

Rapport du projet Automatique



Filtrage de Kalman sous Matlab

Sommaire

Introduction	2
---------------------------	----------

I Présentation du problème.....	3
--	----------

II Première partie	5
---------------------------------	----------

Les bruits de mesures affectant chaque série de mesure sont décorrélés.
Les bruits d'état et de mesure sont également décorrélés.

III Deuxième partie.....	9
---------------------------------	----------

Les bruits de mesures affectant chaque série de mesure sont corrélés.
Les bruits d'état et de mesure sont également décorrélés.
Technique de partage des mesures

IV Troisième partie	11
----------------------------------	-----------

Les bruits de mesures affectant chaque série de mesure sont corrélés.
Les bruits d'état et de mesure sont également corrélés.

Conclusion	13
-------------------------	-----------

Index des figures	14
--------------------------------	-----------

Annexes	15
----------------------	-----------

Annexe 1 : Programme première partie	16
Annexe 2 : Programme deuxième partie	20
Annexe 3 : Programme troisième partie	23

Introduction

Le cours d'automatique de DEA, avec Mr. Bigué, nous a permis de découvrir un nouveau filtrage : le filtrage de Kalman. Après un cours théorique, nous avons mis en pratique ces connaissances en développant sous Matlab un algorithme de Kalman, lors de deux séances de travaux pratiques.

Le filtrage de Kalman, inventé presque en même temps que l'observation d'état donne une solution optimale à une classe de problèmes bien posés et une solution satisfaisante assez souvent (mais pas toujours !). D'un certain point de vue, cette efficacité du filtrage de Kalman loin de ses hypothèses canoniques a fini par occulter son réel champ d'application.

La technique du filtre de Kalman est un outil privilégié de l'estimation des modèles structurels à composantes inobservables. Elle consiste en un ensemble d'équations récursives et procède en deux temps. Premièrement, une étape de filtrage qui permet d'obtenir la meilleure approximation de l'état Z_t du système à la date t , conditionnellement à l'information disponible jusqu'en t . Deuxièmement, une étape de lissage qui donne l'approximation optimale du vecteur d'état à l'instant t , conditionnellement à toute l'information disponible sur l'ensemble de la période, de 1 à T . Le lissage fournit des estimations des diverses composantes inobservables (tendance, cycle et irrégularité) de la série initiale, permettant ainsi de la désagréger date par date.

Pour ce projet d'automatique, le travail fait suite à celui fait en TP. En effet les lignes de codes restent dans l'ensemble les mêmes. Les modifications sont dues à la présence de deux bruits de mesures pour chaque partie au lieu d'un seul.

I Présentation du problème

Au cours de ce projet, on va reprendre un exemple abondamment traité en cours : un processus de Gauss Markov. Le processus considéré est stationnaire ; sa densité spectrale de puissance est donnée par :

$$\phi_S(p) = \frac{2}{-p^2 + 1}$$

Les observations du processus sont entachées de bruits dont les caractéristiques restent à déterminer. A la différence du TP déjà suivi, on dispose de 2 mesures bruitées du même signal à chaque instant. La méthodologie pour résoudre ces deux cas a été présentée en cours.

- ✚ Dans un premier cas, les bruits de mesures affectant chaque série de mesure sont décorrélés. Les bruits d'état et de mesure sont également décorrélés.
- ✚ Dans un second cas, les bruits de mesures affectant chaque série de mesure sont corrélés. Les bruits d'état et de mesure sont décorrélés.
- ✚ Dans un troisième cas, les bruits de mesures affectant chaque série de mesure sont corrélés. Les bruits d'état et de mesure sont corrélés.

Le modèle d'état continu pour ce système s'écrit :

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u$$

Avec $A = -1$
 $B = \sqrt{2}$.
 u est un bruit blanc gaussien.

On échantillonne toutes les 20 ms. On montre que le modèle d'état discret pour ce processus s'écrit :

$$\mathbf{x}_{k+1} = \Phi_{k+1,k} \cdot \mathbf{x}_k + \mathbf{v}_k$$

Avec :

$$\Phi_{k+1,k} = e^{A[kT].T_e} = 0,9802$$

Et

$$v_k = \int_{t_k}^{t_{k+1}} \Phi(t_{k+1}, \tau).B(\tau).u(\tau).d\tau$$

On montre que v_k est aussi une séquence blanche gaussienne, et que sa covariance Q_k s'écrit :

$$Q_k = E[v_k.v_k^T] = \int_{t_k}^{t_{k+1}} \int_{t_k}^{t_{k+1}} \Phi(t_{k+1}, \zeta).B(\zeta).E[u(\zeta).u^T(\eta)].B^T(\eta).\Phi^T(t_{k+1}, \eta).d\zeta.d\eta$$

Dans notre cas :

