



**Intitulé :** Robotique mobile en productique (1/2 module)

**Langue dans laquelle est dispensée le cours :** français

**Crédits ECTS :** 6

**Durées :**

Cours	20h
TDs	14h
TPs	16h

**Compétences acquises :** Le but de ce module est de présenter les problématiques liées à la robotique mobile et son usage de plus en plus fréquent en productique. On parlera donc de géolocalisation, de poursuite et de planification de trajectoire. Le cours sera illustré par des applications.

## Contenu, programme :

- ▶ Dynamique des robots mobiles
  - Modélisation : équations d'Euler-Lagrange
  - Contraintes holonomes et non holonomes
  - Contrôle non linéaire
    - Commande de Lyapunov pour les systèmes conservatifs
    - Linéarisation par retour d'état (*feedback*)
    - Commande géométrique
  - Observateurs à convergence exponentielle
    - Observateur de Luenberger grand gain
    - Filtre de Kalman étendu à gain adaptatif
- ▶ Cinématique des robots mobiles
  - Contrôlabilité : crochets de Lie et formule BCH
  - Planification de trajectoire
  - Méthodes empiriques de planification

**Responsables** : Toufik Bakir, Éric Busvelle

Plusieurs parties de ce cours sont extraites des cours de :

- ▶ *Frédéric JEAN*, (ENSTA) : Algorithmes pour la planification de mouvements en robotique non-holonyme ;
- ▶ *Jean-Pierre LAUMOND*, (LAAS) : Robot motion planning and control – livre collectif
- ▶ *Alessandro DE LUCA*, Sapienza : Wheeled mobile robots

Les exemples sont tirés d'études récentes (toujours en cours pour les trois premières)

- ▶ Chariots transpalettes autonomes (Balyo)
- ▶ Assistance à la conduite (Edies)
- ▶ Moteurs *brushless* (thèse de Khaled Bakir)
- ▶ Traitement robotisé des adventices (ENESAD)
- ▶ Ball & wheel (thèse de Kenneth Sebesta)

## Introduction

Il s'agit plus précisément de l'**observateur de Kalman**

- ▶ **étendu** *i.e.* pour des systèmes non linéaires
- ▶ **grand-gain** : permettant une convergence rapide (exponentielle)
- ▶ **adaptatif** : étant capable de s'adapter aux perturbations
- ▶ **continu-discret** : adapté à des mesures échantillonnées asynchrones

Nous traiterons explicitement du cas multisorties mais afin d'alléger les notations, nous traiterons le cas à 2 sorties : passer de 1 à 2 n'est pas immédiat, mais passer de 2 à 3, 4, ... est évident. On ne parlera pas **ici** d'observabilité et on partira directement d'un système sous forme canonique.

Le système s'écrit

$$\dot{x} = Ax + b(x, u)$$

$$y(t_k) = C_k x(t_k)$$

où  $x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$ ,  $x_i \in \mathbb{R}^{n_i}$  tel que  $\sum_{i=1}^p n_i = n$  avec

$$A = \begin{pmatrix} A_1 & 0 & \cdots & 0 \\ 0 & A_2 & \ddots & 0 \\ 0 & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & A_p \end{pmatrix} \quad \text{où } A_i = \begin{pmatrix} 0 & a_{i,2} & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \cdots & a_{i,n_i} \\ 0 & \cdots & 0 & 0 \end{pmatrix}$$

Sommaire

Introduction

Observateurs

**Observateur de Kalman**

Prédiction

Correction

exemple

Modélisation

Formalisme  
d'Euler-Lagrange

Exemple motivant

Ball & wheel

Exercice

Modèle géométrique

Modèle cinématique

Pekoe et WiFiBot

Voiture

Moteur électrique  
(brushless)

Modèle complet

Montage expérimental

Démarrage du moteur

Vitesse stabilisée

Voiture à moteur thermique

Cinématique

Bilan énergétique

Équations d'état :

Sorties mesurées :

Application du PMP

Planification

le problème...

Stratégies de navigation

Classification

$$b = \begin{pmatrix} b_1(x, u) \\ b_2(x, u) \\ \vdots \\ b_p(x, u) \end{pmatrix} \text{ avec } b_i = \begin{pmatrix} b_{i,1}(x_{i,1}, u) \\ b_{i,2}(x_{i,1}, x_{i,2}, u) \\ \vdots \\ b_{i,n_i}(x, u) \end{pmatrix}$$

On reconnaît bien la forme canonique d'observabilité du cas monosortie généralisée sous forme de matrices blocs.

Le couplage des systèmes se fait dans la dernière ligne de chaque bloc ( $g_{i,n_i}$  dépend de  $x$  en entier).

Le cas des sorties est plus particulier encore : on suppose que le vecteur de sorties s'écrit  $y = Cx$  avec

$$C = \begin{pmatrix} C_1 & 0 & \cdots & 0 \\ 0 & C_2 & \ddots & 0 \\ 0 & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & C_p \end{pmatrix} \text{ et } C_i = (1 \ 0 \ \cdots \ 0)$$

Chaque instant  $t_k$  est un temps où certaines sorties sont mesurées (les mesures sont **échantillonnées**).

La matrice  $C_k$  est constituée des lignes de  $C$  correspondantes aux mesures effectivement disponibles au temps  $t_k$  (on ne suppose pas la **synchronicité** des mesures).

Entre deux instants d'échantillonnage  $[t_{k-1}, t_k[$ , on intègre le système et  $P$  en partant de  $(\hat{x}(t_{k-1}, P(t_{k-1})))$  pour calculer  $(\hat{x}(t_k^-, P(t_k^-)))$

$$\dot{\hat{x}} = A\hat{x} + b(\hat{x}, u)\dot{P} = (A + b(\hat{x}, u)^*)P + P(A + b(\hat{x}, u)^*) + Q_\theta$$

où  $Q_\theta = \theta \Delta_\theta^{-1} Q \Delta_\theta^{-1}$  avec

$$\Delta = \begin{pmatrix} \Delta_1 & 0 & \cdots & 0 \\ 0 & \Delta_2 & \ddots & 0 \\ 0 & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \Delta_p \end{pmatrix} \quad \text{où } \Delta_i = \begin{pmatrix} \frac{1}{\theta^{n^* - n_i}} & 0 \\ 0 & \ddots & \ddots \\ & \ddots & \frac{1}{\theta^{n^* - 1}} \end{pmatrix}$$

A chaque instant de mesure  $t_k$ , on effectue les corrections

$$\begin{aligned}\hat{x}(t_k) &= \hat{x}(t_k^-) + G_k (y(t_k) - C_k x(t_k^-)) \\ G_k &= P(t_k^-) C_k' (C_k P(t_k^-) C_k' + R_k)^{-1} \\ P(t_k) &= (I - G_k C_k) P(t_k^-)\end{aligned}$$

où la matrice  $R_k$  correspond aux covariances des sorties mesurées à l'instant  $t_k$ .

On va traiter un cas concret en dimension 5 et comportant 2 sorties.

