NOM :

Prénom :

TP1 - Numérisation et echnatillonage

Le signal à étudier est du type :

$x(t) = \text{Asin}(2\pi f_0 t)$

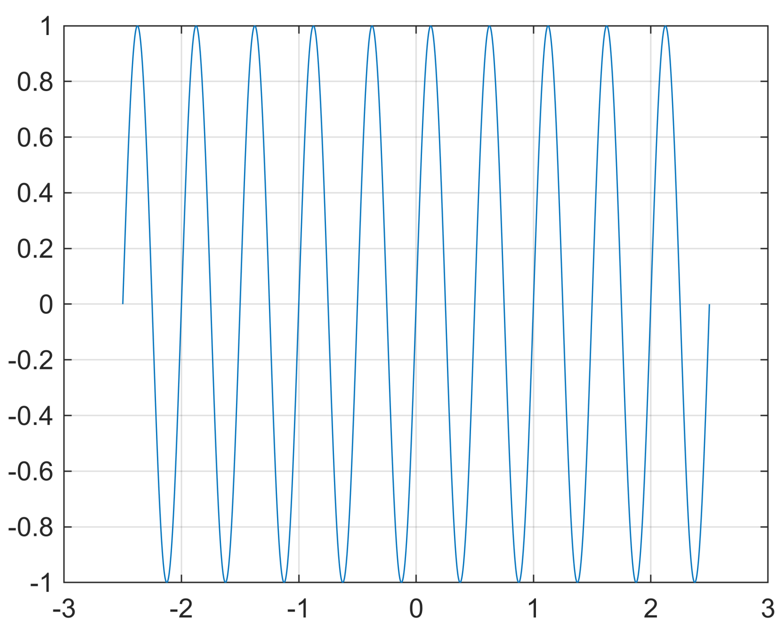
avec *f0*=2000Hz et *A*=1V.

Définition du signal continu et de son spectre

Question 1 :

Calculer puis tracer le signal x(t) sur 10 périodes et centré en t=0, avec une fréquence d'echantillonage de 1000*f0. Celle-ci est suffisament grande pour que par la suite, x(t) soit considéré comme notre signal continu.

```
f0=2000;
fe=1000*f0;
t0=1/2000;
t=-5*t0:1/(1000*2000):+5*t0;
x=sin(2*pi*2000*t);
plot(t,x)
grid on;
```

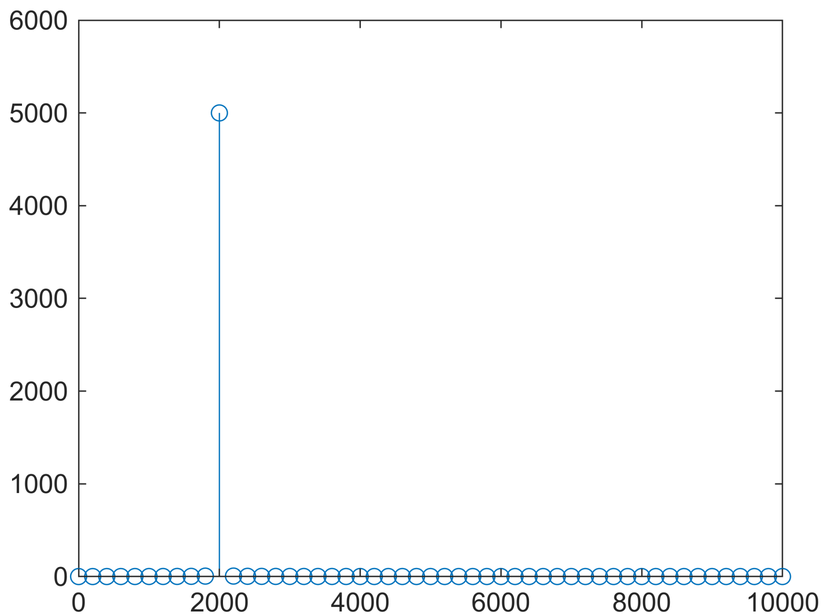


Question 2 :

Calculer puis tracer schématiquement le spectre (non centré, à l'aide de `fft()`) en amplitude |*X(f)*|. Pour l'affichage, utiliser `xlim()` afin de restreindre l'axe des abscisses de 0 à 5 fois à f0. Utiliser la fonction `stem()` pour tracer en barre (voir aide). Retrouve-t-on le comportement attendu ?

