ROBOTIQUE
-ELE4203-

Cours #13: Révision

Enseignant: Jean-Philippe Roberge

ÉCOLE POLYTECHNIQUE MONTRÉAL

Cours #13

- **Présentation du *robot de la semaine:***
  - Rat Neuron-Driven robot
    - *Plusieurs sources*
- **Retour sur le cours #12:**
  - Fin de la cinématique
  - Localisation à l'aide de capteurs
    - Introduction au filtre de Kalman:
      - 1) Estimation d'une quantité fixe à l'aide d'un lot de données
      - 2) Estimation d'une quantité fixe à l'aide d'un processus récursif
      - 3) Estimation d'un état d'un système dynamique à l'aide d'un processus récursif (filtre de Kalman classique)
      - 4) Estimation d'un état d'un système dynamique non-linéaire à l'aide d'un procesus récursif (filtre étendu de Kalman)

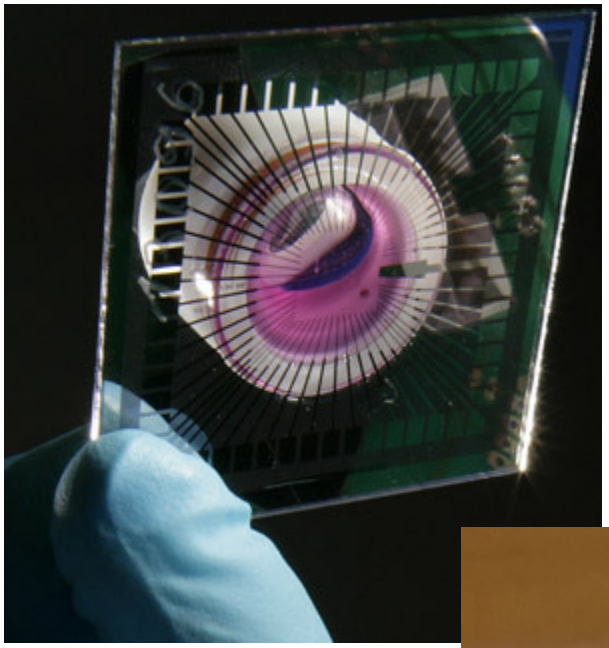
Cours #13

- **Cours #12 (suite):** ***Localisation, contrôle et planification de trajectoires***
  - Contrôle pour le suivi de trajectoires:
    - Contrôleurs classiques de vitesses de translation et de rotation
    - Contrôleur d'Astolfi
- **Cours #13 : Révision de la matière à l'examen**
- **Exercices**

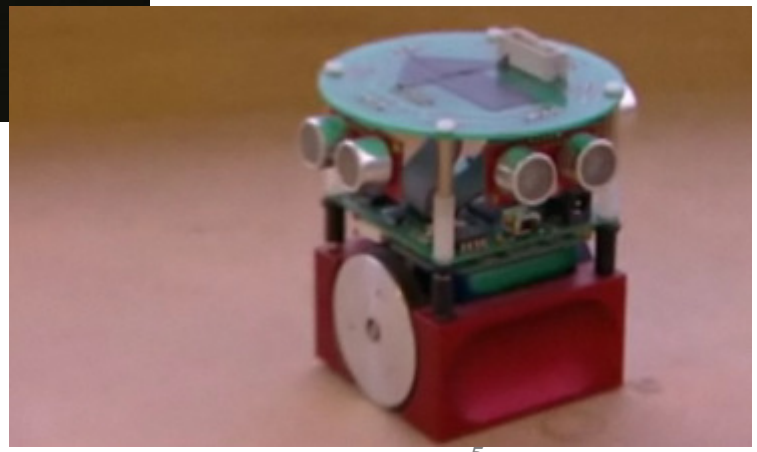
Le robot de la semaine:

Rat brain-driven Robot

Des robots partiellement biologiques…



Robot controlled by Neurons.mp4

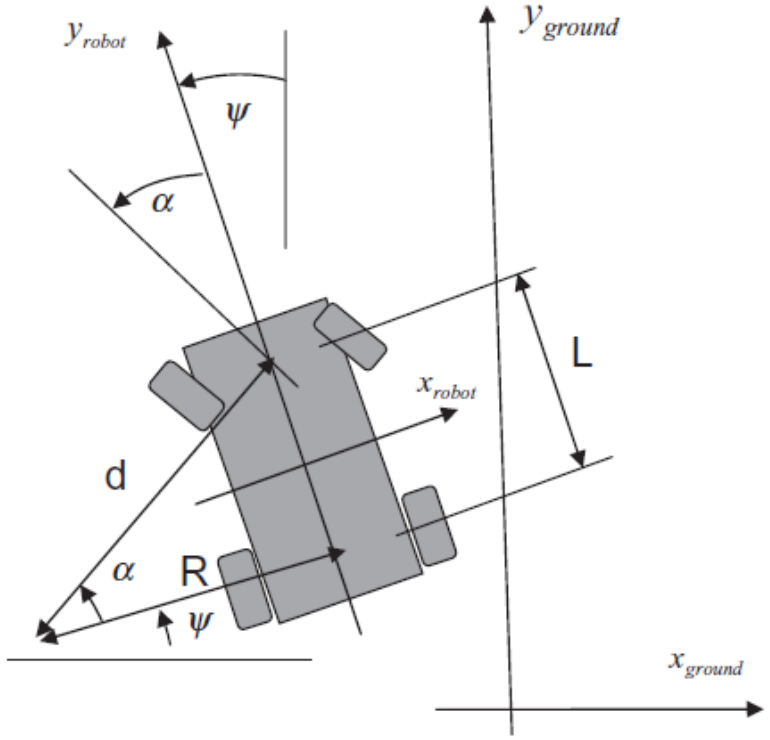


Rat Brain Robot.mp4

Retour sur le cours #12

Cours #12
Cinématique des robots mobiles: un autre exemple (5)

- Suite:
  - ***Image tirée de [1]*



- Vous avez donc obtenu le modèle cinématique du robot mobile:

$$\dot{\xi}_{ground} = \begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\psi}_R \end{bmatrix} = \begin{bmatrix} -\sin(\psi)\, v_{roues\_arr} \\ \cos(\psi)\, v_{roues\_arr} \\ \dfrac{v_{roues\_arr}}{L}\tan(\alpha) \end{bmatrix}$$

- **Quelles variables (variables d'entrées) proposeriez-vous pour contrôler ce véhicule?**
- **Notez qu'en fait, la seule variable sur laquelle vous ne pouvez pas directement agir est l'angle Ψ. Cependant, nous en avons besoin pour résoudre la cinématique!**
- **Encore une fois, nous pourrions penser à utiliser :**

$$\psi = \psi_0 + \int_{t_0}^{\sigma} \dot{\psi}(t)\, dt$$

- **Mais ce n'est certainement pas optimal!**
- **Une meilleure façon de faire serait d'utiliser un *filtre de Kalman* pour fusionner l'information des capteurs disponibles!**

Cours #12
Introduction au filtre de Kalman (1)

- Le filtre de Kalman, petite mise en contexte…

- **Rudolf E. Kálmán est récipiendaire de la *National Medal of Science***

Cours #12
Introduction au filtre de Kalman (3)

- **Maintenant, qu'est-ce que le filtre de Kalman et pourquoi est-t-il si répandu?**
  - Le filtre de Kalman est un algorithme qui utilise une série de mesures (affectées par du bruit blanc) effectuées au fil du temps, et produit des estimations de variables aléatoires qui sont optimales du point de vue statistique.
  - C'est un algorithme récursif qui permet de faire des prédictions (en temps réel) qui minimisent l'erreur au carré.
- **Bien entendu, ce genre de filtre est extrêmement utile dans un contexte de navigation autonome par des robots mobiles.**
  - Il permet entres autres de fusionner l'information provenant de différents capteurs de manière à produire une estimation (e.g. vitesse d'un véhicule) optimale.
  - Le développement du filtre de Kalman est complexe pour un seul cours, l'important est surtout de comprendre le principe.

Cours #12 – Filtre de Kalman: Estimation d'une quantité fixe à l'aide d'un lot de données (1)

- Sans démontrer rigoureusement tout le développement derrière le filtre de Kalman (un cours n'est pas suffisant), ce qui suit permettra du moins de survoler certains fondements derrière la théorie. *Référence: [1]*
  - La démarche sera *incrémentale* dans le sens où nous bâtirons, petit à petit, les principes qui mènent vers la construction du filtre de Kalman.
- Considérons pour commencer un ensemble de données qui sont en fait des mesures d'une certaine quantité fixe:

$$Y_K = Hx + V_K$$

Où Y_K est la mesure, x est la quantité à estimer et V_K est l'erreur

- Avec :

$$Y_K = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \end{bmatrix} \qquad V_K = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_k \end{bmatrix}$$

- L'hypothèse que l'on fait au niveau de l'erreur est que cette dernière est de **moyenne nulle** et:

$$\mathrm{cov}\left(V_K\right) = R$$

Cours #12 – Filtre de Kalman:
Estimation d'une quantité fixe à l'aide d'un lot de données (2)

- À partir des mesures Y_K, trouver la meilleur estimation de x. Pour ce faire, on veut trouver l'estimation de x qui minimise cet indicateur:

$$J = \left(Y_K - H\hat{x}_K\right)^T R^{-1} \left(Y_K - H\hat{x}_K\right)$$

- L'estimation de x qui minimise la valeur de cet indicateur est donnée par:

$$\boxed{\hat{x}_K = \left[H^T R^{-1} H\right]^{-1} H^T R^{-1} Y_K}$$

- Selon cette formulation, vous devez connaître la matrice de covariance des capteurs utilisés pour la mesure.

Cours #12 – Filtre de Kalman: Estimation d'une quantité fixe par un processus récursif (1)

- Maintenant, supposons que de nouvelles données deviennent disponibles et que nous souhaitons utiliser ces dernières afin de parfaire notre estimation de x.
  - Nous pourrions utiliser le même calcul, mais ce n'est pas efficace puisque cela implique de toujours ré-utiliser à chaque opération toutes les données!
  - Pour contrer ce problème, nous développons une méthode récursive.
- La mesure additionnelle est donnée par:

$$y_{K+1} = H_{K+1}x + v_{K+1}$$

- Encore une fois, nous cherchons la meilleure estimation de x que nous pouvons faire à partir des mesures y. Pour ce faire, nous chercherons à minimiser:

$$J = \begin{bmatrix} \left(Y_K - H\hat{x}_{K+1}\right)^T & \left(y_{K+1} - H_{K+1}\hat{x}_{K+1}\right)^T \end{bmatrix} \begin{bmatrix} R^{-1} & 0 \\ 0 & R_{K+1}^{-1} \end{bmatrix} \begin{bmatrix} Y_K - H\hat{x}_{K+1} \\ y_{K+1} - H_{K+1}\hat{x}_{K+1} \end{bmatrix}$$

- L'estimation de x qui minimise la valeur de cet indicateur est donné par:

$$\boxed{\hat{x}_{K+1} = \hat{x}_K + K_{K+1}\left[y_{K+1} - H_{K+1}\hat{x}_k\right]}$$

$$\text{Où: } K_{K+1} = P_K H_{K+1}^T \left[H_{K+1} P_K H_{K+1}^T + R_{K+1}\right]^{-1} \quad \text{et} \quad P_K = \left[H^T R^{-1} H\right]^{-1}$$

Cours #12 – Filtre de Kalman: Estimation d'une quantité fixe par un processus récursif (2)

- Re-visitons ce dernier résultat:

$$\boxed{\hat{x}_{K+1} = \hat{x}_K + K_{K+1}\left[y_{K+1} - H_{K+1}\hat{x}_k\right]}$$

- On remarque que le nouvel estimé de x dépend de l'ancienne valeur de l'estimation, plus une correction qui tient compte de la nouvelle mesure qui devient disponible.

- Aussi, re-visitons la fonction que nous avons minimisée pour trouver la meilleure estimation de x:

$$J = \begin{bmatrix} \left(Y_K - H\hat{x}_{K+1}\right)^T & \left(y_{K+1} - H_{K+1}\hat{x}_{K+1}\right)^T \end{bmatrix} \begin{bmatrix} R^{-1} & 0 \\ 0 & R_{K+1}^{-1} \end{bmatrix} \begin{bmatrix} Y_K - H\hat{x}_{K+1} \\ y_{K+1} - H_{K+1}\hat{x}_{K+1} \end{bmatrix}$$

- Le deuxième élément de J est encore l'erreur au carrée du nouvel estimé, pondéré par la matrice de covariance. Le premier élément est l'erreur au carré du nouvel estimé, basée sur les anciennes mesures.

Cours #12 – Filtre de Kalman: Estimation d'une quantité fixe par un processus récursif (3)

- Considérons un exemple simple:
  - Vous mesurer une distance à l'aide d'un télémètre laser qui prend plusieurs mesures.
  - La variance des mesures que donnent ce capteur est de 0.04
  - Comment pouvez-vous utiliser chacune des nouvelles mesures afin de produire un estimé optimal de la distance?


```
clc; clear all; close all;
R = 0.04;
P = R;
RealValue=5;
y(1) = RealValue + sqrt(R) * randn;
xest(1) = y(1); K(1)=NaN;
for i = 1:49
y(i + 1) = RealValue + sqrt(R) * randn;
P = 1 / ((1 / P) + (1 / R));
K(i+1) = P / (P + R);
xest(i + 1) = xest(i) + K(i+1) * (y(i + 1) - xest(i));
end

%Tracer le graphique:
figure
plot(RealValue*ones(1,numel(y)),['k']);
hold on
plot(y,['r' '.'])
hold on
plot(xest,['b' '--'],'linewidth',2)
legend('Valeur réelle','Mesures','Estimation')
```

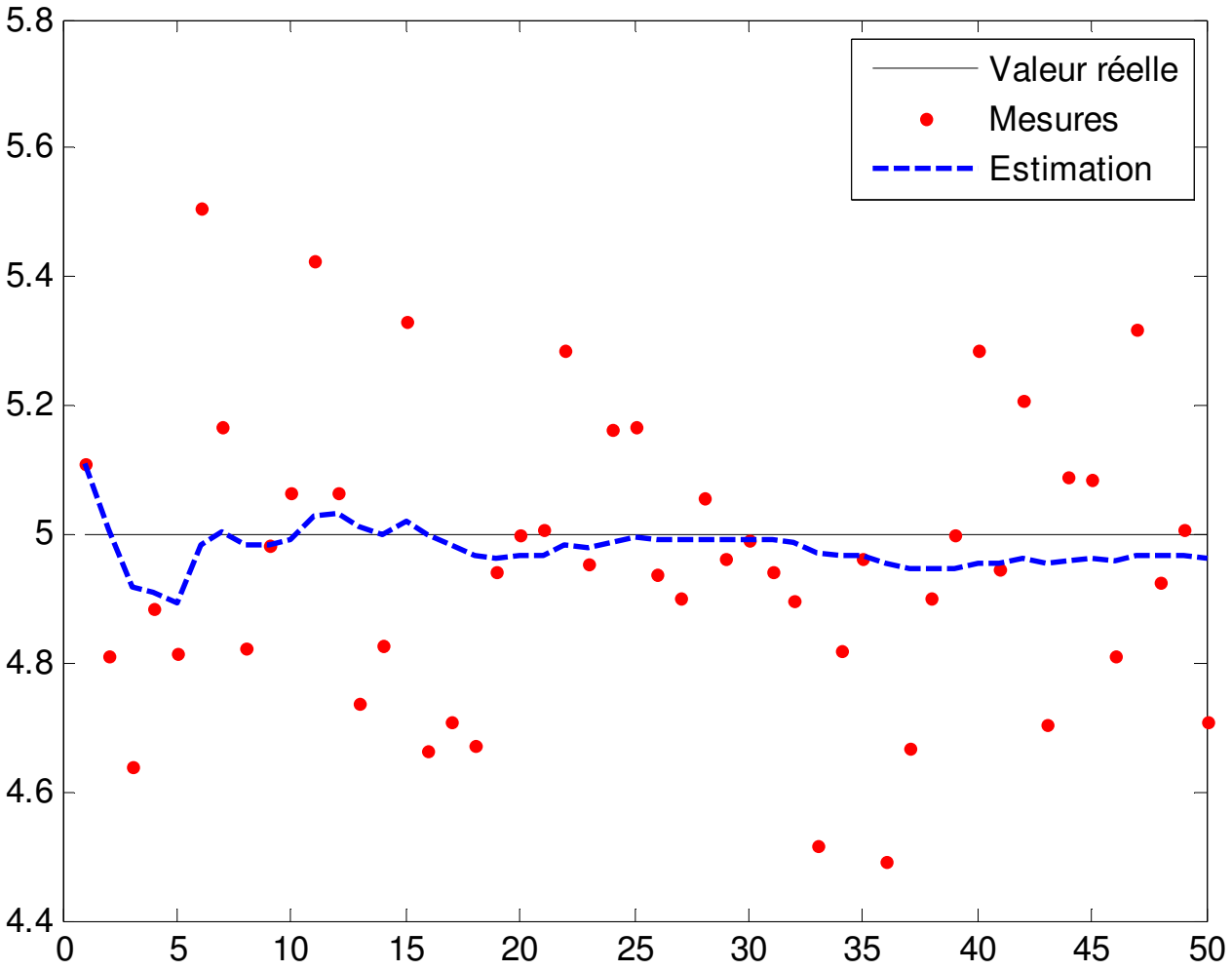
$$\hat{x}_{K+1} = \hat{x}_K + K_{K+1}\left[y_{K+1} - H_{K+1}\hat{x}_k\right]$$

$$K_{K+1} = P_K H_{K+1}^T \left[H_{K+1} P_K H_{K+1}^T + R_{K+1}\right]^{-1}$$

$$P_K = \left[H^T R^{-1} H\right]^{-1}$$

Cours #12 – Filtre de Kalman:
Estimation d'une quantité fixe par un processus récursif (4)

- **Résultats**:



Cours #12 – Filtre de Kalman (ref: [1]): Estimation des états d'un sys. dyn. par un processus récursif (1)

- Le filtre de Kalman est un algorithme qui permet d'estimer x (de manière récursive et assurant un résultat optimal) qui **n'est plus fixe**. On se sert souvent du filtre de Kalman afin d'estimer les états dynamiques d'un système.
- Soit un système représenté sous forme de modèle d'état:

$$X\left((k+1)T\right) = AX\left(kT\right) + Bu\left(kT\right) + Gw\left(kT\right)$$

$$Y\left((k+1)T\right) = HX\left((k+1)T\right) + v\left((k+1)T\right)$$

- Où w est une perturbation (bruit) affectant certains (ou tous) états du système: de moyenne nulle. v est l'erreur de mesure de moyenne nulle. De plus:

$$\mathrm{cov}\left(w\right) = Q \quad \text{et} \quad \mathrm{cov}\left(v\right) = R$$

- Aussi, introduisons la notation suivante:

Y_k signifie la sortie au temps $Y\left(kT\right)$

Y_{k+1} signifie la sortie au temps $Y\left((k+1)T\right)$

$\hat{X}_{k/k}$ signifie l'estimé de X au temps kT étant donné les mesures jusqu'au temps kT

$\hat{X}_{k+1/k}$ signifie l'estimé de X au temps $(k+1)T$ étant donné les mesures jusqu'au temps kT

Cours #12 – Filtre de Kalman (réf: [1]): Estimation des états d'un sys. dyn. par un processus récursif (2)

- Évidemment, on constate que le prochaine état estimé, basé sur la valeur actuelle de l'état est:

$$\hat{X}_{k+1/k} = A\hat{X}_{k/k} + Bu_k$$

- Si aucune mesure n'est disponible, ceci est le meilleur estimé que vous pouvez faire, il est basé sur la valeur précédente de l'état et le fait que l'espérance (valeur la plus probable) du bruit w est 0.
- Si une mesure devient disponible, Kalman a démontré que l'estimé optimal est donné par ce qu'on appelle aujourd'hui le ***filtre de Kalman***:

$$\boxed{\hat{X}_{k+1/k+1} = \hat{X}_{k+1/k} + K_{k+1}\left(Y_{k+1} - H\hat{X}_{k+1/k}\right)}$$

- Donc, l'estimé optimal est donné par la prédiction basée sur le modèle d'état, plus une correction, qui dépend d'un certain gain K:

$$K_{k+1} = P_{k+1/k}H^T\left[HP_{k+1/k}H^T + R\right]^{-1}$$

$$P_{k+1/k} = AP_{k/k}A^T + GQG^T$$

$$P_{k+1/k+1} = \left[I - K(k+1)H\right]P_{k+1/k}$$

Cours #12 – Filtre de Kalman (réf: [1]): Estimation des états d'un sys. dyn. par un processus récursif (3)

- Donc, pour illustrer ces notions, voici encore un petit exemple…
  - Soit un système dynamique discret qui régit le déplacement d'un véhicule:

$$x_1(k+1) = x_1(k) + Tx_2(k)$$

$$x_2(k+1) = x_2(k) + Tu(k) + w(k)$$

Où x_1 est la position et x_2 est la vitesse.

On mesure la position, donc l'équation des mesures:

$$y(k+1) = x_1(k+1) + v(k+1)$$

- Étant donné les mesures de position, nous souhaitons trouvé le meilleur estimé de la position du véhiculet **ET** de la vitesse. De plus, nous savons que la variance du capteur servant à mesuré la position est R=0.04 et la variance du signal qui perturbe le deuxième état est Q=0.00005. Nous considérerons aussi que le système sera excité par le signal d'entré u, seulement pour les dix premières périodes de temps T et que la valeur du signal d'entre sera alors u=0.25.

Cours #12 – Filtre de Kalman (réf: [1]): Estimation des états d'un sys. dyn. par un processus récursif (4)

$$x_1(k+1) = x_1(k) + Tx_2(k)$$

$$x_2(k+1) = x_2(k) + Tu(k) + w(k)$$

Où x_1 est la position et x_2 est la vitesse.

On mesure la position, donc l'équation des mesures:

$$y(k+1) = x_1(k+1) + v(k+1)$$

```
clear all; close all; clc;
nb_iteration=500;
T = 0.2;
A = [1 T; 0 1];
B = [0 T]';
H = [1 0];
G = [0 1]';
Q = 0.00005;
R = 0.02;
x1(1) = 0;
x2(1) = 0;
x1e(1) = 0;
x2e(1) = 0;
xest = [x1e(1) x2e(1)]';
x1p(1) = 0;
x2p(1) = 0;
PE = [R 0; 0 0];
PP = A * PE(1) * A' + Q;
```

$$\hat{X}_{k+1/k+1} = \hat{X}_{k+1/k} + K_{k+1}\left(Y_{k+1} - H\hat{X}_{k+1/k}\right)$$

$$K_{k+1} = P_{k+1/k}H^T\left[HP_{k+1/k}H^T + R\right]^{-1}$$

$$P_{k+1/k} = AP_{k/k}A^T + GQG^T$$

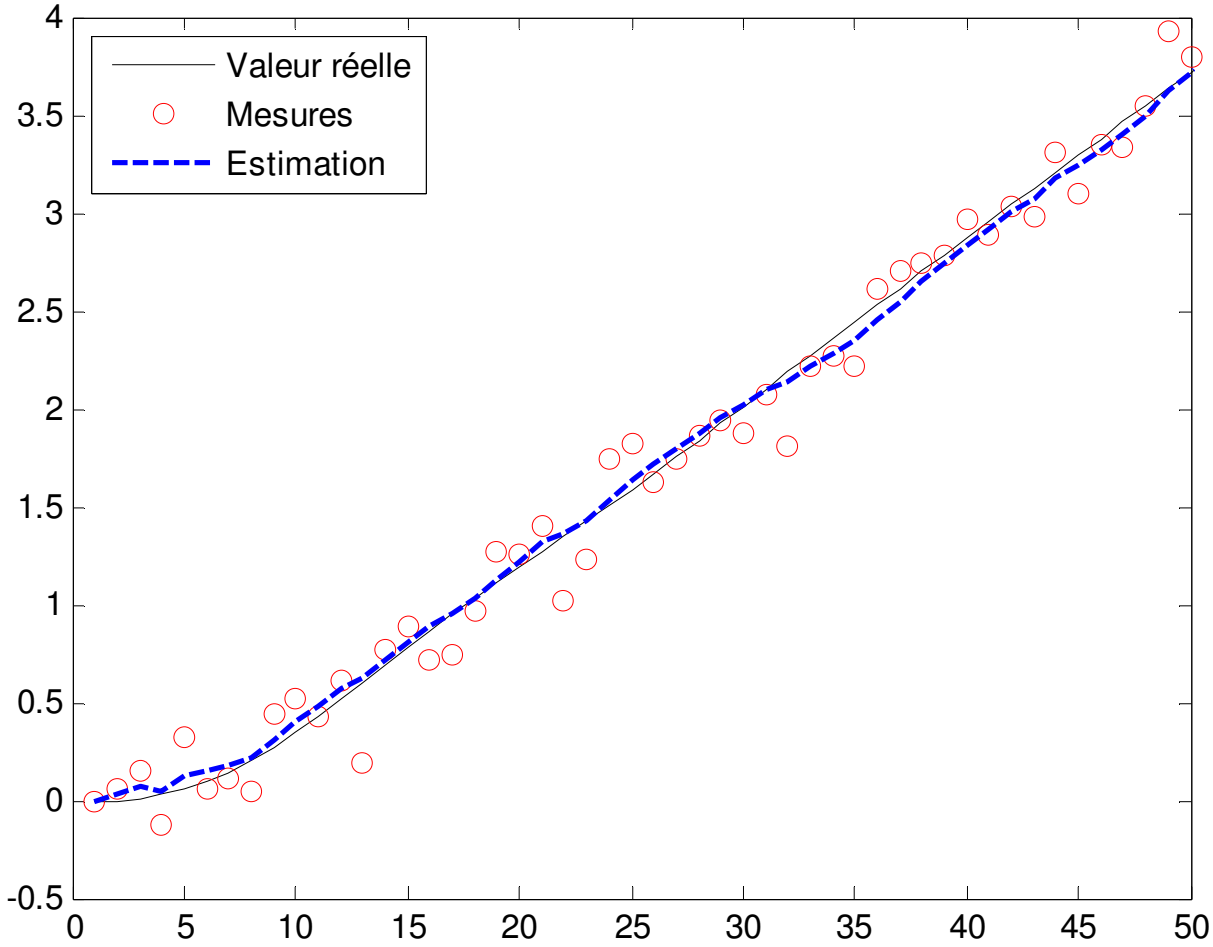
$$P_{k+1/k+1} = \left[I - K(k+1)H\right]P_{k+1/k}$$

- Étant donné les mesures de position, nous souhaitons trouvé le meilleur estimé de la position du véhiculet **ET** de la vitesse. De plus, nous savons que la variance du capteur servant à mesuré la position est R=0.04 et que la la variance du signal d'entré qui perturbe le deuxième état est Q=0.00005. Nous considérerons aussi que le système sera excité par le signal d'entré u, seulement pour les dix premières périodes de temps T et que la valeur du signal d'entre sera alors u=0.25.

```
for i = 1:nb_iteration
    if i < 10
        u = 0.25;
    else
        u = 0;
    end
    x1(i+1) = x1(i) + T * x2(i);
    x2(i+1) = x2(i) + T * u + sqrt(Q) * randn;
    y(i+1) = x1(i + 1) + sqrt(R) * randn;
    PP = A * PE * A' + G * Q * G' ;
    K = PP * H' * inv(H * PP * H' + R);
    PE = [eye(2) - K * H] * PP;
    xpredict = A * xest + B * u;
    xest = xpredict + K * (y(i + 1) - H * xpredict);
    x1e(i + 1) = [1 0] * xest;
    x2e(i + 1) = [0 1] * xest;
end
```

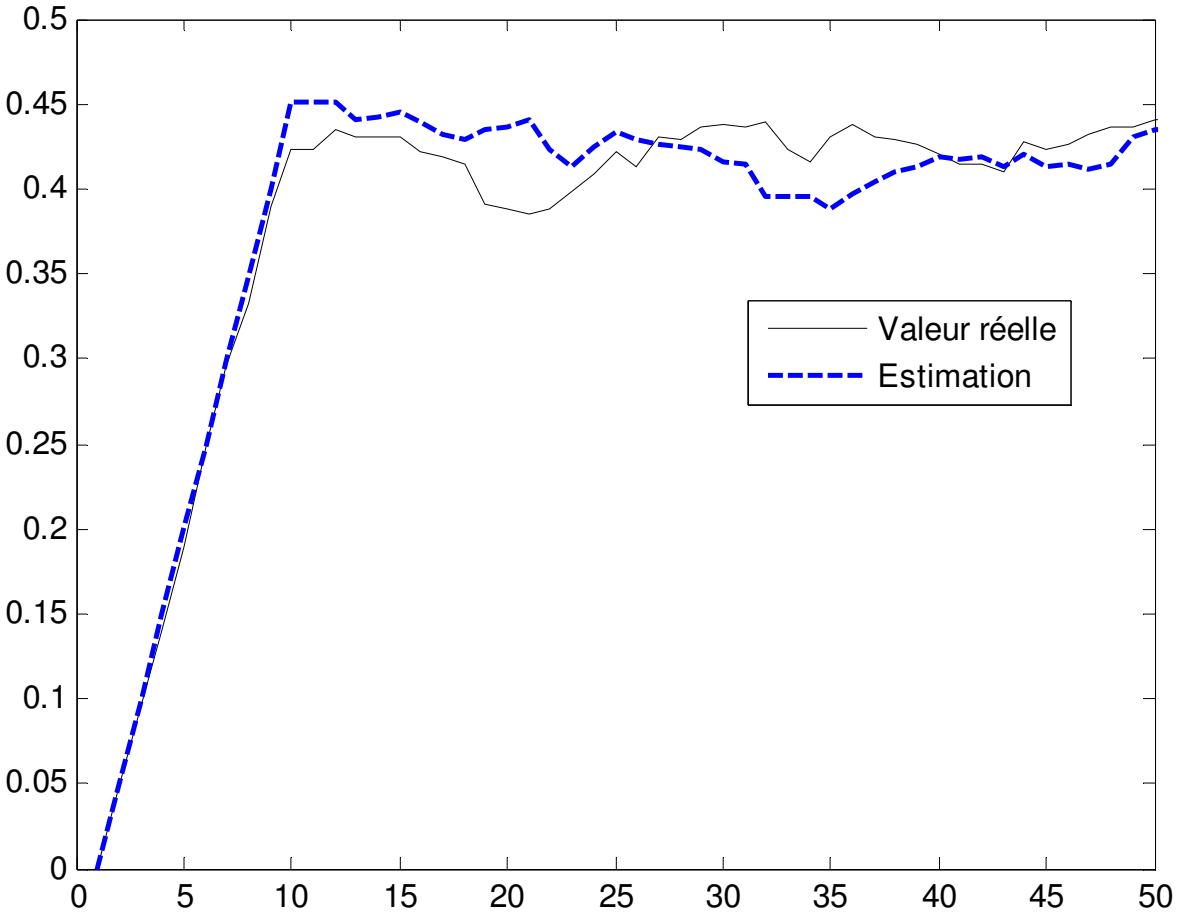
Cours #12 – Filtre de Kalman (réf: [1]): Estimation des états d'un sys. dyn. par un processus récursif (5)

- L'état 1, c'est-à-dire la position (Rappel: nous avions des mesures disponibles):



Cours #12 – Filtre de Kalman (réf: [1]): Estimation des états d'un sys. dyn. par un processus récursif (6)

- L'état 2, c'est-à-dire la vitesse (Rappel: nous **n'**avions **pas** de mesures disponibles):



Cours #12 – Filtre de Kalman (réf: [1]): Estimation des états d'un sys. dyn. par un processus récursif (7)

- Jusqu'à présent, nous avons considéré l'estimation d'états pour des systèmes linéaires invariants. Dans le cas d'un système non-linéaire, il est possible alors d'utiliser le **filtre Kalman étendu** (*extended Kalman filter*).
  - Soit un système non-linéaire:

$$\dot{X} = f(X,U) + Gw, \text{ sous forme discrétisé:}$$
$$X\big((k+1)T\big) = X(kT) + Tf\big(X(kT),U(kT)\big) + TGw(kT)$$
$$Y\big((k+1)T\big) = h\Big(X\big((k+1)T\big)\Big) + v\big((k+1)T\big)$$

  - Le meilleur estimé (c'est le filtre de Kalman étendu):

$$\hat{X}_{k+1/k+1} = \hat{X}_{k+1/k} + K_{k+1}\left(Y_{k+1} - h\left(\hat{X}_{k+1/k}\right)\right)$$
$$P_{k+1/k} = A(k)P_{k/k}A(k)^T + (GT)Q(GT)^T$$
$$P_{k+1/k+1} = \left[I - K_{k+1}H(k)\right]P_{k+1/k}$$
$$K_{k+1} = P_{k+1/k}H_k^T\left[H_kP_{k+1/k}H_k^T + R\right]^{-1}$$

Où:

$$A(k) = I + T\left[\frac{\partial f(X,U)}{\partial X}\right]_{\hat{X}_{k/k},U_k}$$
$$H(k) = \left[\frac{\partial h(X)}{\partial X}\right]_{\hat{X}_{k/k}}$$

Application du filtre de Kalman: Problème du décalage des gyroscopes (I)

- Petit historique; problème traité par plusieurs sources, entres autres Brown, Bona & Smay, Bar-Itzhack…

$$\dot{\Psi}_x = \Omega_z \Psi_y + \varepsilon_x$$
$$\dot{\Psi}_y = \Omega_x \Psi_z - \Omega_z \Psi_x + \varepsilon_y$$
$$\dot{\Psi}_z = -\Omega_x \Psi_y + \varepsilon_z$$

Ψ_x : Erreur de longitude du système de navigation inertielle

Ψ_y : Erreur de latitude du système de navigation inertielle

Ψ_z : $(\text{Erreur d'azimut de la platforme inertielle}) - (\text{Erreur de longitude}) \tan(\text{Latitude})$

$\Omega_x = \Omega \cos(\text{Latitude})$ $\Omega_z = \Omega \cos(\text{Latitude})$

$\varepsilon_x, \varepsilon_y, \varepsilon_z$: Décallage des gyroscopes ("Bias") pour l'axe x,y et z respectivement

- Problème: Le bruit (Epsilon) n'est pas un bruit blanc!

