Question exam TDS TP

# Examen 1 Traitement du signal TP

1. **ouvrir un fichier WAV contenant un son avec des parasites.**

>> [alien,Fs,nbits] = wavread('alien\_bruit1.wav');

>> sptool;



1. **filtrer le son dans Matlab**

Le parasite est entre 6 et 15 +-…

debut = 6\*Fs;

fin = 15\*Fs;

alienBRUIT = alien(debut:fin);



En 2500 y a un parasite…

Fc = 2500;

Wn1 = (Fc-1)/(Fs/2);

Wn2 = (Fc+1)/(Fs/2);

Wn = [Wn1 Wn2];

[b,a]= butter(1,Wn,'stop') ; %on met un coupe bande

alienBRUIT = filter(b,a,alienBRUIT);

sound(ans); %son du signal atténué

alien(debut:fin) = alienBRUIT ;

**Résumé du programme :**

|  |
| --- |
| [alien,Fs,nbits] = wavread('alien\_bruit1.wav');    %>> sptool; % on localise ou est le parasite, entre 6 et 15 seconde    debut = 6\*Fs;  fin = 15\*Fs;  alienBRUIT = alien(debut:fin);    %>> sptool; % spectre de alienBRUIT, on voit un parasite en 2500    Fc = 2500;  Wn1 = (Fc-1)/(Fs/2);  Wn2 = (Fc+1)/(Fs/2);  Wn = [Wn1 Wn2];  [b,a]= butter(1,Wn,'stop') ; %on met un coupe bande  alienBRUIT = filter(b,a,alienBRUIT);  sound(alienBRUIT); %son du signal atténué  alien2 = alien; %garder le son de base pour après comparer…  alien2(debut:fin) = alienBRUIT ; % réinsere alien bruit dans le son de base    sound(alien2,Fs) ; |

(frequence de coupure -3db et puis pente de -6 db \* ordre filtre)

**3) analyser le spectre du signal de début et le signal filtré**

Analyse dans sptool ou bien avec des fft fftshift et plot

|  |
| --- |
| sptool %prendre le temps du son  axeFqce = 0:1/24.538375:Fs; %entre 0 et Fs, si c’était entre 0 et  %Fs/2 il faudrait  %créer un nouveau vecteur adapté qui  %serait  %v=alien(Fs\*(duree/2):Fs\*duree);  subplot(2,1,1);  spectre =fftshift(fft(alien));  plot(axeFqce,abs(spectre));    subplot(2,1,2);  spectre2 =fftshift(fft(alien2));  plot(axeFqce,abs(spectre2)); |

**La différence entre le spectre de sptool et celui-là est que le spectre de sptool affiche le spectre de la puissance.**

**4) refaire le point 2 avec sptool**

Facile, on choisit les mêmes options que dans le filtre question 2 et c’est bon…

**5) Image**Une image fond blanc avec des formes géométriques de couleur, faut passer le tout en noir et blanc et après inverser les couleurs (ce n’est pas noté)…  
Il y a donc toute les formes qui sont présentent, faut labéliser, utiliser la fonction Exentricity de regionprop pour ne garder que les deux lignes…

|  |
| --- |
| im = imread('bwlabel\_exc.jpg');  im =~im2bw(im);  [im,n] = bwlabel(im);  R = regionprops(im,'Eccentricity');    idx = find([R.Eccentricity]>0.5);  BW2 = ismember(im,idx);    imshow(BW2); |

**6) image des fleurs, mettre en niveau de gris de deux manières : avec l'équation et rg2gray puis dire la différence entre les deux résultats**

|  |
| --- |
| clear all;close all;clc;    im = imread('flowers.tif');  im = im2double(im);  im2(:,:) = 0.3\*im(:,:,1)+0.3\*im(:,:,2)+0.3\*im(:,:,3);  subplot(2,1,1);  imshow(im2);    im3 = rgb2gray(im);  subplot(2,1,2);  imshow(im3); |

**Imprécis niveau luminance mais je pense pas que dewulf va faire chier :p…**

**7) enlever les points rouges d’une image**

img(:,:,1)=0;

