11/12/2016

Compte rendu TP2 Estimation Stochastique

Estimation par maximum de vraisemblance moindres carrés

Aurélien Bernier Levalois & Yoann Fleytoux
2A SRI

Modélisation

1) On choisit un estimateur $\widehat{H}[n] = Y[n]$

On obtient donc un biais de $\widehat{H}[n]$ nul et une variance = 0.4, soit la même que le bruit.

2) Le modèle est linéaire en les paramètres inconnus.

3) Y = ah + B avec Y (N,1), A (N,M), h(M,1) et B(N,1)

Avec A =

x[0]	0	.	0
.	.	.	.
.	.	x[0]	0
x[M-1]	.	x[1]	x[0]
.	.	.	x[1]
.	.	.	0
x[N-1]	x[N-2]	.	x[N-M]

4) L'estimateur $\hat{h}$ du maximum de vraisemblance de h :

$$\widehat{H}mle = argmaxhpy|h = \left(A^T \frac{1}{\sigma^2} A\right)^{-1} A^T \frac{1}{\sigma^2} Y$$

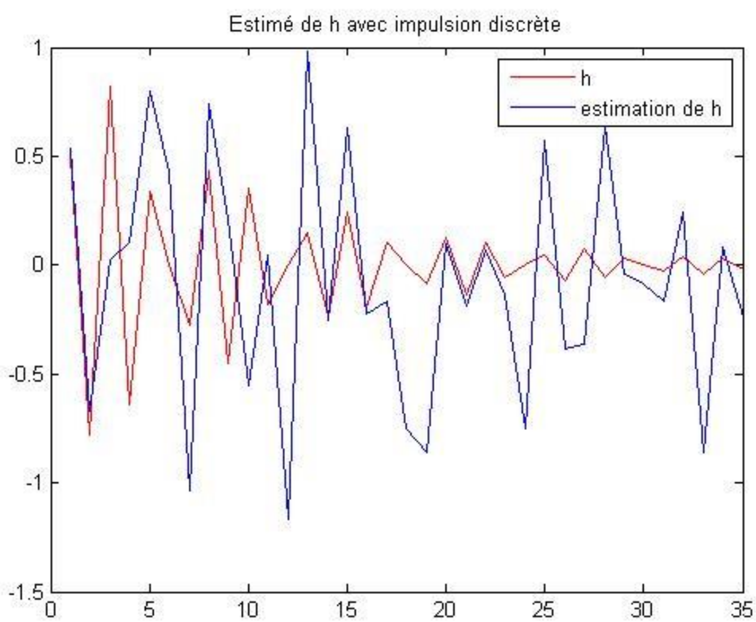
$$Biais\ \widehat{H}mle = 0$$

$$Cov\ \widehat{H}mle = \left(A^T \frac{1}{\sigma^2} A\right)^{-1}$$

Implémentation et analyse sous MATLAB

Estimateur pour une entrée impulsionnelle

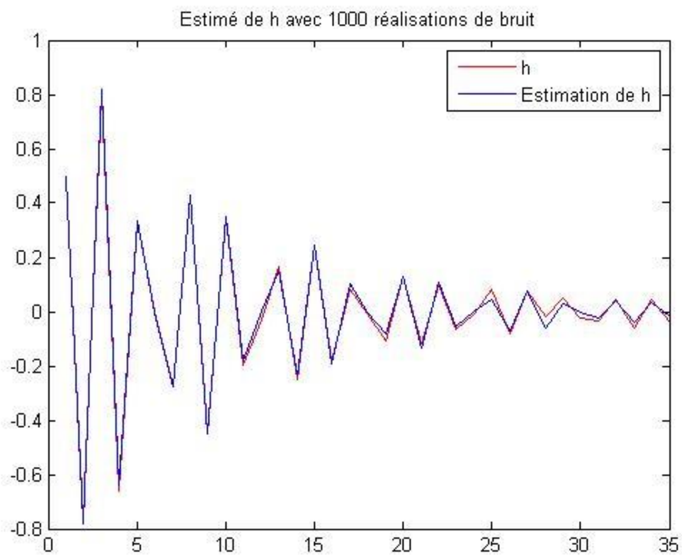
```
%x = signal d'entree
%h = réponse impulsionnelle
load data_MC.mat

%5)
N = length(x);
M = length(h);
delta = zeros(M, 1);
delta(1,1) = 1;
% Calcul de la sortie bruitée
y = syst(delta, h)
estim_h = y(1:M)

%6)
plot(h, 'r');
hold on;
plot(estim_h);
hold off;
title('Estimé de h avec impulsion discrète')
legend('h','estimation de h');

dqn = (1/M)*sum((abs(estim_h-h).^2))
%Pour info : dqn = 0.4482

figure;
```



