Universitè Sultan Moulay Slimane
École nationale des sciences appliquées de Khouribga

Rapport de projet traitement du signal

Filtre de Kalman

Realisé Par :

EL KREM ABDELHAMID

OUASSIF EL MAHDI

EL GATIA HAMZA

ELOUAGDI AYOUB

Encadré Par :

Aboutabit Noureddine

Table des matières

Remerciements

Notre gratitude est pleinement exprimée en faveur de notre encadrant le professeur Mr Aboutabit Noureddine, c'est grace a son agréable qualité d'enseignement, ses qualités humaines, son professionnalisme ainsi que leur bienveillance qu'on a pu acquérir de nouvelles connaissances en Traitement de signal et dans le principe de filtrage.

Introduction

la fonction de filtrage consiste a estimer une information (signal) utile qui est polluée par un bruit. Alors que le filtrage fréquentiel suppose qu´ il existe une s´ eparation entre les réponses fréquentielles du signal utile et du bruit et consiste a trouver une fonction de transfert satisfaisant un gabarit sur le gain de sa réponse fréquentielle (et beaucoup plus rarement sur la courbe de phase), le filtre de Kalman vise a estimer de facon "optimale" l´ état du système linéaire (cet état correspond donc a linformation utile).

Chapitre 1

Modèle Generale

Biographie : Rudolf Kalman

Rudolf Emil Kalman (en hongrois Kálmán Rudolf Emil) (19 mai 1930 á Budapest - 2 juillet 20161) est un mathématicien et un automaticien américain d'origine hongroise, ingénieur en électrotechnique de formation, connu pour l´ invention du filtre de Kalman.

Figure 1.1 – Rudolf Kalman

Rudolf Kalman naît á Budapest en Hongrie. Il obtient son bachelor´ s degree (licence) en 1953 et son master´ s degree en 1954 au MIT, en ingénierie électrique, puis son doctorat en 1957 à l´ université Columbia.
Il travaille ensuite comme chercheur en mathématiques au Research Institute for Advanced Study (en) de Baltimore, de 1958 Ã 1964. De 1964 à 1971, il est professeur à l´ université Stanford,
puis Graduate Research Professor et directeur, au Center for Mathematical System Theory, à l´ université de Floride à Gainesville, de 1971 à 1992. De 1969 Ã 1972, il vient régulièrement à l´ Ecole des Mines de Paris à Fontainebleau où il est conseiller scientifique pour le Centre de recherches en automatique. À partir de 1973, il occupe la chaire de théorie des systèmes mathématiques à l´ École polytechnique fédérale de Zurich (EPFZ).

Histoire

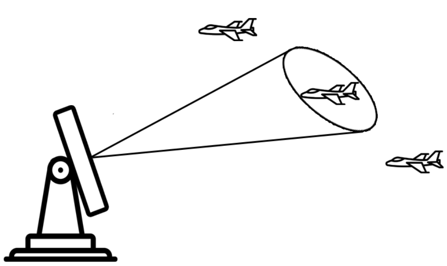
Le filtre doit son nom à l´ émigré hongrois Rudolf E. Kálmán, bien que Thorvald Nicolai Thiele et Peter Swerling aient développé un algorithme similaire plus tôt. Richard S. Bucy de l´ Université de Californie du Sud a contribué à la théorie, ce qui l´ a amenée à être parfois appelée filtre de Kalman-Bucy. Stanley F. Schmidt est généralement reconnu pour avoir développé la première implémentation d´ un filtre de Kalman. Il a réalisé que le filtre pouvait être divisé en deux parties distinctes, avec une partie pour les périodes de temps entre les sorties du capteur et une autre partie pour incorporer les mesures. C´ est lors d´ une visite de Kàlmán au NASA Ames Research Center que Schmidt a vu l´ applicabilité des idées de Kálmán au problème non linéaire d´ estimation de trajectoire du programme Apollo conduisant à son incorporation dans l´ ordinateur de navigation Apollo. Ce filtre de Kalman a été décrit pour la première fois et partiellement développé dans des articles techniques de Swerling (1958), Kalman (1960) et Kalman et Bucy (1961)

Figure 1.2 – NASA

Définition du filtre

Principe

Le filtre de Kalman est une méthode visant à estimer des paramètres d´ un système évoluant dans le temps à partir de mesures bruités. On retrouve ce filtre dans bon nombre de domaines relatifs au traitement du signal, radar, traitement d´ images etc. Un exemple d´ utilisation de ce filtre pourrait être la détermination de la position et de la vitesse d'un véhicule à partir de données GPS fournis par plusieurs satellites.



