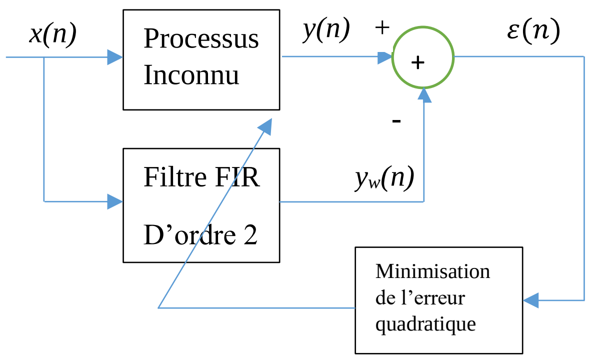
ESIEE AMIENS	FILTRAGE I4 **Travaux dirigés** : n°1 **Remis par** : N. ZITOUNI	**Année** : 2014-2015 **Date** :28-08-14

Le but de ce TD est d'étudier analytiquement, dans un premier temps, le filtrage adaptatif appliqué à la suppression d'interférences, puis dans un second temps, expérimentalement à travers la simulation sur matlab.

EXERCICE I

On donne le système suivant :



1- Exprimer l'expression de l'échantillon du signal estimé $y_w(n)$ en fonction des échantions x(n) et des coefficients du filtre FIR.
2- Exprimer la puissance moyenne de l'erreur quadratique $\varepsilon\ (n)$ calculée sur N points
3- Trouver les coefficients w_i qui minimisent la fonction coût $J(\varepsilon)=E\{\varepsilon^2(n)\}$

EXERCICE II

On considère un problème de soustraction de bruit où u(n) est la référence bruit en entrée du filtre de Wiener. Le signal observé est x(n) = s(n) + b(n). On suppose par ailleurs que l'on dispose d'un lien entre *u(n)* et $\hat{b}(n)$: $\hat{b}(n) = \sum_{i=0}^{P-1} \alpha_i u(n-i)$

1. Donner le schéma synoptique du système et exprimer le vecteur d'intercorrélation Rux en fonction de la matrice de corrélation Ruu et du vecteur des coefficients α,

2. Quel est le filtre de Wiener optimal?

3. Donner l'algorithme du gradient.

4. Quel est le pas optimal théorique μ? Donner une valeur pratique dans le cas où u(n) est un bruit blanc de variance σ2 ?

EXERCICE III

Dans le but de vous familiariser avec les résultats de la corrélation, appliquez la fonction de corrélation

R_{xy}=*xcorr(y,x,L,'unbiased')* à chacun des trois signaux suivants :

x_c=sin(2πn/N) , x_q=square(2πn/N) , x_g=randn(size(n)) ;

où N=50 et n=0 :1000. Pour faire :

1. Calculer et tracer chaque signal dans une fenêtre.
2. Calculer et tracer chaque fonction d'autocorrélation dans une autre fenêtre à côté du signal correspondant.
3. Quelle est l'utilité du paramètre L ?
4. Que doit être le maximum de chaque fonction d'autocorrélation ?
5. Calculer et tracer l'intercorrélation entre x_c et x_q. Prenez garde à l'ordre des signaux.

EXERCICE IV

On souhaite extraire un signal inconnu s(n) fortement perturbé par le réseau électrique de fréquence 50Hz. Pour ce faire, on a mesuré simultanément le signal du réseau x(n) et le signal bruité y(n) en les échantillonnant à la fréquence f_e=10kHz. Pour résoudre ce problème,

1. Charger les signaux contenus dans le fichier xy50hz.txt
   ```
   Signaux=load('xy50hz.txt') ;
   xn=signaux( :,1) ;
   yn=signaux( :,2) ;
   ```
2. Tracez x(n) et y(n)
3. Dessiner le schéma de Wiener ; où se trouve le signal s(n) ?
4. Rechercher s(n) en appliquant l'algorithme de Wiener Hopf avec 2 paramètres.
5. Augmenter le nombre de paramètres P ; Observer leur valeurs et la puissance de s(n) ; Conclure ;
6. Calculer le rapport signal sur bruit : S_{eff}/y_{eff}.