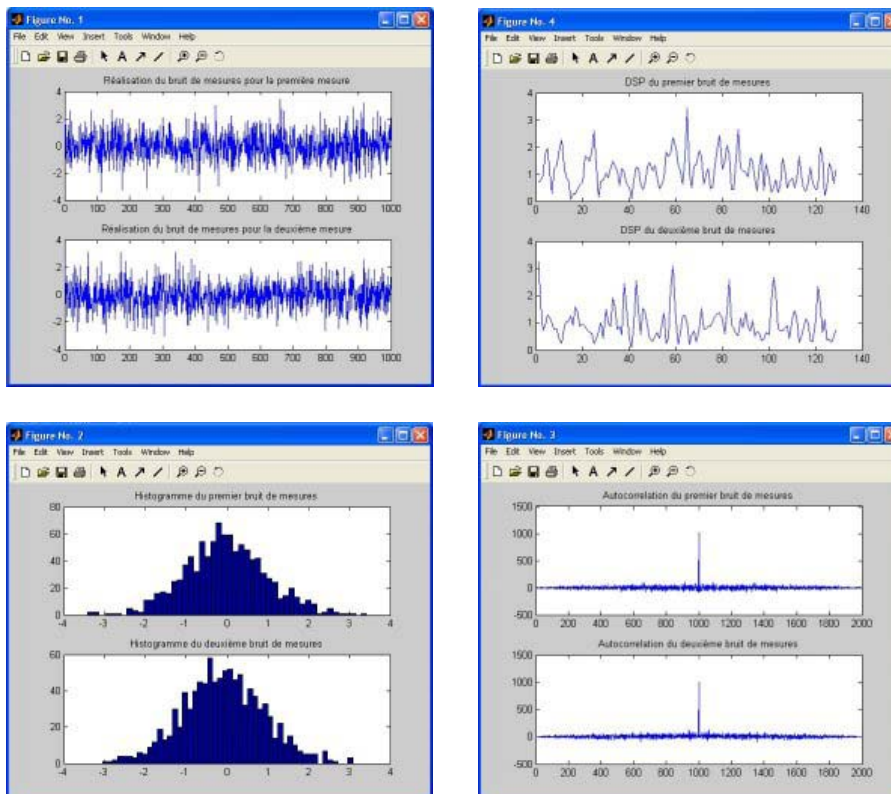
$$Q_k = E[v_k.v_k^T] = \int_0^{0.02} (\sqrt{2}.e^{-\eta})^2 d\eta \approx 0.03921$$

Nous allons maintenant passer à l'application de ce filtrage de Kalman, en examinant l'effet du filtrage pour les trois cas de mesures vu précédemment.

II Première partie

Le fichier de données fourni (TP2003.mat) contient 3 séries de variables (suffixées 1, 2 et 3). La première de ces séries est constituée de 3 variables : « z1 », « zbruite1 » et « bruit1 ». Nous allons l'étudier.

- ✚ **Première question :** « bruit1 » contient les valeurs relatives à une réalisation du bruit de mesures pour chaque mesure (une matrice 2 lignes x 1000 colonnes, 1 ligne par réalisation de bruit). Déterminez rapidement les caractéristiques de ces bruits.



La première figure montre l'évolution temporelle des deux bruits de mesures. Ensuite on peut voir la densité spectrale de ces bruits. L'analyse des histogrammes montre bien que ces deux signaux sont Gaussiens. Leur autocorrelation présentent bien un pic central et la valeur moyenne est presque nulle. Les conditions sont donc satisfaites pour appliquer la méthode de Kalman pour un filtrage.

De plus on obtient les valeurs caractéristiques suivantes :

- ✚ Moyenne du premier bruit : -0.0340
- ✚ Moyenne du deuxième bruit : -0.0892
- ✚ Covariance du premier bruit : 1.0160
- ✚ Covariance du deuxième bruit : 0.9940

- ✚ **Deuxième question :** Une fois les caractéristiques des bruits connues, utilisez l'algorithme de filtrage de Kalman pour débruiter les données « zbruite1 ». Commenter.

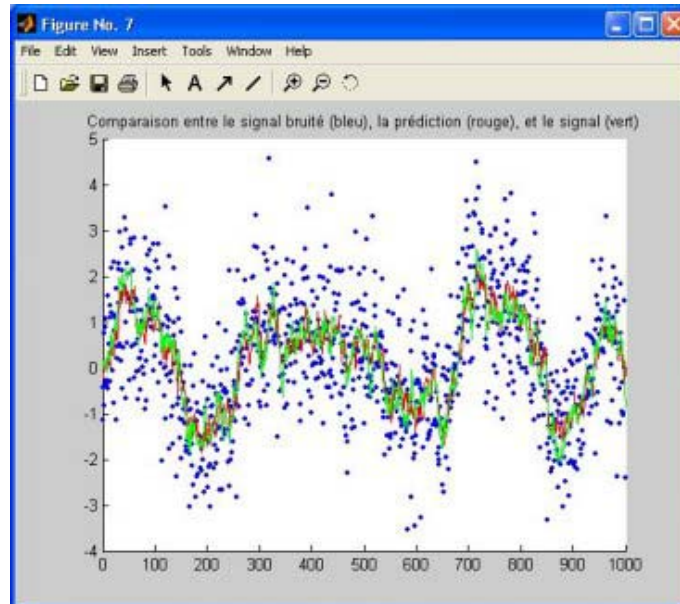
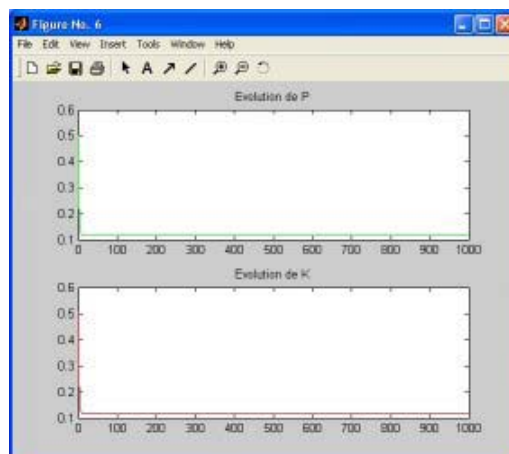


Figure 1 : Résultat du filtrage bruits de mesures et d'état décorrélés

Cette figure montre trois courbes : En bleu, on observe le signal bruité, en rouge la prédiction faite et en vert, le signal non bruité. Ce filtrage de Kalman permet de conclure que, à première vue, la prédiction semble très bonne par rapport au vrai signal non bruité. L'erreur commise est faible.

- ✚ **Troisième question :** Tracer l'évolution temporelle des variables P et K. Commentez.