Points forts

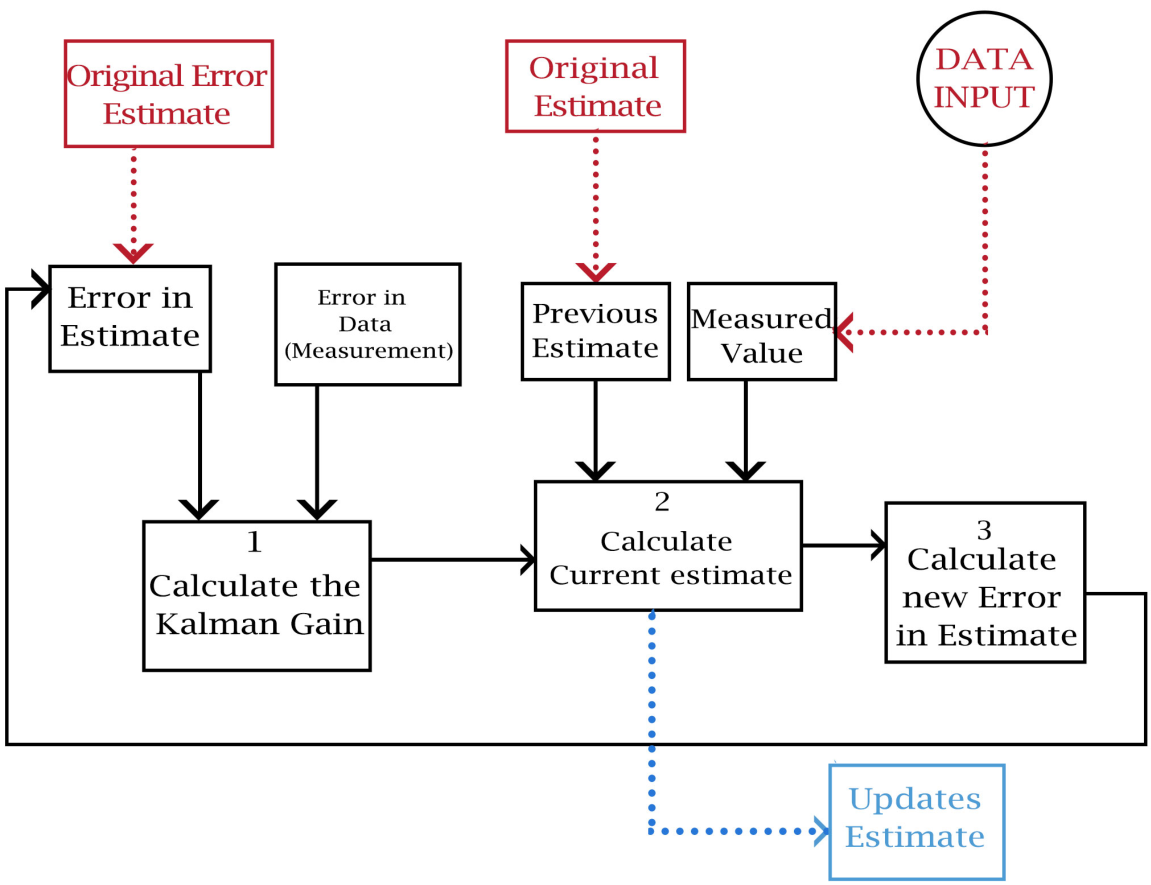
La force de ce filtre est sa capacité de prédiction des paramètres et de rectification des erreurs, non seulement des capteurs, mais aussi du modèle lui-même ! En effet, pour appliquer un filtre de Kalman, il faut avant tout modéliser le système pour lequel on veut estimer les paramètres, de manière linéaire. (des variantes du filtre de Kalman existent pour la prise en compte de modèles non linéaires) Dans une méthode d´ estimation classique (par exemple, la méthode des moindres carrés), une simple erreur dans la modélisation du système entraine inévitablement une erreur au niveau de l´ estimation. La force du filtre de Kalman est d´ intégrer un terme d´ imprécision sur le modèle lui-même, ce qui lui permet de donner des estimations correcte malgré les erreurs de modélisation (pour peu que les erreurs restent raisonnables).

Un autre point fort du filtre de Kalman (mais que l´ on retrouve aussi dans la méthode des moindres carrés par exemple) est sa capacité à déterminer l´ erreur moyenne de son estimation. En effet, l´ outil Kalman fournir un vecteur contenant les paramètres estimés, mais aussi une matrice de covariance de l´ erreur ! Cette matrice nous renseigne donc sur la précision de l´ estimation, ce qui peut être utile dans de nombreuses applications. Un autre atout du filtre de Kalman est que la convergence de cette erreur est garantie !

Le fonctionnement du filtre

1. Une première étape de prédiction de l´ estimation selon le modèle du système. Pour ce faire, le filtre de Kalman reprend l´ estimation précédente des paramètres et de l´ erreur et prédit les nouveaux paramètres et la nouvelle erreur en fonction de la modélisation du système.

2. La seconde étape va faire la mise à jour de cette prédiction grâce aux nouvelles mesures. Ces mesures (par définition bruités) vont permettre d´ obtenir une estimation des paramètres et de l´ erreur à partir de la prédiction faite. Si jamais le modèle comporte des erreurs, cette étape de mise à jour permettra de les rectifier.



Chapitre 2

Modèle Mathematique

Hypothèses

Pour pouvoir appliquer le filtre de Kalman conventionnel il faut :

1. Un système dynamique linéaire bruité. L´ évolution du système peut donc s´ écrire sous la forme :

$$x_{k+1} = F_k x_k + G_k u_k + w_k$$

   où F_k et G_k sont des matrices qui dépendent de la modélisation du système, x_k est un vecteur représentant l´ état du système, u_k est un vecteur représentant les commandes appliquées et w_k est un bruit. Chaque terme est pris à l´ instant k.

