

# Introduction aux processus stochastiques

## Travail Matlab : Filtre de Kalman

### Consignes de style pour le travail

Vous pouvez remettre un document rédigé dans le style d'un bref rapport de laboratoire.

### Suivi d'un véhicule par mesures de distance à des balises.

#### But de l'exercice.

On souhaite pouvoir estimer en permanence la position d'un véhicule en mouvement, sur base uniquement de mesures de distance entre la position (inconnue)  $\mathbf{p}(t) \in \mathbb{R}^2$  du véhicule aux instants  $t = 0, 1, \dots$ , et la position (fixe et connue) de  $m$  balises, placées en  $\mathbf{s}_1 \in \mathbb{R}^2, \dots, \mathbf{s}_m \in \mathbb{R}^2$ . Ainsi, à un instant  $t$  on disposera d'un nouveau vecteur de mesure  $\mathbf{y}(t) \in \mathbb{R}^m$  dont les éléments

$$y_i(t) = \|\mathbf{s}_i - \mathbf{p}(t)\| + v_i(t) = \sqrt{[\mathbf{s}_i - \mathbf{p}(t)]^T [\mathbf{s}_i - \mathbf{p}(t)]} + v_i(t) \quad (1)$$

correspondent aux mesures de distance à la balise  $i$ , avec  $v_i(t)$  un bruit de mesure gaussien.

Le but de l'exercice est de fournir au temps  $t$  une estimation de la position du véhicule au temps  $t$

$$\hat{\mathbf{p}}(t) \simeq \mathbb{E}\{\mathbf{p}(t) \mid \mathbf{y}(0), \mathbf{y}(1), \dots, \mathbf{y}(t)\}, \quad (2)$$

en se servant de l'estimation précédente  $\hat{\mathbf{p}}(t-1)$  et du nouveau vecteur de mesures  $\mathbf{y}(t)$ .

Idéalement, pour minimiser l'espérance de l'erreur quadratique d'estimation de  $\mathbf{p}(t)$  par  $\hat{\mathbf{p}}(t)$ , c'est-à-dire  $\mathbb{E}\{\|\hat{\mathbf{p}}(t) - \mathbf{p}(t)\|^2\}$ , l'égalité dans (2) devrait être stricte. Cependant, l'espérance dans (2) sera calculée de façon approchée, en linéarisant la relation (1) afin d'appliquer la technique du filtre de Kalman.

#### Spécification du système.

On place  $m = 8$  balises en

$$[\mathbf{s}_1 \mid \mathbf{s}_2 \mid \dots \mid \mathbf{s}_8] = \begin{bmatrix} -3 & 0 & 3 & -1.5 & 1.5 & -3 & 0 & 3 \\ 3 & 3 & 3 & 0 & 0 & -3 & -3 & -3 \end{bmatrix} \triangleq \mathbf{S} \in \mathbb{R}^{2 \times 8}. \quad (3)$$

Les bruits de mesure sont i.i.d. (indépendants et identiquement distribués) gaussiens :

$$\mathbf{v}(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_v) \quad \mathbf{\Sigma}_v = 0.1 \mathbf{I}_m, \quad (4)$$

avec  $\mathbf{v}(t) \in \mathbb{R}^m$  le vecteur formé des  $v_i(t)$ , et  $\mathbf{I}_m$  la matrice identité de dimension  $m$ .

La dynamique du véhicule suit

$$\mathbf{p}(t+1) = \mathbf{p}(t) + 0.1\mathbf{u}(t), \quad \mathbf{u}(t+1) = \begin{bmatrix} 0.85 & 0.15 \\ -0.1 & 0.85 \end{bmatrix} \mathbf{u}(t) + \mathbf{w}(t). \quad (5)$$

Le vecteur  $\mathbf{u}(t) \in \mathbb{R}^2$  s'interprète comme la vitesse du véhicule à l'instant  $t$ . La vitesse est contrôlée de manière stochastique par des perturbations  $\mathbf{w}(t) \in \mathbb{R}^2$  i.i.d. gaussiennes :

$$\mathbf{w}(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_w) \quad \mathbf{\Sigma}_w = \mathbf{I}_2. \quad (6)$$

