Observateurs et filtre de Kalman

Fabrice LE BARS

Présentation générale de la matière

Présentation générale de la matière

- Objectifs
  - Etre sensibilisé aux problèmes d’estimation d’état
  - Utiliser différents types d'observateurs d'état
  - Savoir implémenter un filtre de Kalman dans des cas simples
  - Avoir des notions sur ses applications et ses variantes courantes
  - Connaitre le vocabulaire courant autour de la notion de filtre de Kalman

Présentation générale de la matière

- Planning
  - 32C, en général le lundi après-midi, mardi, mercredi matin, et parfois lundi ou jeudi matin
  - Amener ses PC portables
  - Majorité des travaux sur ordinateur (MATLAB)

Présentation générale de la matière

- Planning
  - Révisions et travaux préliminaires (3x2C)
  - Vocabulaire et notions prérequises : vecteurs aléatoires, bruits, gaussiennes, matrices de covariance… (2C?)
  - Formalisation du filtre de Kalman et utilisation (9x2C?)
  - Extended Kalman Filter (2x2C?)
  - Lisseur de Kalman (?)
  - Filtres particulaires, ensemblistes ou autres techniques (?)
  - TE de révision (2C?)
  - TE noté (2C)

Révisions et travaux préliminaires

Révisions et travaux préliminaires

- Révisions
  - Cours d’automatique de 1A :
    - Certaines notions directement réutilisées (équations d'état, simulation avec Euler…),
    - D'autres en versions différentes (observateurs (en 1A ils étaient surtout pour les systèmes linéaires invariants), régulateurs…)

Révisions et travaux préliminaires

- Programme
  - Modélisation de systèmes avec des équations d'état
  - Simulation par méthode d'Euler
  - Coordonnées homogènes
  - Observateur d'état
  - Fusion de données
  - Commande
  - Implémentation en MATLAB : en simulation, et avec un système réel

Révisions et travaux préliminaires

- Modélisation de systèmes avec des équations d'état
  - Le fonctionnement de très nombreux systèmes (voiture, bateau…) de la vie quotidienne peut être modélisé par des équations d’état
  - Equation d’état/représentation d’état :

$$\left\{\begin{array}{rcll} \dot{\mathbf{x}}(t) & = & \mathbf{f}(\mathbf{x}(t),\mathbf{u}(t)) & \text{équation d'évolution} \\ \mathbf{y}(t) & = & \mathbf{g}(\mathbf{x}(t),\mathbf{u}(t)) & \text{équation d'observation} \end{array}\right.$$

  - Variables d’état : en général les variables nécessaires pour dessiner le système à un t donné + celles permettant de prévoir ce qui se passera au t suivant

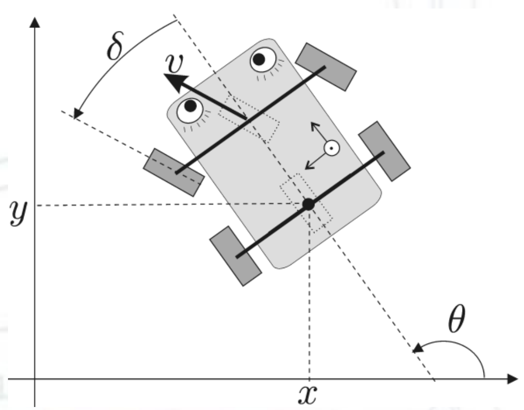
Révisions et travaux préliminaires

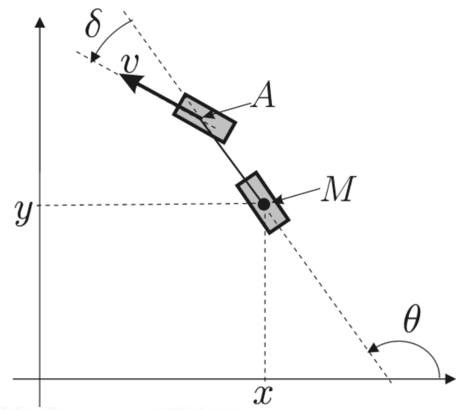
- Modélisation de systèmes avec des équations d'état
  - Etat : vecteur souvent noté $\mathbf{x}$ regroupant les variables d'état
  - Entrées : vecteur souvent noté $\mathbf{u}$ regroupant en général les signaux de commande directement envoyés au système, ou parfois leurs mesures plus ou moins directes
  - Sorties : vecteur souvent noté $\mathbf{y}$ regroupant en général les variables intéressantes mesurées par les capteurs du système

Révisions et travaux préliminaires

- Modélisation de systèmes avec des équations d'état
  - Equation d’évolution : équation différentielle permettant de savoir vers où va se diriger l’état x(t) sachant sa valeur à l’instant présent t et la commande u(t) actuelle
  - Equation d’observation : calcul des sorties y(t) actuelles en fonction de l’état actuel x(t) et la commande actuelle u(t)
  - Exemple : voiture

