**Code Damien “test2.m” 04/01/17 pour débruiter tout un signal**

clear all

close all

clc

% programme qui calcule la lpc sur chaque trame avec une boucle for et concatene les y pour

% générer le signal complet débruité Y

% Choix des paramètres

[X,Fe] = audioread('101.wav');

p = 15; %ordre du prédicteur

T\_trame = 0.150; % durée de la trame en sec

N\_trame = T\_trame\*Fe; %nombre d'échantillons par trame

nX = size(X);

nX = nX(1); %nombre d'échantillons total de X

K = floor(nX/N\_trame);

pX = var(X); %variance de la trame

RSB = 1;

pB = pX/RSB; %variance du bruit

% Définition de V et Y

V = sqrt(pB)\*randn(nX,1); %génération du signal bruit

S = X+V;

for k=1:K-1

x = X(k\*N\_trame:(k+1)\*N\_trame);

v = V(k\*N\_trame:(k+1)\*N\_trame);

s = S(k\*N\_trame:(k+1)\*N\_trame);

% LPC & filtre de Wiener

[A,pe] = lpc(x,p);

G = sqrt(N\_trame\*pe);

num\_wiener = G/sqrt(pB\*N\_trame);

A\_inv = flip(A);

AA = conv(A,A\_inv);

AA(p+1) = AA(p+1)+num\_wiener^2;

AAracine = roots(AA);

size\_AAracine = size(AAracine);

size\_AAracine = size\_AAracine(1);

AAracine\_caus = 1;

j=0;

for i=1:size\_AAracine

if abs(AAracine(i))<1

j = j+1;

AAracine\_caus(j) = AAracine(i);

end

end

AAcaus = poly(AAracine\_caus);

y = filter(num\_wiener,AAcaus,s);

V(k\*N\_trame:(k+1)\*N\_trame) = v;

S(k\*N\_trame:(k+1)\*N\_trame) = s;

Y(k\*N\_trame:(k+1)\*N\_trame) = y;

end

% Graphiques

Max\_x = max(X);

Max\_s = max(S);

Max\_y = max(Y);

M = max([Max\_x,Max\_s,Max\_y]);

M = 1.2\*M;

figure;

subplot(3,1,1)

plot(X);

ylim([-M M]);

title('signal non bruité X');

legend('signal original X');

xlabel('échantillons');

subplot(3,1,2)

plot(S,'k')

ylim([-M M]);

legend('signal bruité S');

title('signal bruité S');

xlabel('échantillons');

subplot(3,1,3)

plot(Y,'r')

ylim([-M M]);

title('signal débruité Y');

legend('signal débruité Y');

xlabel('échantillons');