise everything went well.

04/02/2017

Today we made the fourth replicate for the arduino.
In order to be sure that the arduino can feel the bottle (has its diameter higher than the surface of the FSR sensor), we build a little wood device.

is

To measure we took the following configuration (each number is a new replicate):

1. Clément put the bottle and Léonie record on the computer
2. Clément put the bottle and Daphné record on the computer
3. Clément put the bottle and Daphné record on the computer
4. Daphné put the bottle and Clément record on the computer

After processing all the data, we get this results (you can find the code at the very end of the labnotbook)

Here are our conclusions but the sample size isn't large enough to make statement:

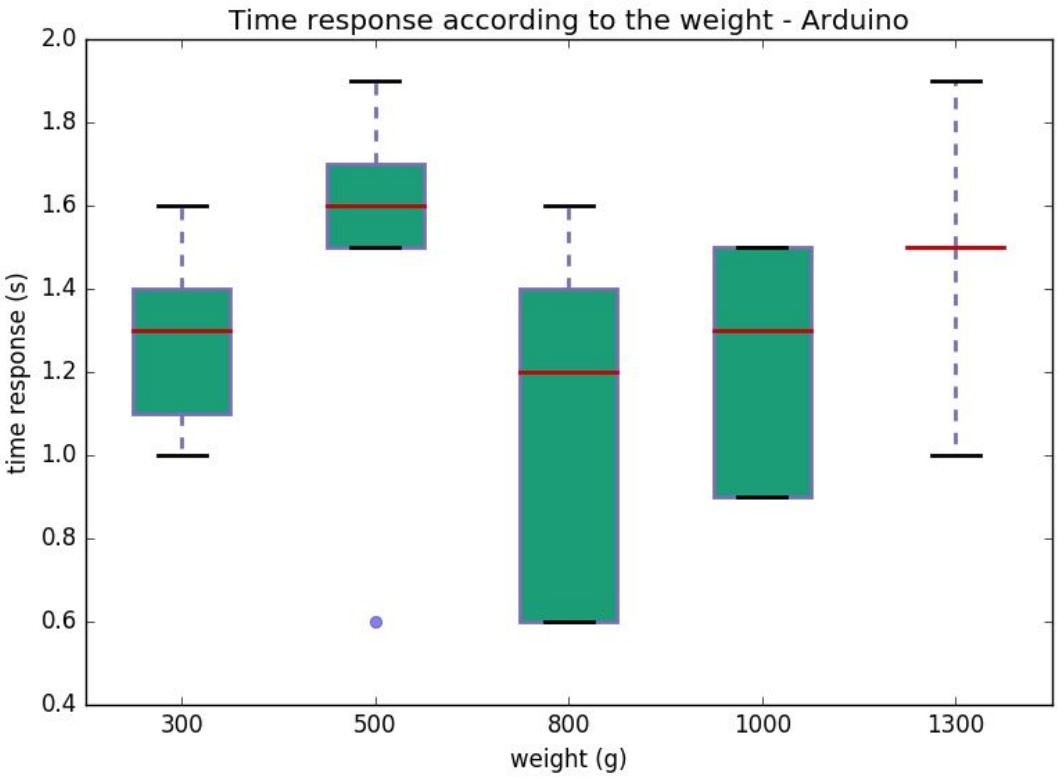
- The median for a human to estimate a weight is around 5s and is not in function of the mass
- There is a biggest variation on for the arduino device than for the human
- The FSR seems to saturate very quickly, giving between 700 and 1000 output even if the mass double.

