TP 2 : FILTRE DE KALMAN ÉTENDU (EKF) : CAR TRACKING PAR CAPTEURS ANGULAIRES

On considère un problème de 'car tracking', dans lequel le modèle d'évolution de l'état discrétisé $X_k = (Z_k, V_k) = (Z_{k,1}, Z_{k,2}, V_{k,1}, V_{k,2})$ (position, vitesse) est le même que dans l'exercice 3 de la feuille de TP n^o 1 (Kalman). On rappelle l'équation de ce modèle d'évolution :

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

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 $k \geq 0$.

La différence de ce cas d'étude par rapport à celui vu précédemment tient au modèle d'observation. Ici, on mesure indirectement la position Z de la voiture grâce à sa direction observée depuis deux capteurs C_1, C_2, où C_1 est placé à l'origine du repère $(O, \mathbf{e}_1, \mathbf{e}_2)$, et C_2 est le point de coordonnées $(0, L)$ (sur l'axe des ordonnées, à distance L de l'origine). On suppose que la voiture reste dans le demi-plan $\{x \geq 0\}$, *i.e.,* un mur est placé sur l'axe $(0y)$ et un crash se produit lorsque $Z_{k,1} \leq 0$. L'observation Y est alors le vecteur des deux angles $\theta_1 = (\mathbf{e}_1, \overrightarrow{C_1 Z})$ et $\theta_2 = (\mathbf{e}_1, \overrightarrow{C_2 Z})$, auquel s'ajoute un bruit gaussien. Autrement dit,

$$Y_k = \Big(\text{atan}(Z_{2,k}/Z_{1,k}), \text{atan}((Z_{2,k} - L)/Z_{1,k})\Big) + \eta_k,$$

où $\eta_k \sim \mathcal{N}(0, R)$, avec $R = \tau^2 \,\text{Id}_2$. Tous les bruits ϵ_k, η_k ainsi que l'état initial sont indépendants.

Le modèle d'observation est non linéaire, il faudra utiliser la version étendue du filtre de Kalman en linéarisant l'équation d'observation.

Exercice 1 (Simulation et reconstruction de l'état à partir des observations brutes).

1. Construire un simulateur de l'état caché X et de l'observation Y à horizon fini T, sur le modèle suivant.

```
   function [X,Y]= voiture_angle_simul(z_0, v_0, L, T, sigma, tau, h)
% Voiture brownienne avec observation des angles seulement:
% Etat: X = (Z_1, Z_2, V_1, V_2) (position vitesse : 4 dimensions)
% Observation: On observe l'angle de la direction de la voiture depuis 2 capteurs
% sur l'axe (0y), le premier en (0,0), le second en (0,L).
% La voiture s'arrete lorsque son abscisse atteint 0 (crash).
% ARGUMENTS
 % z_0: vecteur de taille 2; position initiale (premier element strictement positif
 % v_0: vecteur de taille 2, vitesse initiale.
 % L: position du deuxieme capteur sur l'axe vertical
 % T: horizon temporel: nombre d'iterations.
 % sigma, tau: parametres d'ecart type pour les bruits de
 %           l'equation d'etat et de l'equation d'observation.
 % h : pas de discretisation temporelle
 % %
 % VALEUR:
 % X: une matrice de taille (T,4)
 % Y : une matrice de taille (T,2)

  A = [ 1, 0, h, 0 ;
        0, 1, 0, h ;
        0, 0, 1, 0 ;
        0, 0, 0, 1 ] ;


  Q = sigma^2 * [ h^3/3 , 0 , h^2/2 , 0 ;
```

```
               0,     h^3/3,  0   , h^2/2  ;
               h^2/2,  0   ,  h    ,   0   ;
               0     , h^2/2   , 0     , h ] ;

  R = tau^2 * eye(2);
  X = zeros(0, 4);
  Y = zeros(0,2);
  x = [z_0, v_0];
  for i=1:T
    x = %% Completer
    if x(1)<0
      break;
    end
    X = [X;x];
    y_mean = %% completer
    y = %% completer (simulation de loi normale multivariee)
    Y = [Y;y];
  end
end
```