Reponse : On observe bien que c'est une transformée de fourrier d'un signal sinusoidal avec un pic a f0=2000HZ

```
X=fft(x);
Xf=abs(X);
n=length(Xf);
freq=0:fe/n:(fe-fe/n);
stem(freq,Xf);
xlim([0 5*f0]);
```



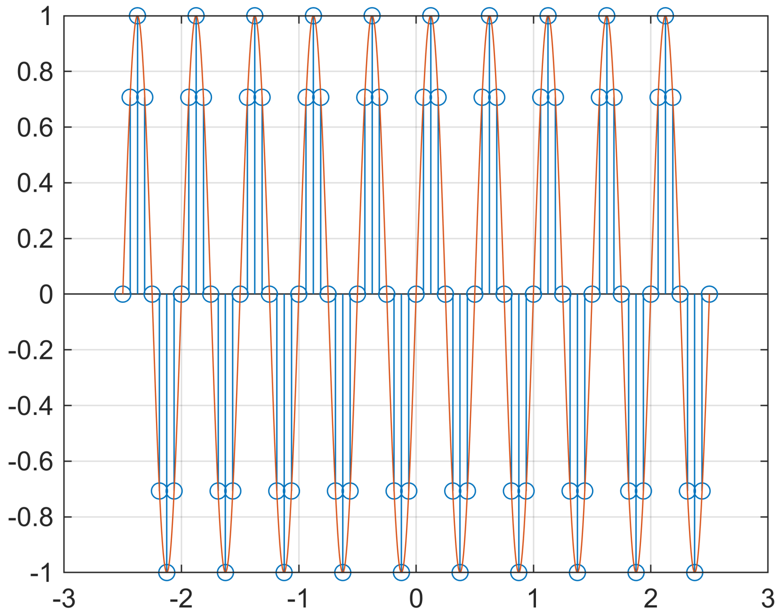
Un premier échantillonnage

Question 3 :

La fréquence d'échantillonnage de *x(t)* est fixée à `fe`=16000Hz, respecte-t-on le critère de Shannon? Déterminer les valeurs des échantillons prélevées sur *x(t)*. Utiliser ces valeurs pour tracer sur un même graphe *x(t)* et *xb(t)*: le signal échantillonné bloqué (le blocage consiste à maintenir la valeur du signal échantillonné constante entre 2 instants d'échantillonnage successifs). Vous pouvez également utiliser `stem()` pour afficher le signal échantilloné.

```
fe1=16000;
t1=-5*t0:1/(16000):+5*t0;
x1=sin(2*pi*2000*t1);
stem(t1,x1)
hold on
plot(t,x)
```

```
grid on;
```