Application #3: Problème du décalage des gyroscopes (III)

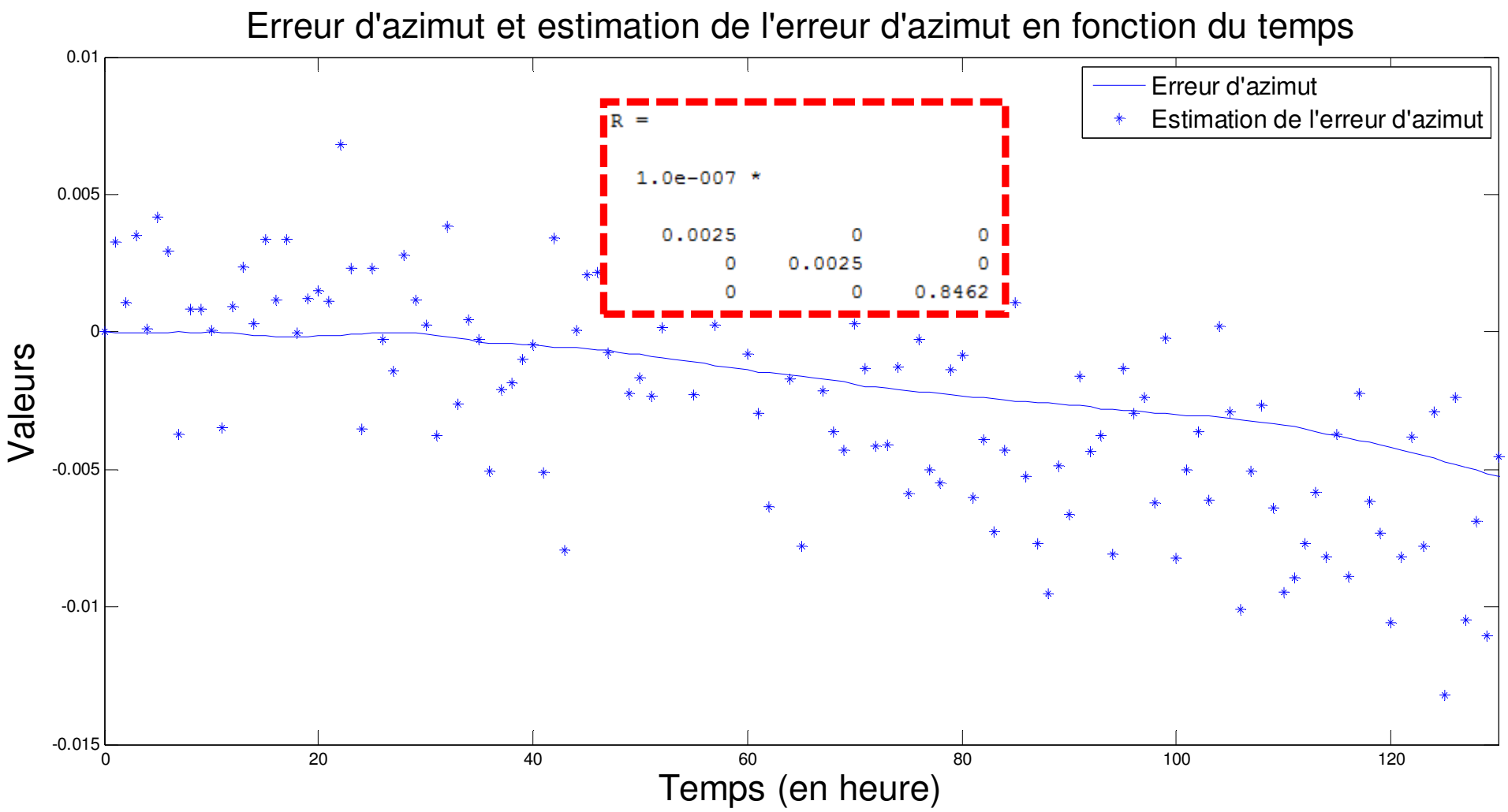
- **Solution** - Augmenter l'état du système de sorte à inclure le bruit dans l'état:

$$\dot{\varepsilon}_x = f_x$$
$$\dot{\varepsilon}_y = f_y$$
$$\dot{\varepsilon}_z = f_z$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & \Omega_z & 0 & 1 & 0 & 0 \\ -\Omega_z & 0 & \Omega_x & 0 & 1 & 0 \\ 0 & -\Omega_x & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{F} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{G} \begin{bmatrix} 0 \\ 0 \\ 0 \\ f_x \\ f_y \\ f_z \end{bmatrix}$$

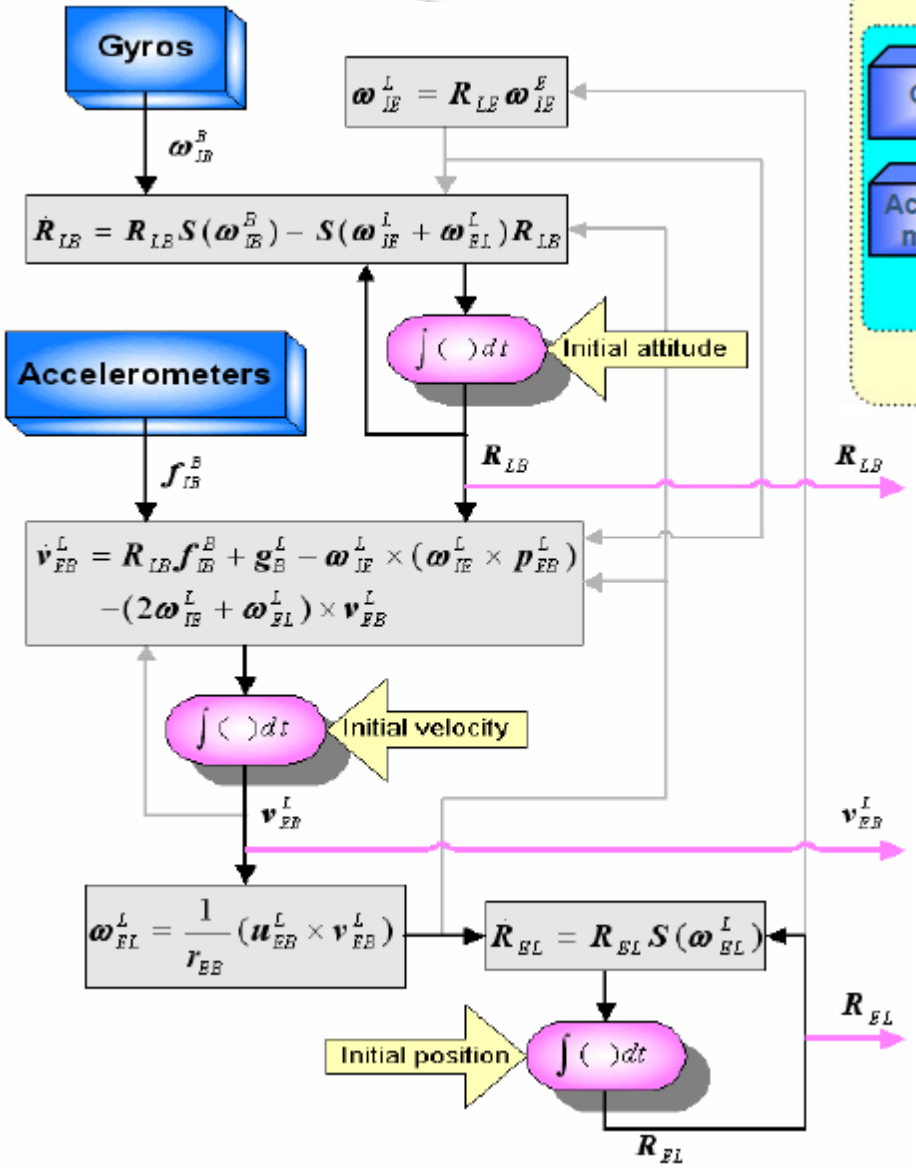
- On choisit les fonctions « f » de sorte que les « Epsilon » soient modélisés tels que des processus de Wiener-Brown indépedants : varient lentement.

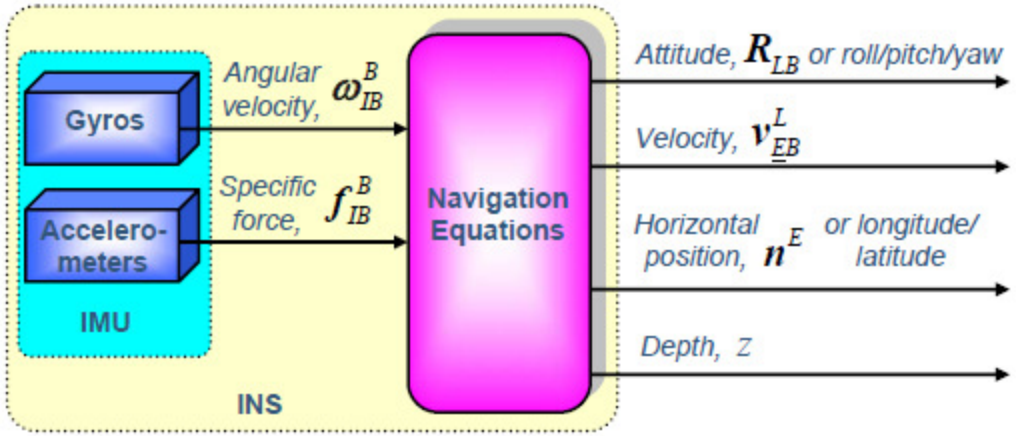
Décalage des gyroscopes: Résultats (I)

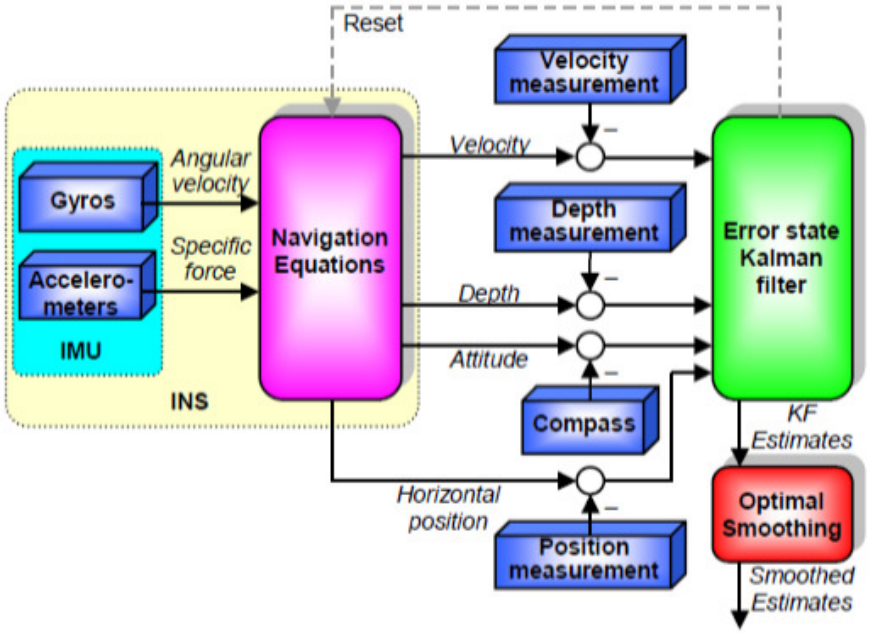


Cours #12 – Filtre de Kalman (réf: [1])

***images tirées de [5]*







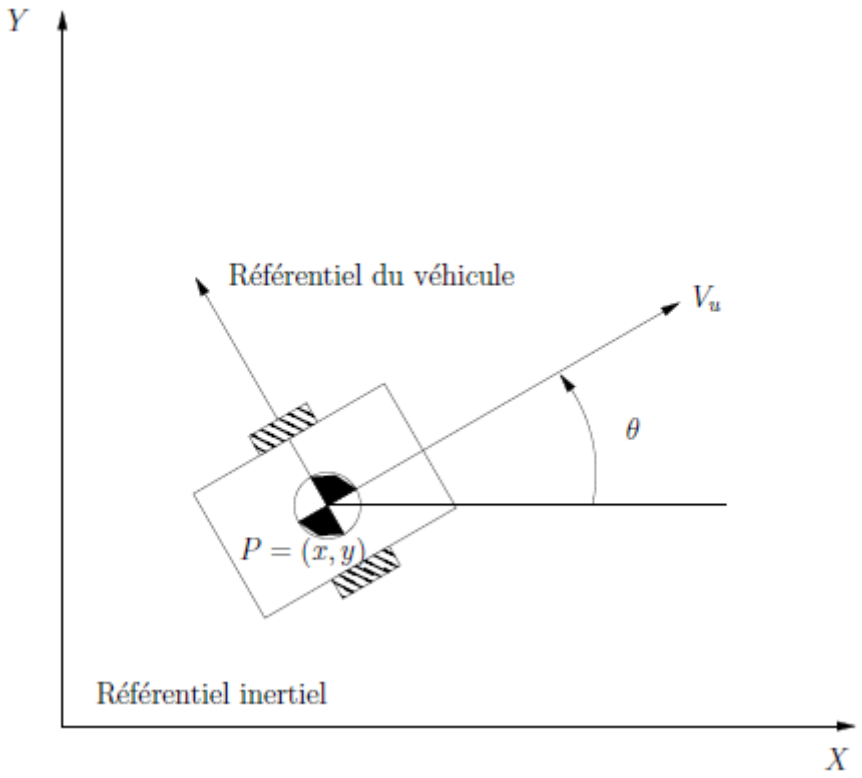
Cours #12
Contrôleur classique de vitesse et rotation (1)

- Puisque nous avons touché le thème de la cinématique et de la localisation, nous en sommes rendus au point où il est nécessaire de parler de contrôle.
  - Le problème auquel nous nous attaquons ici est sommarisé par: Comment synthétiser un signal d'entrée de manière à ce que le robot puisse suivre une trajectoire désirée?

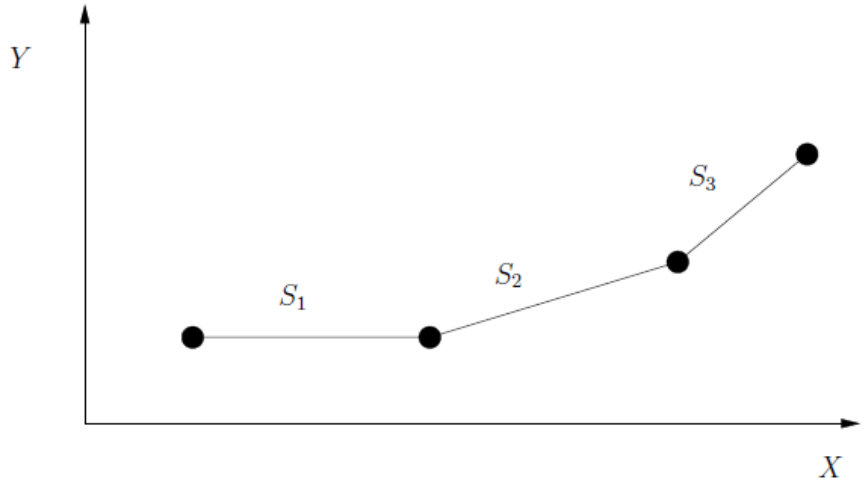
- Nous verrons deux approches répandues, et une troisième approche (fortement basée sur la deuxième) plus différente.

Cours #12
Contrôleur classique de vitesse et rotation (2)

- Considérons pour commencer la situation suivante:
  - On veut pouvoir faire suivre au robot ci-dessous des trajectoires pré-déterminées avec une erreur de suivi minimale.

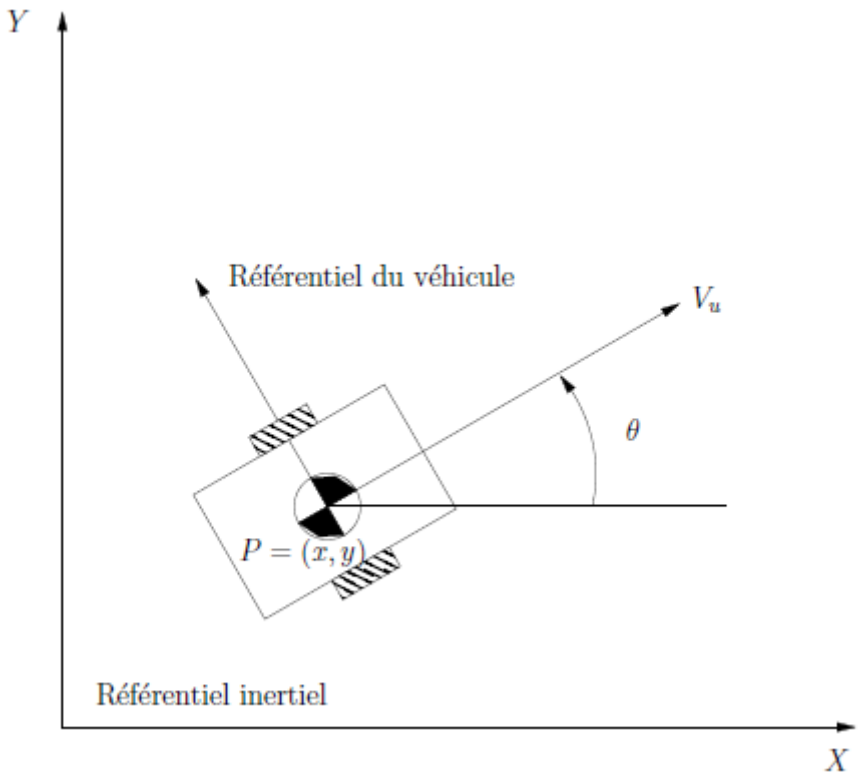


Exemple de trajectoires à suivre:



Cours #12
Contrôleur classique de vitesse et rotation (3)

- Considérons pour commencer la situation suivante:
  - On veut pouvoir faire suivre au robot ci-dessous des trajectoires pré-déterminées avec une erreur de suivi minimale.



- Tel que mentionné au cours #11:

$$\dot{x} = V_u \cos(\theta)$$
$$\dot{y} = V_u \sin(\theta)$$
$$\dot{\theta} = \dot{\theta}$$

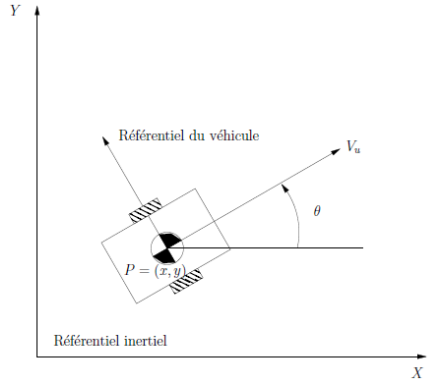
- On peut considérer la dynamique de ce système sous cette forme:

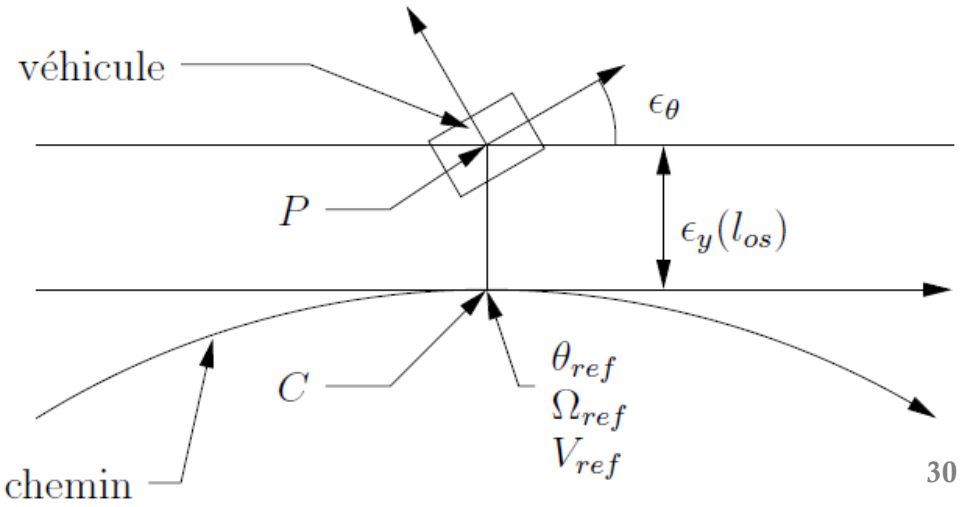
$$\frac{V_u}{U_m} = \frac{K_v}{s\tau_v + 1} \quad \text{et} \quad \frac{\Omega}{\Delta U} = \frac{K_o}{s\tau_o + 1}$$

où: $U_m = U_1 + U_2$ et $\Delta U = U_1 - U_2$

Cours #12
Contrôleur classique de vitesse et rotation (5)

- Dans le contexte du contrôle, on peut parler de différentes erreurs de suivi de trajectoire. Ici, on parle d'erreur de suivi en vitesse, en orientation et l'erreur de suivi latéral (position):





- Par simple géométrie on peut trouver l'expression de l'erreur latérale:

$$l_{os} = -(x - x_d)\sin(\theta_{ref}) + (y - y_d)\cos(\theta_{ref})$$

- En dérivant (voir vos notes de cours pour la démarche):

$$\dot{l}_{os} = -(\dot{x} - \dot{x}_d)\sin(\theta_{ref}) + (\dot{y} - \dot{y}_d)\cos(\theta_{ref})$$

- Soit la loi de commande suivante:

$$V_u^* = V_{ref}$$

$$\Omega^* = \Omega_{ref} - \frac{k_1}{V_{ref}} l_{os} - \frac{k_2}{V_{ref}} \dot{l}_{os}$$

Cours #12
Contrôleur classique de vitesse et rotation (6)

$$V_u^* = V_{ref} \quad \text{et} \quad \Omega^* = \Omega_{ref} - \frac{k_1}{V_{ref}} l_{os} - \frac{k_2}{V_{ref}} \dot{l}_{os}$$

$$\dot{l}_{os} = -(\dot{x} - \dot{x}_d)\sin(\theta_{ref}) + (\dot{y} - \dot{y}_d)\cos(\theta_{ref})$$

- En substituant: $\dot{x} = V_{ref}\cos(\theta) \quad \dot{y} = V_{ref}\sin(\theta) \quad \dot{x}_d = V_{ref}\cos(\theta_{ref}) \quad \dot{y}_d = V_{ref}\sin(\theta_{ref})$

- Dans l'équation ci-dessus, on trouve:

$$\begin{aligned} \dot{l}_{os} &= V_{ref}\left[-(\cos\theta - \cos(\theta_{ref}))\sin(\theta_{ref}) + (\sin\theta - \sin(\theta_{ref}))\cos(\theta_{ref})\right] \\ &= V_{ref}\left[-\cos\theta\sin(\theta_{ref}) + \sin\theta\cos(\theta_{ref})\right] \\ &= V_{ref}\sin(\theta - \theta_{ref}) \end{aligned}$$

- En dérivant une autre fois l'erreur de suivi latéral (pour faire apparaître la vitesse angulaire):

$$\begin{aligned} \ddot{l}_{os} &= V_{ref}\cos(\theta - \theta_{ref})(\Omega^* - \Omega_{ref}) \\ &= V_{ref}\cos(\theta - \theta_{ref})\left(-\frac{k_1}{V_{ref}} l_{os} - \frac{k_2}{V_{ref}} \dot{l}_{os}\right) \\ &= \cos(\theta - \theta_{ref})(-k_1 l_{os} - k_2 \dot{l}_{os}) \end{aligned}$$

- Pour une petite erreur d'angle, le facteur cosinus ≈1. On peut donc simplement déterminer les gains k_1 et k_2 par placement de pôles.

Cours #12
Contrôleur classique de vitesse et rotation (7)

- *Rappel:* Comment placer les pôles?
  - *Pour de petites erreurs d'angle le sinus≈0 et cosinus≈1 dans les équations ci-dessous:*

$$\begin{aligned} \dot{l}_{os} &= V_{ref}\left[-(\cos\theta - \cos(\theta_{ref}))\sin(\theta_{ref}) + (\sin\theta - \sin(\theta_{ref}))\cos(\theta_{ref})\right] \\ &= V_{ref}\left[-\cos\theta\sin(\theta_{ref}) + \sin\theta\cos(\theta_{ref})\right] \\ &= V_{ref}\sin(\theta - \theta_{ref}) \end{aligned}$$

$$\begin{aligned} \ddot{l}_{os} &= V_{ref}\cos(\theta - \theta_{ref})(\Omega^* - \Omega_{ref}) \\ &= V_{ref}\cos(\theta - \theta_{ref})(-\frac{k_1}{V_{ref}}l_{os} - \frac{k_2}{V_{ref}}\dot{l}_{os}) \\ &= \cos(\theta - \theta_{ref})(-k_1 l_{os} - k_2\dot{l}_{os}) \end{aligned}$$

- Donc: $\dot{l}_{os} = 0$ et $\ddot{l}_{os} = -k_1 l_{os} - k_2 \dot{l}_{os}$
- Sous forme de modèle d'états:

$$\underbrace{\begin{bmatrix} \dot{l}_{os} \\ \ddot{l}_{os} \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} 0 & 1 \\ -k_1 & -k_2 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} l_{os} \\ \dot{l}_{os} \end{bmatrix}}_{x}$$

- Pour assurer la stabilité (ici que l'erreur de suivi latérale soit asymptotiquement stable, c'est-à-dire qu'elle converge vers 0 lorsque le temps tend vers l'infini), on doit choisir les valeurs propres de la matrice A de manière à ce qu'elles soient à partie réelle négative.

Cours #12
Contrôleur classique de vitesse et rotation (8)

- *Rappel(suite):*
  - *Les valeurs propres de A sont données par les racines du polynôme caractéristique:*

$$P(\lambda) = \det(\lambda I - A) = \det\left(\begin{bmatrix} \lambda & -1 \\ k_1 & \lambda + k_2 \end{bmatrix}\right) = \lambda^2 + k_2\lambda + k_1$$

- Donc, pour assurer la convergence de l'erreur latéral, n'importe quel gain k_1, k_2>0 assurera que les valeurs propres sont à partie réelle négative.

Cours #12
Contrôleur classique de vitesse et rotation (9)

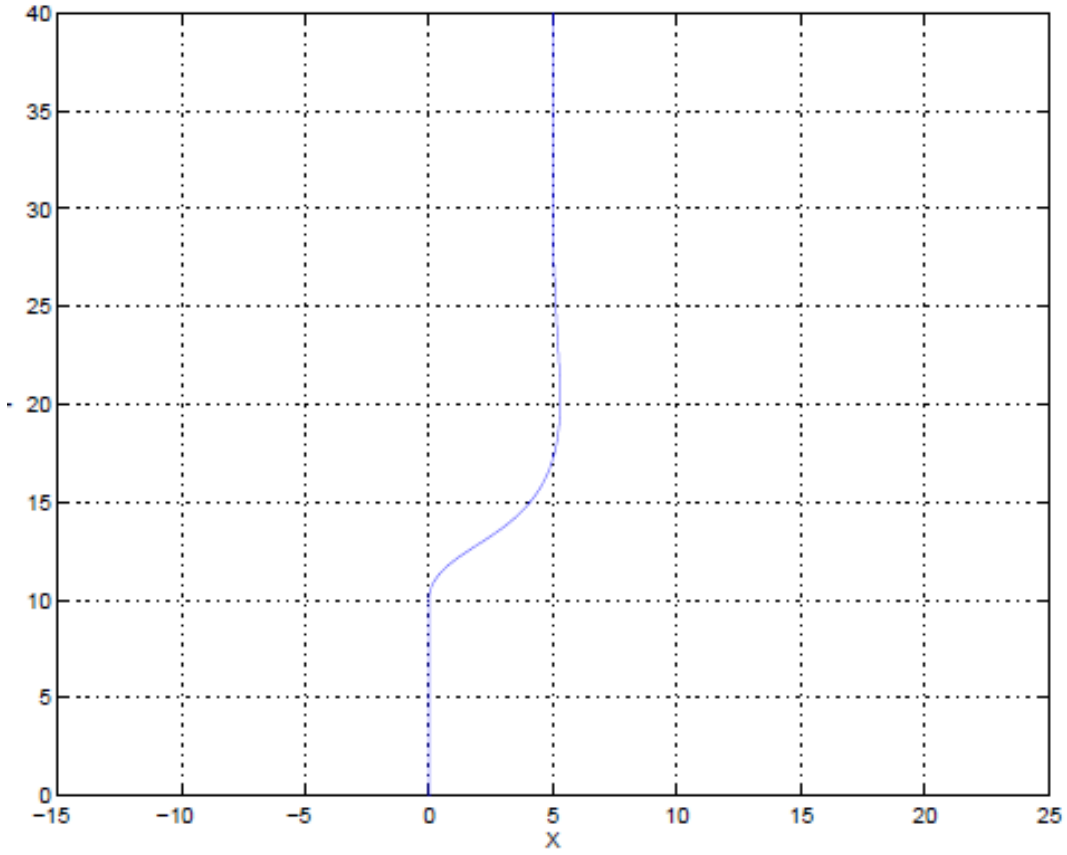
- On rajoute finalement un contrôleur de vitesse pour assurer le suivi de vitesse désiré:

$$U_m = \frac{1}{K_v}\left[V_u + K_p\tau_v(V_u^* - V_u) + K_I\tau_v\int(V_u^* - V_u)dt\right]$$

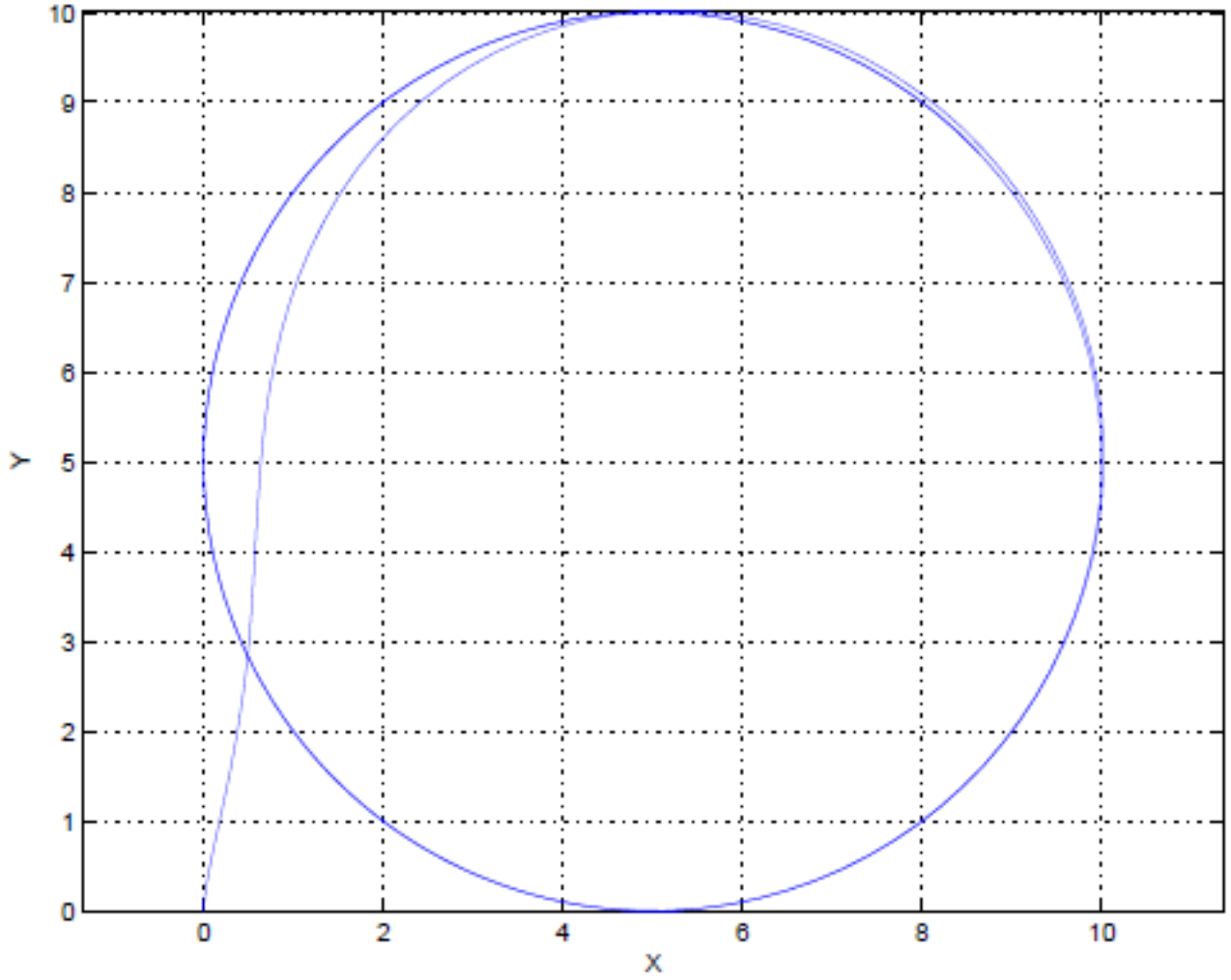
- Et une loi de commande pour la vitesse en rotation:

$$\Delta U = \frac{1}{K_o}[\Omega + K_\theta\tau_o(\Omega^* - \Omega)]$$

Cours #12
Contrôleur classique de vitesse et rotation (10)