2. Des mesures, liées linéairement à l´ état du système, et bruitées :

$$y_k = H_k x_k + v_k$$

   où y_k est le vecteur contenant les mesures effectuées à l´ instant k, H_k est une matrice, et v_k le bruit de mesure.

Algorithme

Prediction l′état suivant d′oiseau avec le dernier état et le mouvement prédit.

$$X_{est} = A * X_{est} + B * u;$$

predict next covariance

$$P = A * P * A' + cov;$$

covariance de mesure observateur prévue

Gain de Kalman

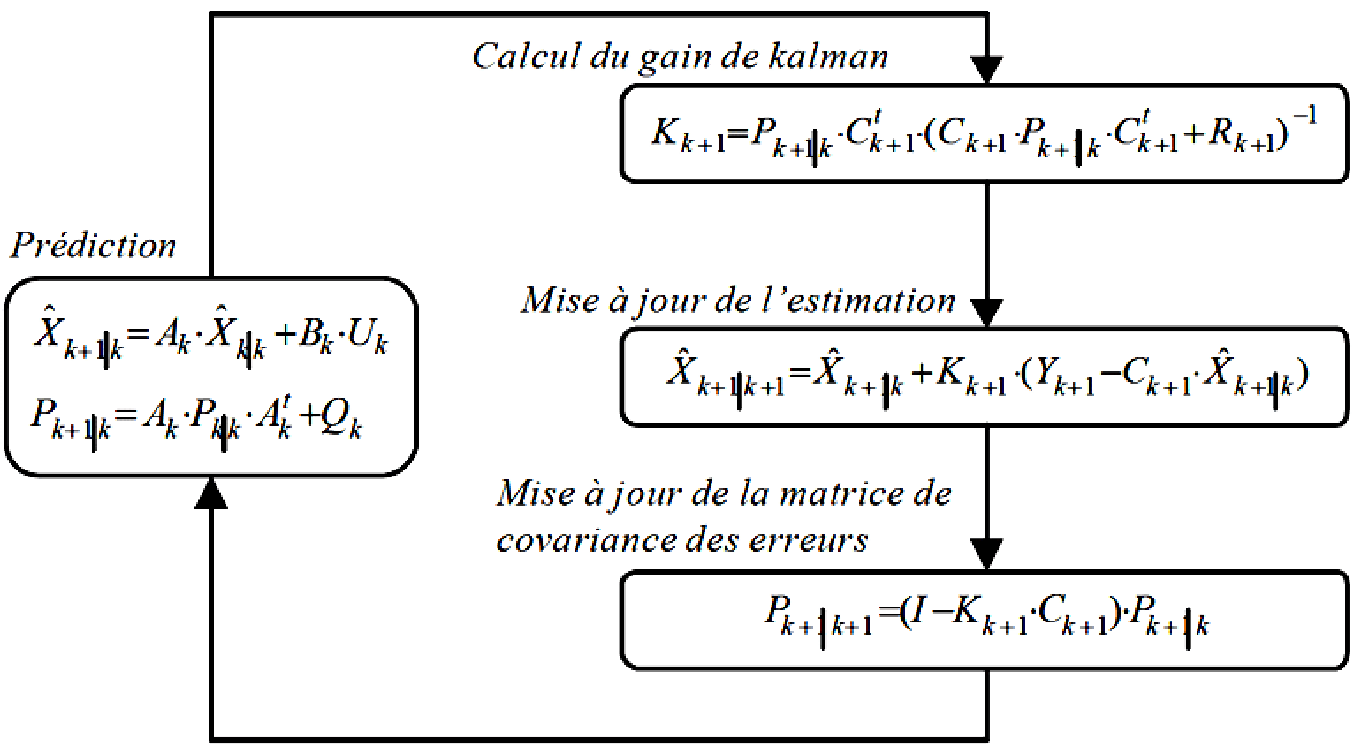
$$K = P * C' * inv(C * P * C' + var);$$

Mettez à jour l′estimation de l′état.

$$X_{est} = X_{est} + K * (X_{pos_obs}(t) - C * X_{est});$$

mettre à jour l′estimation de la covariance.