Le premier graphe montre l'évolution de P et traduit la rapidité du débruitage. On observe alors que le filtrage est très performant au niveau de la vitesse. Le deuxième graphe montre l'évolution de K et traduit la stabilité du filtrage qui s'avère elle aussi très bonne.

- ✚ **Quatrième question :** A partir des données non bruitées (variable « $z1$ »), observez l'évolution temporelle de l'erreur avant et après filtrage. Comparez.

Voici les différentes erreurs relevées :

- Erreur quadratique pour le premier signal avant filtrage : 982.8
- Erreur quadratique pour le deuxième signal avant filtrage : 947.0
- Erreur quadratique après le filtrage : 107.9

On peut donc en conclure que l'erreur quadratique a donc considérablement chuté. Le filtrage de Kalman est donc clairement efficace pour ces mesures de bruits.

- ✚ **Cinquième question :** Comparer le filtrage ainsi réalisé (fusion de données à partir de deux mesures) et le filtrage classique, en n'utilisant par exemple que la première série de mesures.

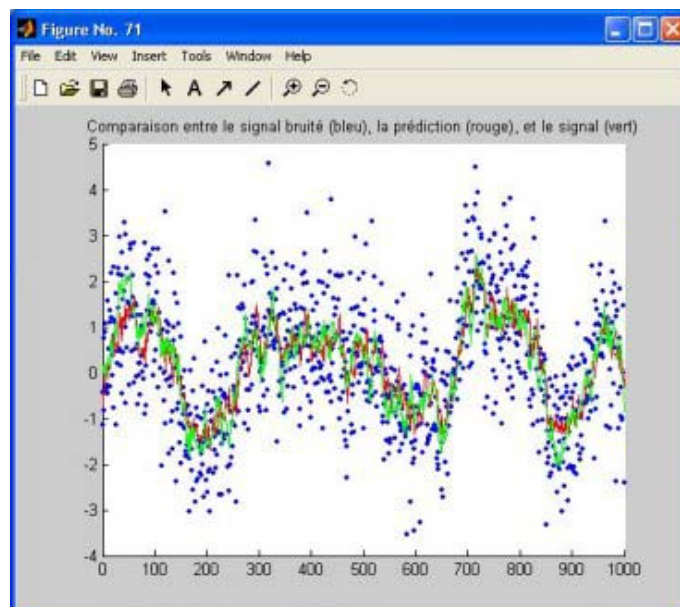


Figure 2 : Résultat du filtrage bruits de mesures et d'état décorrélés sans fusion de données

En observant de près les détails de ce graphe, on peut voir que le filtrage réalisé ici - sans fusion de données - est légèrement moins performant. En effet, la prédiction (en rouge) admet un petit retard par rapport au vrai signal. On observe aussi que la prédiction est trop forte à certains endroits.

Le relevé des erreurs est le suivant :

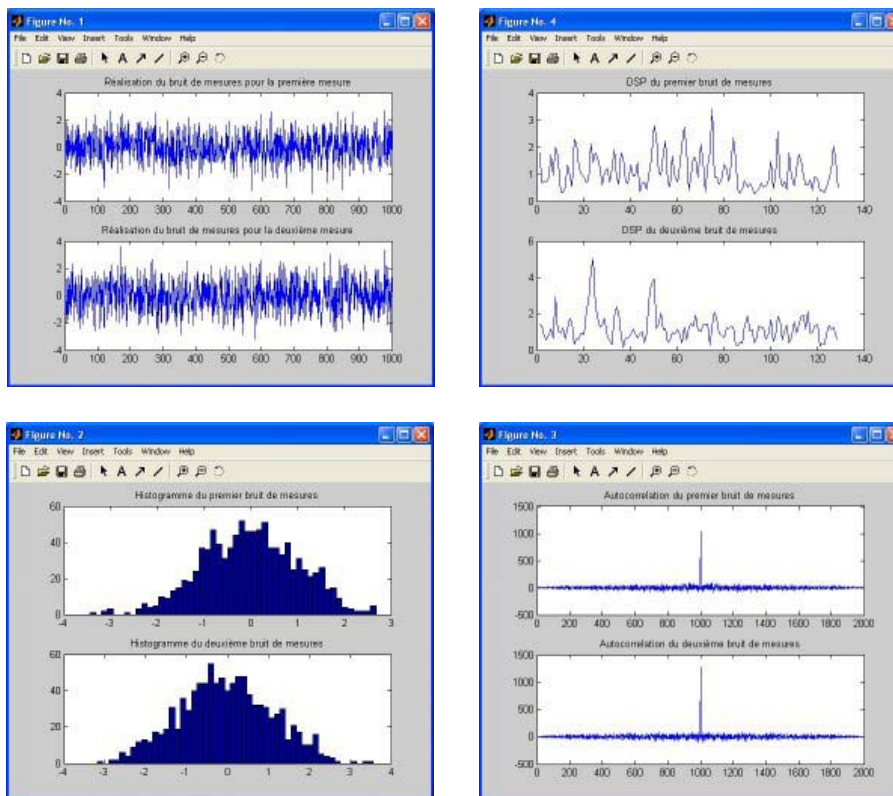
- Erreur quadratique pour le premier signal avant filtrage : 982.8
- Erreur quadratique après le filtrage : 148.0

L'erreur quadratique a donc aussi considérablement chuté, mais moins que lorsque l'on fusionne les données de mesures. Le filtrage est toujours efficace. Par contre la fusion des données apporte quelque chose de plus intéressant au niveau de la prédiction.

II Deuxième partie

Le fichier de données contient une deuxième série de variables, avec une deuxième série de réalisations de bruit d'observations.