# Examen 2 Traitement du signal TP

1. **J'ai eu la même question que l'an dernier avec le cameraman\_bruit ou il fallait faire un filtre moyenneur matrice 3x3 puis faire une fonction avec un "seuil" qui ne filtre que là où il y a des points.**

|  |
| --- |
| %1. J'ai eu la même question que l'an dernier avec le cameraman\_bruit ou il  %fallait faire un filtre moyenneur (filtre utilisant la convolution) matrice 3 puis faire une fonction avec un  %"seuil" qui ne filtre que là où il y a des points.    clc;  clear all;  close all;    im = imread('CAMERAMAN\_BRUIT.tif');  im = im2double(im);  subplot(2,1,1);  imshow(im);    im2 =imfilter(im,[1/9 1/9 1/9;1/9 1/9 1/9;1/9 1/9 1/9]);  subplot(2,1,2);  imshow(im2); |

**Fonction seuil :**

|  |
| --- |
| im = imread('CAMERAMAN\_BRUIT.tif');  im = im2double(im);  subplot(2,1,1);  imshow(im);    im2 = im;  [x y] = size(im);  for i=2:x-1  for j=2:y-1  m =(im(i+1,j+1)+im(i+1,j-1)+im(i-1,j-1)+im(i-1,j+1)+im(i+1,j)+ ...  im(i,j+1)+im(i-1,j)+im(i,j-1))/8;  if ((m-im(i,j)>0.15)||(im(i,j)-m)>0.15)  im2(i,j) = m;  end  end  end    imshow(im2); |

==> **si** la **différence** de niveau de gris entre la **moyenne** sur les pixels entourant chaque pixel **et** la valeur de ce **pixel** est **plus grande** que **0.2** on **remplace** le **pixel** par cette **moyenne**.

(0.2 à été retrouvé après quelques test, 0.25 est en fait plus précis ! (mais flemme de redécoupé l’image…)).

**Je montre le meme avec le filter median :**

|  |
| --- |
| im = imread('CAMERAMAN\_BRUIT.tif');  im = im2double(im);  subplot(2,1,1);  imshow(im);    im3 = medfilt2(im,[3 3]);  subplot(2,1,2);  imshow(im2); |

L’image est lissée, les points ont disparut mais perd certain de ces détails.

****

**2. creer un battement avec deux sons de 440 et 500Hz, le voir, l'enregistrer, l'entendre.  
    filtre de maniere à enlever le battement (ordre max =4 ) avec sptool et avec filter.**

|  |
| --- |
| clear all;  close all;  f = 440;  f2 = 500;  Fe = 22000; %fréquence d'échantillonage  Te=1/Fe; %période d'échantillonage  t = 0 :Te:4; %intervalle de 0 à 4s, avec une période  %d’échantillonnage de 1/Fe  A = 0.5; %amplitude  %1er signal  y = A\*sin(2\*pi\*f\*t); %y signal périodique  %2ème signal  y2 = A\*sin(2\*pi\*f2\*t);  x = (y+y2);  Fc = 500;  Wn1 = (Fc-1)/(Fe/2);  Wn2 = (Fc+1)/(Fe/2);  Wn = [Wn1 Wn2];  [b,a]=butter(1,Wn, 'stop') ;  filter(b,a,x);  sound(x,Fe); %son du signal  plot(t,x);  pause(5);  sound(ans,Fe); %son du signal atténué  plot(t,ans); |

