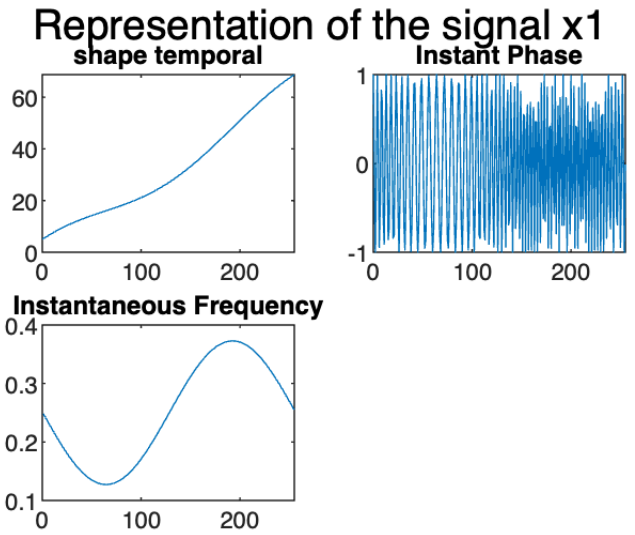
ARSIG : Time Frequency Representation

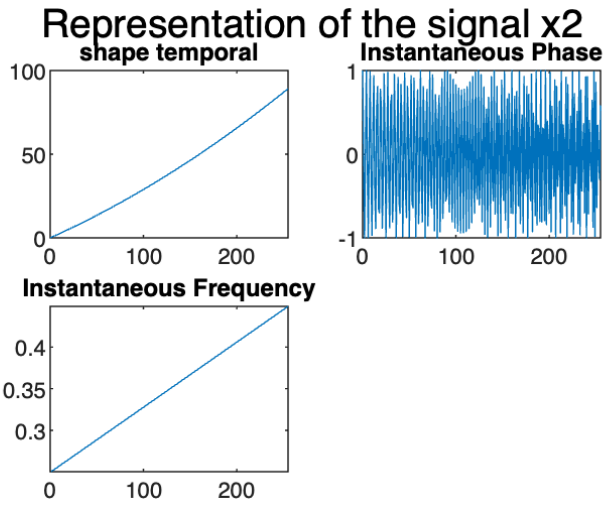
Supervisor : Sebastien Bourguignon
Name : ***Chaitanya Krishna VIRIYALA***

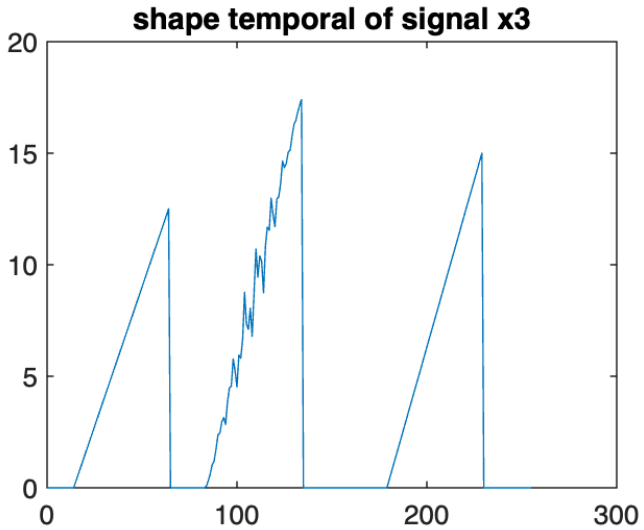
1. Representation in Time Frequency of the simulated signals

1.1 Generation of Signals

For these two mono-component signals x1 and x2, represent their temporal pace, their instantaneous phase and their instantaneous frequency as a function of time.