$$P = (eye(2) - K * C) * P;$$

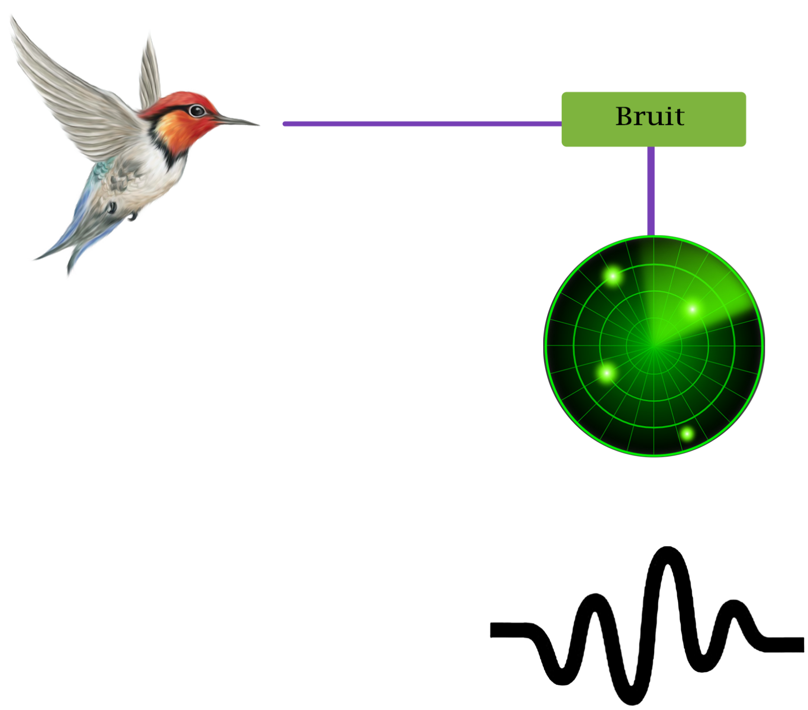


Chapitre 3

Implementation matlab

Probleme

Pour implemention de fitre de Kalman dans un problème partique nous avons considère problème suivant : Problème : nous avons prends cas de poursuit un oiseau qui on veut savoir sa position et que nous avons une appareil numérique qui donne des captures mais avec bruit :



Solution

utilisation d′un filtre pour se corrige signal ou donnée par filtre de Kalman qui un filtre d′estimation optimal. Tout d′abord pour applique les filtre il faut dèfinir les equation qui define le system physique : A un instance donne t on a :

Eq1 : (equation de mouvement) $x = x_0 + t * V + a * t^2/2$.

Eq2 : (equation de vitesse) $V = V + t * a$.

x : position

x_0 : position initail

V :vitesse, a : acceleration

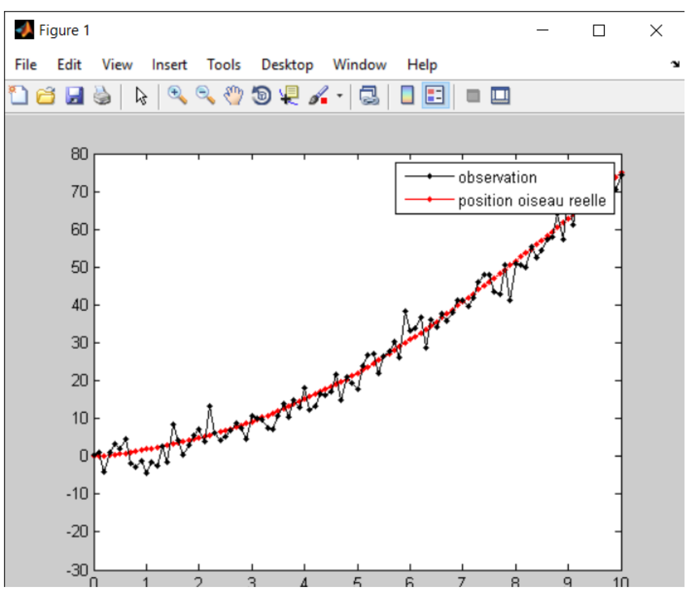
On considere que etat defini par position + vitesse : $X = [x; v]$
Alors systeme sera sous forme : $X = A*X + B*u$ Avec $u(variabledecontrole) = a$, $A = [1t; 01]$ et $B = [t2/2; t]$.
Donc il s′agit d′un systeme lineaire alors filtre de kalman est fausable.

1er etape : (collection des donnees)
Nous avons faire une simulation sous matlab du probleme poser :

```
%% d?finir les variables principales
u = 1.5; % d?finir l'amplitude d'acc?l?ration
X= [0; 0]; %?tat initial - il a deux composantes: [position; vitesse] d'oiseau.
X_est = X;  %x_est de l'estimation initiale de l'emplacement d'oiseau (ce que nous mettons ? jour)
bruitAccelOiseau_amp = 2.05; %bruit de processus: la variabilit? de la vitesse ? laquelle l'oiseau acc?l?re (ecart-type d'acc?l?ration: m?tres / sec ^ 2)
bruitObservation_amp = 3;  %bruit de mesure: incertitude l'observateur (ecart-type de l'emplacement, en m?tres)
var = bruitObservation_amp^2;% var: convertir le bruit de mesure (ecart-type) en matrice de covariance = variance
cov = bruitAccelOiseau_amp^2 * [dt^4/4 dt^3/2; dt^3/2 dt^2]; % cov: convertir le bruit de processus (ecart-type) en matrice de covariance
P = cov; % P : estimation de la variance de position initiale de Quail (matrice de covariance)
```

