
Echant_TFD_Fourier

J.-F Bercher

November 15, 2013

Part I

Introduction à la TFD et l'analyse de Fourier

On aura besoin des fonctions suivantes

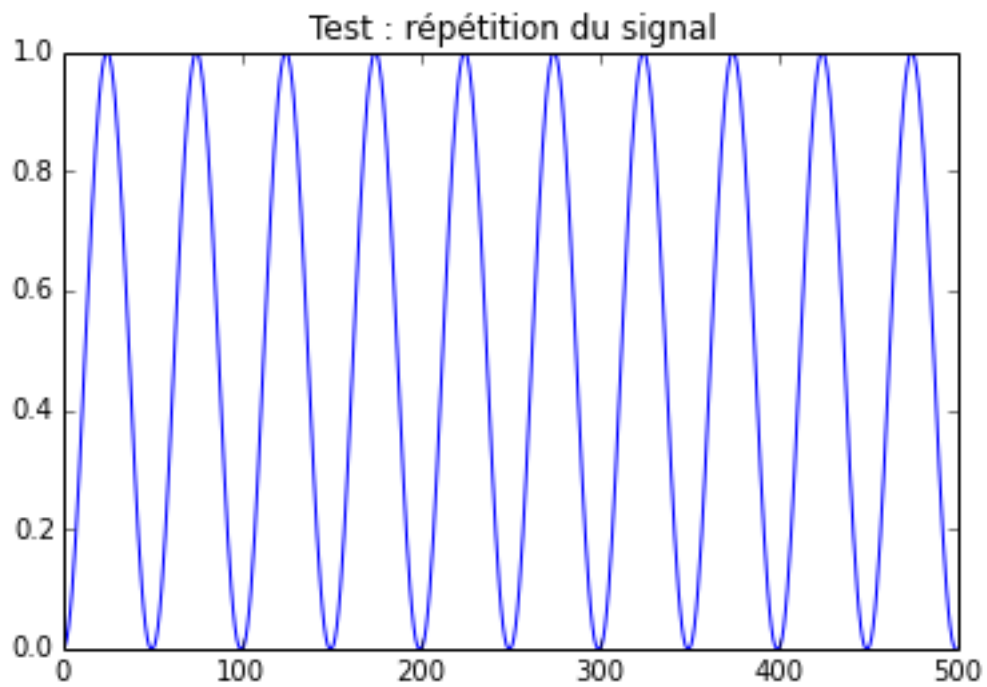
```
In [3]: from scipy.signal import hann
        from numpy.fft import fft, ifft
```

1 Vers la TFD. Dualité échantillonnage-périodisation

1. Créer une fonction *repeat(x,n)* permettant de périodiser un signal. Vous pourrez utiliser pour ce faire le fait que sous Python, $[(x)*n]=((x) (x) \dots (x))$, où x est une liste quelconque.
2. Tester cette fonction, par exemple en prenant $x=randn(10)$, ou $x=hann(50)$,

```
In [79]: def repeat(x,n):
        u""" repeat(x,n) -- x ndarray, n integer
        Répète n fois la séquence x """
        return array([list(x)*n]).flatten()
```