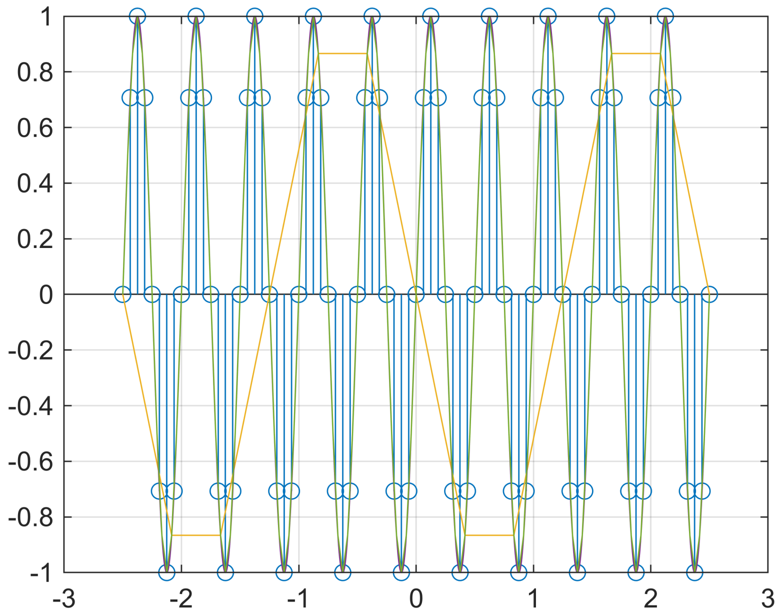


Interprétation temporelle de la notion d'échantillonage

Question 4 :

Quelle est la fréquence `fmax` contenue dans le signal x(t), en déduire la fréquence minimale d'échantillonnage donnée par le critère de Shannon. Stocker cette valeur dans une variable `fShanon`. Echantillonner alors à une fréquence `feOK=6*fShanon` et `fePasOK=12/10*f0`. Tracer sur une même figure le signal continu *x(t)* , bien échnatilloné et mal échantilloné. Interpréter.

```
fePasOK=12/10*f0;
feOK=12*f0;
tPOK=-5*t0:1/(fePasOK):+5*t0;
xPOK=sin(2*pi*2000*tPOK);
plot(tPOK,xPOK)
hold on
plot(t,x)
tOK=-5*t0:1/(feOK):+5*t0;
xOK=sin(2*pi*2000*tOK);
plot(tOK,xOK)
grid on;
```



Interprétation fréquentielle de la notion d'échantillonage

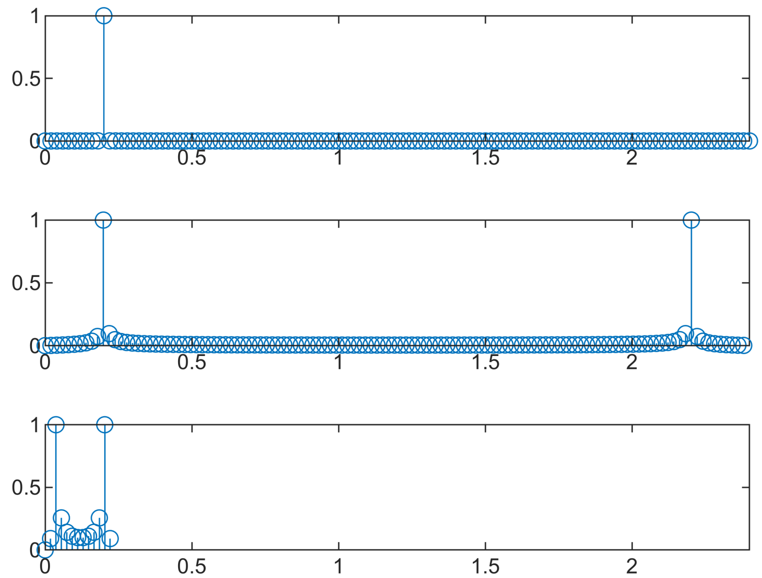
Question 5 :

En conservant le code couleur ci-dessus, calculer puis tracer (en utilisant `subplot()`) schématiquement les spectre (non centrés) en amplitude |*X(f)*| normalisés à 1. Pour l'affichage, utiliser `xlim()` pour obtenir un axe des abscisses communs aux 3 spectres entre 0 et `feOK`. Justifier l'allure des spectres observés.

Reponse : dans les deux premiers graphes on observe un spectre normal car le critère de shanon est bien respecté Dans le cas pas Ok la frequence d'echantillonage est petite donc le critere de shanon n'est pas respecté, on observe alors un repliment de spectre fe-f0

```
subplot(3,1,1)
stem(freq,Xf/max(Xf));
xlim([0 feOK])
subplot(3,1,2)
nOK=length(xOK);
fOK=0:feOK/nOK:(feOK-feOK/nOK);
XOK=fft(xOK);
XfOK=abs(XOK);
stem(fOK,XfOK/max(XfOK))
xlim([0 feOK])
subplot(3,1,3)
nPOK=length(xPOK);
fPOK=0:fePasOK/nPOK:(fePasOK-fePasOK/nPOK);
XPOK=fft(xPOK);
XfPOK=abs(XPOK);
stem(fPOK,XfPOK/max(XfPOK));
```

```
xlim([0 feOK])
```