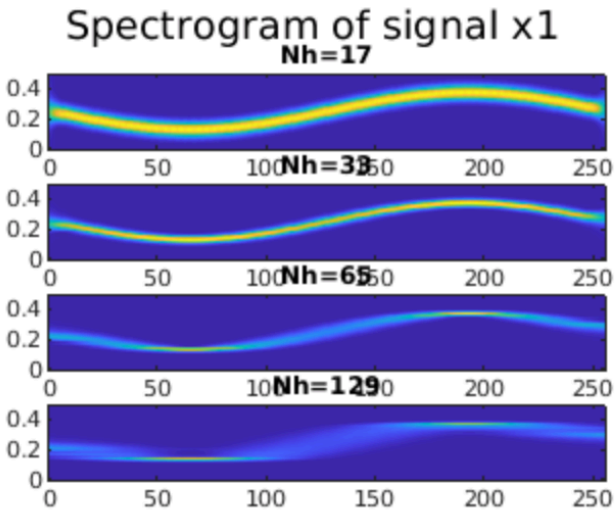


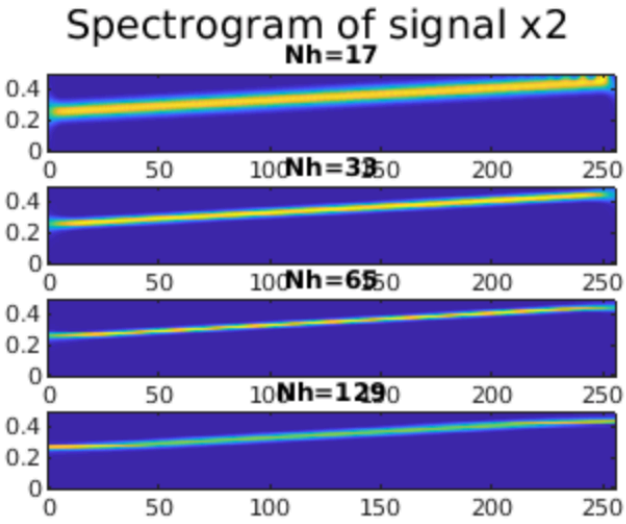


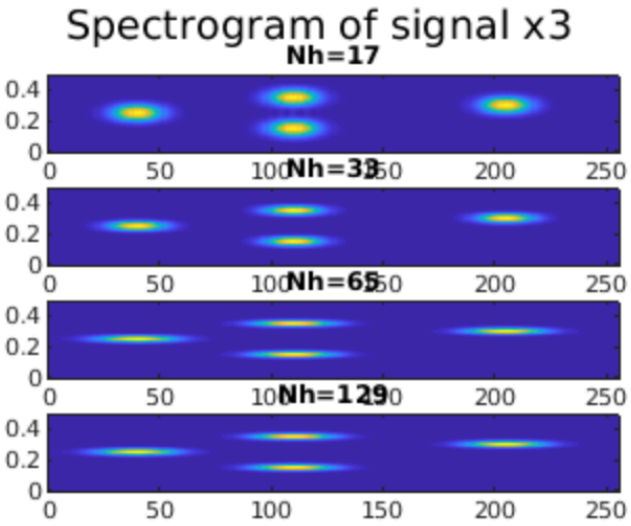
1.2 Time-Frequency Representation

1.2.1 , 1.2.2 and 1.2.3 Spectrogram of signal x1 , x2 and x3

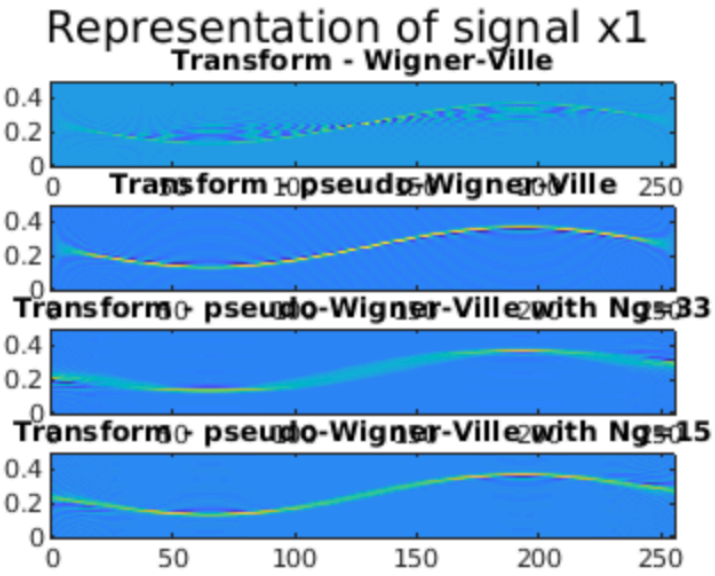
We display the spectrogram, using a Hamming window h of length Nh = 17, 33, 65 and 129. We display the four results in the same figure.

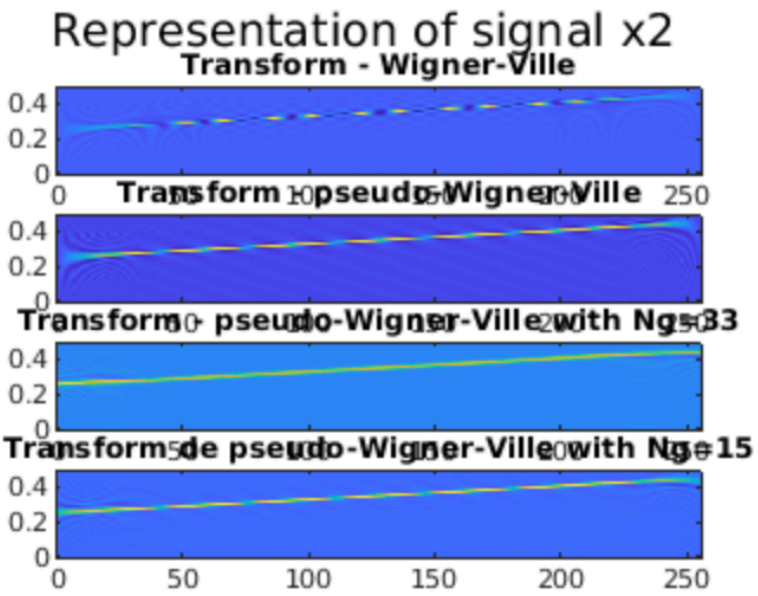


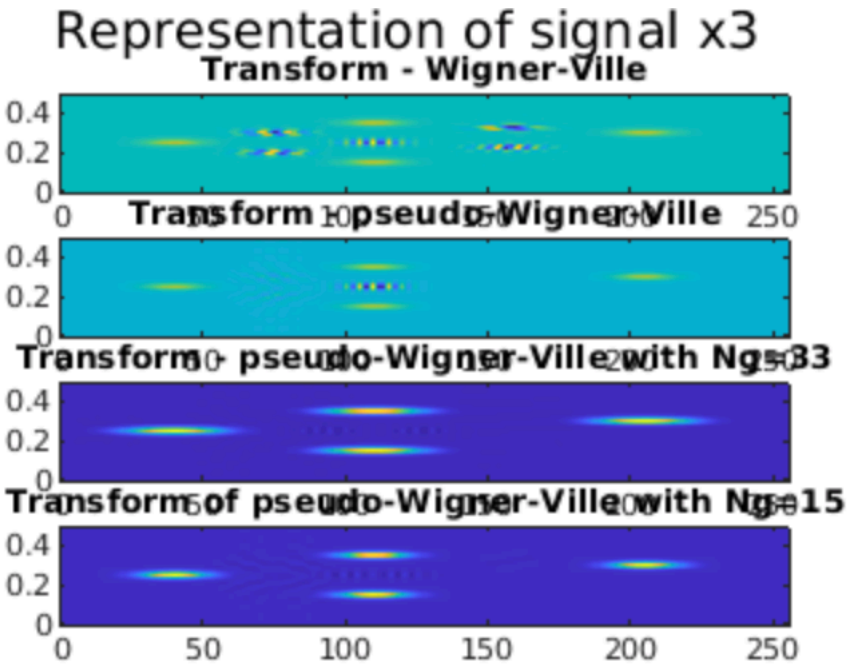




Note :
Using matlab online, so the figure is like this (as it doesn't pop up a separate window like the traditional version)


Representation of signal x1
Transform - Wigner-Ville
Transform pseudo-Wigner-Ville
Transform pseudo-Wigner-Ville with Ng=33
Transform pseudo-Wigner-Ville with Ng=15
0.4
0.2
0
0
50
100
150
200
250


Representation of signal x2
Transform - Wigner-Ville
Transform pseudo-Wigner-Ville
Transform pseudo-Wigner-Ville with Ng=33
Transform de pseudo-Wigner-Ville with Ng=15
0.4
0.2
0
0
50
100
150
200
250


Representation of signal x3
Transform - Wigner-Ville
Transform pseudo-Wigner-Ville
Transform pseudo-Wigner-Ville with Ng=33
Transform of pseudo-Wigner-Ville with Ng=15
0.4
0.2
0
0
50
100
150
200
250

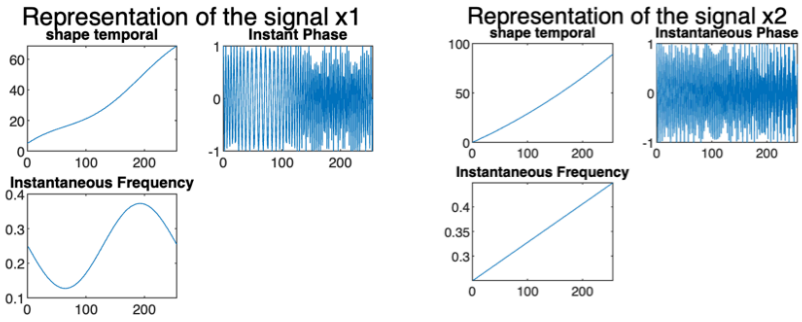
Comments :

