

# Réalisation et étude d'un accordeur de guitare

---

# INTRODUCTION

---



# PLAN

---

## **I] L'accordeur de guitare:**

- 1) Principes d'utilisation
- 2) Fonctionnement d'un accordeur classique

## **II] Acquisition et traitement du son de la corde:**

- 1) Montage et composant
- 2) FFT et TFD

## **III] Correction de fréquence**

## **IV] Annexe**

# 1) Principes d'utilisation



## Accordeurs sans câble:

- Utilisable sur tout les modèles
- Détecte tout les sons
- Plus facile de compréhension



## Accordeurs avec câble:

- Très précis
- Complexe
- Uniquement sur électrique

## 2) Fonctionnement d'un accordeur classique

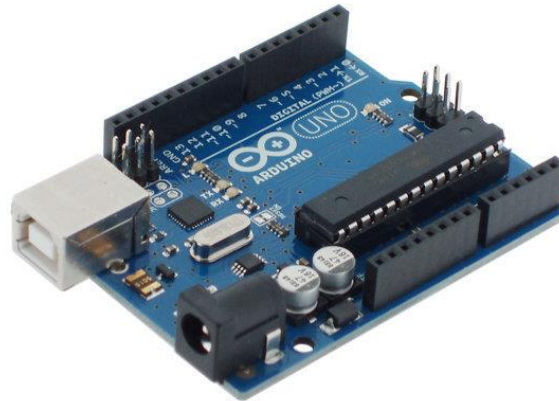


- Acquisition du son
- Traitement du son pour obtenir la fréquence
- Comparaison avec une fréquence de référence
- Indiquer si la fréquence est trop haute ou trop basse par le biais d'une information visuelle

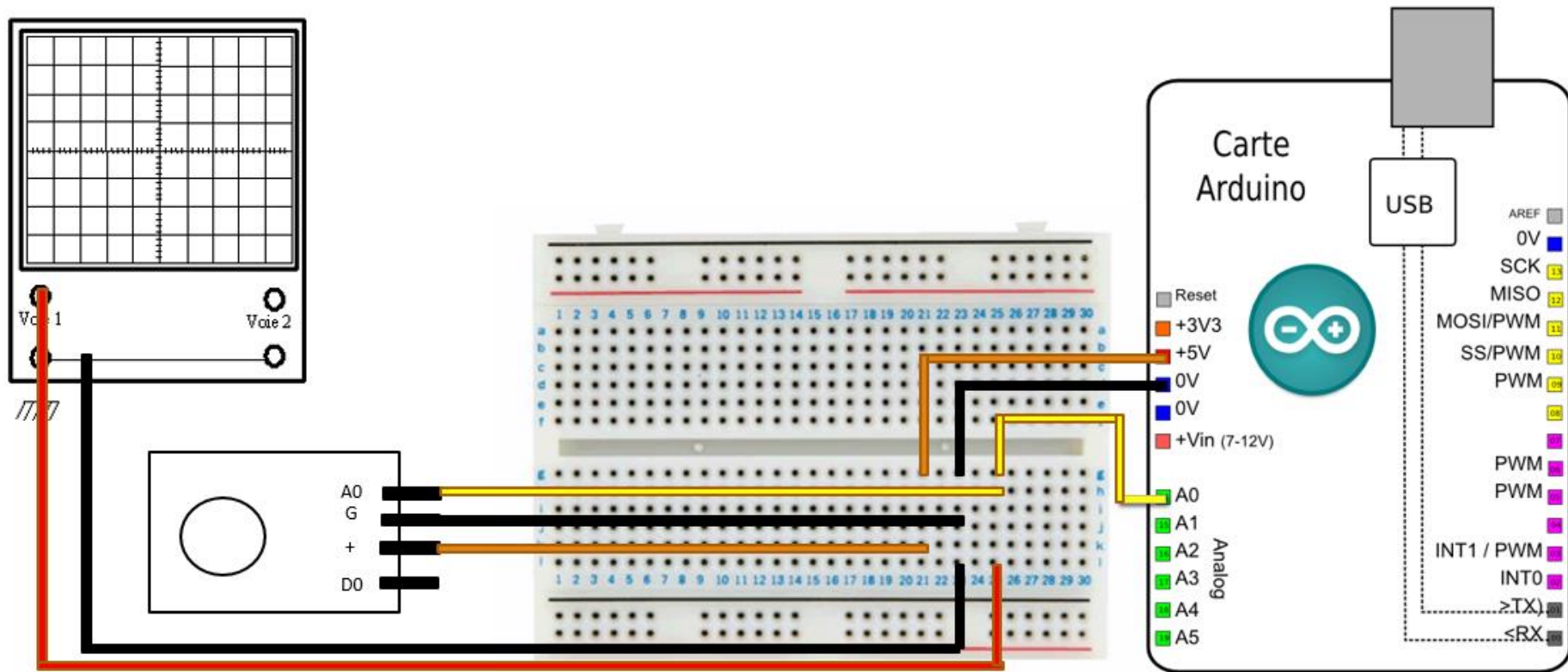
### Accordeur automatique:

- Calculer la correction à effectuer
- Commander des moteurs pour effectuer cette correction
- Recommencer jusqu'à placement dans un intervalle de fréquence voulu

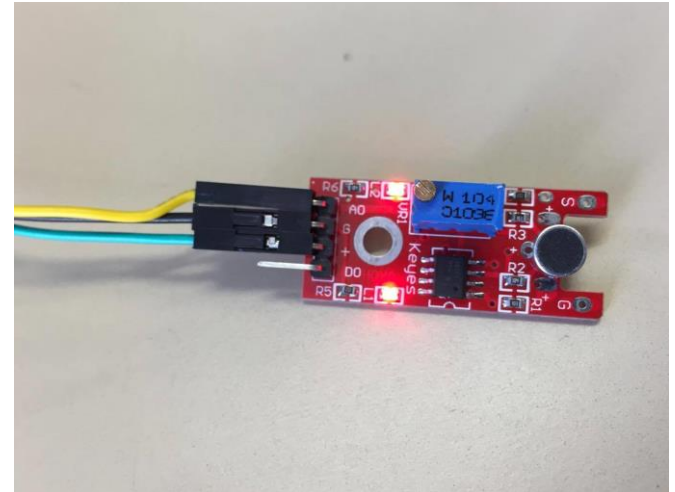
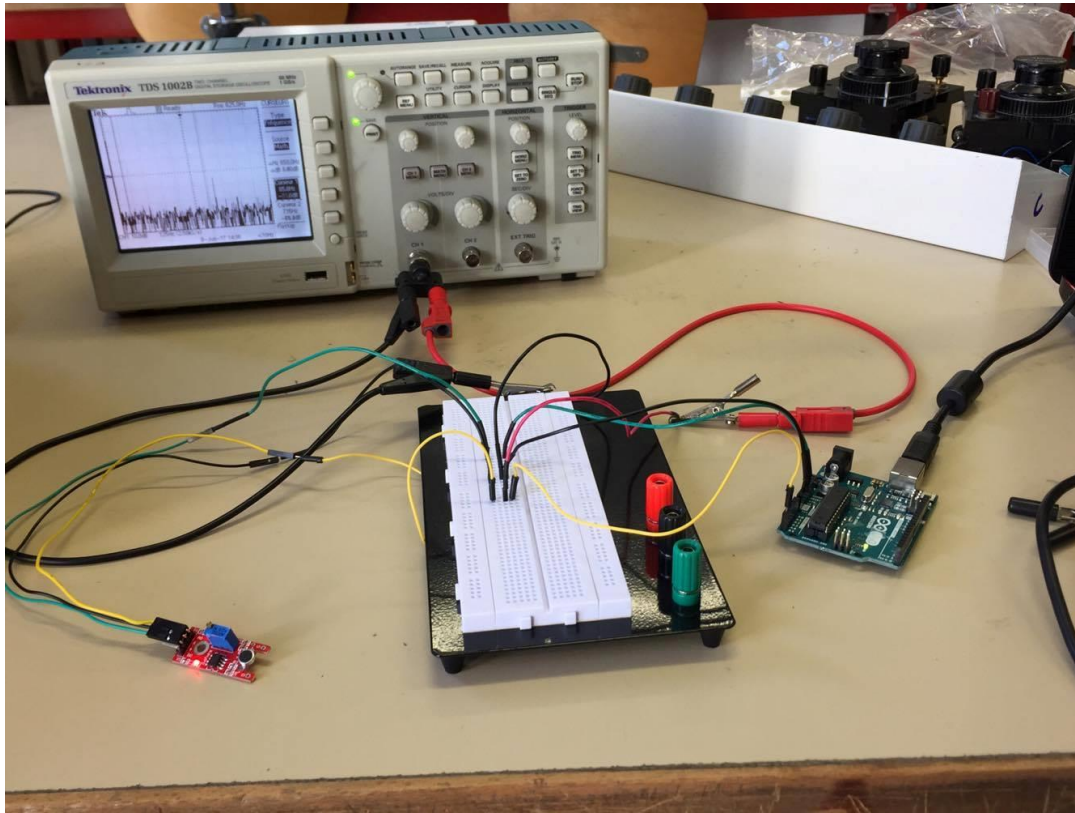
## II] Acquisition et traitement du son de la corde:



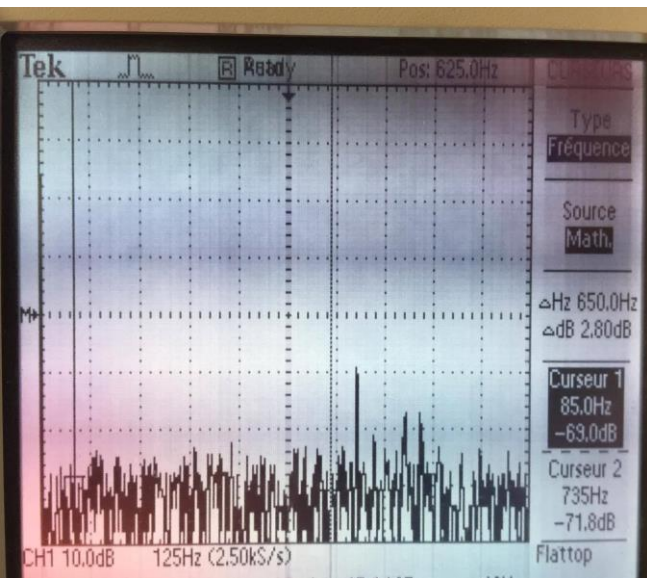
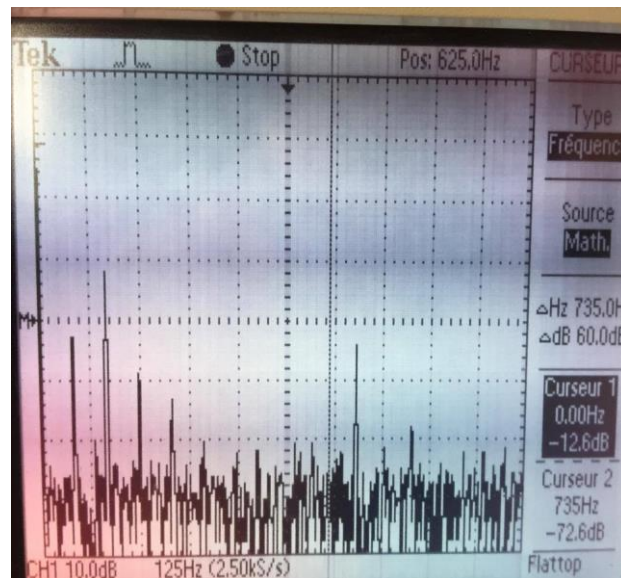
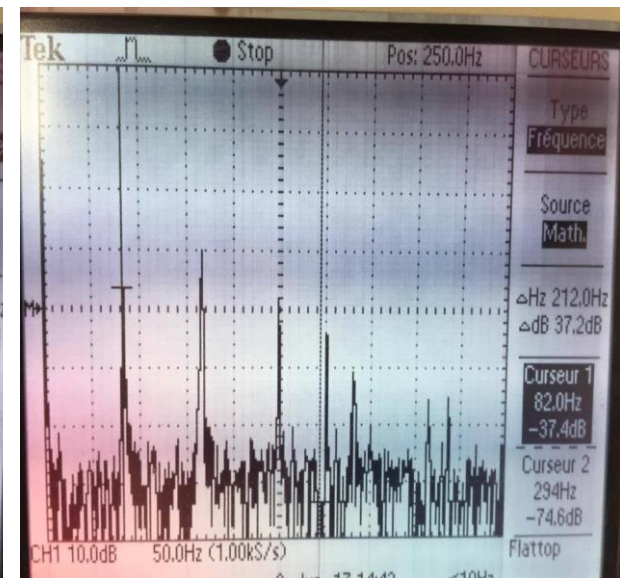
# 1) Montage et composants