Interprétation temporelle et fréquencielle pour 5 fréquences d'echantillonage proche de Shanon

Question 6 :

A l'aide d'une boucle `for` et de `subplot()` (regarder l'aide si nécessaire), tracer pour 5 valeurs de frequence d'échantillonage entre `0.6*fshanon` et `1.9*fshanon` le signal temporel echantilloné et sa TF en vis à vis. Observer le phénomène de repliement de spectre. Afficher également sur ces figures le signal x(t) continu et sa TF. Pour la TF, limitez la fenetre à 2`fshanon`.

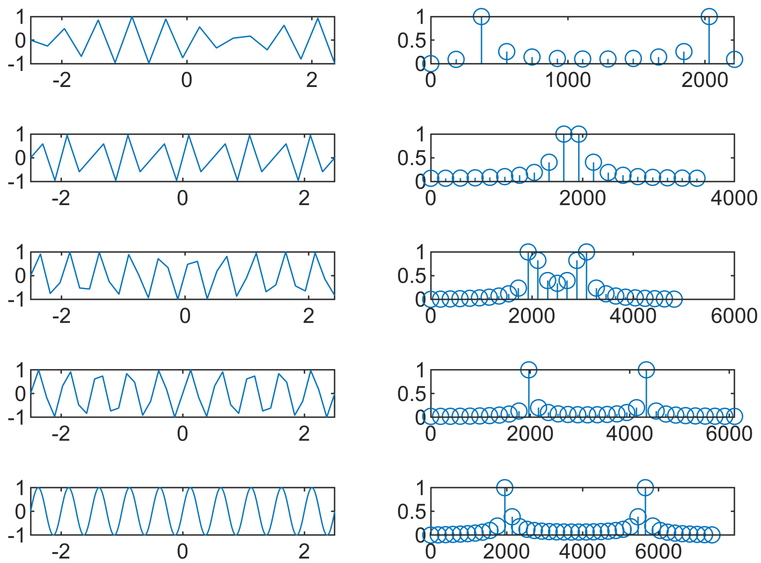
```
listeFrq=linspace(0.6*2*f0,1.9*2*f0,5)
```

```
listeFrq = 1×5
        2400         3700         5000         6300         7600
```

```
for i=1:5
 fei=listeFrq(i);
 ti=-5*t0:1/(fei):+5*t0;
 xi=sin(2*pi*2000*ti);
 plot(ti,xi);
 subplot(5,2,2*i);
 ni=length(xi)
 fi=0:fei/ni:(fei-fei/ni)
 xi=fft(xi)
 xfi=abs(xi)
 stem(fi,xfi/max(xfi))
 subplot(5,2,2*i-1)
 xlim([0 4*f0])
 plot(t,x)
```

```
end
```

```
ni = 13
fi = 1×13
10^3 ×
         0    0.1846    0.3692    0.5538    0.7385    0.9231    1.1077    1.2923 ···
xi = 1×13 complex
  -0.0000 + 0.0000i  -0.1287 + 0.5221i  -2.7479 + 5.2356i   1.0036 - 1.1328i ···
xfi = 1×13
    0.0000    0.5377    5.9129    1.5134    0.8340    0.6486    0.5845    0.5845 ···
ni = 19
fi = 1×19
10^3 ×
         0    0.1947    0.3895    0.5842    0.7789    0.9737    1.1684    1.3632 ···
xi = 1×19 complex
  -0.5681 + 0.0000i  -0.5702 - 0.0842i  -0.5768 - 0.1734i  -0.5897 - 0.2741i ···
xfi = 1×19
    0.5681    0.5763    0.6023    0.6503    0.7299    0.8617    1.0961    1.5963 ···
ni = 26
fi = 1×26
10^3 ×
         0    0.1923    0.3846    0.5769    0.7692    0.9615    1.1538    1.3462 ···
xi = 1×26 complex
  -0.0000 + 0.0000i   0.0048 - 0.0395i   0.0199 - 0.0806i   0.0475 - 0.1251i ···
xfi = 1×26
    0.0000    0.0398    0.0830    0.1338    0.1984    0.2869    0.4193    0.6390 ···
ni = 32
fi = 1×32
10^3 ×
         0    0.1969    0.3937    0.5906    0.7875    0.9844    1.1812    1.3781 ···
xi = 1×32 complex
  -0.2723 + 0.0000i  -0.2702 - 0.0589i  -0.2639 - 0.1204i  -0.2523 - 0.1878i ···
xfi = 1×32
    0.2723    0.2766    0.2900    0.3145    0.3537    0.4150    0.5135    0.6836 ···
ni = 39
fi = 1×39
10^3 ×
         0    0.1949    0.3897    0.5846    0.7795    0.9744    1.1692    1.3641 ···
xi = 1×39 complex
   0.0000 + 0.0000i   0.0060 - 0.0747i   0.0249 - 0.1530i   0.0590 - 0.2392i ···
xfi = 1×39
    0.0000    0.0750    0.1550    0.2464    0.3578    0.5038    0.7118    1.0418 ···
```