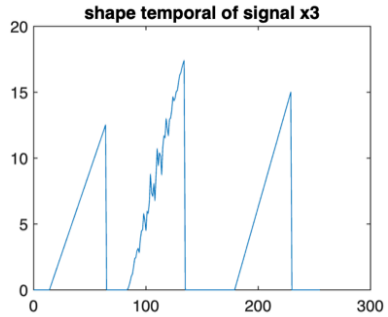
After having analyzed the results of the representations, while on board, for the single-component signals **x1 and x2**

- it is obvious that the **Wigner-Ville method** is the **best** in terms of the **energy** concentration of the signal.
- Because we do not add the windows in the calculation compared to the Wigner-Ville method in order to avoid the mutual constraint between the resolution of the time domain and the resolution of the frequency domain.
- But due to the windowless operation, when analyzing multicomponent signals, the Wigner-Ville method will be disturbed by mutual constraint. So, in **order to improve and solve** this problem, the **Pseudo-Wigner-Ville method smooths** it out by adding a **hamming window** function, so that the time-frequency resolution is fixed.
- Because the time-frequency resolution is only related to the window function. Moreover, the **Pseudo-Wigner-Ville smooth method adds 2 window functions**, **hamming and kaiser** respectively. We have found that when we set the width of the kaiser window smaller (Ng = 15), the result is better.