L'estimation est relativement décevante, on est assez loin de 0.

```
%7)
Nr = 1000;
for i = 1:Nr
    y(:,i) = syst(delta,h);
end
```

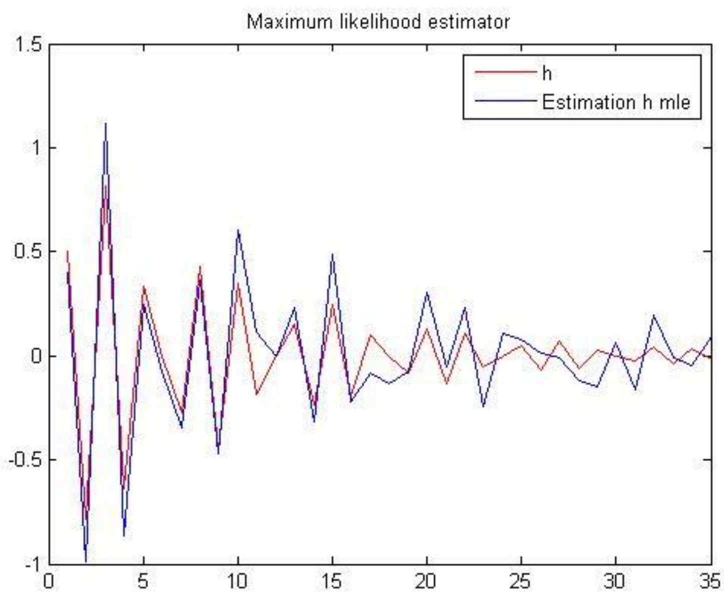
```
estim_h = zeros(M, Nr);
for j = 1:M
    estim_h(j,:) = y(j,:);
end
hist(estim_h(1,1:Nr),100)
m = sum(estim_h,2)/Nr;
mean_emp = mean(m)
var_emp = var(m)
plot(m,'r');
hold on;
plot(h);
hold off;
dqn2 = (1/M)*sum((abs(m-h).^2))
title('Estimé de h avec 1000 réalisations de bruit')
legend('h', 'Estimation de h')
```



L'estimation est ici beaucoup plus précise, les courbes sont presques confondues.

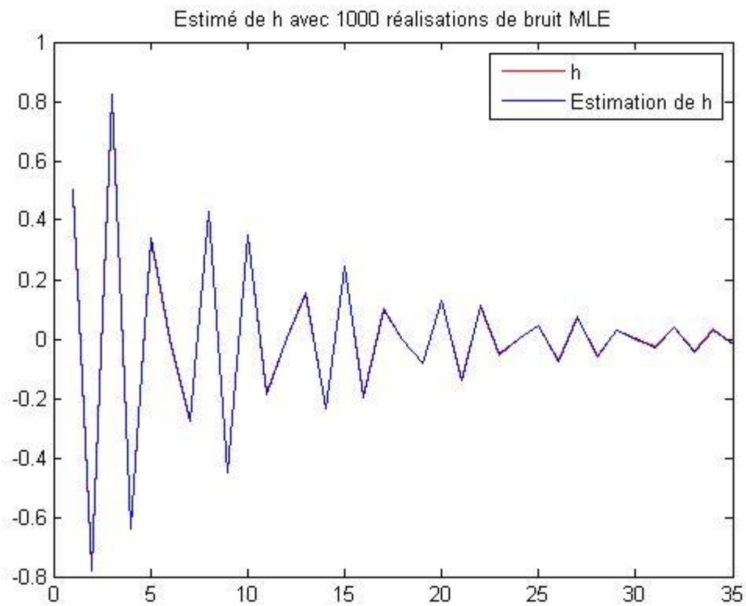
Estimateur du maximum de vraisemblance – Estimateur par moindres carrés

```
%8)
figure;
ylikely = syst(x,h)
likelihood = zeros(1, M)
likelihood(1,1) = x(1,1);
A = toeplitz(x, likelihood);
h_mle = (inv(transpose(A)*A)) * transpose(A) * ylikely;

%9)
plot(h, 'r');
hold on;
plot(h_mle);
hold off;
title('Maximum likelihood estimator');
legend('h', 'Estimation h mle');
dqn = (1/M)*sum((abs(h_mle - h).^2))
%Pour info : dqn = 0.0295
```



Cette estimation est décevante comparée à la question 7, mais le résultat n'est pas trop loin.