Solution :EXO1

1- *La fonction de transfert du filtre FIR est :*

$$H(z) = \sum_{i=0}^{2} w_i z^{-i}$$

Donc l'expression de $y_w(n)$ est :

$$y_w(n) = \sum_{i=0}^{2} w_i x(n-i) = w_o x(n) + w_1 x(n-1) + w_2 x(n-2)$$

2- $\varepsilon^2(n) = (y(n) - y_w(n))^2 = (y(n) - w_0 x(n) - w_1 x(n-1) - w_2 x(n-2))^2$

La fonction coût vaut :

$$J(\varepsilon) = E\{\varepsilon^2(n)\} = \frac{1}{N}\sum_{n=0}^{N-1} \varepsilon^2(n) = \frac{1}{N}\sum_{n=0}^{N-1}(y(n) - w_0 x(n) - w_1 x(n-1) - w_2 x(n-2))^2$$

w_0, w_1 et w_2 qui minimisent la fonction coût s'obtiennent comme suit :

$$\frac{\partial J(\varepsilon)}{\partial w_0} = \frac{2}{N}\sum_{n=0}^{N-1}\big(y(n) - w_0 x(n) - w_1 x(n-1) - w_2 x(n-2)\big)\big(-x(n)\big) = 0$$
$$\frac{\partial J(\varepsilon)}{\partial w_1} = \frac{2}{N}\sum_{n=0}^{N-1}\big(y(n) - w_0 x(n) - w_1 x(n-1) - w_2 x(n-2)\big)\big(-x(n-1)\big) = 0$$
$$\frac{\partial J(\varepsilon)}{\partial w_2} = \frac{2}{N}\sum_{n=0}^{N-1}\big(y(n) - w_0 x(n) - w_1 x(n-1) - w_2 x(n-2)\big)\big(-x(n-2)\big) = 0$$

Ce qui donne :

$$\frac{1}{N}\sum_{n=0}^{N-1} y(n)x(n) = w_0\frac{1}{N}\sum_{n=0}^{N-1} x(n)x(n) + w_1\frac{1}{N}\sum_{n=0}^{N-1} x(n)x(n-1) + w_2\frac{1}{N}\sum_{n=0}^{N-1} x(n)x(n-2)$$

$$\frac{1}{N}\sum_{n=0}^{N-1} y(n)x(n-1) = w_0\frac{1}{N}\sum_{n=0}^{N-1} x(n)x(n-1) + w_1\frac{1}{N}\sum_{n=0}^{N-1} x(n-1)x(n-1) + w_2\frac{1}{N}\sum_{n=0}^{N-1} x(n-1)x(n-2)$$

$$\frac{1}{N}\sum_{n=0}^{N-1} y(n)x(n-2) = w_0\frac{1}{N}\sum_{n=0}^{N-1} x(n)x(n-2) + w_1\frac{1}{N}\sum_{n=0}^{N-1} x(n-1)x(n-2) + w_2\frac{1}{N}\sum_{n=0}^{N-1} x(n-2)x(n-2)$$

Donc :

$$R_{yx}(0) = w_0 R_{xx}(0) + w_1 R_{xx}(1) + w_2 R_{xx}(2)$$
$$R_{yx}(1) = w_0 R_{xx}(1) + w_1 R_{xx}(0) + w_2 R_{xx}(1)$$
$$R_{yx}(2) = w_0 R_{xx}(2) + w_1 R_{xx}(1) + w_2 R_{xx}(0)$$