On rassemble dans  $\mathbf{x}(t) = [\mathbf{p}(t)^T \ \mathbf{u}(t)^T]^T \in \mathbb{R}^4$  les variables d'état. L'état initial est

$$\mathbf{x}(0) \sim \mathcal{N}(\bar{\mathbf{x}}_0, \mathbf{\Sigma}_{x_0}) \quad \bar{\mathbf{x}}_0 = \mathbf{0} \quad \mathbf{\Sigma}_{x_0} = \mathbf{I}_4. \quad (7)$$

## Filtre de Kalman.

Pour un système de la forme

$$\begin{aligned} x(t+1) &= A_t x(t) + B_t w(t) & w(t) &\sim \mathcal{N}(0, \Sigma_w) & x(0) &\sim \mathcal{N}(\bar{x}, \Sigma_{x_0}) \\ y(t) &= C_t x(t) + d_t + v(t) & v(t) &\sim \mathcal{N}(0, \Sigma_v) \end{aligned}$$

avec  $x(0)$ , les  $w(t)$  et les  $v(t)$  mutuellement indépendants, on peut calculer

$$\hat{x}(t) = \mathbb{E}\{x(t)|y(0), y(1), \dots, y(t)\}$$

en utilisant  $\hat{x}(t-1)$  et  $y(t)$ .

On définit tout d'abord

$$\begin{aligned} \hat{x}(t|t-1) &= \mathbb{E}\{x(t)|y(0), y(1), \dots, y(t-1)\} && \text{Prédiction de l'état} \\ \Sigma_{t|t-1} &= \mathbb{E}\{[x(t) - \hat{x}(t|t-1)][x(t) - \hat{x}(t|t-1)]^T\} && \text{Covariance de la prédiction} \\ \hat{x}(t|t) &= \mathbb{E}\{x(t)|y(0), y(1), \dots, y(t)\} \triangleq \hat{x}(t) && \text{Estimation de l'état} \\ \Sigma_{t|t} &= \mathbb{E}\{[x(t) - \hat{x}(t|t)][x(t) - \hat{x}(t|t)]^T\} && \text{Covariance de l'estimation.} \end{aligned}$$

En  $t=0$  on pose  $\hat{x}(0|-1) = \bar{x}$  et  $\Sigma_{0|-1} = \Sigma_{x_0}$ .

En supposant qu'on dispose de  $\hat{x}(t|t-1)$  et  $\Sigma_{t|t-1}$ , on calcule pour l'estimation

$$\begin{aligned} \hat{x}(t|t) &= \hat{x}(t|t-1) + \Sigma_{t|t-1} C_t^T [C_t \Sigma_{t|t-1} C_t^T + \Sigma_v]^{-1} [y(t) - (C_t \hat{x}(t|t-1) + d_t)] , \\ \Sigma_{t|t} &= \Sigma_{t|t-1} - \Sigma_{t|t-1} C_t^T [C_t \Sigma_{t|t-1} C_t^T + \Sigma_v]^{-1} C_t \Sigma_{t|t-1} , \end{aligned}$$

puis on calcule pour la prédiction

$$\hat{x}(t+1|t) = A_t \hat{x}(t|t) , \quad \Sigma_{t+1|t} = A_t \Sigma_{t|t} A_t^T + B_t \Sigma_w B_t^T .$$

## Linéarisation de l'équation de mesure.

L'équation (1) est de la forme  $y(t) = g(x(t)) + v(t)$ , avec  $g(\cdot)$  non linéaire. Pour se ramener à une forme linéaire et appliquer les formules d'estimation et de prédiction du filtre de Kalman, on peut linéariser  $g(\cdot)$  autour du point  $\hat{x}(t|t-1)$ .

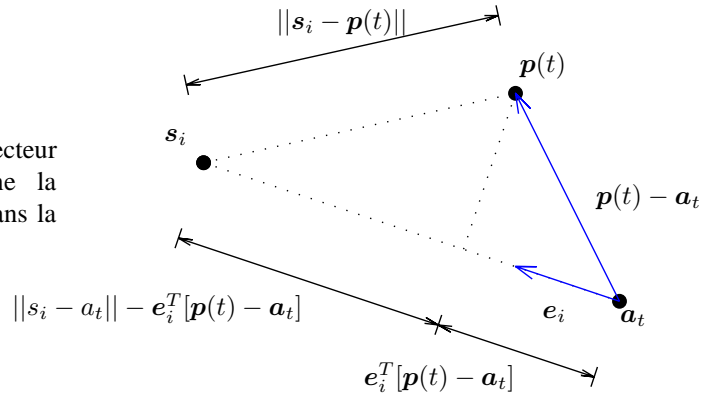
La linéarisation de (1) autour d'un point  $\mathbf{a}_t \in \mathbb{R}^2$  donne