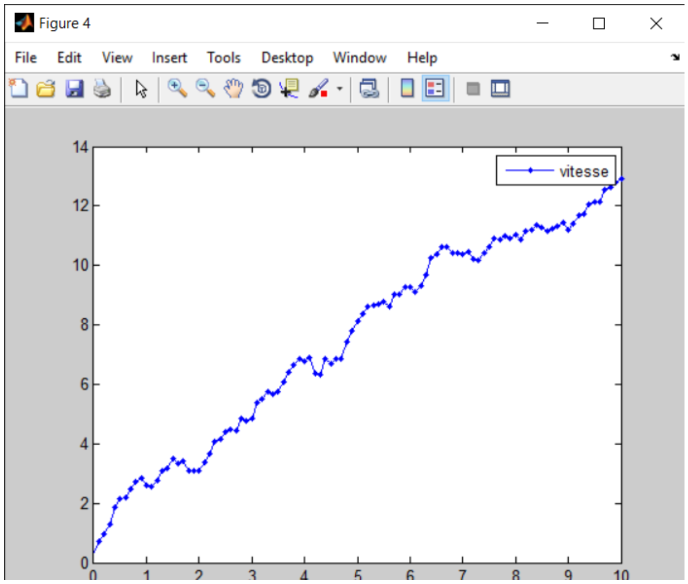
```
%% simulation ce que l'observateur voit au fil du temps
figure(2);clf
figure(1);clf
for t = 0 : dt: duree
    % Generer le vol Oiseau
    bruitAccelOiseau = bruitAccelOiseau_amp * [(dt^2/2)*randn; dt*randn];
    X= A * X+ B * u + bruitAccelOiseau;
    % G?n?rez ce que l'observateur voit
    bruitObservation = bruitObservation_amp * randn;
    y = C * X+ bruitObservation;
    X_pos = [X_pos; X(1)];
    X_pos_obs = [X_pos_obs; y];
    v = [v; X(2)];
    %iteratively plot what the ninja sees
    plot(0:dt:t, X_pos, '-r.')
    plot(0:dt:t, X_pos_obs, '-k.')
    axis([0 duree -30 80])
    legend('observation','position oiseau reelle');
    hold on
    pause(0.01);
end
```

Figure des donnees brute :



Comme montre le figure ci-dessus les donnees que nous avons capter a etait introduit avec un bruit du a capture que nous avons utiliser qui a un incertitude du valeur mesure.

Figure de la vitesse, juste pour montrer qu'elle augmente constamment, en raison de l'acceleration constant(il y a un bruit a cause de acceleration.)



Donc pour faire un balayage du bruit introduit on utilise le filtre de kalman :

```
%% Faites le filtrage kalman
% initier les variables d'estimation
X_pos_est = []; % position estimer d'oiseau.
v_est = []; % vitesse estimer d'oiseau.
X= [0; 0]; % reinitialise etat.
P_est = P;
P_amp_est = [];
predic_etat = [];
predic_var = [];
for t = 1:length(X_pos)
    % Prediction l'etat suivant d'oiseau avec le dernier etat et le mouvement predit.
    X_est = A * X_est + B * u;
    predic_etat = [predic_etat; X_est(1)] ;
    %predict next covariance
    P = A * P * A' + cov;
    predic_var = [predic_var; P] ;
    % covariance de mesure observateur prevue
    % Gain de Kalman
    K = P*C'*inv(C*P*C'+var);
    % Mettez ? jour l'estimation de l'etat.
    X_est = X_est + K * (X_pos_obs(t) - C * X_est);
    % mettre a jour l'estimation de la covariance.
    P =  (eye(2)-K*C)*P;
    % Stoker pour le tracage
    X_pos_est = [X_pos_est; X_est(1)];
    v_est = [v_est; X_est(2)];
    P_amp_est = [P_amp_est; P(1)];
end
```

1 er etape :

faire une predection de etat suivant apartir du equetion introduit par le systeme

$etatestime_{k|k-1} = A * etatestimer_{k-1} + B * u_k;$

$X(k|k-1) = A * X(K-1) + B * u(k);$

2 eme etape :

Puis on fiat une prediction de matrice de covarince P sachant que $P(k-1)$ est realiser :

$P(k|k-1) = AP(k)A' + cov;$

Avec cov est la matrice d′erreur ou de bruit mise en calcule.

3 eme etape :(Kalman gain) :

$$\text{Gain kalman} = \frac{Erreur\_prediction}{Erreur\_prediction + Erreur\_mesure}$$

$$KG = \frac{P * C'}{C * P * C' + R}$$

Avec C est matrice de passage, et R matrice de covarince de mesure.(observation)