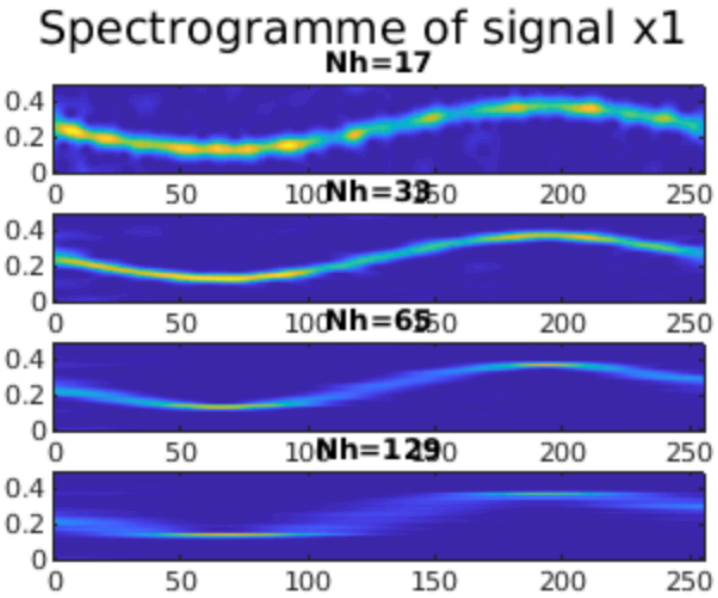
1.3 Influence of Noise

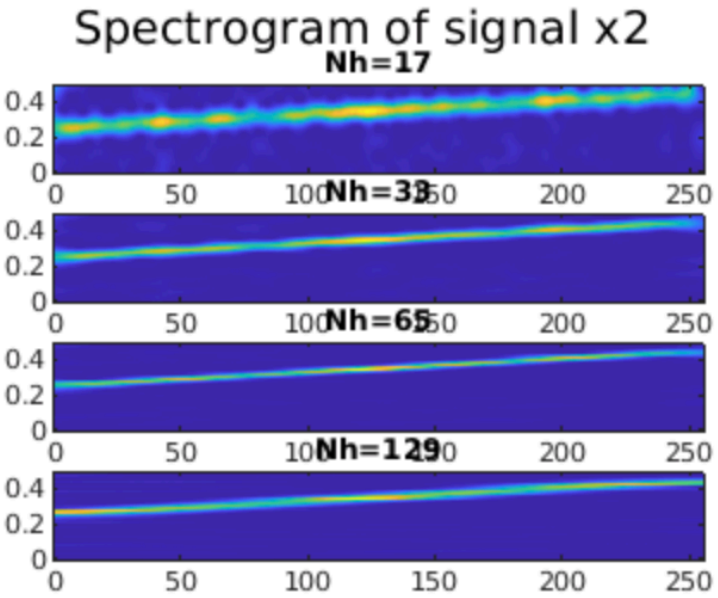
1.3.1 Temporal Signal

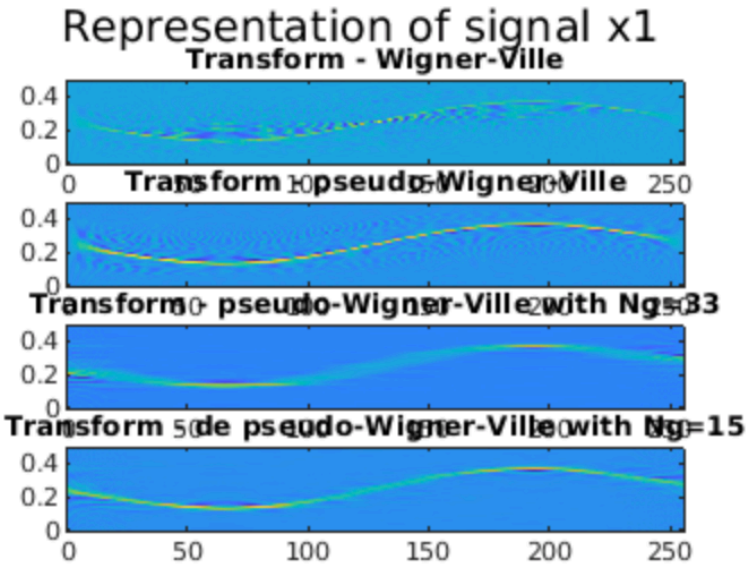


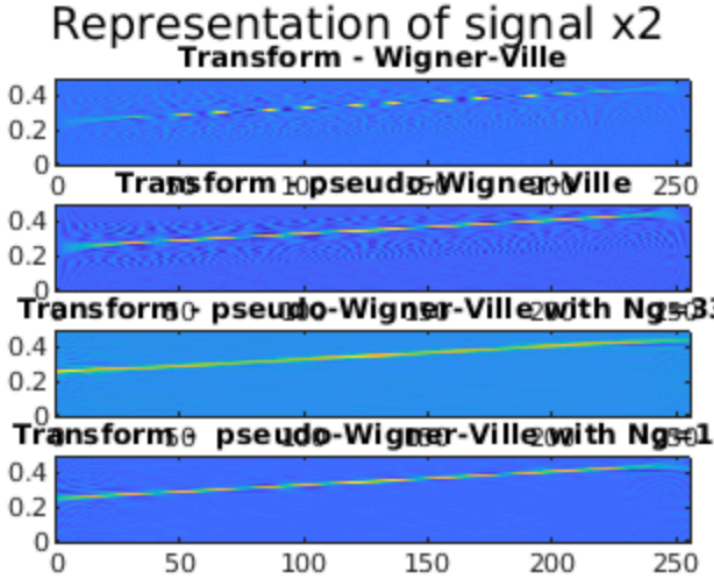


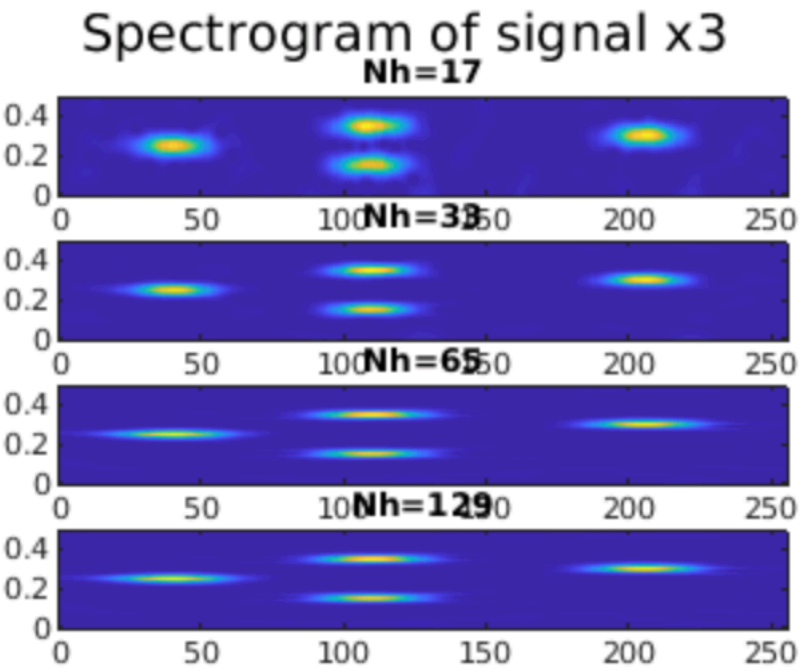
1.3.2 - 1.3.7 Spectogram of signals and their representations

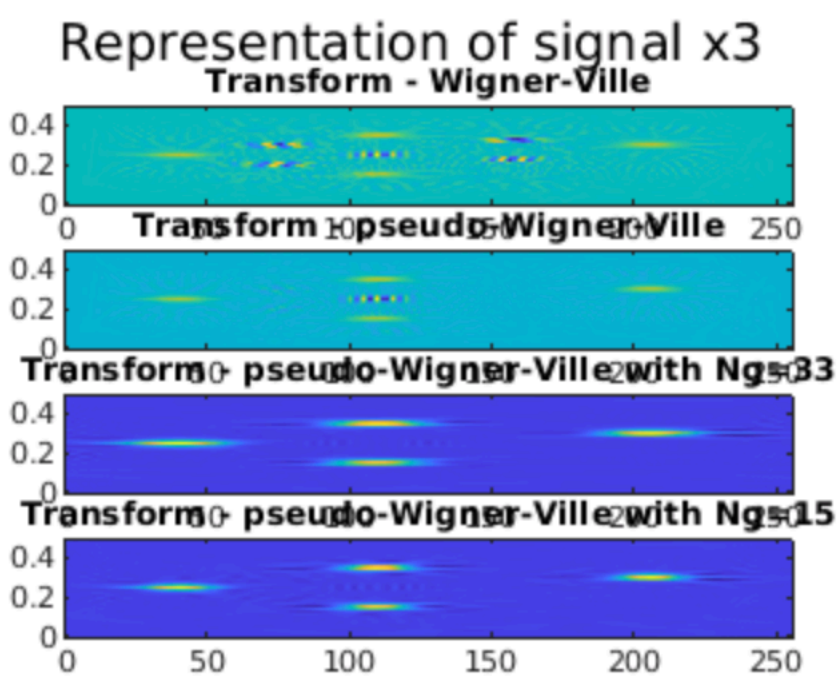









Spectrogram of signal x3
Nh=17
Nh=33
Nh=65
Nh=129
0.4
0.2
0
0
50
100
150
200
250


Representation of signal x3
Transform - Wigner-Ville
Transform - pseudo-Wigner-Ville
Transform - pseudo-Wigner-Ville with Ng=33
Transform - pseudo-Wigner-Ville with Ng=15
0.4
0.2
0
0
50
100
150
200
250

1.3.8 Results

After observing the temporal signals, compared to the spectrograms of three signals :

- it is evident that when the **width of the selected hamming window** function is larger, their robustness to **noise** is **stronger.**

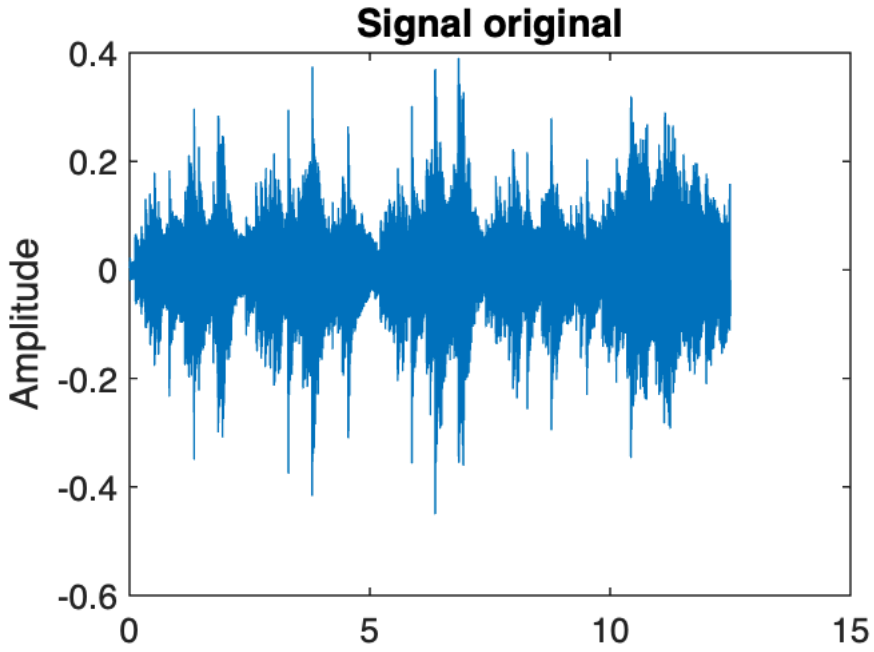
- For representations of three transforms, the **smoothed pseudo-Wigner-Ville method** has the **strongest robustness** when comparing the other methods. In addition, when the width of the selected kaiser window function is smaller, their noise robustness is stronger.