2. En l'absence d'un procédé de filtrage, on pourrait déduire des deux angles $Y = (Y_1, Y_2) = (\theta_{1,obs}, \theta_{2,obs})$ et de la distance L, la position 'observée' Z_{obs} de la voiture en coordonnées cartésiennes. Cette dernière est donnée implicitement par

$$tan(Y_1) = Z_{2,obs}/Z_{1,obs}$$
$$tan(Y_2) = (Z_{2,obs} - L)/Z_{1,obs}$$

   - Montrer que la distance à l'origine observée $r_{obs} = \|OZ_{obs}\|$ s'écrit

$$r_{obs} = L\frac{\cos(Y_2)}{\sin(Y_1 - Y_2)}.$$

   - En déduire l'expression de $(Z_{1,obs}, Z_{2,obs})$ en fonction de L, Y_1, Y_2.
   - Utilisez votre résultat pour tracer sur le même graphique la trajectoire 'réelle' de la voiture et sa trajectoire 'observée' par les deux capteurs, en complétant le code suivant

```
T = 3000;
L = 100;
h = 0.01; %% pas
sigma = 0.5 %% sd acceleration
tau = 0.01 %% sd observation
z_0 = [50,5];
v_0 = [0.5,0.5];
[X, Y] = voiture_angle_simul(z_0, v_0, L, T, sigma, tau, h);

R_obs = %% Completer
Z1_obs = %% completer
Z2_obs = %% completer
Z_obs = [Z1_obs, Z2_obs];

close all
figure(1)
hold on;
plot(Z_obs(:,1), Z_obs(:,2), 'og');
plot(X(:,1), X(:,2), 'b');
legend( 'true state', 'observations');
title('trajectory (Z1,Z2) reconstructed from angular measures (theta1,theta2)')
hold off;
```

Faites varier les paramètres et observez les effets.

Exercice 2 (Filtrage)**.** Dans ce cas d'étude, seule l'équation d'observation est non linéaire.

1. Linéarisez le modèle d'évolution. Autrement dit, en appelant g la fonction telle que $Y_k = g(X_k) + \eta_k$, calculez la diférentielle de dg de la fonction g

2. La différence avec le filtre de Kalman 'standard' réside dans l'étape de mise à jour. En appelant (m^-_{k+1}, P^-_{k+1}) l'espérance et la variance de X_{k+1}, conditionnellement à $y_{1:k}$, on utilise l'approximation

$$\mathcal{L}\Big(\begin{pmatrix} X_{k+1} \\ Y_{k+1} \end{pmatrix} |y_{1:k} \Big) \approx \mathcal{L}\begin{pmatrix} m^-_{k+1} + \delta X^-_{k+1} \\ g(m^-_{k+1}) + dg_{m^-_{k+1}} \delta X^-_{k+1} + \eta_{k+1} \end{pmatrix}$$

où $\delta X^-_{k+1} \sim \mathcal{N}(0, P^-_{k+1})$ et $\eta_{k+1} \sim \mathcal{N}(0, R)$, *i.e.*,

$$\mathcal{L}\Big(\begin{pmatrix} X_{k+1} \\ Y_{k+1} \end{pmatrix} |y_{1:k} \Big) \approx \mathcal{N}\Big(\begin{pmatrix} m^-_{k+1} \\ g(m^-_{k+1}) \end{pmatrix}, \begin{pmatrix} P^-_{k+1} & P^-_{k+1} B^\top \\ B P^-_{k+1} & B P^-_{k+1} B^\top + R \end{pmatrix} \Big)$$

où $B = dg_{m^-_{k+1}}$.

3. En déduire une modification de l'étape de mise à jour du filtre de Kalman en s'inspirant du squelette de fonction suivant

```
function [mu, Pu] = ekf_update(y, mpred, Ppred, ypred, A, B, R)
 % Etape de mise a jour dans le filtre de Kalman etendu:
 % ARGUMENTS
 % mpred, Ppred: moyenne et variance de la loi predictive courante,
 % y: vecteur de taille q: nouvelle observation
 % ypred: l'observation predite: matrice colonne:
 % B: Matrice q*p definissant la dynamique linearisee
 %     de l'etat et du modele d'observation
 % R: Variance des bruits pour l'equation d'observation.de taille 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 = %% Completer (Gain de Kalman)
  mu = %% completer
%% !! difference avec kalman classique: ypred remplace B* xpred.
  Pu = %%
end
```