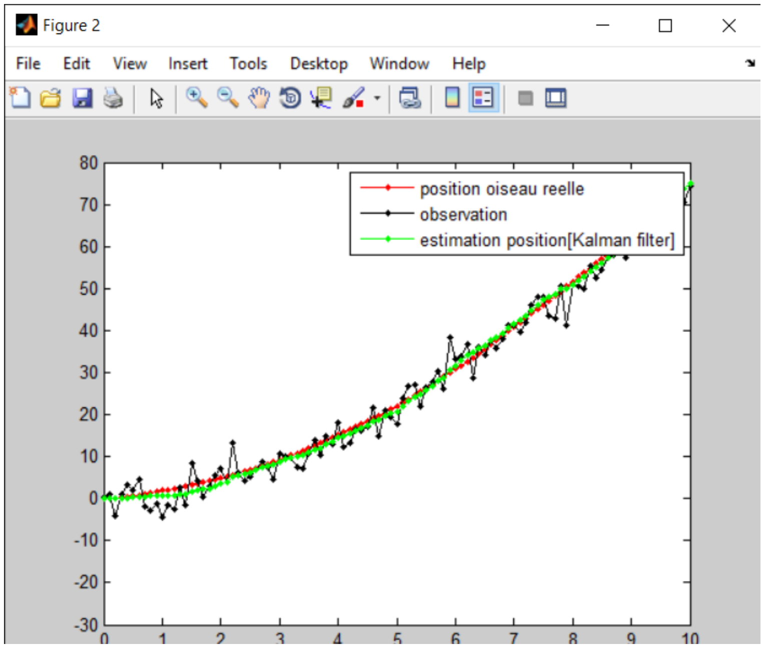
4 eme etape: (update de estimation d'etat):

```
X_estimer_corriger  = X_estimer + KG*(Mesure - C'  * X_estimer)
```

5 eme etape: (update matrice de covarince):

```
P_est_corriger = (I - KG*C)*P_est;
```

Apres les resulta que nous avons obetenus:



Pour les dutributions de probalibilite des resultats :

```
for T = 1:length(X_pos_est)
clf
    x = X_pos_est(T)-5:.01:X_pos_est(T)+5; % plage sur l'axe des x

    %prochaine position pr?dite d'oiseau
    hold on
    m = predic_etat(T); % moyenne
    sigma = predic_var(T); % ecart-type
    y = normpdf(x,m,sigma); % pdf: fonction densite de probabilite.
    y = y/(max(y));
    hl = line(x,y,'Color','m');

    %donn?es mesur?es par l'observateur.
    m = X_pos_obs(T); % moyenne
    sigma = bruitObservation_amp; % ecart-type
    y = normpdf(x,m,sigma); % pdf
    y = y/(max(y));
    hl = line(x,y,'Color','k');

    %estimation de position combin?e
    m = X_pos_est(T); % moyenne
    sigma = P_amp_est(T); % ecart-type
    y = normpdf(x,m,sigma); % pdf
    y = y/(max(y));
    hl = line(x,y, 'Color','g');
    axis([X_pos_est(T)-5 X_pos_est(T)+5 0 1]);


    %position r?elle d'oiseau
    plot(X_pos(T));
    ylim=get(gca,'ylim');
    line([X_pos(T);X_pos(T)],ylim.','linewidth',2,'color','b');
    legend('prediction etat','observation','estimation etat[kalman filter]','position oiseau reelle')
```

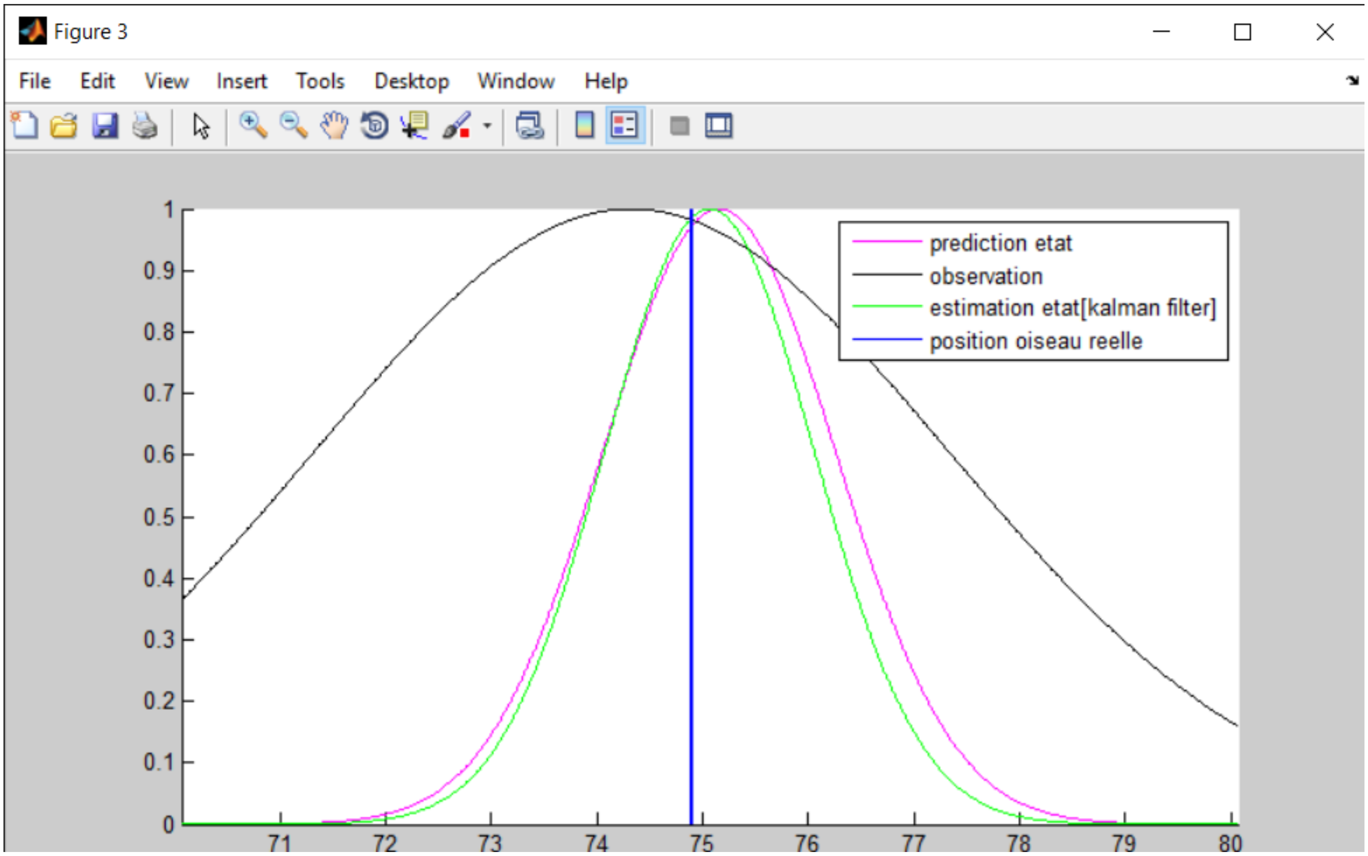
Comme on fait une prediction donc il a toujour l'aleatiore que nous avons contater que tout parametre suit lois normal :

