Filtrage de données accélérométriques et GPS par le filtre de Kalman

Projet 08A08 : Etude et mise en œuvre d'accéléromètres pour des systèmes embarqués de signalisation ferroviaire

Sommaire

Pages

Table des figures

I. Vue d'ensemble

Un accéléromètre est un capteur qui, fixé à un mobile, permet de mesurer l'accélération de ce dernier. Généralement on utilise un GPS pour connaître les positions et vitesses d'un véhicule mais ce qui serait de pouvoir déterminer ces positions et vitesses en partant des données accéléromètriques.
Le filtre de Kalman va nous permettre de reconstruire l'état de notre système en recalant les données des accéléromètres grâce aux mesures GPS.

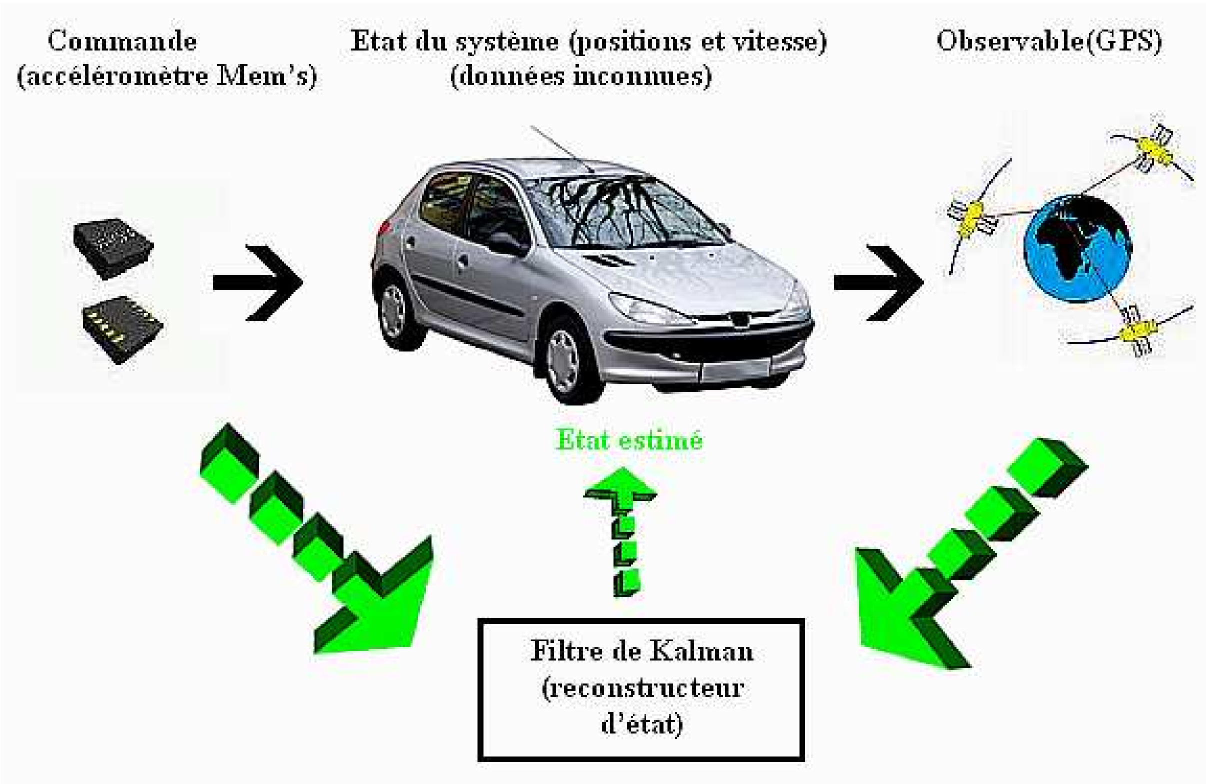

Fig1 : Vue d'ensemble

II. Système

On travaillera avec des valeurs échantillonnées. Notre système est donc défini comme suit :

