TP 1 : FILTRE DE KALMAN : IMPLÉMENTATION

Exercice 1 (Simulation dans le modèle linéaire Gaussien)**.**
On se limite au cas où les paramètres régissant l'évolution de l'état et le modèle d'observation sont indépendants de l'indice 'temporel' $k \geq 0$. On note $\mathcal{M}_{r,s}$ l'ensemble des matrices réelles de taille $r \times s$, $\mathcal{S}_r$ l'ensemble des matrices symétriques semi-définies positives de taille $r \times r$. On rappelle la définition d'un MLG :

$$\begin{cases} X_0 \sim \mathcal{N}(m_0, P_0) \\ X_{k+1} = AX_k + \epsilon_k \\ Y_{k+1} = BX_{k+1} + \eta_{k+1} \end{cases} \tag{1}$$

où $m_0 \in \mathbb{R}^p, P_0 \in \mathcal{S}_r^+$, $A \in \mathcal{M}_{p,p}$, $\epsilon_k \sim \mathcal{N}(0, Q)$, $Q \in \mathcal{S}_p^+$, $B \in \mathcal{M}_{q,r}$, $\eta_k \sim \mathcal{N}(0, R)$, $R \in \mathcal{S}_q^+$, et où les vecteurs aléatoires $(X_0, (\epsilon_k)_{k\geq 0}, (\eta_k)_{k\geq 0})$ sont indépendants.

Les paramètres à spécifier en entrée sont

- Les moyennes et variances de l'état initial m_0, P_0 ;
- Les matrices A, B définissant respectivement l'équation d'évolution de l'état et le modèle d'observation
- les matrices de covariances des bruits Q, R.

1. Relation de récurrence du modèle
   - Écrire une fonction `lingauss_step` sur le modèle suivant, pour générer (X_{k+1}, Y_{k+1}) à partir de l'état courant X_k.

```
function [x, y] = lingauss_step(x_current, A, B, Q, R)
 % Genere (X_{k+1}, Y_{k+1}) dans
 % le modele lineaire gaussien.
 % ARGUMENTS:
 % A, B: matrices definissant la dynamique de l'etat et
 %     le modele d'observation (de tailles resp. (p,p) et (q,p))
 % Q, R: variances des bruits pour l'equation d'etat et
 %     l'equation d'observation.
 % x_current: vecteur de taille p
 % %
 % VALEUR:
 % x: un vecteur colonne (dimensions (p,1))
 % y : un vecteur colonne (dimensions (q,1)
  dim_state = max(size(x_current));
  dim_obs = size(B,1) ;
  x_current = reshape(x_current, dim_state, 1);
 %
 % completer le code
 %
  y = reshape(y , dim_obs,1);
end
```

   - Testez votre fonction pour $p = 3$, $q = 2$, sur le script suivant

```

%% 1.1 parametres par default
x_0 = [1; 2; 3];
A = 0.1 * [1 , 1, 0 ;
        0, 1 , 1 ;
        0, 0 , 1 ]
B = [1, 0, 0;
    0, 1, 1];
cholQ = [0.5, 0.5, 0.5 ;
```

```
        0, 1, 1;
        0, 0, 1];
Q = cholQ*cholQ';
cholR = 0.5*[1, 1;
         0, 2];
R = cholR*cholR';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% 1.2 une iteration dans le modele: lingauss_step
[x,y] = lingauss_step(x_0, A, B, Q, R )
```

2. Écrire une fonction `lingauss_simul` sur le modèle suivant, pour générer le processus $(X_k, Y_k, k = 1 : T)$ à horizon fini T, à partir d'un état initial x_0, sur le modèle suivant

```
function [X, Y] = lingauss_simul(x_0, T, A, B, Q, R)
 % Genere le processus ((X_{k}, Y_{k}), k = 1:T)
 % dans le modele lineaire gaussien
 % ARGUMENTS:
 % A, B: Matrices definissant la dynamique de l'etat et
 %     le modele d'observation (de tailles resp. (p,p) et (q,p))
 % Q, R: Variances des bruits pour l'equation d'etat et
 %     l'equation d'observation.
 % x_0: Etat initial
 % T : nombre d'iterations
 % %
 % VALEUR:
 % X: une matrice de taille (T,p)
 % Y : une matrice de taille (T,q)
 % N.B: p et q sont determines par les arguments passes A et B.
  dim_state = size(A,1);
  dim_obs = size(B,1);
  X = zeros(T,dim_state);
  Y = zeros(T,dim_obs);
  x = x_0;

  for i = 1:T
   %
   % Completez le code : utilisez la fonction
   % lingauss_step de la question precedente
   %
  end

end
```

