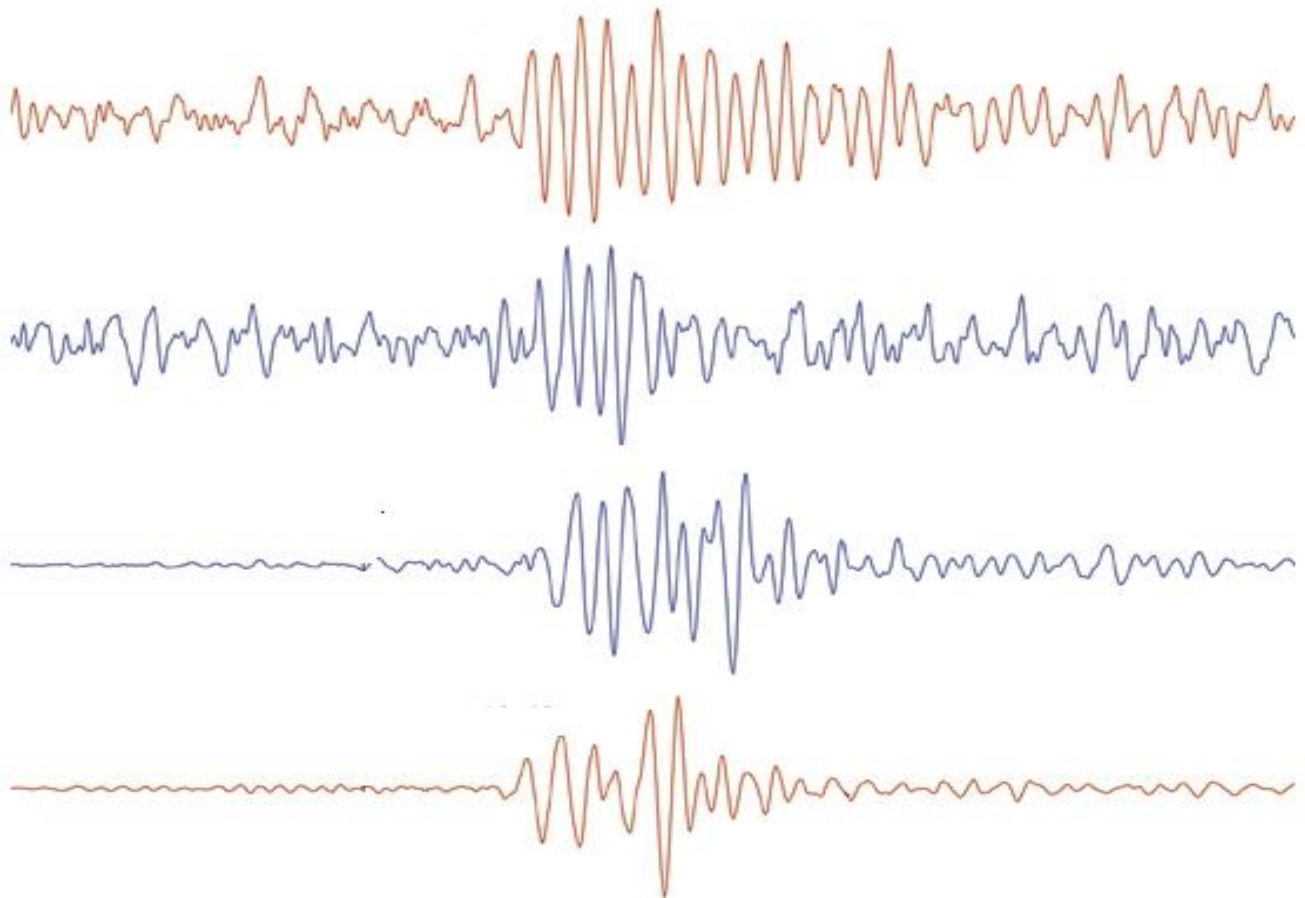


Traitement optimale d'un signal sismique :

(sujet 4)



Professeur : FAKHITA REGRAGUI .

Réaliser par :

Lasri Wafae (IASE) .

Lasri Kawtar (IASE).

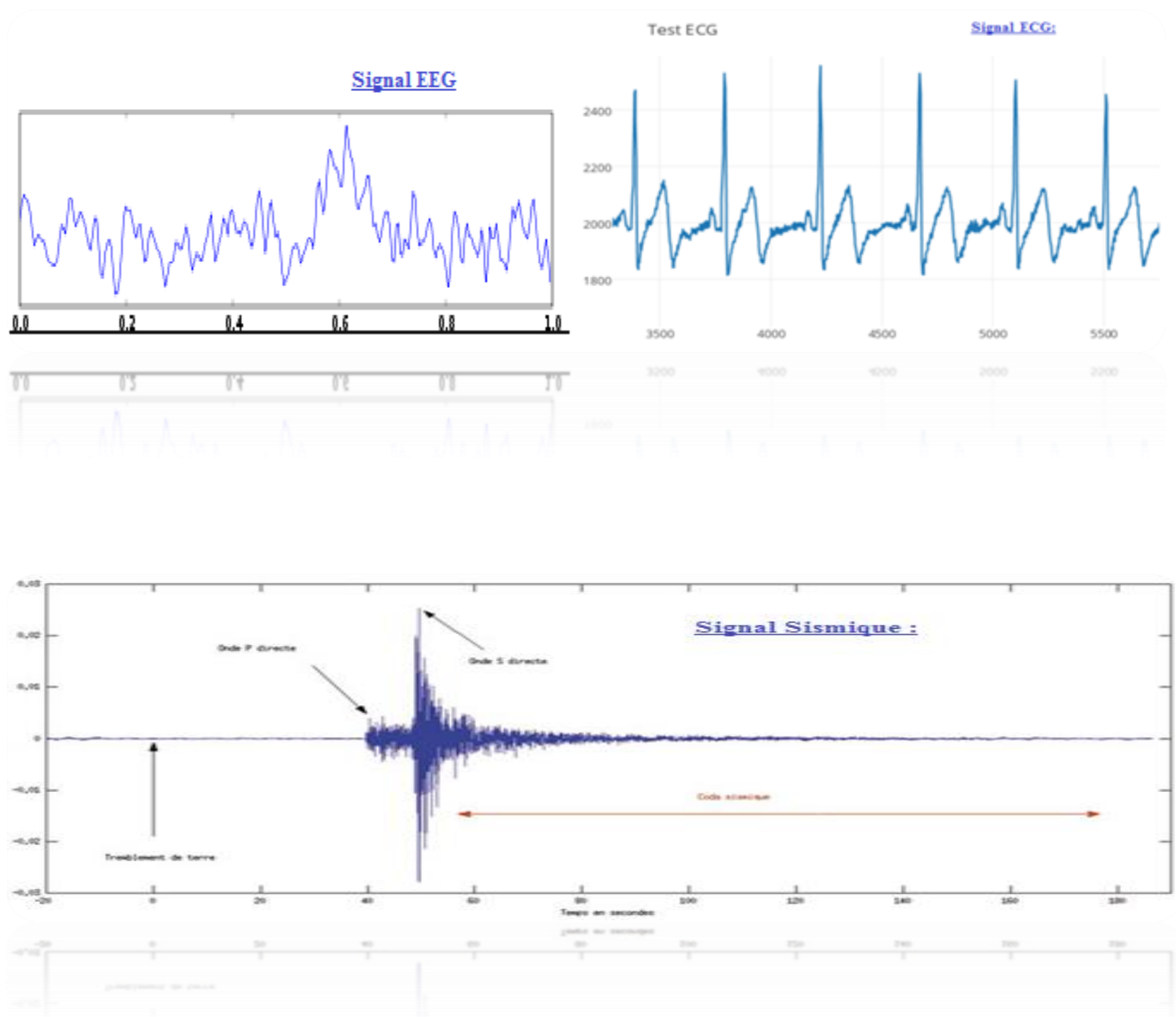
Table de contenu:

Introduction	5
Partie I : Extraction d'une sinusoides d'un signal bruité (modèle AR)...	7
Partie II : Analyse d'un signal sismique :	15
Annexe (programme sous Matlab)	23

Introduction :

Le but du projet est d'analyser des signaux aléatoires et réelles en utilisant des techniques du traitement optimale du signal tel que le Processus Autorégressive, Prédiction linéaire, Filtrage Adaptative et Estimation linéaire.

Ces différentes techniques permettent d'analyser des signaux tels que : signal de parole, signal sismique, signal neurologique et signal cardiaque .



Partie I : Extraction d'une sinusoïde d'un signal bruité par le modèle Autorégressif :

Partie I : Extraction d'une sinusoides d'un signal bruité (modèle AR)

Le processus autorégressif permet de modéliser un signal comme la sortie d'un filtre Soit Y un processus AR(p) on a l'équation suivante :

$$Y_{t+1} = \phi_1 Y_t + \dots + \phi_p Y_{t-p+1} + Z_{t+1} \quad t \in \mathbb{Z}$$

C'est un modèle paramétrique permettant d'estimer les attributs d'un signal.

➤ Enoncé :

Soit le processus autorégressif d'ordre 4 qui est définie par l'équation aux différences :

$$y(n) + a_1 * y(n-1) + a_2 * y(n-2) + a_3 * y(n-3) + a_4 * y(n-4) = b(n)$$

$b(n)$: un bruit blanc gaussien de moyenne nulle et de variance σ^2 .

Les paramètres du filtre (ou modèle) $\{ a_1, a_2, a_3, a_4 \}$ sont choisis tel que :

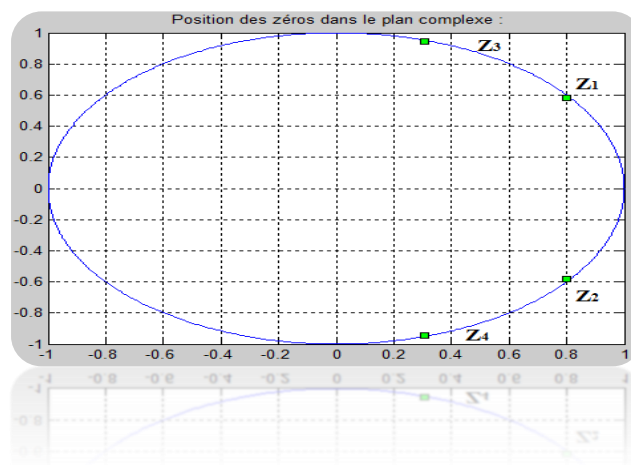
$$A(z) = 1 + a_1 * z^{-1} + a_2 * z^{-2} + a_3 * z^{-3} + a_4 * z^{-4}$$

$A(z)$ possède des zéros (racines) aux positions :

$$\left\{ \begin{array}{l} Z_1 = 0.99 \exp(+0.2\pi j) . \\ Z_2 = 0.99 \exp(-0.2\pi j) . \\ Z_3 = 0.99 \exp(+0.4\pi j) . \\ Z_4 = 0.99 \exp(-0.4\pi j) . \end{array} \right.$$

➤ Analyse:

1. Représentation des zéros du modèle AR dans le plan complexe et déduction des paramètres :



$A(z)$ est un filtre à phase minimale (système stable). En effet, tous les zéros du filtre $A(z)$ se trouvent à l'intérieur du cercle d'unité.

La résolution du système ci-dessus nous permet de déduire les valeurs exactes du vecteur de paramètres $\{a_1, a_2, a_3, a_4\}$:

$$\begin{cases} 1 + a_1 z_1^{-1} + a_2 z_1^{-2} + a_3 z_1^{-3} + a_4 z_1^{-4} = 0 \\ 1 + a_1 z_2^{-1} + a_2 z_2^{-2} + a_3 z_2^{-3} + a_4 z_2^{-4} = 0 \\ 1 + a_1 z_3^{-1} + a_2 z_3^{-2} + a_3 z_3^{-3} + a_4 z_3^{-4} = 0 \\ 1 + a_1 z_4^{-1} + a_2 z_4^{-2} + a_3 z_4^{-3} + a_4 z_4^{-4} = 0 \end{cases}$$

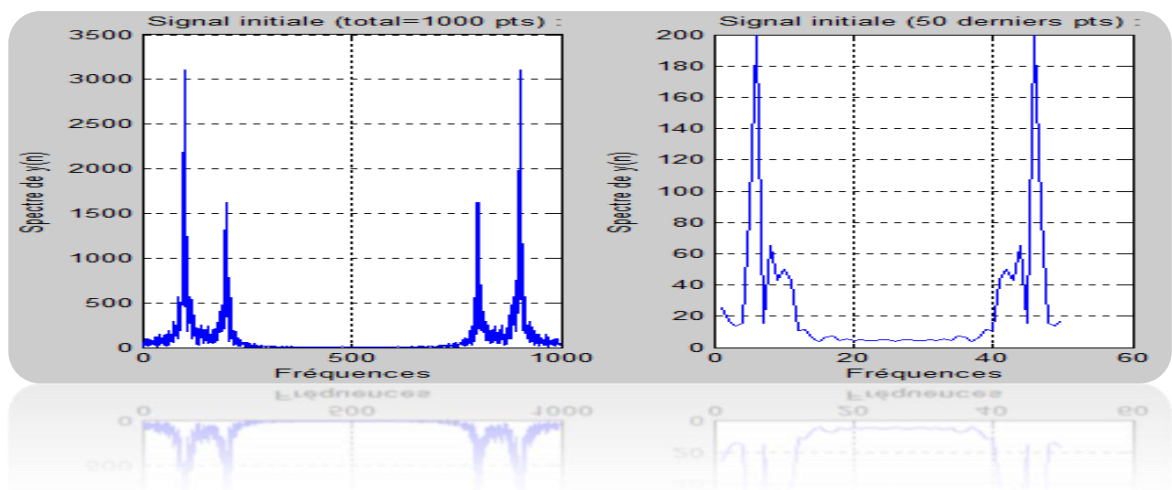
....On trouve :