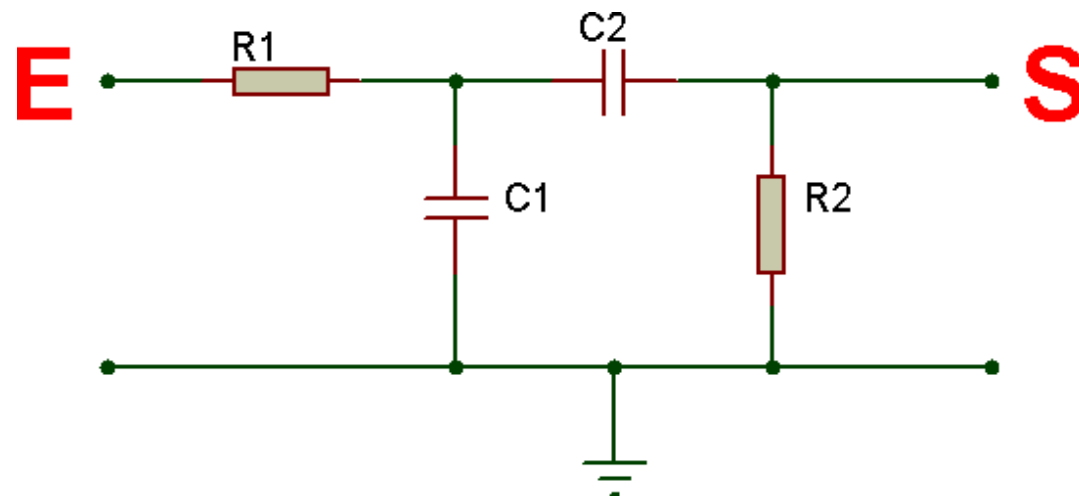




Aucun signalPremière vibration de la corde  
Mi graveDeuxième vibration de la corde  
Mi grave

# Filtrage

Passe bande passif :