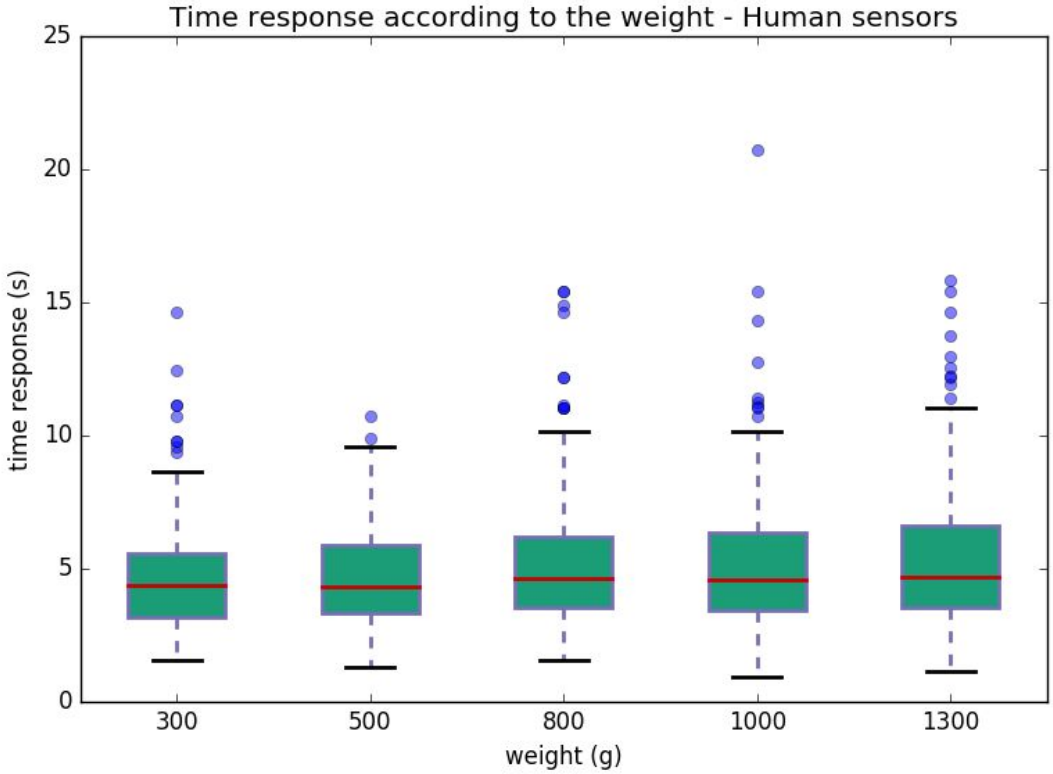
Limits of our experiment :

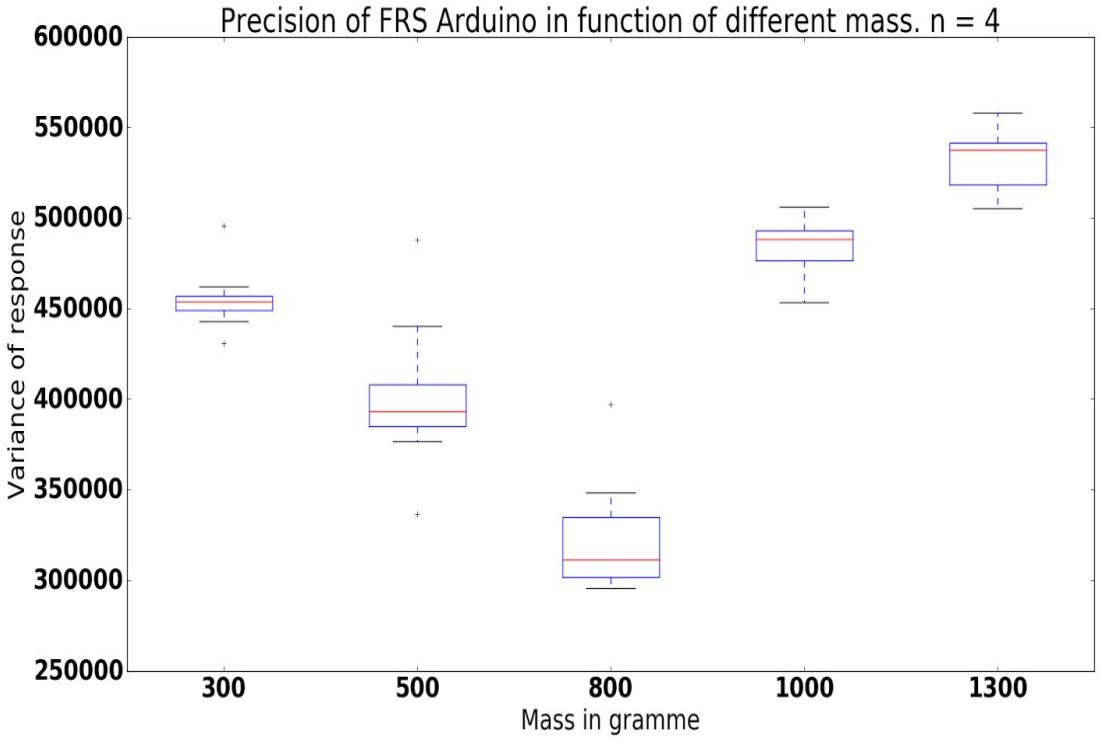
- Position of the bottle on the wood slab
- Shape and the position of the hand of the participant
- Individuality and mood of human.