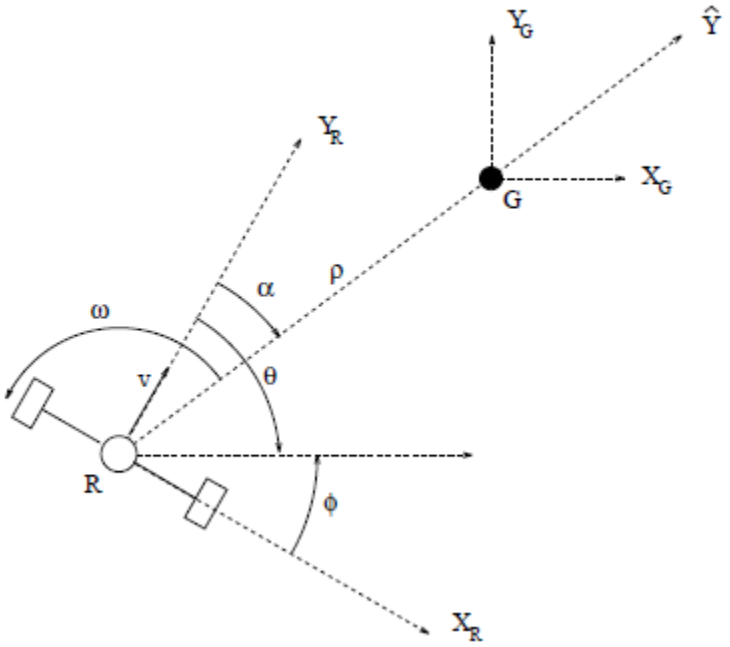
Cours #12
Contrôleur classique de vitesse et rotation (11)



Cours #12
Contrôleur d'Astolfi (voir [2]) (1)

- Depuis le début de notre étude sur la cinématique des robots, nous avons décrit la cinématique en coordonnées cartésiennes.
  - Voici un contrôleur qui considère plutôt une cinématique énoncée en coordonnées sphériques:

- **Considérons le robot ci-bas:**



- Sa cinématique (coordonnées cartésiennes) est donnée par :

$$\dot{x} = \cos(\theta)v$$
$$\dot{y} = \sin(\theta)v$$
$$\dot{\theta} = \omega$$

- Considérons le changement de variable suivant:

$$\rho = \sqrt{x^2 + y^2}$$
$$\alpha = -\theta + \arctan\left(\frac{-y}{-x}\right) \quad \mod\left(\frac{\pi}{2}\right)$$
$$\phi = \frac{\pi}{2} - \theta$$

Cours #12
Contrôleur d'Astolfi (voir [2]) (2)

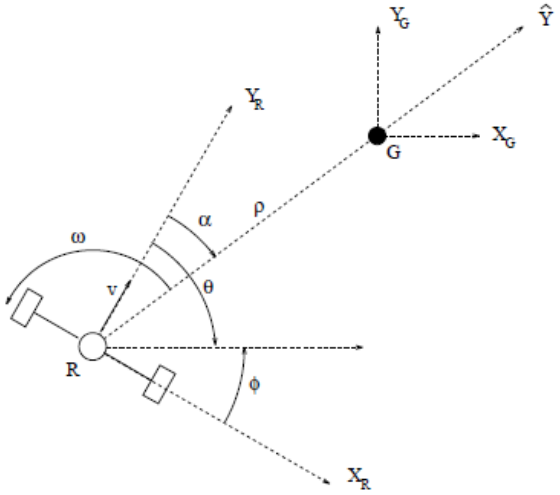
- On pourrait alors démontré (assez facilement) que, dans ce nouveau système de coordonnées polaires, la cinématique est donné par:

$$\dot{\rho} = -\cos(\alpha)v$$
$$\dot{\alpha} = \frac{\sin\alpha}{\rho}v - \omega$$
$$\dot{\phi} = -\omega$$

- **Quelles sont les variables d'entrées de ce systèmes d'après vous? Sur quelles variables seraient-ils logique d'agir pour exciter ce système?**

- **Considérons la loi de commande suivante:**

$$v = k_p\rho$$
$$\omega = k_\alpha\alpha + k_\phi\phi$$



- **La cinématique du robot mobile est alors donnée par:**

$$\dot{\rho} = -k_\rho\rho\cos(\alpha)$$
$$\dot{\alpha} = k_\rho\sin\alpha - k_\alpha\alpha - k_\phi\phi$$
$$\dot{\phi} = -k_\alpha\alpha - k_\phi\phi$$

Cours #12
Contrôleur d'Astolfi (voir [2]) (3)

- Cette dynamique est non-linéaire, on peut toutefois linéariser ce système autour du point d'équilibre: $(\alpha = 0)$
- On obtiendra un modèle linéarisé qui représente la cinématique du robot autour des points d'équilibre dans une certaine région de cet espace 3D appellée la "région d'attraction". Donc, le modèle linéarisé donnera des bons résultats à l'intérieur de cette région d'attraction.
  - Une fois linéarisé, on obtient le modèle d'état suivant:

$$\underbrace{\begin{bmatrix}\dot{\rho}\\ \dot{\alpha}\\ \dot{\phi}\end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix}-k_\rho & 0 & 0\\ 0 & -(k_\alpha - k_\rho) & -k_\phi\\ 0 & -k_\alpha & -k_\phi\end{bmatrix}}_{A}\underbrace{\begin{bmatrix}\rho\\ \alpha\\ \phi\end{bmatrix}}_{x}$$

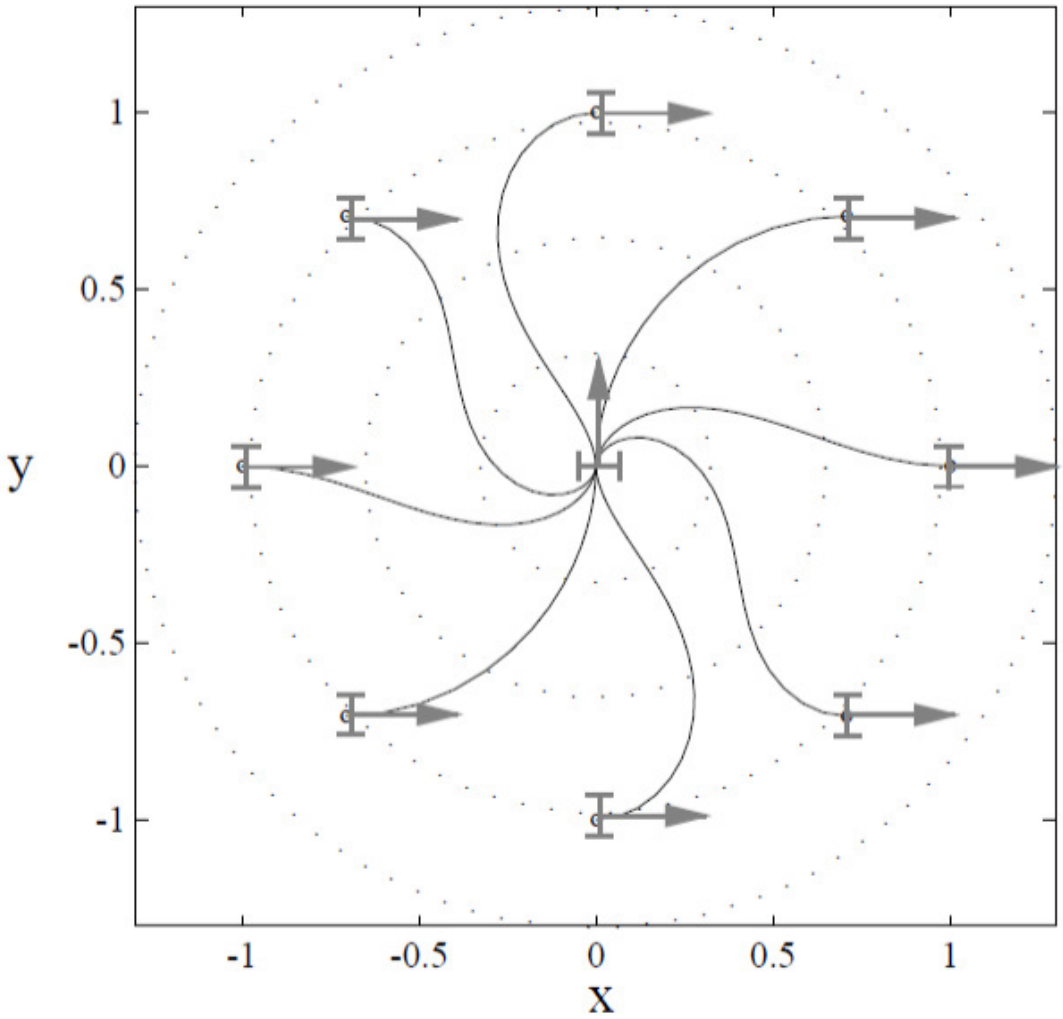
- Les valeurs propres de la matrice A (qui détermine la stabilité du système) sont données par les racines du polynôme caractéristique:

$$P(\lambda) = \det(\lambda I - A) = (\lambda + k_p)\left(\lambda^2 + \lambda(k_\alpha + k_\phi - k_p) - k_\rho k_\phi\right)$$

- Les valeurs propres seront à partie réelle négative ssi:

$$\begin{aligned} k_\rho &> 0\\ k_\phi &< 0\\ k_\alpha + k_\phi - k_\rho &> 0. \end{aligned}$$

Cours #12
Contrôleur d'Astolfi (voir [2]) (4)



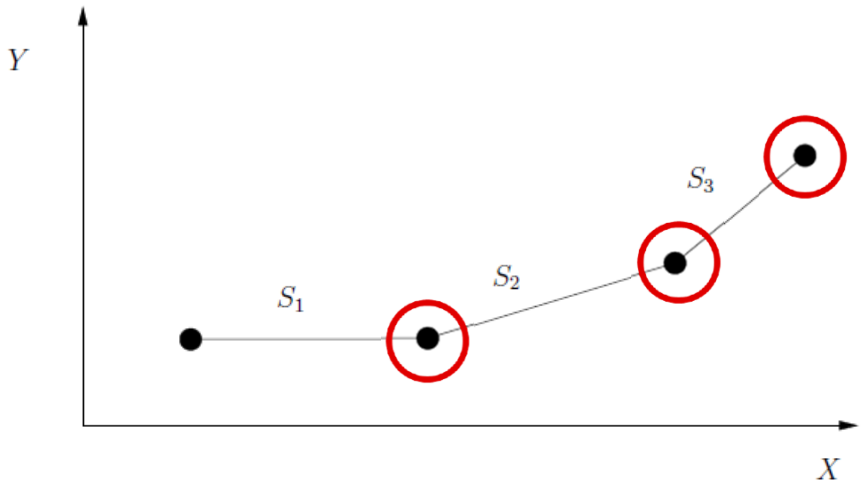
Cours #12
Contrôleur d'Astolfi: un autre contrôleur qui lui ressemble (1)

- Un problème avec le contrôleur d'Astolfi est que plus le robot mobile se rapproche de sa destination, plus ce dernier ralentit. En fait, sa vitesse décroit linéairement en fonction de la distance euclidienne:

$$v = k_\rho \rho$$

$$\omega = k_\alpha \alpha + k_\phi \phi$$

- Une solution pourrait être de déterminer des régions entourant les positions à atteindre à l'intérieur desquelles on considère que le robot a atteint sa destination:



- On aura quand même le même phénomène de vitesse qui décroit/croit drastiquement/décroit/croit drastiquement, etc…
- **Pour régler ce problème, il faut pousser plus loin notre étude du contrôleur d'Astolfi.**

Cours #12
Contrôleur d'Astolfi: un autre contrôleur qui lui ressemble (2)

- Rapellez-vous qu'avec la loi de commande proposé par Astolfi, le système linéarisé s'exprimait sous cette forme:

$$\underbrace{\begin{bmatrix}\dot{\rho}\\ \dot{\alpha}\\ \dot{\phi}\end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix}-k_\rho & 0 & 0\\ 0 & -(k_\alpha - k_\rho) & -k_\phi\\ 0 & -k_\alpha & -k_\phi\end{bmatrix}}_{A} \underbrace{\begin{bmatrix}\rho\\ \alpha\\ \phi\end{bmatrix}}_{x}$$

- Une particularité de ce système linéaire d'ordre 3 est que son modèle d'état est bloc-diagonal, il peut donc être découplé en deux sous-systèmes (un d'ordre 1 et l'autre d'ordre 2).

- **Autre point:** On connait bien les systèmes d'ordre 2, nous connaissons la *forme standard normalisée* des systèmes d'ordre 2:

$$G(s) = \frac{K\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

- **Il pourrait être intéressant de choisir le dénominateur de cette fonction de transfert (i.e. le polynôme caractéristique du deuxième sous-système), de manière à ce que la dynamique angulaire n'oscille pas! (i.e. ζ >= 1)**

Cours #12
Contrôleur d'Astolfi: un autre contrôleur qui lui ressemble (3)

$$\underbrace{\begin{bmatrix}\dot{\rho}\\ \dot{\alpha}\\ \dot{\phi}\end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix}-k_\rho & 0 & 0\\ 0 & -(k_\alpha - k_\rho) & -k_\phi\\ 0 & -k_\alpha & -k_\phi\end{bmatrix}}_{A} \underbrace{\begin{bmatrix}\rho\\ \alpha\\ \phi\end{bmatrix}}_{x}$$

$$G(s) = \frac{K\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} = \frac{Q(s)}{P(s)}$$

- Le polynôme caractéristique du deuxième sous-système (angles) est donné par:

$$P(s) = s^2 + (k_\alpha + k_\phi - k_p)s - k_\rho k_\phi$$

- On a donc:

$$\omega_n = \sqrt{-k_\rho k_\phi} \quad \text{et} \quad 2\zeta\omega_n = k_\alpha + k_\phi - k_p$$

- En substituant pour trouver "zeta":

$$2\zeta\sqrt{-k_\rho k_\phi} = k_\alpha + k_\phi - k_p \quad \Rightarrow \quad \zeta = \frac{k_\alpha + k_\phi - k_p}{2\sqrt{-k_\rho k_\phi}} \geq 1$$

- Donc en plus des critères précédents, on rajoute le critère ci-dessus pour s'assurer que la dynamique angulaire du robot ne soit pas sous-amortie (oscillante en régime transitoire).

Cours #12
Contrôleur d'Astolfi: un autre contrôleur qui lui ressemble (4)

- Donc, le contrôleur n'a pas encore changé de forme, nous avons simplement rajouté un critère à l'ensemble des critères existants:

$$v = k_p \rho$$
$$\omega = k_\alpha \alpha + k_\phi \phi$$

$$\begin{aligned} k_\rho &> 0 \\ k_\phi &< 0 \\ k_\alpha + k_\phi - k_\rho &> 0. \end{aligned}$$

critère supplémentaire: $\frac{k_\alpha + k_\phi - k_p}{2\sqrt{-k_\rho k_\phi}} \geq 1$

Le critère supplémentaire implique le troisième critère d'Astolfi, on peut donc le remplacer

- Soit v_{max} la vitesse de croisière que l'on souhaite avoir pour le robot, et le fait que la dynamique de la vitesse est maintenant simplement un système de premier ordre:

$$\dot{\rho} = -k_\rho \rho \quad \text{et} \quad v = k_\rho \rho$$

- On peut faire en sorte que la vitesse sature à la vitesse max (ou utiliser une saturation) jusqu'à une certaine distance de la trajectoire en choisissant un gain Kp élevé approprié, par exemple, pour une vitesse maximale en pratique de 1,5m/s si on choisi Kp=3:

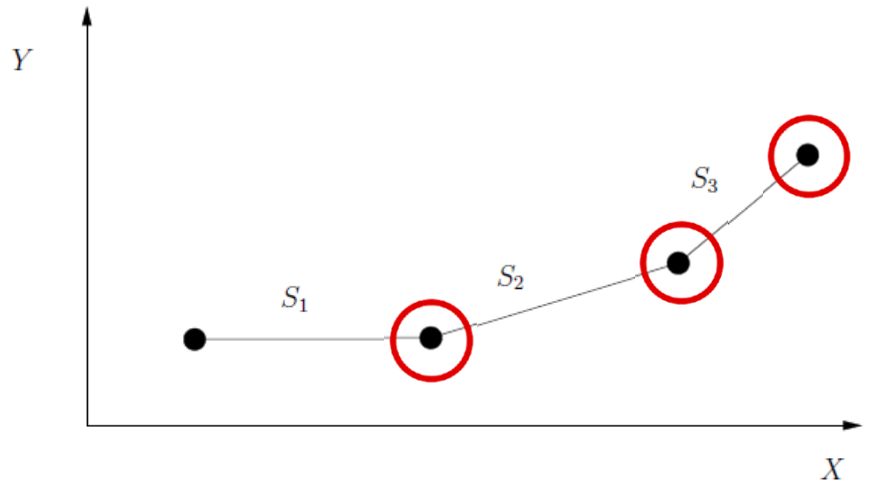
$$v = k_\rho \rho$$
$$v = 3\rho \quad \text{et} \quad 3\rho \geq v_{max} \forall \rho \geq 0{,}5m$$

- En gros, vous pouvez choisir le gain de cette façon:

$$k_\rho = \frac{v_{max}}{\rho_{limite}}$$

Cours #12
Contrôleur d'Astolfi: un autre contrôleur qui lui ressemble (5)

- Dans l'exemple précédent, pour des distances plus petite que 0,5m, le robot mobile ralentira. Si on ne veut jamais subir de décélération, on peut choisir un gain encore plus élevé et/ou combiner l'effet d'un gain élevé avec des régions plus permissives:



- Par contre, si on tolère une dynamique trop rapide en vitesse linéaire, on risque de détériorer la dynamique plus lente de la dynamique angulaire.
  - **Avez-vous des pistes de solutions?**

Cours #12
Contrôleur d'Astolfi: un autre contrôleur qui lui ressemble (6)

- On peut choisir la solution facile qui consiste à choisir les gains de la dynamique angulaire de sorte à ce que la réponse angulaire soit rapide, mais cela est déjà partiellement pris en compte par les critères qu'énonce Astolfi:

$$\begin{aligned} k_\rho &> 0 \\ k_\phi &< 0 \\ k_\alpha + k_\phi - k_\rho &> 0. \end{aligned}$$

critère supplémentaire: $\frac{k_\alpha + k_\phi - k_p}{2\sqrt{-k_\rho k_\phi}} \geq 1$

- On pourrait modifier la loi de commande et bâtir un nouveau contrôleur fortement inspiré du contrôleur d'Astolfi, et qui possèderait cette forme:

$$v = k_p\rho - k_1\alpha - k_2\phi \quad \text{avec} \quad k_{1,2} > 0$$
$$\omega = k_\alpha\alpha + k_\phi\phi$$

- On pourrait démontrer qu'il est possible de trouver des gains k_1 et k_2 qui stabilisent le système linéaire et qui permet donc de moduler la vitesse linéaire en fonction de l'erreur angulaire.

Cours #13:

Révision

Cours #13
Révision (1)

- Pour ce cours, le concept sera:
  - Dans un premier temps de rappeler la matière que nous avons vu et qui est sujette à examen.
  - Ensuite, divers exercices seront présentés: Par thème, au moins un exercice sera résolu au tableau et du temps sera ensuite accordé pour au moins un autre exercice sera à faire individuellement (nous le corrigerons ensemble).

- La matière à l'examen est détaillée dans le PDF que je vous ai envoyé. La présentation d'aujourd'hui est un rappel des majeures parties de cette matière.

Adobe Acrobat
Document

Révision
Dynamique (1)

- La dynamique d'un robot manipulateur peut être développé à partir de la méthode d'Euler-Lagrange:

$$\boldsymbol{\tau} = \frac{d}{dt}\left(\frac{\partial L(\boldsymbol{q},\dot{\boldsymbol{q}})}{\partial \dot{\boldsymbol{q}}}\right) - \frac{\partial L(\boldsymbol{q},\dot{\boldsymbol{q}})}{\partial \boldsymbol{q}}$$

où $L(\boldsymbol{q},\dot{\boldsymbol{q}}) = T(\boldsymbol{q},\dot{\boldsymbol{q}}) - U(\boldsymbol{q})$ est le lagrangien, $T(\boldsymbol{q},\dot{\boldsymbol{q}})$ l'énergie cinétique totale, $U(\boldsymbol{q})$ l'énergie potentielle, $\boldsymbol{q}$ le vecteur des positions aux articulations (coordonnées généralisées) et $\boldsymbol{\tau}$ le vecteur des couples aux articulations (forces généralisées).

L'application de l'équation de Lagrange au robot manipulateur résulte en une équation de la forme :

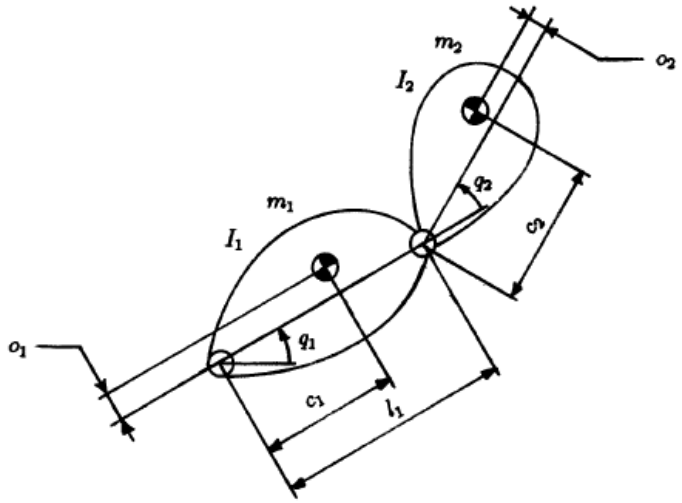
$$\boldsymbol{\tau} = \boldsymbol{D}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{h}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q})$$

S'il y a présence de friction dans les joints, on peut en tenir compte en introduisant un terme de friction $\boldsymbol{b}(\dot{\boldsymbol{q}})$ pour obtenir

$$\boldsymbol{\tau} = \boldsymbol{D}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{h}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{b}(\dot{\boldsymbol{q}})$$

Révision
Dynamique (2)

- Soit un robot RR à deux degrés de liberté:



L'énergie cinétique des deux segments est donnée par

$$T_1 = \frac{1}{2}(m_1 v_1^2 + I_1 \dot{q}_1^2)$$

$$T_2 = \frac{1}{2}(m_2 v_2^2 + I_2[\dot{q}_1 + \dot{q}_2]^2)$$

où I_1 et I_2 sont les inerties des segments 1 et 2 par rapport à leur centre de masse, v_1 et v_2 sont les vitesses de ces centres de masse.

Révision
Dynamique (3)

Les positions des centres de masse des segments 1 et 2 sont donnés par

$$
\begin{aligned}
x_1 &= c_1 \cos q_1 - o_1 \sin q_1 \\
y_1 &= c_1 \sin q_1 + o_1 \cos q_1 \\
x_2 &= l_1 \cos q_1 + c_2 \cos(q_1 + q_2) - o_2 \sin(q_1 + q_2) \\
y_2 &= l_1 \sin q_1 + c_2 \sin(q_1 + q_2) + o_2 \cos(q_1 + q_2)
\end{aligned}
$$

et leur vitesse par

$$
\begin{aligned}
\dot{x}_1 &= -(c_1 \sin q_1 + o_1 \cos q_1)\dot{q}_1 \\
\dot{y}_1 &= (c_1 \cos q_1 - o_1 \sin q_1)\dot{q}_1 \\
v_1^2 &= (c_1^2 + o_1^2)\dot{q}_1^2 \\
\dot{x}_2 &= -l_1 \sin(q_1)\dot{q}_1 - [c_2 \sin(q_1 + q_2) + o_2 \cos(q_1 + q_2)](\dot{q}_1 + \dot{q}_2) \\
\dot{y}_2 &= l_1 \cos q_1 \dot{q}_1 + [c_2 \cos(q_1 + q_2) - o_2 \sin(q_1 + q_2)](\dot{q}_1 + \dot{q}_2) \\
v_2^2 &= l_1^2 \dot{q}_1^2 + (c_2^2 + o_2^2)(\dot{q}_1 + \dot{q}_2)^2 \\
&\quad + 2l_1(c_2 \cos q_2 - o_2 \sin q_2)\dot{q}_1(\dot{q}_1 + \dot{q}_2)
\end{aligned}
$$

Révision
Dynamique (4)

$$v_1^2 = (c_1^2 + o_1^2)\dot{q}_1^2$$

$$\begin{aligned} v_2^2 &= l_1^2\dot{q}_1^2 + (c_2^2 + o_2^2)(\dot{q}_1 + \dot{q}_2)^2 \\ &\quad + 2l_1(c_2\cos q_2 - o_2 \sin q_2)\dot{q}_1(\dot{q}_1 + \dot{q}_2) \end{aligned}$$

L'énergie cinétique des segments est donnée par

$$\begin{aligned} T_1 &= \frac{1}{2}[m_1(c_1^2 + o_1^2) + I_1]\dot{q}_1^2 \\ T_2 &= \frac{1}{2}m_2 l_1^2\dot{q}_1^2 + \frac{1}{2}[m_2(c_2^2 + o_2^2) + I_2](\dot{q}_1 + \dot{q}_2)^2 \\ &\quad + m_2 l_1 c_2 \cos(q_2)\dot{q}_1(\dot{q}_1 + \dot{q}_2) - m_2 l_1 o_2 \sin(q_2)\dot{q}_1(\dot{q}_1 + \dot{q}_2) \end{aligned}$$

En utilisant les paramètres :

$$\begin{aligned} p_1 &= m_1(c_1^2 + o_1^2) + I_1 + m_2 l_1^2 \\ p_2 &= m_2(c_2^2 + o_2^2) + I_2 \\ p_3 &= m_2 l_1 c_2 \end{aligned}$$

on obtient l'énergie cinétique totale :

$$\begin{aligned} T &= T_1 + T_2 \\ &= \frac{1}{2}p_1\dot{q}_1^2 + \frac{1}{2}p_2(\dot{q}_1 + \dot{q}_2)^2 + p_3\cos(q_2)\dot{q}_1(\dot{q}_1 + \dot{q}_2) \\ &\quad - p_4 \sin(q_2)\dot{q}_1(\dot{q}_1 + \dot{q}_2) \end{aligned}$$

Révision
Dynamique (5)

Si la gravité est dans le sens négatif de l'axe des Y, l'énergie potentielle totale est donnée par

$$\begin{aligned} U &= m_1 g(c_1 \sin q_1 + o_1 \cos q_1) \\ &\quad + m_2 g(l_1 \sin q_1 + c_2 \sin(q_1 + q_2) + o_2 \cos(q_1 + q_2)) \end{aligned}$$

En définissant les paramètres

$$\begin{aligned} p_5 &= (m_1 c_1 + m_2 l_1) g \\ p_6 &= m_1 o_1 g \\ p_7 &= m_2 c_2 g \\ p_8 &= m_2 o_2 g \end{aligned}$$

on obtient

$$U = p_5 \sin q_1 + p_6 \cos q_1 + p_7 \sin(q_1 + q_2) + p_8 \cos(q_1 + q_2)$$

Révision
Dynamique (6)

- On peut alors obtenir la dynamique par la méthode d'Euler-Lagrange pour chacun des liens:

$$\boldsymbol{\tau} = \frac{d}{dt}\left(\frac{\partial L(\boldsymbol{q},\dot{\boldsymbol{q}})}{\partial \dot{\boldsymbol{q}}}\right) - \frac{\partial L(\boldsymbol{q},\dot{\boldsymbol{q}})}{\partial \boldsymbol{q}}$$

$$\begin{aligned}
\tau_1 &= (p_1 + p_2 + 2p_3 \cos q_2 - 2p_4 \sin q_2)\ddot{q}_1 \\
&\quad + (p_2 + p_3 \cos q_2 - p_4 \sin q_2)\ddot{q}_2 \\
&\quad - (p_3 \sin q_2 + p_4 \cos q_2)\dot{q}_2(2\dot{q}_1 + \dot{q}_2) \\
&\quad + p_5 \cos q_1 - p_6 \sin q_1 + p_7 \cos(q_1 + q_2) - p_8 \sin(q_1 + q_2) \\
\tau_2 &= (p_2 + p_3 \cos q_2 - p_4 \sin q_2)\ddot{q}_1 + p_2\ddot{q}_2 + (p_3 \sin q_2 + p_4 \cos q_2)\dot{q}_1^2 \\
&\quad + p_7 \cos(q_1 + q_2) - p_8 \sin(q_1 + q_2)
\end{aligned}$$

Révision
Dynamique (7)

- Sous forme matricielle:

$$
\begin{aligned}
\tau_1 &= (p_1 + p_2 + 2p_3\cos q_2 - 2p_4\sin q_2)\ddot{q}_1 \\
&\quad + (p_2 + p_3\cos q_2 - p_4\sin q_2)\ddot{q}_2 \\
&\quad - (p_3\sin q_2 + p_4\cos q_2)\dot{q}_2(2\dot{q}_1 + \dot{q}_2) \\
&\quad + p_5\cos q_1 - p_6\sin q_1 + p_7\cos(q_1+q_2) - p_8\sin(q_1+q_2) \\
\tau_2 &= (p_2 + p_3\cos q_2 - p_4\sin q_2)\ddot{q}_1 + p_2\ddot{q}_2 + (p_3\sin q_2 + p_4\cos q_2)\dot{q}_1^2 \\
&\quad + p_7\cos(q_1+q_2) - p_8\sin(q_1+q_2)
\end{aligned}
$$

$$
\boldsymbol{D}(q) = \begin{bmatrix} p_1 + p_2 + 2p_3\cos q_2 - 2p_4\sin q_2 & p_2 + p_3\cos q_2 - p_4\sin q_2 \\ p_2 + p_3\cos q_2 - p_4\sin q_2 & p_2 \end{bmatrix}
$$

$$
\boldsymbol{h}(q,\dot{q}) = \begin{bmatrix} -(p_3\sin q_2 + p_4\cos q_2)\dot{q}_2(2\dot{q}_1 + \dot{q}_2) \\ (p_3\sin q_2 + p_4\cos q_2)\dot{q}_1^2 \end{bmatrix}
$$

$$
\boldsymbol{g}(q) = \begin{bmatrix} p_5\cos q_1 - p_6\sin q_1 + p_7\cos(q_1+q_2) - p_8\sin(q_1+q_2) \\ p_7\cos(q_1+q_2) - p_8\sin(q_1+q_2) \end{bmatrix}
$$

Révision
Dynamique (8)

- Sous forme matricielle (autre forme):

$$\boldsymbol{D}(q) = \begin{bmatrix} p_1 + p_2 + 2p_3\cos q_2 - 2p_4\sin q_2 & p_2 + p_3\cos q_2 - p_4\sin q_2 \\ p_2 + p_3\cos q_2 - p_4\sin q_2 & p_2 \end{bmatrix}$$

$$\boldsymbol{h}(q,\dot{q}) = \begin{bmatrix} -(p_3\sin q_2 + p_4\cos q_2)\dot{q}_2(2\dot{q}_1 + \dot{q}_2) \\ (p_3\sin q_2 + p_4\cos q_2)\dot{q}_1^2 \end{bmatrix}$$

$$\boldsymbol{g}(q) = \begin{bmatrix} p_5\cos q_1 - p_6\sin q_1 + p_7\cos(q_1+q_2) - p_8\sin(q_1+q_2) \\ p_7\cos(q_1+q_2) - p_8\sin(q_1+q_2) \end{bmatrix}$$

$$\boldsymbol{h}(q,\dot{q}) = \boldsymbol{C}(q,\dot{q})\dot{q}$$

$$\boldsymbol{C}(q,\dot{q}) = \begin{bmatrix} -(p_3\sin q_2 + p_4\cos q_2)\dot{q}_2 & -(p_3\sin q_2 + p_4\cos q_2)(\dot{q}_1 + \dot{q}_2) \\ (p_3\sin q_2 + p_4\cos q_2)\dot{q}_1 & 0 \end{bmatrix}$$

Révision
Commande des robots manipulateurs (1)

- Nous savons désormais que la dynamique d'un robot manipulateur peut être obtenue à l'aide de la méthode d'Euler-Lagrange. Celle-ci peut s'exprimer sous la forme matricielle:

$$\tau = D(q)\ddot{q} + h(q,\dot{q}) + g(q)$$

- Quel genre de loi de commande pouvons-nous utiliser afin de régler le couple τ à appliquer à chacune des articulations?
  - Il y a plusieurs stratégies possibles, dépendament du contexte, dans le cadre du cours, nous verrons en particulier:
    - PID axe par axe (joint par joint)
    - PD avec compensation de la gravité
    - Couple pré-calculé
  - On pourra discuter aussi des stratégies suivante, bien que ce ne soit pas matière à l'examen:
    - Contrôle hybride force position et commande en impédance
    - Commande adaptative

Révision
Commande des robots manipulateurs (2)

La commande PID axe par axe est le plus souvent utilisée pour le déplacement de point à point. Comme la consigne est constante par morceau, considérons l'intervalle de temps pendant lequel la consigne est constante. Ainsi $\boldsymbol{q}_d(t) = \text{cte}$ et $\dot{\boldsymbol{q}}_d(t) = 0$. Dans le cas où la commande résulte en un système stable, en régime permanent $\boldsymbol{q}(t) \rightarrow \boldsymbol{q}_d$. et $\dot{\boldsymbol{q}}(t) \rightarrow 0$. Considérons la dynamique d'un robot sans friction :

$$\boldsymbol{\tau} = \boldsymbol{D}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q})$$

Révision
Commande des robots manipulateurs (3)

En régime permanent et toujours sous l'hypothèse de stabilité, $\boldsymbol{g}(\boldsymbol{q}) \rightarrow$ cte, $\boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \rightarrow 0$ et la matrice $\boldsymbol{D}(\boldsymbol{q})$ est presque constante. En utilisant un contrôleur PID axe par axe de la forme

$$\boldsymbol{\tau} = \boldsymbol{K}_P(\boldsymbol{q}_d - \boldsymbol{q}) + \boldsymbol{K}_I \int_0^t (\boldsymbol{q}_d - \boldsymbol{q})dt + \boldsymbol{K}_D(\dot{\boldsymbol{q}}_d - \dot{\boldsymbol{q}})$$

où $\boldsymbol{K}_P$, $\boldsymbol{K}_I$ et $\boldsymbol{K}_D$ sont des matrices diagonales dont les éléments sont les gains PID pour chacun des axes, il est possible de stabiliser le système. Ainsi en régime permanent, la partie intégrale I compensera la gravité alors que les composantes PD influenceront la transitoire. Comme la matrice $\boldsymbol{D}(\boldsymbol{q})$ est fonction de la configuration du robot, différents points de consigne auront des transitoires différentes.