3. Exemple élémentaire : marche aléatoire.
Pour tester votre simulateur, essayez-le sur le modèle le plus simple possible, *i.e.,* le cas où X suit une marche aléatoire et Y consiste en l'observation bruitée de certaines composantes de X.
On fixe $p = 2, q = 3$, et on note $X_k = (X_{k,1}, X_{k,2}, X_{k,3}), Y_k = (Y_{k,1}, Y_{k,2})$. On considère le cas particulier de (1) où pour $k \geq 0$, pour $i \in \{1, 2, 3\}$, $j =\in \{1, 2\}$,

$$\begin{cases} X_{k+1,i} = X_{k,i} + \epsilon_{k,i}, \text{ avec } \quad \epsilon_{k,i} \sim \mathcal{N}(0, \tau^2) \\ Y_{k+1,j} = X_{k+1,j} + \eta_{k,j}, \text{ avec } \eta_{k,j} \sim \mathcal{N}(0, \sigma^2). \\ \qquad (\epsilon_{k,1}, \epsilon_{k,2}, \epsilon_{k,3}) \text{ indépendants}, \ (\eta_{k,1}, \eta_{k,2}) \text{ indépendants} \end{cases} \tag{2}$$

- Écrire ce modèle sous la forme (1), en spécifiant les paramètres A, B, Q, R.
- Simuler ensuite $T = 100$ observations et afficher les trajectoires en vous inspirant du script (incomplet) suivant

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%A: Simulation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% A-1 parametres \'a modifier
sigma = 10;
tau = 40;
x_0 = [1,2,-1];
T = 500;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% A-2: parametres structurels
A = eye(3);
B = %% Completer
Q = sigma^2 * %% Completer
R = tau^2 * %% Completer
[X, Y] = lingauss_simul(x_0, T, A, B, Q, R);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%B: visualisation, eventuellement 'step by step'
close all;
figure(1)
hold on;
plot(X(:,1), X(:,2), 'b');
plot(Y(:,1), Y(:,2),'og');
legend('State', 'Observation')
title('Random walk observed with uncorrelated noise')
hold off;

for i = 1:2
figure(i+1)
hold on;
plot(X(:,i), 'b');
plot(Y(:,i), 'og');
title(strcat('component ', num2str(i) ));
legend('State', 'Observation');
hold off;
end

% figure(4)
% disp('<push any key to proceed to next time steps>');
% clc;
% hold on;
% for k=1:T
%  if rem(k,10)==1
%    plot(X(1:k, 1), X(1:k,2), 'b') ;
%    plot( Y(1:k, 1), Y(1:k, 2),'og');
%    drawnow;
%    pause ;
%   end
% end
% hold off;
```

- Que représentent les figures obtenues ? Faites varier les paramètres sigma, tau, T et observez l'effet de ces modifications.

Exercice 2 (Filtre de Kalman)**.** Dans cet exercice on implémente le filtre proprement dit.

1. Écrire une fonction `kalman_predict` qui effectue l'étape de prédiction dans le filtre de Kalman, sur le modèle suivant

```
function [mpred, Ppred] = kalman_predict( m, P, A, Q )
 % effectue l'etape de prevision dans le filtre de Kalman.
 % Calcule la loi de prediction [X_{k+1} | Y_{1:k}]
 % ARGUMENTS:
 % m, P: moyenne et variance de la loi de filtrage courante
 %     vecteur et matrice de taille p
 % A : matrice p*p definissant la dynamique de l'etat
 % Q : variance des bruits (matrice p*p) pour l'equation d'etat
 % VALEUR:
 % mpred: vecteur colonne p * 1: moyenne de la loi de prediction.
 % Ppred: matrice p*p: covariance de la loi de prediction.
 dim_state = max(size(m)) ;
 m = reshape(m, dim_state, 1) ;
 mpred = %% Completer
 Ppred = %% Completer
end
```

2. Écrire maintenant une fonction `kalman_update` pour l'étape de mise à jour, sur le modèle suivant