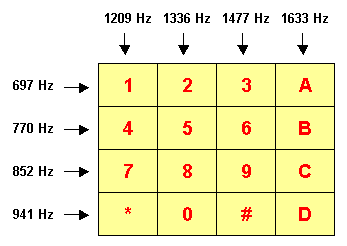
# Examen 3 Traitement du signal TP

**1) Ouvrir une image : "pleiades\_i.jpg " se trouvant sur le réseau, représentant un champs d'étoiles, il faut pourvoir isoler les étoiles les plus brillantes.**

|  |
| --- |
| % Isoler les étoiles les plus brillante  clear all;  close all;  clc;    img = imread('pleiades\_i.jpg');  imshow(img)  img = im2bw(img,graythresh(img));  [img,n] = bwlabeln(img); %labeliser l'image  stats =regionprops(img,'Area'); %région de l'image en Area  %(surface de la région)  idx = find([stats.Area] > 100); %trouver les area>100(surface de px)  BW2 = ismember(img,idx); %ismember créera une copie de la matrice  %img et ne gardera que les éléments  %compris dans idx de cette matrice  figure, imshow(BW2) |

**2) Ouvrir un fichier son : "dtmf.wav", en sachant que les touches dtmf sont composées d'une fréquence ligne additionnée à une fréquence colonne, déterminer à partir du fichier son sur quelles touches on a appuyé. Il nous donne un tableau des fréquences DTMF pour nous aider.**

**Utiliser specgram (specrogram) pour voir les fréquences en fonction du temps.**



**Méthode specgram**

|  |
| --- |
| clear all;  close all;  clc;    [dtmf,fs]=wavread('dtmf5.wav');    specgram(dtmf,2048,fs); %2048 contraste de couleur |

**Méthode plus précise**

|  |
| --- |
| clear all;  close all;  clc;    [dtmf,fs]=wavread('dtmf1.wav');    duree = (length(dtmf)-1)/fs; %diviser le nbre échantillons par la Fe pour obtenir la  %duréé arrondie donc fausse le résultat  %duree = 1.3999375; %la choper dans sptool    y=fftshift(fft(dtmf));  axeF=0:1/duree:fs/2; %axe des fréquences    v=y(fs\*(duree/2):fs\*duree);  plot(axeF,abs(v)); |

Il s’agit donc d’un 1.

Maintenant dtmf5

|  |
| --- |
| clear all;  close all;  clc;    [dtmf,fs]=wavread('dtmf5.wav');    duree = (length(dtmf)-1)/fs; %diviser le nbre échantillons par la Fe pour obtenir la  %duree = 1.3999375; %la choper dans sptool    y=fftshift(fft(dtmf));  axeF=0:1/duree:fs/2; %axe des fréquences    v=y(fs\*(duree/2):fs\*duree);  plot(axeF,abs(v)); |

Il s’agit d’un 5.

1. **Faire un signal composé de 10 sinus de 2 secondes chacun (gamme musicale). La fréquence du premier sinus est de 150hz, les suivantes sont les octaves (300, 600,...). La fréquence d'échantillonage est de 8000Hz. Il faut prédire les fréquences fantomes qui apparaissent et prouver avec Matlab.**

|  |
| --- |
| clear all;  close all;  clc;    fe = 8000;  t = 0:1/fe:2;  f=150;    signal = [];  for i=1:10  s = sin(2\*pi\*f\*t);  signal = [signal s];  f= f\*2;  end    sound(signal,fe);    %Foirage à la 6eme octave, non respect de shannon  % 150 \* 2^5 = 4800 ==> 2fmax >8000  %(2\*4800 < 8000)  %voir dans sptool |

1. **Enlever des points d'une image avec bwlabel, ismember, regionprops.**

**On va faire ça avec "pleiades\_i.jpg " et enlever les images les plus grande/brillante.**

**Si on nous demande de supprimer des petits points sur une image :**

|  |
| --- |
| % Isoler les étoiles les plus brillante  clear all;  close all;  clc;    img = imread('pleiades\_i.jpg');  imshow(img)  img = im2bw(img,graythresh(img));  [img,n] = bwlabeln(img); %labeliser l'image  stats =regionprops(img,'Area'); %région de l'image en Area  %(surface de la région)  idx = find([stats.Area] > 100); %trouver les area>100(surface de px)  BW2 = ismember(img,idx); %ismember créera une copie de la matrice  %img et ne gardera que les éléments  %compris dans idx de cette matrice  figure, imshow(BW2) |

(On ne garde que les gros)

**BONUS :** **bwareaopen(img,nbpixel)** fait la même chose en une ligne pour supprimer les petits points en enlevant des points blancs sur un fond noir.

