

Représentation Temps-Fréquence : travaux pratiques

Yassine Jamoud, Samy Haffoudhi

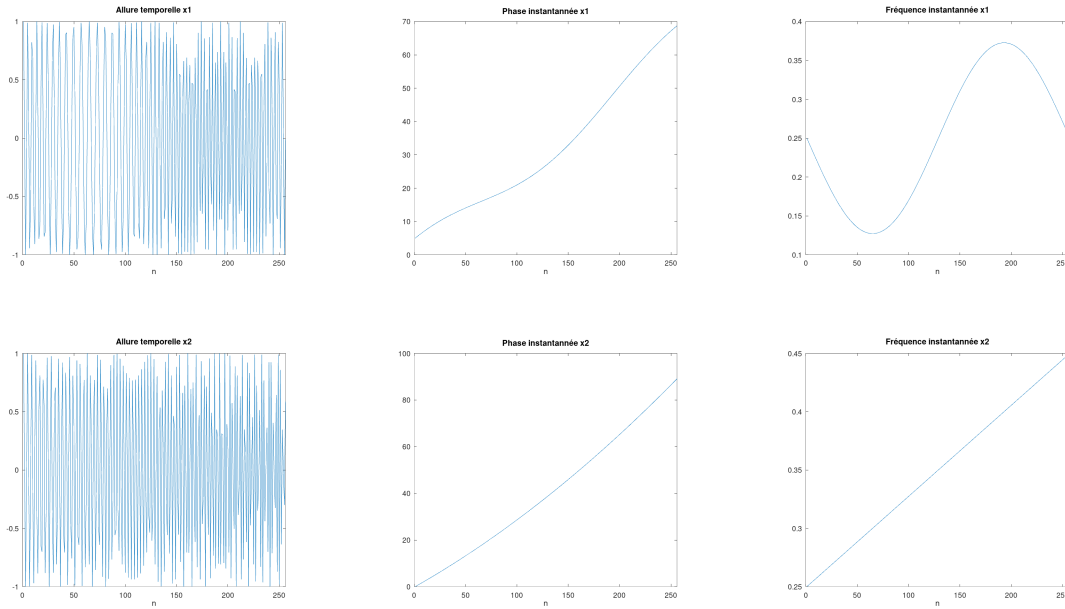
10 octobre 2021

2 Représentation temps-fréquence de signaux simulés

2.1 Génération de signaux

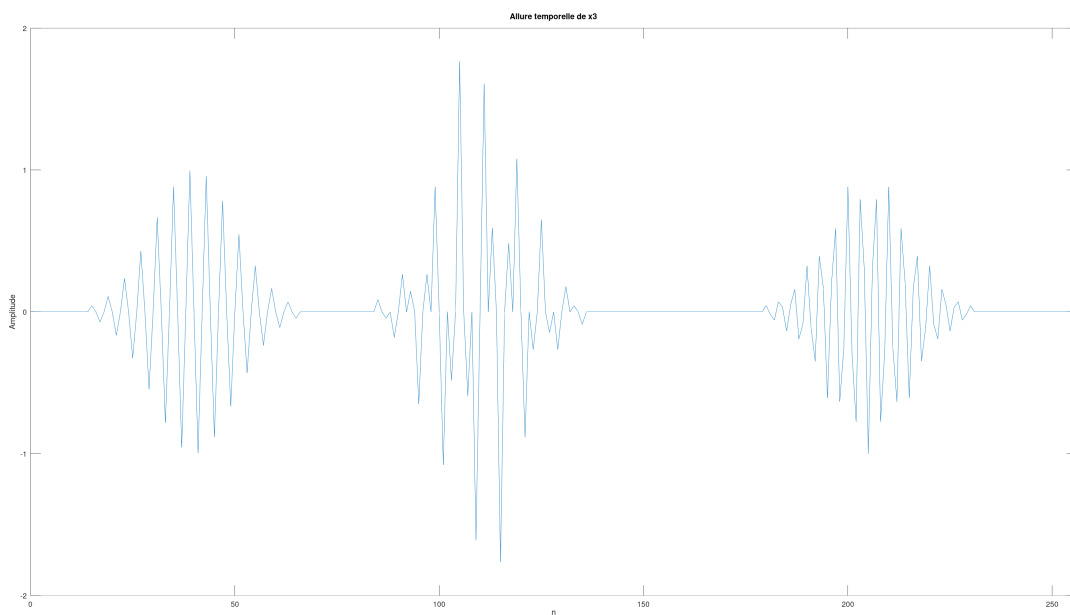
Représentons l'allure temporelle, la phase instantanée et la fréquence instantanée des signaux mono-composantes x_1 et x_2 :