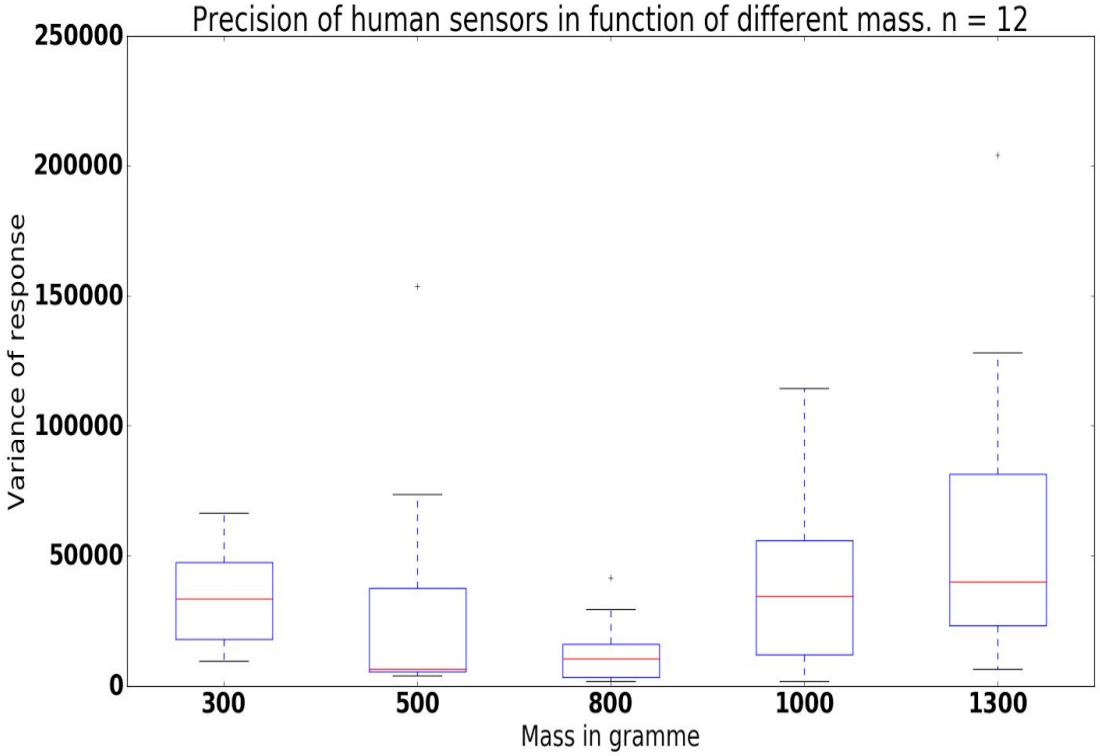
Further experiments to do :