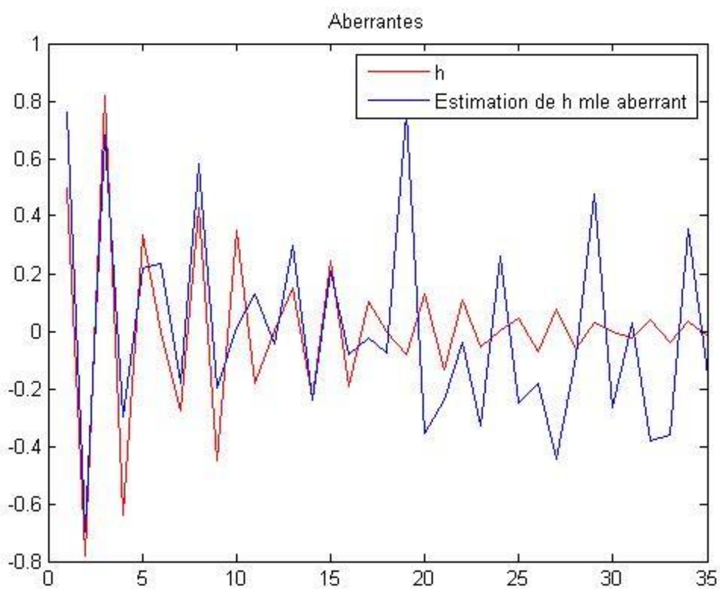
```
%10)
for i = 1:Nr
    ylikely(:,i) = syst(x,h);
end
estim_h_likelihood = (inv(transpose(A)*A)) * transpose(A) * ylikely;
m_likelihood = sum(estim_h_likelihood,2)/Nr;
m_likelihood_emp = mean(m_likelihood);
var_likelihood_emp = var(m_likelihood);

figure;
plot(h,'r');
hold on;
plot(m_likelihood);
hold off;
title('Estimé de h avec 1000 réalisations de bruit MLE')
legend('h', 'Estimation de h')
```



Ici, le résultat est presque parfait : les deux courbes sont presque confondues.

Données aberrantes

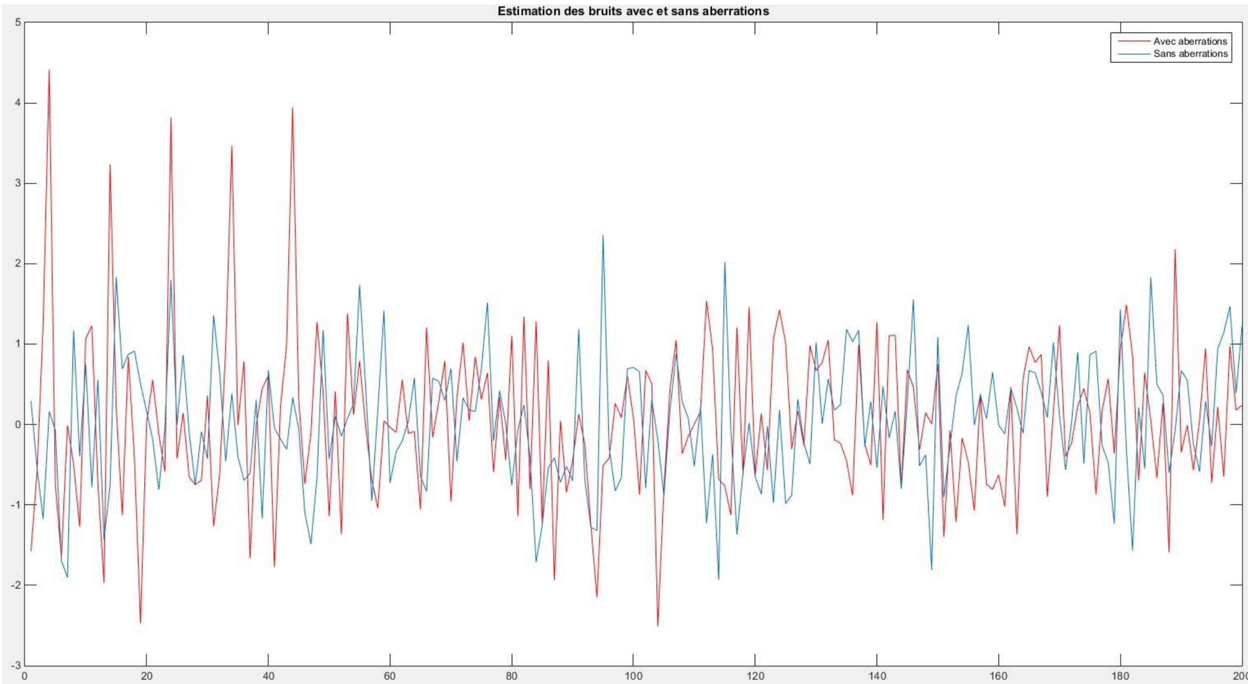
```
%11)
y_aberrant = syst(x,h);
y_aberrant(4) = 5;
y_aberrant(14) = 5;
y_aberrant(24) = 5;
y_aberrant(34) = 5;
y_aberrant(44) = 5;

estim_h_mle_aberrant = (inv(transpose(A)*A)) * transpose(A) * y_aberrant;

figure;
plot(h, 'r')
hold on;
plot(estim_h_mle_aberrant);
hold off;
title('Aberrantes');
legend('h', 'Estimation de h mle aberrant');

dqn = (1/M)*sum((abs(estim_h_mle_aberrant - h).^2));
%Pour info : dqn = 0.0331
```



On obtient un résultat très different de h.