Révision
Commande des robots manipulateurs (4)

Le deuxième contrôleur présenté est un contrôleur PD avec compensation de gravité dont la commande est donnée par

$$\boldsymbol{\tau} = \boldsymbol{K}_P(\boldsymbol{q}_d - \boldsymbol{q}) + \boldsymbol{K}_D(\dot{\boldsymbol{q}}_d - \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q})$$

où $\boldsymbol{K}_P$ et $\boldsymbol{K}_D$ sont des matrices diagonales dont les éléments sont les gains PD pour chacun des axes et $\boldsymbol{g}(\boldsymbol{q})$ les couples associés à la force gravitationnelle. Ainsi, la gravité est compensée et non plus traitée comme une perturbation. À basse vitesse, $\boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \approx 0$, ce type de commande permet de suivre avec assez de précision une trajectoire où $\boldsymbol{q}_d(t)$ et $\dot{\boldsymbol{q}}_d(t)$ sont des fonctions continues.

Révision
Commande des robots manipulateurs (5)

Pour ce contrôleur, la dynamique est compensée et découplée en utilisant la loi de commande :

$$\boldsymbol{\tau} = \boldsymbol{D}(\boldsymbol{q})\left[\ddot{\boldsymbol{q}}_d + \boldsymbol{K}_D(\dot{\boldsymbol{q}}_d - \dot{\boldsymbol{q}}) + \boldsymbol{K}_P(\boldsymbol{q}_d - \boldsymbol{q})\right] + \boldsymbol{h}(\dot{\boldsymbol{q}}, \boldsymbol{q}) + \boldsymbol{g}(\boldsymbol{q})$$

Loi de commande

$$= D(q)\ddot{q} + h(\dot{q}, q) + g(q)$$

Dynamique du robot

où $\boldsymbol{K}_P$ et $\boldsymbol{K}_D$ sont des matrices diagonales dont les éléments sont les gains PD pour chacun des axes ; $\boldsymbol{D}(\boldsymbol{q})$, $\boldsymbol{h}(\dot{\boldsymbol{q}}, \boldsymbol{q})$ et $\boldsymbol{g}(\boldsymbol{q})$ proviennent de la dynamique du robot. Ainsi, la dynamique des erreurs articulaires est découplée :

$$D(q)\left[(\ddot{q}_d - \ddot{q}) + \boldsymbol{K}_D(\dot{q}_d - \dot{q}) + \boldsymbol{K}_P(q_d - q)\right] = 0$$

Révision
Commande des robots manipulateurs (6)

- Considérons encore une fois les équations de la dynamique d'un robot manipulateur:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau$$

- Le couple τ est la variable d'entrée du système, i.e. la variable que vous réglez pour exciter le système. Notons "u" cette variable d'entrée:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = u$$

- L'idée du couple pré-calculé est de compenser pour la dynamique du système en choisissant le couple qui excitera les moteurs tel que:

$$u = M(q)a_q + C(q,\dot{q})\dot{q} + g(q)$$

- Où a_q est une variable à déterminer. La dynamique du système avec cette entrée se réduit alors à:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = M(q)a_q + C(q,\dot{q})\dot{q} + g(q)$$

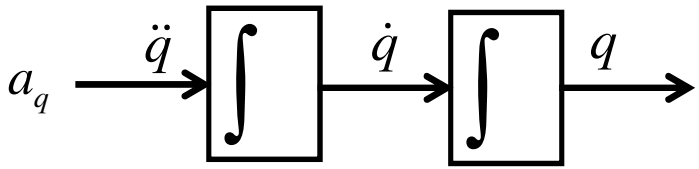
$$\ddot{q} = a_q$$

Révision
Commande des robots manipulateurs (7)

- La dynamique du système avec cette entrée devient donc simplement:

$$\ddot{q} = a_q$$

- Ceci est beaucoup plus simple puisque nous sommes désormais en présence d'un système **linéaire** de deuxième ordre. Le problème du contrôle de notre système robotique est, à ce point-ci, seulement réduit au contrôle d'un système de type "double-intégrateur":



- Maintenant, l'entrée a_q peut stabiliser aisément ce système linéaire de deuxième ordre. a_q peut être choisi de la forme suivante:

$$a_q = \ddot{q}_d + K_D\left(\dot{q}_D - \dot{q}\right) + K_P\left(q_D - q\right)$$

$$\text{avec } K_D \text{ et } K_P > 0$$

- Donc l'entrée globale du système est donc:

$$u = M(q)\left(\ddot{q}_d + K_D\left(\dot{q}_d - \dot{q}\right) + K_P\left(q_d - q\right)\right) + C(q,\dot{q})\dot{q} + g(q)$$

Commande stabilisante

Révision
Commande des robots manipulateurs (8)

Pour ce contrôleur, la dynamique est compensée et découplée en utilisant la loi de commande :

$$\boldsymbol{\tau} = \boldsymbol{D}(\boldsymbol{q})\left[\ddot{\boldsymbol{q}}_d + \boldsymbol{K}_D(\dot{\boldsymbol{q}}_d - \dot{\boldsymbol{q}}) + \boldsymbol{K}_P(\boldsymbol{q}_d - \boldsymbol{q})\right] + \boldsymbol{h}(\dot{\boldsymbol{q}}, \boldsymbol{q}) + \boldsymbol{g}(\boldsymbol{q})$$

Loi de commande

$$= D(q)\ddot{q} + h(\dot{q}, q) + g(q)$$

Dynamique du robot

où $\boldsymbol{K}_P$ et $\boldsymbol{K}_D$ sont des matrices diagonales dont les éléments sont les gains PD pour chacun des axes ; $\boldsymbol{D}(\boldsymbol{q})$, $\boldsymbol{h}(\dot{\boldsymbol{q}}, \boldsymbol{q})$ et $\boldsymbol{g}(\boldsymbol{q})$ proviennent de la dynamique du robot. Ainsi, la dynamique des erreurs articulaires est découplée :

$$D(q)\left[(\ddot{q}_d - \ddot{q}) + \boldsymbol{K}_D(\dot{q}_d - \dot{q}) + \boldsymbol{K}_P(q_d - q)\right] = 0$$

Révision
Planification de trajectoires (1)

- Lors des cours précédents, nous avons étudiés les géométries des manipulateurs sériels. Nous nous sommes attardés à développer la cinématique directe et inverse, nous avons abordé la dynamique ainsi que certaines stratégies de contrôle.

- Un aspect que nous n'avons toutefois pas encore considéré est la **planification de trajectoires**.

- En effet, pour cette section, nous nous attarderons à définir la trajectoire d'un robot manipulateur, c'est-à-dire: $q_d, \dot{q}_d, \ddot{q}_d$

- Pour ce faire nous considérerons deux modes d'opération distincts:

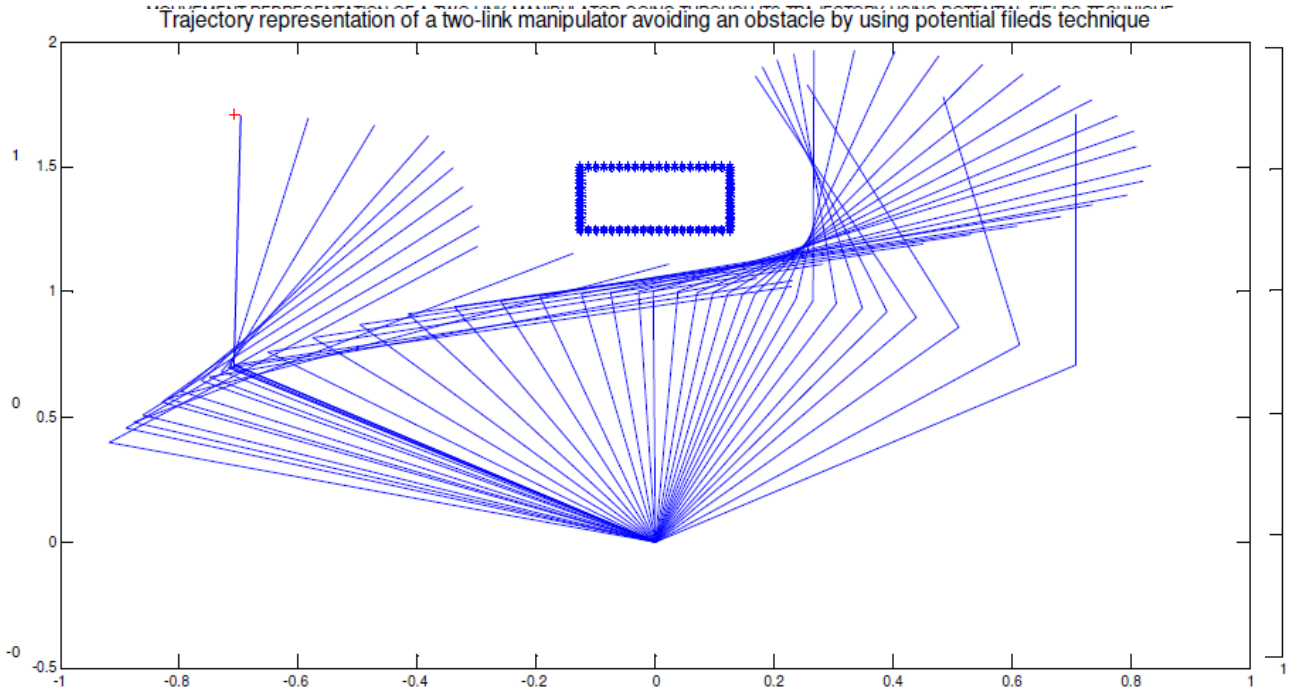
  - Le mode **point par point**, où les points sont relativement éloignés les uns des autres
  - Le mode **continu**, où les points sont suffisamment rapprochés pour former une trajectoire quasi-continue.

- Pour chacun de ces cas, nous verrons comment générer une trajectoire cohérente.

Révision
Planification de trajectoires (2)

- Voici une notion qui **n'est pas à l'ordre du cours**, il s'agit de l'utilisation de champs de potentiels pour diriger les robots.

- Lorsqu'il y a des obstacles dans l'environnement du robot, c'est une approche relativement simple conceptuellement qui consiste à utiliser un champ potentiel de répulsion et un champ d'attraction:



Révision
Planification de trajectoires (3)

- Pour ce mode, voici les profils typiques de position, vitesse et accélération entre deux points q_o et q_1:

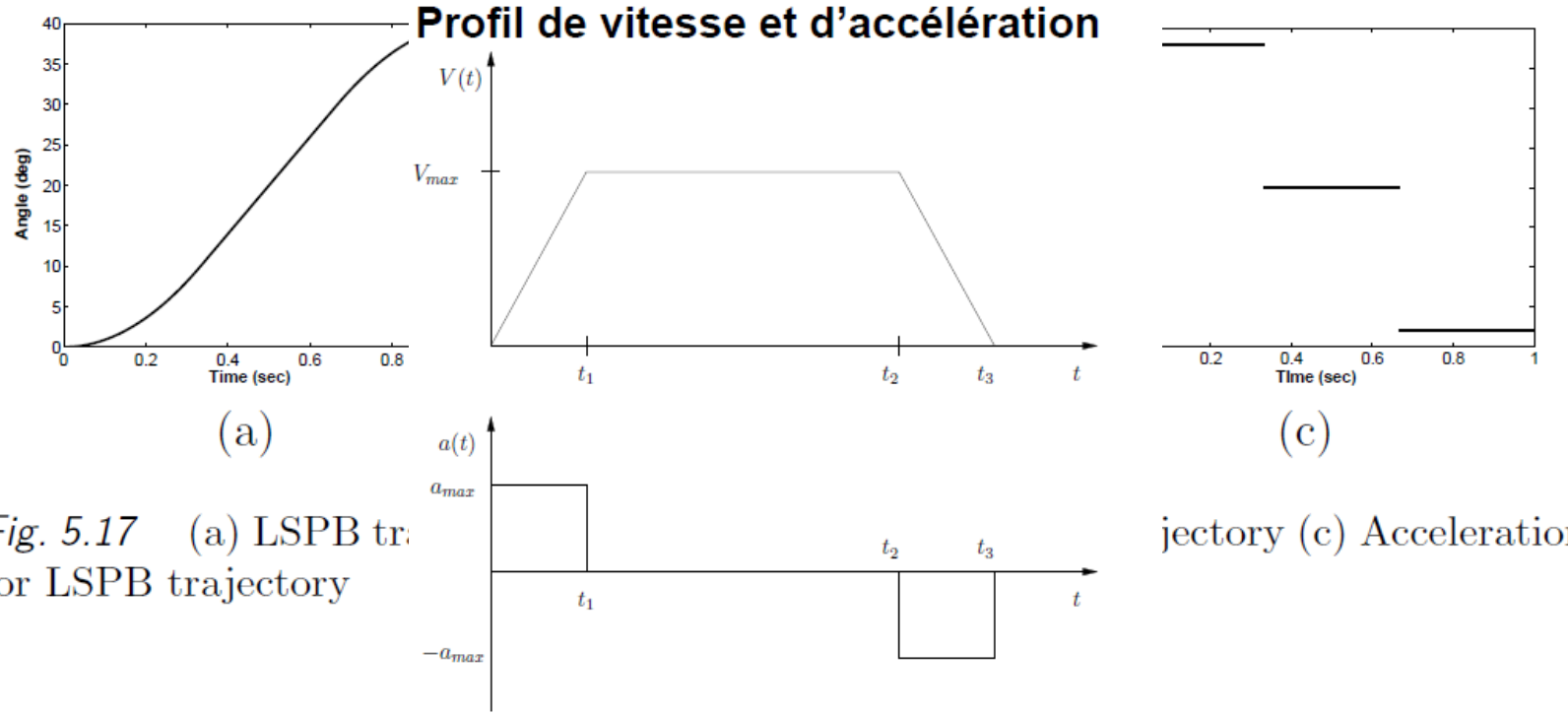


Fig. 5.17 (a) LSPB tr… jectory (c) Acceleration for LSPB trajectory

- Développons les équations permettant de définir la trajectoire articulaire

Révision
Planification de trajectoires (4)

Si V_{max} et a_{max} nous sont donnés, et que la position y doit varier de y_0 à y_1, on a

$$\begin{aligned} t_1 &= V_{max}/a_{max} \\ t_3 - t_2 &= V_{max}/a_{max} \\ V_{max}\left(\frac{t_1}{2} + t_2 - t_1 + \frac{t_3 - t_2}{2}\right) &= y_1 - y_0 \\ V_{max}\left(-\frac{t_1}{2} + t_2 + \frac{t_3 - t_2}{2}\right) &= y_1 - y_0 \end{aligned}$$

Révision
Planification de trajectoires (5)

Mais comme $t_1 = t_3 - t_2$, on a

$$\begin{aligned} V_{max} t_2 &= y_1 - y_0 \\ t_2 &= \frac{y_1 - y_0}{V_{max}} \end{aligned}$$

Ici, il faut que $t_2 > t_1$. On a donc

$$y(t) = \begin{cases} y_0 + \frac{1}{2} a_{max} t^2 & \text{si} \quad 0 < t \leq t_1 \\ y_0 + t a_{max} t_1 - \frac{1}{2} a_{max} t_1^2 & \text{si} \quad t_1 < t \leq t_2 \\ y_0 + t a_{max} t_1 - \frac{1}{2} a_{max} t_1^2 - \frac{1}{2} a_{max} (t - t_2)^2 & \text{si} \quad t_2 < t \leq t_3 \end{cases}$$

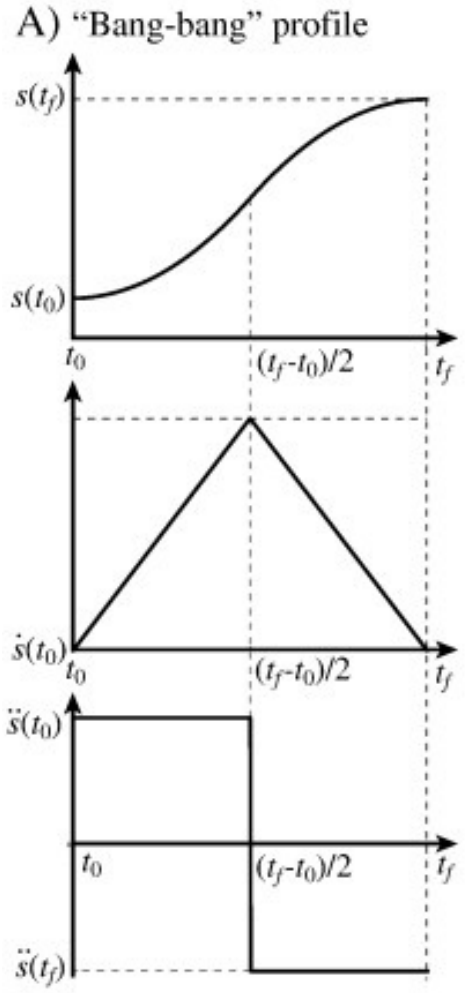
Révision
Planification de trajectoires (6)

- Dans un contexte où:
  - Les mouvements saccadés ne sont pas un problème pour le système robotique et les produits manipulés
  - La rapidité est très importante

- On peut considérer, au lieu du LSPB, d'élaborer des trajectoires à temps minimal.
  - Les trajectoires à temps minimal sont les trajectoires les plus rapides pour amener le robot d'une configuration articulaire q_0 à une configuration articulaire q_1.

Révision
Planification de trajectoires (7)

- De plus, le profil de vitesse n'est pas nécessairement trapézoïdal:



Il sera trapézoïdale seulement si V_{max} est atteinte lors du déplacement.

Révision
Planification de trajectoires (8)

- Une autre approche pour définir des trajectoires en mode point-par-point consiste à utiliser des polynômes d'ordre 3:

$$q_d(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

- Ces polynômes ont quatres paramètres (quatre degrés de liberté) et on peut donc, en général, leur imposer quatre contraintes.
  - On peut par exemple, donner des contraintes sur la position initiale et finale (2) ainsi que sur la vitesse initiale et finale (2)

- La vitesse articulaire est donnée simplement en dérivant:

$$\dot{q}_d(t) = a_1 + 2a_2 t + 3a_3 t^2$$

Révision
Planification de trajectoires (9)

$$q_d(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$
$$\dot{q}_d(t) = a_1 + 2a_2 t + 3a_3 t^2$$

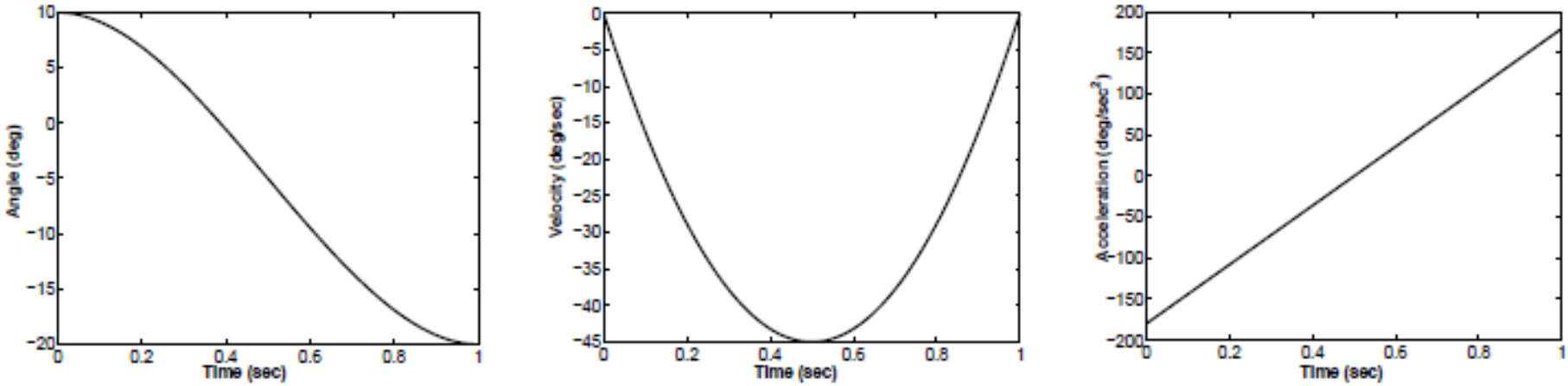
- À l'aide de ces équations, on bâtit le système de quatre équations à quatre inconnues:

$$q_0 = a_0 + a_1 t_0 + a_2 t_0^2 + a_3 t_0^3$$
$$\dot{q}_0 = a_1 + 2a_2 t_0 + 3a_3 t_0^2$$
$$q_f = a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3$$
$$\dot{q}_f = a_1 + 2a_2 t_f + 3a_3 t_f^2$$

- On définit donc les polynômes simplement en résolvant le système:

$$\begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 \\ 0 & 1 & 2t_0 & 3t_0^2 \\ 1 & t_f & t_f^2 & t_f^3 \\ 0 & 1 & 2t_f & 3t_f^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} q_0 \\ \dot{q}_0 \\ q_f \\ \dot{q}_f \end{bmatrix}$$

Révision
Planification de trajectoires (10)



(a) (b) (c)

Fig. 5.13 (a) Cubic polynomial trajectory (b) Velocity profile for cubic polynomial trajectory (c) Acceleration profile for cubic polynomial trajectory

Révision
Planification de trajectoires (11)

- Nous avons vu que, puisqu'un polynôme cubique possède quatre paramètres, il est possible de spécifier des contraintes sur les positions articulaires de départ et d'arrivé ainsi que sur les vitesses articulaires de départ et d'arrivée.
  - De tels polynômes donnent des trajectoires continues en position et en vitesse.
  - Cependant, elle admet des discontinuités au niveau de l'accélération aux points de départ et d'arrivé.
  - Ces discontinuités au niveau de l'accélération créé des impulsions au niveau du jerk (dérivée de l'accélération), ce qui peut exciter les modes vibratoires du manipulateur.
    - Ceci est indésirable, puisque ça peut causer une perte de précision au niveau du suivi de trajectoire et/ou causer des ennuis mécaniques en fatigue.
- Une façon de remédier à ces problèmes consiste à utiliser les polynômes quintiques.

Révision
Planification de trajectoires (12)

- Les polynômes quintiques sont des polynômes de 5ième ordre de la forme:

$$q_d(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5$$

- Puisque le polynôme possède 6 paramètres, il est possible, en général, de lui donner 6 contraires:
  - 2 pour la position initiale et finale
  - 2 pour la vitesse initiale et finale
  - 2 pour l'accélération initiale et finale

- La vitesse et l'accélération articulaires sont données par:

$$\dot{q}_d(t) = a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4$$
$$\ddot{q}_d(t) = 2a_2 + 6a_3 t + 12a_4 t^2 + 20a_5 t^3$$

Révision
Planification de trajectoires (13)

- Les contraintes sont donc tout simplement données par:

$$q_0 = a_0 + a_1 t_0 + a_2 t_0^2 + a_3 t_0^3 + a_4 t_0^4 + a_5 t_0^5$$
$$\dot{q}_0 = a_1 + 2a_2 t_0 + 3a_3 t_0^2 + 4a_4 t_0^3 + 5a_5 t_0^4$$
$$\ddot{q}_0 = 2a_2 + 6a_3 t_0 + 12a_4 t_0^2 + 20a_5 t_0^3$$
$$q_f = a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 + a_4 t_f^4 + a_5 t_f^5$$
$$\dot{q}_f = a_1 + 2a_2 t_f + 3a_3 t_f^2 + 4a_4 t_f^3 + 5a_5 t_f^4$$
$$\ddot{q}_f = 2a_2 + 6a_3 t_f + 12a_4 t_f^2 + 20a_5 t_f^3$$

- Il suffit donc de résoudre:

$$\begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 & t_0^4 & t_0^5 \\ 0 & 1 & 2t_0 & 3t_0^2 & 4t_0^3 & 5t_0^4 \\ 0 & 0 & 2 & 6t_0 & 12t_0^2 & 20t_0^3 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 0 & 0 & 2 & 6t_f & 12t_f^2 & 20t_f^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} q_0 \\ \dot{q}_0 \\ \ddot{q}_0 \\ q_f \\ \dot{q}_f \\ \ddot{q}_f \end{bmatrix}$$

Révision
Planification de trajectoires (14)

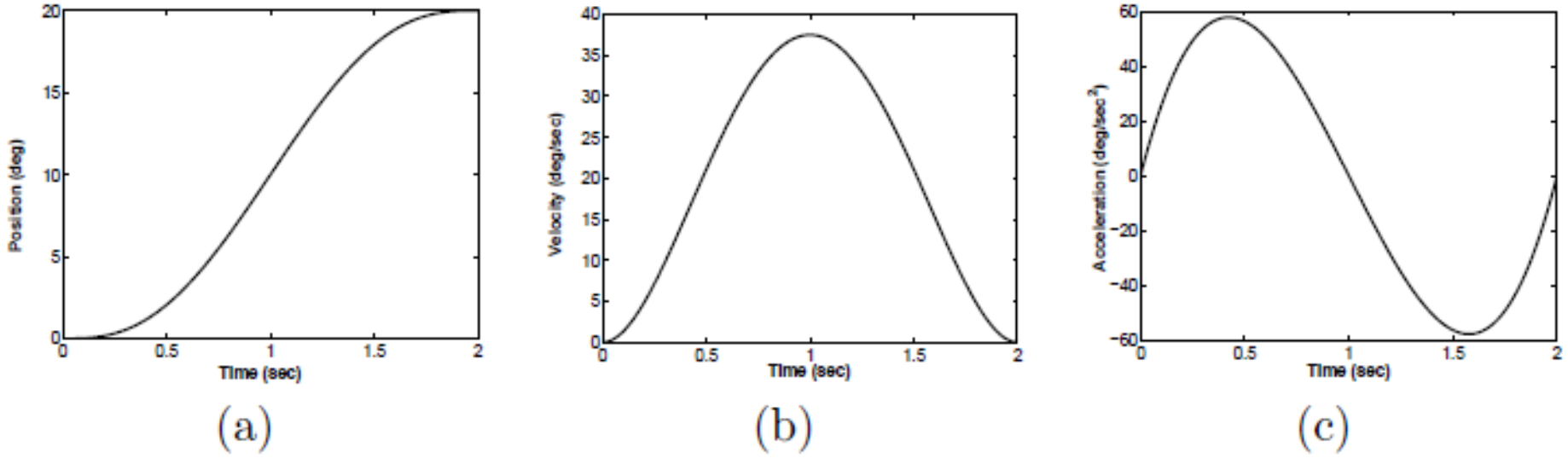


Fig. 5.15 (a) Quintic Polynomial Trajectory. (b) Velocity Profile for Quintic Polynomial Trajectory. (c) Acceleration Profile for Quintic Polynomial Trajectory

Révision
Planification de trajectoires (15)

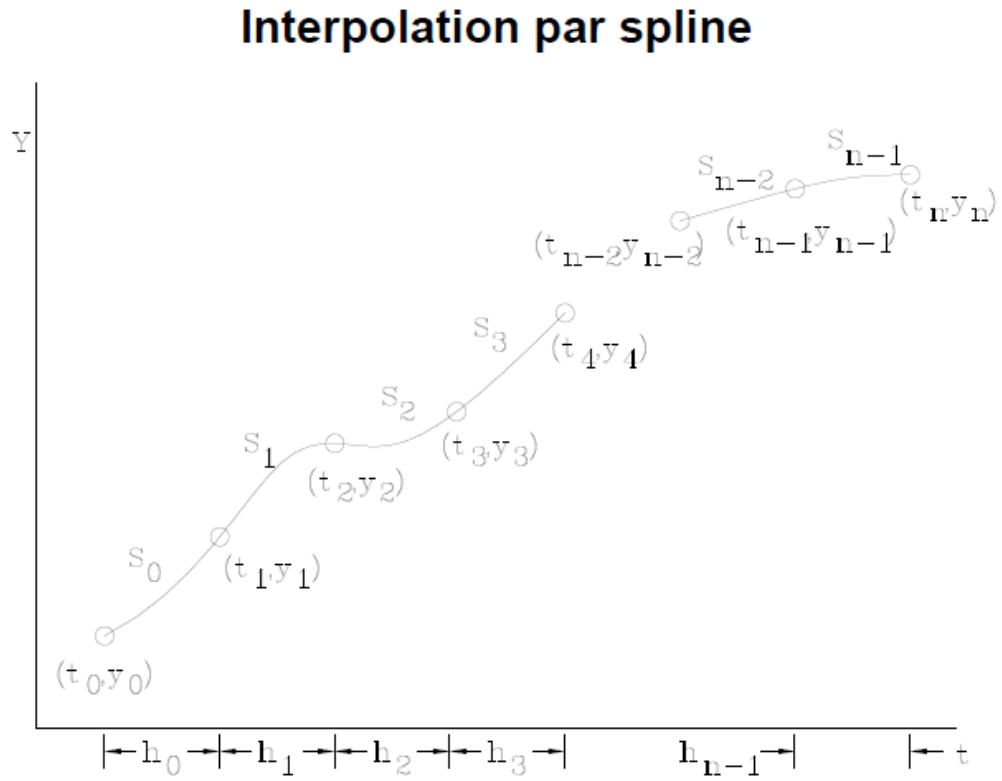
- Lorsque le robot fonctionne en mode continu, les points à atteindre ne sont plus des points singuliers situés à une distance considérable.
  - Il s'agit maintenant de points très rapprochés et nous devons donc considérer des trajectoires définies différemment.
    - En particulier, puisque plusieurs points seront interpolés en peu de temps, il serait pratique d'avoir des trajectoires connectant ces points qui seraient continues en vitesse et en accélération aux points de jonction.
    - Pour ce faire, nous utiliserons une ***spline cubique***.

- Une spline, c'est une fonction définie par morceaux à l'aide de polynômes.
  - Une spline cubique est donc une fonction définie à l'aide de polynômes d'ordre 3 de forme:

$$q_d(t_0) = a_0 + a_1(t - t_0) + a_2(t - t_0)^2 + a_3(t - t_0)^3$$

Révision
Planification de trajectoires (16)

- Donc, pour interpoler *n+1* points nous utiliserons *n* polynômes qui seront continus en vitesse et en accélération aux points de jonctions.



- Comment peut-on construire la spline cubique avec ces conditions de continuité?