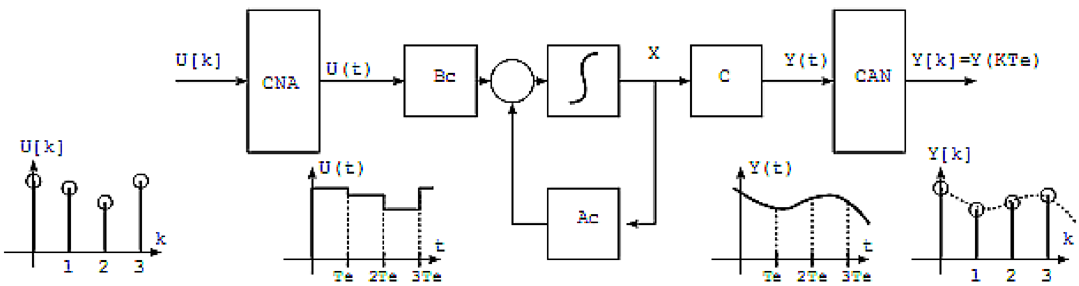

Fig2: Représentation du système

Avec U[k] et Y[k] respectivement la commande et l'observable de notre système. Le vecteur U contient les données du capteur (accéléromètre 3 axes) et Y est relatif aux données GPS : ces deux vecteurs seront donc de dimension 3x1.
L'état inconnu du système est représenté par le vecteur X qui contient les positions ainsi que les vitesses sur les axes x, y et z. X est donc de taille 6x1.
Unités de mesure
Les composantes de X et Y sont en m et ceux de U en m/s^2

III. Définitions des matrices et vecteur intervenant dans le filtre

1. Modèle d'évolution

L'évolution du système est de la forme : $\mathbf{X_k=A*X_{k-1}+B*U_{k-1}+M*W_{k-1}}$ avec:

- A_{6x6}= exp(Ac*Te), relie l'état courant à l'état précédent.
- B_{6x3}= exp(Bc*Te), relie l'entrée de a commande U à l'état X.
- M_{6x3} le bruit sur l'état X à l'état X.

W_{3x1} est le vecteur contenant les bruits du capteur sur les axes x, y et z ou encore l'erreur sur la commande U.
Ce modèle peut se réécrire sous la forme suivante :

$$\overbrace{\begin{pmatrix}x\\y\\z\\vx\\vy\\vz\end{pmatrix}_k}^{X_k} = \overbrace{\begin{pmatrix}1&0&0&Te&0&0\\0&1&0&0&Te&0\\0&0&1&0&0&Te\\0&0&0&1&0&0\\0&0&0&0&1&0\\0&0&0&0&0&1\end{pmatrix}}^{A}\overbrace{\begin{pmatrix}x\\y\\z\\vx\\vy\\vz\end{pmatrix}_{k-1}}^{X_{k-1}} + \overbrace{\begin{pmatrix}Te^2/2&0&0\\0&Te^2/2&0\\0&0&Te^2/2\\Te&0&0\\0&Te&0\\0&0&Te\end{pmatrix}}^{B}\overbrace{\begin{pmatrix}ax\\ay\\az\end{pmatrix}_{k-1}}^{U_{k-1}} + \overbrace{\begin{pmatrix}Te^2/2&0&0\\0&Te^2/2&0\\0&0&Te^2/2\\Te&0&0\\0&Te&0\\0&0&Te\end{pmatrix}}^{M}\overbrace{\begin{pmatrix}wx\\wy\\wz\end{pmatrix}_{k-1}}^{W_{k-1}}$$

NB: Les indices k et k-1 représentent respectivement l'état suivant et l'état précédent

2. Modèle d'observation

Ce modèle relie les données du GPS (l'observable) à l'état. Le GPS fournit les cordonnées cartésiennes sur les axes x, y et z du mobile. Donc ce modèle nous fournira les positions (sur les axes x, y et z) de l'état ainsi que le bruit de mesure sur le GPS ($\mathbf{v_{k-1}}$).