FIGURE 1 – allure temporelle, phase et fréquence instantanée de x_1 et x_2



Représentons maintenant l'allure du signal x_3 , somme de quatre atomes gaussiens :

FIGURE 2 – Allure temporelle du signal x_3



Nous avons obtenus ces différentes figures à l'aide du script présenté en **Annexe A** et par le calcul au préalable de $f_i(t) = \frac{d\phi_i(t)}{dt}$.

2.2 Représentation temps-fréquence

Pour chaque signal nous allons maintenant représenter :

- Le spectrogramme en utilisant une fenêtre de Hamming de longueur $N_h = 17, 33, 65$ et 129 .
- La transformée de Wigner-Ville
- La transformée de pseudo-Wigner-Ville en utilisant uen fenêtre de Kaiser avec $N_h = 63$.
- La transformée de pseudo-Wigner-Ville lissée avec des fenêtres de Kaiser de longueur $N_g = 33$ et $N_h = 63$, puis $N_g = 15$ et $N_h = 63$.

On obtient les figures suivantes :

FIGURE 3 – Tracés x_1

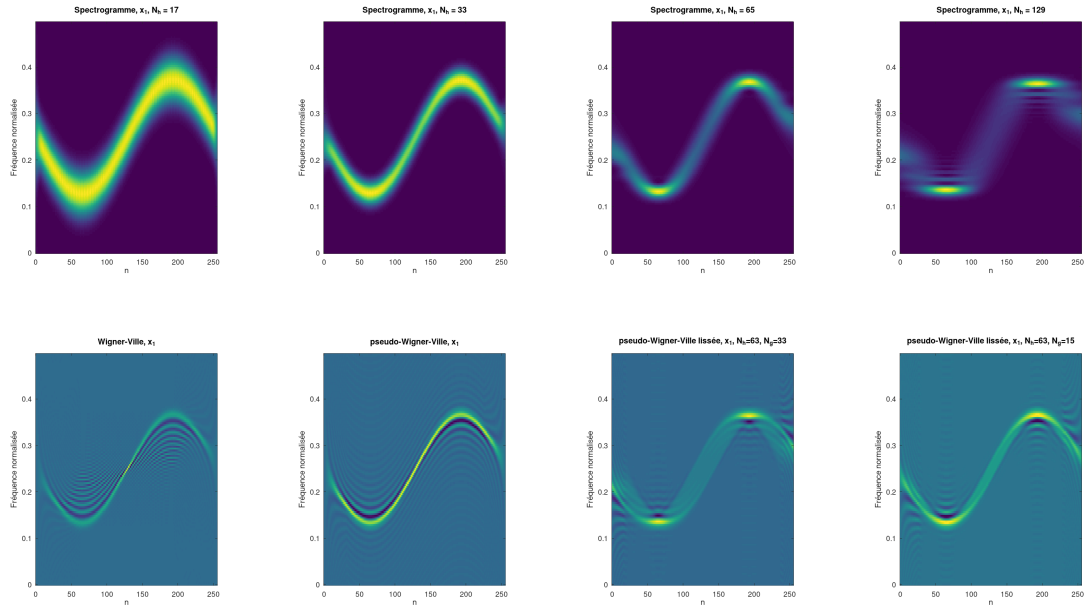


FIGURE 4 – Tracés x_2

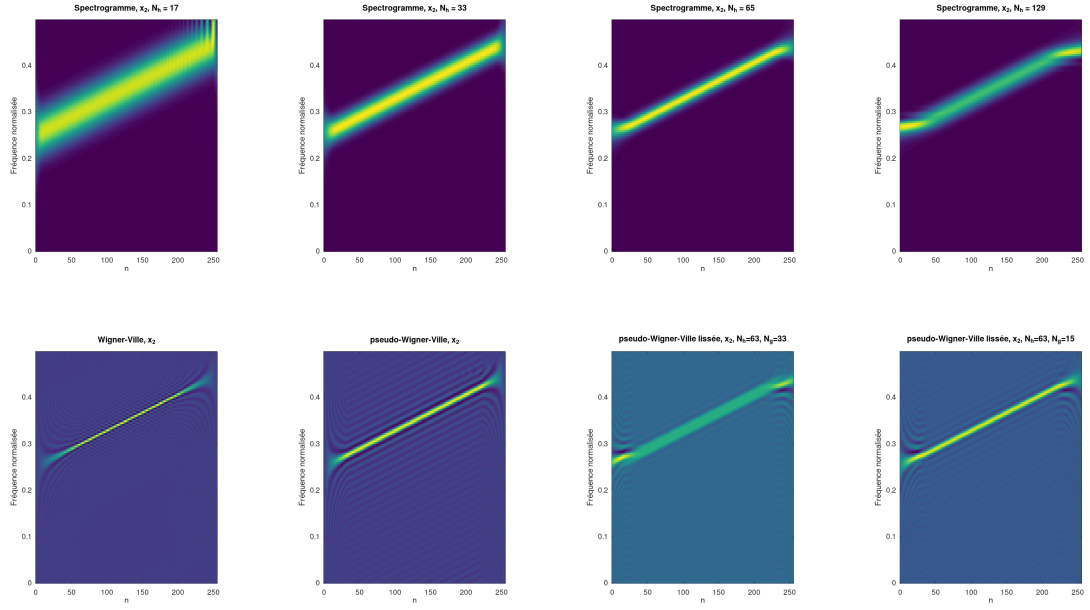
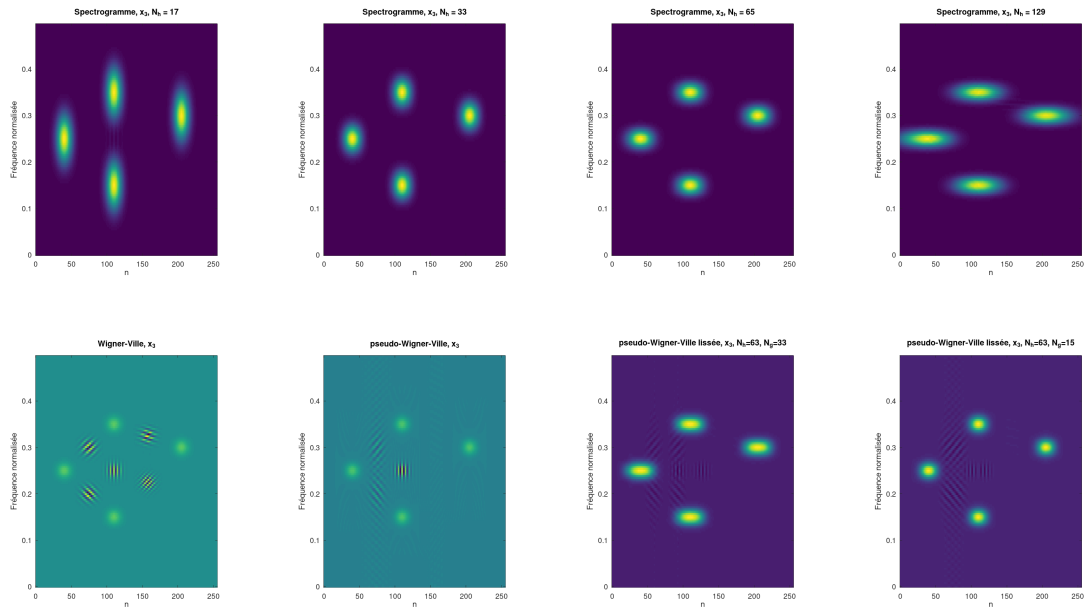