Restitution du signal par transformée inverse

Question 6 :

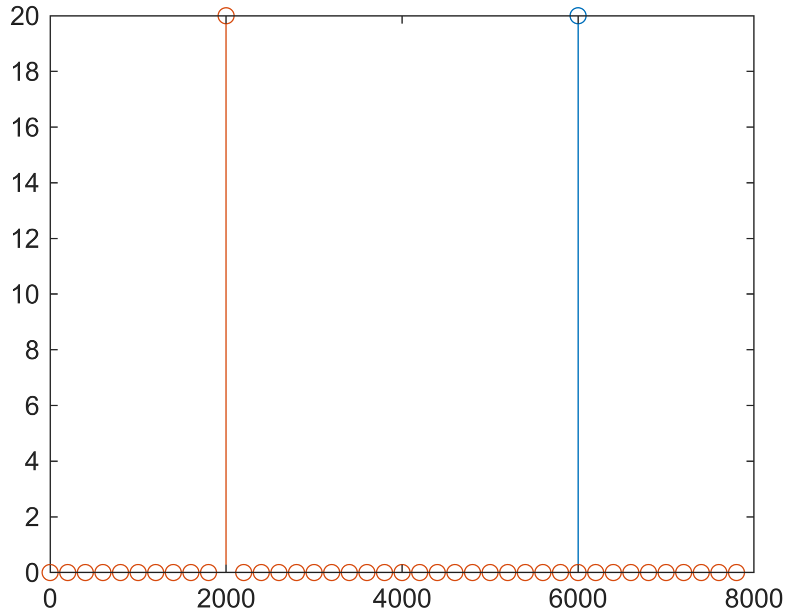
Calculer puis tracer schématiquement le spectre (non centré) en amplitude |*X(f)*| entre 0 et fe=8000 Hz. Justifier l'allure du spectre observé. Définir ensuite le filtre H(f) de type porte qui ne conserve que les fréquences comprises en 0 et fe/2. Afficher sur une même figure |*X(f)*| *et* |*X(f)filtre*|=|*X(f)*|x*H(f).*

```
fe6 = 8000;
te6 = 1/fe6;
A=1;
t6=-5*t0:1/(fe6):(5*t0-te6); %il faut enlever une periode (le dernier point)
x6=A*sin(2*pi*f0*t6);
N=length(x6);
X = fft(x6, N);
f = 0:fe6/N:(fe6-fe6/N)
```

```
f = 1×40
         0         200         400         600         800        1000 ···
```

```
X_amplitude = abs(X);
% Définir le filtre passe-bas
H = zeros(1, N);             % Initialiser le filtre
H(f <= fe6 / 2) = 1;         % Conserver les fréquences entre 0 et fe/2
% Spectre filtré
X_filtre = X .* H;           % Appliquer le filtre
X_filtre_amplitude = abs(X_filtre);
figure
stem(f, X_amplitude); hold on
```

```
stem(f, X_filtre_amplitude);
```