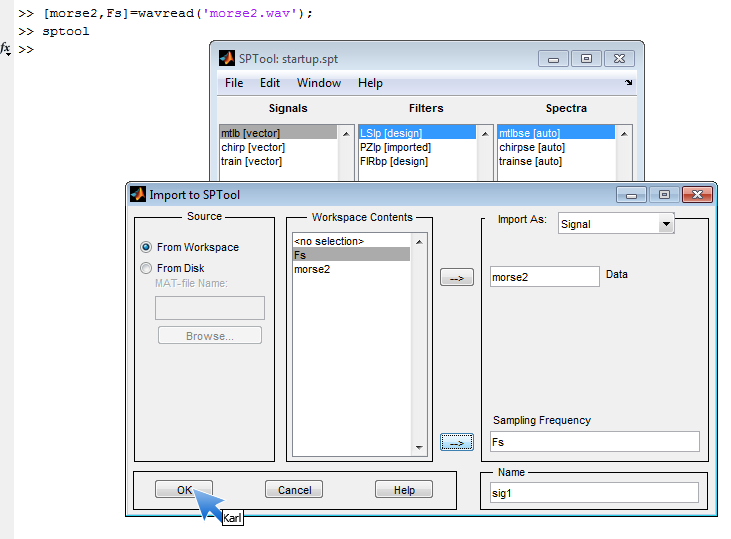
**Si on nous demande de supprimer des gros éléments :**

|  |
| --- |
| % Isoler les étoiles les plus brillante  clear all;  close all;  clc;  img = imread('pleiades\_i.jpg');  %région de l'image en Area  %(surface de la région)  idx = find([stats.Area] > 10); %trouver les area>100(surface de px)  BW2 = ismember(img,idx); %ismember créera une copie de la matrice  %img et ne gardera que les éléments  %compris dans idx de cette matrice  figure, imshow(BW2) |

1. **On dispose d'un fichier genre morse.wav avec plein de parasites.  
   Le but est de tout filtré à l'aide SPTool et sans SPTool ;  
   Bien entendu chaque résultat devrait être affiché (spectre, signal, etc);  
   Il faudra donc utilisé fft, Filter & Butter...**

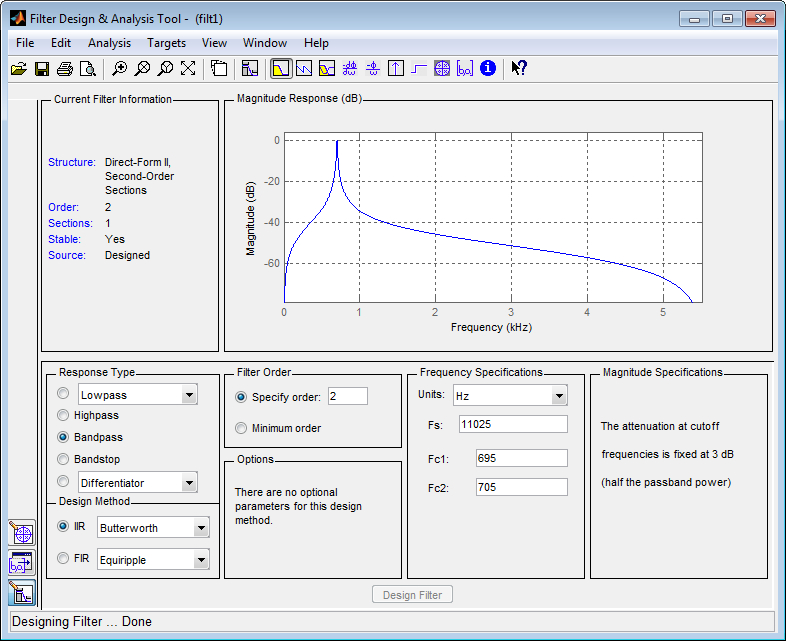
|  |
| --- |
| [morse2,Fs,nbits] = wavread('morse2.wav');    %>> sptool; % on localise ou est le parasite, entre 6 et 15 seconde    Fc = 680;  Wn = Fc/(Fs/2);    [b,a]= butter(2,Wn,'low') ; %on met un coupe bande  morsedebruite = filter(b,a,morse2);    Fc = 720;  Wn = Fc/(Fs/2);    [b,a]= butter(2,Wn,'high') ; %on met un coupe bande  morsedebruite = filter(b,a,morsedebruite);    sound(morsedebruite); |