$\mathbf{a} =$

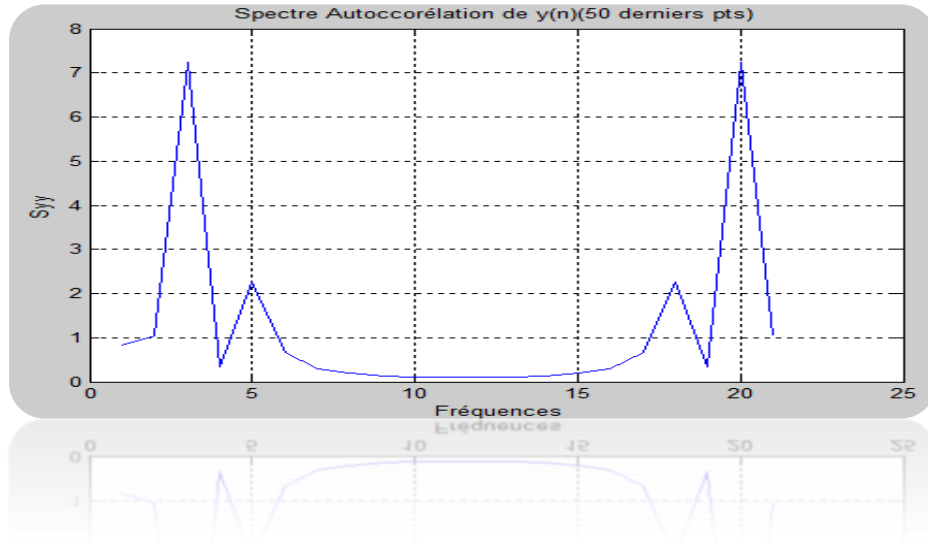
$$\begin{bmatrix} -2.2137 & + & 0.0000i \\ 2.9403 & - & 0.0000i \\ -2.1697 & + & 0.0000i \\ 0.9606 & - & 0.0000i \end{bmatrix}$$

2. Signal sinusoïdale bruité par un bruit blanc de moyenne nulle et de variance 1 :

Afin d'éliminer les effets transitoires du signal $y(n)$, on ne considère que ses 50 derniers points (Dans ce qui suit $y = y(50)$) :



3. Fonction d'autocorrélation de $y(n)$:



Le spectre de l'autocorrélation reflète la contribution qu'apporte chaque pics à la puissance moyenne du signal.

4. Estimation des paramètres du modèle AR et σ^2 :

Pour estimer les paramètres du modèle AR, on a besoins de résoudre les équations normales. Il existe plusieurs méthodes de résolution , parmi eux :

- Calcul de l'inverse de la matrice d'autocorelation .
- Algorithme de Levinson .
- Méthode de Yulewalker .

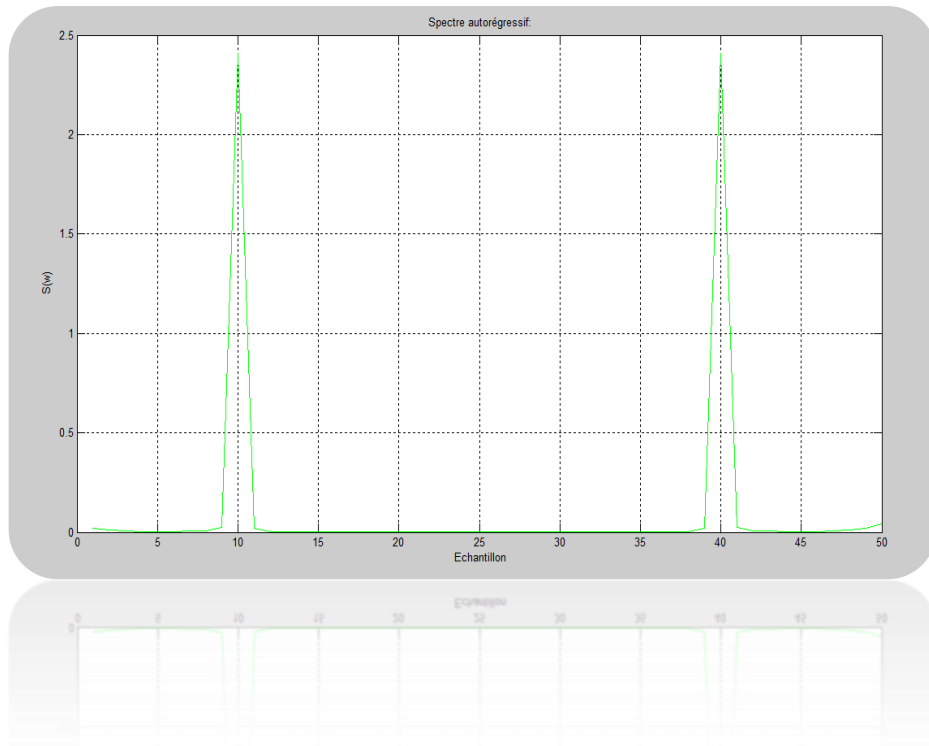
En utilisant l'algorithme de Levinson qui nécessite moins d'opérations ,on trouve :

```
a =  
-2.2137 + 0.0000i  
2.9403 - 0.0000i  
-2.1697 + 0.0000i  
0.9606 - 0.0000i  
  
a1 =  
1.0000 -2.2059 2.9171 -2.1449 0.9460  
  
e =  
0.0131
```

On remarque que les coefficients obtenus par la méthode de Levinson(a1) sont à peut-prêt égales aux valeurs exactes calculées (a) dans question 1 ,avec une erreur de 0.0131.

5 . Spectre autorégressif avec un ordre M=4 :

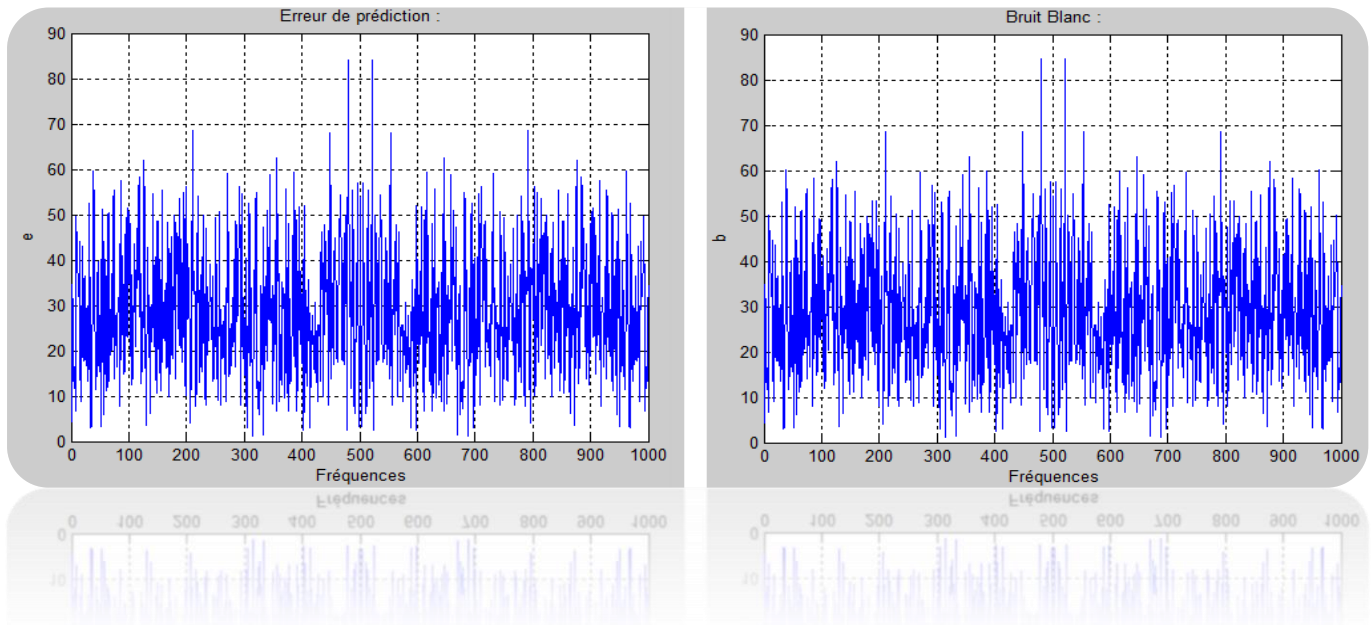
Le spectre autorégressif est donné par la formule : $S(\omega) = \sigma^2 / |1 + \sum_{p=1}^M a(p) e^{i\omega p}|^2$



