Ashly Abigail Huidobro
Oswaldo Alejandro Viviano Posadas
Jacobo Vazquez Santiago
Jaimes Romano José Eduardo

Martes, 12 de marzo de 2019

Simulacíon de un Espectro de ^{1}H para una señal con acoplamientos

Se desarrollo el siguiente código adaptado del link:
www.pybonacci.org/2012/09/29/transformada-de-fourier-discreta-en-python-con-scipy

Con el objetivo de simular acoplamientos de 0 a 6 hidrógenos vecinos.

El código se desarrolló sobre código Python 3.6

```
# -*- coding: utf-8 -*-
"""
Created on Sun Mar 10 18:22:16 2019

@author: Eduardo Jaimes

codigo adaptado de: https://www.pybonacci.org/2012/09/29/transformada-de-
fourier-discreta-en-python-con-scipy/

"""

import matplotlib.pyplot as plt
import numpy as np
from numpy import pi
from scipy.fftpack import fft, fftfreq

f = 500 #Hz <-- NOTA: Aqui hay un pequeño problema sobre el código


x1 = 6.0 #ppm desplazamiento químico # cambiar esta varible


n4 = (2**8) # de puntos
t4 = np.linspace(0, 1, n4)# ajuste de puntos extención de 256 puntos en
0-1
dt4 = t4[1] - t4[0]#delta

# multiplicidad

nH = 6 # número de protones que ve Cambiar esta variable
nH = nH/2
if nH == 2:
    y4 = np.cos(2 * pi * f * t4) + 0.5*np.cos(2 * pi * 2 * f * t4)-1
elif nH == 0.5:
    y4 = np.cos(2 * pi * f * t4)+np.cos(2 * pi * f * t4)
elif nH ==1.5:
    y4 = nH*np.cos(2 * pi * f * t4)+np.cos(2 * pi * 2* f * t4)
elif nH == 2.5:
```

Ashly Abigail Huidobro
Oswaldo Alejandro Viviano Posadas
Jacobo Vazquez Santiago
Jaimes Romano José Eduardo

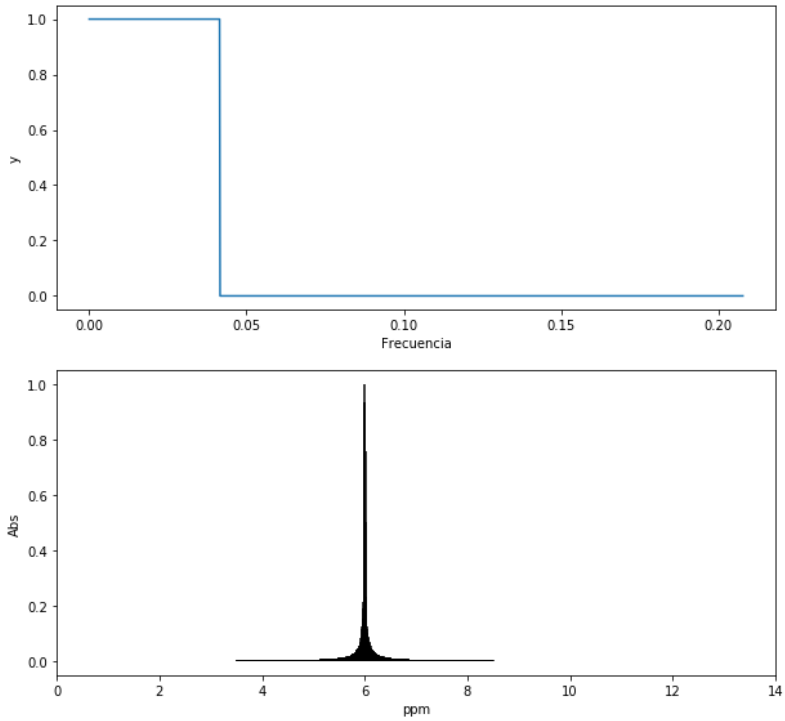
Martes, 12 de marzo de 2019

```
    y4 = (np.cos(2 * pi*f * t4) + 0.5*np.cos(2 * pi * f* 2 * t4)+
0.25*np.cos(2 * pi *f* 3 * t4))
elif nH ==3:
    y4 = (np.cos(2 * pi*f * t4) + 0.5*np.cos(2 * pi * f* 2 * t4)+
0.25*np.cos(2 * pi *f* 3 * t4))-1
elif nH ==1:
    y4 = nH*np.cos(2 * pi * f * t4)-1
elif nH == 0:
    y4 = nH*np.cos(2 * pi * f * t4)+1

#Transformada de fourier

y5 = y4 * np.blackman(n4)
t4 = np.linspace(0, 0.2 + 2 * dt4, 5 * n4)
y4 = np.append(y4, np.zeros(4 * n4))
y5 = np.append(y5, np.zeros(4 * n4))
Y4 = fft(y4) / (n4)
frq4 = fftfreq(5 * n4, 0.2)

#Gráficas
fig = plt.figure(figsize=(10, 20))

ax1 = fig.add_subplot(411)
ax1.plot(t4, y4,)

plt.xlabel('Frecuencia')
plt.ylabel('y')
ax2 = fig.add_subplot(412)
ax2.vlines((frq4 + x1), 0, abs(Y4) #transformada de Fourier Espectro de
RMN
plt.xlim(right=14)
plt.xlim(left=0)
plt.xlabel('ppm')
plt.ylabel('Abs')
```