Révision
Planification de trajectoires (17)

- Énonçons de façon plus clair les contraintes définissant la spline cubique:
  - Soit une fonction f(t) définie dans l'intervalle [a,b] avec $a=t_0 < t_1 < \ldots < t_n = b$ et un ensemble de points prédéterminés, alors:

$1-S$ est la spline cubique, donné par S_j, dans l'intervalle $[t_j, t_{j+1}]$ pour chaque $j = 0,1,\ldots,n-1$

$2-S_j(t_j) = f(t_j) = y_j$ pour chaque $j = 0,1,\ldots,n$

$3-S_{j+1}(t_{j+1}) = S_j(t_{j+1})$ pour chaque $j = 0,1,\ldots,n-2$

$4-\frac{d}{dt}S_{j+1}(t_{j+1}) = \frac{d}{dt}S_j(t_{j+1})$ pour chaque $j = 0,1,\ldots,n-2$

$5-\frac{d^2}{dt^2}S_{j+1}(t_{j+1}) = \frac{d^2}{dt^2}S_j(t_{j+1})$ pour chaque $j = 0,1,\ldots,n-2$

6 – On impose les vitesse aux extrémités de la spline:

$$\frac{d}{dt}S_0(t_0) = \frac{d}{dt}f(t_0) \quad \text{et} \quad \frac{d}{dt}S_n(t_n) = \frac{d}{dt}f(t_n)$$

Révision
Planification de trajectoires (18)

- Méthode de résolution:
  - Vous connaissez les a_j quasi-directement, vous connaissez les h_j et les pentes imposées au début et à la fin des splines, donc:
    - Vous notez les a_j
    - Vous posez le système d'équation sous forme matricielle et résolvez pour trouver les c_j
    - Vous trouvez les b_j et les d_j à l'aide des relations établies plus tôt, i.e.:

$$b_j = \frac{1}{h_j}\left(a_{j+1} - a_j\right) - \frac{h_j}{3}\left(2c_j + c_{j+1}\right)$$

$$d_j = \frac{c_{j+1} - c_j}{3h_j}$$

Révision
Planification de trajectoires (19)

- Ces équations peuvent être résolues sous forme matricielle:

$$Ax = b$$

$$\boldsymbol{A} = \begin{bmatrix} 2h_0 & h_0 & \cdots & \cdots & \cdots & \cdots & 0 \\ h_0 & 2(h_0+h_1) & h_1 & 0 & \cdots & \cdots & 0 \\ 0 & h_1 & 2(h_1+h_2) & h_2 & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & h_{n-2} & 2(h_{n-2}+h_{n-1}) & h_{n-1} \\ 0 & \cdots & \cdots & \cdots & 0 & h_{n-1} & 2h_{n-1} \end{bmatrix}$$

$$\boldsymbol{b} = \begin{bmatrix} \frac{3}{h_0}(a_1-a_0) - 3\frac{d}{dt}f(t_0) \\ \frac{3}{h_1}(a_2-a_1) - \frac{3}{h_0}(a_1-a_0) \\ \vdots \\ \frac{3}{h_{n-1}}(a_n-a_{n-1}) - \frac{3}{h_{n-2}}(a_{n-1}-a_{n-2}) \\ 3\frac{d}{dt}f(t_n) - \frac{3}{h_{n-1}}(a_n-a_{n-1}) \end{bmatrix}$$

$$\boldsymbol{x} = \begin{bmatrix} c_0 & \cdots & c_n \end{bmatrix}^T$$

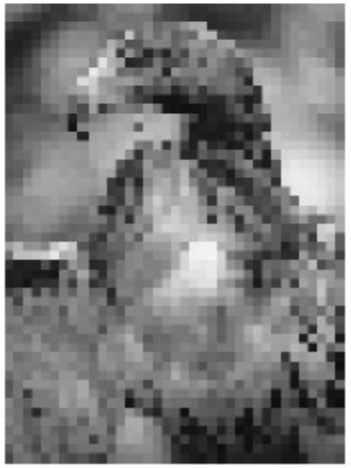
Révision
Traitement de bas niveau (1)

- **Autres notions:**
  - Effet de la résolution:

résolution originale

pixel 2× original

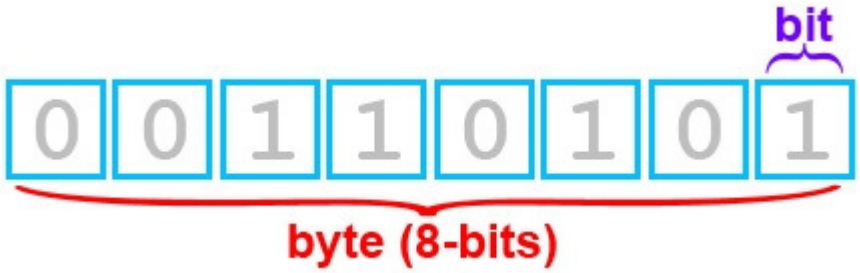
pixel 4× original



pixel 8× original

Révision
Traitement de bas niveau (2)

- **Autres notions (suite):**
  - L'intensité de l'image *réelle* sera donc discrétisée en surface, étant donnée que celle-ci sera constituée de pixels.
  - L'intensité sera aussi discrétisée par sa représentation en bits (nombre binaire).



Decimal pattern (Hex Value)	Binary numbers
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10 – (A)	1010
11 – (B)	1011
12 – (C)	1100
13 – (D)	1101
14 – (E)	1110
15 – (F)	1111
16 – (10)	10000

Souvent, un considérera une image numérique en tons de gris représentés en 1-byte (8 bits): 2^8 = 256 niveaux de représentation de gris possibles (0 à 255).

Révision
Traitement de bas niveau (3)

- **Autres notions (suite):**
  - Effet du nombre de bits:

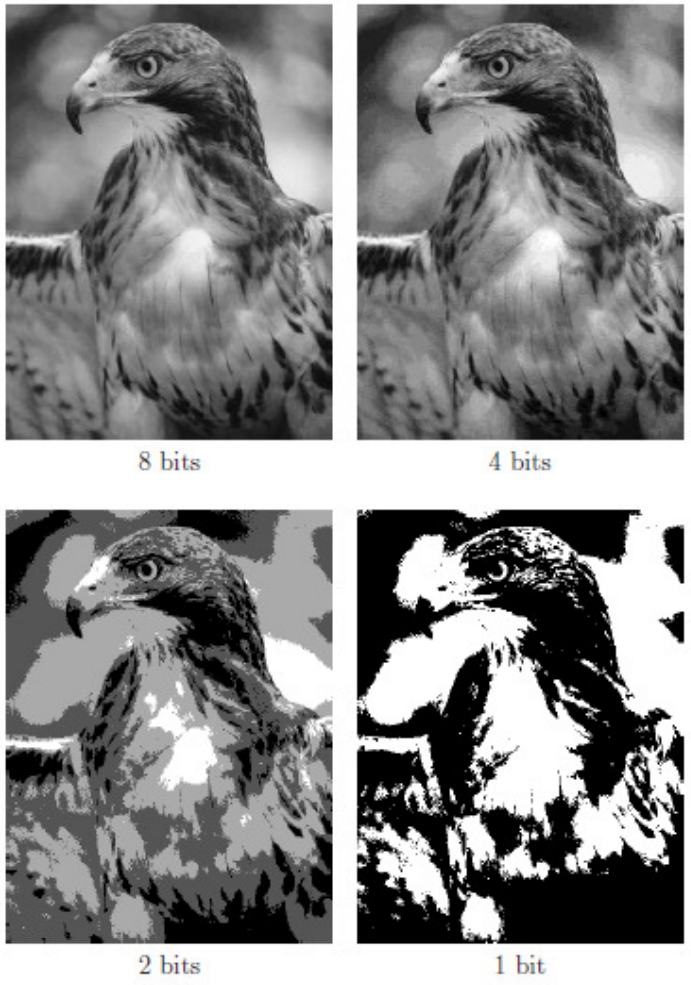


FIG. 1.3 – Influence du nombre de bits pour les niveaux de gris

Révision
Traitement de bas niveau (4)

- **Éclairage:**
  - Dépendamment des opérations que vous désirez effectuer, vous devrez choisir un **éclairage** approprié. Ici, on présente quatre techniques d'éclairage:
    - L'éclairage **diffus** utilisé pour les objets lisses ayant des surfaces régulières
    - L'éclairage **direct** utilisé souvent pour détecter des défauts de surface.

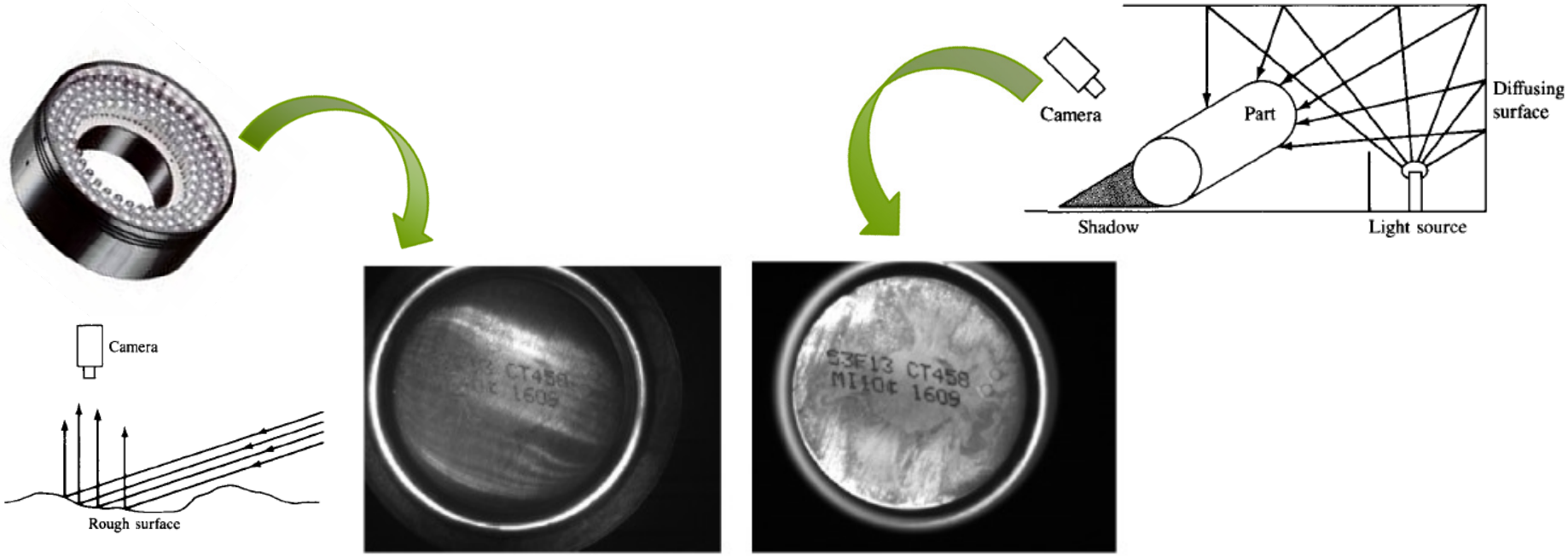
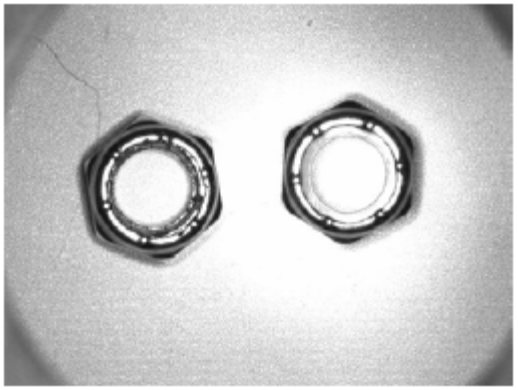
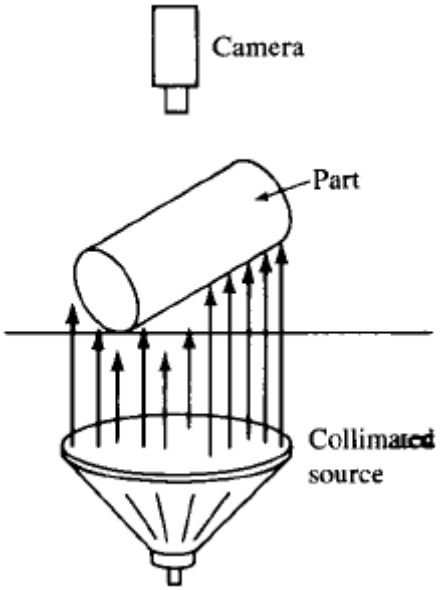


Fig. 6 – Bottom of a soda can. Left: illuminated with a bright field ring light, but shows poor contrast, uneven lighting, and specular reflections. Right: imaged with diffuse light, creating an even background allowing the code to be read.

Révision
Traitement de bas niveau (5)

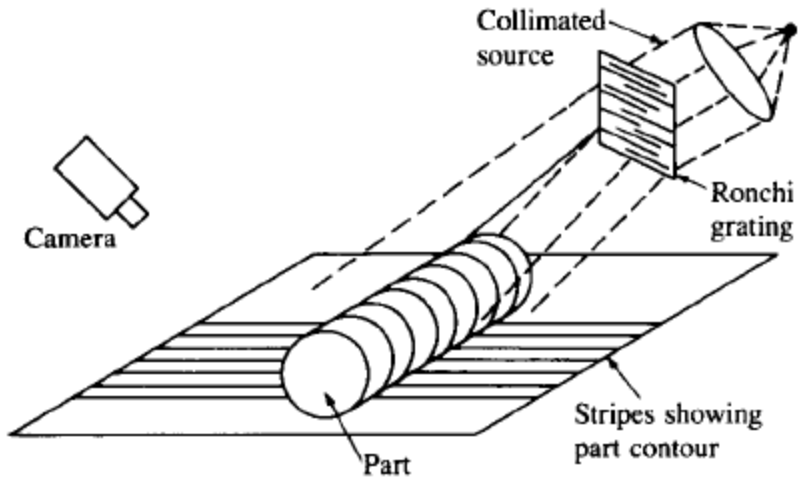
- **Éclairage (suite):**
  - Dépendamment des opérations que vous désirez effectuer, vous devrez choisir un **éclairage** approprié. Ici, on présente quatre techniques d'éclairage:
    - L'éclairage **arrière** qui permet de bien générer des images en noir et blanc





Révision
Traitement de bas niveau (6)

- **Éclairage (suite):**
  - Dépendamment des opérations que vous désirez effectuer, vous devrez choisir un **éclairage** approprié. Ici, on présente quatre techniques d'éclairage:
    - L'éclairage **structuré** par bande ou par grille. La déformation du patron permet d'identifier les caractéristiques de l'objet.



Révision
Traitement de bas niveau (7)

- **Distance entre les pixels:**

- Considérons les pixels p p, q et z de coordonnées (i, j), (s, t) et (u, v) respectivement, nous appelons *D* une fonction de distance si:

1. $D(p,q) \geq 0$ avec $D(p,q) = 0$ ssi $p = q$
2. $D(p,q) = D(q,p)$
3. $D(p,z) \leq D(p,q) + D(q,z)$

- **Distance euclidienne:**

$$D_e(p,q) = \sqrt{(i-s)^2 + (j-t)^2}$$

Dans le domaine des $D_e \leq 3$:

			3			
	$\sqrt{8}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{8}$	
	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$	
3	2	1	0	1	2	3
	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$	
	$\sqrt{8}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{8}$	
			3			

Révision
Traitement de bas niveau (8)

- **Distance entre les pixels (suite):**

- **La distance D_4 ou *city-blocks* est donnée par:**

$$D_4(p,q) = |i-s| + |j-t|$$

Dans le domaine des $D_4 \leq 2$:

$$\begin{array}{ccccc} & & 2 & & \\ & 2 & 1 & 2 & \\ 2 & 1 & 0 & 1 & 2 \\ & 2 & 1 & 2 & \\ & & 2 & & \end{array}$$

- **La distance D_8 ou *chess-board* est donnée par:**

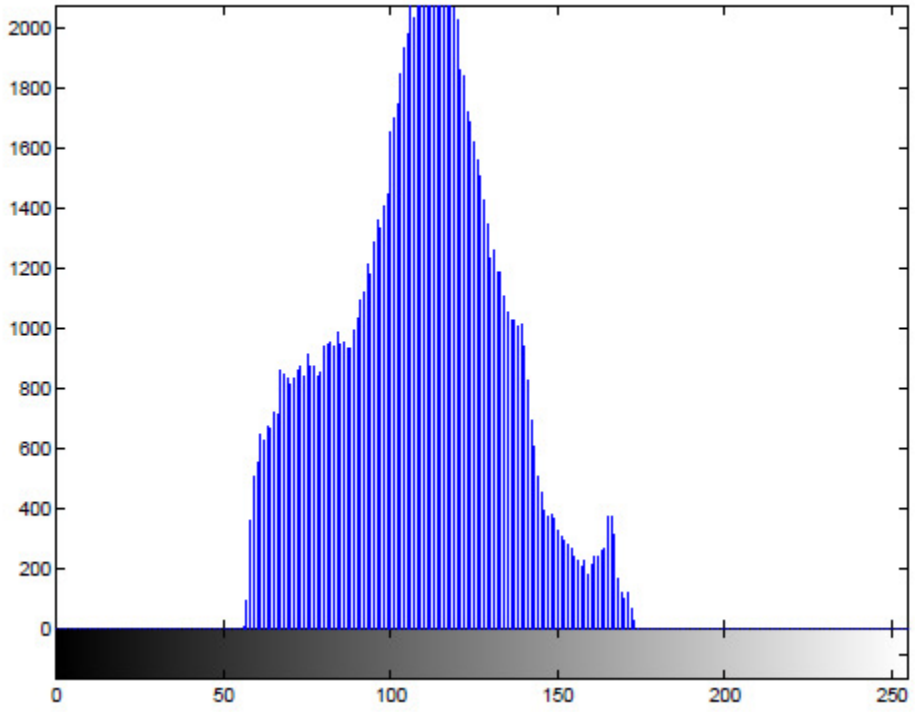
$$D_8(p,q) = \max(|i-s|, |j-t|)$$

Dans le domaine des $D_8 \leq 2$:

$$\begin{array}{ccccc} 2 & 2 & 2 & 2 & 2 \\ 2 & 1 & 1 & 1 & 2 \\ 2 & 1 & 0 & 1 & 2 \\ 2 & 1 & 1 & 1 & 2 \\ 2 & 2 & 2 & 2 & 2 \end{array}$$

Révision
Traitement de bas niveau (9)

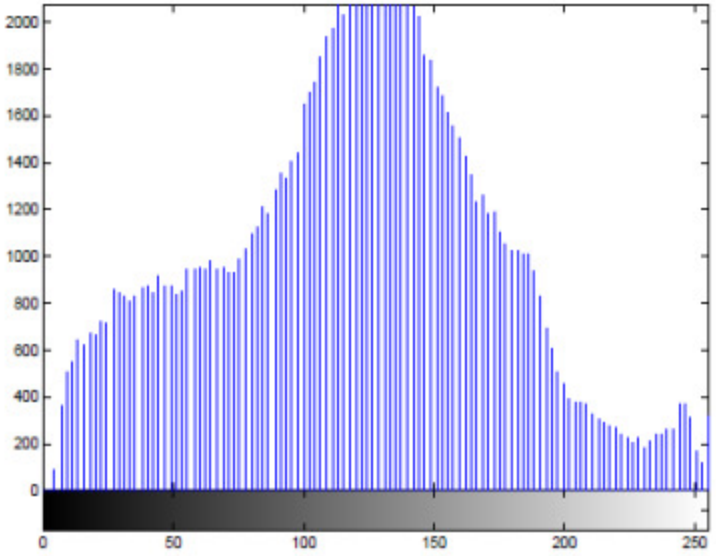
- Considérez la paire image-histogramme suivante:



- Il y a peu de nuances au niveau des tons de gris et cela se voit autant au niveau de l'histogramme qu'au niveau de l'image.

Révision
Traitement de bas niveau (10)

- Il est possible de retrouver une image plus claire en augmentant le contraste:

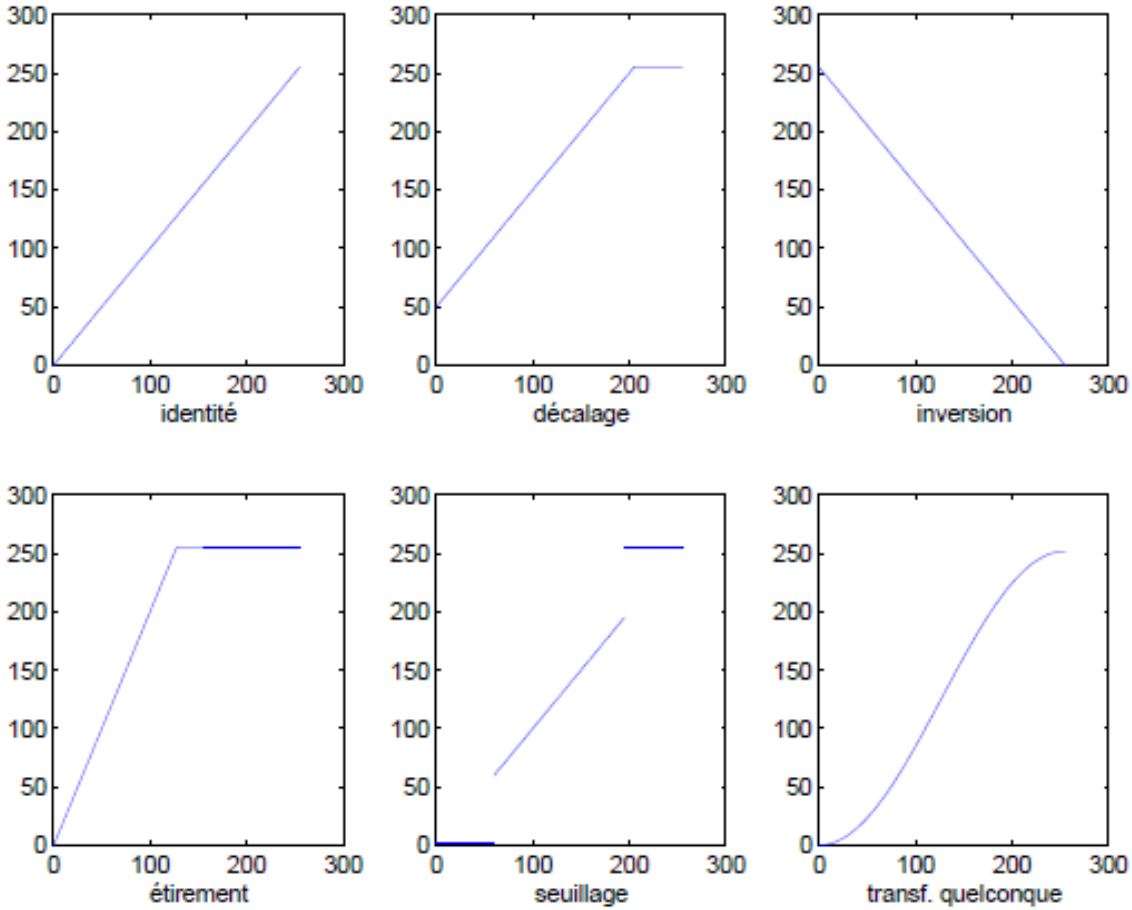


- Il suffit d'effectuer une *transformation* afin d'augmenter le contraste. Dans ce cas-ci, il s'agissait d'un *décalage* et d'un *étirement* qui se traduisent par:

$$I' = (I - 55)\frac{255}{170 - 55}$$

Révision
Traitement de bas niveau (11)

- Autres types de transformation d'histogramme:



Révision
Traitement de bas niveau (12)

- **Seuillage par histogramme**
  - Considérons cette image et son histogramme:

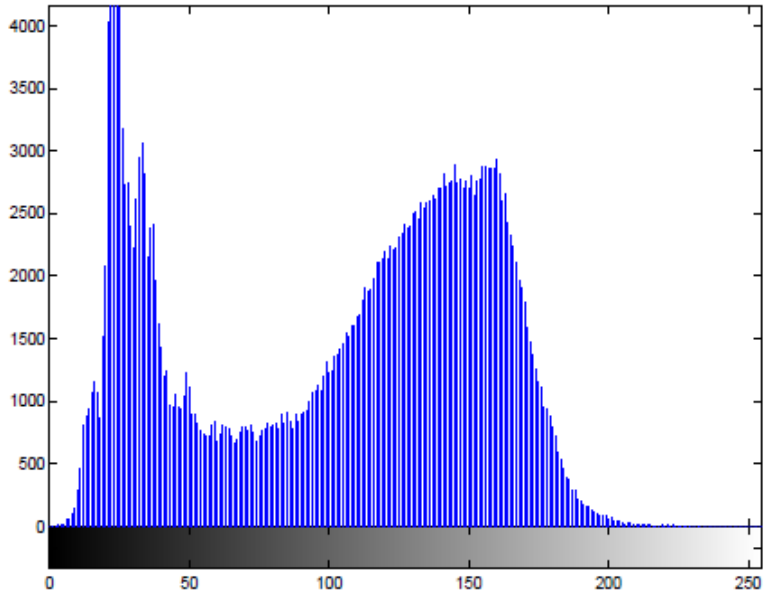


FIG. 2.8 – Histogramme d'une image avant le seuillage

  - On observe que la distribution d'intensité possède deux modes!

Révision
Traitement de bas niveau (13)

- La méthode d'Otsu permet de trouver une valeur seuil automatiquement.
  - Celle-ci est basée sur l'hypothèse que l'image provient de deux distributions gaussiennes: l'une pour la partie claire et l'autre pour la partie foncée.
  - L'hypothèse est donc que l'histogramme de l'image réelle est une superposition de la distribution gaussienne de la partie pâle avec la distribution de la partie foncée.
    - Cette méthode donnera donc de bons résultats lorsqu'on détecte effectivement dans l'histogramme une distribution *à gauche* et une autre *à droite* avec des moyennes significativement différente.
- **L'idée derrière la méthode d'Otsu est de trouvé un seuil t créant deux groupes dont les variances pondérées sont les plus petites possible.**

Révision
Traitement de bas niveau (14)

- ***"L'idée derrière la méthode d'Otsu est de trouvé un seuil t créant deux groupes dont les variances pondérées sont les plus petites possible."***
- Donc on veut minimiser ceci:

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t)$$

- Avec:

$$q_1(t) = \sum_{i=1}^{t} P(i) \qquad q_2(t) = \sum_{i=t+1}^{I} P(i) = 1 - q_1(t)$$

- On pourrait donc utiliser simplement ces équations et trouver la valeur de t qui minimise la variance $\sigma_w^2(t)$de manière itérative.
  - Il existe tout de même une manière plus efficace.

Révision
Traitement de bas niveau (15)

- En effet, puisque Otsu a démontré que minimiser:

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t)$$

- Revient à maximiser:

$$\sigma_b^2 = q_1(t)[1 - q_1(t)][\mu_1(t) - \mu_2(t)]^2$$

- Avec $\mu_1(t)$ la moyenne du groupe 1 et $\mu_2(t)$ la moyenne du groupe 2:

$$\mu_1(t) = \frac{\sum_{i=0}^{t} iP(i)}{q_1(t)}$$

$$\mu_2(t) = \frac{\sum_{i=t+1}^{255} iP(i)}{q_2(t)}$$

Révision
Traitement de bas niveau (16)

FIG. 2.10 – Image binaire après seuillage : méthode de Otsu

Révision
Traitement de bas niveau (17)

- Nous venons d'introduire la notion de *masque* pour traiter les images affectée de bruit de type poivre et sel.
  - Un masque est un opérateur sur l'intensité *I* d'un pixel *p* permettant de calculer une nouvelle intensité *I'* en fonction de l'intensité de *p* et de ses voisins. Un masque peut être de dimension *n x n.*
- Forme général d'un masque (ici, 3x3):

$$\begin{aligned} I'(i,j) &= w_1 I(i-1,j+1) + w_2 I(i,j+1) + w_3 I(i+1,j+1) \\ &\quad + w_4 I(i-1,j) + w_5 I(i,j) + w_6 I(i+1,j) \\ &\quad + w_7 I(i-1,j-1) + w_8 I(i,j-1) + w_9 I(i,j+1) \end{aligned}$$

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Révision
Traitement de bas niveau (18)

- **Lissage:**
  - Pour *lisser* une image, on peut utiliser la moyenne du pixel *p* et des pixels dans le voisinnage de *p*. Par exemple, si on utilise $N_8(p)$, on aura $w_i=1/9$.
  - Le lissage a pour avantage d'améliorer les images bruitées, mais le désavantage de rendre floues les arrêtes et dertains détails pointus.
  - De manière similaire, si on considère encore plus de voisins (en utilisant par exemple un masque 5x5) on aura des arrêtes encore plus floues.

Révision
Traitement de bas niveau (19)

- **Détection d'arrêtes par gradient :**
  - Pour détecter les arrêtes dans une image, on peut dériver partiellement l'intensité en x et en y en utilisant un filtre approprié:

$$\left(\frac{\partial I}{\partial x}\right)$$

-1	0	1

  - Cependant, le filtre ci-dessus sera très sensible au bruit, on peut donc utiliser la moyenne des gradients:

$$\left(\frac{\partial I}{\partial x}\right)$$

ou

-1	0	1
-1	0	1
-1	0	1

-1	0	1
-2	0	2
-1	0	1

  - Remarque:
    - On a qu'a transposer ces gradients pour obtenir l'information dans le sens horizontal: $\left(\frac{\partial I}{\partial y}\right)$

Révision
Traitement de bas niveau (20)

- **Le gradient total et sa direction sont donnés par:**

$$|\nabla I| = \sqrt{\frac{\partial I}{\partial x}^2 + \frac{\partial I}{\partial y}^2}$$

$$\theta = \text{atan2}\left(\frac{\partial I}{\partial y}, \frac{\partial I}{\partial x}\right)$$

- **D'autres masques…**

- **<u>Le Laplacien</u> est défini par:** $\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$

  - Il peut aussi être calculé par un masque:

0	1	0
1	-4	1
0	1	0

Révision
Traitement de bas niveau (21)

- **D'autres masques…**

- Pour rehausser les contours d'une image, on peut utiliser le masque ci-dessous:

$$\frac{1}{1+\alpha}\begin{bmatrix} -\alpha & \alpha-1 & -\alpha \\ \alpha-1 & \alpha+5 & \alpha-1 \\ -\alpha & \alpha-1 & -\alpha \end{bmatrix}$$

FIG. 2.12 – Figure originale et après application du rehaussement ($\alpha = 0.5$)

Révision
Traitement de haut niveau (1)

- Classification VS Segmentation:
  - *Segmentation*: Regroupement de pixels qui possèdent entre eux des caractéristiques similaires (Contours, intensité, etc…) sans utiliser d'information contextuelle.
  - *Classification:* Classification des régions d'une image basée sur la reconnaissance de formes ou de caractéristiques communes basée sur l'information contextuelle.

Révision
Traitement de haut niveau (2)

- La première méthode de classification que nous étudierons sera la *méthode des moments*.
  - C'est une méthode très répandue (Industrie, jeux vidéos, films).

- **Introduisons d'abord ce qu'est, mathématiquement, un moment d'ordre (p+q) :**

$$M_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q I(x,y) dx dy$$

- Puisque nous sommes dans le domaine discret, les deux intégrales seront remplacées par des sommations (approximation rectangulaire de l'intégration), donc en pratique:

$$m_{pq} = \sum_i \sum_j i^p j^q I(i,j)$$

Révision
Traitement de haut niveau (3)

- **Aussi, les moments d'ordre 1 sont particulièrement intéressants…:**

$$m_{10} = \sum_i \sum_j i^p I(i,j)$$

$$m_{01} = \sum_i \sum_j j^q I(i,j)$$

- **… entres autres puisqu'ils permettent de calculer le *centre de masse*** $(\bar{i}, \bar{j})$**:**

$$\bar{i} = \frac{\sum_i \sum_j i^p I(i,j)}{\sum_i \sum_j I(i,j)} = \frac{m_{10}}{m_{00}}$$

$$\bar{j} = \frac{\sum_i \sum_j j^q I(i,j)}{\sum_i \sum_j I(i,j)} = \frac{m_{01}}{m_{00}}$$

Révision
Traitement de haut niveau (4)

- **Aussi, les moments d'ordre 2 sont une analogie du moments d'inertie en dynamique:**

$$m_{20} = \sum_i \sum_j i^2 I(i,j)$$

$$m_{02} = \sum_i \sum_j j^2 I(i,j)$$

Rappel, moment d'inertie:

$$J = \sum_i r_i^2 m_i$$

- **Aussi, les moments centraux sont définis par:**

$$\mu_{pq} = \sum_i \sum_j \left(i - \bar{i}\right)^p \left(j - \bar{j}\right)^q I(i,j)$$

- **Rappel:**

$$\bar{i} = \frac{\sum_i \sum_j i^p I(i,j)}{\sum_i \sum_j I(i,j)} = \frac{m_{10}}{m_{00}} \qquad \bar{j} = \frac{\sum_i \sum_j j^q I(i,j)}{\sum_i \sum_j I(i,j)} = \frac{m_{01}}{m_{00}}$$

Révision
Traitement de haut niveau (5)

$$\mu_{pq} = \sum_i \sum_j \left(i - \bar{i}\right)^p \left(j - \bar{j}\right)^q I(i,j)$$

- **Le moment central normalisé:**

$$\frac{\mu_{pq}}{(\mu_{00})^{\gamma}}$$

- **Avec**

$$\gamma = \frac{(p+q)}{2} + 1$$

Révision
Traitement de haut niveau (6)

- **L'ensemble des moments invariants de *Hu*:**

$$
\begin{aligned}
\phi_1 &= \eta_{20} + \eta_{02} \\
\phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\
\phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\
\phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\
\phi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\
&\quad +(3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{30})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\
\phi_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\
\phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\
&\quad +(3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]
\end{aligned}
$$

- **Remarques:**
  - Le premier de ces moments invariants est analogue au moment d'inertie, tel que mentionné plus tôt.
  - Les six premiers moments invariants sont aussi invariant en réflexion, il est donc nécessaire d'utiliser le 7ième si l'on souhaite détecter les images mirroirs.