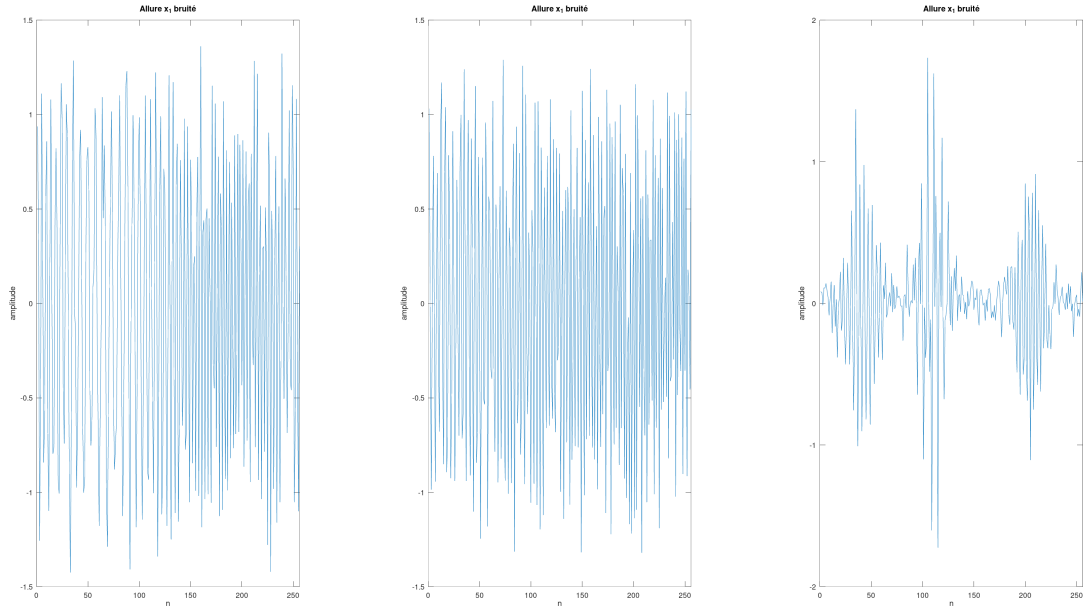
FIGURE 5 – Tracés x_3



2.3 Influence du bruit

Perturbons les signaux à l'aide d'un bruit additif gaussien pour un rapport signal sur bruit de 10dB, on obtient les allures temporelles suivantes :

FIGURE 6 – Allure temporelles des signaux bruités



Les différentes représentations suivantes deviennent alors (avec x_4 correspondant à x_1 bruité, x_5 à x_2 bruité et x_6 à x_3 bruité) :