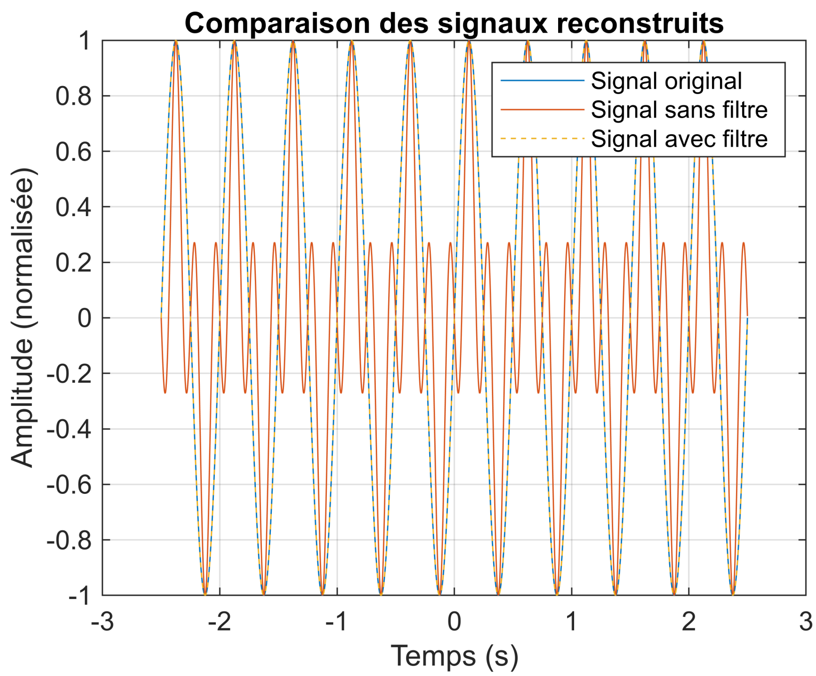
Question 7 :

Restituer alors le signal sans filtre anti-repliement et avec, le tout sur l'ensemble du signal temporel (utiliser `ifft(x,n)` cf. l'aide). Représenter alors sur une même figure le sigal initial x(t), ainsi que les 2 signaux reconstruits avec et sans filtre anti repliement. Normaliser à 1 ces signaux afin qu'ils aient la même amplitude. Conclure sur l'emploi du filtre anti-repliement.

```
% Restituer les signaux avec et sans filtre
x_RSF = ifft(X, length(x));         % Signal reconstruit sans filtre
x_RAF = ifft(X_filtre, length(x));  % Signal reconstruit avec filtre

% Normalisation des signaux pour une comparaison équitable
x_RSFnormalise = real(x_RSF) / max(abs(x_RSF));
x_RAFnormalise = real(x_RAF) / max(abs(x_RAF));

% Tracé des signaux
figure;
plot(t, x); hold on; % Signal original
plot(t, x_RSFnormalise);hold on; % Signal sans filtre
plot(t, x_RAFnormalise,'--'); % Signal avec filtre
title('Comparaison des signaux reconstruits');
xlabel('Temps (s)');
ylabel('Amplitude (normalisée)');
legend('Signal original', 'Signal sans filtre', 'Signal avec filtre');
grid on;
```



Question ouverte : restituer au mieux un signal avec un détecteur limité

Question Bonus :

Le signal à étudier est du désormais multifréquence du type :

$$x(t) = A_1 \sin(2\pi f_1 t) + B_2 \sin(2\pi f_2 t) + B_3 \sin(2\pi f_3 t)$$

avec A1=1 B2=0.5 et B3=0.2 et f1=2000 f2=4000 et f3=6000 Hz.

Le détecteur utilisé pour l'acquérir a une bande passante de fe=9000 Hz.

A l'aide de la démarche présentée dans ce cours, restituer au mieux le signal initial.