$R_{xx}(i) = R_{xx}(-i)$: *La fonction d'autocorrélation est une fonction paire*

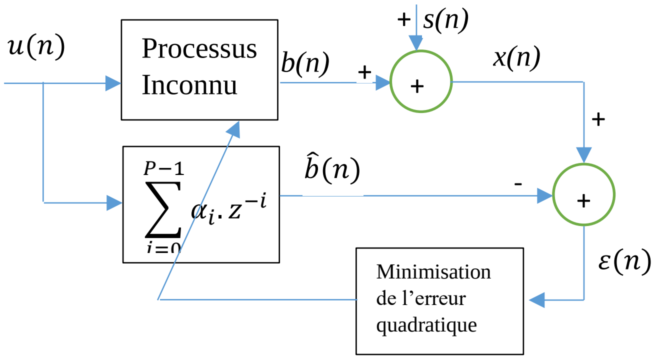
Donc :

$$\begin{pmatrix} R_{yx}(0) \\ R_{yx}(1) \\ R_{yx}(2) \end{pmatrix} = \begin{pmatrix} R_{xx}(0) & R_{xx}(1) & R_{xx}(2) \\ R_{xx}(1) & R_{xx}(0) & R_{xx}(1) \\ R_{xx}(2) & R_{xx}(1) & R_{xx}(0) \end{pmatrix} . \begin{pmatrix} w_0 \\ w_1 \\ w_2 \end{pmatrix}$$

Ce qui donne $\begin{pmatrix} w_0 \\ w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} R_{xx}(0) & R_{xx}(1) & R_{xx}(2) \\ R_{xx}(1) & R_{xx}(0) & R_{xx}(1) \\ R_{xx}(2) & R_{xx}(1) & R_{xx}(0) \end{pmatrix}^{-1} . \begin{pmatrix} R_{yx}(0) \\ R_{yx}(1) \\ R_{yx}(2) \end{pmatrix}$

Solution EXOII

1-Le schéma synoptique du système est :



2- Le filtre de Wiener optimal est donné par l'équation : $\alpha = R_{uu}^{-1}.R_{xu}$

Avec $\boldsymbol{\alpha}$ *vecteur de dimension P et* R_{uu} *est la matrice d'autocorrélation de u*

$$R_{uu}(i) = \frac{1}{N}\sum_{n=0}^{N-1} x(n-1)x(n-i)$$

Et

$$R_{xu}(i) = \frac{1}{N}\sum_{n=0}^{N-1} x(n-1)u(n-i)$$

3- L'algorithme du gradient (appelé aussi RLMS : Recursive Least Mean Square) permet de calculer le vecteur $\boldsymbol{\alpha}$ *des coefficients du filtre adaptatif. Il est donné par :*

$$\boldsymbol{\alpha_n = \alpha_{n-1} + \mu(x_n - U_n^T.\alpha_{n-1})U_n^T}$$

Avec

* $\boldsymbol{\alpha_n}$ *vecteur des coefficients du filtre adaptatif calculés à l'instant nTe :* $\alpha_n = \begin{pmatrix} \alpha_0 \\ \cdots \\ \alpha_{P-1} \end{pmatrix}$.

* $\boldsymbol{x_n}$*, dernier échantillon mesuré en sortie.*

* $\boldsymbol{U_n^T}$ *est le transposé du vecteur d'entrée.* $\boldsymbol{U_n^T = [u_n\ u_{n-1}\ u_{n-2}\ ..\ u_{n-P+1}]}$

* $\boldsymbol{\mu}$ *est le pas d'adaptation*

*$\boldsymbol{x_n - U_n^T.\alpha_{n-1}}$ *est l'erreur de prédiction à priori.*

4- La valeur optimale théorique du pas d'adaptation doit être $\boldsymbol{0 < \mu < \frac{2}{P\sigma_x^2}}$