Spectogram with Nh = 129 has the highest robustness to noise>

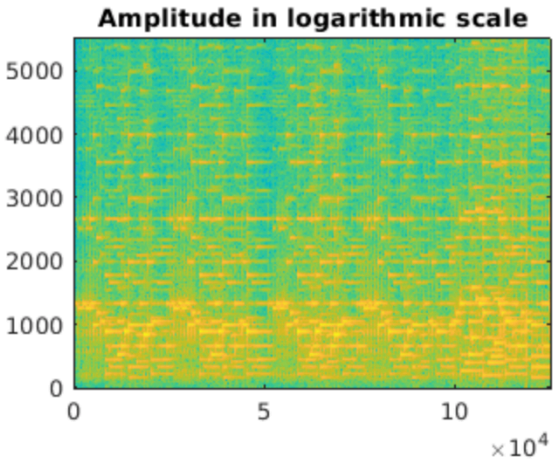
In the representaions of transforms, smoothed Pseudo Wigner Ville with Ng = 15 has the highest robustness towards noise.

2. Detection and reconstruction of a musical score (number 3 in the TP)

3.2.2 Load the signal and view it graphically



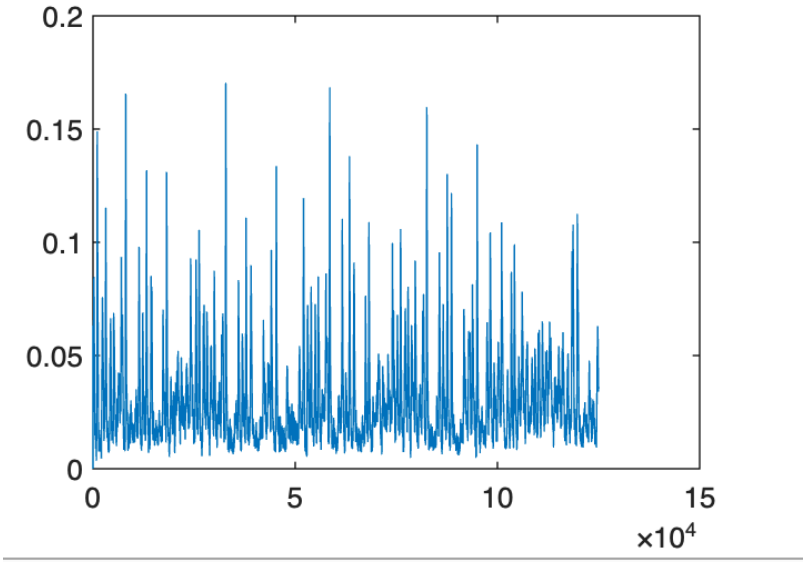
With the spectrogram in logarithmic scale,
- we see that the main notes over time and frequency are well arranged in the image and we can observe the time duration approximately of each note along the horizontal axis.



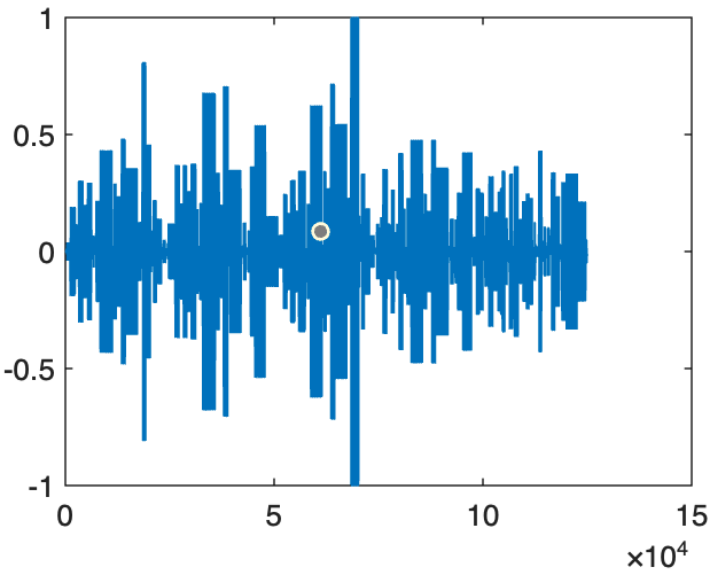
Check the pattern of yellow, quite interesting because these are the ones which vary the notes and which allow us to hear the very soft and very elegant notes.

3.2.3

We display the image of the Kolmogorov distance, from this image we can know that the number of peaks correspond to the number of note changes. Since the minimum peak is almost 0.037, we take 0.035 as the threshold given to detect peaks.