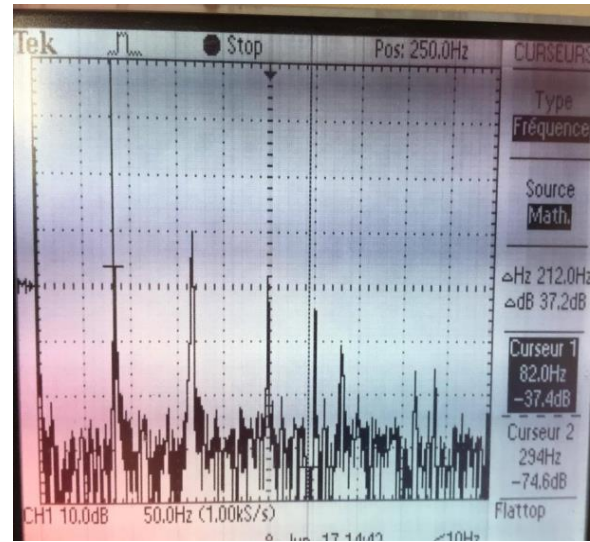
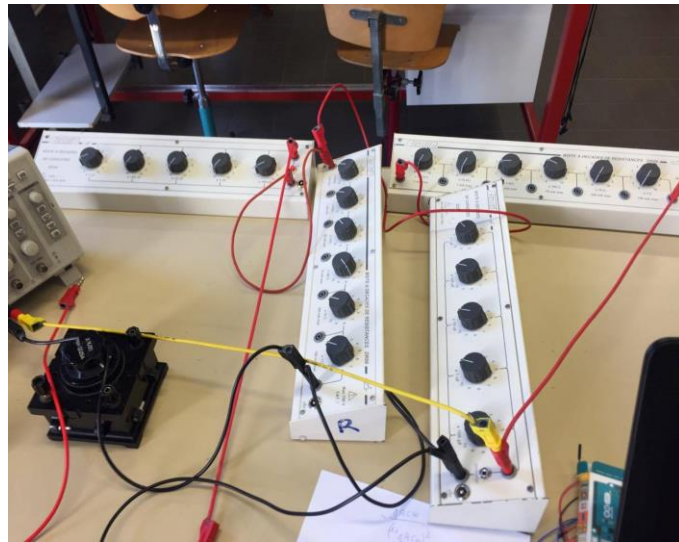
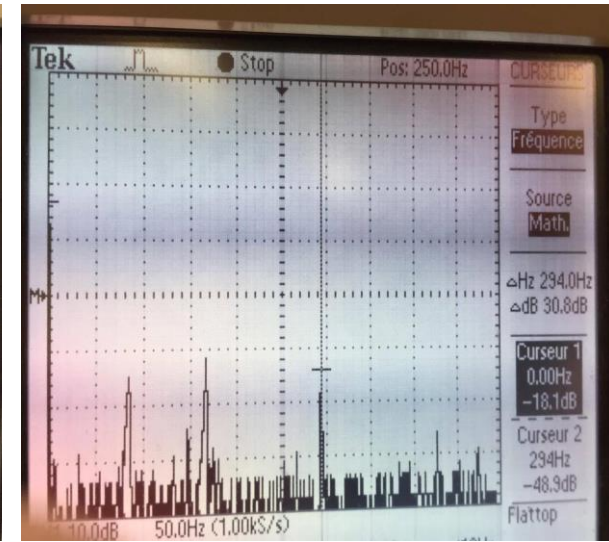
Fréquences de coupure :

$$F_{cB} = \frac{1}{2\pi R_1 C_1}$$

$$F_{cH} = \frac{1}{2\pi R_2 C_2}$$

$$C_1 = C_2 = 1\mu F \text{ et } R_1 = 1700\Omega, R_2 = 2300\Omega$$

$$F_{cB} = 69Hz \quad F_{cH} = 93Hz$$

Avant filtrageAprès filtrage

# Variation de fréquence lors de l'accordage:

Précision de l'appareil :  $\pm 0,5\text{Hz}$

Tour:	-1	-3/4	-1/2	-1/4	0	1/4	1/2	3/4	1
Fréquence	74	76	77,5	79,5	81,5	83	85	87	89



Précision sur la position voulu :  $\pm 1\text{ Hz}$  donc variation de  $\pm 1/8$  de tour

## 2) FFT et TFD

Principe pour une fonction temporelle  $x(t)$ :

- discrétiser la fonction temporelle,
- tronquer la fonction temporelle,
- discrétiser la fonction fréquentielle.

$$X(f) = \int_{-\infty}^{+\infty} x(t) e^{-j2\pi f t} dt$$

On approche l'intégrale par une somme d'aires de rectangles de durée  $T_e$  entre  $[0, (N-1)T_e]$  :