# BLA BLA

L'équation d'Euler-Lagrange s'écrit :

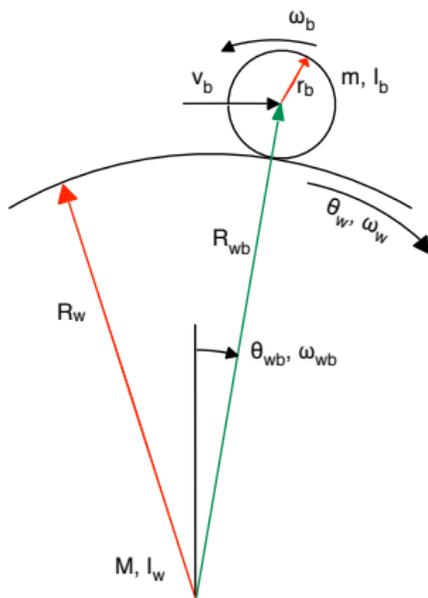
$$\frac{d}{dt} \left[ \frac{\partial L}{\partial \dot{q}} \right] - \frac{\partial L}{\partial q} + \frac{\partial J}{\partial \dot{q}} = \tau_0$$

où le Lagrangien du système est défini par

$$L(q, \dot{q}) = U(\dot{q}) - T(q)$$

et

- ▶  $U(\dot{q})$  (ou  $U(q, \dot{q})$ ) est l'énergie cinétique ;
- ▶  $T(q)$  est l'énergie potentielle ;
- ▶  $J(\dot{q})$  étant la perte d'énergie ;
- ▶  $\tau_0$  est la force ou le couple appliqué.



Ball & wheel : une balle sur une roue (jante)

### 1. Coordonnées :

$$q = \begin{pmatrix} \theta_{wb} \\ \theta_w \end{pmatrix}$$

### 2. Couple

$$\tau_0 = \begin{pmatrix} 0 \\ \tau_w \end{pmatrix}$$

### 3. Energies cinétique et potentielle

$$U(\dot{q}) = \frac{1}{2} I_w \omega_w^2 + \frac{1}{2} I_b \omega_b^2 + \frac{1}{2} m v_b^2$$

$$T(q) = mg(R + r) \cos(\theta_{wb})$$

### 4. Perte d'énergie : négligée ( $J(\dot{q}) = 0$ )

A cela s'ajoute les conditions de roulement sans frottement permettant d'exprimer  $v_b$  et  $\omega_b$  en fonction des variables d'état.

$$v_w = \omega_w R$$

$$v_w = \omega_b r + v_b$$

$$v_b = \dot{\theta}_{wb}(R + r)$$

ce qui donne

$$\omega_w R = \omega_b r + \dot{\theta}_{wb}(R + r)$$

ou encore

$$\omega_b = \frac{1}{r}(\omega_w R - \dot{\theta}_{wb}(R + r))$$

On substitue ces équations dans l'expression de l'énergie :

$$U(\dot{q}) = \frac{1}{2} I_w \omega_w^2 + \frac{1}{2} I_b \left( \frac{1}{r} (\omega_w R - \dot{\theta}_{wb} (R + r)) \right)^2 + \frac{1}{2} m (\dot{\theta}_{wb} (R + r))^2$$

$$T(q) = mg(R + r) \cos(\theta_{wb})$$

et on obtient le Lagrangien.

Après calculs, on obtient l'équation du second ordre du système mécanique :

Modèle de la balle sur la roue

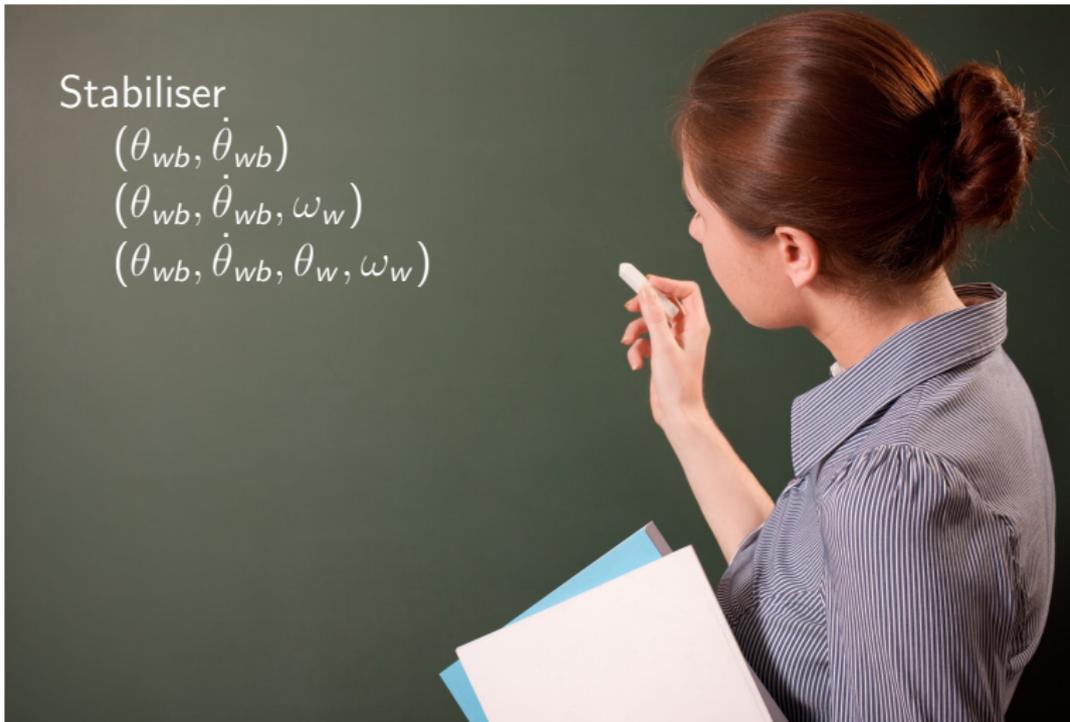
$$\ddot{\theta}_{wb} = \left( \frac{1}{R + r} \right) \left[ \frac{I_b R \tau_w + mg \sin(\theta_{wb}) (I_w r^2 + I_b R^2)}{I_b I_w + I_w m r^2 + I_b m R^2} \right]$$

Stabiliser

$$(\theta_{wb}, \dot{\theta}_{wb})$$

$$(\theta_{wb}, \dot{\theta}_{wb}, \omega_w)$$

$$(\theta_{wb}, \dot{\theta}_{wb}, \theta_w, \omega_w)$$



Première indication (pour la deuxième) :

$$\omega_w = \frac{(I_b + mr^2)\tau_w + I_b R m g \sin(\theta_{wb})}{I_b I_w + I_w m r^2 + I_b m R^2}$$

Indication ultime :

$$z_1 = \varphi = a\dot{\theta}_{wb} - \alpha\omega_w$$

Première indication (pour la deuxième) :

$$\omega_w = \frac{(I_b + mr^2)\tau_w + I_b R m g \sin(\theta_{wb})}{I_b I_w + I_w m r^2 + I_b m R^2}$$

Indication ultime :

$$z_1 = \varphi = a\dot{\theta}_{wb} - \alpha\omega_w$$

- ▶ Robot mobile = point dans  $E$ , l'espace des configurations
- ▶ Habituellement,  $E \subset \mathbb{R}^n$

### Exemple : l'unicycle

Le robot à une roue :

$$(x, y, \theta) \in E = \mathbb{R}^2 \times S^1 \text{ ou } \mathbb{R}^3$$

- ▶ Obstacles =  $O \subset E$  donc espace de travail =  $E \setminus O$

Usuellement, les équations du robot sont des équations différentielles liées à la mécanique

$$\dot{q}(t) = F_0(q(t)) + \sum_{i=1}^p F_i(q(t)) u_i(t)$$

système pour lequel

- ▶  $F_0(q(t))$  est un champ de vecteur appelé le *drift* ;
- ▶  $F_i(q(t))$ ,  $i = 1, \dots, n$  sont les champs de vecteurs qui sont les directions de contrôle