FIGURE 7 – Tracés x_1 bruité

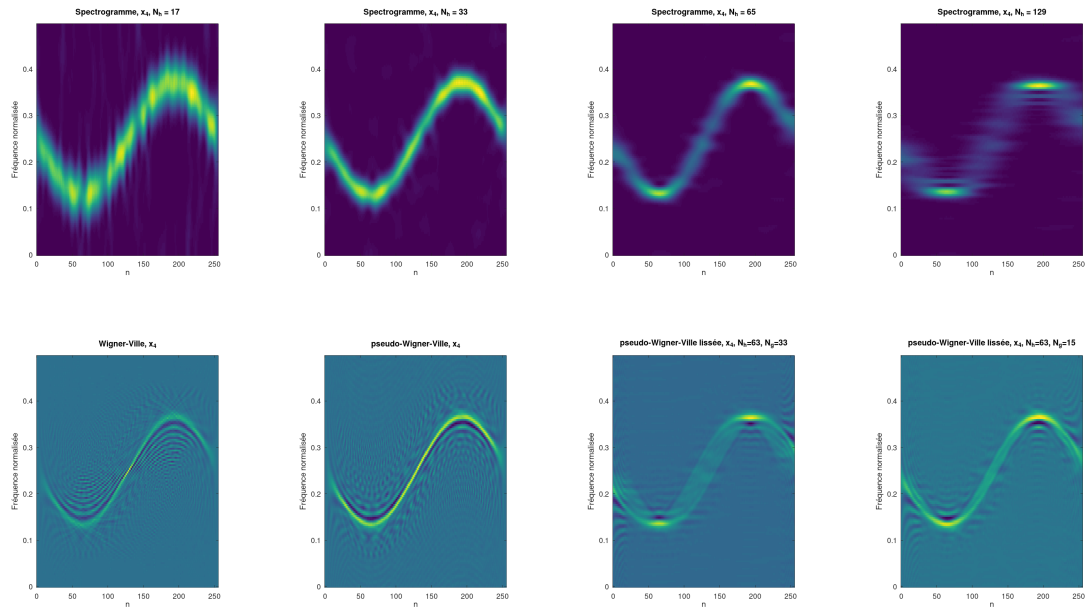


FIGURE 8 – Tracés x_2 bruité

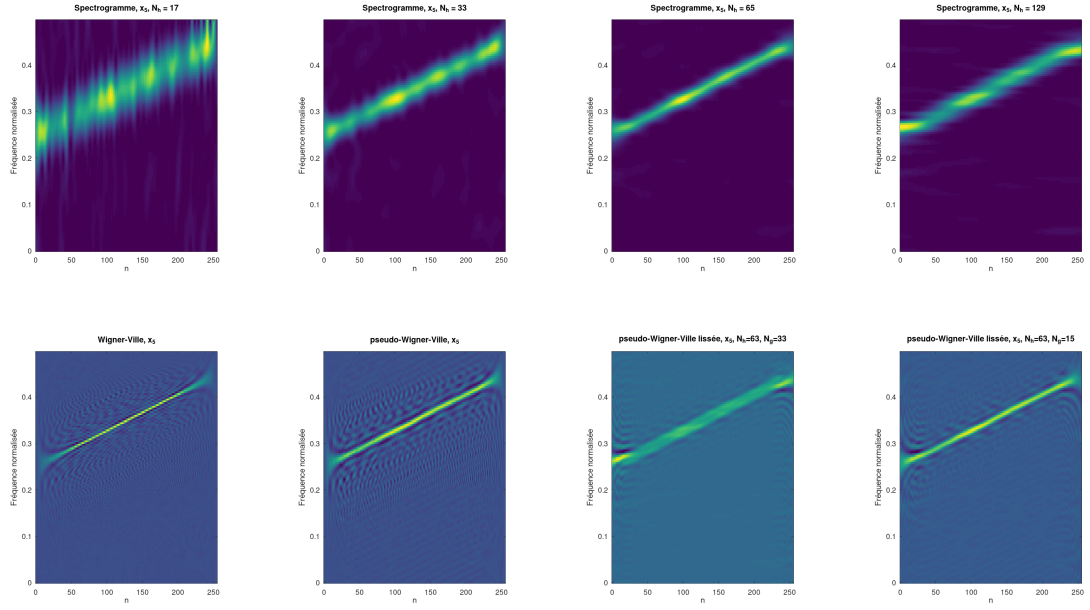
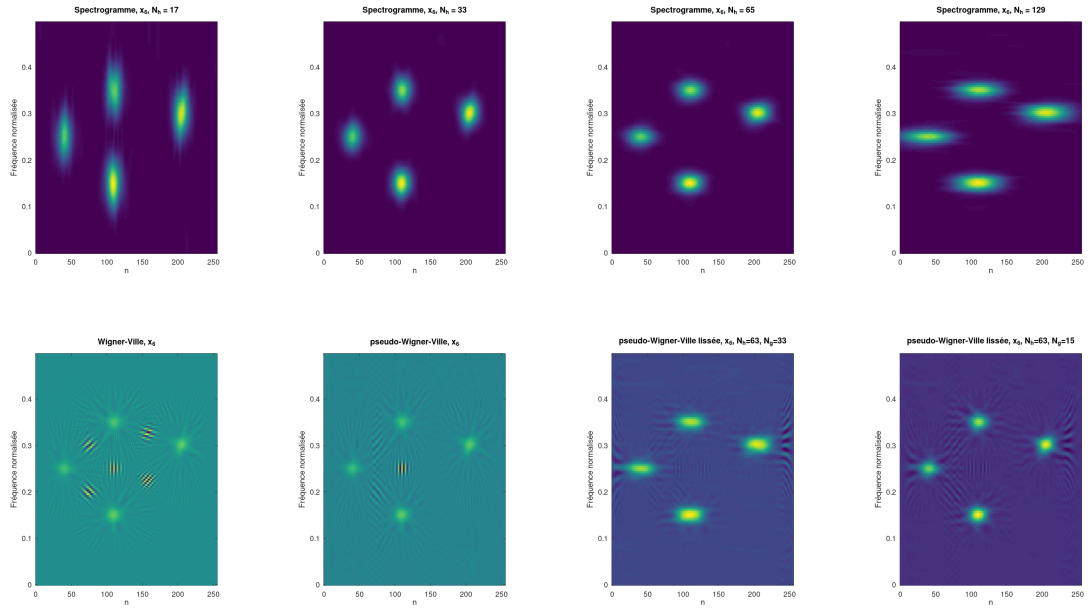


