TP Master SAM – 06/11/2020

« Localisation, cartographie, planification, fusion de données »

Le filtrage de Kalman

Développement, Utilisation et Compréhension

I) Développement du filtrage de Kalman

L'objectif principal de ce TP est de réaliser le développement d'un filtre de Kalman étendu (avec un modèle non linéaire).

Dans un premier temps, vous devrez prendre en main un ensemble de codes sources sous Matlab. Dans ces codes, vous trouverez toutes les fonctions permettant de mettre en œuvre le filtre de Kalman.

Cet ensemble de codes se décompose en 5 groupes de fonctions et de données :

1. Les données brutes et réelles provenant de capteurs embarqués sur un véhicule (capteurs proprioceptifs) ainsi que de données provenant d'un GPS. Toutes ces données sont datées.
   a. Répertoire SensorData

2. Les données de référence de la piste sur laquelle la collecte des données a été réalisée. Ces pistes se situent à Satory.
   a. Répertoire TracksData

3. Les fonctions concernant la mise en forme et le formatage des données provenant des capteurs embarqués pour leur intégration dans un ensemble de structures dédiées facilement utilisable.
   a. Chargement_BraquageRoue.m
   b. Chargement_Compass_KVH.m
   c. Chargement_Gyro_KVH.m
   d. Chargement_INS_VG400.m
   e. Chargement_Odometre.m
   f. ChargementGPS_A12.m, ChargementGPS_AG132.m et ChargementGPS_Sagitta.m
   g. ChargementPistesReferences.m
   h. ChargementDonneesFUDOLO.m et ChargementDonneesFUDOLOSynchro5Hz.m

4. Les fonctions correspondant au filtre de Kalman.

a. InitBruitGPS.m

b. InitialisationFiltres.m

c. InitialisationPosition.m

d. EstimationKalmanNL.m

```
function [X_E,P_E,Sk]=EstimationKalmanNL(P_P,Hk,X_P,Y_GPS,Q_GPS);

% P_P: Matrice de varcov sur prediction
% Hk: Matrice de eqution de mesure, sélectionne variable de l'état qui sont
estimées (mise à jour)
% X_P: Prédiction utilisé
% Y_GPS: données GPS, (X,Y)
% Q_GPS: bruit de GPS

    %==================================
    % ETAPE D'ESTIMATION : instant k/k
    %==================================

   % Mettre ici les équations correspondant au cours
```

e. ModeleEvolutionGPSNL.m

```
%*******************************************************************************
% ModeleEvolutionGPSNL
% Module de calcul des différentes matrices et structures utilisées par l'EKF
% (c) LIVIC-IFSTTAR
%
% Fichier           : $RCSfile: ModeleEvolutionGPSNL.m,v $
% Auteur            : Dominique Gruyer
% Version           : $Revision: 0.00 $

% Auteurs :
%  Dominique Gruyer:      1) Développement du code principal
%                         2) Calcul des étapes suivantes:
%                            matrice jacobienne du modele d'évolution: Fk = dfk/dxk
%                            matrice jacobienne du modele d'évolution: Bk = dfk/duk
%                            matrice de mesure: Hk
%                            matrice de bruit d'état et de mesure
%*******************************************************************************
function [Fk,Hk,Bk,Q_systeme,Q_GPS] =
ModeleEvolutionGPSNL(var_S,var_teta_c,teta_c,VarSysteme,ModeGPS,Valide,SigmaA,Sigma
B,Phi,Psi);

   %================================================================
   % DEFINITION DU MODELE D'EVOLUTION ET DE LA PREMIERE PREDICTION
   %================================================================
    % calcul de la première position prédite du véhicule
   %=======================================================
   R = 0.3;                        % rayon de la roue
   E = 1;                          % longueur de l'essieu
   pas_codeur=.1954;

    % matrice Jacobienne du modèle d'évolution: Fk = dfk/dxk
    %=======================================================

   Mettre ici les équations correspondant au cours

    % matrice Jacobienne du modèle d'évolution: Bk = dfk/duk
    %=======================================================

   Mettre ici les équations correspondant au cours

    % matrice de mesure: Hk
    %=======================================================

   Mettre ici les équations correspondant au cours
```

```
%=========================================================
% CALCUL DE LA MATRICE DE BRUIT SUR L'ENTREE DU SYSTEME
%=========================================================

Mettre ici les équations correspondant au cours

%=================================================================
% Construction de la matrice de bruit du GPS
% Cette matrice est construite à partir des sigmas a et b et de la
% rotation fournie par le GPS dans le cas où l'indicateur de validité
% d'une mesure est à 1
%=================================================================
%  Mode GPS, 1 gps naturel, 2 différentiel EGNOS, 0 masquage
%  Valide, 0 et 1
%=================================================================
```