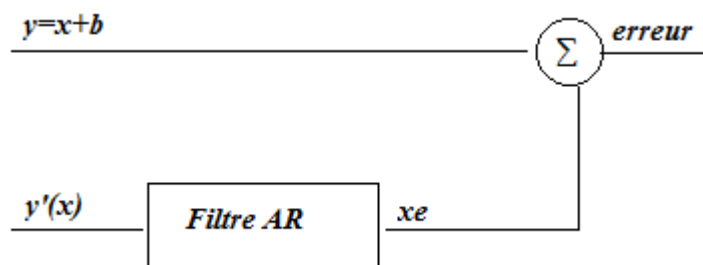
Le spectre AR ne restitue pas les amplitudes de la sinusoïde .En effet, il nous permet pas d'estimer la puissance du signal contrairement au périodogramme.

On remarque que le spectre est plus lisse que si on l'avait tracer en utilisant la transformer de Fourier.

6. Signal d'erreur de prédiction : $e(n)=y(n)-y_e(n)$

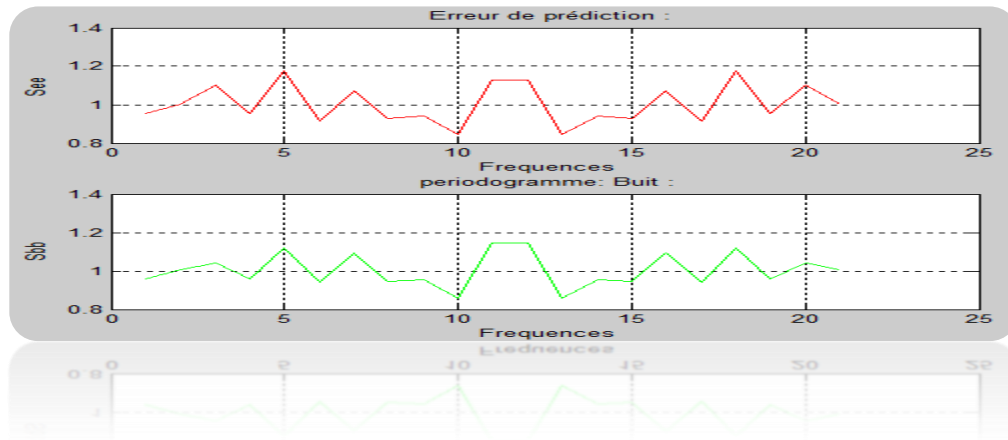


On sait que le spectre d'un bruit blanc contient toutes les fréquences. En d'autres termes, chaque pic représente une fréquence, ceci est illustré par la figure du bruit blanc avec lequel on a bruité le signal sinusoïdale (introduit l'effet aléatoire à la sinusoïde).



On remarque que le spectre du bruit a l'aspect d'un bruit blanc, ce qui est logique puisqu'on estime le signal sinusoïdale et l'erreur= $y-x_e$ = bruit blanc.

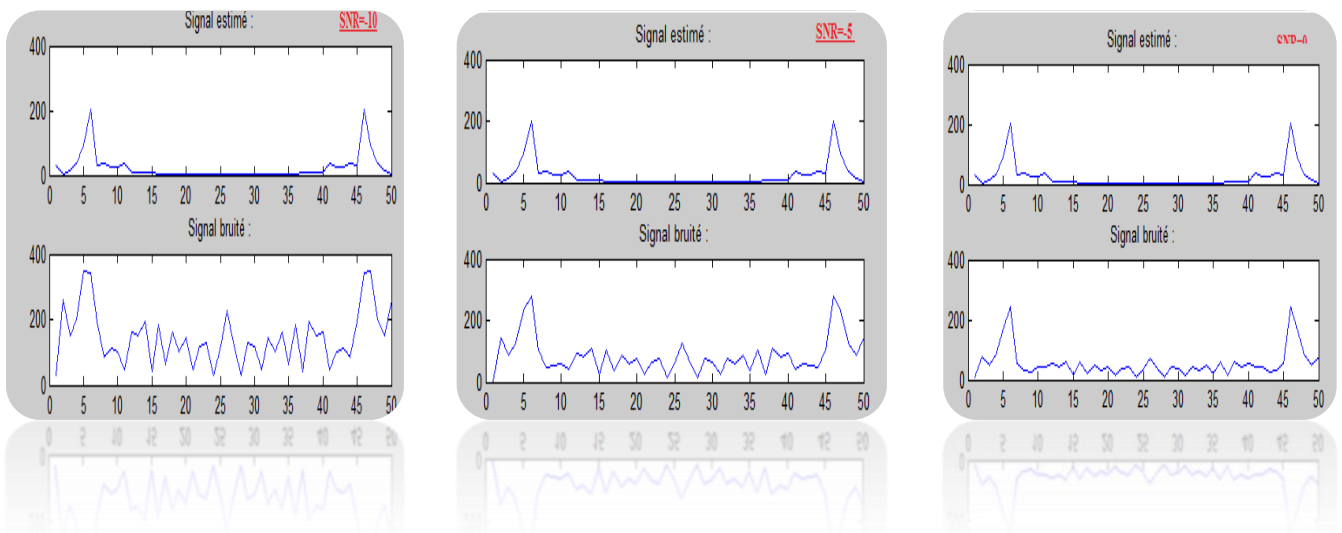
7. Signal d'autocorrélation de l'erreur et du bruit :



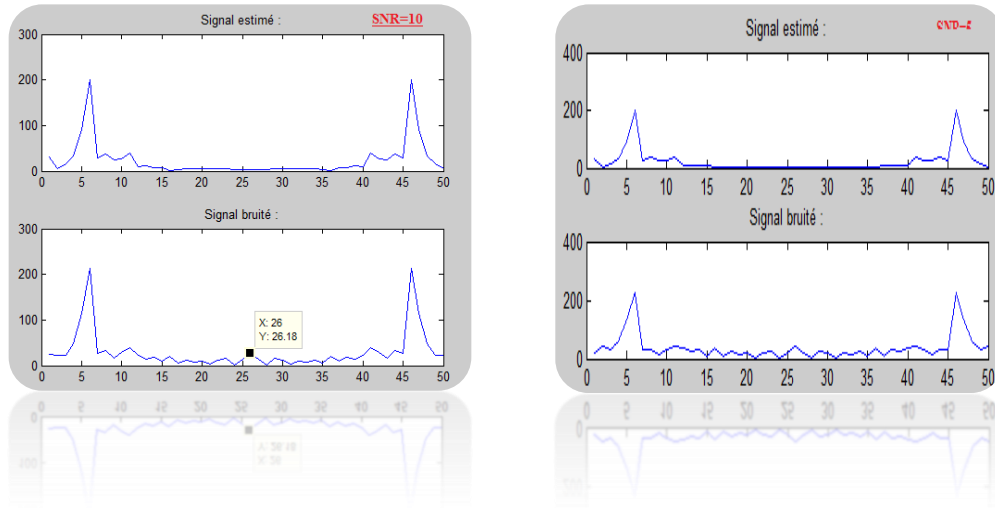
Cette figure accentue les remarques faite dans la question précédente qui montrent que l'erreur quadratique n'est autre que le bruit blanc avec lequel on a bruité la sinusoïde.