$$y_i(t) \simeq \|\mathbf{s}_i - \mathbf{a}_t\| - \frac{(\mathbf{s}_i - \mathbf{a}_t)^T}{\|\mathbf{s}_i - \mathbf{a}_t\|} (\mathbf{p}(t) - \mathbf{a}_t) + v_i(t) . \quad (8)$$

$\mathbf{a}_t$  correspond à la prédiction de la position, i.e. aux 2 premiers éléments du vecteur  $\hat{\mathbf{x}}(t|t-1)$ .

L'expression (8) a l'interprétation géométrique ci-contre.

$\mathbf{e}_i \triangleq (\mathbf{s}_i - \mathbf{a}_t) / \|\mathbf{s}_i - \mathbf{a}_t\|$  est un vecteur unitaire, et  $\mathbf{e}_i^T [\mathbf{p}(t) - \mathbf{a}_t]$  donne la projection du vecteur  $[\mathbf{p}(t) - \mathbf{a}_t]$  dans la direction  $\mathbf{e}_i$ .



## Questions

1. Définir les matrices  $A$  et  $B$  qui permettent de mettre (5) sous la forme

$$\mathbf{x}(t+1) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{w}(t), \quad \text{avec } \mathbf{x}(t) \triangleq \begin{bmatrix} \mathbf{p}(t) \\ \mathbf{u}(t) \end{bmatrix}.$$

Présenter la réponse sous la forme d'une fonction Matlab

```
function [xbar0, sigmax0, A, B, sigmaw] = dynamics_data
```

qui renvoie  $\bar{\mathbf{x}}_0, \Sigma_{x_0}, \mathbf{A}, \mathbf{B}, \Sigma_w$  conformément aux spécifications du système.

Ecrire également une fonction

```
function [ss, sigmav] = measurements_data
```

qui renvoie la matrice  $S$  définie en (3), et  $\Sigma_v$ .

2. La fonction `simulate_system` fournie en annexe permet de générer sur  $T$  pas de temps une séquence d'états  $\mathbf{x}(t)$  et de mesures  $\mathbf{y}(t)$ , avec en argument  $T$  et les sorties des fonctions `dynamics_data` et `measurements_data`. Les fonctions `figure_trajectory` et `figure_measurements` également fournies tracent la trajectoire du véhicule, la position des balises, et les mesures recueillies en fonction du temps.

Générer une séquence d'états et de mesures de longueur  $T = 50$ , et les figures correspondantes.

Les mesures semblent-elles cohérentes avec la trajectoire ? Le bruit de mesure est-il visible ?

3. Définir la matrice  $C_t$  et le vecteur  $\mathbf{d}_t$  qui permettent de mettre (8) sous la forme

$$\mathbf{y}(t) = \mathbf{C}_t\mathbf{x}(t) + \mathbf{d}_t + \mathbf{v}(t).$$

Exprimer  $C_t$  et  $\mathbf{d}_t$  en fonction du vecteur  $\hat{\mathbf{x}}(t|t-1)$  autour duquel la linéarisation s'effectue.

Présenter la réponse sous la forme d'une fonction

```
function [C, ybar] = lin_meas(xhat_predict, ss)
```

qui prend en argument  $\hat{\mathbf{x}}(t|t-1)$ , la matrice  $S$ , et qui renvoie la matrice  $C_t$  et un vecteur  $\bar{\mathbf{y}}_t$  tel que

$$\bar{\mathbf{y}}_t = \mathbf{C}_t\hat{\mathbf{x}}_{t|t-1} + \mathbf{d}_t,$$

expression qu'on retrouve dans la formule de mise à jour de  $\hat{\mathbf{x}}(t|t)$  du filtre de Kalman.