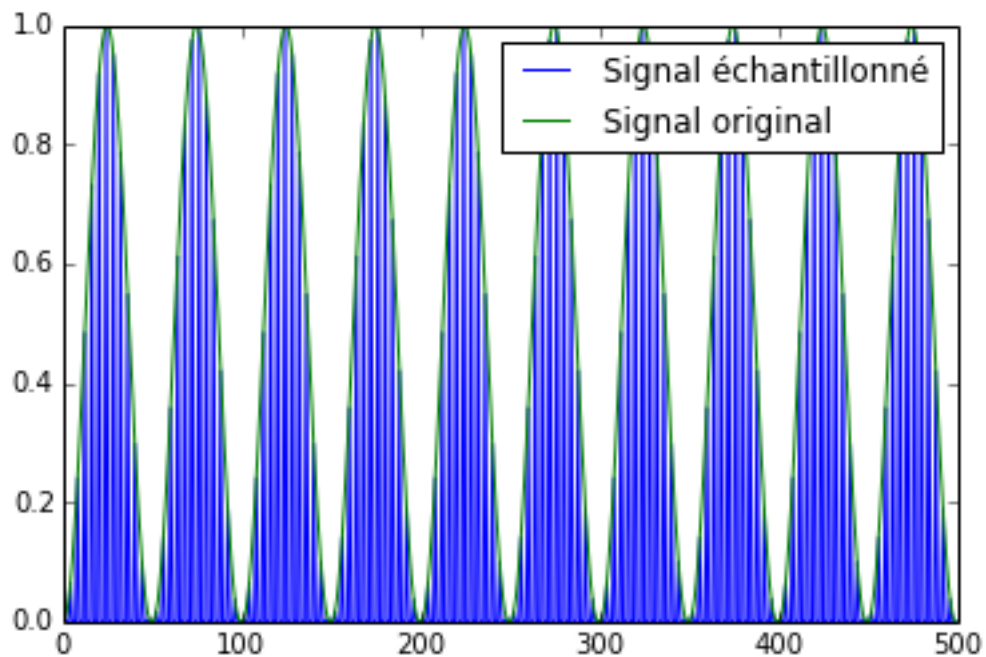
```
In [12]: # Test repeat
w=hann(50,sym=True)
rr=array(repeat(list(w),10))
#r=rr.reshape(rr.size,1)
r=rr.flatten()
figure(1)
clf()
plot(r)
title('Test : répétition du signal' )
show()
```



1. Créer une fonction *echant*(*x*,*step*) permettant de sous échantillonner un signal, en ne retenant qu'un point tous les -step_.
2. Tester cette fonction, par exemple en prenant *x=range(100)*, *x=randn(10)*, ou *x=hann(50)*,

```
In [25]: def echant(x, step):
          z=zeros(x.shape)