8. L'impact du rapport signal sur bruit sur le spectre du signal. :

SNR est le rapport signal/bruit (Signal to Noise Ratio) permettant de mesurer la qualité de transmission .Lorsque le SNR augmente , le bruit devient de plus en plus faible (négligeable) : $SNR = P_s/P_b$ ou $SNR_{db} = 10 \log(P_s/P_b)$.



Pour des SNR faibles (-10/-5/0) on a que la sinusoïde est noyée dans le bruit (bruit fort) .



Pour des SNR important(5/10) le bruit est réduit et négligeable par rapport au signal .

Lorsque SNR augmente le signal est plus présent que le bruit et vice versa. Dans notre cas, le meilleur SNR est 10 car on a les mêmes pics dans le signal estimé que dans le signal bruité , aussi les mêmes amplitudes , même si on remarque l'existence de quelques pics dans la région [15 35] qui sont de faibles amplitudes devant ceux qui sont significatifs. Ces pics montrent qu'on a encore du bruit mais avec une puissance petite vue qu'ils ont des amplitudes faibles (ne dépassent pas 27) . Pour un processus AR , plus le SNR augmente et plus résolution est meilleure .

Conclusion :

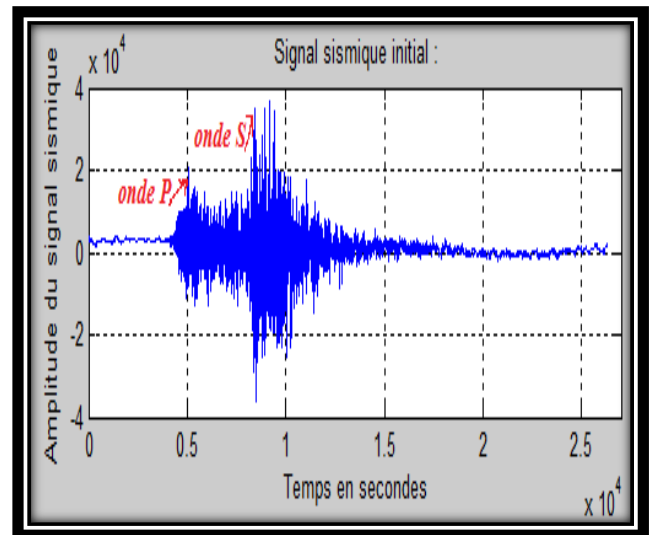
On a pu appliquer le processus Autoregressif linéaire sur un signal sinusoidale bruité qu'on a généré .Les résultats trouvés expérimentalement rejoignent celles théoriques .En effet , les paramètres du filtre AR calculer théoriquement en utilisant la méthode Cramer sont proches de ceux trouvés expérimentalement avec Levinson .On a aussi montrer que le spectre de l'erreur de prédiction n'est autre que le bruit blanc gaussien .On déduit qu'on peu utiliser ces résultats pour faire une extraction d'une sinusoide d'un signal bruité , ceci en considérant que :le signal primaire = le signal bruité , le signal secondaire = bruit extrait dans cette partie .

Au final on a visualisé l'impacte qu'a le SNR sur le spectre : plus le SNR est faible plus la puissance du signal est faible et celle du bruit est forte .

Partie II : Analyse d'un signal sismique :

Partie II : Analyse d'un signal sismique : « Sujet4 »

L'onde P et S caractérisent les ondes de volume, elles se propagent dans la masse terrestre et sont réfléchies /réfractées par les limites des couches de densité différentes et par la surface. Connaissant la célérité de ses ondes ,on peut connaître le lieu ou le séisme s'est produit et la date de l'arrivée du séisme.



Un signal sismique est complexe et aléatoire, il peut être représenté comme une superposition d'ondes de fréquences variées dont l'amplitude est plus au moins importante. Ce signal est enregistré à l'aide des sismographes puis envoyé aux stations centrales pour le traiter, avant de l'envoyer moyennant des réseaux de télécommunications, il faut tout d'abord le coder en utilisant ce qu'on appelle la modélisation paramétrique qui consiste à extraire quelques coefficients représentant le signal au lieu d'envoyer tous les échantillons. C'est en utilisant ces coefficients que la station centrale va restituer le signal sismique, alors comment choisir ces coefficients ?

Tout d'abord, il faut régler le problème de la sélection de modèle qui consiste à trouver l'ordre optimal du modèle AR pour le fixé afin de déterminer ces coefficients.

Différents critères permettent de déterminer l'ordre p optimal à partir du signal à modéliser, tels le FPE (Final Prediction Error), l'AIC (Information Theoretic Criterion) proposés par Akaike. Dans notre cas on va utiliser que la FPE.

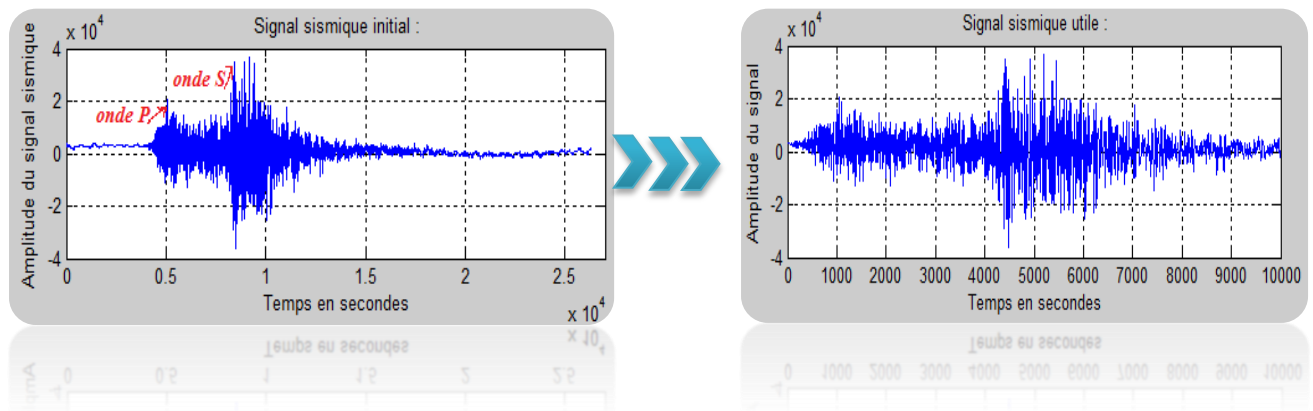
Voici les étapes à suivre pour avoir l'ordre optimale :

- On détermine le modèle paramétrique (AR).
- On prédit alors le signal de synthèse correspondant au modèle trouvé.
- On calcul la FPE et on déduit l'ordre minimale.