FIGURE 9 – Tracés x_3 bruité



3 Détection et reconstruction d'une partition musicale

1. On charge le fichier à l'aide de la fonction `audioread` de matlab
2. Représentons l'allure temporelle et le spectrogramme en échelle logarithmique de ce signal :
- 3.
- 4.
- 5.

A Scripts Représentation temps-fréquence de signaux simulés

```
1 addpath(genpath('tftb-0.2'));
2
3 N = 256;
4 n = 0:N-1;
5
6 phi1 = 0.25 * n + 5 * cos(2 * pi * n / N);
7 phi2 = 0.25 * n + 0.1 * n.^2/N;
8
9 f1 = 0.25 - 5 * 2 * pi / N * sin(2 * pi * n / N);
10 f2 = 0.25 + 0.2 * n / N;
11
12 x1 = cos(2 * pi * phi1);
13 x2 = cos(2 * pi * phi2);
14
15 figure(1);
16 subplot(2, 3, 1);
17 plot(x1);
18 title('Allure temporelle x1')
19 xlim([0 N])
20 xlabel('n')
21 subplot(2, 3, 2);
22 plot(phi1);
23 title('Phase instantann e x1')
24 xlim([0 N])
25 xlabel('n')
26 subplot(2, 3, 3);
27 plot(f1);
28 title('Fr quence instantann e x1')
29 xlim([0 N])
30 xlabel('n')
31 subplot(2, 3, 4);
32 plot(x2);
33 title('Allure temporelle x2')
34 xlim([0 N])
35 xlabel('n')
36 subplot(2, 3, 5);
37 plot(phi2);
38 title('Phase instantann e x2')
39 xlim([0 N])
40 xlabel('n')
41 subplot(2, 3, 6);
42 plot(f2);
43 title('Fr quence instantann e x2')
44 xlim([0 N])
45 xlabel('n')
46
47 T1 = 15;
```

```

48 T2 = 85;
49 T3 = 180;
50 Nh = 51;
51 th = (0:Nh-1);
52 h = gausswin(Nh)';
53 x3 = zeros(1,N);
54 x3(T1:T1+Nh-1) = h.*cos(2*pi*0.25*th);
55 x3(T2:T2+Nh-1) = h.*cos(2*pi*0.15*th);
56 x3(T2:T2+Nh-1) = x3(T2:T2+Nh-1) + h.*cos(2*pi*0.35*th);
57 x3(T3:T3+Nh-1) = h.*cos(2*pi*0.3*th);
58
59 figure(2);
60 plot(x3);
61 xlabel('n')
62 xlim([0 N])
63 title('Allure temporelle de x3')
64 ylabel('Amplitude')
65
66 x1_bruit = ajoute_bruit(x1,10);
67 x2_bruit = ajoute_bruit(x2,10);
68 x3_bruit = ajoute_bruit(x3,10);
69
70 figure(3);
71 subplot(1, 3, 1);
72 plot(x1_bruit);
73 title('Allure x_1 bruit ')
74 xlim([0 N])
75 xlabel('n')
76 ylabel('amplitude')
77 subplot(1, 3, 2);
78 plot(x2_bruit);
79 title('Allure x_1 bruit ')
80 xlim([0 N])
81 xlabel('n')
82 ylabel('amplitude')
83 subplot(1, 3, 3);
84 plot(x3_bruit);
85 title('Allure x_1 bruit ')
86 xlim([0 N])
87 xlabel('n')
88 ylabel('amplitude')
89
90 x = [x1; x2; x3; x1_bruit; x2_bruit; x3_bruit];
91 for j = 1:6
92     figure(3 + j);
93
94     Nh = [17 33 65 129];
95     for i = 1:4
96         h = hamming(Nh(i));
97         [tfrx,T,F] = tftb_spectrogram(x(j, :)',N,h);
98         subplot(2, 4, i);
99         imagesc(T,F,tfrx); axis xy
100         title(sprintf('Spectrogramme, x_%i, N_h = %i', j, Nh(i)))
101         xlabel('n')
102         ylabel('Fr quence normalis e')
103     end
104
105     [tfrx,T,F] = tftb_wvd(x(j, :)',N);
106     subplot(2, 4, 5);