On a l'équation suivante $\mathbf{Y_{k-1}=C*X_{k-1}+v_{k-1}}$

$$\overbrace{\begin{pmatrix}y_x\\y_y\\y_z\end{pmatrix}_{k-1}}^{Y_{k-1}} = \overbrace{\begin{pmatrix}1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&1&0&0&0\end{pmatrix}}^{C}\overbrace{\begin{pmatrix}x\\y\\z\\vx\\vy\\vz\end{pmatrix}_{k-1}}^{X_{k-1}} + \overbrace{\begin{pmatrix}v_x\\v_y\\v_z\end{pmatrix}_{k-1}}^{v_{k-1}}$$

3. La matrice de covariance de l'état

Les erreurs sur l'état sont indépendantes aussi on peut commencer à prendre des mesures à partir de n'importe quel point. Donc à l'instant zéro la matrice s'écrira sur cette forme :

$$P_0 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

4. Matrice de covariance de mesure sur le GPS

On considère que les erreurs de mesure sur le GPS sont indépendantes les unes des autres. On aura donc :

$$R = \begin{pmatrix} \sigma^2_{xgps} & 0 & 0 \\ 0 & \sigma^2_{ygps} & 0 \\ 0 & 0 & \sigma^2_{zgps} \end{pmatrix}$$

5. Matrice de covariance du système

On considère aussi que les bruits de mesure sur chaque axe du capteur sont indépendants les uns des autres :

$$Q = \begin{pmatrix} \sigma^2_{ax} & 0 & 0 \\ 0 & \sigma^2_{ay} & 0 \\ 0 & 0 & \sigma^2_{az} \end{pmatrix}$$

IV. Filtrage

Le filtre de Kalman a deux phases distinctes : **Prédiction** et **Mise à jour**. La phase de prédiction utilise l'état estimé de l'instant précédent pour produire une estimation de l'état courant. Dans l'étape de mise à jour, les observations de l'instant courant sont utilisées pour corriger l'état prédit dans le but d'obtenir une estimation plus précise.

1. Prédiction

Dans cette phase nous partirons de la valeur de l'état initial (pour nous la valeur initiale est toujours nulle :

$$X_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Connaissant donc X_0 nous pourrons prédire l'état suivant grâce à cette équation :
$\mathbf{X_{k/k-1}=A*X_{k-1/k-1}+B*U_{k-1}}$
L'état suivant sera ainsi prédit avec une certaine erreur qu'on pourra prédire à son tour:
$\mathbf{P_{k/k-1}=A*P_{k-1/k-1} + M*Q_{k-1}*M^T}$
On se rappellera aussi que :

$$P_0 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

2. <u>L'innovation et le gain</u>

L'innovation (vecteur 3x1) portera sur la différence entre la mesure et la prédiction :
$\mathbf{In=Ygps_{k-1} - C*X_{k/k-1}}$
Ygps est le vecteur contenant les coordonnées GPS sur les axes x, y et z. C'est en fait l'observable du système (se reporter à Fig1).
La matrice In portera donc sur la différence entre ce qui est réellement observé du système (GPS) et l'observation des valeurs prédites ($\mathbf{C*X_{k/k-1}}$).
Cette innovation sera à son tour pondérée par le gain de Kalman K_k qui représente la quantité d'information supplémentaire à apporter à la correction lors d'une nouvelle mesure. Le gain de Kalman toujours compris entre 0 et 1, joue un rôle très important dans l'étape de correction, il permet de connaître la confiance que l'on peut apporter à la mesure par rapport à la prédiction.
Le gain s'exprimera comme suit :
$\mathbf{K_k=P_{k/k-1}*C^T*(C*P_{k/k-1}*C^T + R)^{-1}}$

3. <u>La mise à jour</u>

Après la correction entre la mesure donnée par le GPS et la prédiction, on met à jour l'état suivant ($\mathbf{X_{k/k-1}}$ qui était prédit) par la relation :
$\mathbf{X_{k/k}=X_{k/k-1} + K_k*In}$
En même temps on mettra à jour l'erreur suivante ($\mathbf{P_{k/k-1}}$ qui était prédite) avec la relation :
$\mathbf{P_{k/k}= (I - K_k*C)*P_{k/k-1}}$
Avec I la matrice identité de taille 6x6.