$X_{etimer} \rightarrow N(m, \sigma)$

$Y \rightarrow N(m, \sigma)$

$X_{etimer_corriger} \rightarrow N(m, \sigma)$

Figure suivant montre les fonction de dutribution de probabilite :



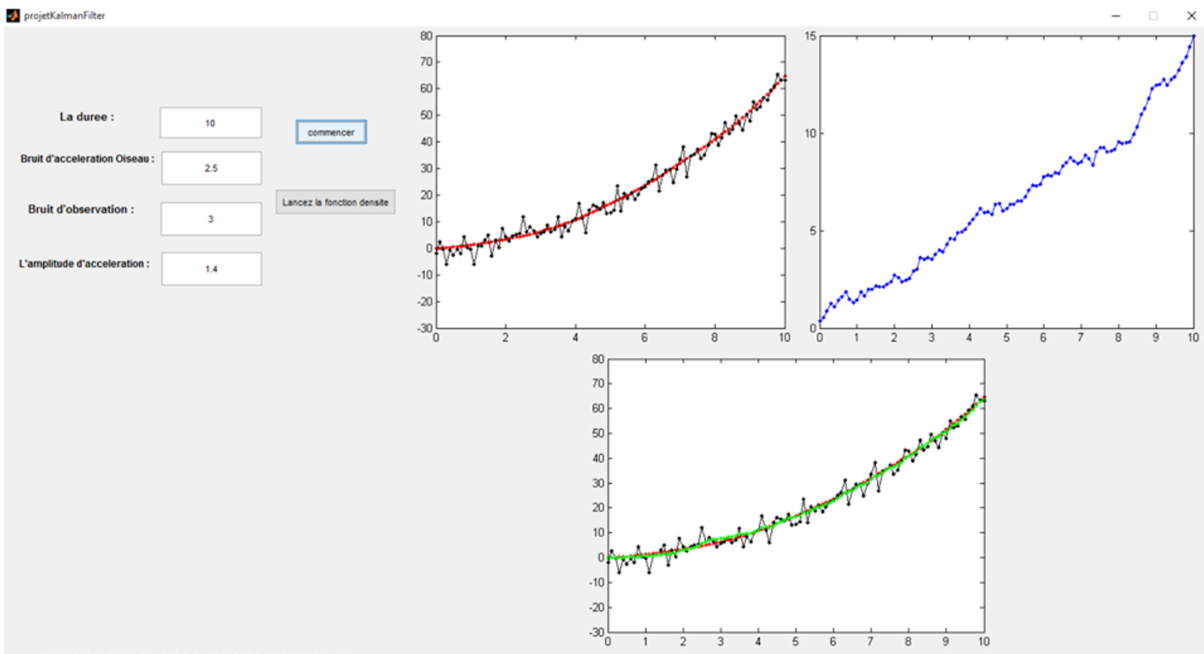
Chapitre 4

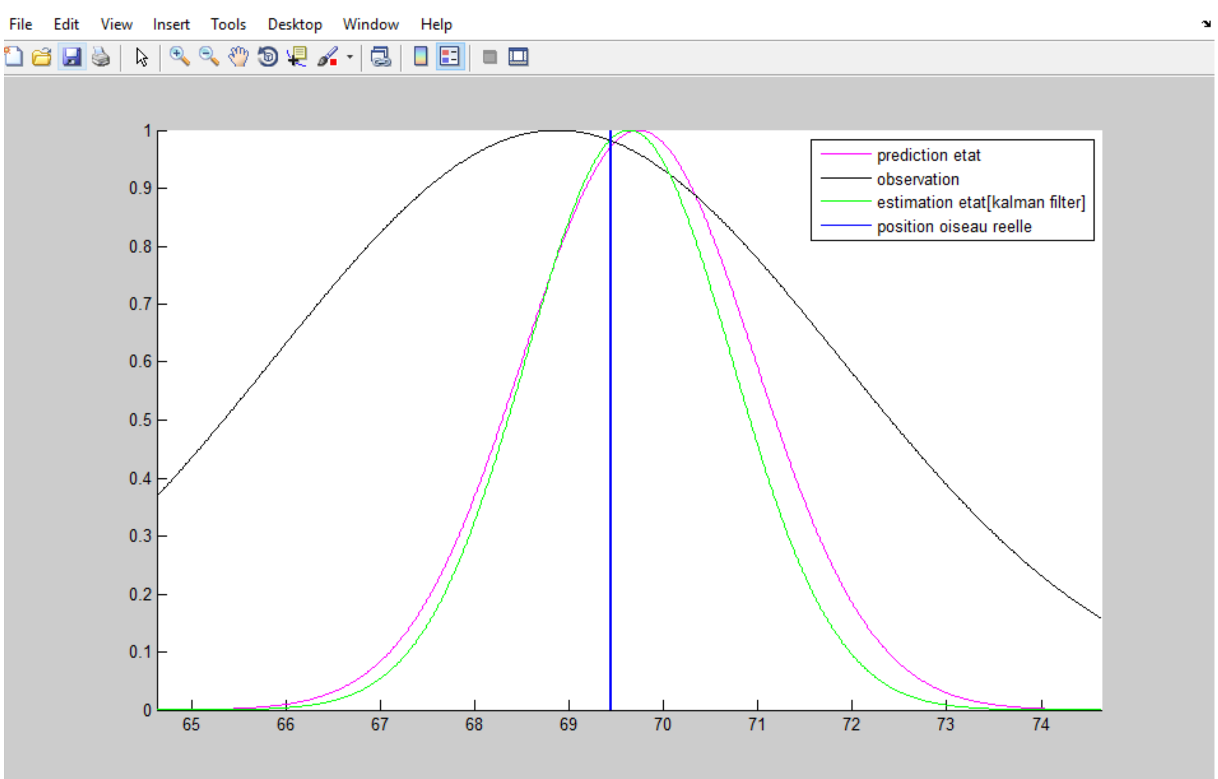
interface graphique

On realise l′interface graphique qui permet a l′utilisateur de saisir quatre valeurs : La duree utilisee pour appliquer le filtre.

- Le bruit de l′acceleration d′un oiseau que nous suivons.
- Le bruit de l′ibservation prise sur l′appareil.
- Amplitude d′acceleration.

Nous representons l′observation et la mesure dans un graphique1 Ensuite, Nous representons dans un graphique1 en ajoutant un filtre de Kalman Ensuite, nous representons dans le graphique3 vitesse
Ensuite, Nous representons dans un graphique4 fonction de densite


projetKalmanFilter
La duree :
10
commencer
Bruit d'acceleration Oiseau :
2.5
Bruit d'observation :
3
Lancez la fonction densite
L'amplitude d'acceleration :
1.4


File Edit View Insert Tools Desktop Window Help
prediction etat
observation
estimation etat[kalman filter]
position oiseau reelle

Chapitre 5

Conclusion

Le filtre de Kalman est donc une méthode d′estimation et de prédiction puissante prenant en compte les modelisations du systeme. Neanmoins, ce filtre n′est pas forcement l′outil a appliquer dans tous les cas. En effet, comme nous l′avons vu, le developpeur a besoin de modeliser le systeme assez precisement afin de designer un filtre efficace. Le probleme est que certains systemes sont difficilement modelisable et, encore moins lineairement. Dans le cas ou la modelisation est trop approximative, le filtre n′est pas assez performant et l′erreur des estimations ne convergera pas assez rapidement, elle restera grande.