➤ Prédiction linéaire et ordre optimale :

La prédiction linéaire consiste à prédire un échantillon à l'instant n à partir de K échantillons précédents .

Dans ce qui suit on considère que la partie du signal importantes càd la partie ou les ondes sismiques sont importantes (4000 :14000) :



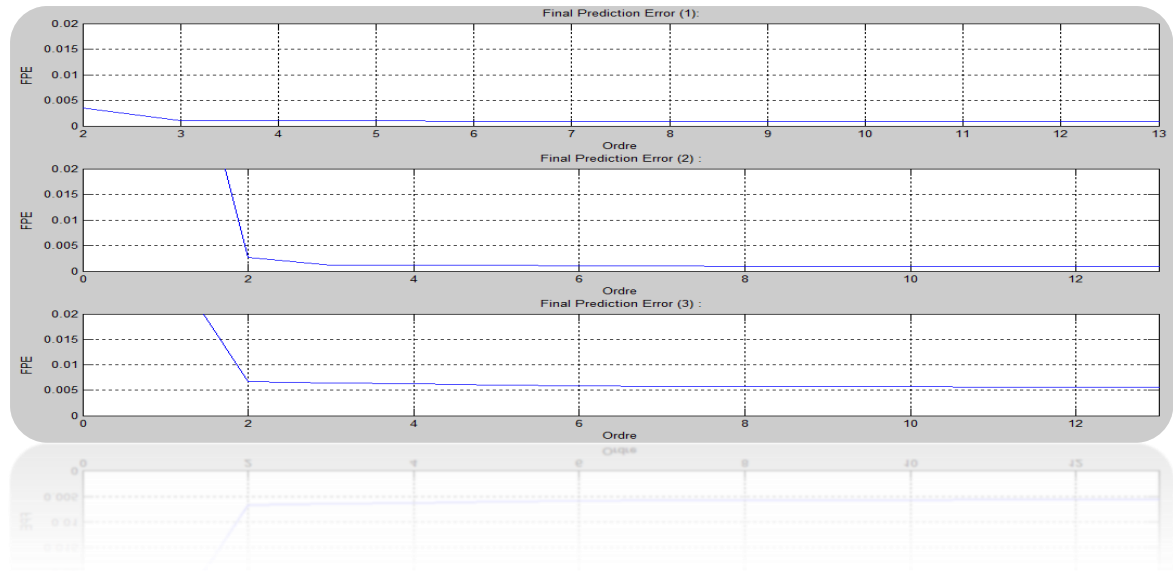
Le signal sismique est un signal continu non stationnaire, pour faire apparaitre l'aspect stationnaire d'un signal on le subdivise à des ségments stationnaires.

En utilisant le critère de AKAIKE (FPE :Final Prediction Error) :

$$FPE(k) = ((N+p+1)/(N-p-1)) * \sigma^2$$

Avec p l'ordre et σ^2 l'estimation de la variance du bruit et N longueur du signal .

La figure ci-dessus montre la FPE en considérant plusieurs blocs du signal utilisé au début au milieu et à la fin.

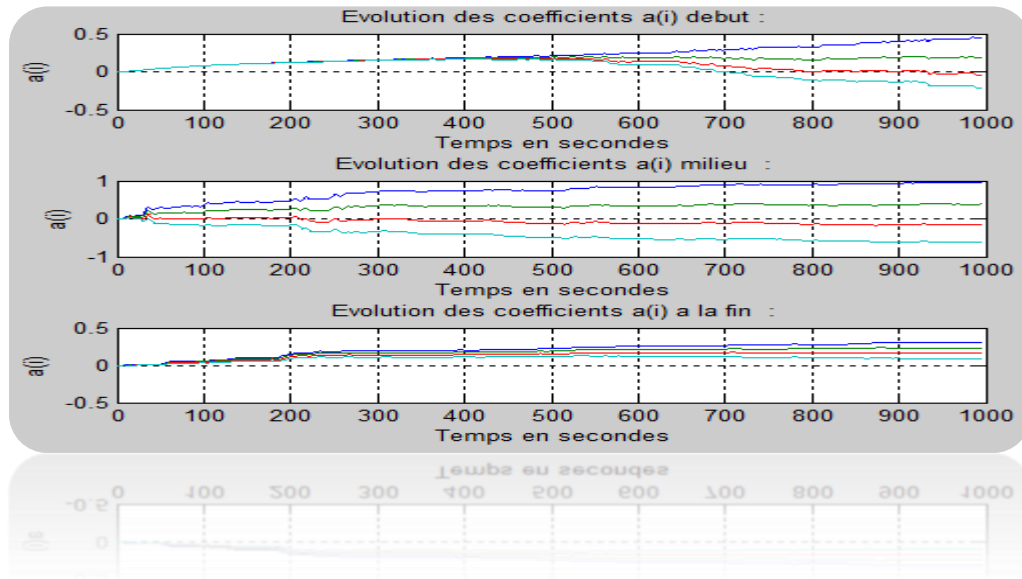


❖ L'ordre optimal est : $p=4$ puisque la FPE est stable (cste) pour des ordres supérieures à 4.

➤ Modélisation du signal sismique par les coefficients d'un filtre autoregressif :

On fixe l'ordre à 4.

Après avoir segmenter le signal utilisé on applique l'algorithme du LMS pour optimiser les paramètres du filtre et minimiser l'erreur de prédiction. On remarque qu'en agissant sur le pas d'adaptation ($\mu=10^{-9}$) les coefficients du modèle évoluent en fonction du temps et tendent à converger rapidement.



On a un filtre d'ordre 4 et donc on s'attend à visualiser une variation de 4 coefficients qui est le cas ici.