f. ModeleEvolutionNL.m

```
%*****************************************************************************************
% ModeleEvolutionNL
% Module de calcul des différentes matrices et structures utilisées par l'EKF
%
% Fichier           : $RCSfile: ModeleEvolutionNL.m,v $
% Auteur            : Dominique Gruyer

%
% Auteurs :
%  Dominique Gruyer:  1) Développement du code principal
%                     2) Calcul des étapes suivantes:
%                        matrice Jacobienne du modèle d'évolution: Fk = dfk/dxk
%                        matrice Jacobienne du modèle d'évolution: Bk = dfk/duk
%                        matrice de mesure: Hk
%                        matrice de bruit d'état et de mesure
%*****************************************************************************************
**/
function
[Fk,Hk,Bk,Q_systeme]=ModeleEvolutionNL(var_S,var_teta_c,teta_c,VarSysteme,Psi);

   %=================================================================
   % DEFINITION DU MODELE D'EVOLUTION ET DE LA PREMIERE PREDICTION
   %=================================================================
        % calcul de la première position prédite du véhicule
   %=========================================================
   R = 0.3;                          % rayon de la roue
   E = 1;                            % longueur de l'essieu
   pas_codeur=.1954;

        % matrice Jacobienne du modèle d'évolution: Fk = dfk/dxk
        %=========================================================

        Mettre ici les équations correspondant au cours

        % matrice Jacobienne du modèle d'évolution: Bk = dfk/duk
        %=========================================================

        Mettre ici les équations correspondant au cours

        % matrice de mesure: Hk
        %=========================================================

        Mettre ici les équations correspondant au cours

   %=========================================================
   % CALCUL DE LA MATRICE DE BRUIT SUR L'ENTREE DU SYSTEME
   %=========================================================

        Mettre ici les équations correspondant au cours
```

g. PredictionKalmanNL

```
function [X_P,P_P]=PredictionKalmanNL(var_S,var_teta,Fk,X_E,P_E,Q_systeme,AngleRoue);

% Calcul de la prédiction X(k/k-1)
%=================================================

      Mettre ici les équations correspondant au cours

% calcul de la variance de la prediction P(k/k-1)
%=================================================

   Mettre ici les équations correspondant au cours
```

Seules les fonctions **d,e,f** et **g** sont à mettre à jour afin d'obtenir un filtre fonctionnel.

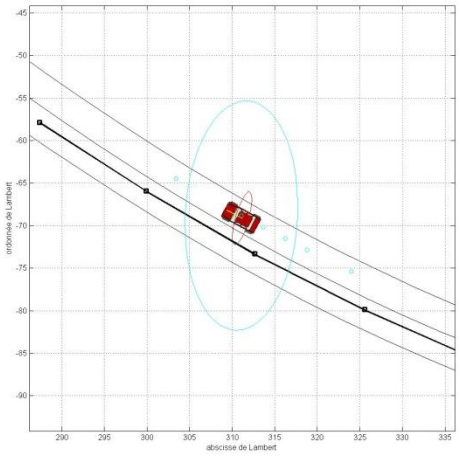
Le modèle d'évolution utilisé est un modèle de type « *bicyclette* » :

```
    X_P  =   [  (X_E(1) + var_S*cos(X_E(3) + (var_teta/2))*cos(AngleRoue));
                (X_E(2) + var_S*sin(X_E(3) + (var_teta/2))*cos(AngleRoue));
                (X_E(3) + var_teta)];
```

5. Les fonctions utilitaires et les fonctions d'affichage
   a. AffichageVoitureEKF.m
   b. ModeleVoiture.m
   c. DisplayStateVectorResults.m
   d. DrawEllipse.m
   e. ChangeAngleInterval.m
   f. ConcerterLL2XY.m
   g. UpdateRecordStructures

II) Mise en œuvre du filtre sur données réelles

Une fois que les fonctions du filtre sont mises à jour, vous pouvez les utiliser sur le jeu de données fournit dans ce TP. Faite fonctionner ce filtre dans diverses configurations : avec et sans correction des données GPS afin de voir la dérive du filtre au court du temps.



III) Compréhension du fonctionnement

Cette dernière étape consiste à évaluer votre compréhension du fonctionnement du filtre de Kalman étendu et à pouvoir exposer clairement ses avantages, ses inconvénients ainsi les limites de fonctionnement. Dans cette partie vous devrez présenter clairement les données utilisées (phase d'initialisation, bruits sur les données provenant des capteurs, bruit du système …) et donner les résultats (courbes) représentant les différents fonctionnements.

Dans cette partie faite varier les grandeurs suivantes et interpréter les résultats :

- Qualité de l'initialisation (vecteur d'état et matrices de variance-covariance)
- Impact de la matrice de bruit du système
- Impact de la matrice de bruit de mesure