- ✚ **Première question :** « bruit2 » contient les valeurs relatives à une réalisation du bruit de mesures pour chaque mesure (une matrice 2 lignes x 1000 colonnes, 1 ligne par réalisation de bruit). Déterminez rapidement les caractéristiques de ces bruits.



La première figure montre l'évolution temporelle des deux bruits de mesures. Ensuite on peut voir la densité spectrale de ses bruits.

L'analyse des histogrammes montre bien que ces deux signaux sont, pour cette deuxième série de mesures, Gaussiens. Leur autocorrelation présentent encore un pic central et la moyenne est quasi nulle. Les conditions sont donc satisfaites pour appliquer la méthode de Kalman pour un filtrage.

De plus on obtient les valeurs caractéristiques suivantes :

- ✚ Moyenne du premier bruit : -0.0123
- ✚ Moyenne du deuxième bruit : -0.0462
- ✚ Covariance du premier bruit : 1.0395
- ✚ Covariance du deuxième bruit : 1.2727



- ✚ **Deuxième question :** Une fois les caractéristiques des bruits connues, utilisez l'algorithme de filtrage de Kalman **avec la technique de partage des mesures** pour débruiter les données « zbruite2 ». Commentez.

Pour cette partie, les deux bruits de mesures sont corrélés. Il est donc nécessaire, avant d'appliquer le filtrage de Kalman, de les décorrélés. L'idée, comme on l'a vu à la fin du cours, est de diagonaliser la matrice R de covariance du bruit de mesure. La première étape est donc de calculer les termes de la matrice R, ensuite l'algorithme de Kalman est modifié légèrement, puis on lance les calculs afin de filtrer le signal.

Voici le résultat obtenu :

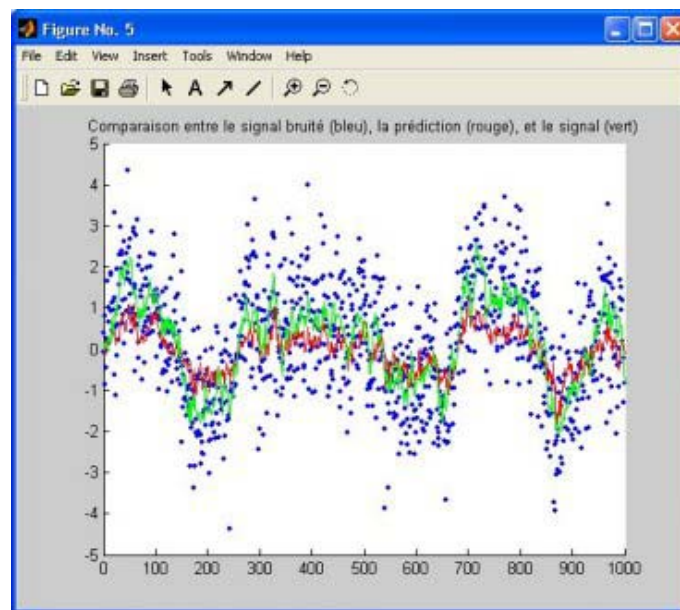


Figure 3 : Résultat du filtrage bruits de mesures corrélés et bruits d'état décorrélés. Partage des mesures.

Le résultat est assez satisfaisant, la prédiction est proche du signal non bruité. Mais pour conclure sur la qualité du filtrage, regardons les erreurs quadratiques avant et après filtrage.

- ✚ **Troisième question :** A partir des données non bruitées (variable « z2 »), observez l'évolution temporelle de l'erreur avant et après filtrage. Comparez.
- ✚ **Quatrième question :** Comparez avec le cas précédent

Voici les différentes erreurs relevées des erreurs :

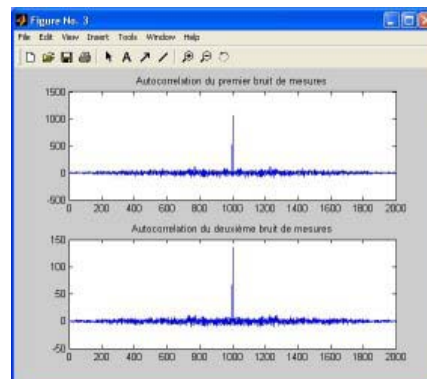
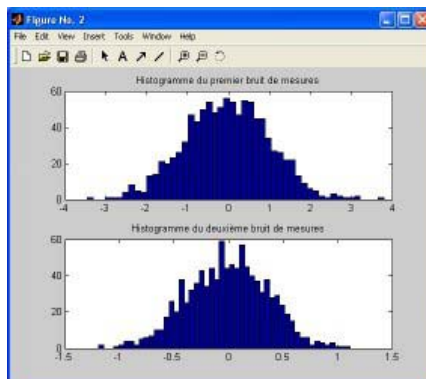
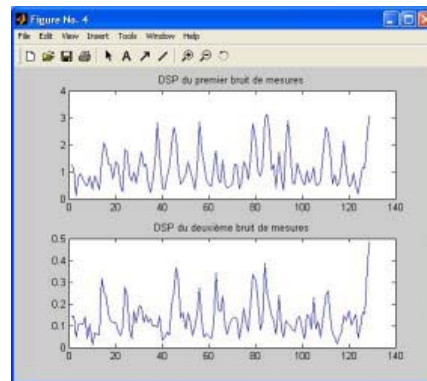
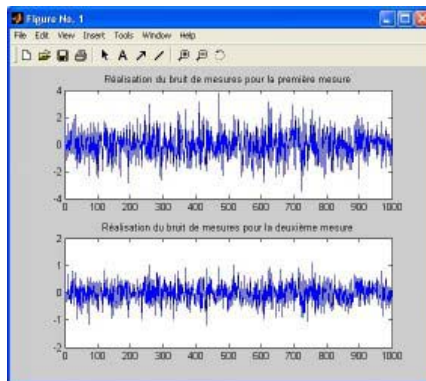
- Erreur quadratique pour le premier signal avant filtrage : 1015.7
- Erreur quadratique pour le deuxième signal avant filtrage : 1922
- Erreur quadratique après le filtrage : 404.1

L'erreur quadratique en fin de filtrage est bien plus grande que pour l'exemple précédent. Même si le filtrage semble bon, il est meilleur pour la première série de mesure (partie 1) dont l'erreur en fin de filtrage était 9 fois plus faible. On voit aussi que la prédiction ne suit pas aussi bien le signal que précédemment.