4. Déduire des questions précédentes un une fonction de filtrage en complétant le squelette suivant.

```
  function [M, Parray] = ekf_voiture(Y, m_0, P_0, L, sigma, tau, h)
 % Filtre de Kalman pour la voiture a capteurs directionnels
 % Filtre sequentiellement les observations Y
 % pour reconstruire le signal X
 %%%
 % ARGUMENTS:
 % Y: les observations : une matrice T * 2
 % m_0: vecteur colonne 2 * 1 : moyenne a priori de l'etat initial
 % P_0: matrice 2*2: variance a priori de l'instant initial
 % L : ecartement des deux capteurs sur l'axe 0y
 % sigma, tau: bruits sur l'etat et l'observation (cf pendule_step.m)
 % h: pas de discretisation
 %%%
```

```
% VALEUR:
% M: Matrice T * 4: les moyennes a posteriori
% Parray: Array de dimension 4*4*T les matrices de covariances
%         des lois de filtrages successives
 dim_state = 4;
 dim_obs = 2;
 T = size(Y,1);
 A = [ 1, 0, h, 0 ;
       0, 1, 0, h ;
       0, 0, 1, 0 ;
       0, 0, 0, 1 ] ;

 Q = sigma^2 * [ h^3/3 , 0 , h^2/2 , 0 ;
                0,    h^3/3, 0  , h^2/2 ;
                h^2/2, 0  , h   ,  0  ;
                0    , h^2/2  , 0   , h ] ;

 R = tau^2 * eye(dim_obs);
 P = P_0; m = reshape(m_0,dim_state,1) ;
 M = zeros(T,dim_state);
 Parray = zeros(dim_state,dim_state, T);
 for k=1:T
  %%%%%%%%%%%%%%%%%%
  %% prediction : comme dans Kalman (equation lineaire)
  %%%%%%%%%%%%%%%%%%
  mpred = %% completer
  Ppred = %% completer
  %%%%%%%%%%%%%%%%%%%
  %% mise a jour
  %%%%%%%%%%%%%%%%%%%
  %% Differentielle de g au point mpred
  r1 = mpred(1)^2 + mpred(2)^2;
  r2 = mpred(1)^2 + ( mpred(2) - L)^2;
  B = %% completer
  ypred = %% completer
  [mu, Pu] = ekf_update(Y(k,:), mpred , Ppred, ypred, A, ...
                B, R);
  %% reaffectation de l'etat courant
  m = mu; P = Pu;
         %% stockage
  M(k,: ) = mu';
  Parray(:,:,k) = P;
 end
end
```

5. Visualisation : Testez votre filtre en vous inspirant du script suivant (après avoir simulé [X, Y] comme dans l'exercice 1) :

```
% Priors sur l'etat initial
m_0 = [60,60,0,0];
P_0 = 100 * eye(4);

% filtrage
[M, Parray] = ekf_voiture(Y, m_0, P_0, L, sigma, tau, h=0.01);

R_obs = %% completer comme dans l'exercice 1
```

```
Z1_obs = %% completer comme dans l'exercice 1
Z2_obs = %% completer comme dans l'exercice 1
Z_obs = [Z1_obs, Z2_obs];

close all
figure(2)
hold on;
plot(M(:,1), M(:,2),'r');
plot(X(:,1), X(:,2), 'b');
plot(Z_obs(:,1), Z_obs(:,2), 'og');
legend('filtered', 'true state', 'observations');
title('trajectory (Z1,Z2) reconstructed from angular measures
(theta1,theta2) and from the kalman filter')
hold off;
```

6. pour $i = 1, 2$, tracez sur le même graphique l'évolution de l'état $Z_{k,i}$ $(kleT)$, avec les observations associées (reconstruites comme à l'exercice 1), et la moyenne de la loi de filtrage $(m_{k,i}, k \leq T)$. Indiquez des intervalles de crédibilité a posteriori en vous inspirant du TP précédent (Kalman). Faites de même avec l'évolution de la vitesse $V_{k,i}, k \leq T$ (pour laquelle vous n'avez pas d'observations).