Révision
Traitement de haut niveau (7)

- Donc, pour chacun des objets que l'on peut retrouver dans une image, on peut calculer la valeur des ϕ_i. Après la segmentation en régions et pour les objets détectés (ici un objet est considéré comme le groupe de pixels ayant la même étiquette), on peut calculer les valeur des ϕ_i et faire une recherche par comparaison avec les ϕ_i des objets de notre base de données.

- De plus, une fois la pièce localisée, nous pouvons calculer l'angle de l'axe principal à l'aide de l'équation:

$$\theta = \frac{1}{2}\operatorname{atan2}\left(2\mu_{11}, \mu_{20} - \mu_{02}\right)$$

L'axe principale est l'axe autour duquel le moment d'inertie (c'est plutôt le moment d'ordre 2) est minimal.

Révision
Traitement de haut niveau (8)

- Une autre méthode pour classer les différents objets d'une image consiste simplement à comparer un à un les objets d'une image avec un (ou plusieurs) patron(s). Le patron est une image m0 X n0 dont l'intensité des pixels est donnée par $T_i(k,j)$ où i désigne l'objet considéré.
  - Pour quantifier la qualité de l'appariemment entre l'objet de l'image et le patron, on utilise un indicateur de performance. Par exemple:

$$p_i(x,y) = \sum_{k=1}^{m0}\sum_{j=1}^{n0}\left|I(x+k,y+j) - T_i(k,j)\right| \qquad \geq 0$$

  - À partir de chaque pixel (x,y) de l'image, on vérifie ainsi la corrélation avec le patron en parcourant une surface m0 X n0.

- ***Avantages:*** -Facile à implenter, s'applique autant pour les images noirs et blancs que pour les tons de gris.
- ***Désavantages:*** -Sensible à l'intensité moyenne, problèmes si l'intensité est à un facteur d'échelle différent.

Révision
Traitement de haut niveau (9)

- Donc remédier à ce dernier désavantage, on peut normaliser l'indicateur de performance en fonction de l'intensité moyenne. Pour ce faire, considérons les normes suivantes:

$$||T_i|| = \left[\sum_{k=1}^{m0}\sum_{j=1}^{n0} T_i^2(k,j)\right]^{\frac{1}{2}}$$

$$||I_{x,y}|| = \left[\sum_{k=1}^{m0}\sum_{j=1}^{n0} I^2(k+x,j+y)\right]^{\frac{1}{2}}$$

- On peut alors définir l'indice de corrélation normalisé:

$$\sigma_i(x,y) = \frac{\sum_{k=1}^{m0}\sum_{j=1}^{n0} I(k+x,j+y)T_i(k,j)}{||T_i|| \cdot ||I_{x,y}||} \quad où \quad 0 \le \sigma_i(x,y) \le 1$$

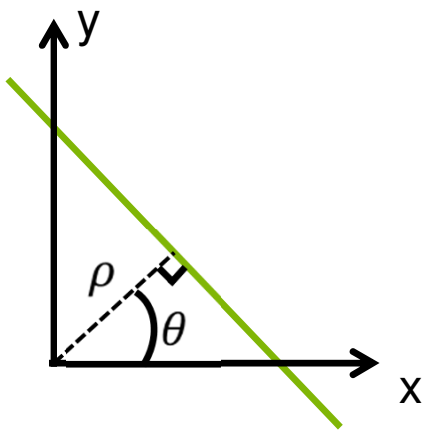
  - Ici, plus le patron et l'objet de l'image seront corrélés, plus l'indicateur de corrélation sera près de 1. Un patron et une pièce identique, mais d'intensité d'échelles différentes résulteront quand même en un indice de corrélation égale à 1.

Révision
Traitement de haut niveau (10)

- Considérons tout d'abord la recherche de segments de droite dans une image.
  - Une droite, comme nous avons vu précedemment peut être paramétrisée:

$$x\cos\theta + y\sin\theta = d$$

  - Où θ est l'angle de la droite et d la distance entre l'origine et le point d'interception perpendiculaire à la droite.



- Comme l'image est de dimension finie et que $0 \leq \theta \leq 2\pi$, on peut simplement discrétiser les paramètres θ et d pour former un tableau 2D.
- Pour chaque combinaison, on vérifie chaque pixel égale à 1 de l'image pour savoir s'il appartient à la droite et, le cas échéant, on enregistre son vote.
- Les cellules du tableau qui possèdent le plus grand nombre de votes caractérisent les droites qui pourraient potentiellement se trouver dans l'image.

Révision
Traitement de haut niveau (11)

- Il est possible de rechercher un cercle de façon similaire en utilisant l'équation:

$$(x - c_x)^2 + (y - c_y)^2 = R^2$$

- On discrétise l'espace des paramètres (R, cx, cy) et on obtient un tableau 3D, le principe est alors le même:
  - Pour chacun des éléments du tableau, on enregistre le nombre de votes, c'est-à-dire le nombre de pixels à 1 appartenant au cercle correspondant à ces valeurs de paramètres.
    - On peut donc identifier les cercles (ou les portions de cercles) dans l'image de cette façon.

Révision
Traitement de haut niveau (12)

- Le modèle de projection orthogonal est le modèle le plus simple que nous pouvons utiliser pour retrouver la pose d'un objet en particuler, par rapport à un repère de référence.
  - La condition par contre pour utiliser un tel modèle est que les objets soient tous situés dans un plan **parallèle** au plan image.

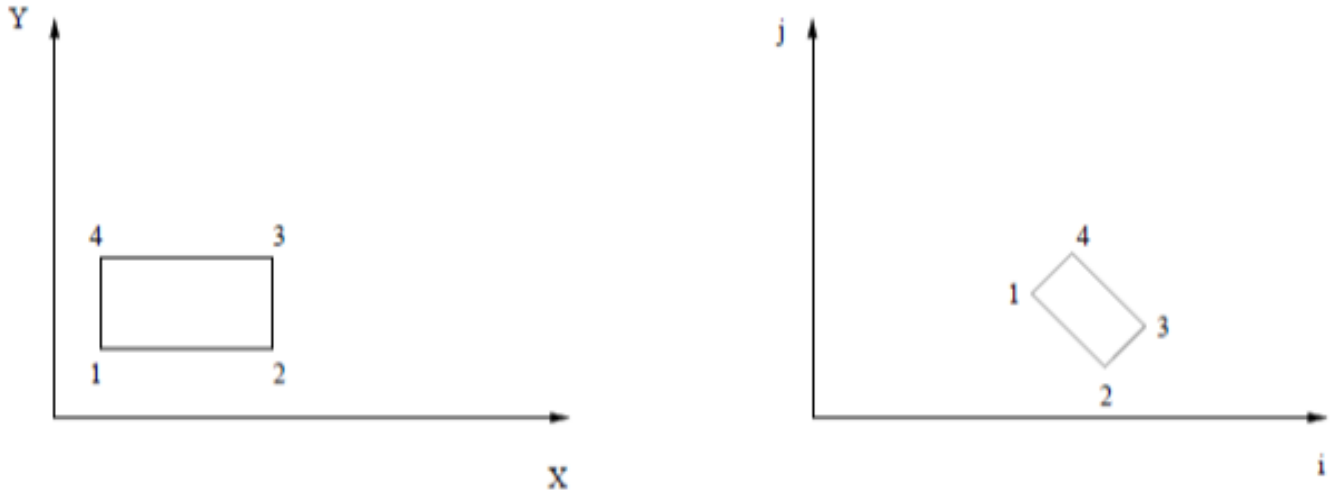
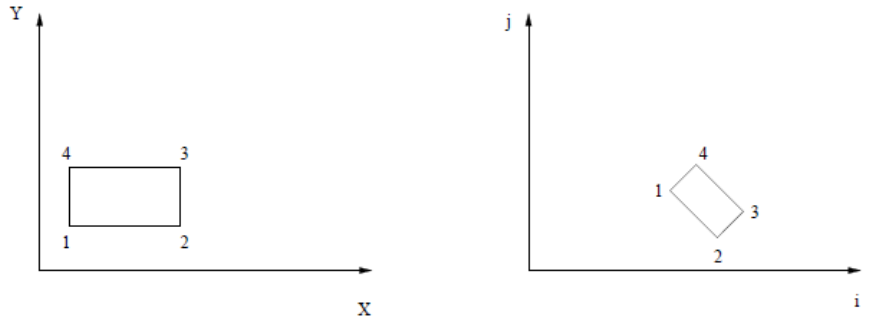


FIG. 3.5 – Objet et sa projection orthogonale

Révision
Traitement de haut niveau (13)



- Dans cette situation, l'objet de l'image est celui du monde réelle ayant subi une **homothétie**, une **rotation** et une **translation**. Ainsi, la transformation entre un point (x,y) et un point (i,j) est donnée par:

$$\begin{bmatrix} i \\ j \\ 1 \end{bmatrix} = \begin{bmatrix} E\cos(\theta) & -E\sin(\theta) & a \\ E\sin(\theta) & E\cos(\theta) & b \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Où E,θ,a et b sont des paramètres à déterminer.

Révision
Traitement de haut niveau (14)

- En utilisant simplement les substitutions c=E*cos(θ) et d=E*sin(θ):

$$\begin{bmatrix} i \\ j \\ 1 \end{bmatrix} = \begin{bmatrix} E\cos(\theta) & -E\sin(\theta) & a \\ E\sin(\theta) & E\cos(\theta) & b \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} i \\ j \\ 1 \end{bmatrix} = \begin{bmatrix} c & -d & a \\ d & c & b \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$i = cx - dy + a$$

$$j = dx + cy + b$$

- Pour un point (x_k,y_k) nous mesurons (i_k,j_k), on peut donc ré-écrire ces équations en fonction des inconnues a,b,c et d:

$$\begin{bmatrix} i_k \\ j_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_k & -y_k \\ 0 & 1 & y_k & x_k \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$$

- Comme il y a 4 inconnues, il faudra 2 points pour déterminer a,b,c et d. Pour plus de deux points, on peut améliorer notre précision dans la mesure en utilisant une solution des moindres carrés.

Révision
Traitement de haut niveau (15)

- Comme il y a 4 inconnues, il faudra 2 points pour déterminer a,b,c et d. Pour plus de deux points, on peut améliorer notre précision dans la mesure en utilisant une solution des moindres carrés (voir notes de cours).
- Une fois les paramètres a,b,c et d identifiés:

$$\begin{bmatrix} i \\ j \\ 1 \end{bmatrix} = \begin{bmatrix} E\cos(\theta) & -E\sin(\theta) & a \\ E\sin(\theta) & E\cos(\theta) & b \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} i \\ j \\ 1 \end{bmatrix} = \begin{bmatrix} c & -d & a \\ d & c & b \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\begin{aligned} i &= cx - dy + a \\ j &= dx + cy + b \end{aligned}$$

- On a simplement qu'à résoudre:

$E = \sqrt{c^2 + d^2}$	$\Rightarrow$	homothétie
$\theta = \text{atan2}(d, c)$	$\Rightarrow$	Angle de la rotation
a	$\Rightarrow$	Translation en x
b	$\Rightarrow$	Translation en y

Révision
Traitement de haut niveau (16)

$$\begin{bmatrix} i \\ j \\ 1 \end{bmatrix} = \begin{bmatrix} E\cos(\theta) & -E\sin(\theta) & a \\ E\sin(\theta) & E\cos(\theta) & b \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} i \\ j \\ 1 \end{bmatrix} = \begin{bmatrix} c & -d & a \\ d & c & b \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

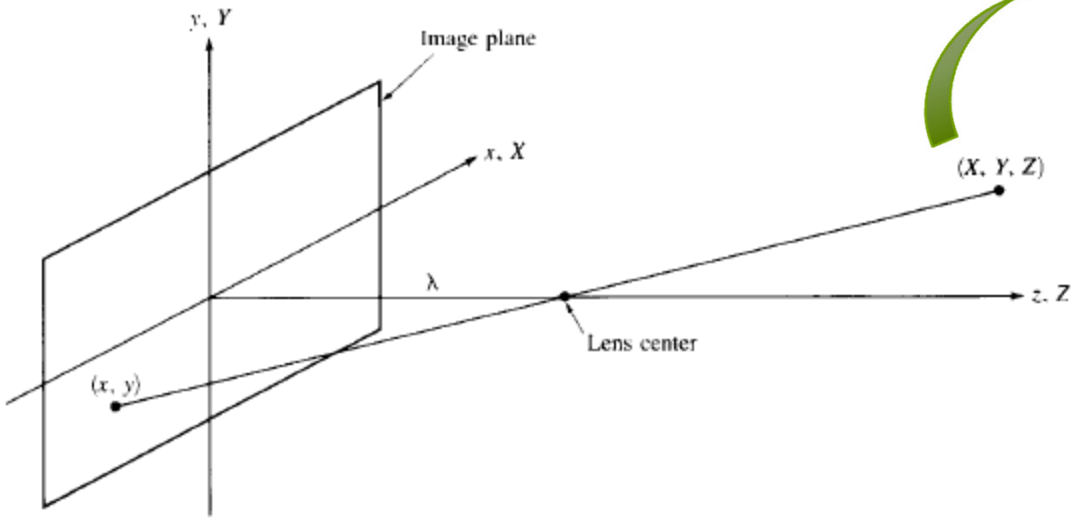
$$\begin{aligned} i &= cx - dy + a \\ j &= dx + cy + b \end{aligned}$$

- Finalement, pour retrouver les coordonnées réelles (x,y) il faut calculer l'inverse de la matrice de transformation homogène (de l'homothétie + rotation + translation). Les coordonnées réelles sont alors données par:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{\cos(\theta)}{E} & \frac{\sin(\theta)}{E} & -\frac{\sin(\theta)b + \cos(\theta)a}{E} \\ -\frac{\sin(\theta)}{E} & \frac{\cos(\theta)}{E} & \frac{\sin(\theta)a - \cos(\theta)b}{E} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ j \\ 1 \end{bmatrix}$$

Révision
Traitement de haut niveau (17)

- Par triangles semblables:



$$\frac{x}{\lambda} = \frac{-X}{Z-\lambda} = \frac{X}{\lambda - Z}$$
$$\Rightarrow x = \frac{\lambda X}{\lambda - Z}$$

$$\frac{y}{\lambda} = \frac{-Y}{Z-\lambda} = \frac{Y}{\lambda - Z}$$
$$\Rightarrow y = \frac{\lambda Y}{\lambda - Z}$$

- Supposons que nous souhaitons représenter cette transformation mathématiquement à l'aide d'une matrice de transformation homogène (comme nous avons fait depuis le début du cours), quelle particularité remarquez-vous ici?

Il s'agit d'une transformation non-linéaire.

Révision
Traitement de haut niveau (18)

- Étant donné cette particularité, la matrice de transformation s'exprime sous une forme à laquelle nous sommes moins habitué, en regard de ce que nous avons vu depuis le début du cours.

- La transformation permettant d'exprimer la transformation entre les points (X,Y,Z) et (x,y) se nomme la *transformation perspective:*

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-1}{\lambda} & 1 \end{bmatrix}$$

Révision
Traitement de haut niveau (19)

- Autre remarque concernant la *transformation perspective:*

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-1}{\lambda} & 1 \end{bmatrix} \quad \text{et} \quad \left(P \ {}^{C}T_{U} \right)$$

- Après multiplication, la matrice de transformation globale aura cette forme:

$$\boldsymbol{P}\,{}^{C}\boldsymbol{T}_{U} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ 0 & 0 & 0 & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

Aussi a_{44} ne sera jamais = 0. En fait, a_{44}=1 pour les transformations ici considérées.

Ainsi,

$$\begin{bmatrix} kx \\ ky \\ 0 \\ k \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ 0 & 0 & 0 & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} \begin{bmatrix} a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Révision
Traitement de haut niveau (20)

- Continuant le développement *transformation perspective:*

$$\left(P\ {}^{C}T_{U}\right)$$

$$\begin{bmatrix} kx \\ ky \\ 0 \\ k \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ 0 & 0 & 0 & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} \begin{bmatrix} a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

- On pourrait démontré ici qu'en multipliant P et T, a_{44} sera toujours égal à 1. Donc, les équations écrites sous forme d'un système:

$$\begin{aligned} a_{11}X + a_{12}Y + a_{13}Z + a_{14} - a_{41}xX - a_{42}xY - a_{43}xZ &= x \\ a_{21}X + a_{22}Y + a_{23}Z + a_{24} - a_{41}yX - a_{42}yY - a_{43}yZ &= y \end{aligned}$$

$$\begin{bmatrix} X_k & Y_k & Z_k & 1 & 0 & 0 & 0 & 0 & -x_kX_k & -x_kY_k & -x_kZ_k \\ 0 & 0 & 0 & 0 & X_k & Y_k & Z_k & 1 & -y_kX_k & -y_kY_k & -y_kZ_k \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{14} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{24} \\ a_{41} \\ a_{42} \\ a_{43} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \end{bmatrix}$$

Révision
Traitement de haut niveau (21)

Pour un point (X_k, Y_k, Z_k) et sa projection (x_k, y_k), on aura

$$\begin{bmatrix} X_k & Y_k & Z_k & 1 & 0 & 0 & 0 & 0 & -x_kX_k & -x_kY_k & -x_kZ_k \\ 0 & 0 & 0 & 0 & X_k & Y_k & Z_k & 1 & -y_kX_k & -y_kY_k & -y_kZ_k \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{14} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{24} \\ a_{41} \\ a_{42} \\ a_{43} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \end{bmatrix}$$

Si $n \geq 6$ points sont disponibles, on peut solutionner au sens des moindres carrés l'équation :

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -x_1X_1 & -x_1Y_1 & -x_1Z_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -y_1X_1 & -y_1Y_1 & -y_1Z_1 \\ X_2 & Y_2 & Z_2 & 1 & 0 & 0 & 0 & 0 & -x_2X_2 & -x_2Y_2 & -x_2Z_2 \\ 0 & 0 & 0 & 0 & X_2 & Y_2 & Z_2 & 1 & -y_2X_2 & -y_2Y_2 & -y_2Z_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -x_nX_n & -x_nY_n & -x_nZ_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -y_nX_n & -y_nY_n & -y_nZ_n \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{14} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{24} \\ a_{41} \\ a_{42} \\ a_{43} \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ \vdots \\ \vdots \\ x_n \\ y_n \end{bmatrix}$$

Révision
Robotique mobile – Généralités (1)

- Lors de la conception d'un robot mobile, plusieurs moyen de locomotion peuvent être choisis afin de rendre possible le déplacement du robot.
  - La plupart du temps, il est intéressant de constater à quel point les mécanismes de locomotion sont inspirés de la nature… En voici quelques-uns (image tirée de [4]).

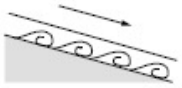
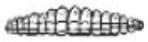
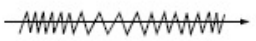
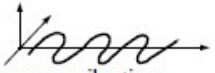
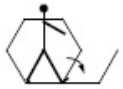
Type of motion	Resistance to motion	Basic kinematics of motion
Flow in a Channel 	Hydrodynamic forces	Eddies 
Crawl 	Friction forces	Longitudinal vibration 
Sliding 	Friction forces	Transverse vibration 
Running 	Loss of kinetic energy	Oscillatory movement of a multi-link pendulum 
Jumping 	Loss of kinetic energy	Oscillatory movement of a multi-link pendulum 
Walking 	Gravitational forces	Rolling of a polygon (see figure 2.2) 

Figure 2.1
Locomotion mechanisms used in biological systems.

126

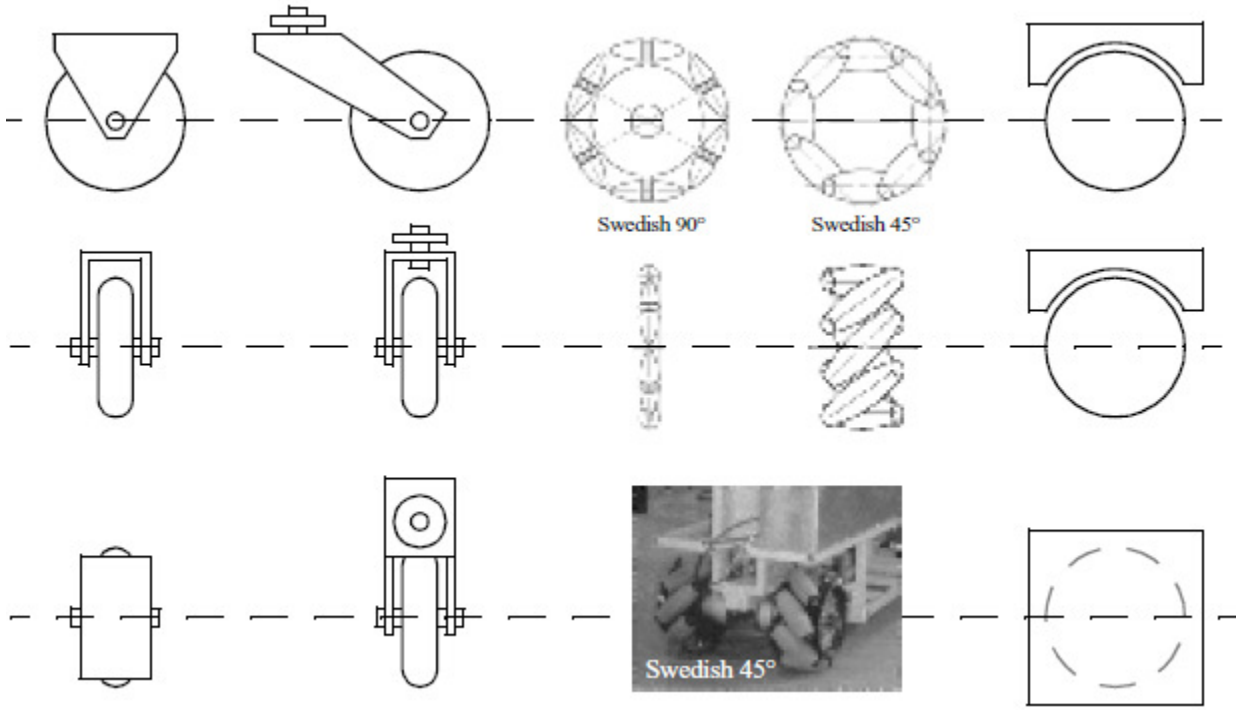
- **Il manque pourtant un moyen de locomotion important, lequel?**

La roue!

(On ne la retrouve pas tel quel dans la nature!)

Révision
Robotique mobile – Généralités (2)

- Dans le cadre de ce module sur les robots mobiles, nous nous attarderons surtout sur les robots mobiles qui se déplacent à l'aide de roues.
  - Plusieurs types de roues existent cependant:

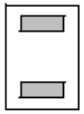
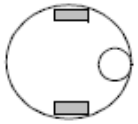
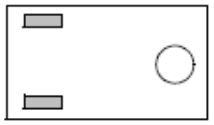
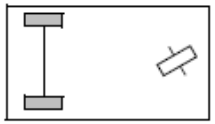
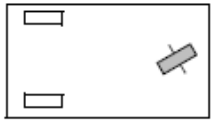
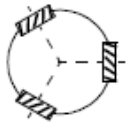
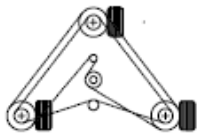


Aussi, tant qu'à parler de type de roue, pour vous ouvrir l'esprit:

http://www.radio-canada.ca/emissions/decouverte/2011-2012/Reportage.asp?idDoc=211681

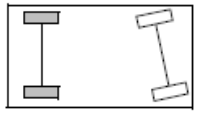
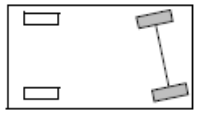
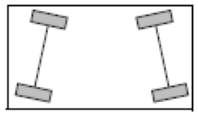
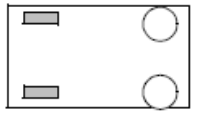
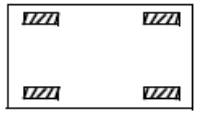
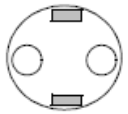
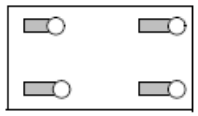
Révision
Robotique mobile – Généralités (3)

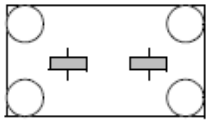
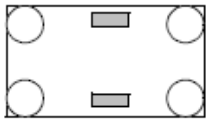
- Non seulement existe-t-il plusieurs types de roues, il existe aussi plusieurs géométries de roue (image) adapté de [4]:

# of wheels	Arrangement	Description	Typical examples
2		One steering wheel in the front, one traction wheel in the rear	Bicycle, motorcycle
		Two-wheel differential drive with the center of mass (COM) below the axle	Cye personal robot
3		Two-wheel centered differential drive with a third point of contact	Nomad Scout, smartRob EPFL
		Two independently driven wheels in the rear/front, 1 unpowered omnidirectional wheel in the front/rear	Many indoor robots, including the EPFL robots Pygmalion and Alice
		Two connected traction wheels (differential) in rear, 1 steered free wheel in front	Piaggio minitrucks
		Two free wheels in rear, 1 steered traction wheel in front	Neptune (Carnegie Mellon University), Hero-1
		Three motorized Swedish or spherical wheels arranged in a triangle; omnidirectional movement is possible	Stanford wheel Tribolo EPFL, Palm Pilot Robot Kit (CMU)
		Three synchronously motorized and steered wheels; the orientation is not controllable	"Synchro drive" Denning MRV-2, Georgia Institute of Technology, I-Robot B24, Nomad 200

Révision
Robotique mobile – Généralités (4)

- Géométries de roues (suite)

4		Two motorized wheels in the rear, 2 steered wheels in the front; steering has to be different for the 2 wheels to avoid slipping/skidding.	Car with rear-wheel drive
		Two motorized and steered wheels in the front, 2 free wheels in the rear; steering has to be different for the 2 wheels to avoid slipping/skidding.	Car with front-wheel drive
		Four steered and motorized wheels	Four-wheel drive, four-wheel steering Hyperion (CMU)
		Two traction wheels (differential) in rear/front, 2 omnidirectional wheels in the front/rear	Charlie (DMT-EPFL)
		Four omnidirectional wheels	Carnegie Mellon Uranus
		Two-wheel differential drive with 2 additional points of contact	EPFL Khepera, Hyperbot Chip
		Four motorized and steered castor wheels	Nomad XR4000

# of wheels	Arrangement	Description	Typical examples
6		Two motorized and steered wheels aligned in center, 1 omnidirectional wheel at each corner	First
		Two traction wheels (differential) in center, 1 omnidirectional wheel at each corner	Terregator (Carnegie Mellon University)

Icons for the each wheel type are as follows:	
	unpowered omnidirectional wheel (spherical, castor, Swedish);
	motorized Swedish wheel (Stanford wheel);
	unpowered standard wheel;
	motorized standard wheel;
	motorized and steered castor wheel;
	steered standard wheel;
	connected wheels.

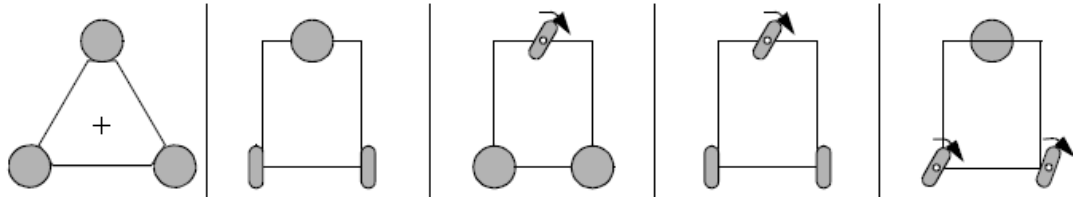
Révision
Robotique mobile – Généralités (5)

- **<u>Donnons ici quelques définitions:</u>**

- Le **degré de mobilité** d'un robot mobile est le nombre de variables de la pose sur lesquelles il est possible d'agir simplement en changeant la vitesse des roues.

- Le **degré de directionnalité** (*steerability*) d'un robot mobile est le nombre de variables supplémentaires de la pose sur lesquelles vous pouvez agir après avoir changer les angles des roues pivotantes.

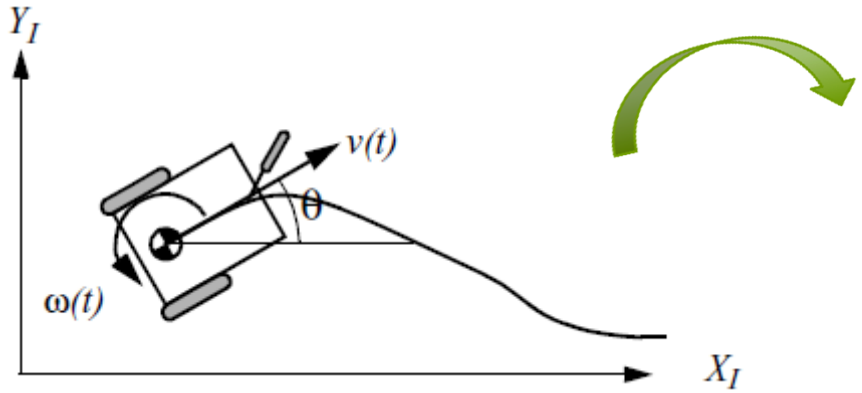
- Le **degré de manoeuvrabilité** est simplement (**le degré de mobilité + le degré de directionnalité**).

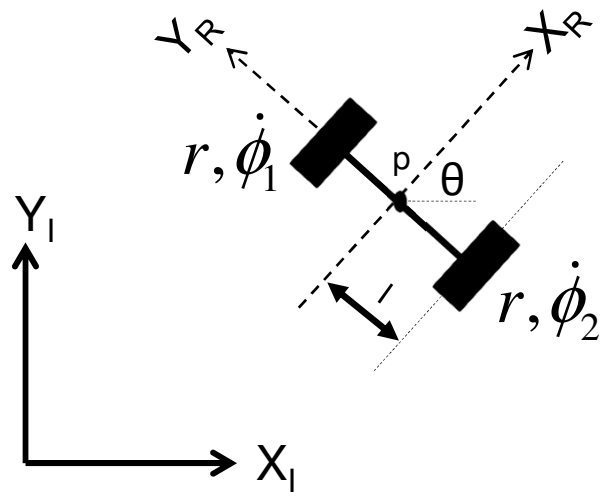


Omnidirectional	Differential	Omni-Steer	Tricycle	Two-Steer
δ_M =3	δ_M =2	δ_M =3	δ_M =2	δ_M =3
δ_m =3	δ_m =2	δ_m =2	δ_m =1	δ_m =1
δ_s =0	δ_s =0	δ_s =1	δ_s =1	δ_s =2

Révision
Robotique mobile – Cinématique (1)

- Considérons un premier exemple:





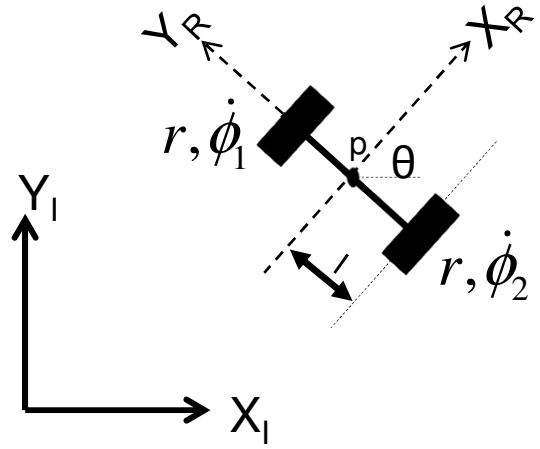
- Lorsque nous travaillons dans le contexte de la robotique mobile, le problème de la cinématique directe réfère plutôt à la question: Étant donné la géométrie du robot et la vitesse de ses roues, comment le robot se déplace-t-il?
  - Pour le robot ci-dessus, plus particulièrement, le problème se résume à répondre à la question: Étant donné les vitesses des deux roues de ce robot et sa géométrie, quel sera la vitesse de sa pose exprimée dans le repère de référence?

Révision
Robotique mobile – Cinématique (2)

- Nous avons déjà mentionné que:

$$\dot{\xi}_I = \underbrace{\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{{}^{I}T_R} \dot{\xi}_R = \left({}^{R}T_I\right)^T \dot{\xi}_R$$

- Nous supposons ici que nous connaissons l'orientation initiale θ.
  - Notez que la matrice de transformation ITR changera au fur et à mesure que le robot se déplacera. Pour mettre à jour la valeur de θ, on pourrait simplement ajouter l'intégrale de la vitesse angulaire
    - Nous verrons comment on peut le faire encore plus efficacement à l'aide du calcul de l'odométrie basé sur la fusion de plusieurs capteurs.
- La démarche consiste donc tout d'abord à déterminer la contribution de chacune des deux roues sur la pose exprimée dans le repère local (repère *R*):



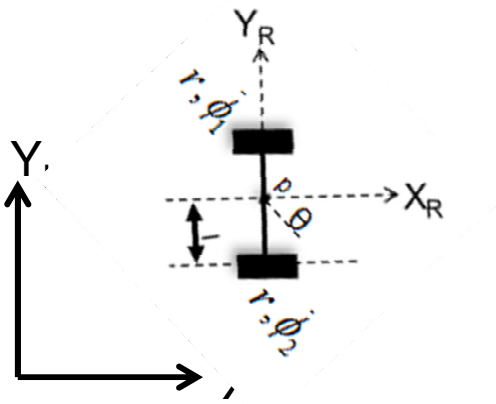
- Pour ce faire, nous imaginons que θ = 0 et calculons la contribution de chacune des roues sur la vitesse de déplacement en x, en y et sur la vitesse de rotation.