II Troisième partie

Le fichier de données contient une troisième série de variables, avec en particulier une troisième série de réalisations de bruit d'observations.

- ✚ **Première question :** « bruit3 » contient les valeurs relatives à une réalisation du bruit de mesures pour chaque mesure (une matrice 2 lignes x 1000 colonnes, 1 ligne par réalisation de bruit). La covariance entre U et le premier bruit de mesure est « covU1 », entre U et le deuxième bruit de mesure « covU2 ». Déterminez rapidement les caractéristiques des bruits de mesure



La première figure montre l'évolution temporelle des deux bruits de mesures. Ensuite on peut voir la densité spectrale de ses bruits.

L'analyse des histogrammes montre bien que ces deux signaux sont, pour cette troisième série de mesures, Gaussiens, leur autocorrélation présentent encore un pic central et la moyenne est presque nulle. Les conditions sont donc satisfaites pour appliquer la méthode de Kalman pour un filtrage.

On obtient les valeurs caractéristiques suivantes :

- ✚ Moyenne du premier bruit : -0.0366
- ✚ Moyenne du deuxième bruit : -0.0089
- ✚ Covariance du premier bruit : 1.0612
- ✚ Covariance du deuxième bruit : 0.1362

- ✚ **Deuxième question :** Une fois les caractéristiques des bruits connues, utilisez l'algorithme de filtrage de Kalman pour débruiter les données « zbruite3 ». Commentez.

Dans cet exemple, les bruits de mesures affectant chaque série de mesure sont corrélés et les bruits d'état et de mesure sont aussi corrélés. Le premier algorithme sera modifier au niveau de la matrice R qui sera alors diagonale avec les valeurs covU1 et covU3 dans sa diagonale.

On obtient le résultat suivant :

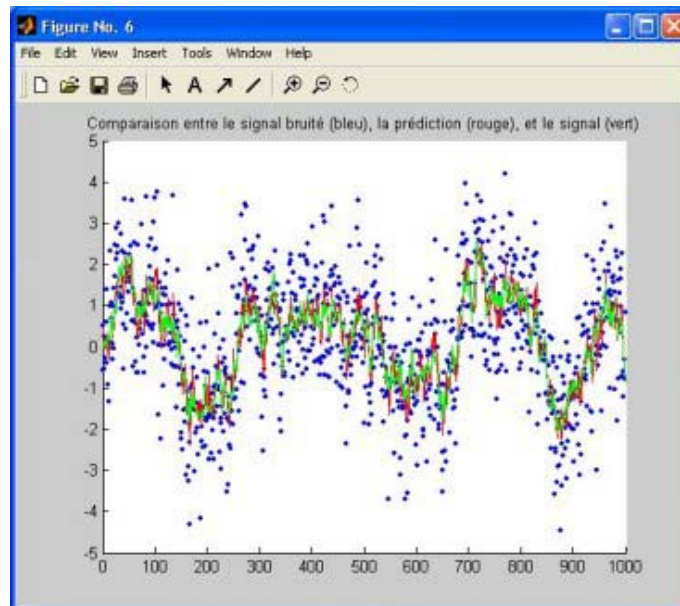


Figure 4 : Résultat du filtrage bruits de mesures et d'état corrélés

Le résultat est encore très satisfaisant, la prédiction est proche du signal non bruité. Regardons les différentes erreurs.

- ✚ **Troisième question :** A partir des données non bruitées (variable « z3 »), observez l'évolution temporelle de l'erreur avant et après filtrage. Comparez.

Voici les différentes erreurs relevées des erreurs :

- Erreur quadratique pour le premier signal avant filtrage : 1152.6
- Erreur quadratique pour le deuxième signal avant filtrage : 146.51
- Erreur quadratique après le filtrage : 93.13

L'erreur quadratique a chuté, cependant un des bruits avait déjà une erreur quadratique faible. Ce filtrage a l'air d'être le plus efficace. Cependant, cette grande baisse est peut-être due à la faible valeur de l'erreur quadratique du deuxième signal avant filtrage. On ne peut donc pas vraiment comparer ce filtrage avec les deux précédents dont les erreurs quadratiques des bruits de mesures étaient relativement proches

Conclusion

Ces séances de TP et ce dernier TP test, transformé en projet, nous ont permis de découvrir une nouvelle technique de filtrage qu'est celle de Kalman. De plus la programmation se faisait sous l'environnement de calcul Matlab ; cela nous a donc permis de nous améliorer au niveau de l'écriture du langage.

Ces domaines d'application sont aujourd'hui très vastes et les résultats obtenus, comme dans notre exemple de débruitage de signaux, montrent leurs grandes possibilités.

Les trois programmes constituant chaque partie de ce projet sont disponibles sur la disquette jointe. Il s'y trouve également le rapport en PDF.