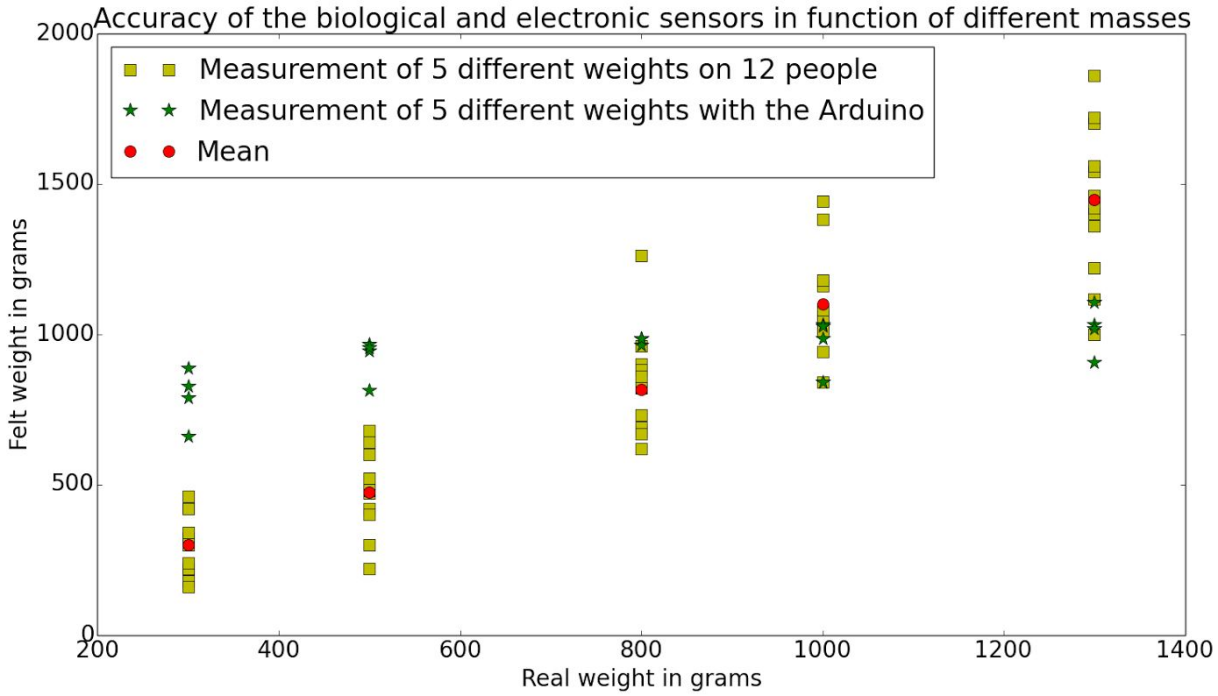
- More replicate for humans and arduino to reduce the noise.


Time response according to the weight - Arduino
time response (s)
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
300
500
800
1000
1300
weight (g)


Time response according to the weight - Human sensors
time response (s)
25
20
15
10
5
0
300
500
800
1000
1300
weight (g)


Precision of FRS Arduino in function of different mass. n = 4
600000
550000
500000
450000
400000
350000
300000
250000
Variance of response
300
500
800
1000
1300
Mass in gramme


Precision of human sensors in function of different mass. n = 12
250000
200000
150000
100000
50000
0
Variance of response
300
500
800
1000
1300
Mass in gramme



Code for the precision graph :

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

import matplotlib.pyplot as plt
import pylab

match = {4 : 300, 11: 300, 13: 300, 17: 300, 20: 300, 7 : 500, 14 : 500, 22: 500, 23: 500, 24:
500, 1 : 800, 12: 800, 16: 800, 18: 800, 19: 800, 2: 1000, 5: 1000, 9: 1000, 10: 1000, 15 :
1000, 3: 1300, 6 : 1300, 8 : 1300, 21 : 1300, 25 : 1300}

font = {'weight' : 'bold',
        'size'   : 25}
plt.rc('font', **font)

plt.ylabel('Variance of response')
plt.xlabel('Mass in gramme')
plt.title("Precision of human sensors in function of different mass. n = 12")
#human
a = open("human.csv", "r")
human = []
for i in a:
    human.append(i.replace("\n", "").split(","))

number = 1
means = {300 : [], 500 : [], 800 : [], 1000 : [], 1300 : []}
```

```
tovariances = {300 : [], 500 : [], 800 : [], 1000 : [], 1300 : []}
variances = []
masses = []
for hu in human:
    if str(number) == hu[0]:
        means[match[int(hu[1])]] += [int(hu[2])]
    else:
        for mass in means:
               mean = sum(means[mass]) / 5
               var = 0
               for value in means[mass]:
                      var += (value - mean)**2
               var = (1./float(len(means[mass]))) * var
               if "." in hu[0]:
                      #plt.plot(mass, var, "*", markersize=15)
                      hello = "coucou"
               else: #plt.plot(mass, var,"v", markersize=15)
                      tovariances[mass] += [var]
        means = {300 : [], 500 : [], 800 : [], 1000 : [], 1300 : []}
        number = hu[0]
        means[match[int(hu[1])]] += [int(hu[2])]
for val in tovariances:
    variances.append(tovariances[val])