P : L'ordre du filtre

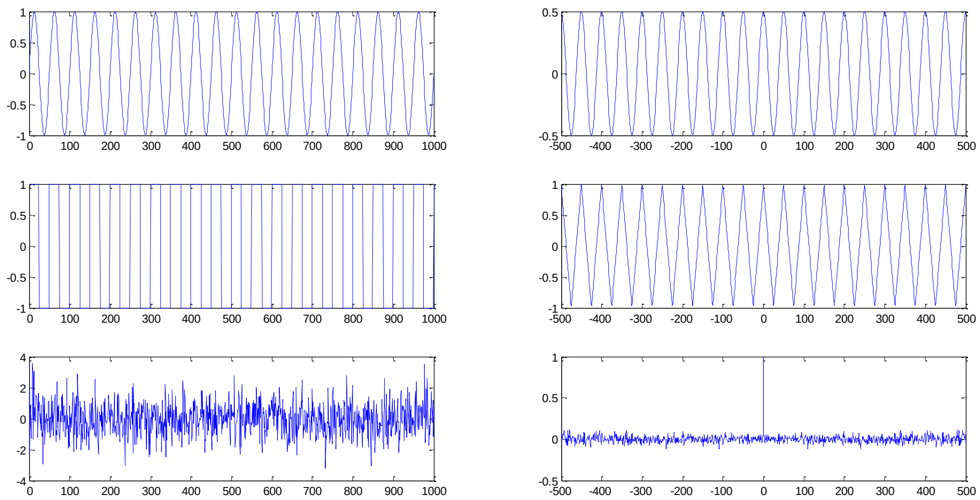
$\boldsymbol{\sigma_u^2}$ *: La variance du signal d'entrée c'est-à-dire sa puissance moyenne.*

Valeur pratique $\boldsymbol{\mu = \frac{\mu_0}{P\sigma^2}}$ *Avec* $\boldsymbol{\mu_0 \approx (0.01, \ldots, 0.01)}$

Solution EXOIII

```
n=0:1000;
    N=50;
%calcul des signaux
    xc=sin(2*pi*n/N);
    xq=square(2*pi*n/N);
    xg=randn(size(n));
%tracé des signaux
    subplot(321),plot(n,xc);
    subplot(323),plot(n,xq);
```

```
    subplot(325),plot(n,xg);
%caclcul des autocorrélations
    [Rxcxc,tau]=xcorr(xc,500,'unbiased');
    [Rxqxq,tau]=xcorr(xq,500,'unbiased');
    [Rxgxg,tau]=xcorr(xg,500,'unbiased');
%tracé des autocorrélations
    subplot(322),plot(tau,Rxcxc);
    subplot(324),plot(tau,Rxqxq);
    subplot(326),plot(tau,Rxgxg);
%comparaison de la puissance et de l'autocorrélation en zéro
      Rxgxg(501)
    sum(xg.^2)/length(n)   %/ L'autocorrélation en zéro vaut la puissance
                             moyenne du signal
```



```
%calcul des intercorrélations
    [Rxcxq,tau]=xcorr(xc,xq,500,'unbiased');
    [Rxqxc,tau]=xcorr(xq,xc,500,'unbiased');
%tracé des intercorrélations
    subplot(322),plot(tau,Rxcxq);
    subplot(324),plot(tau,Rxqxc);
```

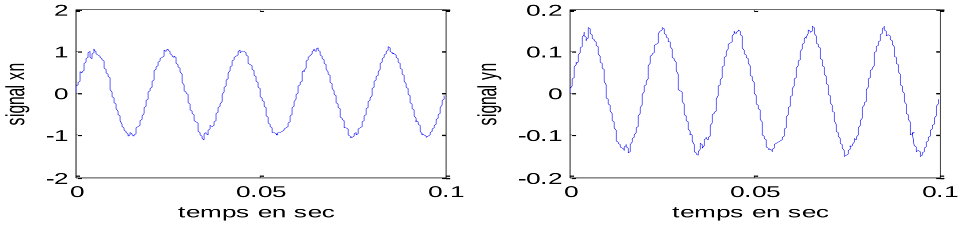
Solution EXO IV

1.

```
signaux=load('C:\MATLAB\Travail\xy50hz.txt');
xn=signaux(:,1);
yn=signaux(:,2);
```

