TP - Filtrage De Kalman : application à la poursuite de cible mobile Telecom SudParis - MAT4501

10/12/2019

Dans ce TP, on s'intéresse au problème de poursuite d'une cible mobile à partir de mesures bruitées. **Un compte-rendu rédigé en LaTeXest à rendre pour le vendredi 20 décembre, 23h59m59s.**

1 Scénario de poursuite

1.1 Modélisation du Problème

On désire poursuivre une cible, en supposant que la dynamique de celle-ci est "à peu près" linéaire et sa vitesse est constante. En réalité, ce n'est jamais le cas car la dynamique de la cible est inconnue : à tout moment, elle peut tourner ou accélérer. C'est pour cette raison qu'un modèle probabiliste se prête bien à la modélisation de la dynamique de la cible. Par ailleurs, nous disposons d'un capteur qui relève des informations sur la cible (signal observé) avec une période T_e, ces informations étant une fonction des états cachés de la cible. Le capteur n'est pas parfait ce qui induit des erreurs de mesure. Le filtre de Kalman cherche à annuler le caractère aléatoire dû aux erreurs de mesure et à la dynamique inconnue de la cible.

Nous modélisons le problème sous forme de 2 équations, appelées **équation d'état** (qui porte sur la dynamique de la cible) et **équation d'observation** (qui porte sur les mesures du capteur).
On s'intéresse dans un premier temps à la position dans le plan de la cible et à sa vitesse. Une réalisation du vecteur d'état à l'instant $t = k$ s'écrit donc $x_k = [p_x, \dot{p}_x, p_y, \dot{p}_y]^T_{t=k}$, où p_x est l'abscisse de la cible et $\dot{p_x}$ la vitesse en abscisse de la cible (idem pour p_y, en ordonné).
Le capteur utilisé mesure la position en abscisse et en ordonné de la cible. Une observation est donc un vecteur colonne de dimension 2. Le problème peut s'écrire :

$$X_k = \underbrace{\begin{pmatrix} 1 & T_e & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T_e \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\mathbf{F}_k} X_{k-1} + U_k, \tag{1}$$

$$Y_k = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{\mathbf{H}_k} X_k + V_k, \tag{2}$$

$$X_0 \sim \mathcal{N}(\mathbf{m}_{0|0}, \mathbf{P}_{0|0}), \tag{3}$$

avec $U_k \sim \mathcal{N}(\mathbf{0}_{4\times1}, \mathbf{Q})$ (on a $\dim(U_k) = 4 \times 1$ et $\dim(\mathbf{Q}) = 4 \times 4$), $V_k \sim \mathcal{N}(\mathbf{0}_{2\times1}, \mathbf{R})$ (on a $\dim(V_k) = 2 \times 1$ et $\dim(\mathbf{R}) = 2 \times 2$). Dans les problématiques de poursuite, on choisit généralement $\mathbf{Q}$ sous la forme $\mathbf{Q} = \sigma_Q^2 \begin{pmatrix} \frac{T_e^3}{3} & \frac{T_e^2}{2} & 0 & 0 \\ \frac{T_e^2}{2} & T_e & 0 & 0 \\ 0 & 0 & \frac{T_e^3}{3} & \frac{T_e^2}{2} \\ 0 & 0 & \frac{T_e^2}{2} & T_e \end{pmatrix}$ et $\mathbf{R} = \begin{pmatrix} \sigma_{p_x}^2 & 0 \\ 0 & \sigma_{p_y}^2 \end{pmatrix}$. On suppose que la suite de vecteurs $[U_k, V_k]^T$ est indépendante, indépendante de X_0 et que U_k et V_i sont indépendants quelques soient k et i ($\Rightarrow \text{Covar}(U_k, V_i) = 0$).

1.2 Interprétation

L'équation (1) assure par exemple que la position de la cible à l'instant $k+1$ est la position de la cible à l'instant k augmentée de sa vitesse, plus un terme de bruit de moyenne nulle. L'équation (2) assure que les informations observées sont les positions de la cible à l'instant k plus un terme d'erreur, de moyenne nulle lui aussi. Vérifiez rapidement cette interprétation. Notez qu'ici les matrices $\mathbf{F}$, $\mathbf{H}$, $\mathbf{Q}$, et $\mathbf{R}$ ne dépendent pas de k.