          z[0:len(x):step]=x[0:len(x):step]
          return z
```

```
In [26]: # test echant
          L=len(r)
          t=arange(L)
          figure(2)
          clf()
          plot(t,echant(r,4),color='blue',label=u'Signal échantillonné')
          plot(t,r,color='green',label='Signal original')
          legend()
          show()
```



1.1 Manipulations -

Vous disposez d'un signal $x(n)$, échantillonné à $F_e = 32$.

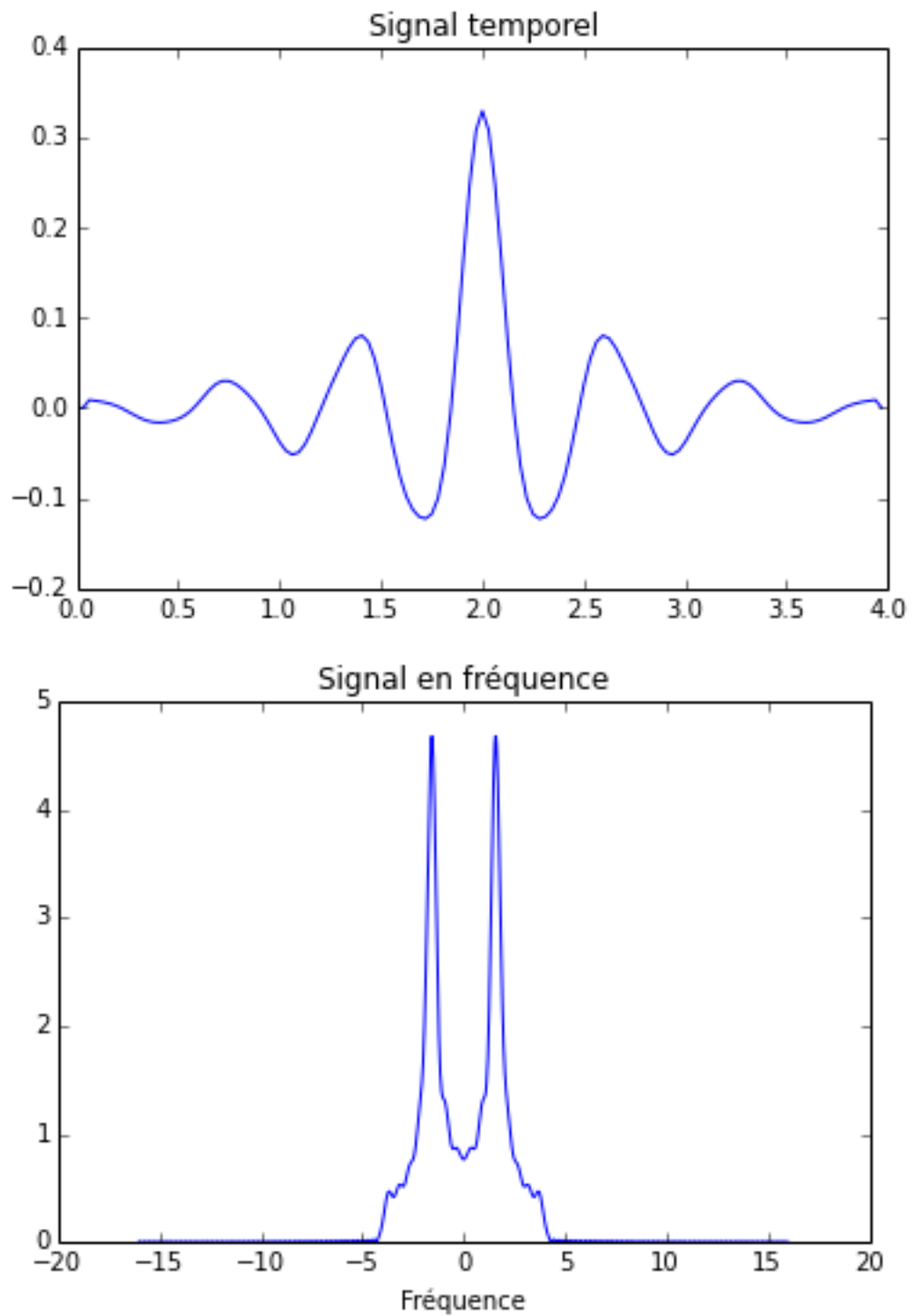
1) Chargez ce signal par

```
f=numpy.load('signal.npz')
#f.keys()
x=f['x'].flatten()
```