V. Codage sur Matlab

```
%Définition des matrices constantes intervenant dans le filtre de Kalman
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Te=1;
A=[1   0   0   Te   0   0;
   0   1   0    0   Te  0;
   0   0   1    0   0  Te;
   0   0   0    1   0   0;
   0   0   0    0   1   0;
   0   0   0    0   0   1;];

B=[(Te^2)/2      0        0;
      0      (Te^2)/2     0;
      0          0    (Te^2)/2;
      Te         0        0;
      0          Te       0;
      0          0       Te;];
 M=B;

 C= [1  0   0   0   0   0;
     0  1   0   0   0   0;
     0  0   1   0   0   0];


%variance sur xgps,ygps et zgps (erreur de mesure de +/- 2m sur chaque axe pour le GPS)
sigmaxgps=2;
sigmaygps=2;
sigmazgps=2;

% Matrice de covariance de mesure du GPS
R=[(sigmaxgps)^2          0                  0;
         0           (sigmaygps)^2            0;
         0                0             (sigmazgps)^2];

%Matrice de covariance P
P=[0 0 0 0 0 0;
   0 0 0 0 0 0;
   0 0 0 0 0 0;
   0 0 0 0 0 0;
   0 0 0 0 0 0;
   0 0 0 0 0 0];
```

2. Calcul de la matrice de covariance Q

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Estimation du bruit du capteur
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%accélération à l'état statique (unité en g)
a_statique=acceleration0;
ax0=a_statique(:,1);
ay0=a_statique(:,2);
az0=a_statique(:,3);
% la covariance à l'état statique (variance de bruit)
sigmax=var(9.81*ax0);
sigmay=var(9.81*ay0);
sigmaz=var(9.81*az0);
%Matrice de covariance du système (contient les covariances de la commande)
Q=[sigmax      0        0;
           0       sigmay 0;
           0          0       sigmaz];
```

3. Expression de la commande et initialisation du vecteur d'état

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Expression des vecteurs et initialisation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Expression de la commande (accélérations sur les axes x,y et z)
U=9.81*[ax ay az];
X(:,1)=[0 0 0 0 0 0]';%Initialisation du vecteur d'état à 0
```

4. Le filtre de Kalman : Estimation de la position sur x, y et z

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Filtre de Kalman avec les données capteur et GPS (recalage avec le GPS)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


indice=recal_agps;

t=0;
 for j=2:N1
  for n=1:N2
 X(:,j)=A*X(:,j-1) + B*U(j-1,:)';

 %Prédiction de la Matrice de covariance d'état
 Ppred=A*P*A' + M*Q*M';
 P=Ppred;

 if (j==indice(n))
 %Innovation sur les mesures du GPS
 In(:,j)=Y_gps(:,n)-(C*X(:,j));
 % On détermine le gain
 K=Ppred*C'*inv(C*Ppred*C' + R);

 %Mise à jour de la matrice de covariance d'erreur
 Pnew=(eye(6) - K*C)*Ppredk;
 Ppredk=Pnew;
 Pk=Ppredk;
 %Mise à jour de l'état
 Xnew(:,j)=X(:,j) + K*In(:,j);
 X=Xnew;
 Y(:,j)=C*X(:,j);
 %Modèle d'observation
else

end

end
end

%observation des positions x,y et z prédites (vecteur colonne)
Ykalm=Y;
xkalm=Ykalm(:,1);
ykalm=Ykalm(:,2);
zkalm=Ykalm(:,3);
```

On fait une innovation que lorsque la commande arrive au même moment que l'observable ou tout juste après

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Reconstruction de la trajectoire avec les 2 filtres de Kalman précédents et le GPS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 figure(1)
 plot3(xgps,ygps,zgps,'*g')
 plot3(xkalm,ykalm,zkalm,'or')
 hold off
 grid
 xlabel('x');
 ylabel('y');
 zlabel('z');
```