1.3 Préliminaires

Nous avons vu en cours que le modèle (1)-(2) constituait une chaîne de Markov caché dont les transitions sont données par $f_{k|k-1}(x_k|x_{k-1}) = \mathcal{N}(x_k; \mathbf{F}x_{k-1}; \mathbf{Q})$ et $g_k(y_k|x_k) = \mathcal{N}(y_k; \mathbf{H}x_k; \mathbf{R})$. On rappelle que l'estimateur optimal à l'instant k au sens de l'erreur quadratique moyenne, connaissant les k premières observations $y_{0:k}$ est donné par

$$\hat{x}_{k|k} = m_{k|k} = \mathrm{E}[X_k|y_{0:k}]; \tag{4}$$

et on note

$$P_{k|k} = \mathrm{Covar}[X_k|y_{0:k}] \tag{5}$$

1. Rappeler comment dans le modèle (1)-(2), on passe de $\hat{x}_{k|k}$ et $P_{k|k}$ à $\hat{x}_{k+1|k+1}$ et $P_{k+1|k+1}$. Nous allons maintenant implémenter ces équations.

2 But du TP

Le but du TP est de réaliser les taches suivantes :

— Initialiser les paramètres du modèle (matrices $\mathbf{F}$, $\mathbf{Q}$, $\mathbf{H}$, $\mathbf{R}$, longueur du scénario, ...).

— Générer une trajectoire aléatoire à partir de l'équation (1), qui servira de vérité terrain.

— Générer un jeu d'observations de la trajectoire crée, à partir de l'équation (2).

— Implémenter de façon récursive le filtre de Kalman, afin de retrouver la trajectoire cachée à partir de la trajectoire observée.

— Tracer la vraie trajectoire dans le plan, la trajectoire relevée par le capteur et la trajectoire estimée par le filtre de Kalman.

Vous serez guidé sur les programmes à écrire.

Chaque programme ne nécessite pas plus de 10 lignes de commande et peut être écrit dans le langage de votre choix (Python,Matlab,...).

3 Partie Pratique 1 : Poursuite en coordonnées cartésiennes

3.1 Implémentation du Filtre de Kalman sur données synthétiques

1. Créer un répertoire TP_FDK, ouvrir MATLAB ou Python et créer un fichier « principal.m »dans le dossier crée.

2. Initialiser les variables suivantes : $T\_e = 1$ (période du capteur), $T = 100$ (longueur du scénario), $sigma\_Q = 1$, $sigma\_px = 30$, $sigma\_py = 30$, puis créer les matrices $\mathbf{F}$, $\mathbf{Q}$, $\mathbf{H}$ et $\mathbf{R}$ dont les valeurs sont données dans la partie 1.1. Initialiser $x\_init = [3, 40, -4, 20]^T$ qui servira à initialiser notre trajectoire, puis $x\_kalm = x\_init$ (qui correspond à $\hat{x}_{0|0}$) et $P\_kalm = \mathbf{I}_{4\times 4}$ (qui correspond à $P_{0|0}$) qui servent à initialiser la moyenne et la matrice de covariance du filtre de Kalman. On suppose donc ici qu'on connaît à peu près l'état initial de la cible puisque $x\_kalm = x\_init$.

3. Ecrire la fonction $vecteur\_x = creer\_trajectoire(\mathbf{F}, \mathbf{Q}, x\_init, T)$ qui prend en entrée les matrices $\mathbf{F}$, $\mathbf{Q}$, x_{init} et T et qui retourne une matrice $vecteur\_x$ de dimension $4 \times T$, à partir de l'équation (1). On obtient donc une trajectoire aléatoire. Une fois créée, appeler cette fonction dans le programme principal à la suite de l'initialisation de toute les variables.

4. Ecrire la fonction $vecteur\_y = creer\_observations(\mathbf{H}, \mathbf{R}, vecteur\_x, T)$ qui prend en entrée les matrices $\mathbf{H}$ et $\mathbf{R}$, une matrice $vecteur\_x$ de dimension $4 \times T$, la durée du scénario T et qui retourne des observations bruitées de la trajectoire x créée précédemment. On obtient donc une matrice $vecteur\_y$ de dimension $2 \times T$. Appeler cette fonction dans le programme principal.

5. Tracer sur une même figure la vraie trajectoire et la trajectoire observée, d'une couleur différente. Cette étape vous permettra de vérifier que vos fonctions sont écrites correctement.

6. Ecrire la fonction $[x\_kalm\_k, P\_kalm\_k] = filtre\_de\_kalman(\mathbf{F}, \mathbf{Q}, \mathbf{H}, \mathbf{R}, y\_k, x\_kalm\_prec, P\_kalm\_prec)$ qui retourne le vecteur $x\_kalm\_k = \hat{x}_{k|k}$ et la matrice $P\_kalm\_k = P_{k|k}$ à l'instant k à partir des matrices **F**, **Q**, **H**, **R**, de la nouvelle observation $y\_k = vecteur\_y(:,k)$ à l'instant k (vecteur 2×1), du vecteur $x\_kalm\_prec = \hat{x}_{k-1|k-1}$ et de la matrice $P\_kalm\_prec = P_{k-1|k-1}$ estimées à l'instant précédent (bien distinguer la partie prédiction et mise-à-jour).