Révision
Robotique mobile – Cinématique (3)

- Étant donné la géométrie simple de ce robot, la contribution des roues sur la vitesse de translation et d'orientation est assez simple à calculer.
  - Pour la roue 1, sa vitesse en m/s est donnée par:

    $v_1 = r\dot{\phi}_1$ où r est le rayon de la roue en mètre et $\dot{\phi}$ la vitesse angulaire de la roue en $^{rad}/_{sec}$
  - Étant donné que le point P se situe en plein milieu des deux roues, la contribution de la roue 1 pour la vitesse en translation selon l'axe x du repère local sera la moitié de la vitesse de la roue 1:

$$\dot{x}_{r1} = \frac{r\dot{\phi}_1}{2}$$



- Le même raisonnement pour la roue 2 nous mène aussi à: $\dot{x}_{r2} = -\frac{r\dot{\phi}_2}{2}$

Attention aux signes!

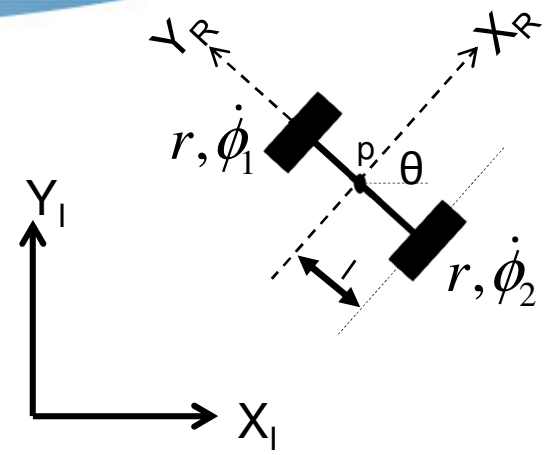
- Clairement, aucune des roues n'amènent de contribution selon l'axe y du repère local!
- Finalement la contribution de chacune des roues sur la vitesse de l'orientation:

$$\dot{\theta} = -\frac{r\dot{\phi}_1}{2l} - \frac{r\dot{\phi}_2}{2l}$$

Révision
Robotique mobile – Cinématique (4)

- Donc, en résumé:

$$\dot{\xi} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{r\dot{\phi}_1}{2} - \frac{r\dot{\phi}_2}{2} \\ 0 \\ -\frac{r\dot{\phi}_1}{2l} - \frac{r\dot{\phi}_2}{2l} \end{bmatrix}$$



- **La cinématique directe de ce robot est donc donnée directement par:**

$$\dot{\xi}_I = \underbrace{\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{{}^{I}T_R} \begin{bmatrix} \frac{r\dot{\phi}_1}{2} - \frac{r\dot{\phi}_2}{2} \\ 0 \\ -\frac{r\dot{\phi}_1}{2l} - \frac{r\dot{\phi}_2}{2l} \end{bmatrix}$$

- **Il s'agit de la vitesse (de la pose) du robot, exprimée dans le repère universel en fonction de la vitesse de chacun de ces moteurs.**

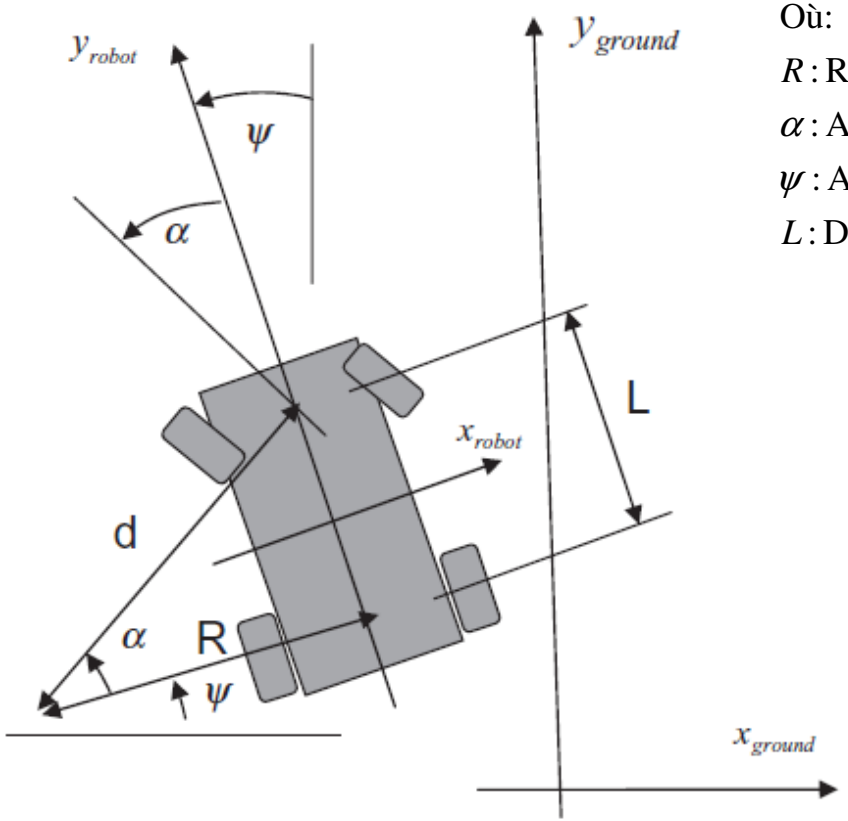
- **De plus (rappel), on pourrait calculer θ simplement de cette façon:**

$$\theta(t) = \theta_0 + \int_{t_0}^{t} \dot{\theta}(t)\, dt$$

***Quelle sera une des sources d'erreur considérable si l'on se base sur ce calcul pour déterminer θ?*

Révision
Robotique mobile – Cinématique (5)

- Avant, d'aller plus loin, considérons un autre exemple de robot mobiles:
  - ***Image tirée de [1]*



Où:

R : Rayon de courbure instantané

α : Angle de la direction p-r-à l'axe longitudinal (y) du robot

ψ : Angle de l'axe longitudinal du robot p-r-à l'axe y du repère de référence

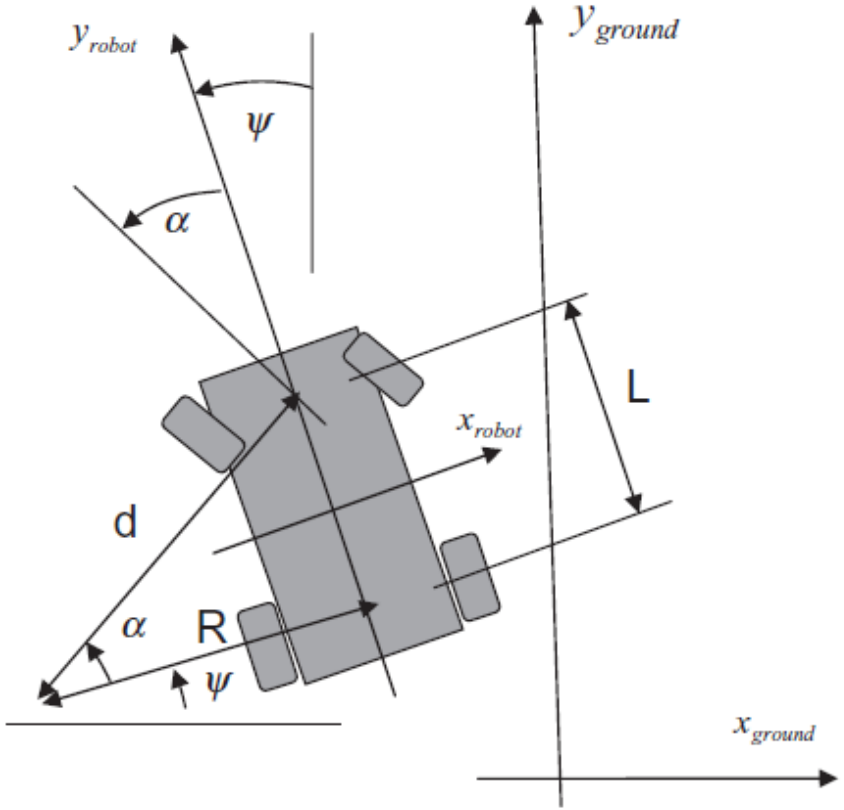
L : Distance entre l'axe de rotation des roues arrières et devant lorsque α=0

- Les roues arrières propulsent le véhicle, tandis que les roues avant le dirigent.
- Par simple géométrie, nous savons que:

$$\tan(\alpha) = \frac{L}{R} \quad \Rightarrow \quad R = \frac{L}{\tan(\alpha)}$$

Révision
Robotique mobile – Cinématique (6)

- Suite:
  - **Image tirée de [1]



- Aussi par simple constatation:

$$v_{roues\_arr} = R\frac{d\psi}{dt} = R\dot{\psi} \quad \Rightarrow \quad \dot{\psi} = \frac{v_{roues\_arr}}{R}$$

- En substituant pour le rayon de courbure instantanné:

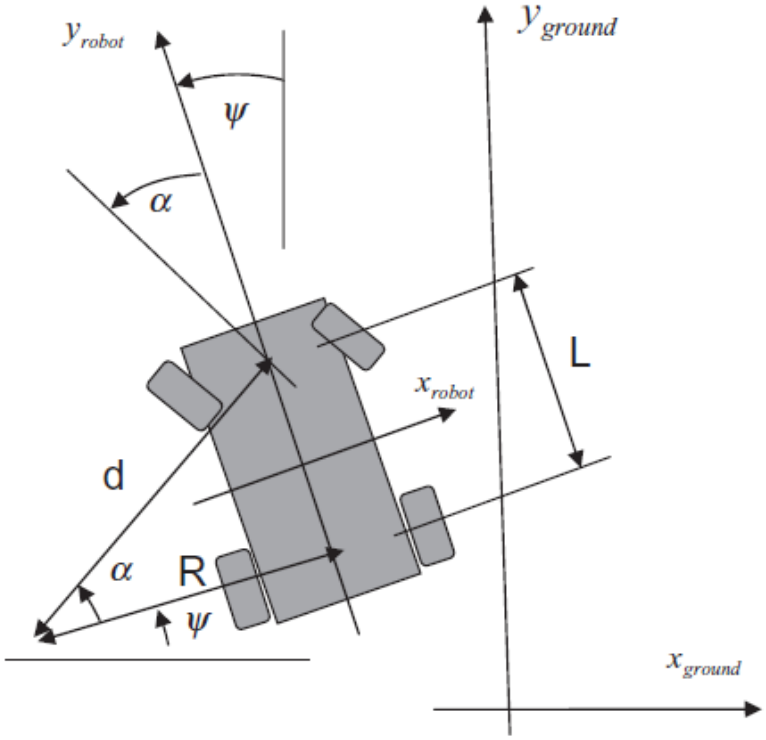
$$\dot{\psi} = \frac{v_{roues\_arr}}{L/\tan(\alpha)} = \frac{v_{roues\_arr}}{L}\tan(\alpha)$$

- Finalement, on remarque directement la contribution des roues arrières sur le déplacement en translation exprimé dans le repère du robot:

$$\dot{x}_{robot} = 0$$
$$\dot{y}_{robot} = v_{roues\_arr}$$

Révision
Robotique mobile – Cinématique (7)

- Suite:
  - ***Image tirée de [1]*



- On connait donc la vitesse de la pose exprimée dans le repère du robot, c'est-à-dire:

$$\dot{\xi}_R = \begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\psi}_R \end{bmatrix} = \begin{bmatrix} 0 \\ v_{roues\_arr} \\ \frac{v_{roues\_arr}}{L}\tan(\alpha) \end{bmatrix}$$

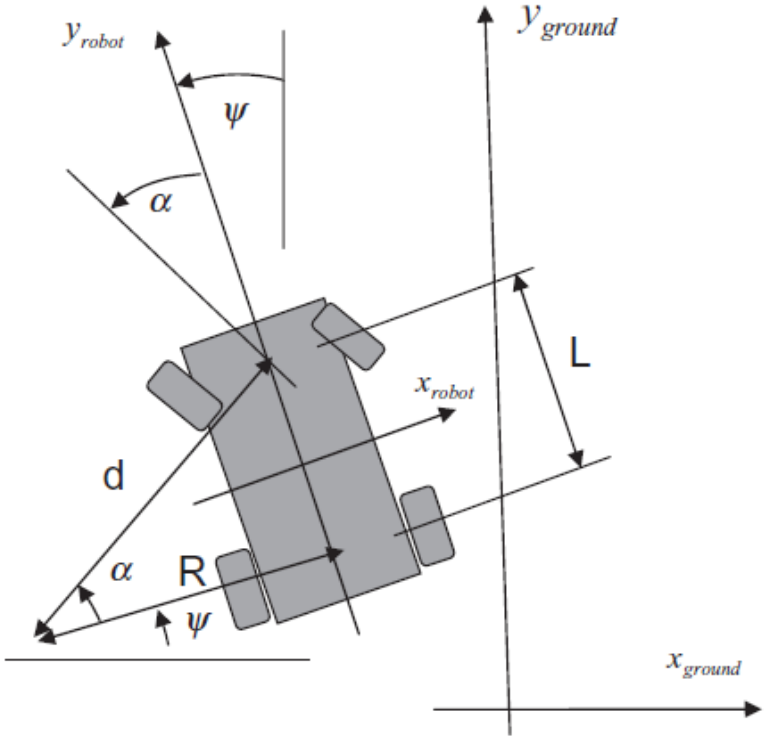
  - Donc, pour transformer la vitesse de la pose du robot dans le repère de référence (universel):

$$\dot{\xi}_R = \underbrace{\begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{{}^{R}T_{ground}} \dot{\xi}_{ground} \quad \Rightarrow \quad \dot{\xi}_{ground} = \left({}^{R}T_{ground}\right)^{-1} \dot{\xi}_R$$

$$\dot{\xi}_{ground} = \underbrace{\begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{{}^{ground}T_{R}} \begin{bmatrix} 0 \\ v_{roues\_arr} \\ \frac{v_{roues\_arr}}{L}\tan(\alpha) \end{bmatrix} = \begin{bmatrix} -\sin(\psi)v_{roues\_arr} \\ \cos(\psi)v_{roues\_arr} \\ \frac{v_{roues\_arr}}{L}\tan(\alpha) \end{bmatrix}$$

Révision
Robotique mobile – Cinématique (8)

- Suite:
  - ***Image tirée de [1]*



- Vous avez donc obtenu le modèle cinématique du robot mobile:

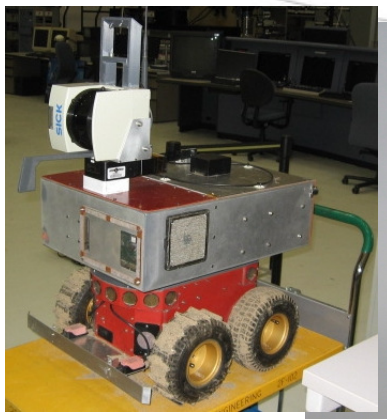
$$\dot{\xi}_{ground} = \begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\psi}_R \end{bmatrix} = \begin{bmatrix} -\sin(\psi)\, v_{roues\_arr} \\ \cos(\psi)\, v_{roues\_arr} \\ \dfrac{v_{roues\_arr}}{L}\tan(\alpha) \end{bmatrix}$$

- **Quelles variables (variables d'entrées) proposeriez-vous pour contrôler ce véhicule?**
- **Notez qu'en fait, la seule variable sur laquelle vous ne pouvez pas directement agir est l'angle Ψ. Cependant, nous en avons besoin pour résoudre la cinématique!**
- **Encore une fois, nous pourrions penser à utiliser :**

$$\psi = \psi_0 + \int_{t_0}^{\sigma} \dot{\psi}(t)\, dt$$

- **Mais ce n'est certainement pas optimal!**
- **Une meilleure façon de faire serait d'utiliser un *filtre de Kalman* pour fusionner l'information des capteurs disponibles!**

Révision
Robotique mobile – Trajectoires (1)



Révision
Robotique mobile – Trajectoires (2)

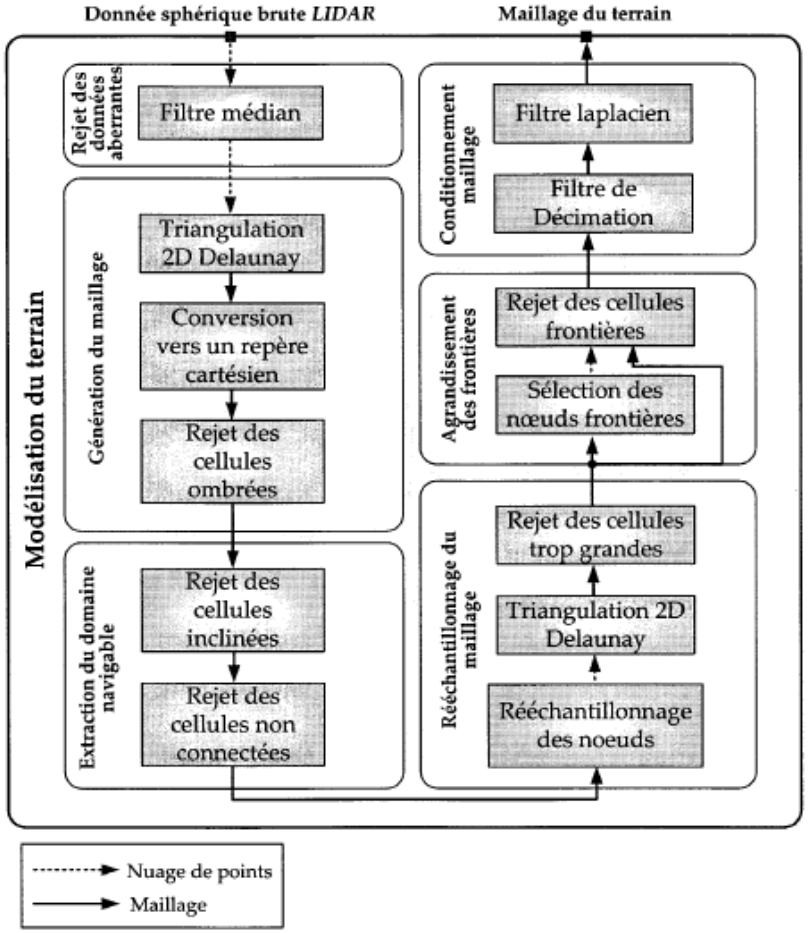
- La planification de trajectoire dans le contexte de la robotique mobile autonome se fait très différemment que la planification de trajectoire pour les robots sériels fixes.

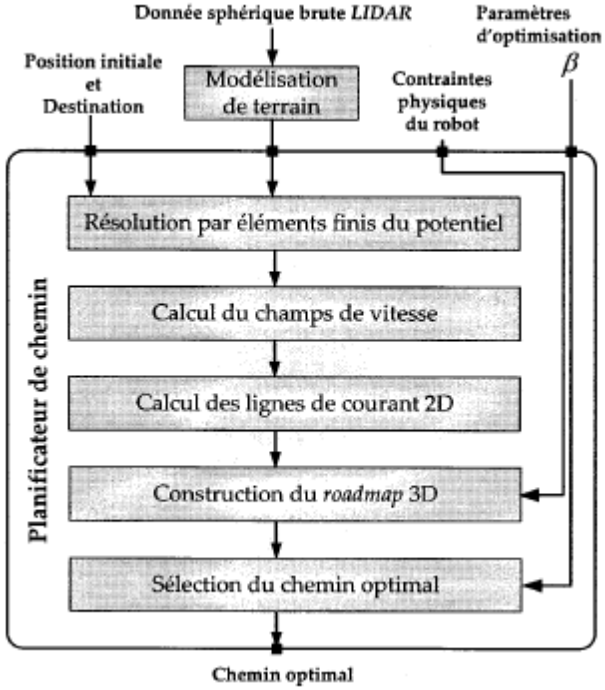
Révision
Robotique mobile – Trajectoires (3)

- En effet, étant donné que le robot se déplace et que l'environnement n'est pas connu *a priori*, il faut user de techniques fort différentes.
  - Entres autres, il faut utiliser des capteurs pour saisir les formes de l'environnement à traver lequel le robot se déplacera.
  - Il faut tenter de localiser le robot dans un repère de référence.
  - Il faut éviter les obstacles qui ne sont pas non plus connus *a priori*
  - La minimisation de l'énergie revêt ici une importance grandiose(!) en général
  - Il faut enfin créer une trajectoire navigable pour le robot mobile.
    - Ensuite un contrôleur se chargera de commander le robot afin qu'il assure un bon suivi de trajectoire.

Révision
Robotique mobile – Trajectoires (4)

- Par exemple, voici *grosso modo* une technique que nous développons à l'ASC:



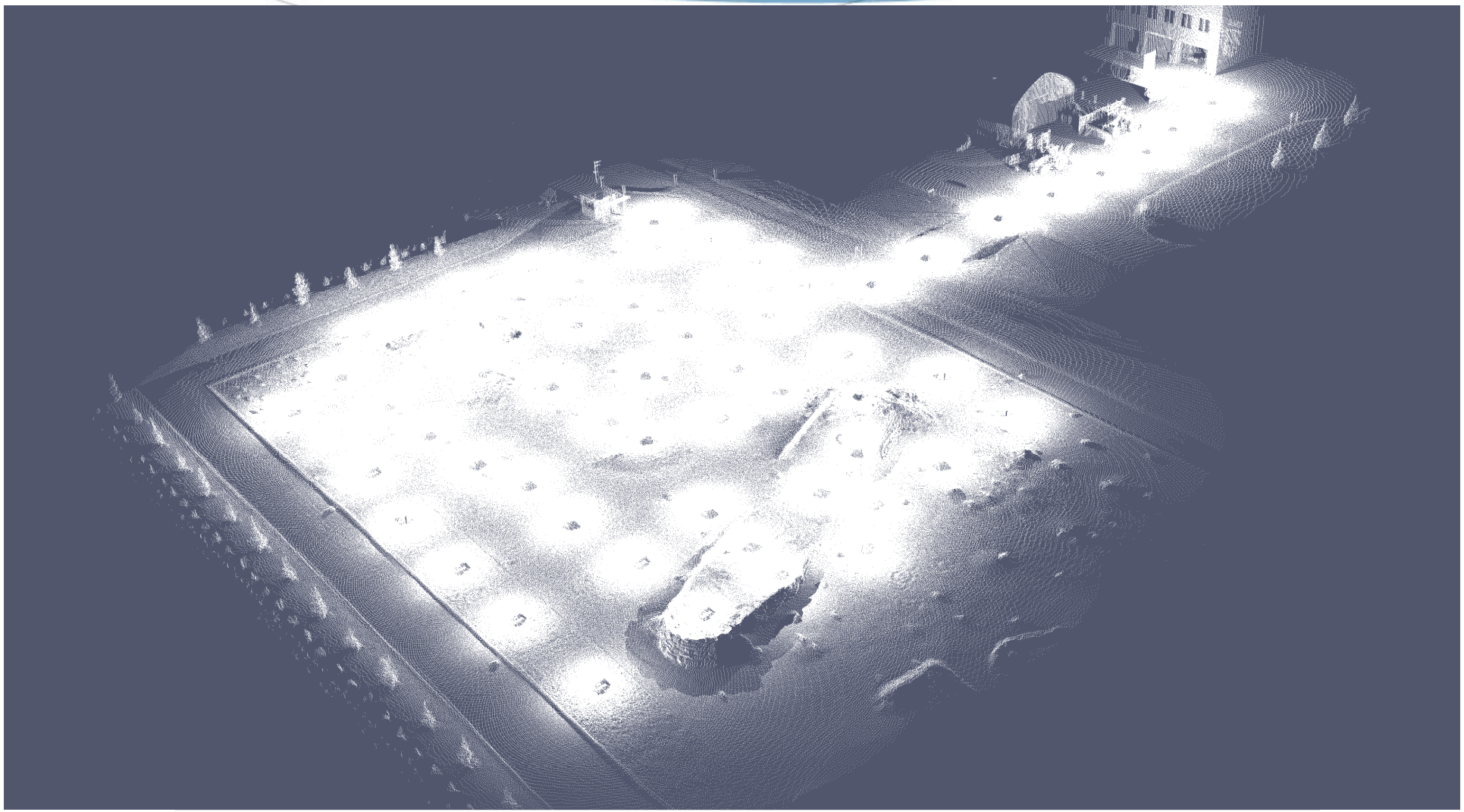


**Ces figures sont tirés du mémoire de maîtrise de David Gingras:
Planification de chemins pour robot mobile explorateur de planète, 2010

Révision
Robotique mobile – Trajectoires (5)

- Le “Terrain de Mars”, laboratoire extérieur sur lequel nous effectuons nos expériences:

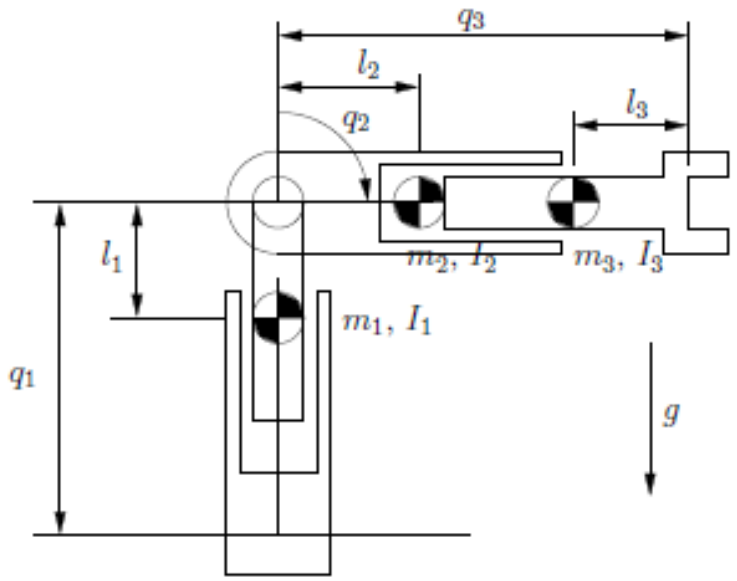
Révision
Robotique mobile – Trajectoires (6)



EXERCICES

Révision
Exercices – Dynamique (1)

Considérons le robot PRP de la figure ci-dessous.



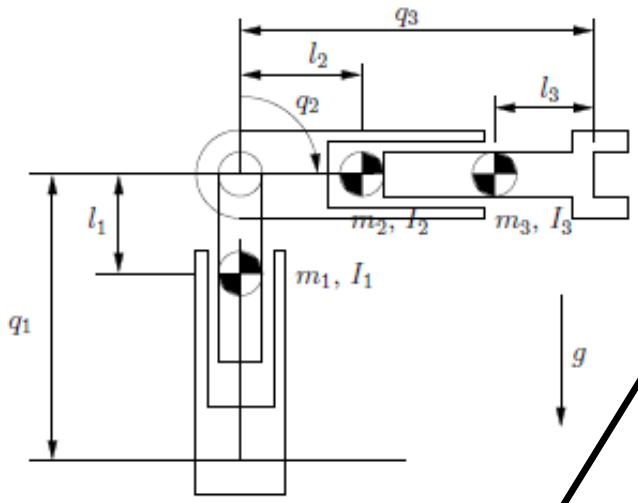
Les positions des centres de masse sont données par

$$
\begin{aligned}
x_1 &= 0 \\
y_1 &= q_1 - l_1 \\
x_2 &= l_2 \sin(q_2) \\
y_2 &= q_1 + l_2 \cos(q_2) \\
x_3 &= (q_3 - l_3) \sin(q_2) \\
y_3 &= q_1 + (q_3 - l_3) \cos(q_2)
\end{aligned}
$$

Calculer : a) le lagrangien de ce robot b) l'équation de la dynamique pour τ_2.

Révision
Exercices – Dynamique (1)

Considérons le robot PRP de la figure ci-dessous.



Les positions des centres de masse sont données par

$$\begin{aligned} x_1 &= 0 \\ y_1 &= q_1 - l_1 \\ x_2 &= l_2 \sin(q_2) \\ y_2 &= q_1 + l_2 \cos(q_2) \\ x_3 &= (q_3 - l_3)\sin(q_2) \\ y_3 &= q_1 + (q_3 - l_3)\cos(q_2) \end{aligned}$$

Calculer : a) le lagrangien de ce robot b) l'équation de la dynamique pour τ_2.

$$\begin{aligned} \dot{x}_1 &= 0 \\ \dot{y}_1 &= \dot{q}_1 \\ \dot{x}_2 &= l_2\dot{q}_2\cos(q_2) \\ \dot{y}_2 &= \dot{q}_1 - l_2\dot{q}_2\sin(q_2) \\ \dot{x}_3 &= \dot{q}_3\sin(q_2) + (q_3 - l_3)\dot{q}_2\cos(q_2) \\ \dot{y}_3 &= \dot{q}_1 + \dot{q}_3\cos(q_2) - (q_3 - l_3)\dot{q}_2\sin(q_2) \end{aligned}$$

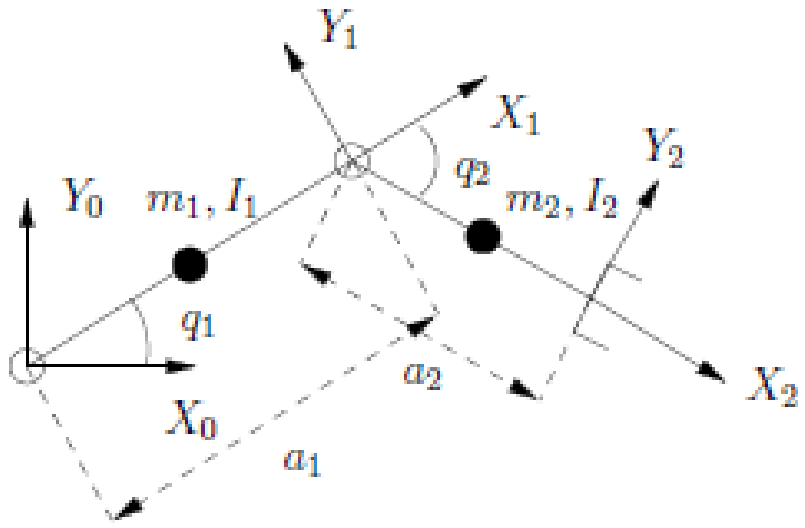
$$\begin{aligned} v_1^2 &= \dot{q}_1^2 \\ v_2^2 &= \dot{q}_1^2 + l_2^2\dot{q}_2^2 - 2l_2\dot{q}_1\dot{q}_2\sin(q_2) \\ v_3^3 &= \dot{q}_1^2 + (q_3 - l_3)^2\dot{q}_2^2 + \dot{q}_3^2 - 2\dot{q}_1(q_3 - l_3)\dot{q}_2\sin(q_2) + 2\dot{q}_1\dot{q}_3\cos(q_2) \end{aligned}$$

$$\begin{aligned} T_1 &= \frac{1}{2}m_1\dot{q}_1^2 \\ T_2 &= \frac{1}{2}m_2\left[\dot{q}_1^2 + l_2^2\dot{q}_2^2 - 2l_2\dot{q}_1\dot{q}_2\sin(q_2)\right] + \frac{1}{2}I_2\dot{q}_2^2 \\ T_3 &= \frac{1}{2}m_3\left[\dot{q}_1^2 + (q_3 - l_3)^2\dot{q}_2^2 + \dot{q}_3^2 - 2\dot{q}_1(q_3 - l_3)\dot{q}_2\sin(q_2) + 2\dot{q}_1\dot{q}_3\cos(q_2)\right] + \frac{1}{2}I_3\dot{q}_2^2 \end{aligned}$$

$$\begin{aligned} L &= \frac{1}{2}\left[m_1 + m_2 + m_3\right]\dot{q}_1^2 + \frac{1}{2}m_2\left[l_2^2\dot{q}_2^2 - 2l_2\dot{q}_1\dot{q}_2\sin(q_2)\right] + \frac{1}{2}(I_2 + I_3)\dot{q}_2^2 \\ &\quad + \frac{1}{2}m_3\left[(q_3 - l_3)^2\dot{q}_2^2 + \dot{q}_3^2 - 2\dot{q}_1(q_3 - l_3)\dot{q}_2\sin(q_2) + 2\dot{q}_1\dot{q}_3\cos(q_2)\right] \\ &\quad - m_1g(q_1 - l_1) - m_2g[q_1 + l_2\cos(q_2)] - m_3g[q_1 + (q_3 - l_3)\cos(q_2)] \\ \frac{\partial L}{\partial \dot{q}_2} &= m_2l_2^2\dot{q}_2 - m_2l_2\dot{q}_1\sin(q_2) + (I_2 + I_3)\dot{q}_2 + m_3\left[(q_3 - l_3)^2\dot{q}_2 - \dot{q}_1(q_3 - l_3)\sin(q_2)\right] \\ \frac{\partial L}{\partial q_2} &= -m_2l_2\dot{q}_1\dot{q}_2\cos(q_2) - m_3\left[\dot{q}_1(q_3 - l_3)\dot{q}_2\cos(q_2) + \dot{q}_1\dot{q}_3\sin(q_2)\right] \\ &\quad + [m_2l_2 + m_3(q_3 - l_3)]g\sin(q_2) \\ \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_2}\right) &= [m_2l_2^2 + I_2 + I_3 + m_3(q_3 - l_3)^2]\ddot{q}_2 - m_2l_2\ddot{q}_1\sin(q_2) - m_2l_2\dot{q}_1\dot{q}_2\cos(q_2) \\ &\quad + m_3\left[2(q_3 - l_3)\dot{q}_3\dot{q}_2 - \ddot{q}_1(q_3 - l_3)\sin(q_2) - \dot{q}_1\dot{q}_3\sin(q_2) - \dot{q}_1(q_3 - l_3)\dot{q}_2\cos(q_2)\right] \\ \tau_2 &= [m_2l_2^2 + I_2 + I_3 + m_3(q_3 - l_3)^2]\ddot{q}_2 - m_2l_2\ddot{q}_1\sin(q_2) \\ &\quad + m_3\left[2(q_3 - l_3)\dot{q}_3\dot{q}_2 - \ddot{q}_1(q_3 - l_3)\sin(q_2)\right] - [m_2l_2 + m_3(q_3 - l_3)]g\sin(q_2) \end{aligned}$$

Révision
Exercices – Dynamique (2)

L'effecteur du robot RR de la figure ci-dessous doit se déplacer le long d'une trajectoire définie dans le plan XY.