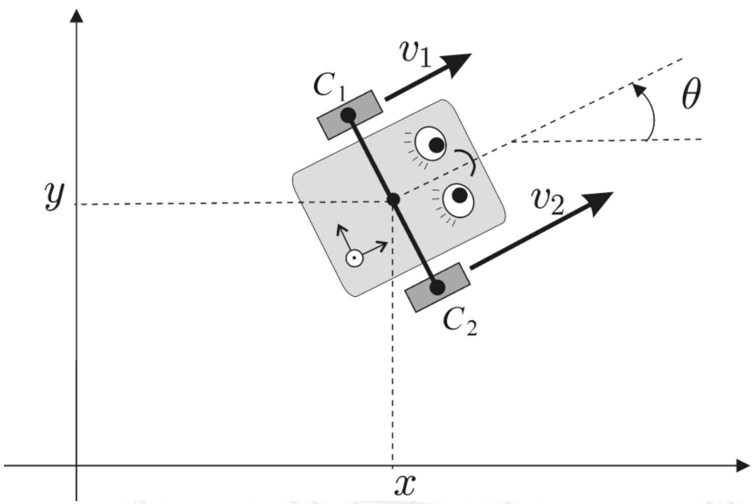
$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{v} \\ \dot{\delta} \end{pmatrix} = \begin{pmatrix} v \cos\delta \cos\theta \\ v \cos\delta \sin\theta \\ \frac{v \sin\delta}{L} \\ u_1 \\ u_2 \end{pmatrix}$$





Révisions et travaux préliminaires

- Modélisation de systèmes avec des équations d'état
  - Exemple : char



$$\begin{cases} \dot{x} = \frac{r\alpha_1 u_1 + r\alpha_2 u_2}{2} \cos\theta \\ \dot{y} = \frac{r\alpha_1 u_1 + r\alpha_2 u_2}{2} \sin\theta \\ \dot{\theta} = \frac{r\alpha_2 u_2 - r\alpha_1 u_1}{l} \end{cases}$$

$$\begin{aligned} v &= \frac{v_1+v_2}{2} \\ \omega &= \frac{v_2-v_1}{l} \\ \omega_1 &= \alpha_1 u_1 \\ \omega_2 &= \alpha_2 u_2 \\ v_1 &= r\omega_1 \\ v_2 &= r\omega_2 \end{aligned}$$

r Rayon des roues

Distance entre roues

Révisions et travaux préliminaires

- Modélisation de systèmes avec des équations d'état
  - Exemple : autre type de char

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} v \cos \theta \\ v \sin \theta \\ u_1 \\ u_2 \end{pmatrix}$$

  - Exemple : modèle de robot simple et assez général évoluant en 2.5D (e.g. quadrirotor, sous-marin…), souvent utilisé en post-traitement

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \begin{pmatrix} u_1 \cos u_4 - u_2 \sin u_4 \\ u_1 \sin u_4 + u_2 \cos u_4 \\ u_3 \end{pmatrix}$$

Révisions et travaux préliminaires

- Simulation par méthode d'Euler
  - Une fois qu’on a trouvé des équations d’état pour un système, il est bon de faire une simulation pour voir si elles représentent bien son comportement
  - Vu que l’équation d’évolution est une équation différentielle, on peut utiliser une méthode d’intégration numérique comme la méthode d’Euler :

Avec $\mathbf{x}\left(t+dt\right) \simeq \mathbf{x}\left(t\right) + dt.\dot{\mathbf{x}}(t)$

Vu l’équation d’évolution $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))$

On a $\mathbf{x}\left(t+dt\right) \simeq \mathbf{x}\left(t\right) + dt.\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))$

Révisions et travaux préliminaires

- Simulation par méthode d'Euler en MATLAB
  - Dans le code MATLAB, on va noter le vecteur d'état x comme un vecteur MATLAB, e.g. pour un modèle char $\mathbf{x} = (x, y, \theta, v)$:

$$\begin{array}{ccc} \text{MATLAB} & \rightarrow & \text{Théorie} \\ \mathbf{x}(1) & \rightarrow & x \\ \mathbf{x}(2) & \rightarrow & y \\ \mathbf{x}(3) & \rightarrow & \theta \\ \mathbf{x}(4) & \rightarrow & v \end{array}$$

  - Les fonctions d'évolution f et d'observation g seront codées comme des fonctions MATLAB : e.g. pour f d'un modèle char

```
function xdot = f(x,u)
xdot = [x(4)*cos(x(3));
     x(4)*sin(x(3));
     u(1);
     u(2)];
end
```

Révisions et travaux préliminaires

- Simulation par méthode d'Euler en MATLAB
  - Une simulation en MATLAB pour un modèle char :

```
x = [0;0;0;10];
u = [0;0];
dt = 0.01;
t = 0;
while (t < 30)
    x = x+f(x,u)*dt;
    clf; draw(x);
    pause(dt);
    t = t+dt;
end
```

Révisions et travaux préliminaires

- Coordonnées homogènes
  - Pour dessiner un système simulé, on a souvent des rotations et des translations d'éléments à faire
  - Pour combiner ces 2 types d'opérations facilement, on peut utiliser le formalisme des coordonnées homogènes

    e.g. si on veut faire une rotation de theta puis une translation de x,y, il nous faut définir la matrice

$$\mathbf{R} = \begin{pmatrix} \cos\theta & -\sin\theta & x \\ \sin\theta & \cos\theta & y \\ 0 & 0 & 1 \end{pmatrix}$$