**Pekee** est un robot classique avec

- ▶ deux roues parallèles motorisées indépendantes ;
- ▶ une roue morte (castor) pour assurer la stabilité

$$\begin{cases} \dot{x}(t) &= \frac{1}{2} (v_1(t) + v_2(t)) \cos \theta(t) \\ \dot{y}(t) &= \frac{1}{2} (v_1(t) + v_2(t)) \sin \theta(t) \\ \dot{\theta}(t) &= \frac{1}{L} (v_1(t) - v_2(t)) \\ \dot{v}_1(t) &= u_1(t) \\ \dot{v}_2(t) &= u_2(t) \end{cases}$$

où

- ▶  $(x(t), y(t))$  est le point milieu des deux roues, espacées de  $L$
- ▶  $\theta(t)$  est la direction des deux roues
- ▶  $v_1(t)$  et  $v_2(t)$  sont les vitesses des roues :  $|v_j(t)| \leq v_{\max}$

$$\begin{cases} \dot{x}(t) &= v(t) \cos \theta(t) \\ \dot{y}(t) &= v(t) \sin \theta(t) \\ \dot{\theta}(t) &= \omega(t) \end{cases}$$

où on a le changement de variable sur les contrôles

- ▶  $v(t) = \frac{1}{2} (v_1(t) + v_2(t)) =$  vitesse de la voiture ;
- ▶  $\omega(t) = \frac{1}{L} (v_1(t) - v_2(t)) =$  vitesse de la rotation des roues ;

et

$$q(t) = \begin{pmatrix} x(t) \\ y(t) \\ \theta(t) \end{pmatrix}$$

**WiFibot** est un robot un peu plus complexe dans sa conception mécanique mais qui est strictement équivalent à **Pekee**.

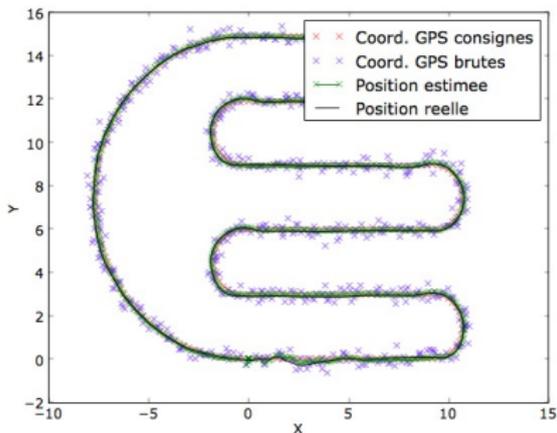
Soit

- ▶  $x(t), y(t)$  le point central entre les roues avant...
- ▶ supposées rouler à vitesse  $w(t)$ .
- ▶  $\varphi(t)$  l'angle entre les roues avant ...
- ▶ ...et la direction de la voiture  $\theta(t)$ .

$$\begin{cases} \dot{x}(t) &= w(t) \cos \varphi(t) \cos \theta(t) \\ \dot{y}(t) &= w(t) \cos \varphi(t) \sin \theta(t) \\ \dot{\theta}(t) &= w(t) \sin \varphi(t) \\ \dot{w}(t) &= u_1(t) \\ \dot{\varphi}(t) &= u_2(t) \end{cases}$$

où  $|u_1(t)| \leq u_{1,\max}$  et  $|u_2(t)| \leq u_{2,\max}$  On suppose que  $\varphi(t) \leq \varphi_{\max}$  ce qui induit un rayon de braquage minimal.

L'ENESAD a développé un robot sur une architecture de Quad : le modèle de ce robot est le même que celui de la voiture.



- ▶ **robots holonomes.** Pas de contraintes liées au déplacement du robot lui-même : tout chemin dans  $E$  est admissible.
- ▶ **robots non holonomes.** Contraintes *cinématiques*, les chemins admissibles doivent vérifier une contrainte liée au déplacement

$$\dot{q}(t) \in V_{q(t)} \subsetneq \mathbb{R}^n \quad \forall t$$

## Exemple

Contrainte de roulement sans glissement :

$$\dot{q}(t) = v \begin{pmatrix} \cos \theta(t) \\ \sin \theta(t) \\ 0 \end{pmatrix} + \omega \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

donc

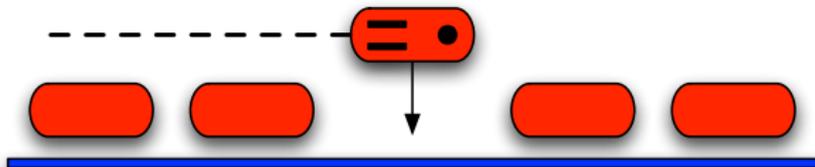
$$\dot{x} \sin \theta - \dot{y} \cos \theta = 0$$

On peut approcher une trajectoire non admissible (qui viole les contraintes non holonome) en utilisant les outils du contrôle géométrique.

Exemples :

- ▶ Un créneau en voiture, avec ou sans remorque
- ▶ Un voilier qui remonte le vent
- ▶ Un planeur qui cherche à profiter des courants ascendants

Prenons le premier cas : il s'agit de déplacer un robot-unicycle orthogonalement à la direction de ses roues !



1. On note  $F_1 = \begin{pmatrix} \cos \theta(t) \\ \sin \theta(t) \\ 0 \end{pmatrix}$  et  $F_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$  les deux directions admissibles ;

2. On calcule  $[F_1, F_2] = \begin{pmatrix} \cos \theta(t) \\ \sin \theta(t) \\ 0 \end{pmatrix} = F_3$  la direction engendrée pas les crochets ;

3.  $(F_1, F_2, F_3)$  est une base de  $\mathbb{R}^3$  donc inutile de calculer d'autres crochets ; au passage, le système est localement contrôlable ;

4. On se déplace dans la direction  $F_3$  en appliquant successivement  $F_1, -F_2, -F_1$  puis  $F_2$  puisque

$$\begin{aligned} q(t + 4T) &= e^{T F_2(q)} e^{-T F_1(q)} e^{-T F_2(q)} e^{T F_1(q)} q(t) \\ &= e^{T^2 [F_1, F_2](q)} q(t) + o(T^2) \end{aligned}$$

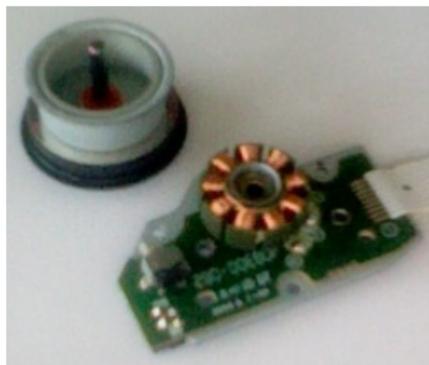
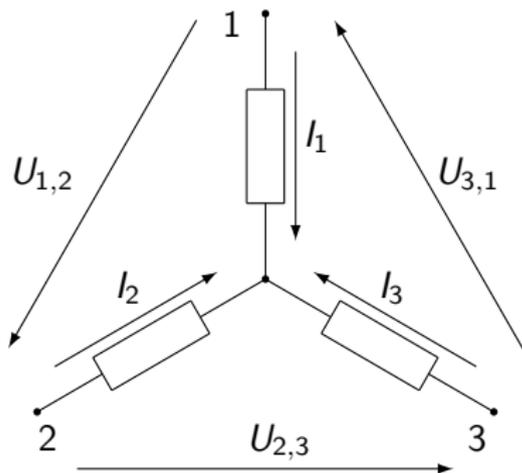
En robotique mobile, la norme pour un robot est d'être non holonome :

- ▶ tous les robots à roues
- ▶ tous les robots à patte (c'est pire!)
- ▶ les robots nageurs, rampants, volants...
- ▶ robots en apesanteur (conservation du moment angulaire)
- ▶ robots bioinspirés : anguilles, insectes, ...