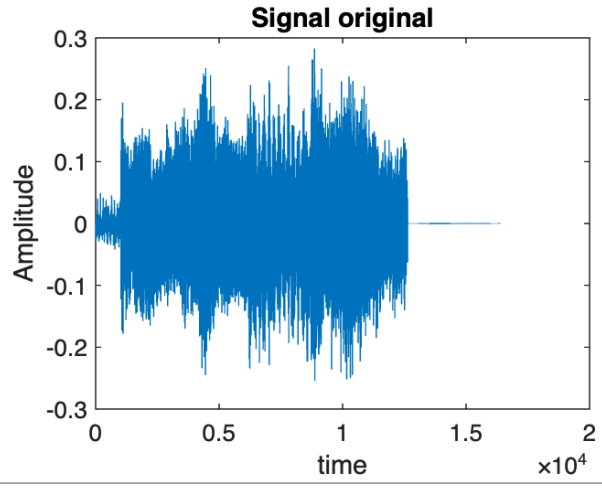


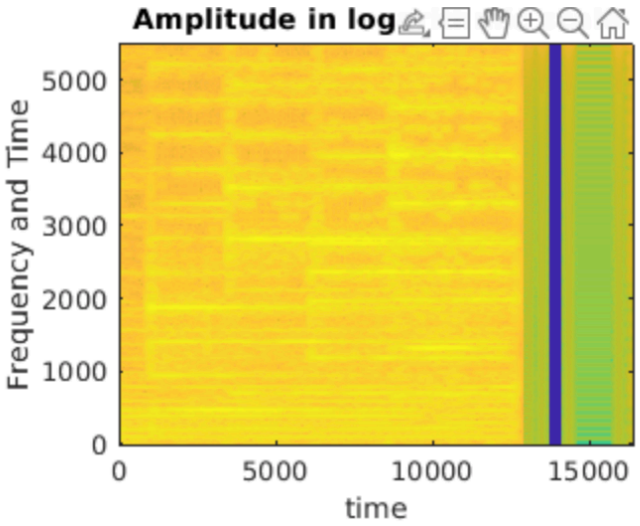
3.2.5

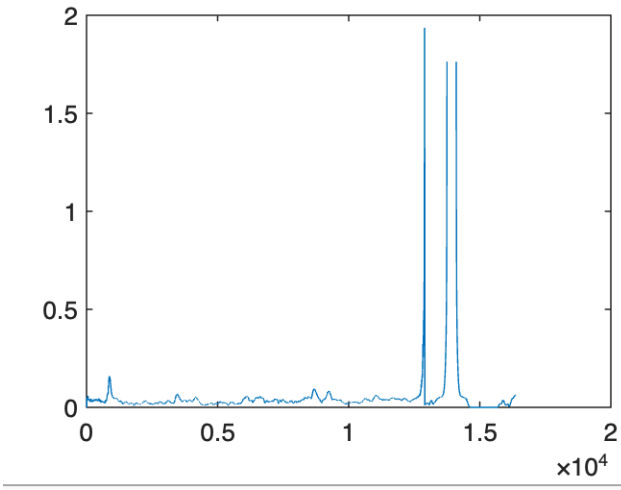


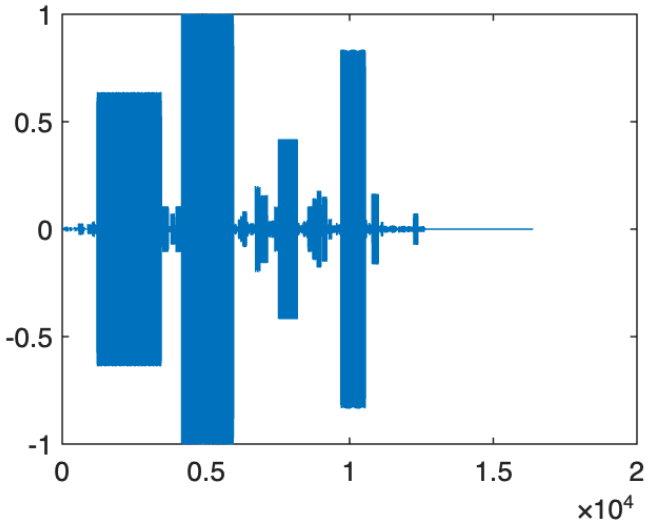
Observation : The Resynthesized signal has the **same notes** and **lengths** as the original signal, but it seems to us to be stiff and starched.

3.2.6 BONUS









Observations:

The above steps were replicated but for a different input PinkFloyd,

- although **some echoes** or reverberations may be heard in the original signal, there is **no information** about it in the re-synthesized signal.
- That is to say that **we will lose some information** during this process.