On observe

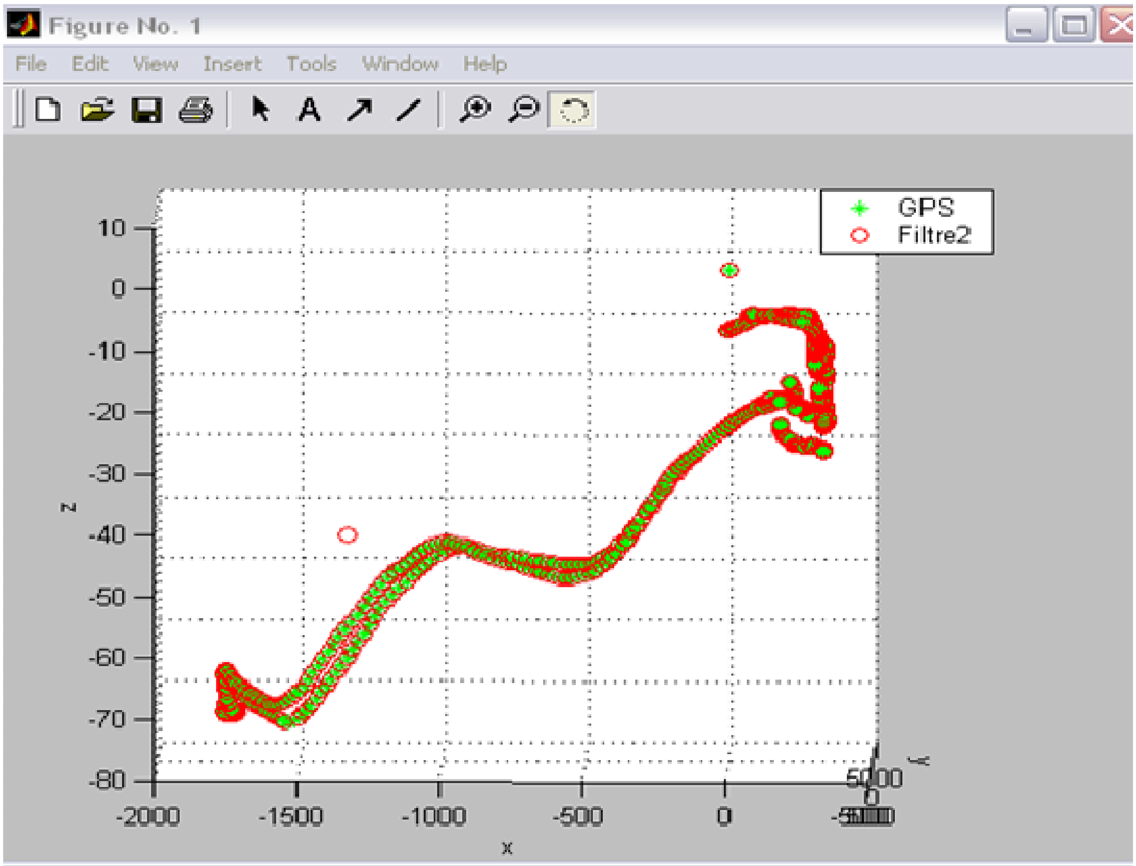

Fig3 : Superposition de la courbe donnée par le GPS et celle donnée après filtrage par filtre de Kalman

Le GPS représente la courbe en vert et les données issues du filtre de kalman sont représentées en rouge. On observe que le filtre suit bien le GPS.