## Remarque

Pour tous ces robots, le chemin le plus simple pour aller d'un point *A* à un point *B* **n'est pas** la ligne droite !

Très utilisés en robotique !



La partie électrique du modèle est donnée par trois équations, une pour chaque phase  $j = 1, 2, 3$  du moteur :

$$U_j - U_0 = R \cdot I_j + L \frac{dI_j}{dt} + E_j \quad (1)$$

où

- ▶  $U_j - U_0$  est la tension dans la phase  $j$  où  $U_0$  est le potentiel au centre du montage en étoile ;
- ▶  $I_j$  est l'intensité dans la phase  $j$  ;
- ▶  $E_j$  est la force contre-électromotrice dans la phase  $j$  ;
- ▶  $R$  et  $L$  sont la résistance et l'inductance du stator.

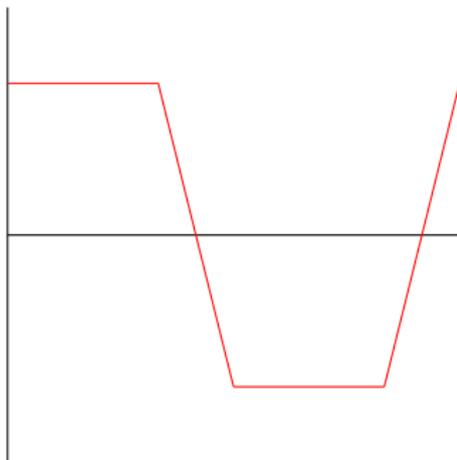
et remarquons que

$$I_1 + I_2 + I_3 = 0 \quad (2)$$

$$E_j = \dot{\theta}_e F(\theta_e - \frac{2\pi}{3}(j-1))$$

$$T_E = \frac{1}{\omega_m} \sum_{j=1}^3 E_j I_j$$

$$= \sum_{j=1}^3 F(\theta_e - \frac{2\pi}{3}(j-1)) I_j$$



Exemple de f.c.e.m. trapézoïdale :

$$F(\theta) = \begin{cases} 1 & 0 \leq \theta < \frac{2\pi}{3} \\ 1 - \frac{6}{\pi}(\theta - \frac{2\pi}{3}) & \frac{2\pi}{3} \leq \theta < \pi \\ -1 & \pi \leq \theta < \frac{5\pi}{3} \\ -1 + \frac{6}{\pi}(\theta - \frac{5\pi}{3}) & \frac{5\pi}{3} \leq \theta < 2\pi \end{cases}$$

La variable  $\theta_e$  est proportionnelle à l'angle du rotor, plus précisément,

$$\theta_e = \frac{p}{2}\theta_m$$

où  $\theta_m$  est la position du rotor et

$$\omega_m = \frac{d\theta_m}{dt}$$

est la vitesse angulaire du rotor.

La dernière équation du modèle traduit la partie mécanique du système :

$$J \frac{d\omega_m}{dt} = -B\omega_m + T_E - T_L$$

où

- ▶  $J$  est l'inertie du rotor ;
- ▶  $B$  est le coefficient de frottement ;
- ▶  $T_L$  est le couple de charge ;
- ▶  $T_E$  est le couple électromagnétique :

$$T_E = \sum_{j=1}^3 F(\theta_e - \frac{2\pi}{3}(j-1)) I_j$$

$$L \frac{dl_{j,k}}{dt} = -R.l_{j,k} + U_{j,k} - \omega_m \Phi(\theta_e - \frac{2\pi}{3}(j-1))$$

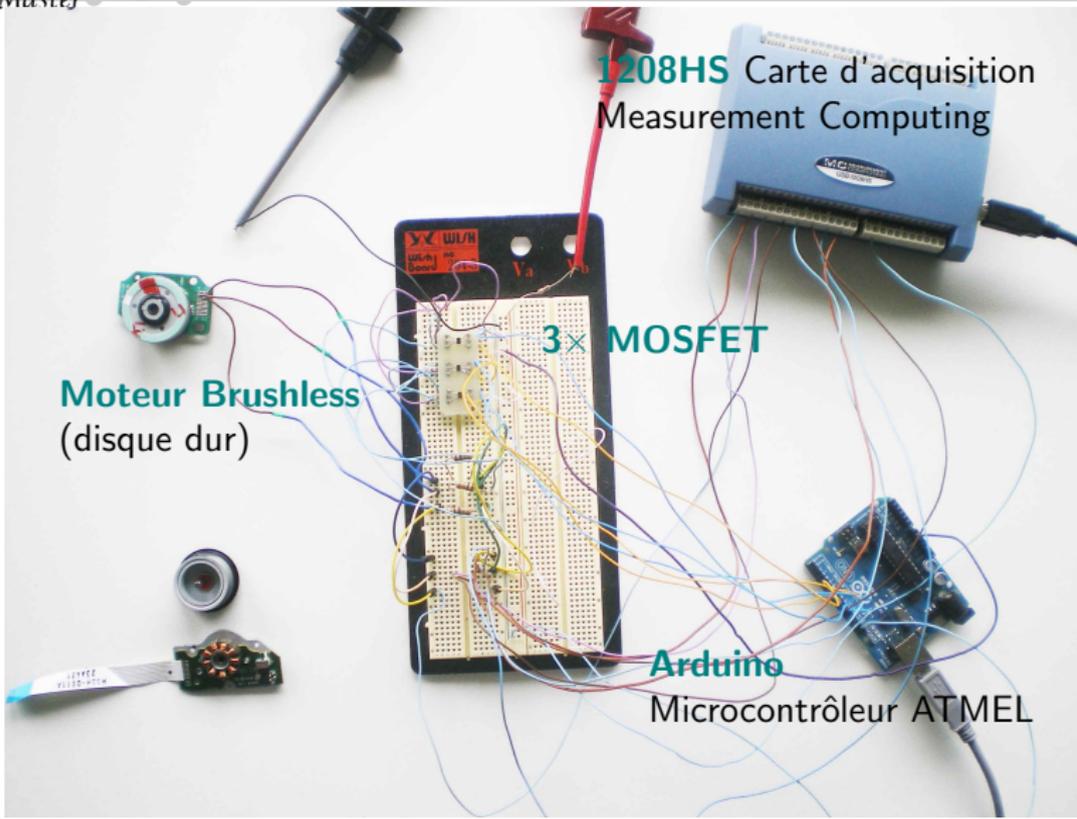
$$\frac{d\theta_m}{dt} = \omega_m$$

$$J \frac{d\omega_m}{dt} = -B\omega_m + T_E - T_L$$

pour  $(j, k) \in \{(1, 2), (2, 3), (3, 1)\}$  où  $l_{j,k} = l_j - l_k$  et  
 $U_{j,k} = U_j - U_k$  et où

$$\Phi(\theta_e) = F(\theta_e) - F(\theta_e - \frac{2\pi}{3})$$

- Observateurs
  - Observateur de Kalman
  - Prédiction
  - Correction
- exemple
- Modélisation
  - Formalisme d'Euler-Lagrange
  - Exemple motivant
  - Ball & wheel
  - Exercice
  - Modèle géométrique
  - Modèle cinématique
  - Pekoe et WiFiBot
  - Voiture
  - Moteur électrique (brushless)
  - Modèle complet
- Montage expérimental**
  - Démarrage du moteur
  - Vitesse stabilisée
- Voiture à moteur thermique
- Cinématique
- Bilan énergétique
- Équations d'état :
- Sorties mesurées :
- Application du PMP
- Planification
- le problème...
- Stratégies de navigation
- Classification