```
function [mu, Pu] = kalman_update(y, mpred, Ppred, A, B, Q, R)
 % Etape de mise a jour dans le filtre de Kalman:
 % ARGUMENTS
 % mpred, Ppred: moyenne et variance de la loi predictive courante,
 %             valeur renvoyee par 'kalman_predict'
 % y: vecteur de taille q: nouvelle observation
 % A, B: Matrices p*p et q*p definissant la dynamique de l'etat
 %     et le modele d'observation
 % Q, R: Variances des bruits pour l'equation d'etat
 %     et l'equation d'observation.de taille p et q
 % VALEUR:
 % mu, Pu : moyenne et matrice de covariance de
 %        la loi de filtrage apres l'etape de mise a jour

  dim_state = size(A,1);
  dim_obs = size(B,1);
  S = %% Completer
  K = %% Gain de Kalman (Completer)
  mu = %% Completer
  Pu = %% Completer
end
```

3. Écrire enfin une fonction `kalman_filter` en complétant le modèle ci-dessous

```
function [M, Parray] = kalman_filter(Y, m_0, P_0, A, B, Q, R)
 % Filtre de Kalman:
 % Filtre sequentiellement les observations Y
 % pour reconstruire le signal X
 %%%
 % ARGUMENTS:
 % Y: les observations : une matrice T * q
 % m_0: vecteur colonne p * 1 : moyenne a priori de l'etat initial
 % P_0: matrice p*p: variance a priori de l'instant initial
 % A, B: matrices definissant la dynamique de l'etat
 %    et le modele d'observation
 % Q, R: variances des bruits pour l'equation d'etat
 %     et l'equation d'observation.
```

```
 %%%
 % VALEUR:
 % M: Matrice T * p: les moyennes a posteriori
 % Parray: Array de dimension p*p*T les matrices de covariances
 %        des lois de filtrages successives
  dim_state = size(A,1); %% dimension des etats
  dim_obs = size(B,1); %% dimension des observations
  T = size(Y,1);

%% initialisation
  P = P_0; m = m_0 ;
  M = zeros(T,dim_state);
  Parray = zeros(dim_state,dim_state, T);
  for k=1:T
    %% prediction
         % Completer
    %% mise a jour
         % Completer
    %% reaffectation de l'etat courant
      m = mu; P = Pu;
    %% stockage
      M(k,: ) = mu';
      Parray(:,:,k) = P;
  end
end
```

4. Essayez votre filtre sur les données Y générées à l'exercice 1. Vous pourrez vous inspirer du code suivant pour visualiser les résultats.
   N.B : certains objets définis à l'exercice 1 sont réutilisés ici, comme A, B, P, Q, X, Y par exemple.

```
  % Priors sur l'etat initial
m_0 = [0,0,0];
P_0 = 100 * eye(3);

% filtrage
[M, Parray] = kalman_filter(Y, m_0, P_0, A, B, Q, R);
% Visualisation
close all;
for i = 1:3
  figure(i)
  hold on;
  plot(1:T, M(:,i), 'r');
  plot(1:T, X(:,i), 'b');
  sd = reshape(sqrt(Parray(i,i,:) ), T,1);
  plot(1:T, M(:,i) + 1.96 * sd, 'y');
  if i <3
    plot(1:T, Y(:,i), 'g');
    legend('filtered', 'state', 'conf bands', 'observation')
  else
    legend('filtered', 'state', 'conf bands')
  end
  plot(1:T, M(:,i) - 1.96 * sd, 'y');
  title(strcat('component ', num2str(i)))
  hold off;
end

figure(4)
```

```
hold on;
plot(M(:,1), M(:,2),'r');
plot(X(:,1), X(:,2), 'b');
plot(Y(:,1), Y(:,2), 'og');
title('trajectory of the two first components')
legend('filtered', 'state', 'observations')
hold off;
```

- Que représentent les deux bandes jaunes ? Interprétez la figure 3.
- Faites varier les paramètres sigma, tau, T, et observez les conséquences.

5. On modifie le problème d'observation de la manière suivante : au lieu d'observer la 'projection' de la marche aléatoire sur le plan (O, x_1, x_2), on observe sa projection sur un plan ayant subi deux rotations successives d'angle ψ_B, θ_B autour des axes (Oy) et $(0x)$). Cette modification correspond à une nouvelle matrice d'observation B définie comme suit (valeurs de θ_B et ψ_B arbitraires) :