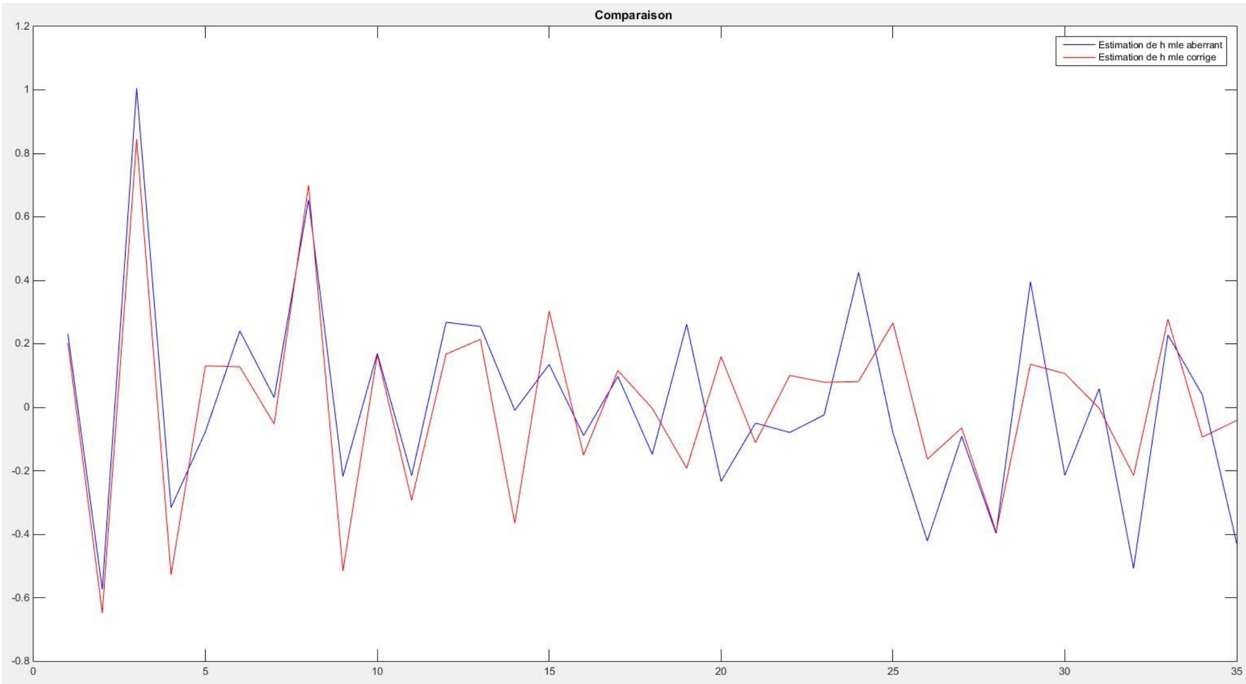
```
%12)
ynAberr = syst(x,h);
h_estime_mle_nAberr = (inv(A'*A))*A'*ynAberr;
%Valeurs normales
yEstimNormal = syst(x, h_estime_mle_nAberr);
%Valeurs aberrantes
yEstimAberr = syst(x, estim_h_mle_aberrant);

%Bruit
estim_bruit = ynAberr - yEstimNormal;
estim_bruit_aberrant = y_aberrant - yEstimAberr;

figure;
```

```
plot(estim_bruit_aberrant, 'r')
hold on;
plot(estim_bruit, 'b');
hold off;
title('Estimation des bruits avec et sans aberrations');
legend('Avec aberrations', 'Sans aberrations');
```



On peut se servir de la différence de bruit entre avec et sans aberrations pour tenter de corriger l'estimation en supprimant ces aberrations via le choix d'un seuil de différence.

```
%13)

%On prend un seuil de différence acceptable d'1.5

for i=1:200
    if (estim_bruit_aberrant(i) > 1.5)
        y_aberrant(i) = 0; %Supprime
    end
    if (estim_bruit_aberrant(i,1) < -1.5)
        y_aberrant(i) = 0; %Supprime
    end
end

estim_h_mle_fix = (inv(transpose(A)*A)) * transpose(A) * y_aberrant;


figure;
plot(estim_h_mle_aberrant, 'b');
hold on;
plot(estim_h_mle_fix, 'r');
hold off;
title('Comparaison ');
legend('Estimation de h mle aberrant', 'Estimation de h mle corrige');
```



On obtient donc une estimation beaucoup plus proche après avoir corrigé à l’aide du seuil d’aberration.