ANNEXE :

```
clc
clear all
close all
addpath(genpath('tftb-0.2'));
%% Question 2.1: Generation of signal
N=256;
n=linspace(0,N-1,N);   %N=256 et n=0,...,N-1
phi1=0.25*n+5*cos(2*pi*n/N);
phi2=0.25*n+0.1*(n.^2)/N;
x1=cos(2*pi*phi1);
x2=cos(2*pi*phi2);
f1=diff(phi1)./diff(n);       % frequence: d(phi)/d(t)
f2=diff(phi2)./diff(n);

T1 = 15; T2 = 85; T3 = 180;
Nh = 51; th = (0:Nh-1); h = gausswin(Nh)';
x3 = zeros(1,N);
x3(T1:T1+Nh-1) = h.*cos(2*pi*0.25*th);
x3(T2:T2+Nh-1) = h.*cos(2*pi*0.15*th);
x3(T2:T2+Nh-1) = x3(T2:T2+Nh-1) + h.*cos(2*pi*0.35*th);
x3(T3:T3+Nh-1) = h.*cos(2*pi*0.3*th);
%representation of the signal x1
subplot(2,2,1);
plot(n,phi1);
title('Allure temporal');
subplot(2,2,2);
plot(n,x1);       %phase
title('Instant Phase');
subplot(2,2,3);
plot(n(2:N),f1);    % frequency
title('Instantaneous Frequency');
suptitle('Representation of the signal x1');


%representation of the signal x2
figure(2);
subplot(2,2,1);
plot(n,phi2);
title('Allure temporal');
```

```
subplot(2,2,2);
plot(n,x2);       %phase
title('Instantaneous Phase');
subplot(2,2,3);
plot(n(2:N),f2);    % frequency
title('Instantaneous Frequency');
suptitle('Representation of the signal x2');


%representation of the signal x1
phi3 = zeros(1,N);
phi3(T1:T1+Nh-1) = 0.25*th;
phi3(T2:T2+Nh-1) = 0.15*th;
phi3(T2:T2+Nh-1) = x3(T2:T2+Nh-1) +0.35*th;
phi3(T3:T3+Nh-1) = 0.3*th;
figure(3);
plot(n,phi3);
title('Allure temporal of signal x3');



%% Question 2.2: Time-Frequency representation
% Spectrogram of signal x1
Nh1=17;
Nh2=33;
Nh3=65;
Nh4=129;
h1= hamming(Nh1);
[tfrx11,t11,f11] = tftb_spectrogram(x1,N,h1);
h2= hamming(Nh2);
[tfrx12,t12,f12] = tftb_spectrogram(x1,N,h2);
h3= hamming(Nh3);
[tfrx13,t13,f13] = tftb_spectrogram(x1,N,h3);
h4= hamming(Nh4);
[tfrx14,t14,f14] = tftb_spectrogram(x1,N,h4);
figure(4)
subplot(4,1,1);
imagesc(t11,f11,tfrx11); axis xy;
title('Nh=17');
subplot(4,1,2);
imagesc(t12,f12,tfrx12); axis xy;
title('Nh=33');
subplot(4,1,3);
imagesc(t13,f13,tfrx13); axis xy;
title('Nh=65');
subplot(4,1,4);
imagesc(t14,f14,tfrx14); axis xy;
```

```
title('Nh=129');
suptitle('Spectrogram of signal x1');


% Spectrogram of signal x2
[tfrx21,t21,f21] = tftb_spectrogram(x2,N,h1);
[tfrx22,t22,f22] = tftb_spectrogram(x2,N,h2);
[tfrx23,t23,f23] = tftb_spectrogram(x2,N,h3);
[tfrx24,t24,f24] = tftb_spectrogram(x2,N,h4);
figure(5)
subplot(4,1,1);
imagesc(t21,f21,tfrx21); axis xy;
title('Nh=17');
subplot(4,1,2);
imagesc(t22,f22,tfrx22); axis xy;
title('Nh=33');
subplot(4,1,3);
imagesc(t23,f23,tfrx23); axis xy;
title('Nh=65');
subplot(4,1,4);
imagesc(t24,f24,tfrx24); axis xy;
title('Nh=129');
suptitle('Spectrogram of signal x2');
% Spectrogram of signal x3
[tfrx31,t31,f31] = tftb_spectrogram(x3,N,h1);
[tfrx32,t32,f32] = tftb_spectrogram(x3,N,h2);
[tfrx33,t33,f33] = tftb_spectrogram(x3,N,h3);
[tfrx34,t34,f34] = tftb_spectrogram(x3,N,h3);
figure(6)
subplot(4,1,1);
imagesc(t31,f31,tfrx31); axis xy;
title('Nh=17');
subplot(4,1,2);
imagesc(t32,f32,tfrx32); axis xy;
title('Nh=33');
subplot(4,1,3);
imagesc(t33,f33,tfrx33); axis xy;
title('Nh=65');
subplot(4,1,4);
imagesc(t34,f34,tfrx34); axis xy;
title('Nh=129');
suptitle('Spectrogram of signal x3');


% Representation of the signal x1
[tfrx1_1,t1_1,f1_1] = tftb_wvd(x1,N);
beta=2.5;
L=63;
```

```
h_pwv=kaiser(L,beta);
[tfrx1,t1,f1] = tftb_pwvd(x1,N,h_pwv);


Ng1=33;
Nhp=63;
Ng2=15;
h_spwv=kaiser(Nhp,beta);
g1=kaiser(Ng1,beta);
g2=kaiser(Ng2,beta);


[tfrx_11,t_11,f_11] = tftb_spwvd(x1,N,g1,h_spwv);
[tfrx_21,t_21,f_21] = tftb_spwvd(x1,N,g2,h_spwv);


figure(7)
subplot(4,1,1);
imagesc(t1_1,f1_1,tfrx1_1); axis xy;
title('Transform - Wigner-Ville');
subplot(4,1,2);
imagesc(t1,f1,tfrx1); axis xy;
title('Transform - pseudo-Wigner-Ville');
subplot(4,1,3);
imagesc(t_11,f_11,tfrx_11); axis xy;
title('Transform - pseudo-Wigner-Ville with Ng=33');
subplot(4,1,4);
imagesc(t_21,f_21,tfrx_21); axis xy;
title('Transform - pseudo-Wigner-Ville with Ng=15');
suptitle('Representation of signal x1');


% Representation of signal x2
[tfrx2_1,t2_1,f2_1] = tftb_wvd(x2,N);
[tfrx2,t2,f2] = tftb_pwvd(x2,N,h_pwv);
[tfrx_12,t_12,f_12] = tftb_spwvd(x2,N,g1,h_spwv);
[tfrx_22,t_22,f_22] = tftb_spwvd(x2,N,g2,h_spwv);


figure(8);
subplot(4,1,1);
imagesc(t2_1,f2_1,tfrx2_1); axis xy;
title('Transform - Wigner-Ville');
subplot(4,1,2);
imagesc(t2,f2,tfrx2); axis xy;
title('Transform - pseudo-Wigner-Ville');
subplot(4,1,3);
imagesc(t_12,f_12,tfrx_12); axis xy;
```

```
title('Transform - pseudo-Wigner-Ville with Ng=33');
subplot(4,1,4);
imagesc(t_22,f_22,tfrx_22); axis xy;
title('Transform de pseudo-Wigner-Ville with Ng=15');
suptitle('Representation of signal x2');


% Representation of signal x3
[tfrx3_1,t3_1,f3_1] = tftb_wvd(x3,N);
[tfrx3,t3,f3] = tftb_pwvd(x3,N,h_pwv);
[tfrx_13,t_13,f_13] = tftb_spwvd(x3,N,g1,h_spwv);
[tfrx_23,t_23,f_23] = tftb_spwvd(x3,N,g2,h_spwv);


figure(9);
subplot(4,1,1);
imagesc(t3_1,f3_1,tfrx3_1); axis xy;
title('Transform - Wigner-Ville');
subplot(4,1,2);
imagesc(t3,f3,tfrx3); axis xy;
title('Transform - pseudo-Wigner-Ville');
subplot(4,1,3);
imagesc(t_13,f_13,tfrx_13); axis xy;
title('Transform - pseudo-Wigner-Ville with Ng=33');
subplot(4,1,4);
imagesc(t_23,f_23,tfrx_23); axis xy;
title('Transform of pseudo-Wigner-Ville with Ng=15');
suptitle('Representation of signal x3');




clc
clear all
close all
addpath(genpath('tftb-0.2'));
%% Question 2.3 Influence of noise
N=256;
n=linspace(0,N-1,N);  %N=256 et n=0,...,N-1
phi1=0.25*n+5*cos(2*pi*n/N);  %allure
phi2=0.25*n+0.1*(n.^2)/N;
x1=cos(2*pi*phi1);  %phase
x2=cos(2*pi*phi2);
f1=diff(phi1)./diff(n);  % frequency: d(phi)/d(t)
f2=diff(phi2)./diff(n);
```

```
T1 = 15; T2 = 85; T3 = 180;
Nh = 51; th = (0:Nh-1); h = gausswin(Nh)';
x3 = zeros(1,N);
x3(T1:T1+Nh-1) = h.*cos(2*pi*0.25*th);
x3(T2:T2+Nh-1) = h.*cos(2*pi*0.15*th);