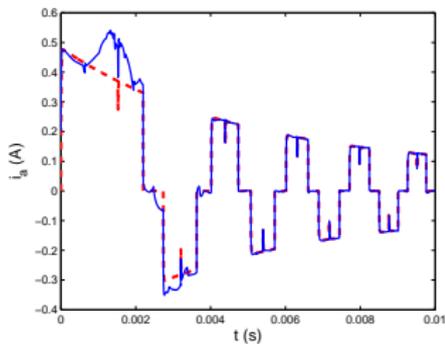
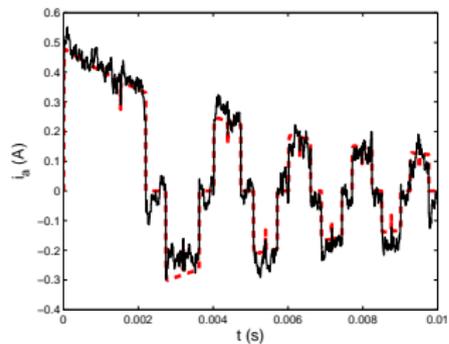


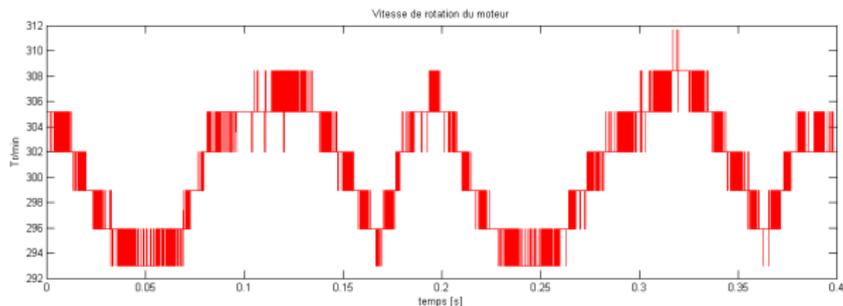
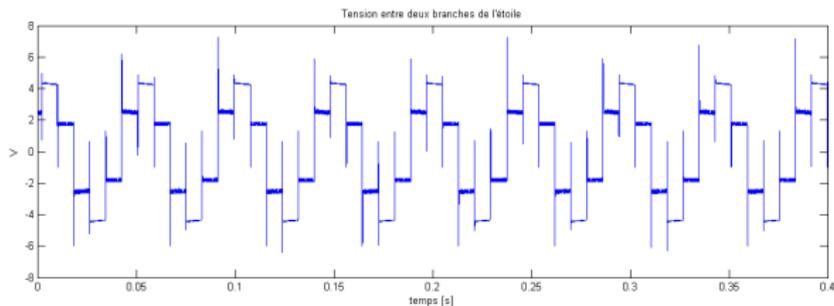
1208HS Carte d'acquisition  
Measurement Computing

3x MOSFET

Moteur Brushless  
(disque dur)

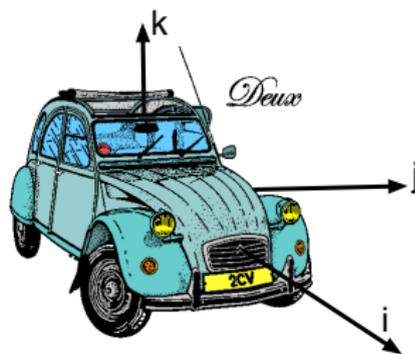
Arduino  
Microcontrôleur ATMEL



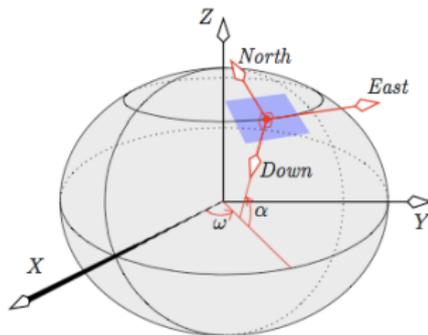


On utilise deux repères :

Un repère inertiel  $\widehat{ijk}$  attaché à la voiture.



Un référentiel global *NED*  
(North-East-Down) (cf Lambert 93)



L'accélération est proportionnelle à la somme des forces :

$$M_a \dot{v}_i = \Sigma F = F_{thrust} - F_{drag} - F_{roll} - F_{gravity}$$

$$F_{drag} = \frac{1}{2} \rho C_d A v_i^2$$

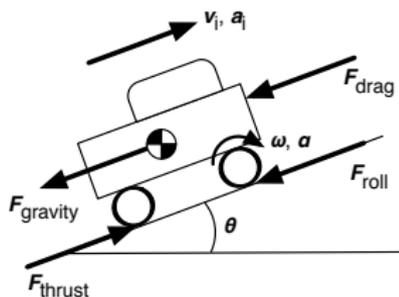
$$F_{roll} = C_{rr} (Mg \cos(\theta) + S_k v_i^2)$$

$$F_{gravity} = Mg \sin(\theta)$$

où

$$M_a = \frac{J}{r_w^2} + M$$

( $M_a$  est la masse accélérée équivalente)



$S_j$  and  $S_k$ , the road straightness coefficients, represent the inverse of road curvature :

$$a_{j,b} = S_j v_i^2$$

$$a_{k,b} = S_k v_i^2$$

Likewise, the rate of change of  $\psi$  and  $\theta$  is dependent on the road straightness coefficients and forward velocity :

$$\dot{\psi} = S_j v_i$$

$$\dot{\theta} = S_k v_i$$

$$\dot{N} = \cos(\psi) \cos(\theta) v_i$$

$$\dot{E} = \sin(\psi) \cos(\theta) v_i$$

$$\dot{D} = -\sin(\theta) v_i$$

The engine turns at a fixed rate with respect to the wheels, neutral gear and differential withstanding. In a normal car transmission, this ratio is different between gears but otherwise constant.