```



```

107 imagesc(T,F,tfrx); axis xy
108 title(sprintf('Wigner-Ville , x_%i', j))
109 xlabel('n')
110 ylabel('Fr quence normalis e')
111
112 Nh = 63;
113 h = kaiser(Nh);
114 [tfrx,T,F] = tftb_pwvd(x(j, :)',N,h);
115 subplot(2, 4, 6);
116 imagesc(T,F,tfrx); axis xy
117 title(sprintf('pseudo-Wigner-Ville , x_%i', j))
118 xlabel('n')
119 ylabel('Fr quence normalis e')
120
121 Ng = [33, 15];
122 for k = 1:2
123     h = kaiser(Nh);
124     g = kaiser(Ng(k));
125     [tfrx,T,F] = tftb_spwvd(x(j, :)',N, g, h);
126     subplot(2, 4, 6 + k);
127     imagesc(T,F,tfrx); axis xy
128     title(sprintf('pseudo-Wigner-Ville liss e , x_%i , N_h=%i , N_g=%i', j, Nh, Ng
(k)))
129     xlabel('n')
130     ylabel('Fr quence normalis e')
131 end
132 end

```

B Scripts Détection et reconstruction d'une partition musicale

```

1 addpath(genpath('tftb-0.2'));
2
3 [x, Fs] = audioread('furElise_court.wav');
4
5 N = length(x);
6 t = 0:1/Fs:(N-1)/Fs;
7
8 % subplot(2,1,1);
9 % plot(t,x);
10 % title('Allure temporelle')
11 % xlabel('temps (s)')
12 % xlim([0 t(N)])
13 % ylabel('Amplitude')
14
15 Nh = 513;
16 Nf = 8*2^nextpow2(Nh);
17 h = hamming(Nh);
18 [tfrx,T,F] = tftb_spectrogram(x',Nf,h);
19
20 subplot(2, 1, 2);
21 imagesc(T,F,log(tfrx)); axis xy;
22 colorbar;
23 title('Spectrogramme en chelle logarithmique')
24 xlabel('n')
25 ylabel('Fr quence normalis e')
26
27 tau = 10;

```

```

28 seuil = 0.1;
29 I = zeros(1, N - 2 * tau);
30 for t = tau+1:N-tau
31     R1 = tfrx(:, t-tau:t);
32     R1 = R1 / sum(abs(R1));
33     R2 = tfrx(:, t:t+tau);
34     R2 = R2 / sum(abs(R2));
35
36     I(t) = sum(abs(R1 - R2));
37 end
38
39 [maxima, ruptures] = findpeaks(I, 'MinPeakHeight', 0.04, 'MinPeakProminence', 0.003);
40 Nf=2^nextpow2(N);
41 axe_freq=linspace(0, Fs-Fs/Nf, Nf);
42
43 notes = [];
44 durees = [];
45 val=0;
46 for i = 1:length(ruptures)-1
47     x_plage = x(ruptures(i):ruptures(i+1));
48     TF = fft(x_plage, Nf);
49     TF = abs(TF).^2;
50     [maxima, lambda] = max(TF);
51     f = axe_freq(lambda);
52     ecart = tab_freq_valeurs - f;
53     [minimum, indice] = min(abs(ecart));
54     notes = [notes; tab_freq_noms(indice,:)];
55     nb_points = ruptures(i+1)-ruptures(i);
56     durees = [durees; nb_points/Fs];
57 end
58
59 s = genere_morceau(notes, durees, Fs);
60 audiowrite('eliseTest.wav', s, Fs);

```