4. Implémenter l'estimation d'état par le filtre de Kalman sous la forme d'une fonction

```
function [xx_hat, err] = estimate_states(...
    yy, xbar0, sigmax0, A, B, sigmaw, ss, sigmav)
```

qui prend en argument les mesures  $\mathbf{yy} \in \mathbb{R}^{8 \times T}$  fournies par la sortie `measurements` de la fonction `simulate_system`, d'autres arguments définis aux points précédents, et donne en sortie une matrice  $\mathbf{xx\_hat} \in \mathbb{R}^{4 \times T}$  dont les colonnes  $1 \leq i \leq T$  correspondent aux estimations  $\hat{\mathbf{x}}(t|t)$  en  $t = i - 1$ , ainsi qu'un vecteur  $\mathbf{err} \in \mathbb{R}^T$  dont l'élément  $i$  vaudra  $\hat{\epsilon}_t$  pour  $t = i - 1$ , avec

$$\hat{\epsilon}_t \triangleq \{(\Sigma_{t|t})_{11} + (\Sigma_{t|t})_{22}\}^{1/2}.$$

(En fait la somme des 2 premiers éléments diagonaux de  $\Sigma_{t|t}$  correspond à [une approximation de] l'erreur quadratique moyenne de l'estimation de la position.)

Ne pas perdre de vue qu'il faut appeler `lin_meas` à chaque itération pour mettre à jour  $C_t$  et  $\bar{\mathbf{y}}_t$ , afin d'évaluer  $\hat{\mathbf{x}}(t|t)$ ,  $\Sigma_{t|t}$ ,  $\hat{\mathbf{x}}(t+1|t)$  et  $\Sigma_{t+1|t}$ .

5. Superposer la trajectoire prédite (extraite de `xx_hat`) à la trajectoire exacte (extraite de `states` en sortie de `simulate_system` et déjà tracée). Commenter brièvement.  
Créer une nouvelle figure présentant l'erreur d'estimation de position réalisée  $\|\hat{\mathbf{p}}(t|t) - \mathbf{p}(t)\|$  en fonction du temps, et comparer à l'erreur prédite  $\hat{\epsilon}_t$ .
6. (Facultatif) Jouer sur différents paramètres : la variance du bruit de mesure, le nombre et la localisation des balises (qu'on peut par exemple placer aléatoirement), etc, et observer (qualitativement) en quoi les performances de l'estimation pour les simulations ainsi générées varient.

## Annexe : Codes Matlab

```
function [states, measurements] = ...
    simulate_system(T, xbar0, sigmax0, A, B, sigmaw, ss, sigmav);
%T : number of time steps, including time 0
n = 4; %number of state variables
m = size(ss,2); %number of measurement variables
states = zeros(n,T);
measurements = zeros(m,T);
%generate state x0 == [p0; u0]
%where p0 in R^2 is the initial position,
% u0 in R^2 is the initial velocity.
states(:,1) = xbar0 + (chol(sigmax0)')*randn(n,1);
%generate measurement y0 with noise v0
%y0(j) = ||p0 - ss(j)|| + v0(j) : measurement j at time 0
cholsigmav = chol(sigmav)';
vt = cholsigmav*randn(m,1); % measurement noise
measurements(:,1) = sqrt(sum((states(1:2,1)*ones(1,m)-ss).^2,1))' + vt;
%generate states and measurements
cholsigmaw = chol(sigmaw)';
for ii = 2:T
    wt = cholsigmaw*randn(2,1); %driving noise for the dynamics
    states(:,ii) = A*states(:,ii-1) + B*wt;
    vt = cholsigmav*randn(m,1); %measurement noise
    measurements(:,ii) = ...
        sqrt(sum((states(1:2,ii)*ones(1,m)-ss).^2,1))' + vt;
end

function figure_trajectory(ss, xx)
% draw the position of the beacons
m = size(ss,2); %number of beacons
figure('name','trajectory');
hold on
scolors = jet(m); % one color per beacon
for ii = 1:m
    plot(ss(1,ii),ss(2,ii),'s','color',scolors(ii,:));
end
% draw the trajectory
plot(xx(1,1),xx(2,1),'bo'); % encircle the initial position
plot(xx(1,:),xx(2:,:),'b:');
xlabel('X');
ylabel('Y');
axis equal

function figure_measurements(yy)
% draw the measurements
[m,T] = size(yy);
figure('name','measurements');
hold on
scolors = jet(m); % one color per beacon
for ii = 1:m
    plot([0:T-1],yy(ii,:),'color',scolors(ii,:));
end
xlabel('time');
ylabel('measurements');
```

*Bon travail.*