$$\dot{\omega}_e = \dot{v}_i N_{spd}$$

$$a_{i,m} = a_{i,b} + \frac{M}{M_a} g \sin(\theta)$$

$$a_{j,m} = a_{j,b}$$

$$a_{k,m} = a_{k,b} - g \cos(\theta)$$

La puissance entrante est le volume de fuel/seconde fois le rendement énergétique par litre :

$$P_{in} = k \dot{V}_{fuel}$$

La puissance récupérée par le système est la force de locomotion fois la vitesse :

$$P_{out} = F_{thrust} v_i$$

On définit l'efficacité par

$$\eta = \frac{P_{out}}{P_{in}}$$

La puissance fournie est également le couple fois la vitesse angulaire

$$P_{out} = T_e \omega_e$$

si bien que

$$T_e = F_{thrust} \frac{v_i}{\omega_e}$$

$$\dot{x}_i = v_i$$

$$\dot{v}_i = \frac{1}{M_a} \left( F_{thrust} - \text{sign}(v_i) \left( \frac{1}{2} \rho C_d A v_i^2 + M C_{rr} (g \cos(\theta) + S_k v_i^2) \right) - M g \sin(\theta) \right)$$

$$\dot{\psi} = S_j v_i$$

$$\dot{\theta} = S_k v_i$$

$$\dot{N} = v_i \cos(\psi) \cos(\theta)$$

$$\dot{E} = v_i \sin(\psi) \cos(\theta)$$

$$\dot{D} = -v_i \sin(\theta)$$

$$\dot{E}_{in} = P_{in}$$

$$\dot{\theta}_e = w_e$$

$$\text{et } \dot{F}_{thrust} = \dot{S}_j = \dot{S}_k = \dot{P}_{in} = \dot{\omega}_e = \dot{b}_i = \dot{b}_j = \dot{b}_k = 0$$

$$x_{i,m} = x_i$$

$$a_{i,m} = \frac{1}{M_a} \left( F_{thrust} - \text{sign}(v_i) \left( \frac{1}{2} \rho C_d A v_i^2 + M C_{rr} (g \cos(\theta) + S_k v_i^2) \right) \right)$$

$$a_{j,m} = S_j v_i^2$$

$$a_{k,m} = -g \cos(\theta) - S_k v_i^2$$

$$DEM_{\psi} = \psi$$

$$DEM_{alt} = -D$$

$$GPS_{lat} = N$$

$$GPS_{lon} = E$$

$$GPS_{alt} = -D$$

$$INJ_{duration} = E_{in}$$

$$INJ_{time} = \theta_e$$

### Exemple d'application

- ▶ Atteindre une vitesse donnée  $v^* = 140\text{km/h}$  ...
- ▶ ... en un temps donné  $T_f = 30\text{s}$ ...
- ▶ ... en minimisant la consommation d'essence.

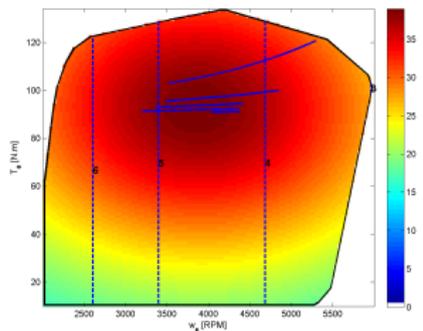
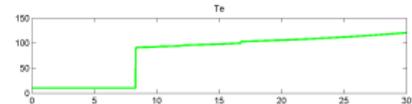
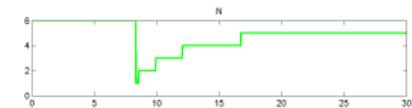
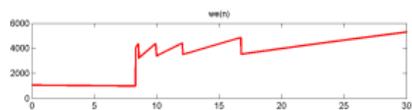
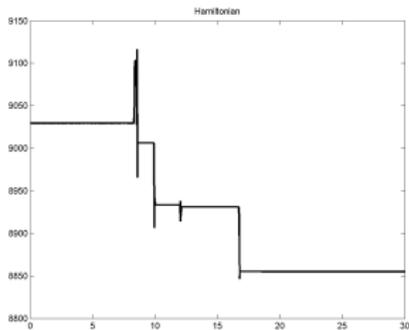
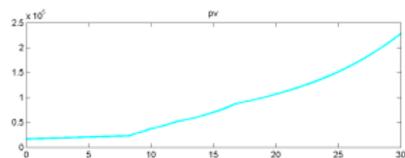
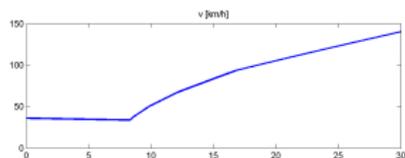
La voiture est légère, aérodynamique et assez sportive (smart roadster). Elle est munie d'une boîte de vitesse performante, à 6 rapports.

Le Hamiltonien s'écrit

$$\mathcal{H} = -p_0 \omega_e T_e \varphi(\omega_e, T_e) + \frac{p_v}{M} \left( \frac{n T_e}{R} - \frac{1}{2} \rho C_d A v^2 - C_{rr} m g \right)$$

Le problème est résolu explicitement par le PMP.

La solution est...



## Problème de la Planification de Mouvements

Étant donnés deux configurations  $q_0$  et  $q_1$  dans  $E$ , trouver un chemin admissible  $q(t)$ ,  $t \in [0, T]$  tel que

$$q(0) = q_0 \text{ et } q(T) = q_1$$

Nombreuses méthodes, que l'on peut classifier<sup>1</sup>

- ▶ Navigation par but
- ▶ Navigation par amers
- ▶ Action associée à un lieu
- ▶ Navigation topologique
- ▶ Navigation métrique

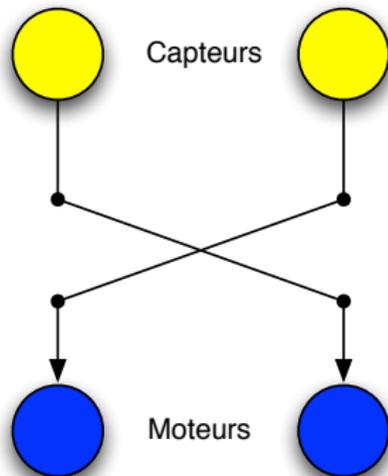
---

1. O. Trullier, S. Wiener, A. Berthoz, and J. A. Meyer. Biologically-based artificial navigation systems : Review and prospects. *Progress in Neurobiology*, 51, pp.483–544, 1997

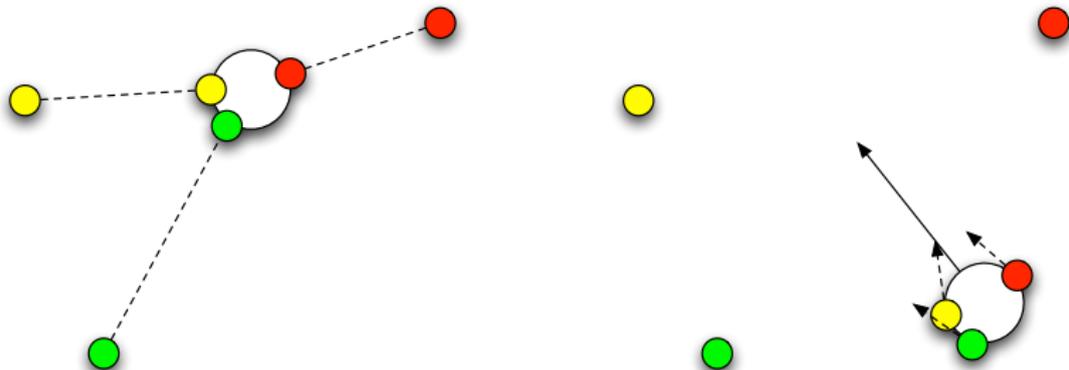
Exemple : s'approcher d'une source lumineuse, d'un émetteur, suivre un chemin magnétique ou noir sur fond blanc, etc...

La méthode est particulièrement simple pour un robot qui tourne grâce à un différentiel de vitesse de rotation des roues. Il n'y a pas besoin de traitement numérique.

Ce type de contrôle est assimilable à un réflexe, ou au comportement de certains insectes.



Il s'agit d'atteindre un but non matériel, représenté par une configuration d'amers, mémorisée par le robot.