7. Ecrire des lignes de commande à la suite, dans le programme principal permettant d'estimer les états cachés de la cible à chaque instant à partir du jeu d'observations dont on dispose, et de la fonction $filtre\_de\_kalman$. Pensez à mémoriser les états estimés à chaque instant dans une matrice $x\_est(:,k)$

8. Calculer l'erreur quadratique à chaque instant définie par $err\_quadra(k) = (x_k - \hat{x}_{k|k})^T(x_k - \hat{x}_{k|k})$, puis moyenner sur le temps afin d'obtenir une unique valeur, $erreur\_moyenne = \frac{1}{T}\sum_{k=1}^{T}\sqrt{err\_quadra(k)}$.

9. Tracer sur une première figure la position vraie, estimée et observée en abscisse en fonction du temps ; sur une seconde figure, la position vraie, estimée et observée en ordonnée en fonction du temps ; et enfin la vraie trajectoire, la trajectoire observée et la trajectoire estimée par le filtre de Kalman.

10. Faites varier le bruit de processus σ_Q puis les bruits de mesure σ_{px} et σ_{py}. Commentaires sur l'erreur moyenne ? Ne pas hésiter à augmenter σ_{px} et σ_{py} pour observer la qualité de la restitution, à partir d'un signal très bruité.

3.2 Application

Lors d'un meeting aérien, nous avons filmé la trajectoire d'un avion de ligne, puis celle d'un avion de voltige. On souhaite extraire la trajectoire de ces avions sur ordinateur, de façon automatique. Dans un premier temps, on a appliqué un algorithme de détection d'objets sur chacune des séquences vidéos, basé sur la couleur des avions, qui retourne des observations bruités de la trajectoire de chaque avion : $vecteur\_y\_avion\_ligne$ et $vecteur\_y\_avion\_voltige$ téléchargeable à l'adresse suivante : `http://www-public.it-sudparis.eu/~petetin/cours.html`, Filtrage de Kalman.

Contrairement au cas précédent, il se peut qu'à un instant donné k, l'algorithme de détection n'ait pas détecté l'avion (c'est le cas si l'avion passe derrière un nuage par exemple). Dans ce cas, le vecteur d'observation à l'instant k retourne $vecteur\_y\_avion\_ligne(:,k) = [NaN, NaN]$ (NaN signifie Not a Number).

1. Comment modifier le filtre pour prendre en compte cette absence de détection ? Pour répondre à la question, on supposera qu'il y'a non-détection à l'instant k et on cherchera le "meilleur" estimateur de x_k, à partir de $y_{0:k-1}$.

2. Modifier alors la fonction $filtre\_de\_kalman$, charger les vecteurs d'observations ($vecteur\_y\_avion\_ligne$) et $vecteur\_y\_avion\_voltige$ dans le programme principal et estimer les deux trajectoires. Comparer les trajectoires estimées avec les vraies trajectoires ($vecteur\_x\_avion\_ligne$) et $vecteur\_x\_avion\_voltige$. Commenter.

3. Constatez vous une différence de performance entre l'estimation de la trajectoire de l'avion de ligne et celle de l'avion de voltige ? Si oui, expliquez pourquoi.

4. Prendre un cachet d'aspirine.

4 Partie Pratique 2 : Poursuite Angle-Distance

Dans certaines applications, comme en radar par exemple, nous n'observons pas la position en abscisse et en ordonnée de la cible mais l'angle et la distance de la cible par rapport au capteur (coordonnées polaires). On suppose ici (comme ce qu'on a fait implicitement précédemment) que notre capteur est situé à l'origine du repère $(0, 0)$. Créer un fichier *principal_2.m* et copier-coller les lignes de commandes du premier fichier dans le second. Nous modifierons au fur et à mesure les commandes concernées.

1. Exprimer l'angle et la distance de la cible en fonction de ses coordonnées cartésiennes p_x et p_y. On se souviendra des formules de trigonométrie apprises en classe de 3ème.

2. Remplacer alors dans l'équation (2) le terme $\mathbf{H}x_k$ par un vecteur de dimension 2×1 approprié à cette nouvelle modélisation. Que devient la loi $g_k(y_k|x_k)$?

3. La matrice $\mathbf{R}$ a toujours la même forme mais nous changeons les valeurs des bruits de mesure, i.e $\mathbf{R} = \begin{pmatrix} \sigma_{angle}^2 & 0 \\ 0 & \sigma_{dist}^2 \end{pmatrix}$ avec $\sigma_{angle} = pi/180$ et $\sigma_{dist} = 10$.
   Créer une nouvelle fonction $vecteur\_y = creer\_observations\_radar(\mathbf{R}, vecteur\_x, T)$ permettant de générer un jeu d'observation radar bruité à partir d'une trajectoire $vecteur\_x$, puis placer la dans le programme principal 2.
   (On reprendra la fonction $vecteur\_y = creer\_observations(\mathbf{H}, \mathbf{R}, vecteur\_x, T)$ en remplaçant le produit $\mathbf{H} \times vecteur\_x(:, k)$ par un terme adéquat).