Importer la matrice de morse2 dans sptool :

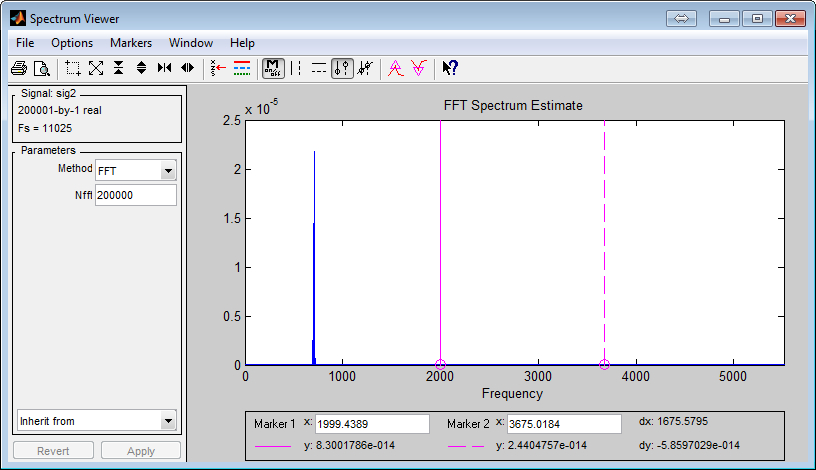


Ici nous souhaitons garder la Fréquence de 700Hz et atténuer tout le reste (bruit de fond de mer)

Créer le filtre « Bandstop » en méthode « Butterworth » d’ordre 2, Fs 11025, Fc1 695, Fc2 705



Voici le spectre final :



1. **Le son à filtrer était « comic\_bruit.wav », qui contenait deux fréquences parasites.**

**La solution était d'appliquer un filtre coupe-bande pour chaque fréquence parasite. Je devais réaliser le filtrage dans SpTool et dans un fichier m, donc avec butter et filter.**

**Il fallait aussi définir le type de parasite et le type de filtre à utiliser, en spécifiant l'ordre qu'on allait utiliser. Mr Ernotte avait l'air d'insister sur la bonne utilisation de specgram, aussi. Faites attentions à ça aussi Wink Je devais aussi faire une représentation fft du signal.**

# Examen 4 Traitement du signal TP

**1. Un fichier avec de la parole et on entendait un tuuuuuut (bruit)**  
Méthode la même que pour l'alien  ===> avec specgram(son,512,Fs)  ==> distinguer les fréquences parasite ici 400 et 950  +/-  
==> choisir le bon coupe bande ===> si c'est 400  prendre 300 - 500   et lui expliquer pq le choix de telles frequences (frequence de coupure -3db et puis pente de -6 db \* ordre filtre)

**2. Question sur les images**  
Carré de 300 Rouge

|  |
| --- |
| clc ;  clear all ;  im=imread('flowers.tif') ;  subplot(1,2,1);  imshow(im);  im(1:300,1:300,2:3)=0;  im(1:300,1:300,1)=255;  imshow(im); |

**3. Utiliser improvide un truc du style je me rappelle plus trop et detecter les bords avec Edge voila**

**Détecter des bords resortant :**

|  |
| --- |
| I=imread('pout.tif');  h=fspecial('sobel');  im = imfilter(I,h);  imshow(im); |

**Détecter les bords:**

|  |
| --- |
| I=imread('pout.tif');  edge(I,'canny'); |

|  |
| --- |
| I=imread('pout.tif');  edge(I,'sobel'); |