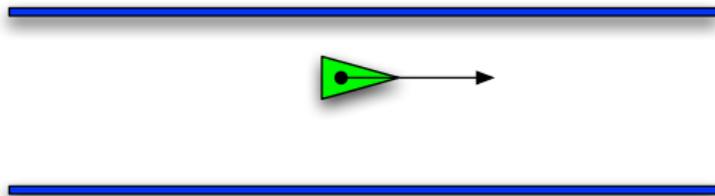
Pour chaque amer, on calcule un vecteur tangent au robot dont la longueur algébrique est proportionnelle à la distance de l'amer. La direction à prendre est la somme vectorielle des vecteurs obtenus. Cette méthode serait utilisée par les abeilles. Elle peut-être utilisée pour un robot muni de caméras.

Les solutions précédentes ne tiennent pas compte des obstacles à éviter.

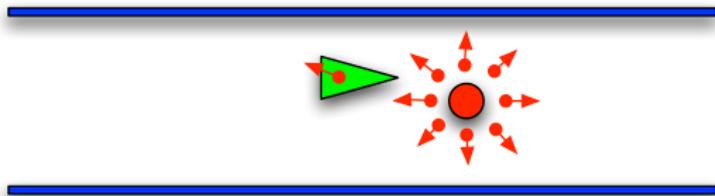
On va détailler trois approches :

1. La méthode du potentiel, basée sur des champs de répulsions ;
2. La méthode du champ d'occupation, basée sur une approche probabiliste (et la formule de Bayes) ;
3. La méthode de la fenêtre glissante, qui est une méthode proche de la commande prédictive et que l'on étudiera ultérieurement..

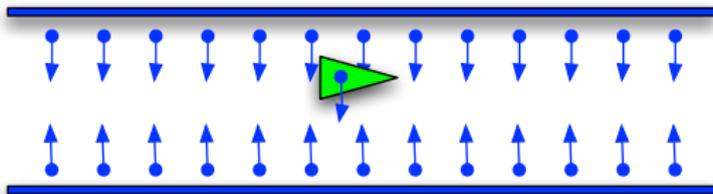
- ▶ Le robot doit aller tout droit...
- ▶ ...mais un poteau fait obstacle...
- ▶ ...et les parois sont infranchissables.
- ▶ On fait la somme vectorielle des champs répulsifs...
- ▶ ... et on obtient une trajectoire qui ne passe entre l'obstacle et la paroi.



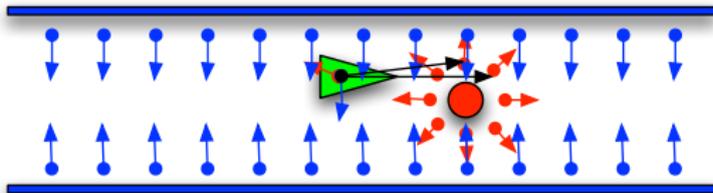
- ▶ Le robot doit aller tout droit...
- ▶ ...mais un poteau fait obstacle...
- ▶ ...et les parois sont infranchissables.
- ▶ On fait la somme vectorielle des champs répulsifs...
- ▶ ... et on obtient une trajectoire qui ne passe entre l'obstacle et la paroi.



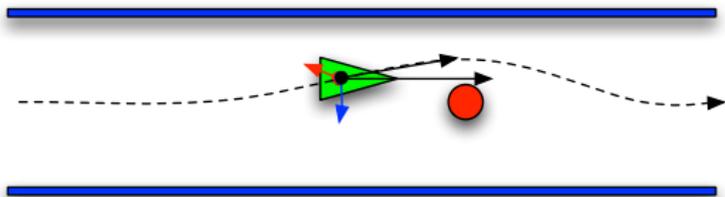
- ▶ Le robot doit aller tout droit...
- ▶ ...mais un poteau fait obstacle...
- ▶ ...et les parois sont infranchissables.
- ▶ On fait la somme vectorielle des champs répulsifs...
- ▶ ... et on obtient une trajectoire qui ne passe entre l'obstacle et la paroi.



- ▶ Le robot doit aller tout droit...
- ▶ ...mais un poteau fait obstacle...
- ▶ ...et les parois sont infranchissables.
- ▶ On fait la somme vectorielle des champs répulsifs...
- ▶ ... et on obtient une trajectoire qui ne passe entre l'obstacle et la paroi.

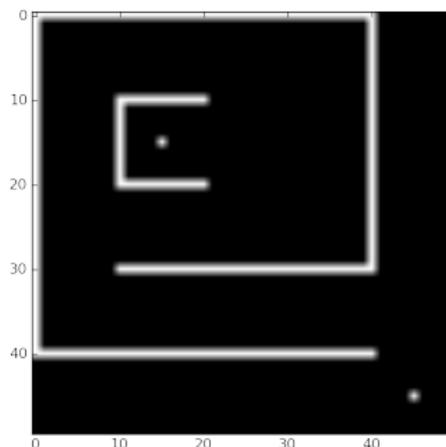


- ▶ Le robot doit aller tout droit...
- ▶ ...mais un poteau fait obstacle...
- ▶ ...et les parois sont infranchissables.
- ▶ On fait la somme vectorielle des champs répulsifs...
- ▶ ... et on obtient une trajectoire qui ne passe entre l'obstacle et la paroi.



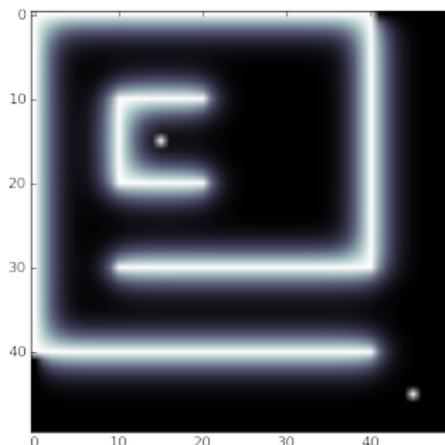
La méthode proposée est une variante, qui permet d'aller un peu plus loin en matière de planification que l'évitement d'obstacle. On considère le cas d'un robot qui doit trouver une cible cachée dans un labyrinthe. Ici, tout est supposé connu (carte des lieux...).

- ▶ Tous les obstacles sont des sources de froid, le sol est à température ambiante
- ▶ La cible est une source de chaleur, la chaleur se propage dynamiquement
- ▶ Le robot suit le gradient de température pour remonter jusqu'à la cible



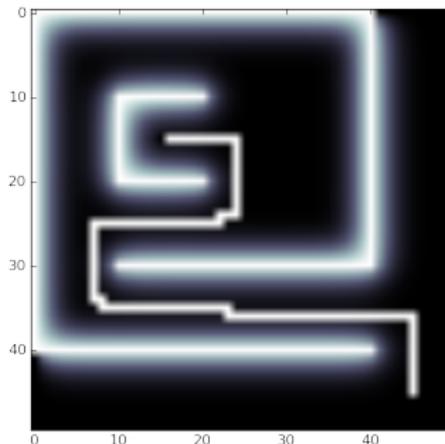
La méthode proposée est une variante, qui permet d'aller un peu plus loin en matière de planification que l'évitement d'obstacle. On considère le cas d'un robot qui doit trouver une cible cachée dans un labyrinthe. Ici, tout est supposé connu (carte des lieux...).

- ▶ Tous les obstacles sont des sources de froid, le sol est à température ambiante
- ▶ La cible est une source de chaleur, la chaleur se propage dynamiquement
- ▶ Le robot suit le gradient de température pour remonter jusqu'à la cible



La méthode proposée est une variante, qui permet d'aller un peu plus loin en matière de planification que l'évitement d'obstacle. On considère le cas d'un robot qui doit trouver une cible cachée dans un labyrinthe. Ici, tout est supposé connu (carte des lieux...).