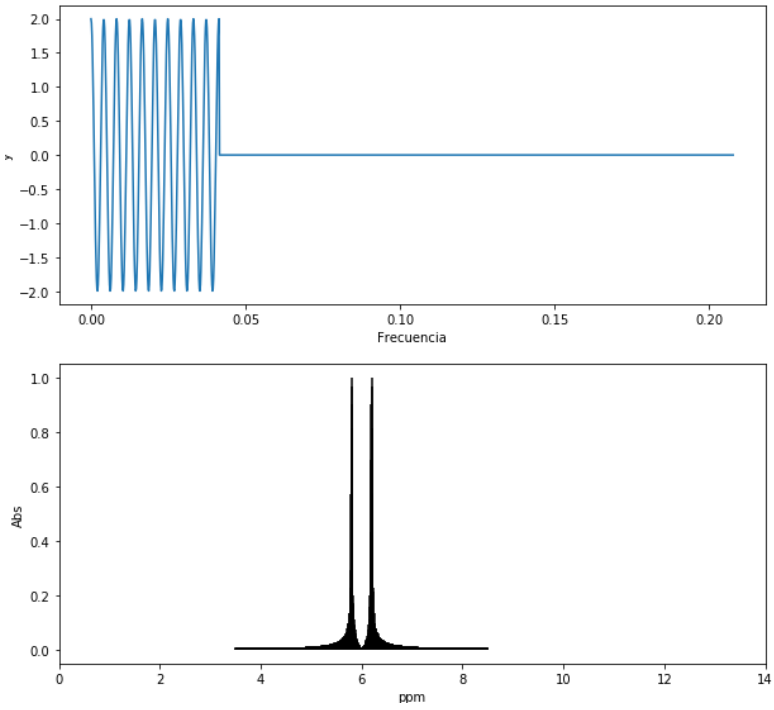
Ashly Abigail Huidobro
Oswaldo Alejandro Viviano Posadas
Jacobo Vazquez Santiago
Jaimes Romano José Eduardo

Martes, 12 de marzo de 2019

Señal para 0 Hidrógenos vecinos



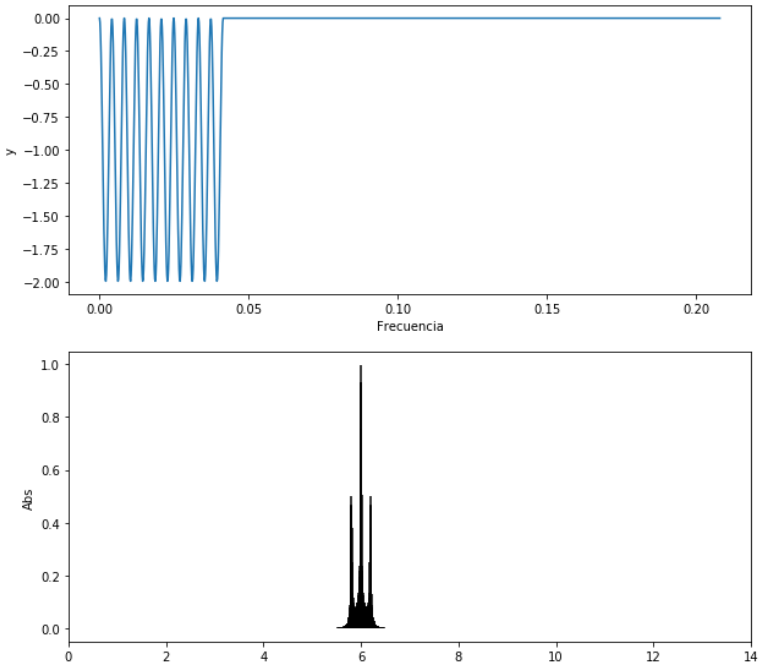
Señal y acoplamiento para 1 Hidrógeno vecino