Le signal est alors chargé dans l'environnement sous le nom x . Visualisez x dans le domaine temporel et fréquentiel. Quelle est sa durée temporelle ? Quelle est approximativement la bande occupée ?

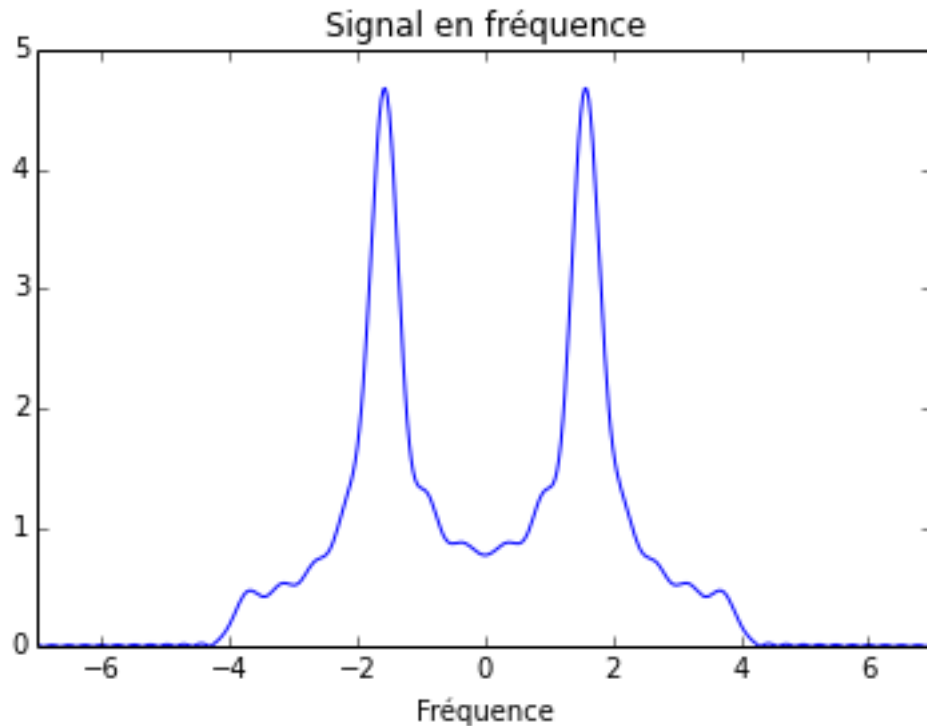
```
In [67]: #===== LE TP =====
f=numpy.load('signal.npz')
#f.keys()
x=f['x'].flatten()
# On décide que la fréquence d'échantillonnage Fe vaut 32
Fe, Te = 32, 1/32
N=len(x)
M=8*N #Utilisé comme longueur des fft
# Et on définit les vecteurs t et f
t=arange(N)*Te
f=(arange(M)/M-1/2)*Fe

figure(3)
#subplot(121)
plot(t,x)
title('Signal temporel')
figure(4)
#subplot(122)
plot(f,abs(fftshift(fft(x,M))))
title('Signal en fréquence')
xlabel('Fréquence')
show()
```