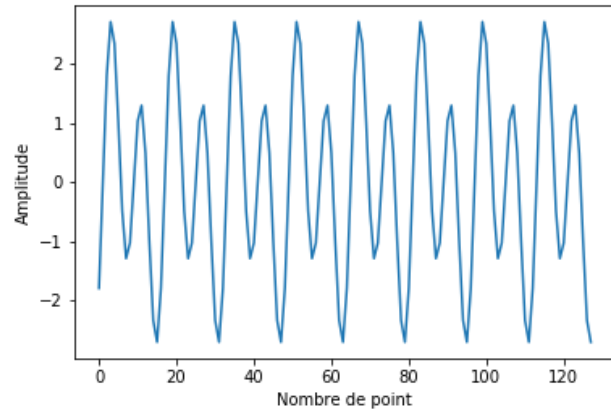
$$X(f) \simeq T_e \sum_{n=0}^{N-1} x(nT_e) e^{-j2\pi f n T_e}$$

Les fréquences valent

$$\text{donc : } F_k = k * \frac{F_e}{N}$$

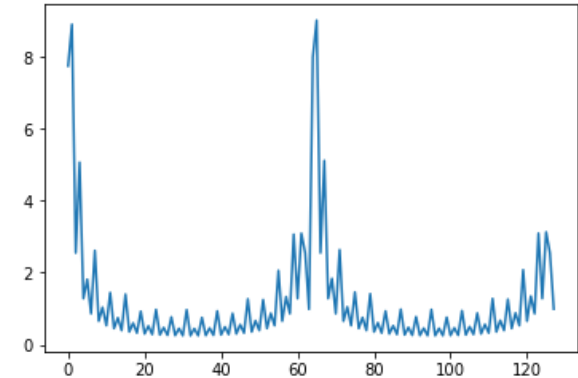
$$X(f_k) \simeq T_e \sum_{n=0}^{N-1} x(nT_e) e^{-\frac{j2\pi n k}{N} f_e T_e} \simeq T_e \sum_{n=0}^{N-1} x(nT_e) e^{-\frac{j2\pi n k}{N}}$$

## Simulation du programme :

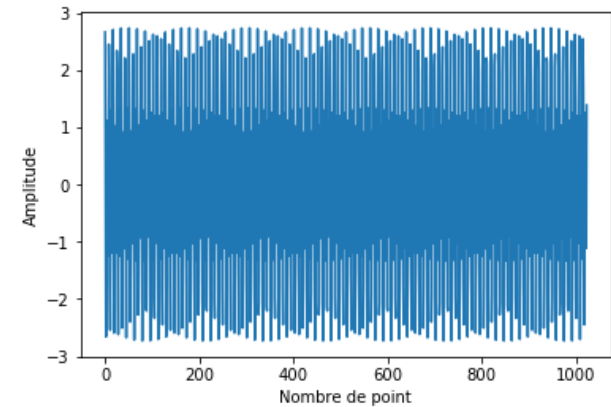


Signal simulé

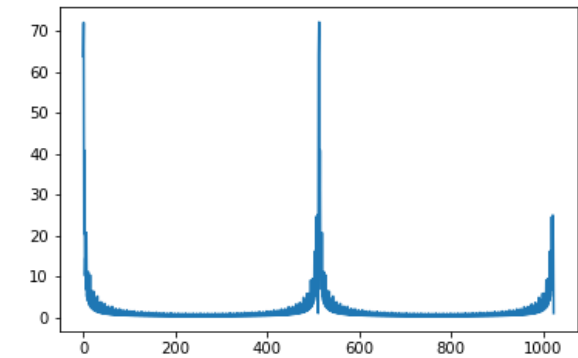
Simulation pour un  
nombre de points  $N=2^7$



Spectre



Simulation pour un  
nombre de points  $N=2^{10}$





## III] Correction de fréquence



Système d'acquisition du  
couple

## Mesure du couple minimum nécessaire pour faire tourner les mécaniques

Acquisition sur une guitare acoustique :

Distance(cm)	19,5	18	16	14
Force (N)	0,5	0,58	0,67	0,75
Couple (N.m)	0,097	0,104	0,107	0,105

Montée en  
fréquence

Distance(cm)	19,5	18	16	14
Force (N)	0,38	0,45	0,52	0,6
Couple (N.m)	0,074	0,081	0,083	0,084