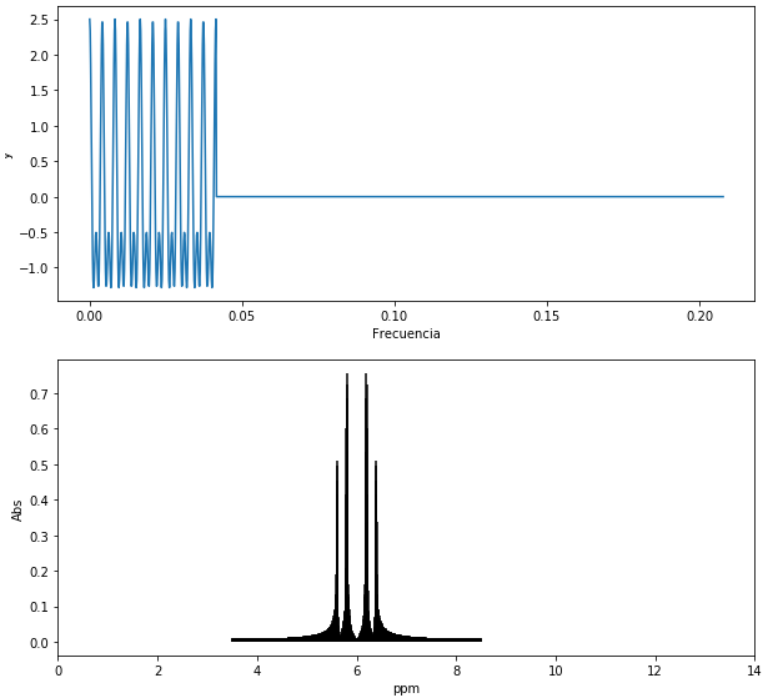
Ashly Abigail Huidobro
Oswaldo Alejandro Viviano Posadas
Jacobo Vazquez Santiago
Jaimes Romano José Eduardo

Martes, 12 de marzo de 2019

Señal y acoplamiento para 2 Hidrógenos vecino



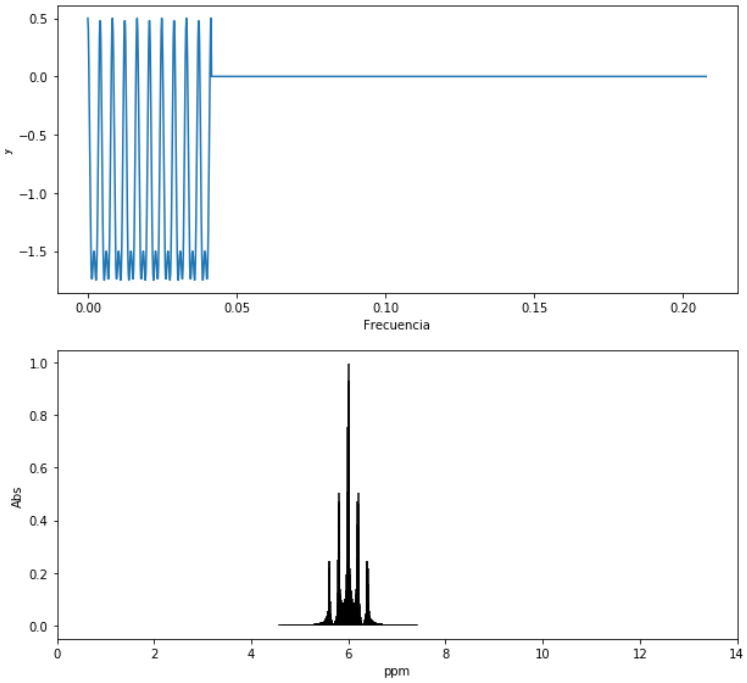
Señal y acoplamiento para 3 Hidrógenos vecino



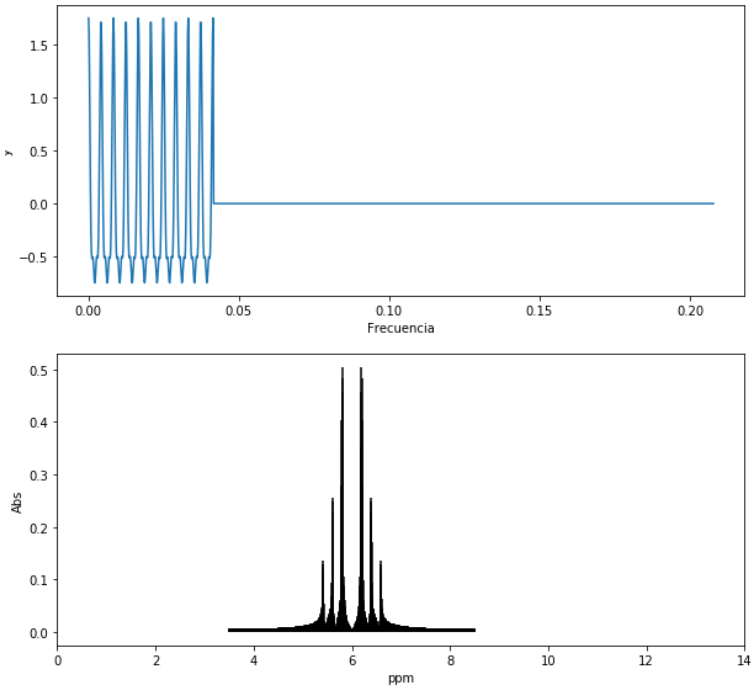
Ashly Abigail Huidobro
Oswaldo Alejandro Viviano Posadas
Jacobo Vazquez Santiago
Jaimes Romano José Eduardo

Martes, 12 de marzo de 2019

Señal y acoplamiento para 4 Hidrógenos vecino



Señal y acoplamiento para 5 Hidrógenos vecinos



Ashly Abigail Huidobro
Oswaldo Alejandro Viviano Posadas
Jacobo Vazquez Santiago
Jaimes Romano José Eduardo

Martes, 12 de marzo de 2019

Señal y acoplamiento para 6 Hidrógenos vecinos