Zoomons un peu sur cette représentation :

```
In [68]: figure(4)
         xf=fftshift(fft(x,M))
         plot(f,abs(xf))
         title('Signal en fréquence')
         xlabel('Fréquence')
         xlim([-7, 7])
         show()
```



On voit donc que la bande de fréquence est de l'ordre de 4.1 Hz

La durée temporelle, quant-à-elle, est de l'ordre de 4 s.

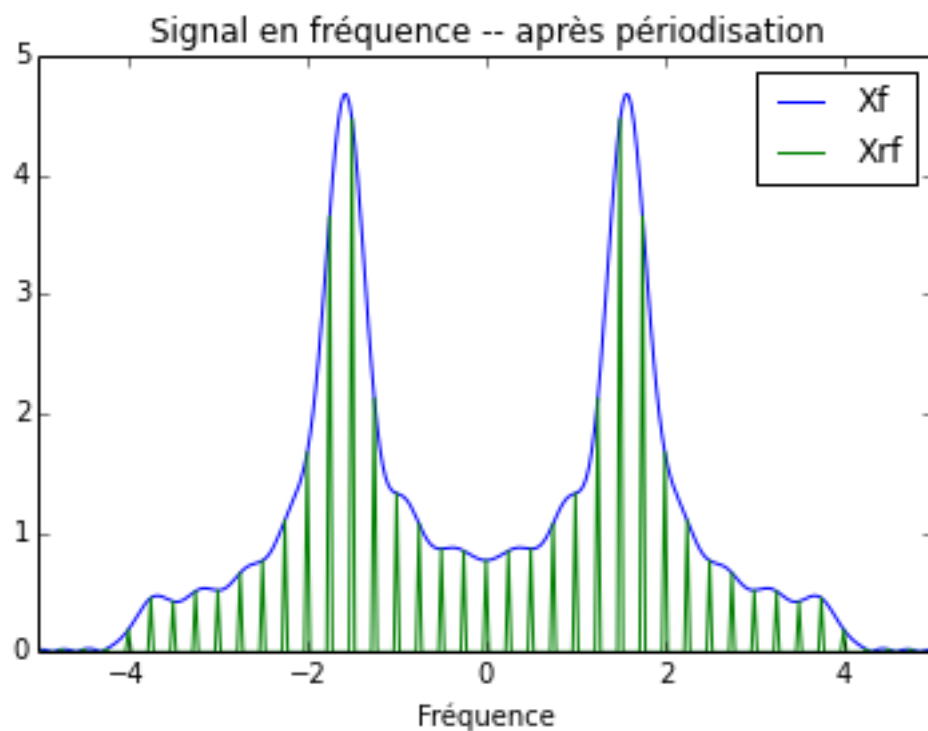
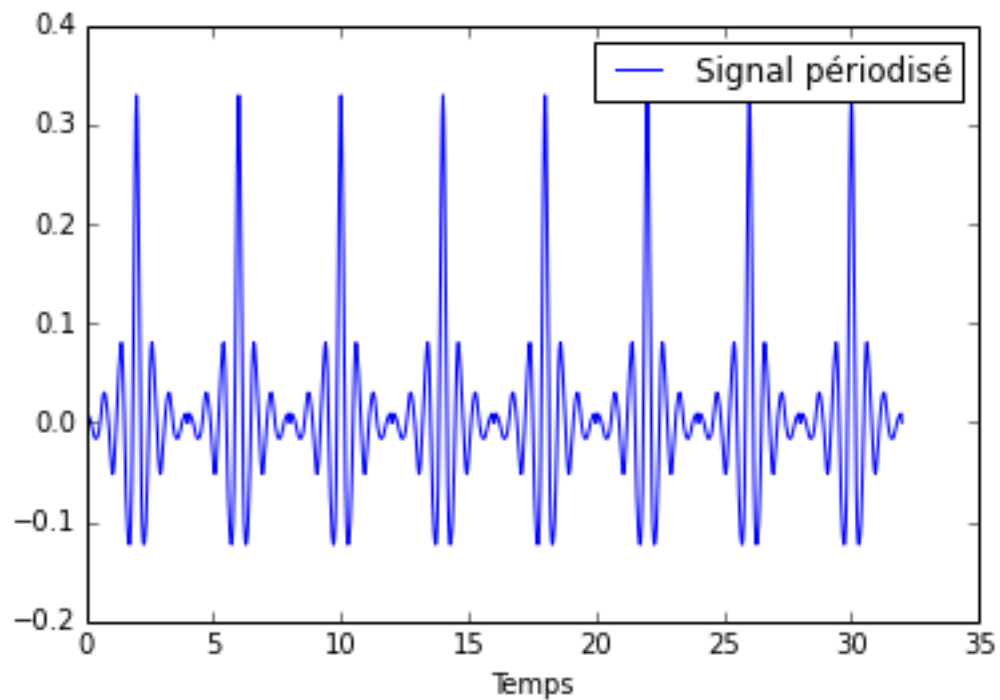
- 2) On étudie d'abord l'effet d'une répétition du signal. Créez un nouveau signal, $x_r(n)$ et répétant 8 fois le motif $x(n)$ (fonction repeat). Visualisez le signal temporel, puis comparez les réponses en fréquence de $x(n)$ et $x_r(n)$. Conclusions.

```
In [90]: xr=repeat(x,8)
tr=arange(len(xr))*Te
xrf=fftshift(fft(xr,M))

figure(5)
clf()
plot(tr,xr, label='Signal périodisé')
#plot(arange(len(xr))*Te,xr, label='Signal périodisé')
#
#stem(t,xe, linefmt='g-', markerfmt='bo', basefmt='b-')
#plot(t,xe, label=u'Signal sous échantillonné')
xlabel('Temps')
legend()

figure(6)
plot(f,abs(xf),label='Xf')
plot(f,abs(xrf)/8,label='Xrf')
legend()
title('Signal en fréquence -- après périodisation')
xlabel('Fréquence')
xlim([-5,5])
(-5, 5)
```