Descente en  
fréquence

# Vérification sur une guitare électrique

	<u>Couple moyen</u> <u>(N.m) :</u>	<u>Ecart type :</u>
<u>Guitare acoustique</u> : Montée	0,103	0,003
<u>Guitare acoustique</u> : Descente	0,080	0,003
<u>Guitare électrique</u> : Montée	0,970	0,001
<u>Guitare électrique</u> : Descente	0,956	0,001

Comparaison du couple minimum entre une guitare  
électrique et une guitare acoustique

# Conclusion

---



```

import math as mt
import matplotlib.pyplot as plt
from Oksinus import *

def embrouille(expo, F2):
    FFTre=[0]*max
    print("\n\nexpo=", expo)
    for i in range(expo):
        double=int(2**i)
        #print("i=%s et double=%s" % (i, double))
        for j in range(double):
            #print("\tj=%s, indice[j]=%s" % (j, indice[j]))
            indice[j]=indice[j]*2
            #print("\tj=%s, indice[j]=%s" % (j, indice[j]))
            indice[j+double]=indice[j]+1
            #print("\tj+double=%s, indice[j+double]=%s" % (j+double, indice[j+double]))
    for i in range(NbPts):
        FFTre[i]=F2[indice[i]]
    print(FFTre)
    return FFTre

def entrelace(Le):
    long=len(Le)
    expo=int(mt.log2(long))
    Ls=[None]*long
    print("long=%s, expo=%s, Ls=%s" % (long, expo, Ls))

    for j in range(long):
        nBin=""
        n=j
        for i in range(expo-1):
            nBin=nBin+str(n%2)
            n=n//2
        nBin=nBin+str(n%2)

        poids=len(nBin)-1
        indice=0
        for bit in nBin:
            indice=indice + int(bit)*2**(poids)
            poids=poids-1
        print("indice=%s, n=%s" % (indice, n))
        Ls[j]=Le[indice]
    return Ls

```

```

def calcul(NbCol, FFTre):
    FFTim=[0]*max
    FFTmod=[0]*max

    for NumCol in range(NbCol):
        NbTre=2**(NbCol-NumCol)
        NbPap=int(2**(NumCol-1))

        for NumTre in range(NbTre):
            BasePap=NumTre*2**NumCol

            for OffsetPap in range(NbPap-1):

                angle=2*mt.pi*(OffsetPap/2**NumCol)
                co=mt.cos(angle)
                si=mt.sin(angle)

                NumPap=BasePap+OffsetPap

                re=co*FFTre[NumPap+NbPap]+si*FFTim[NumPap+NbPap]
                im=co*FFTim[NumPap+NbPap]-si*FFTre[NumPap+NbPap]

                FFTre[NumPap+NbPap]=FFTre[NumPap]-re
                FFTim[NumPap+NbPap]=FFTim[NumPap]-im

                FFTre[NumPap]=FFTre[NumPap]+re
                FFTim[NumPap]=FFTim[NumPap]+im

    for i in range(NbPts):
        FFTre[i]=FFTre[i]/NbPts
        FFTim[i]=FFTim[i]/NbPts
        FFTmod[i]=mt.sqrt(FFTre[i]**2 + FFTim[i]**2)
    return FFTmod

```

```
N=10
max=2*N
NbPts=2*N
print("Il y a %s points"%NbPts)
rapportCycl=0.5

FH=[i for i in range(int(NbPts*rapportCycl))]
FB=[i for i in range(int(NbPts*rapportCycl),NbPts)]
F=FH+FB

NbPer=5000
F=[mt.sin(2*mt.pi*NbPer*(i-1)/NbPts)+2*mt.sin(2*mt.pi*NbPer*2*(i-1)/NbPts) for i in range(NbPts)]
#print(F)

indice=[0]*max
print()
print()
F=sinus(N,NbPer,16)
F=[i for i in range(2*N)]
plt.plot(F)
plt.show()

FFTre=entrelace(F) #embrouille(N,F)
plt.plot(FFTre)
plt.show()

#print("Embrouille: ",FFTre)

FFTmod=calcul(N,FFTre)
#print(FFTmod)

plt.plot(FFTmod)
plt.show()
```