Révisions et travaux préliminaires

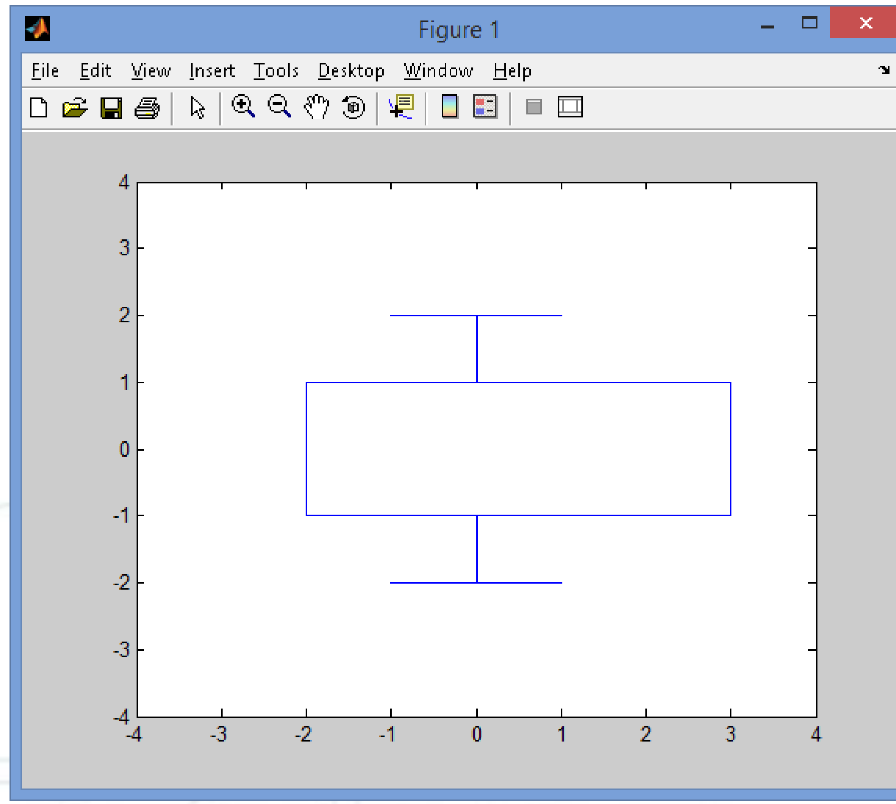
- Coordonnées homogènes
  - Le motif 2D des lignes représentant notre système au repos (**x=0**) devra être défini comme une matrice

    M=[coordonnées x…;coordonnées y…;1…]
  - Les 1 de la 3ème colonne sont nécessaires pour que la multiplication par la matrice **R** fonctionne comme attendu

Révisions et travaux préliminaires

- Coordonnées homogènes en MATLAB
  - Dans le code MATLAB, e.g. pour un modèle char pour **x=0**

```
function draw(x)
M = [1 -1   0   0 -2 -2 0 0 -1 1 0 0 3   3   0;
    -2 -2 -2 -1 -1   1 1 2   2 2 2 1 1 -1 -1;
     1   1   1   1   1   1 1 1   1 1 1 1 1   1   1];
R = [ cos(x(3)) -sin(x(3))  x(1);
      sin(x(3))  cos(x(3))  x(2);
      0          0          1];
M = R*M;
plot(M(1,:),M(2,:),'b');
end
```



Révisions et travaux préliminaires

- Coordonnées homogènes en MATLAB
  - Fonctions MATLAB souvent utiles pour le dessin

```
figure; % Create a new figure
clf; % Clear the figure
hold on; % Ensure all the plot() go on the same figure
axis([-5,5,-5,5]); % Set the limits on the axis
axis square; % Make the axis orthonormal
```

Révisions et travaux préliminaires

- Observateur d'état
  - Sur un système réel, il est parfois difficile de mesurer correctement ou directement certaines variables : capteurs imprécis, trop coûteux, pas utilisables des les conditions où on est…

    Exemple : Le GPS marche bien quand on roule sur une route dégagée mais dans un long tunnel il ne marche pas du tout

  - Il faut donc trouver des méthodes théoriques et numériques pour évaluer ces valeurs : on appelle observateur d'état ces méthodes

Révisions et travaux préliminaires

- Observateur d'état
  - En automatique 1A, vous avez peut-être vu les observateurs de Luenberger pour les systèmes linéaires invariants, on peut en faire d'autres selon les spécificités du système étudié :
    - Dead-reckoning (navigation à l'estime) : en utilisation les équations d'état pour prédire la position à t+dt à partir de celle à t
    - Calculs par relations géométriques : relations de distance et/ou d'angles à des amers (objets de position plus ou moins connue)
    - Filtre de Kalman : détaillé dans la suite…
  - Note : souvent en robotique, on va confondre estimation d'état, observateur d'état avec localisation, estimation de position car bien souvent c'est la position qui nous intéresse parmi les variables d'état