Table des figures

Figure 1 : Résultat du filtrage bruits de mesures et d'état décorrelés	5
Figure 2 : Résultat du filtrage bruits de mesures et d'état décorrelés sans fusion de données	6
Figure 3 : Résultat du filtrage bruits de mesures corrélés et bruits d'état décorrelés. Partage des mesures	9
Figure 4 : Résultat du filtrage bruits de mesures et d'état corrélés	11

Annexes

Annexe 1 : Première partie du programme

```

%%%%%%%%%%
% TP EXAM de Filtrage de Kalman %
% SOUBIGOU Antoine %
%%%%%%%%%%

clc; clear; close all;

%%%%%%%%%%
% Première partie %
%%%%%%%%%%

%%%%%%%%%%
% Mise en place des variables utilisées %
%%%%%%%%%%

load TP2003.mat
b1=bruit1(:,1);
b11=bruit1(:,2);
zb1=zbruit1(1,:);
zb11=zbruit1(2,:);

%%%%%%%%%%
% Tracé et calcul des différentes caractéristiques des bruits %
%%%%%%%%%%

figure(1)
subplot(211);
plot(b1);
title('Réalisation du bruit de mesures pour la première mesure');
subplot(212);
plot(b11);
title('Réalisation du bruit de mesures pour la deuxième mesure');
pause;

% Moyenne des deux bruits de mesures
disp('Moyenne des bruits de mesures')
MoyennePremierBruit = mean(b1) % -0.0340
MoyenneDeuxiemeBruit = mean(b11) % -0.0892
pause;

% Sont-ils gaussien ? Oui, les graphes le montrent
figure(2)
subplot(211);
hist(b1,50)
title('Histogramme du premier bruit de mesures');
subplot(212);
hist(b11,50)
title('Histogramme du deuxième bruit de mesures');
pause;

```

```
%autocorrelation
figure(3)
subplot(211);
plot(xcorr(b1))
title('Autocorrelation du premier bruit de mesures');
subplot(212);
plot(xcorr(b11))
title('Autocorrelation du deuxième bruit de mesures');
pause;

%Covariance des bruits de mesures
disp('Covariance des bruits de mesures')
CovariancePremierBruit = cov(b1)      % 1.0160
CovarianceDeuxiemeBruit = cov(b11)   % 0.9940
pause;

%DSP des bruits de mesures
figure(4)
subplot(211);
plot(psd(b1))
title('DSP du premier bruit de mesures');
subplot(212);
plot(psd(b11))
title('DSP du deuxième bruit de mesures');
pause;

% Variance des bruits de mesures identique à la covariance
disp('Variance des bruits de mesures')
VariancePremierBruit = var(b1)
VarianceDeuxiemeBruit = var(b11)
pause;

% représentation fréquentielle du bruit:
figure(5)
subplot(211);
plot(abs(fft(b1)))
title('Représentation fréquentielle du premier bruit');
subplot(212);
plot(abs(fft(b11)))
title('Représentation fréquentielle du deuxième bruit de mesures');
pause;

%%%%%%%%%%%%%%
% Implémentation de kalman %
%%%%%%%%%%%%%%

% Débruitage des données 'zbruite1' dans un premier temps.
% Elles sont décorrélés

% Mise en place des variables connues

R=[VariancePremierBruit 0;0 VarianceDeuxiemeBruit]; % Variance des bruits de mesure
Q=0.03921; % Covariance du bruit d'état
Te=0.02; % Echantillonnage de 20 ms
F=exp(-Te); % F=phi
Pinit=1000; % On le met grand pour le début
xinit=0;
```

```

H=[1;1];
eq11=0;
eq12=0;
eq2=0;

for i=1:1000
    K(i,:)=Pinit*H'*(H*Pinit*H'+R)^(-1);           % équation de prédiction
    x(i)=xinit+K(i,:)*(zbruite1(:,i)-H*xinit);      % équation de correction avec la mesure
    P(i)=Pinit-K(i,:)*H*Pinit;                    % correction de la variance de l'erreur de prédiction
    xinit=F*x(i);                                  % prédiction
    Pinit=F*P(i)*F'+Q;                             % prédiction

    % Calcul de l'erreur quadratique :
    eq11=eq11+(z1(i)-zbruite1(1,i))^2;            % calcul de l'erreur premier bruit avant filtrage
    eq12=eq12+(z1(i)-zbruite1(2,i))^2;            % calcul de l'erreur deuxième bruit avant filtrage
    eq2=eq2+(z1(i)-x(i))^2;                       % calcul de l'erreur quadratique après filtrage

end

% Résultat de l'algorithme avec la fusion de données

figure(6)
hold on
plot(zbruite1(1,:), 'b.')
plot(x, 'r')
plot(z1, 'g')
title('Comparaison entre le signal bruité (bleu), la prédiction (rouge), et le signal (vert)');
pause;
hold off

% Evolution temporelle des variables P et K :

figure(7)
subplot(211);
plot(P, 'g')
title('Evolution de P') % Ca traduit la rapidité du débruitage
subplot(212);
plot(K(:,1))
plot(K(:,2), 'r')
title('Evolution de K') % et la stabilité du système
pause;

% Calcul de l'erreur quadratique avant et après filtrage:

eq11 % 982.88
eq12 % 947.08
eq2 % 107.93
pause;

% L'erreur quadratique a donc considérablement chutée.
% Le filtrage est donc clairement efficace.

% Comparaison du filtrage ainsi réalisé avec un filtrage classique:
% Implémentation de Kalman (on n'utilise que la première série de mesures).