x3(T2:T2+Nh-1) = x3(T2:T2+Nh-1) + h.*cos(2*pi*0.35*th);
x3(T3:T3+Nh-1) = h.*cos(2*pi*0.3*th);


%add the noise
x1=ajoute_bruit(x1,10);
x2=ajoute_bruit(x2,10);
x3=ajoute_bruit(x3,10);


%% Question 2.2: Representations temps-frequence
% Spectrogram of signal x1
Nh1=17;
Nh2=33;
Nh3=65;
Nh4=129;
h1= hamming(Nh1);
[tfrx11,t11,f11] = tftb_spectrogram(x1,N,h1);
h2= hamming(Nh2);
[tfrx12,t12,f12] = tftb_spectrogram(x1,N,h2);
h3= hamming(Nh3);
[tfrx13,t13,f13] = tftb_spectrogram(x1,N,h3);
h4= hamming(Nh4);
[tfrx14,t14,f14] = tftb_spectrogram(x1,N,h4);
figure(4)
subplot(4,1,1);
imagesc(t11,f11,tfrx11); axis xy;
title('Nh=17');
subplot(4,1,2);
imagesc(t12,f12,tfrx12); axis xy;
title('Nh=33');
subplot(4,1,3);
imagesc(t13,f13,tfrx13); axis xy;
title('Nh=65');
subplot(4,1,4);
imagesc(t14,f14,tfrx14); axis xy;
title('Nh=129');
suptitle('Spectrogramme of signal x1');


% Spectrogram of signal x2
[tfrx21,t21,f21] = tftb_spectrogram(x2,N,h1);
[tfrx22,t22,f22] = tftb_spectrogram(x2,N,h2);
```

```
[tfrx23,t23,f23] = tftb_spectrogram(x2,N,h3);
[tfrx24,t24,f24] = tftb_spectrogram(x2,N,h4);
figure(5)
subplot(4,1,1);
imagesc(t21,f21,tfrx21); axis xy;
title('Nh=17');
subplot(4,1,2);
imagesc(t22,f22,tfrx22); axis xy;
title('Nh=33');
subplot(4,1,3);
imagesc(t23,f23,tfrx23); axis xy;
title('Nh=65');
subplot(4,1,4);
imagesc(t24,f24,tfrx24); axis xy;
title('Nh=129');
suptitle('Spectrogram of signal x2');
% Spectrogram of signal x3
[tfrx31,t31,f31] = tftb_spectrogram(x3,N,h1);
[tfrx32,t32,f32] = tftb_spectrogram(x3,N,h2);
[tfrx33,t33,f33] = tftb_spectrogram(x3,N,h3);
[tfrx34,t34,f34] = tftb_spectrogram(x3,N,h3);
figure(6)
subplot(4,1,1);
imagesc(t31,f31,tfrx31); axis xy;
title('Nh=17');
subplot(4,1,2);
imagesc(t32,f32,tfrx32); axis xy;
title('Nh=33');
subplot(4,1,3);
imagesc(t33,f33,tfrx33); axis xy;
title('Nh=65');
subplot(4,1,4);
imagesc(t34,f34,tfrx34); axis xy;
title('Nh=129');
suptitle('Spectrogram of signal x3');


% Reprsentation of signal x1
[tfrx1_1,t1_1,f1_1] = tftb_wvd(x1,N);
beta=2.5;
L=63;
h_pwv=kaiser(L,beta);
[tfrx1,t1,f1] = tftb_pwvd(x1,N,h_pwv);


Ng1=33;
Nhp=63;
Ng2=15;
```

```
h_spwv=kaiser(Nhp,beta);
g1=kaiser(Ng1,beta);
g2=kaiser(Ng2,beta);


[tfrx_11,t_11,f_11] = tftb_spwvd(x1,N,g1,h_spwv);
[tfrx_21,t_21,f_21] = tftb_spwvd(x1,N,g2,h_spwv);


figure(7)
subplot(4,1,1);
imagesc(t1_1,f1_1,tfrx1_1); axis xy;
title('Transform - Wigner-Ville');
subplot(4,1,2);
imagesc(t1,f1,tfrx1); axis xy;
title('Transform - pseudo-Wigner-Ville');
subplot(4,1,3);
imagesc(t_11,f_11,tfrx_11); axis xy;
title('Transform - pseudo-Wigner-Ville with Ng=33');
subplot(4,1,4);
imagesc(t_21,f_21,tfrx_21); axis xy;
title('Transform - de pseudo-Wigner-Ville with Ng=15');
suptitle('Representation of signal x1');


% Representation of signal x2
[tfrx2_1,t2_1,f2_1] = tftb_wvd(x2,N);
[tfrx2,t2,f2] = tftb_pwvd(x2,N,h_pwv);
[tfrx_12,t_12,f_12] = tftb_spwvd(x2,N,g1,h_spwv);
[tfrx_22,t_22,f_22] = tftb_spwvd(x2,N,g2,h_spwv);


figure(8);
subplot(4,1,1);
imagesc(t2_1,f2_1,tfrx2_1); axis xy;
title('Transform - Wigner-Ville');
subplot(4,1,2);
imagesc(t2,f2,tfrx2); axis xy;
title('Transform - pseudo-Wigner-Ville');
subplot(4,1,3);
imagesc(t_12,f_12,tfrx_12); axis xy;
title('Transform - pseudo-Wigner-Ville with Ng=33');
subplot(4,1,4);
imagesc(t_22,f_22,tfrx_22); axis xy;
title('Transform -  pseudo-Wigner-Ville with Ng=15');
suptitle('Representation of signal x2');
```

```
% Representation of signal x3
[tfrx3_1,t3_1,f3_1] = tftb_wvd(x3,N);
[tfrx3,t3,f3] = tftb_pwvd(x3,N,h_pwv);
[tfrx_13,t_13,f_13] = tftb_spwvd(x3,N,g1,h_spwv);
[tfrx_23,t_23,f_23] = tftb_spwvd(x3,N,g2,h_spwv);


figure(9);
subplot(4,1,1);
imagesc(t3_1,f3_1,tfrx3_1); axis xy;
title('Transform - Wigner-Ville');
subplot(4,1,2);
imagesc(t3,f3,tfrx3); axis xy;
title('Transform - pseudo-Wigner-Ville');
subplot(4,1,3);
imagesc(t_13,f_13,tfrx_13); axis xy;
title('Transform - pseudo-Wigner-Ville with Ng=33');
subplot(4,1,4);
imagesc(t_23,f_23,tfrx_23); axis xy;
title('Transform - pseudo-Wigner-Ville with Ng=15');
suptitle('Representation of signal x3');



%% 3 Detection and reconstruction of a musical partition
clc
clear
addpath(genpath('tftb-0.2'));
load('freq_notes.mat');
%% 3.2.1
%[y,Fs] = audioread('furElise_court.wav'); % Fs = 5513
[y,Fs] = audioread('furElise.wav');
figure;plot(y)
title('Signal original')
%% 3.2.2
% sound(y,Fs);
Nf = 1024;
Nh = 513;
h = hamming(Nh);
[tfrx,t,f] = tftb_spectrogram(y,Nf,h);
Axe_f=(0:length(f)-1)/length(f)*Fs;
figure;imagesc(t,Axe_f,log(tfrx)); axis xy
title('Amplitude in logarithmic scale')
%% 3.2.3
% to choose
tau = 10;
```

```
I = zeros(1,length(y)-2*tau);  % We allocate a zero vector
for the Kolmogorov distance
[tfry,t,f] = tftb_wvd(y,Nf); % We take the Wigner-Ville
method to calculate the RTFs


%% We calcule the distance I(t)
for k = 1+tau:length(t)-tau
R1 = tfrx(:,k-tau:k-1); % size(R1) =512*6
R2 = tfrx(:,k+1:k+tau); % length(R2)=512


sum_R1 = sum(abs(R1(:)));
   sum_R2 = sum(abs(R2(:)));

   R1_tilde = R1/sum_R1;
   R2_tilde = R2/sum_R2;

   I(k) = sum(abs(R1_tilde(:)-R2_tilde(:)));
end
figure;plot(I)


%% We calculate the frequency for  which the TF + grande
[pks,locs] = findpeaks(I,'MinpeakHeight',0.035);


N = length(y);
locs = [1,locs,N];
npks = length(locs);
tab_freq = zeros(npks-1,1);
amp = zeros(npks-1,1);




for i = 1:npks-1
    Axe_freq = linspace(0,Fs-Fs/Nf,Nf);
    yfft = fft(y(locs(i):locs(i+1)),Nf);
    [maxyfft,index_maxyfft] = max(abs(yfft));

    amp(i) = maxyfft;
    tab_freq(i) = Axe_freq(index_maxyfft);

    for j = 1:25
        if abs(tab_freq(i)-tab_freq_valeurs(j))<1
```

```
            tab_freq(i)=tab_freq_valeurs(j);
        end
    end

    tab_duree(i) = (locs(i+1)-locs(i)-1)/Fs;
end


[morceau,t] = genere_morceau(tab_freq,tab_duree,amp,Fs);
sound(morceau,Fs);
figure;
plot(morceau)
```