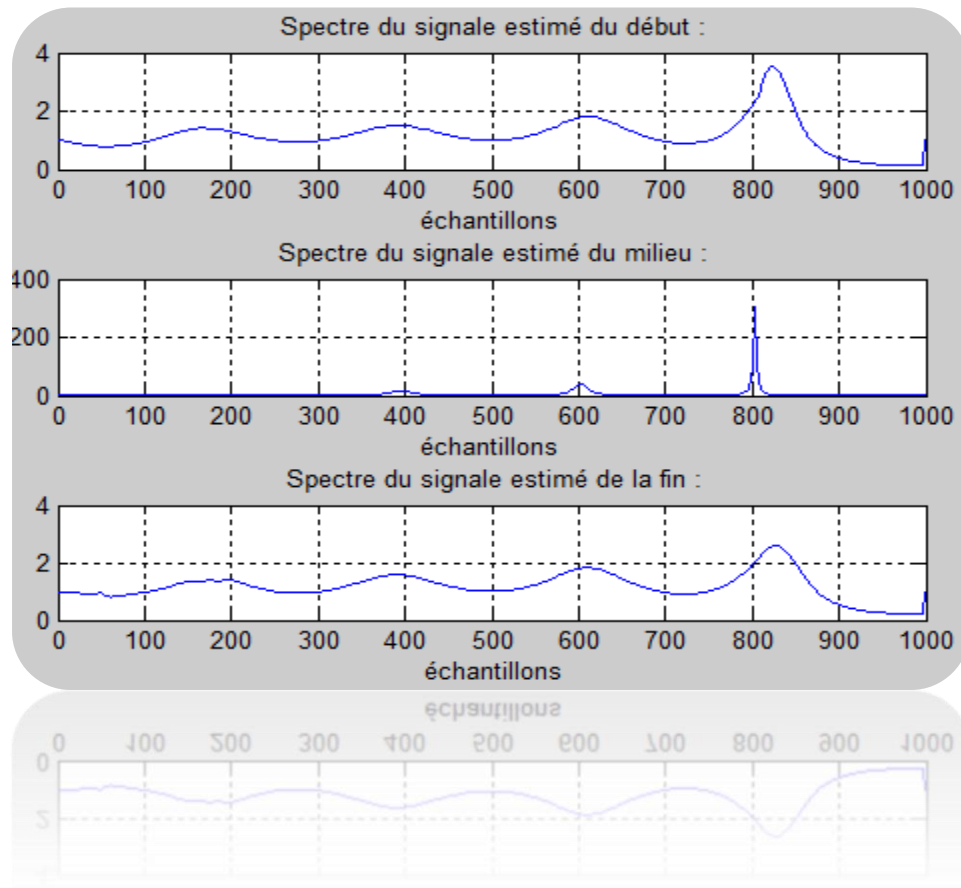
On a qu'au début du signal les coefficients ont la même variance et après un certain temps ou encore après l'onde primaire, ils commencent à varier selon le critère LMS avec des valeurs comprises entre -0.5 et 0.5.

Au milieu du signal on remarque de très fortes variations. En effet, les coefficients prennent des valeurs entre -1 et 1, ce qui est logique puisqu'au milieu d'un signal sismique ou encore après l'onde secondaire (S), on a une énergie très grande, des fluctuations et des tremblements de terre plus fort qu'au début.

A la fin du signal on remarque que la variation des coefficients commencent à diminuer car le domaine de variation retourne comme il l'était au début.

Ceci est logique puisque c'est la fin du tremblement de terre donc la fin des fluctuations et donc le sismographe recommence peu à peu enregistrer des événements normaux.

➤ Spectre de quelques segments du signal :



Au début , le spectre a une énergie maximale au environ de 4 avec des pics assez larges ce qui montre une variation faible jusqu'au dernier pic ou la largeur se rétrécies reflétant le début des fluctuations(séisme) .La faible variation au début est logique puisque le séisme n'est pas encore fort lors de cette partie d'enregistrement.

Au milieu du signal sismique le sismographe a enregistré un signal dont le spectre montre une énergie maximale de 100 avec des pics très minces ce qui montre une forte variation du signal.

A la fin, l'énergie diminue et les pics s'atténuent (energie max =2) et la largeur des pics augmente car le tremblement est entrain de s'adoucir ce qui résulte à des faibles fluctuations .

Conclusion :

La résolution du problème de la sélection du modèle en utilisant la prédiction linéaire ainsi que le critère AKAIKE FPE a abouti à un ordre de filtre AR optimale $p=4$. Après avoir modélisé le signal sismique pour un ordre fixé à p , on a pu déterminer les quatre coefficients qui vont être coder et envoyer par les réseaux de communications à la station centrale qui va les utilisés pour restituer le signal sismique originale afin de le traiter.

Annexe :

❖ Programme partie 1 sous Matlab:

```
close all
clear all
%1)
n=0:1/100:2*pi;
v=cos(n)+i*sin(n);
Z1=0.99*exp(0.2*pi*i);
Z2=0.99*exp(-0.2*pi*i);
Z3=0.99*exp(0.4*pi*i);
Z4=0.99*exp(-0.4*pi*i);
figure
plot(v, 'b')
hold on
grid on
plot(Z1, '--
rs', 'linewidth', 1, 'markersize', 5, 'markerfacecolor', 'g', 'markeredgec
olor', 'k');
hold on
plot(Z2, '--
rs', 'linewidth', 1, 'markersize', 5, 'markerfacecolor', 'g', 'markeredgec
olor', 'k');
hold on
plot(Z3, '--
rs', 'linewidth', 1, 'markersize', 5, 'markerfacecolor', 'g', 'markeredgec
olor', 'k');
hold on
plot(Z4, '--
rs', 'linewidth', 1, 'markersize', 5, 'markerfacecolor', 'g', 'markeredgec
olor', 'k');
title(' Position des zéros dans le plan complexe :');
m=[(1/Z1) ((1/Z1).^2) ((1/Z1).^3) ((1/Z1).^4); (1/Z2) ((1/Z2).^2)
((1/Z2).^3) ((1/Z2).^4); (1/Z3) ((1/Z3).^2) ((1/Z3).^3)
((1/Z3).^4); (1/Z4) ((1/Z4).^2) ((1/Z4).^3) ((1/Z4).^4)];
a=inv(m)*[-1;-1;-1;-1];
%2)
N=1000;
b=randn(1,N); % Bruit blanc de moyen 0 et de varriance 1 .
y=zeros(1,N);
for k=1:N;
    if k<5
        som=0;
        for m=1:k-1
            som=som-a(m)*y(k-m)
        end
    end
end
```

```

        else
            som=0;
            for m=1:4
                som=som-a(m)*y(k-m);
            end
        end
        y(k)=b(k)+som;
    end
    tft=abs(fft(y));
    figure
    subplot(121)
    plot(tft);
    title(' Signal initial (total=1000 pts) :')
    xlabel(' Frequences')
    ylabel(' Spectre de y(n) ')
    grid on

    subplot(122)
    signale=y(N-49:N);
    tf=abs(fft(signale));
    w=1:N;
    plot(w(1:25),tf(1:25));
    plot(tf);
    title(' Signal initial (50 derniers pts) :')
    xlabel(' Frequences')
    ylabel(' Spectre de y(n) ')
    grid on
%3)
    figure
    r=autocorr(signale);
    ft1=fft(r);
    plot(abs(ft1))
    title(' Spectre Autocorrélation de y(n) (50 derniers pts) ')
    xlabel(' Frequences')
    ylabel(' Syy ')
    grid on
%4)
    [a1,e]=levinson(r,4);
%5)
    Ne=50;
    S=zeros(1,Ne);
    for m=1:Ne
        som=1;
        for j=1:4
            som=som+a1(j)*exp(1i*(2*pi/Ne)*m*j);
        end
        S(m)=e/(som*conj(som));
    end
    figure
    plot(S,'g');