2.

```
fe=10000;
N=length(xn);
n=0:N-1;
t=n/fe;
subplot(221),plot(t,xn);
subplot(222),plot(t,yn);
```

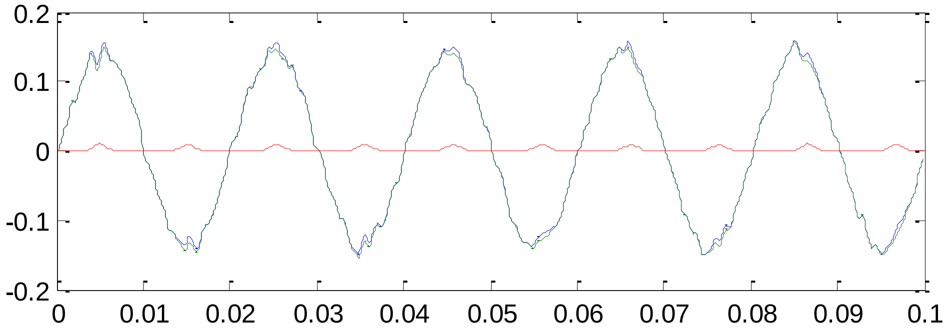


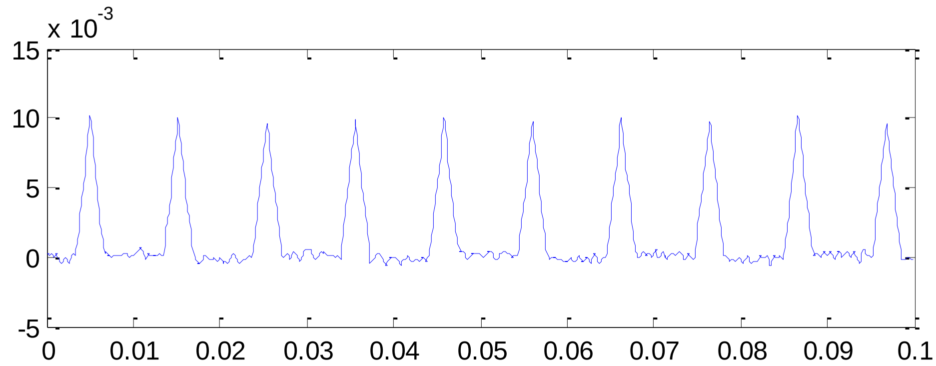
3.

```
%                                  sn
%           |================|  yw   |
%   xn=====>|proces inconnu |====>+===|
%       |   |================|         |yn
%       |                              +====|
%       |    |=======|==========|^yw   |     |
%       ====>|filtre |de Wiener |=====|     |epsilon(n)
%            |=======|==========|           |
%                    |=====================|
%
% Dans le cas optimal, le signal erreur epsilon sera égal à sn ( signal
% recherché). dans ce cas W=inv(Rxx)*Ryx  et sn=epsilon=yn-yw
%
```

4.

```
x=xn;
y=yn;
P=10 ;
[Rxx,tau1]=xcorr(x,x,P-1,'none');
[Ryx,tau2]=xcorr(y,x,P-1,'none');
Rxx=Rxx(max(tau1)+1:2*max(tau1)+1);
Rxx=toeplitz(Rxx);
Ryx=Ryx(max(tau2)+1:2*max(tau2)+1);
W=inv(Rxx)*Ryx
yw=filter(W,[1,0],x);
s=y-yw;
clg ;
subplot(211),plot(t,y,t,yw);
subplot(211),plot(t,s);
```






```
x=xn;
y=yn;
Seff=((1/N)*sum(s.^2))
Yeff=((1/N)*sum(y.^2))

RSN=Seff/Yeff
```