- ▶ Tous les obstacles sont des sources de froid, le sol est à température ambiante
- ▶ La cible est une source de chaleur, la chaleur se propage dynamiquement
- ▶ Le robot suit le gradient de température pour remonter jusqu'à la cible



L'espace est "pixélisé" en sites  $s \in S$  : chaque point  $X_s$  de l'espace est initialement affecté d'une probabilité d'occupation

$$p_s^0 = \mathbb{P}(X_s = 1)$$

chaque mesure vient apporter une nouvelle information.

On note  $Y_s^N = (Y_s(1), \dots, Y_s(N))$  les  $N$  premières mesures disponibles sur le site  $s$ . Chaque mesure  $Y_s(i)$  donne une information  $p_s(i) = \mathbb{P}(X_s = 1 / Y_s(i))$ . On note

$$p_s^N = \mathbb{P}(X_s = 1 / Y_s^N)$$

On veut calculer  $p_s^N$  en fonction de  $p_s^{N-1}$ . Pour éliminer la constante de normalisation, on calculera plutôt

$$L_s^N = \log\left(\frac{p_s^N}{1 - p_s^N}\right)$$

en fonction de  $L_s^{N-1}$ .

$$\begin{aligned}
 p_s^N &= \mathbb{P}(X_s = 1/Y_s^N) = \mathbb{P}(X_s = 1/Y_s^{N-1}, Y_s(N)) \\
 &= \frac{\mathbb{P}(Y_s(N)/X_s = 1, Y_s^{N-1})\mathbb{P}(X_s = 1/Y_s^{N-1})}{\mathbb{P}(Y_s(N)/Y_s^{N-1})} \\
 &= \frac{\mathbb{P}(Y_s(N)/X_s = 1)\mathbb{P}(X_s = 1/Y_s^{N-1})}{\mathbb{P}(Y_s(N)/Y_s^{N-1})} \\
 &= \frac{\mathbb{P}(X_s = 1/Y_s(N))\mathbb{P}(Y_s(N))\mathbb{P}(X_s = 1/Y_s^{N-1})}{\mathbb{P}(X_s = 1)\mathbb{P}(Y_s(N)/Y_s^{N-1})}
 \end{aligned}$$

On en déduit

$$\frac{p_s^N}{1 - p_s^N} = \frac{\mathbb{P}(X_s = 1/Y_s(N))\mathbb{P}(X_s = 1/Y_s^{N-1})\mathbb{P}(X_s = 0)}{\mathbb{P}(X_s = 1)\mathbb{P}(X_s = 0/Y_s(N))\mathbb{P}(X_s = 0/Y_s^{N-1})}$$

Finalement :

$$L_s^N = L_s^{N-1} + \log \frac{p_s(N)}{1 - p_s(N)} - \log \frac{p_s^0}{1 - p_s^0}$$

1. le premier terme est la connaissance à actualiser ;
2. le second terme est la nouvelle information.
3. le troisième terme est l'information *a priori*

Le robot se déplace en privilégiant les cases à faible probabilité d'occupation.

A noter qu'on pourrait améliorer cet algorithme par l'utilisation d'un champ de Gibbs.

Sommaire

Introduction

Observateurs

Observateur de Kalman

Prédiction

Correction

exemple

Modélisation

Formalisme  
d'Euler-Lagrange

Exemple motivant

Ball & wheel

Exercice

Modèle géométrique

Modèle cinématique

Pekee et WiFibot

Voiture

Moteur électrique  
(brushless)

Modèle complet

Montage expérimental

Démarrage du moteur

Vitesse stabilisée

Voiture à moteur thermique

Cinématique

Bilan énergétique

Équations d'état :

Sorties mesurées :

Application du PMP

Planification

le problème...

Stratégies de navigation

Classification

Finalement :

$$L_s^N = L_s^{N-1} + \log \frac{p_s(N)}{1 - p_s(N)} - \log \frac{p_s^0}{1 - p_s^0}$$

1. le premier terme est la connaissance à actualiser ;
2. le second terme est la nouvelle information.
3. le troisième terme est l'information *a priori*

Le robot se déplace en privilégiant les cases à faible probabilité d'occupation.

A noter qu'on pourrait améliorer cet algorithme par l'utilisation d'un champ de Gibbs.

[Sommaire](#)[Introduction](#)[Observateurs](#)[Observateur de Kalman](#)[Prédiction](#)[Correction](#)[exemple](#)[Modélisation](#)[Formalisme  
d'Euler-Lagrange](#)[Exemple motivant](#)[Ball & wheel](#)[Exercice](#)[Modèle géométrique](#)[Modèle cinématique](#)[Pekée et WiFibot](#)[Voiture](#)[Moteur électrique  
\(brushless\)](#)[Modèle complet](#)[Montage expérimental](#)[Démarrage du moteur](#)[Vitesse stabilisée](#)[Voiture à moteur thermique](#)[Cinématique](#)[Bilan énergétique](#)[Équations d'état :](#)[Sorties mesurées :](#)[Application du PMP](#)[Planification](#)[le problème...](#)[Stratégies de navigation](#)[Classification](#)

Finalement :

$$L_s^N = L_s^{N-1} + \log \frac{p_s(N)}{1 - p_s(N)} - \log \frac{p_s^0}{1 - p_s^0}$$

1. le premier terme est la connaissance à actualiser ;
2. le second terme est la nouvelle information.
3. le troisième terme est l'information *a priori*

Le robot se déplace en privilégiant les cases à faible probabilité d'occupation.

A noter qu'on pourrait améliorer cet algorithme par l'utilisation d'un champ de Gibbs.

Sommaire

Introduction

Observateurs

Observateur de Kalman

Prédiction

Correction

exemple

Modélisation

Formalisme  
d'Euler-Lagrange

Exemple motivant

Ball & wheel

Exercice

Modèle géométrique

Modèle cinématique

Pekee et WiFibot

Voiture

Moteur électrique  
(brushless)

Modèle complet

Montage expérimental

Démarrage du moteur

Vitesse stabilisée

Voiture à moteur thermique

Cinématique

Bilan énergétique

Équations d'état :

Sorties mesurées :

Application du PMP

Planification

le problème...

Stratégies de navigation

Classification

Finalelement :

$$L_s^N = L_s^{N-1} + \log \frac{p_s(N)}{1 - p_s(N)} - \log \frac{p_s^0}{1 - p_s^0}$$

1. le premier terme est la connaissance à actualiser ;
2. le second terme est la nouvelle information.
3. le troisième terme est l'information *a priori* (si  $p_s^0 = \frac{1}{2}$ , il vaut 0) ;

Le robot se déplace en privilégiant les cases à faible probabilité d'occupation.

A noter qu'on pourrait améliorer cet algorithme par l'utilisation d'un champ de Gibbs.

Finalemnt :

$$L_s^N = L_s^{N-1} + \log \frac{p_s(N)}{1 - p_s(N)} - \log \frac{p_s^0}{1 - p_s^0}$$

1. le premier terme est la connaissance à actualiser ;
2. le second terme est la nouvelle information.
3. le troisième terme est l'information *a priori* (si  $p_s^0 = \frac{1}{2}$ , il vaut 0) ;

Le robot se déplace en privilégiant les cases à faible probabilité d'occupation.

A noter qu'on pourrait améliorer cet algorithme par l'utilisation d'un **champ de Gibbs**.