```
thetaB = pi/3
psiB = pi/3;
B =[1, 0, 0;
    0, 1, 0] * ...
   [ 1, 0, 0 ;
     0 , cos(thetaB), -sin(thetaB) ;
     0, sin(thetaB), cos(thetaB)] * ...
   [  cos(psiB), 0 , -sin(psiB) ;
      0  ,    1 , 0 ;
      sin(psiB), 0, cos(psiB)]
```

- Faites un croquis rapide de la situation
- Mettez en évidence le fait que le filtrage fournit maintenant de l'information sur les trois composantes du signal X (à partir d'une observation en 2D).

Exercice 3 (Car Tracking)**.** On modélise l'évolution d'une voiture 'sans conducteur' dans un plan en supposant que l'accélération est un "bruit blanc". Avec la théorie de l'intégrale stochastique, on formalise ceci en disant que la vitesse $V(t) = (V_1(t), V_2(t)), t \in \mathbb{R}^+$ est "l'intégrale d'un bruit blanc", c'est à dire un mouvement brownien. En intégrant encore ce mouvement brownien, on obtient la loi de la position $Z(t) = (Z_1(t), Z_2(t))$ de la voiture dans le plan. On note t_k les instants d'observations, $t_k = kh$ où $h > 0$ est le pas de temps, $k \in \mathbb{N}$, $Z_k = (Z_{k,1}, Z_{k,2}) = (Z_1(t_k), Z_2(t_k))$, $V_k = (V_{k,1}, V_{k,2}) = (V_1(t_k), V_2(t_k))$. Sans rentrer dans les détails de calcul stochastique, on obtient le modèle linéaire suivant sur le processus des états discrétisé $X_k = (Z_k, V_k) = (Z_{k,1}, Z_{k,2}, V_{k,1}, V_{k,2})$:

$$\begin{aligned} Z_{k+1,i} &= Z_{k,i} + hV_{k,i} + \sigma^2 \zeta_{k,i} \\ V_{k+1,i} &= V_{k,i} + \sigma^2 \omega_{k,i}, \qquad i = 1, 2. \end{aligned} \tag{3}$$

où $\epsilon_{k,i} = (\zeta_{k,i}, \omega_{k,i}) \sim \mathcal{N}\left(0, \begin{pmatrix} h^3/3 & h^2/2 \\ h^2/2 & h \end{pmatrix}\right)$ et où $\epsilon_{k,1} \perp\!\!\!\perp \epsilon_{k,2}$, pour $k \geq 0$.

1. Écrire le système (3) sous la forme canonique $X_{k+1} = AX_k + \epsilon_k$, où on précisera A et la matrice de covariance Q du bruit $\epsilon_k = (\epsilon_{k,1}, \epsilon_{k,2})$.
2. L'observation Y_k à chaque instant t_k, $k \geq 1$ est constituée de la position Z_k, à laquelle s'ajoute un bruit de mesure η_k, gaussien, de covariance $R_k = \tau^2 \,\mathrm{Id}_2$. Préciser la matrice B de l'équation d'observation $Y_k = BX_k + \eta_k$ dans cette situation.
3. En s'inspirant des exercices 1 et 2, simuler le processus discrétisé $(X_k, Y_k), k = 1 : T$, et représenter graphiquement l'évolution de la position et de la vitesse. Pour la position, on représentera sur le même graphique les observations et l'état, sur le même modèle que dans l'exercice 1.
4. Filtrez les observations Y. En vous inspirant de l'exercice 2, montrez empiriquement que le filtre de Kalman permet de 'reconstruire' la position et la vitesse à partir de la seule donnée des positions bruitées.