Le modèle cinématique direct est donnée par

$$\begin{bmatrix} C_{12} & -S_{12} & 0 & a_2 C_{12} + C_1 a_1 \\ S_{12} & C_{12} & 0 & a_2 S_{12} + S_1 a_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Le modèle cinématique inverse du robot pour une configuration *coude en haut* est donné par

$$\begin{aligned} q_2 &= \text{atan2}(-\sqrt{1-K^2}, K) \\ K &= \frac{p_x^2 + p_y^2 - a_2^2 - a_1^2}{2a_1 a_2} \\ q_1 &= \text{atan2}(p_y, p_x) - \text{atan2}(S_2 a_2, C_2 a_2 + a_1) \end{aligned}$$

et, pour dx et dy, la matrice jacobienne est donnée par

$$\begin{aligned} {}^0\boldsymbol{J} &= \begin{bmatrix} -a_2 S_{12} - S_1 a_1 & -a_2 S_{12} \\ a_2 C_{12} + C_1 a_1 & a_2 C_{12} \end{bmatrix} \\ {}^0\boldsymbol{J}^{-1} &= \frac{1}{a_1 a_2 S_2} \begin{bmatrix} a_2 C_{12} & a_2 S_{12} \\ -a_2 C_{12} - C_1 a_1 & -a_2 S_{12} - S_1 a_1 \end{bmatrix} \end{aligned}$$

On a les valeurs suivantes : $a_1 = 1.0m$ et $a_2 = 0.5m$.

Au temps $t = 1$sec, la position désirée est $x_d(1) = 0.5m$, $y_d(1) = 0.5m$; la vitesse désirée est $\dot{x}_d(1) = 1m/s$, $\dot{y}_d(1) = 1m/s$; et l'accélération désirée est $\ddot{x}_d(1) = -1m/s^2$, $\ddot{y}_d(1) = -1m/s^2$.

Calculer $\boldsymbol{q}_d(1)$, $\dot{\boldsymbol{q}}_d(1)$ et $\ddot{\boldsymbol{q}}_d(1)$.

Révision
Exercices – Dynamique (2)

Le modèle cinématique direct est donnée par

$$\begin{bmatrix} C_{12} & -S_{12} & 0 & a_2 C_{12} + C_1 a_1 \\ S_{12} & C_{12} & 0 & a_2 S_{12} + S_1 a_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Le modèle cinématique inverse du robot pour une configuration *coude en haut* est donné par

$$\begin{aligned} q_2 &= \text{atan2}(-\sqrt{1-K^2}, K) \\ K &= \frac{p_x^2 + p_y^2 - a_2^2 - a_1^2}{2a_1 a_2} \\ q_1 &= \text{atan2}(p_y, p_x) - \text{atan2}(S_2 a_2, C_2 a_2 + a_1) \end{aligned}$$

et, pour dx et dy, la matrice jacobienne est donnée par

$$\begin{aligned} {}^0\boldsymbol{J} &= \begin{bmatrix} -a_2 S_{12} - S_1 a_1 & -a_2 S_{12} \\ a_2 C_{12} + C_1 a_1 & a_2 C_{12} \end{bmatrix} \\ {}^0\boldsymbol{J}^{-1} &= \frac{1}{a_1 a_2 S_2} \begin{bmatrix} a_2 C_{12} & a_2 S_{12} \\ -a_2 C_{12} - C_1 a_1 & -a_2 S_{12} - S_1 a_1 \end{bmatrix} \end{aligned}$$

On a les valeurs suivantes : $a_1 = 1.0m$ et $a_2 = 0.5m$.
Au temps $t = 1$sec, la position désirée est $x_d(1) = 0.5m$, $y_d(1) = 0.5m$; la vitesse désirée est $\dot{x}_d(1) = 1m/s$, $\dot{y}_d(1) = 1m/s$; et l'accélération désirée est $\ddot{x}_d(1) = -1m/s^2$, $\ddot{y}_d(1) = -1m/s^2$.
Calculer $\boldsymbol{q}_d(1)$, $\dot{\boldsymbol{q}}_d(1)$ et $\ddot{\boldsymbol{q}}_d(1)$.

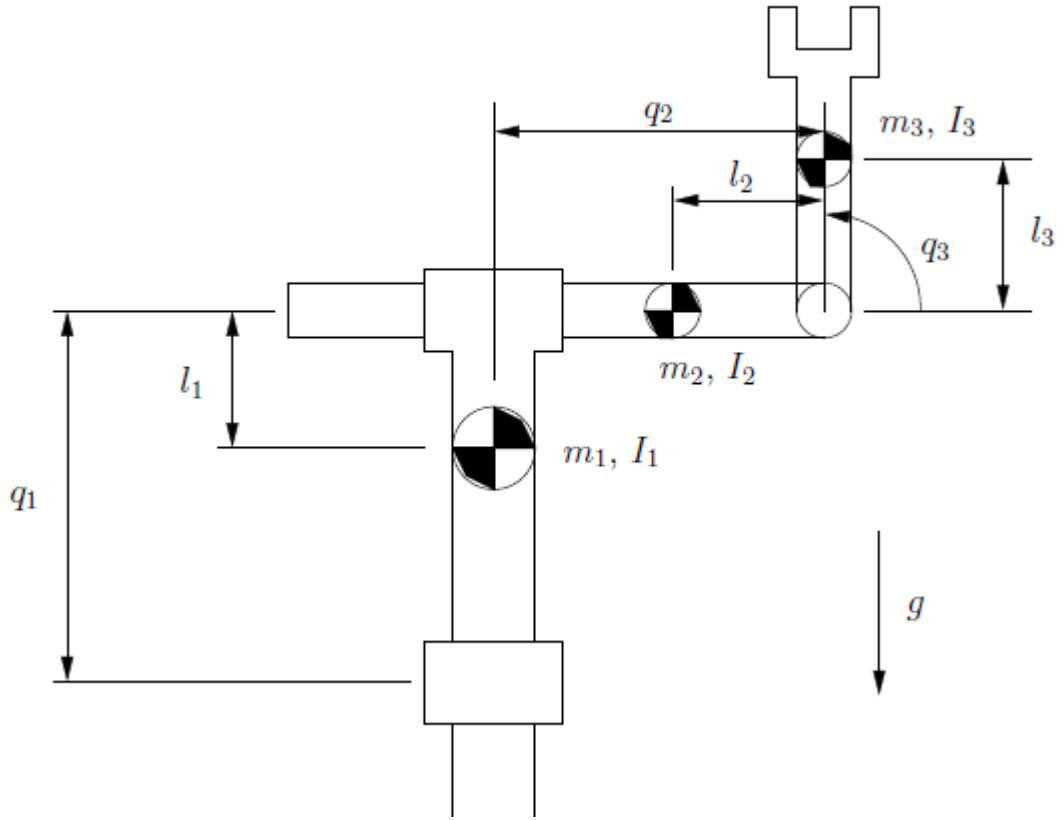
$$\begin{aligned} K &= -0.75 \\ q_{2d}(1) &= -2.4189 \text{ rad ou } 3.8643 \text{ rad (-138.6° ou 221.4°)} \\ q_{1d}(1) &= 1.2721 \text{ rad (72.9°)} \\ \boldsymbol{q}_d(1) &= \begin{bmatrix} 1.2721 \\ -2.4189 \end{bmatrix} \text{ rad} \end{aligned}$$

$$\begin{aligned} {}^0\boldsymbol{J}^{-1} &= \begin{bmatrix} -0.6220 & 1.3780 \\ 1.5119 & 1.5119 \end{bmatrix} \\ \dot{\boldsymbol{q}}_d(1) &= {}^0\boldsymbol{J}^{-1} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \text{ m/s} \\ &= \begin{bmatrix} 0.7559 \\ 3.0237 \end{bmatrix} \text{ rad/s} \end{aligned}$$

$$\begin{aligned} {}^0\dot{\boldsymbol{J}} &= \begin{bmatrix} -a_2 C_{12}(\dot{q}_1 + \dot{q}_2) - a_1 C_1 \dot{q}_1 & -a_2 C_{12}(\dot{q}_1 + \dot{q}_2) \\ -a_2 S_{12}(\dot{q}_1 + \dot{q}_2) - a_1 S_1 \dot{q}_1 & -a_2 S_{12}(\dot{q}_1 + \dot{q}_2) \end{bmatrix} \\ &= \begin{bmatrix} 1.0000 & -0.7775 \\ 1.0000 & 1.7225 \end{bmatrix} \\ \ddot{\boldsymbol{q}}_d(t) &= {}^0\boldsymbol{J}^{-1}(\boldsymbol{q}_d) \left\{ \begin{bmatrix} \ddot{x}_d(t) \\ \ddot{y}_d(t) \end{bmatrix} - {}^0\dot{\boldsymbol{J}}(\boldsymbol{q}_d)\dot{\boldsymbol{q}}_d(t) \right\} \\ \ddot{\boldsymbol{q}}_d(1) &= {}^0\boldsymbol{J}^{-1} \left\{ \begin{bmatrix} -1 \\ -1 \end{bmatrix} \text{ m/s}^2 - {}^0\dot{\boldsymbol{J}}\dot{\boldsymbol{q}}_d(1) \right\} \\ &= \begin{bmatrix} -10.907 \\ -7.343 \end{bmatrix} \text{ rad/s}^2 \end{aligned}$$

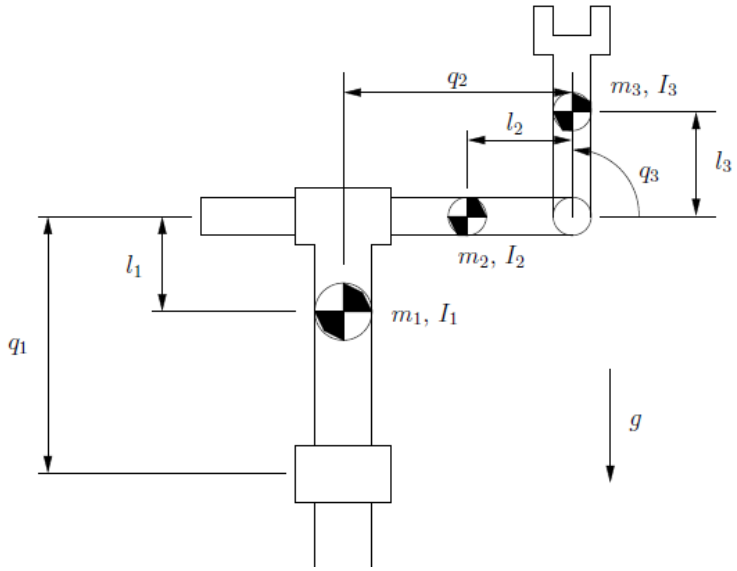
Révision
Exercices – Dynamique (3)

Développer le modèle dynamique du robot de la figure ci-dessous.



Révision
Exercices – Dynamique (3)

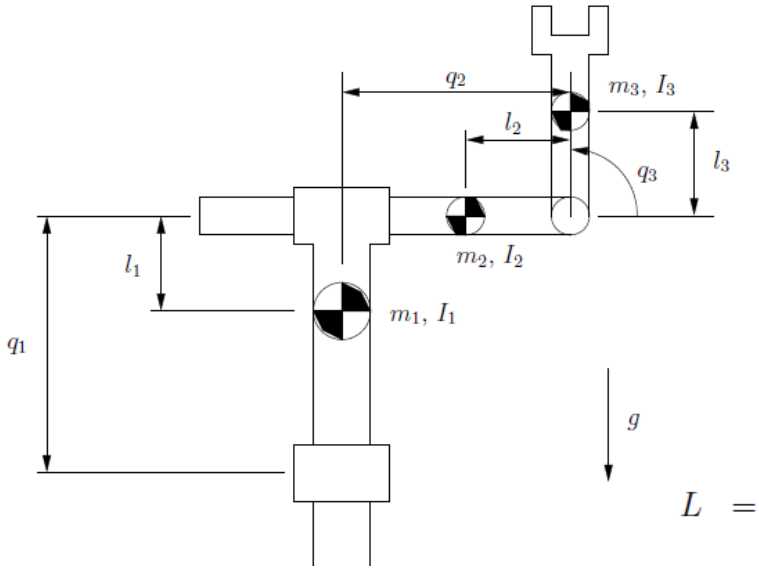
Développer le modèle dynamique du robot de la figure ci-dessous.



$$
\begin{aligned}
x_1 &= 0 \\
y_1 &= q_1 - l_1 \\
x_2 &= q_2 - l_2 \\
y_2 &= q_1 \\
x_3 &= q_2 + l_3\cos(q_3) \\
y_2 &= q_1 + l_3\sin(q_3) \\
v_1^2 &= \dot{q}_1^2 \\
v_2^2 &= \dot{q}_1^2 + \dot{q}_2^2 \\
v_3^2 &= [\dot{q}_2 - l_3\sin(q_3)\dot{q}_3]^2 + [\dot{q}_1 + l_3\cos(q_3)\dot{q}_3]^2 \\
&= \dot{q}_1^2 + \dot{q}_2^2 + 2l_3\dot{q}_3[\cos(q_3)\dot{q}_1 - \sin(q_3)\dot{q}_2] + l_3^2\dot{q}_3^2 \\
T_1 &= \frac{1}{2}m_1\dot{q}_1^2 \\
T_2 &= \frac{1}{2}m2[\dot{q}_1^2 + \dot{q}_2^2] \\
T_3 &= \frac{1}{2}m_3\left\{\dot{q}_1^2 + \dot{q}_2^2 + 2l_3\dot{q}_3[\cos(q_3)\dot{q}_1 - \sin(q_3)\dot{q}_2]\right\} + \frac{1}{2}[m_3l_3^2 + I_3]\dot{q}_3^2 \\
U_1 &= m_1g(q_1 - l_1) \\
U_2 &= m_2gq_1 \\
U_3 &= m_3g[q_1 + l_3\sin(q_3)]
\end{aligned}
$$

Révision
Exercices – Dynamique (3)

Développer le modèle dynamique du robot de la figure ci-dessous.



$$L = \frac{1}{2}[m_1+m_2+m_3]\dot{q}_1^2 + \frac{1}{2}[m_2+m_3]\dot{q}_2^2 + \frac{1}{2}[m_3 l_3^2 + I_3]\dot{q}_3^2 + m_3 l_3 \dot{q}_3[\cos(q_3)\dot{q}_1 - \sin(q_3)\dot{q}_2] - [m_1+m_2+m_3]g q_1 + m_1 g l_1 - m_3 g l_3 \sin(q_3)$$

$$\frac{\partial L}{\partial \dot{q}_1} = [m_1+m_2+m_3]\dot{q}_1 + m_3 l_3 \dot{q}_3 \cos(q_3)$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_1}\right) = [m_1+m_2+m_3]\ddot{q}_1 + m_3 l_3 \ddot{q}_3 \cos(q_3) - m_3 l_3 \dot{q}_3^2 \sin(q_3)$$

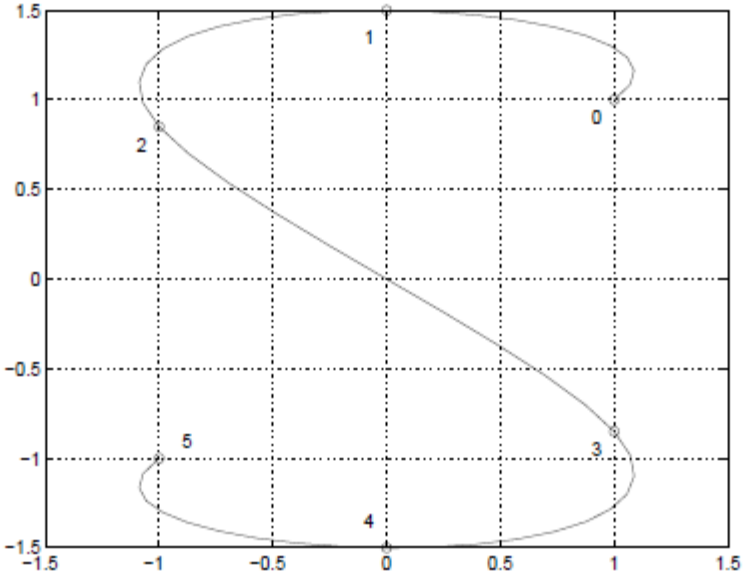
$$\frac{\partial L}{\partial q_1} = -[m_1+m_2+m_3]g$$

$$\tau_1 = [m_1+m_2+m_3]\ddot{q}_1 + m_3 l_3 \ddot{q}_3 \cos(q_3) - m_3 l_3 \dot{q}_3^2 \sin(q_3) + [m_1+m_2+m_3]g$$

Révision
Exercices – Planification de trajectoires (1)

Un robot doit suivre la trajectoire de la figure ci-dessous. La trajectoire doit être générée à l'aide d'une interpolation par spline. Calculer les valeurs de x et y ainsi que la vitesse tangentielle à $t = 2.5$ (les coordonnées sont en mètres).

Point	0	1	2	3	4	5
t	0	1	2	3	4	5
x	1	0	-1	1	0	-1
y	1	1.5	0.85	-0.85	-1.5	-1
c_x	0	-0.8182	3.2727	-3.2727	0.8182	0
c_y	0	-0.6545	-0.8318	0.8318	0.6545	0



Révision
Exercices – Planification de trajectoires (2)

Le profil de vitesse trapézoïdal a un désavantage : le profil d'accélération est discontinu. Pour avoir un profil d'accélération continu, on peut utiliser le profil de la figure ci-dessous.

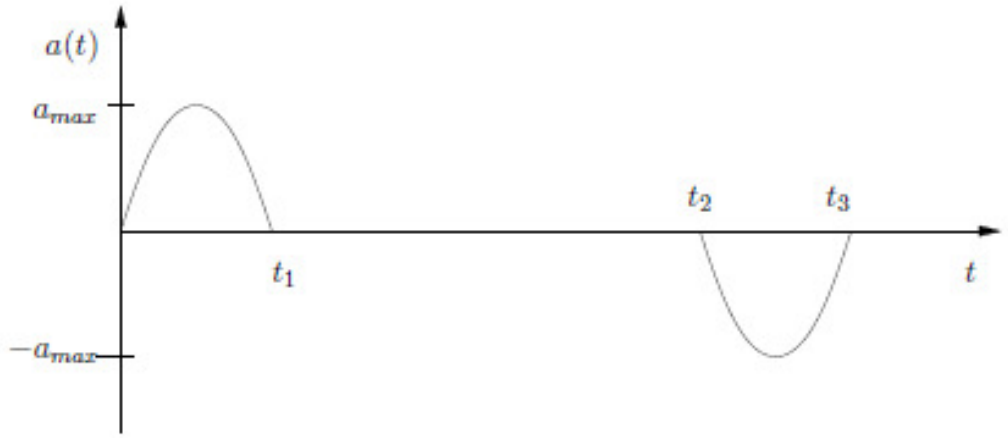


FIG. 2 – Profil d'accélération

Ici, les portions du profil d'accélération non nulles sont des paraboles. On a aussi $t_3 - t_2 = t_1$.

a. Déterminer le profil de vitesse $V(t)$ correspondant pour $V(0) = 0$.

b. Donner la distance parcourue de $t = 0$ à $t = t_3$.

Révision
Vision (1)

Considérons l'image de la figure ci-dessous après segmentation (le pixel en haut à gauche est numéroté (1,1) et celui en bas à droite (10,10)).

	1				j	$\rightarrow$				10
1	0	1	0	0	0	0	0	0	0	0
	1	1	0	0	0	2	2	0	0	0
	1	1	0	0	2	2	2	2	0	0
	1	1	0	0	2	2	2	2	0	0
i	0	1	0	0	0	2	2	0	0	0
$\downarrow$	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	0	3	3	3	0	0	4	4	0
	0	3	3	3	3	3	0	4	4	0
10	0	0	0	0	0	0	0	0	0	0

FIG. 4 – Résultat de la segmentation en régions d'une image binaire après seuillage

Expliquer, calculs à l'appui, comment caractériser et comparer les différentes régions de cette image (les « objets » identiques doivent être détectés) et donner leur position. Proposer une méthode qui fonctionnerait aussi après avoir déplacé les « objets » (rotations et translations).

Révision
Vision (2)

Une caméra est placée à angle par rapport à une surface (le plan image n'est pas parallèle au plan de travail). Tous les objets sur cette surface ont la même coordonnée $Z = z_0$. Certains de ces objets sont des balises de coordonnées connues : (x_{bk}, y_{bk}, z_0) (où $k = 1$ à n avec n le nombre de balises). Expliquer comment utiliser ces balises pour l'étalonnage de la caméra avec une projection perspective. Une fois la caméra étalonnée, expliquer comment on peut retrouver la coordonnée d'un objet (x_o, y_o, z_0) à partir de sa coordonnée image (i_o, j_o).

Révision
Vision (2)

a) Après avoir déterminé les centroïdes des balises en coordonnées pixel (après seuillage, étiquetage ou autre technique), il faut déterminer les éléments de la matrice :

$$\boldsymbol{P}\,{}^{C}\boldsymbol{T}_{U} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ 0 & 0 & 0 & 0 \\ a_{41} & a_{42} & a_{43} & 1 \end{bmatrix}$$

par moindres carrés (avec $n \geq 6$ balises) :

$$\begin{bmatrix} x_{b1} & y_{b1} & z_0 & 1 & 0 & 0 & 0 & 0 & -i_{b1}x_{b1} & -i_{b1}y_{b1} & -i_{b1}z_0 \\ 0 & 0 & 0 & 0 & x_{b1} & y_{b1} & z_0 & 1 & -j_{b1}x_{b1} & -j_{b1}y_{b1} & -j_{b1}z_0 \\ x_{b2} & y_{b2} & z_0 & 1 & 0 & 0 & 0 & 0 & -i_{b2}x_{b2} & -i_{b2}y_{b2} & -i_{b2}z_0 \\ 0 & 0 & 0 & 0 & x_{b2} & y_{b2} & z_0 & 1 & -j_{b2}x_{b2} & -j_{b2}y_{b2} & -j_{b2}z_0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{bn} & y_{bn} & z_0 & 1 & 0 & 0 & 0 & 0 & -i_{bn}x_{bn} & -i_{bn}y_{bn} & -i_{bn}z_0 \\ 0 & 0 & 0 & 0 & x_{bn} & y_{bn} & z_0 & 1 & -j_{bn}x_{bn} & -j_{bn}y_{bn} & -j_{bn}z_0 \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{14} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{24} \\ a_{41} \\ a_{42} \\ a_{43} \end{bmatrix} = \begin{bmatrix} i_{b1} \\ j_{b1} \\ i_{b2} \\ j_{b2} \\ \vdots \\ \vdots \\ i_{bn} \\ j_{bn} \end{bmatrix}$$

Révision
Vision (2)

b) comme on connaît le $Z = z_0$ des points des objets, on a

$$\begin{bmatrix} ki_o \\ kj_o \\ 0 \\ k \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ 0 & 0 & 0 & 0 \\ a_{41} & a_{42} & a_{43} & 1 \end{bmatrix} \begin{bmatrix} x_o \\ y_o \\ z_0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} ki_o \\ kj_o \\ k \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{41} & a_{42} & a_{43} & 1 \end{bmatrix} \begin{bmatrix} x_o \\ y_o \\ z_0 \\ 1 \end{bmatrix}$$

Les inconnues sont x_o, y_o et k. On a donc

$$k \begin{bmatrix} i_o \\ j_o \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{41} & a_{42} \end{bmatrix} \begin{bmatrix} x_o \\ y_o \end{bmatrix} + \begin{bmatrix} a_{13} & a_{14} \\ a_{23} & a_{24} \\ a_{43} & 1 \end{bmatrix} \begin{bmatrix} z_0 \\ 1 \end{bmatrix}$$

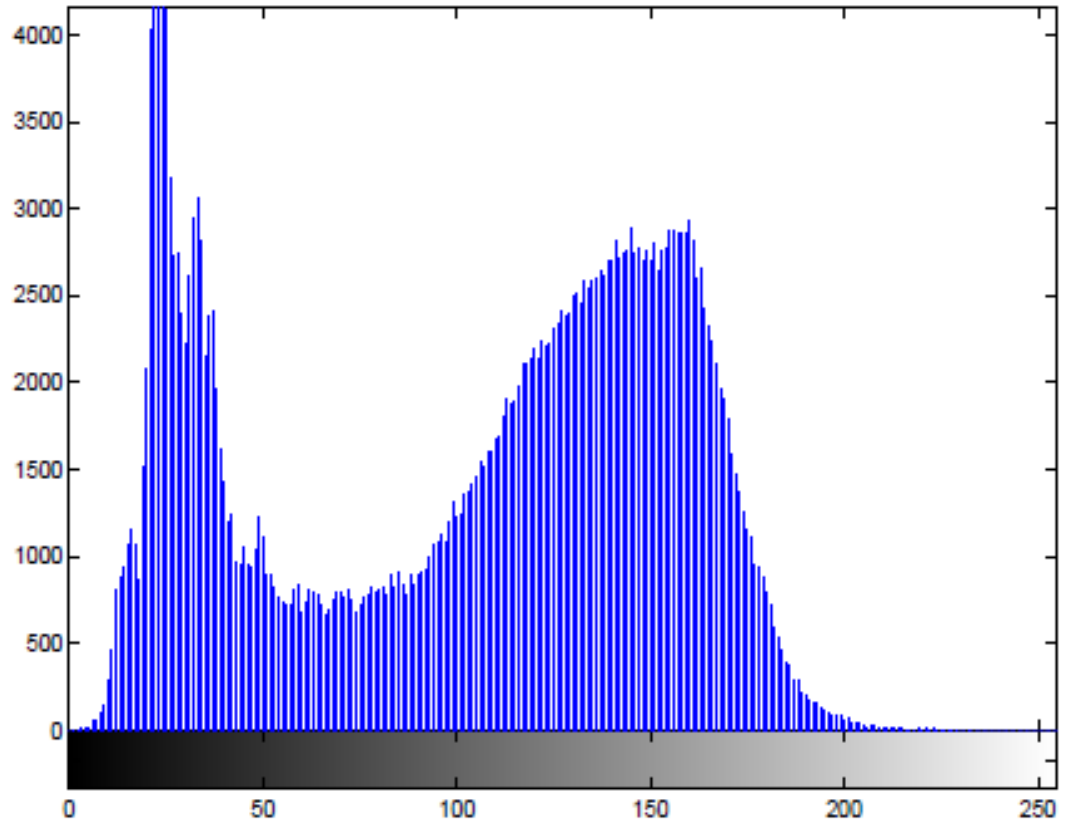
En isolant x_o, y_o et k, on obtient

$$\begin{bmatrix} -a_{11} & -a_{12} & i_o \\ -a_{21} & -a_{22} & j_o \\ -a_{41} & -a_{42} & 1 \end{bmatrix} \begin{bmatrix} x_o \\ y_o \\ k \end{bmatrix} = \begin{bmatrix} a_{13} & a_{14} \\ a_{23} & a_{24} \\ a_{43} & 1 \end{bmatrix} \begin{bmatrix} z_0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x_o \\ y_o \\ k \end{bmatrix} = \begin{bmatrix} -a_{11} & -a_{12} & i_o \\ -a_{21} & -a_{22} & j_o \\ -a_{41} & -a_{42} & 1 \end{bmatrix}^{-1} \begin{bmatrix} a_{13} & a_{14} \\ a_{23} & a_{24} \\ a_{43} & 1 \end{bmatrix} \begin{bmatrix} z_0 \\ 1 \end{bmatrix}$$

Révision
Vision (3)

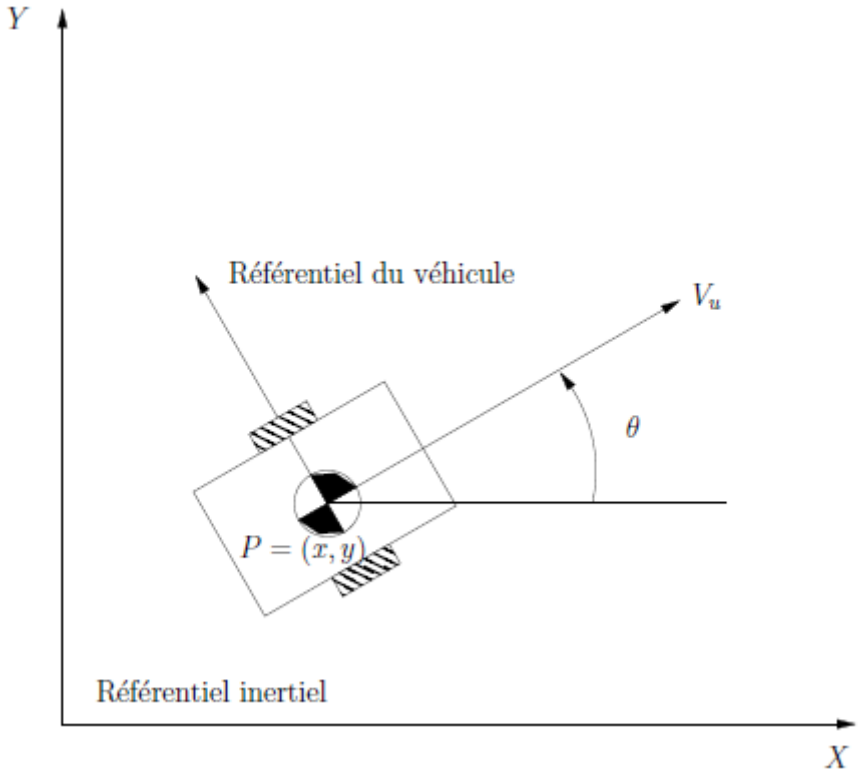
On veut détecter des objets dans une image ayant l'histogramme suivant :



Suggérer une technique pour y arriver et justifier votre choix.

Révision
Robots mobiles (1)

- Soit le véhicule ci-dessous. Nous désirons faire en sorte que ce véhicule soit apte à suivre des trajectoires.
  - Expliquez la méthode que vous utiliseriez et mentionnez les capteurs qui vous seraient nécessaires.



Révision
Robots mobiles (2)

- **Vrai ou faux?**
  - Le contrôleur d'Astolfi classique:
    $$v = k_p \rho$$
    $$\omega = k_\alpha \alpha + k_\phi \phi$$
    $$\begin{aligned} k_\rho &> 0 \\ k_\phi &< 0 \\ k_\alpha + k_\phi - k_\rho &> 0. \end{aligned}$$
    garanti qu'il n'y aura pas d'oscillations au niveau de la dynamique angulaire d'un robot mobile.
  - Le filtre de Kalman permet de "filtrer" le signal d'entré d'un système dynamique et assure une erreur de suivi nulle en régime permanent.
  - Le filtre de Kalman étendu permet d'estimer de façon optimal l'état d'un système dynamique non-linéaire à partir de mesures provenant d'un ou plusieurs capteurs.

- Nommez deux désavantages liés à l'utilisation de la caméra stéréoscopique pour la navigation des robots mobiles en terrain inconnu.

- Deux ingénieurs s'obstinent au sujet de la localisation d'un robot mobile: Le premier prétend qu'il serait pratique de se localiser en utilisant l'odométrie du robot (c'est-à-dire en se servant des encodeurs des roues), tandis que le deuxième prétend qu'il serait mieux d'utiliser une centrale inertielle et résoudre les équations de navigation inertielle. Selon vous, lequel a raison? Avez-vous une meilleure solution?

Références

[1] Leonard, J.E., Durrant-Whyte, H.F., *Directed Sonar Sensing for Mobile Robot Navigation*. Norwood, MA, Kluwer Academic Publishers, 1992.

[2] Borenstein, J., Everet,t H.R., Feng, L., *Navigating Mobile Robots, Systems and Techniques*. Natick, MA, A.K. Peters, Ltd., 1996.

[3] Cox, I.J., Wilfong, G.T. (editors)*, Autonomous Robot Vehicles*. New York, Spinger- Verlag, 1990.

[4] Siegwart, R. Nourbakhsh, I., *Introduction to Autonomous Mobile Robots.* MIT Press, 2004.