print(variances)


box_leg = ['300', '500', '800', '1000', '1300']
#plt.boxplot(variances)
pylab.xticks([1,2,3,4,5], box_leg)

plt.show()
```

Code pour le response time :

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

import numpy as np
import matplotlib.pyplot as plt
import csv
import pylab

speed_arduino = [[] for i in range(5)]
speed_arduinos = [[] for i in range(5)]


with open('arduino.csv', 'rt') as csvfile:
       reader = csv.reader(csvfile)
    for row in reader:
```

```
path = row[0]
if '4' in path:
speed_arduino[0].append(float(row[1]))
if '11' in path:
speed_arduino[0].append(float(row[1]))
if '13' in path:
speed_arduino[0].append(float(row[1]))
if '17' in path:
speed_arduino[0].append(float(row[1]))
if '11' in path:
speed_arduino[0].append(float(row[1]))
if '20' in path:
speed_arduino[0].append(float(row[1]))
if '7' in path:
speed_arduino[1].append(float(row[1]))
if '14' in path:
speed_arduino[1].append(float(row[1]))
if '22' in path:
speed_arduino[1].append(float(row[1]))
if '23' in path:
speed_arduino[1].append(float(row[1]))
if '24' in path:
speed_arduino[1].append(float(row[1]))
if '1' in path:
speed_arduino[2].append(float(row[1]))
if '12' in path:
speed_arduino[2].append(float(row[1]))
if '16' in path:
speed_arduino[2].append(float(row[1]))
if '18' in path:
speed_arduino[2].append(float(row[1]))
if '19' in path:
speed_arduino[2].append(float(row[1]))
if '2' in path:
speed_arduino[3].append(float(row[1]))
if '5' in path:
speed_arduino[3].append(float(row[1]))
if '9' in path:
speed_arduino[3].append(float(row[1]))
if '10' in path:
speed_arduino[3].append(float(row[1]))
if '15' in path:
speed_arduino[3].append(float(row[1]))
if '3' in path:
speed_arduino[4].append(float(row[1]))
if '6' in path:
speed_arduino[4].append(float(row[1]))
if '8' in path:
speed_arduino[4].append(float(row[1]))
if '21' in path:
```

```
                speed_arduino[4].append(float(row[1]))
                if '25' in path:
                speed_arduino[4].append(float(row[1]))
print(speed_arduino)
for temps in range(len(speed_arduino)):
        speed_arduinos[0].append(int(speed_arduino[temps][0])*10**-3)
        speed_arduinos[1].append(int(speed_arduino[temps][1])*10**-3)
        speed_arduinos[2].append(int(speed_arduino[temps][2])*10**-3)
        speed_arduinos[3].append(int(speed_arduino[temps][3])*10**-3)
        speed_arduinos[4].append(int(speed_arduino[temps][4])*10**-3)
print(speed_arduinos)

fig = plt.figure(1, figsize=(9,6))
ax = fig.add_subplot(111)
bpo = ax.boxplot(speed_arduinos, patch_artist=True)

for box in bpo['boxes']:
        box.set( color='#7570b3', linewidth=2)
        box.set( facecolor = '#1b9e77' )
for whisker in bpo['whiskers']:
        whisker.set(color='#7570b3', linewidth=2)
for cap in bpo['caps']:
        cap.set(color='#000000', linewidth=2)
for median in bpo['medians']:
        median.set(color='#CC0000', linewidth=2)
for flier in bpo['fliers']:
        flier.set(marker='o', color='#e7298a', alpha=0.5)

BoxName = ['300', '500', '800', '1000', '1300']
pylab.xticks([1,2,3,4,5],BoxName)
plt.xlabel('weight (g)')
plt.ylabel('time response (s)')
plt.title('Time response according to the weight - Arduino')
plt.savefig('timeresponses_arduino.png')
plt.show()
```