```

% Valeurs connues:

```
R=CovariancePremierBruit;    % Variance du premier bruit de mesure
H=1;
K=0;
Q=0.03921;    % Covariance du bruit d'état
Te=0.02;    % Echantillonnage de 20 ms
F=exp(-Te);    % F=phi
Pinit=1000;
xinit=0;
eq1=0;
eq2=0;

for i=1:1000
    K(i)=Pinit*(Pinit+R)^(-1);    % équation de prédiction
    x(i)=xinit+K(i)*(zbruite1(1,i)-xinit);    % équation de correction avec la mesure
    P(i)=Pinit-K(i)*Pinit;    % correction de la variance de l'erreur de prédiction
    xinit=F*x(i);    % prédiction
    Pinit=F*P(i)*F'+Q;    % prédiction

    % erreur quadratique
    eq1=eq1+(z1(i)-zbruite1(1,i))^2 ;    % calcul de l'EQ avant filtrage
    eq2=eq2+(z1(i)-x(i))^2;    % calcul de l'EQ après filtrage
end
```

% Résultat du filtrage

```
figure(8)
hold on
plot(zbruite1(1,:), 'b.')
plot(x, 'r')
plot(z1, 'g')
title('Comparaison entre le signal bruité (bleu), la prédiction (rouge), et le signal (vert)');
pause;
hold off
```

% Evolution temporelle des variables P et K :

```
figure(9)
subplot(211);
plot(P, 'g')
title('Evolution de P')    % ça traduit la rapidité du débruitage
subplot(212);
plot(K, 'r')
title('Evolution de K')    % et la stabilité du système
pause;
```

% Calcul de l'erreur quadratique avant et après filtrage:

```
eq1    % 982.88
eq2    % 148.03
pause;
```

```
% L'erreur quadratique a donc considérablement chutée, mais moins que lorsque
% l'on fusionne les données de mesure.
% Le filtrage est toujours efficace.
% Dans ce cas, la fusion des données apporte quelque chose de plus intéressant.
% Appel à la deuxième partie du sujet
Partie2
```

Annexe 2 : Deuxième partie du programme

```

%%%%%%%%%%
%
% Deuxième partie %
%
%%%%%%%%%%

clc; clear; close all;

%%%%%%%%%%
% Mise en place des variables utilisées %
%%%%%%%%%%

load TP2003.mat
b2=bruit2(:,1);
b22=bruit2(:,2);
zb2=zbruite2(1,:);
zb22=zbruite2(2,:);

%%%%%%%%%%
% Tracé et calcul des différentes caractéristiques des bruits %
%%%%%%%%%%

figure(1)
subplot(211);
plot(b2);
title('Réalisation du bruit de mesures pour la première mesure');
subplot(212);
plot(b22);
title('Réalisation du bruit de mesures pour la deuxième mesure');
pause;

% Moyenne des deux bruits de mesures
disp('Moyenne des bruits de mesures')
MoyennePremierBruit = mean(b2)           % -0.0123
MoyenneDeuxiemeBruit = mean(b22)        % -0.0462
pause;

% Sont-ils gaussien ?   Oui, les graphes le montrent
figure(2)
subplot(211);
hist(b2,50)
title('Histogramme du premier bruit de mesures');
subplot(212);
hist(b22,50)
title('Histogramme du deuxième bruit de mesures');
pause;

% Autocorrelation
figure(3)
subplot(211);
plot(xcorr(b2))
title('Autocorrelation du premier bruit de mesures');
subplot(212);
plot(xcorr(b22))
title('Autocorrelation du deuxième bruit de mesures');
pause;

% Covariance des bruits de mesures

```

```

disp('Covariance des bruits de mesures')
CovariancePremierBruit = cov(b2)      % 1.0395
CovarianceDeuxiemeBruit = cov(b22)   % 1.2727
pause;

% DSP des bruits de mesures
figure(4)
subplot(211);
plot(psd(b2))
title('DSP du premier bruit de mesures');
subplot(212);
plot(psd(b22))
title('DSP du deuxième bruit de mesures');
pause;

% Variance des bruits de mesures
disp('Variance des bruits de mesures')
VariancePremierBruit = var(b2)
VarianceDeuxiemeBruit = var(b22)
covariance12 = cov(bruit2(:,1),bruit2(:,2))
pause;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Implémentation de kalman avec partage des mesures %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Voici les deux bruits de mesures qui sont corrélés
%
% b2=bruit2(:,1) et b22=bruit2(:,2);
%
% On va diagonaliser la matrice R de covariance du bruit
%
%      (
%      ( E(b2²)   E(b2.b22) )
%  R = (
%      ( E(b22.b2) E(b22²) )
%      (
%
% Calcul des différents termes de la matrice puis mise à jour de zbruite2

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Création des nouveaux bruits de mesures %
% Pour le partage des mesures (vu en cours) %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% x1 et x2 vont remplacer les deux colonnes de zbruite2

for i=1:1000
    x1(i)=0;
    x2(i)=0;
end

% On calcul les nouvelles valeurs décorréllés pour les bruits de mesures
% On note alors x1 la nouvelle valeur de b2
% Et x2 la nouvelle valeur de b22
% La relation est tirée de l'exemple d'application vu en cours

for i=1:1000
    x1(i) = zbruite2(1,i);