Out [90]:



On peut vérifier que l'intervalle en les raies est de 0.25, soit l'inverse de la période du signal. On observe ainsi que la périodisation induit un échantillonnage en fréquence, et que si la périodisation a été effectuée avec un pas de T , alors cet échantillonnage fait apparaître des raies fréquentielles tous les $1/T$

$$\text{Périodisation à } T \rightarrow \text{échantillonnage à } \frac{1}{T}$$

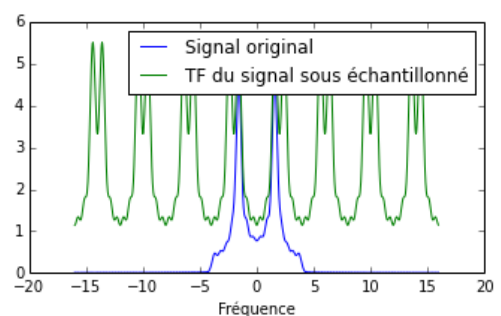
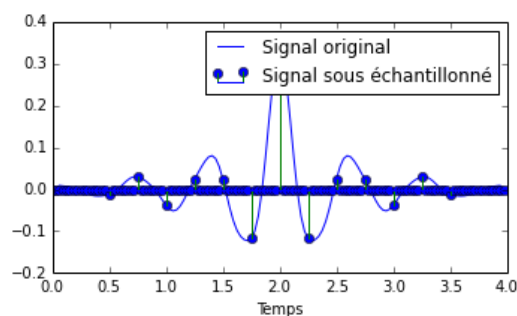
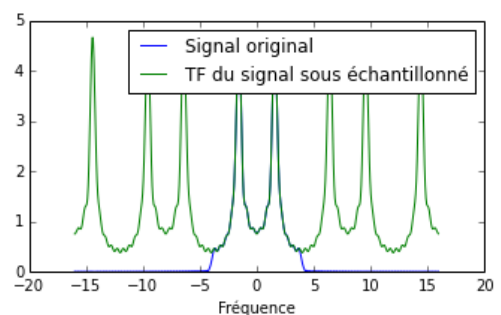
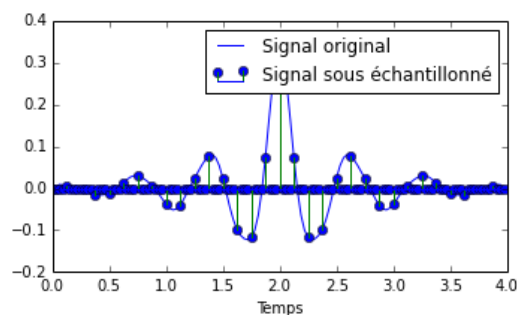
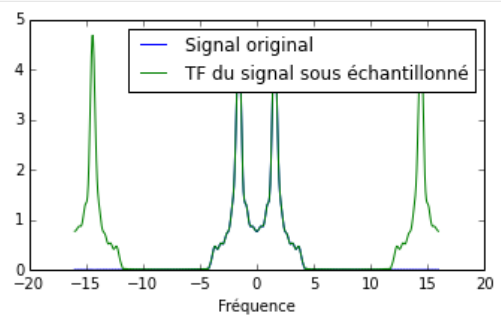
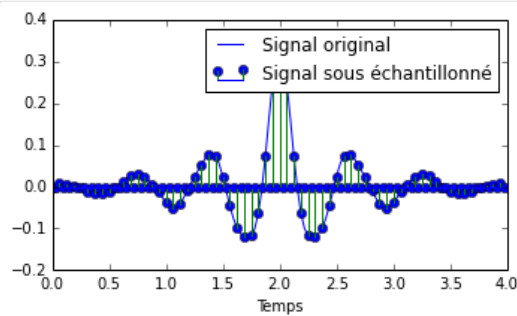
Remarque : on pourra vérifier que pour la TFD, on a en réalité

$$w_L(n) \Leftrightarrow L \times w_{\frac{1}{T}}(k)$$

- 3) On s'intéressera ensuite aux effets de l'échantillonnage : rééchantillonnez le signal aux fréquences $F_{se} = 16$, $F_{se} = 8$, $F_{se} = 4$ (créez les signaux $x_{e1}(n)$, $x_{e2}(n)$ et $x_{e3}(n)$), en utilisant la fonction `echant`. Visualisez les signaux temporels, et comparez les réponses fréquentielles (toujours sur $[-F_e/2, F_e/2]$, avec $F_e = 32$, la fréquence d'échantillonnage initiale).

```
In [120]: def sech(x,k,M=M):
            """ Sous échantillonnage de x par un facteur k
            Rend x sous échantillonné et sa TF """
            xe=echant(x,k)
            xef=fftshift(fft(xe,M)).flatten()
            return (xe, xef)
```

```
In [124]: num=7
            for k in (2, 4, 8):
                figure(num+1,figsize=(12, 3))
                subplot(1,2,1)
                plot(t,x, label='Signal original')
                stem(t,sech(x,k)[0],linefmt='g-',markerfmt='bo', basefmt='b-',label=u'Signal sou
                xlabel('Temps')
                legend()
                #
                subplot(1,2,2)
                plot(f,abs(xf), label='Signal original')
                xef=sech(x,k)[1]
                plot(f,k*abs(xef),label='TF du signal sous échantillonné')
                # Le facteur k ci-dessus prend en compte la puissance perdue dans le sous éch
                xlabel('Fréquence')
                legend()
                num+=1
```



On observe ainsi que l'échantillonnage entraîne une périodisation en fréquence, et que s'échantillonnage a été effectué avec un pas de T_e , alors cette périodisation est de période $F_e = 1/T_e$

$$\text{échantillonnage à } T_e \rightarrow \text{Périodisation à } F_e = \frac{1}{T_e}$$