4. Peut-on utiliser le filtre de Kalman pour estimer le vecteur d'état de la cible à chaque instant ? Pourquoi ?

5. A l'instant k, l'équation d'observation s'écrit

$$y_k \quad = \quad \begin{pmatrix} f(x_k) \\ g(x_k) \end{pmatrix} + v_k, \tag{6}$$

   où $f, g : \mathbb{R}^4 \to \mathbb{R}$, sont des fonctions déterminées à la question 2 de cette partie. Nous allons chercher à nous ramener à un modèle semblable à l'équation (2).
   - Commencer par linéariser les fonctions f et g autour du vecteur $\hat{x}_{k|k-1} = \mathbf{F}\hat{x}_{k-1|x-1}$, et justifier le choix de $\hat{x}_{k|k-1}$ pour cette linéarisation.
     On rappelle la formule de Taylor au première ordre autour du vecteur a pour une fonction de plusieurs variables :

$$f(x) \quad \approx \quad f(a) + \nabla f(a)(x - a), \tag{7}$$

     avec, si $x = [x_1, x_2, x_3, x_4]$, $\nabla f(a) = [\frac{\partial f}{\partial x_1}(a), \cdots, \frac{\partial f}{\partial x_4}(a)]$.
   - Réécrire l'équation d'observation (2) sous la forme

$$y_k \quad = \quad \begin{pmatrix} f(\hat{x}_{k|k-1}) \\ g(\hat{x}_{k|k-1}) \end{pmatrix} + \mathbf{H}(\hat{x}_{k|k-1})(x_k - \hat{x}_{k|k-1}) + v_k, \tag{8}$$

     où $\mathbf{H}(\hat{x}_{k|k-1})$ est une matrice 2×4 obtenue grâce à la linéarisation précédente et qui dépend de $\hat{x}_{k|k-1}$

6. Ecrire alors une nouvelle fonction (**voir indication**)
   $filtre\_de\_kalman\_radar(\mathbf{F}, \mathbf{Q}, \mathbf{R}, y\_k, x\_kalm\_prec, P\_kalm\_prec)$ (qui cette fois ci ne prend plus la matrice $\mathbf{H}$ en entrée, il faut la redéfinir à chaque fois dans cette même fonction), qui permet d'estimer les états cachés de la cible à l'instant k.

7. Tracer la vraie trajectoire et la trajectoire estimée. Commenter, en faisant varier les paramètres de bruit. Vous venez en réalité d'implémenter l'EKF (Extended Kalman Filter - Filtre de Kalman étendu) qui permet d'utiliser le filtre de Kalman pour n'importe quel modèle. Quels peuvent être les inconvénients de l'EKF ? D'autres solutions existent pour résoudre ce problème comme l'UKF (Unscented Kalman Filter) ou encore le filtrage particulaire, qui sont cependant plus gourmands en terme de coûts de calcul.

Indication pour la question 6 : du fait de l'équation (8), il n'est pas possible d'appliquer directement le filtre de Kalman (nous ne sommes pas dans les conditions de l'équation (2)). On pose, pour se ramener au modèle classique

$$y'_k = y_k - \begin{pmatrix} f(\hat{x}_{k|k-1}) \\ g(\hat{x}_{k|k-1}) \end{pmatrix} + \mathbf{H}\hat{x}_{k|k-1} \text{ (on translate } y_k). \tag{9}$$

Toutes les équations vues en cours restent inchangées, sauf l'estimation de l'innovation, où l'on a :

$$\begin{aligned} \tilde{y}_k &= y'_k - \mathbf{H}\hat{x}_{k|k-1} \\ &= y_k - \begin{pmatrix} f(\hat{x}_{k|k-1}) \\ g(\hat{x}_{k|k-1}) \end{pmatrix} \end{aligned} \tag{10}$$

et non plus

$$\tilde{y}_k = y_k - \mathbf{H}\hat{x}_{k|k-1}.$$

5 Une extension à la poursuite de plusieurs cibles

On discutera à la fin du TP à l'adaptation des méthodes précédentes pour la problématique de pistage multi-cibles dans le cadre d'un capteur infra-rouge.

1. À travers les séquences vidéos qui seront projetées, quels sont les principaux obstacles à l'utilisation directe du filtre de Kalman pour la poursuite multi-cibles ?

2. Que faudrait il faire pour pouvoir utiliser simultanément plusieurs filtres de Kalman, afin de pister chaque cible ?