Nom & Prénom de l'étudiant : ……
Classe : 4IoSys

Mixed & analogue Signal Processing

Atelier 1

Objectifs :

1. Manipuler les outils de base de Matlab pour l'affichage des signaux
2. Manipuler les outils de base de Matlab de calcul
3. Etudier les caractéristiques des signaux : moyenne temporelle, Variance, écart type.
4. Simuler sous Matlab le process de numérisation d'un signal sinusoïdal
5. Tracer et analyser un signal sinusoïdal échantillonné
6. Varier la fréquence d'échantillonnage d'un signal
7. Appliquer le théorème d'échantillonnage : Théorème de Shannon
8. Tracer, manipuler et analyser des signaux audios
9. Tracer et analyser un signal quantifié
10. Evaluer la performance de la numérisation des signaux audios

A rendre:

- Document Atelier 1 avec les figures des courbes obtenues et bien commentées

Partie 1 : Echantillonnage d'un signal périodique

On souhaite simuler l'échantillonnage d'un signal sinusoïdal avec différentes valeurs de fréquence d'échantillonnage.

Pour générer et afficher un signal sinusoïdal, exécuter le code Matlab suivant :

```
clear all
close all

D=1; % duration of simulation in second
F=10 % Signal Frequency in Khz
t=0 :1/1000 :D; % Time vector
Signal=sin(2*pi*F*t); % signal calculation

subplot(3,1,1);
plot(t,Signal),hold on; grid on; title('Original analogue signal')
```

Répondre à ces questions :

1) Quel est le type de ce signal
   a. Classification temporelle

……

   b. Classification énergétique

……

2) Quels sont les paramètres de ce signal
   a. Fréquence : ……
   b. Période : ……
   c. Moyenne : ……

Faire l'échantillonnage en utilisant les fréquences :

a) F1= 3*F ,

```
Fe = 30; % Sampling frequency in Hz
Te = 1/Fe
t1=0 :1/Fe :D;
Signal_sampled=sin(2*pi*F*t1);
subplot(3,1,2);plot(t1,Signal_sampled);
```

Afficher le résultat obtenu.

Insérer ici le résultat obtenu

Comparer et commenter le résultat

……

Quelle caractéristique de signal est perdue lors de l'échantillonnage ? (périodicité, valeur crète …)

……

b) F2=1.5*F.

Changer la valeur de la fréquence d'échantillonnage.

Afficher le résultat obtenu.

Insérer ici le résultat obtenu

……

……

……

……

Partie 2 : Echantillonnage d'un signal composite

On souhaite maintenant vérifier le théorème de Shannon en utilisant un signal analogique composé.

Exécuter le code suivant :

Bande de fréquences de ce signal : ……

Fréquence maximale de ce signal : ……

Fréquence limite d'échantillonnage : ……

```
clear all
close all
F=160000;
Fe = 16000 ; % Fréquence d'échantillonnage
F1 = 400 ;
F2 = 2000 ;
F3 = 5000 ;
M=1024;
compressing_factor=2;
% Calcul des échantillons
t=0:1/F:M/F ;
x_original=sin(2*pi*F1*t)+0.7*sin(2*pi*F2*t)+0.4*sin(2*pi*F3*t) ;
figure(1)
subplot(3,1,1);
plot((1:1:1000),x_original(1:1:1000)), hold on;
title('Signal Sampling');legend('original signal');

……
```

Echantillonner et afficher le signal obtenu :

```
k=0 :1/Fe:M/Fe ;
x=sin(2*pi*F1*k)+0.7*sin(2*pi*F2*k)+0.4*sin(2*pi*F3*k) ;
plot((1:10:100*10),x(1:100),'-o'), hold on;
legend('original signal','Sampled signal')
```

Insérer ici le résultat obtenu

Compresser le signal avec un facteur de 2 puis de 4.

```
y=downsample(x,compressing_factor);
plot((1:10*compressing_factor:1000),y(1:100/compressing_factor),'
-o'),hold on;
legend('original signal','compressed signal')
```

Insérer ici le résultat obtenu

Afficher, Comparer et Commenter

……

……

……

……

Partie 3 : Echantillonnage d'un signal audio

Exécuter le code Matlab suivant :

```
close all
fs = 8000;
tmax = 4;
nbits = 8;
nchan = 1;
Recorder = audiorecorder(fs, nbits, nchan);
record(Recorder);
fprintf(1,' Recording . . . \n ');
pause(tmax);
stop(Recorder);
% convert to floating - point vector and play back
yi = getaudiodata(Recorder, 'int16' );
y = double(yi);
y = y/ max( abs(y));
plot(y);
sound(y, fs);
```

1. Echantillonner votre voix en variant la valeur de Fs et nbits.
   Commenter

……

……

……

……

2. Enregistrer des séquences audio en utilisant la commande :

```
audiowrite('TP2record.wav',y,fs);
```

Insérer ici le résultat obtenu

Ajouter cette partie de code et exécuter le code Matlab suivant :

```
% quantization levels
t=0:(1/fs):tmax;
t=t(1:end-1);
n=4;
y_max=max(y);
y_min=min(y);
yquan=y/y_max(1);
```

1. Quantifier votre voix en variant la valeur de nombre des niveaux de quantification Commenter.

……

……

Insérer ici le résultat obtenu