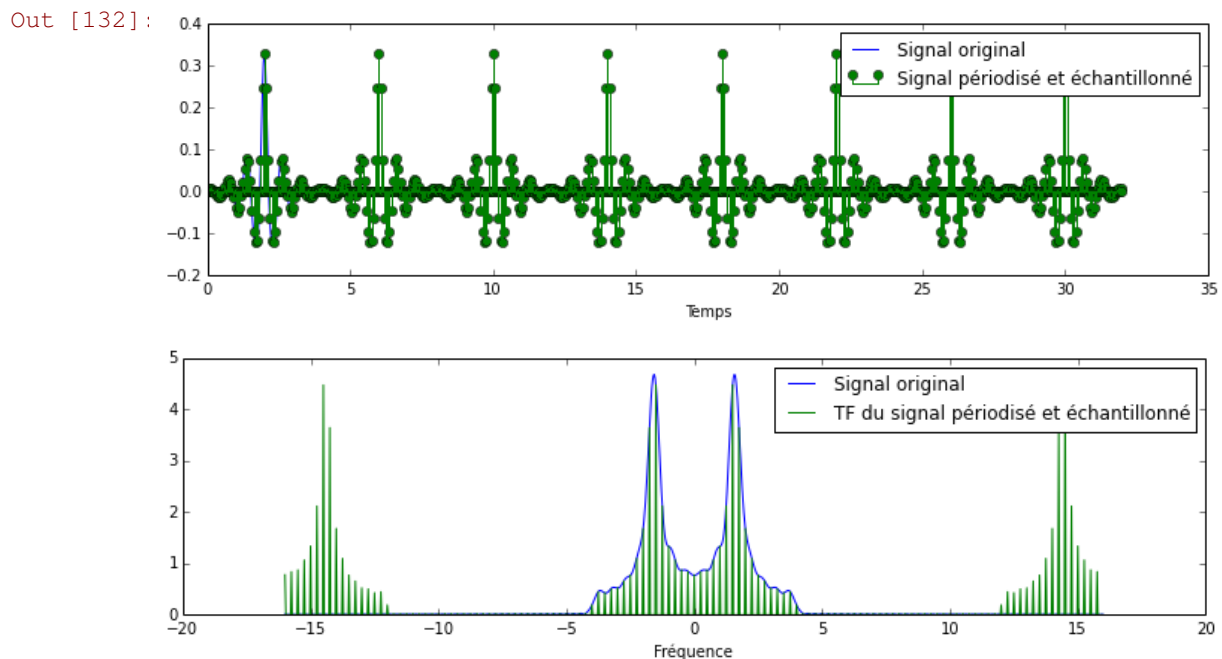
En outre, on vérifie bien la condition d'échantillonnage de Shannon : dès que la fréquence d'échantillonnage devient inférieur à $2 \times$ la bande, alors il peut se produire du *recouvrement*. On termine en examinant ce qu'il se passe lorsque l'on a à la fois échantillonnage et périodisation. Très naturellement, les deux effets se conjuguent et on associe alors à une séquence périodique échantillonnée une autre séquence périodique échantillonnée.

- 4) Créez enfin un signal périodique échantillonné $xre(n)$, en périodisant le signal initial (en créant par exemple 8 périodes) puis en échantillonnant le signal résultant. Analysez le signal obtenu en temps et en fréquence.
- Conclusions.

```
In [132]: num=11
xre=echant(repeat(x,8),2)
figure(num+1,figsize=(12, 3))
plot(t,x, label='Signal original')
stem(tr,xre, linefmt='g-', markerfmt='go', basefmt='g-', label=u'Signal périodisé et échantillonné')
xlabel('Temps')
legend()

#
figure(num+2,figsize=(12, 3))
plot(f,abs(xf), label='Signal original')
plot(f,abs(fftshift(fft(xre,M)))/4,label='TF du signal périodisé et échantillonné')
# Le facteur k ci-dessus prend en compte la puissance perdue dans le sous échantillonné
xlabel('Fréquence')
legend()

<matplotlib.legend.Legend at 0x7f5483426c10>
```



Ces deux séquences échantillonnées périodiques sont reliées par une transformée de Fourier. Il n'est utile de calculer cette transformée de Fourier que sur *une* période de $N = T/T_e$ points (puisque le signal est périodique), et il lui correspondra une TF périodique avec une période également sur $F_e/F = T/T_e = N$ points. On arrive alors à une correspondance entre deux séquences de N points. C'est l'essence de la TFD.

1.2 Trucs sans rapport

$$\begin{aligned} ab &= c f s s \\ f v d h &= d f d f \end{aligned}$$

Toto

Truc

$$+1 \\ E = mc^2 \tag{1}$$

```
In [23]: ind=where(isnan(y))  
         y[ind]=1
```