Pour palier Ã ce probleme de modelisation lineaire du systeme, un filtre de Kalman etendu a ete developpe et permet de prendre en compte une modelisation non lineaire. Neanmoins, cette technique a quelques defauts. En premier lieu, la covariance de l′erreur (la precision des estimations) ne converge pas obligatoirement (comme c′etait le cas avec une modelisation lineaire). Le second defaut est sont cout calculatoire plus important. En effet, de nouvelles matrices couteuses rentre en jeux (les matrices des derivÃ©s partielles des equations d′etats et de mesures modelisant le systeme), ce qui peut etre une limite a son utilisation, surtout dans des systemes embarques tres restreint au niveau de la puissance de calcul.

Une autre limite importante d′une telle methode est que le filtre de Kalman permet de prendre en compte uniquement un modele de bruit Gaussien. Le bruit peut en general etre modelise de faÃ§on Gaussienne, mais dans certains cas, un autre type de bruit est requit (notamment en traitement d′images ou l′on utilise frequemment des bruits de Poisson). Cette restriction limite donc l′utilisation du filtre de Kalman.

Le filtre de Kalman est donc une methode d′estimation interessante, mais qui n′est utilisable que lorsque l′on peut decrire assez precisement notre systeme. S′il est impossible de trouver une modelisation correcte du systeme, il est alors preferable de se tourner vers d′autres methodes (comme la methode de Monte-Carlo par exemple qui est une methode statistique, mais qui requiert une importante puissance de calcul)