```



```

title(' Spectre autorégressif: ')
xlabel(' Frequences')
ylabel(' S(w) ')
grid on
%6)
y3=signale
N=50 %N=1000 pour tout le signal bruité
ye=zeros(1,N);
ye(1)=0;
ye(2)=-a1(2)*y3(1);
ye(3)=-a1(2)*y3(2)-a1(3)*y3(1);
ye(4)=-a1(2)*y3(3)-a1(3)*y3(2)-a1(4)*y3(1);
for k=5:N
    ye(k)=-a1(2)*y3(k-1)-a1(3)*y3(k-2)-a1(4)*y3(k-3)-a1(5)*y3(k-4);
end
erreur=y3-ye;
figure
ff1=abs(fft(erreur));
plot(ff1)
title(' Erreur de prédiction : ')
xlabel(' Frequences')
ylabel(' e')
grid on
%pour tous le signal bruité
%erreure=y-ye;
%figure
%ff1=abs(fft(erreure));
%plot(ff1)
%title(' Erreur de prédiction : ')
%xlabel(' Fréquences')
%ylabel(' e')
%grid on
%figure
%ffb=abs(fft(b));
%plot(ffb)
%title(' Bruit Blanc : ')
%xlabel(' Fréquences')
%ylabel(' b')
%grid on

%7)
r1=autocorr(erreur);
r2=autocorr(b);
ff2=abs(fft(r1));
ff3=abs(fft(r2));
figure
subplot(211)
plot(ff2,'r')
title(' Erreur de prédiction : ')
xlabel(' Frequences')

```

```

ylabel(' See ')
grid on
subplot(212)
plot(ff3,'g')
title(' periodogramme: Buit : ')
xlabel(' Frequences')
ylabel(' Sbb')
grid on
%8) snr
[H,sigma]=levinson(y3,6);
ye6=zeros(1,50);
ye6(1)=0;
ye6(2)=-H(2)*y3(1);
ye6(3)=-H(2)*y3(2)-H(3)*y3(1);
ye6(4)=-H(2)*y3(3)-H(3)*y3(2)-H(4)*y3(1);
ye6(5)=-H(2)*y3(4)-H(3)*y3(3)-H(4)*y3(2)-H(5)*y3(1);
ye6(6)=-H(2)*y3(5)-H(3)*y3(4)-H(4)*y3(3)-H(5)*y3(2)-H(6)*y3(1);
for k=7:N
    ye6(k)=-H(2)*y3(k-1)-H(3)*y3(k-2)-H(4)*y3(k-3)-H(5)*y3(k-4)-
H(6)*y3(k-5)-H(7)*y3(k-6);
end
psignal=mean(abs(ye6).^2);
snr=-10:5:10;
for e=1:length(snr)
    sig(e)=sqrt(psignal/(10^(snr(e)/10)));
    bruit(e,:)=sig(e);
    pbruit=mean(abs(bruit(e,:).^2));
    sigbruit(e,:)=y3+bruit(e,:);
    figure
    subplot (211)
    plot(abs(fft(ye6)))
    title(' Signal a estimé : ')
    subplot(212)
    plot(abs(fft(sigbruit(e,:))))
    title(' Signal bruité : ')
end

```

❖ Programme partie 2 sous Matlab:

```

close all
clc
% Partie 1 :
y=load('sujet4.txt');
N=length(y);
plot(y);
title(' Signal sismique initial :')
xlabel(' Temps en secondes')
ylabel(' Amplitude du signal sismique ')
grid on
axis([0 27000 -40000 40000]);
x1=y(4000:13999);

```

```

step=N/10; % Pas de la segmentation .
x1=reshape(y,step,N/step); % Segmentation du Signal sismique
initial .
No=20;
FPE=zeros (No,N/step); % Final Prediction Error
for i=1:N/step
    fpe=zeros (1,No);
    for M=1:No
        rxx=autocorr(x1(:,i));
        [a e]=levinson(rxx,M); % déterminer le modèle
paramétrique (AR)
        n=length(x1(:,i));
        fpe(M)=e*(N+M+1)/(N-M-1);
    end
    FPE(:,i)=fpe;
end
figure
subplot(311)
plot(FPE(:,2));
title(' Final Prediction Error (1):')
xlabel(' Ordre ')
ylabel(' FPE ')
grid on
axis([2 13 0 0.02]);
subplot(312)
plot(FPE(:,3));
title(' Final Prediction Error (2) :')
xlabel(' Ordre ')
ylabel(' FPE ')
grid on
axis([0 13 0 0.02]);
subplot(313)
plot(FPE(:,5));
title(' Final Prediction Error (3) :')
xlabel(' Ordre ')
ylabel(' FPE ')
grid on
axis([0 13 0 0.02]);
% Partie 2 :
ordre=4;
figure
subplot(211)
plot(y);
title(' Signal sismique initial :')
xlabel(' Temps en secondes')
ylabel(' Amplitude du signal sismique ')
grid on
axis([0 27000 -40000 40000]);
x1=zeros(1,1000);
x1=y(4000:13999);

```

```

N=length(x1);
step=N/10;
seg=reshape(x1,step,N/step);
subplot(212)
plot(x1);
title(' Signal sismique utile :')
xlabel(' Temps en secondes')
ylabel(' Amplitude du signal ')
grid on
mu=0.0000000001; % Pas d'adaptation .
[H1 Y1lms]=lms(seg(:,1)',ordre,mu); % LMS : Lean Mean Square
[H2 Y2lms]=lms(seg(:,7)',ordre,mu); % LMS : Lean Mean Square
[H3 Y3lms]=lms(seg(:,9)',ordre,mu); % LMS : Lean Mean Square
[S1]=sp(H1,step,ordre) % Spectre
[S2]=sp(H2,step,ordre)
[S3]=sp(H3,step,ordre)
figure
subplot(311)
plot(H1);
title(' Evolution des coefficients a(i) debut :')
xlabel(' Temps en secondes')
ylabel(' a(i) ')
grid on
subplot(312)
plot(H2);
title(' Evolution des coefficients a(i) milieu :')
xlabel(' Temps en secondes')
ylabel(' a(i) ')
grid on
subplot(313)
plot(H3);
title(' Evolution des coefficients a(i) a la fin :')
xlabel(' Temps en secondes')
ylabel(' a(i) ')
grid on
figure
subplot(311)
plot(S1);
title(' Spectre du signale estimé du début :')
xlabel(' échantillons ')
grid on
subplot(312)
plot(S2);
title(' Spectre du signale estimé du milieu :')
xlabel(' échantillons ')
grid on
subplot(313)
plot(S3);
title(' Spectre du signale estimé de la fin :')
xlabel(' échantillons')

```

grid on

Avec une fonction pour LMS :

```
function [h, ye, e] = lms(y, ordre, mu)
xi = y(1:length(y)-ordre); % Signal d'entrée .
xr = y(ordre+1:length(y)); % Signal de référence .
N = length(xi);
h = zeros(N+1, ordre); % Vecteur des coeffisicients de prédiction .
e(ordre) = 0;
for i = ordre+1:N
    ye(i) = 0;
    for j = 1:ordre
        ye(i) = ye(i) + h(i, j)' * xi(1, i-j+1); % Estimation du signal .
        h(i+1, j) = h(i, j) + mu * e(i-1) * xi(1, i-j+1); % Adaptation des
coeffiscent h .
    end
    e(i) = xr(1, i) - ye(i); % Erreur de prédiction .
end end
```

Avec une fonction sp qui calcul le spectre AR :

```
function [S] = sp(h, step, ordre)
Ne = step;
S = zeros(1, Ne);
for m = 1:Ne
    som = 1;
    for j = 1:ordre
        som = som + h(m) * exp(-1i * (2*pi/Ne) * m * j);
    end
    S(m) = 1 / (som * conj(som));
end
end
```