```

```
x2(i) = -( (zbruite2(1,i) + zbruite2(2,i))/2 ) * zbruite2(1,i) + ( (zbruite2(1,i) + zbruite2(2,i))/2 ) * zbruite2(2,i);
end
```

```
% Puis on crée finalement w, semblable à zbruite2 mais dont les mesures sont partagées
```

```
for i=1:1000
    w(1,i)=x1(i);
    w(2,i)=x2(i);
end
```

```
% Valeurs connues
```

```
R=[VariancePremierBruit 0; % Variance des bruits de mesure (combinaison linéaire des anciennes mesures)
    0 -(covariance12(1,2))^2 * VariancePremierBruit + (VariancePremierBruit)^2 * VarianceDeuxiemeBruit];
Q=0.03921; % Covariance du bruit d'état
Te=0.02; % Echantillonnage de 20 ms
F=exp(-Te); % F=phi
Pinit=1000;
xinit=0;
H=[1;1];
eq11=0;
eq12=0;
eq2=0;
```

```
for i=1:1000
    K(i,:)=Pinit*H'*(H*Pinit*H'+R)^(-1); % équation de prédiction
    x(i)=xinit+K(i,:)*(w(:,i)-H*xinit); % équation de correction avec la mesure
    P(i)=Pinit-K(i,:)*H*Pinit; % correction de la variance de l'erreur de prédiction
    xinit=F*x(i); % prédiction
    Pinit=F*P(i)*F'+Q; % prédiction
    % Calcul de l'erreur quadratique :
    eq11=eq11+(z2(i)-w(1,i))^2; % calcul de l'EQ avant filtrage
    eq12=eq12+(z2(i)-w(2,i))^2; % calcul de l'EQ avant filtrage
    eq2=eq2+(z2(i)-x(i))^2; % calcul de l'EQ après filtrage
end
```

```
end

% Evolution de l'erreur avant et après filtrage :
% On compare le signal bruité avec notre correction.
```

```
figure(5)
hold on
plot(w(1,:), 'b.')
plot(x, 'r')
plot(z2, 'g')
title('Comparaison entre le signal bruité (bleu), la prédiction (rouge), et le signal (vert)');
pause;
hold off
```

```
% Calcul de l'erreur quadratique avant et après filtrage:
```

```
eq11 % 1015.7
eq12 % 1269.7
eq2 % 432.83
pause;
```

```
% L'erreur quadratique a donc chutée.
% Le filtrage est efficace, cependant ce filtre reste peu précis.
% Appel à la troisième partie du sujet
Partie3
```


Annexe 3 : Troisième partie du programme

```

%%%%%%%%%%
%
% Troisième partie %
%
%%%%%%%%%%

```

```
clc; clear; close all;
```

```

%%%%%%%%%%
% Mise en place des variables utilisées %
%%%%%%%%%%

```

```

load TP2003.mat
b3=bruit3(:,1);
b33=bruit3(:,2);
zb3=zbruite3(1,:);
zb33=zbruite3(2,:);

```

```

%%%%%%%%%%
% Tracé et calcul des différentes caractéristiques des bruits %
%%%%%%%%%%

```

```

figure(1)
subplot(211);
plot(b3);
title('Réalisation du bruit de mesures pour la première mesure');
subplot(212);
plot(b33);
title('Réalisation du bruit de mesures pour la deuxième mesure');
pause;

```

```

% Moyenne des deux bruits de mesures
disp('Moyenne des bruits de mesures')
MoyennePremierBruit = mean(b3)           % -0.0366
MoyenneDeuxiemeBruit = mean(b33)        % -0.0089
pause;

```

```

% Sont-ils gaussien ?   Oui, les graphes le montrent
figure(2)
subplot(211);
hist(b3,50)
title('Histogramme du premier bruit de mesures');
subplot(212);
hist(b33,50)
title('Histogramme du deuxième bruit de mesures');
pause;

```

```

% Autocorrelation
figure(3)
subplot(211);
plot(xcorr(b3))
title('Autocorrelation du premier bruit de mesures');
subplot(212);
plot(xcorr(b33))
title('Autocorrelation du deuxième bruit de mesures');
pause;

```

```

% Covariance des bruits de mesures
disp('Covariance des bruits de mesures')
CovariancePremierBruit = cov(b3)           % 1.0612
CovarianceDeuxiemeBruit = cov(b33)        % 0.1362
pause;

% DSP des bruits de mesures
figure(4)
subplot(211);
plot(psd(b3))
title('DSP du premier bruit de mesures');
subplot(212);
plot(psd(b33))
title('DSP du deuxième bruit de mesures');
pause;

% Variance des bruits de mesures
disp('Variance des bruits de mesures')
VariancePremierBruit = var(b3)
VarianceDeuxiemeBruit = var(b33)
pause;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Implémentation de kalman %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Valeurs connues

R=[covU1 0;           % Variance des bruits de mesure
  0 covU2];
Q=0.03921;           % Covariance du bruit d'état
Te=0.02;             % Echantillonnage de 20 ms
F=exp(-Te);          % F=phi
Pinit=1000;
xinit=0;
H=[1;1];
eq11=0;
eq12=0;
eq2=0;

for i=1:1000
    K(i,:)=Pinit*H*(H*Pinit*H'+R)^(-1);           % équation de prédiction
    x(i)=xinit+K(i,:)*(zbruite3(:,i)-H*xinit);     % équation de correction avec la mesure
    P(i)=Pinit-K(i,:)*H*Pinit;                     % correction de la variance de l'erreur de prédiction
    xinit=F*x(i);                                   % prédiction
    Pinit=F*P(i)*F'+Q;                               % prédiction

    % Calcul de l'erreur quadratique :
    eq11=eq11+(z3(i)-zbruite3(1,i))^2;              % calcul de l'EQ avant filtrage
    eq12=eq12+(z3(i)-zbruite3(2,i))^2;              % calcul de l'EQ avant filtrage
    eq2=eq2+(z3(i)-x(i))^2;                          % calcul de l'EQ après filtrage
end

% Evolution de l'erreur avant et après filtrage

```

% On compare le signal bruité avec notre correction.

```
figure(5)
hold on
plot(zbruite3(1,:), 'b.')
plot(x, 'r')
plot(z3, 'g')
title('Comparaison entre le signal bruité (bleu), la prédiction (rouge), et le signal (vert)');
pause;
hold off
```

% Calcul de l'erreur quadratique avant et après filtrage

```
eq11 % 1152.6
eq12 % 146.51
eq2 % 93.13
pause;
```

% L'erreur quadratique a chuté, cependant un des bruits avait déjà une erreur quadratique faible.

```
close all;
```