



# ROBOTIQUE

## -ELE4203-

*Cours #12: Introduction à la robotique mobile (partie 2):*  
**Localisation, contrôle et planification de trajectoires**

**Enseignant: Jean-Philippe Roberge**



# Cours #12

## ◆ Retour sur les notions du cours #11 :

- ◆ Modèle de la caméra trou d'épingle (modèle du sténopé ou “*pinhole camera*”)
- ◆ Types de robot mobile
- ◆ Types de déplacements et moyens de locomotion
- ◆ Configuration et nombre de roues.
- ◆ Définition de *Manoeuvrabilité*, *mobilité* et *directionnabilité (steerability)*
- ◆ Cinématique des robots mobiles
- ◆ Rappel des capteurs classiques utilisés en robotique mobile.

# Cours #12

- ◆ **Présentation du *robot de la semaine*:**

- ◆ ACE: *Autonomous City Explorer*

- ◆ Institute of Automatic Control Engineering – Technical University of Munich

- ◆ **Cours #12: *Localisation, contrôle et planification de trajectoires***

- ◆ Un autre exemple de cinématique directe d'un robot mobile

- ◆ Continuation de la discussion sur la 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

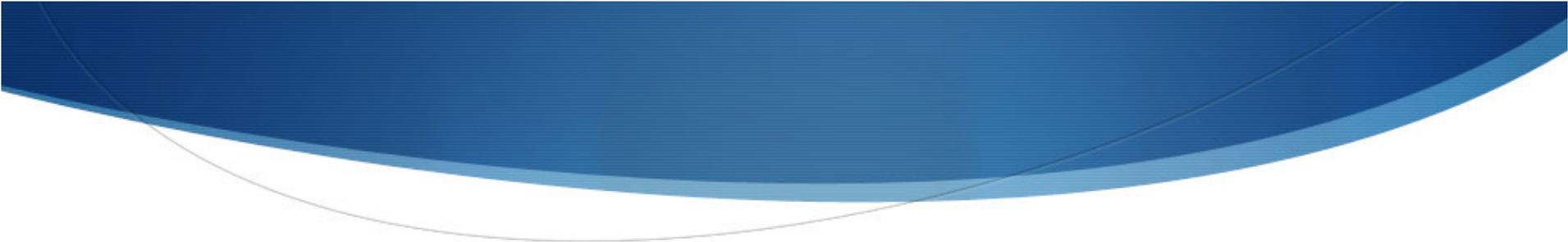
- ◆ 2) Estimation d'une quantité fixe à l'aide d'un processus récursif

- ◆ 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 processus récursif (filtre étendu de Kalman)

# Cours #12

- ◆ 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
    - ◆ Contrôleur d'Astolfi modifié
  - ◆ Concrètement, comment faire?
- ◆ Prochain cours : révision

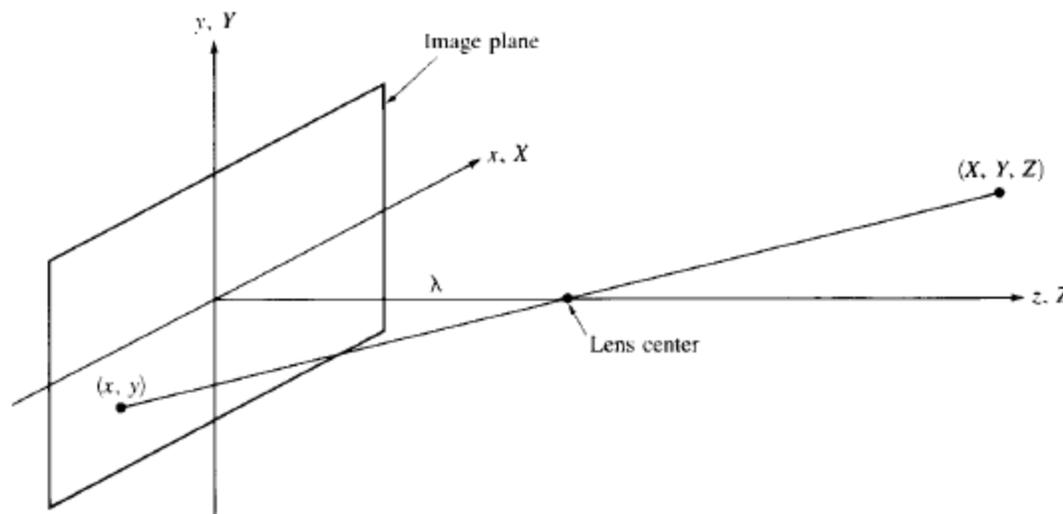


# Retour sur le cours #11

# Rappel Cours #11

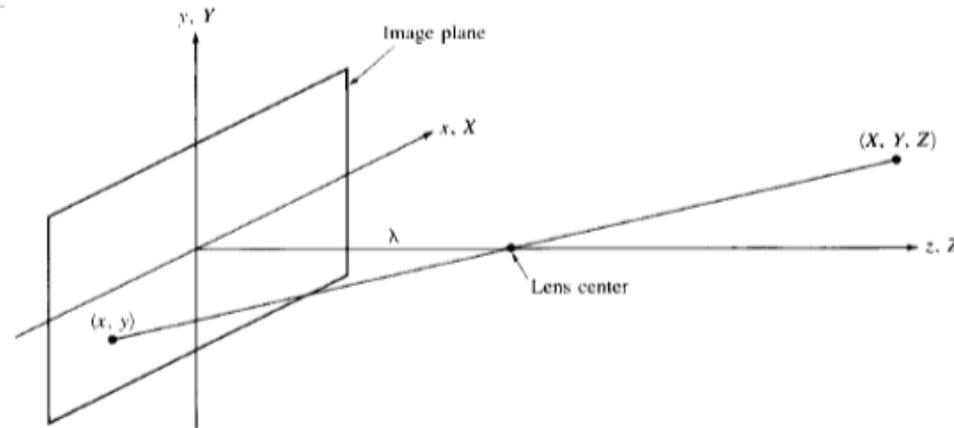
## Modèle de la caméra – Trou d'épingle (1)

- Lorsque les objets ne sont pas dans le même plan, il est possible de se référer à un deuxième modèle simple de caméra, i.e. le modèle de la caméra trou d'épingle (modèle du sténopé ou *pinhole*).
- Le principe est illustré ci-dessous (image tirée de [1]):



# Rappel Cours #11

## Modèle de la caméra – Trou d'épingle (2)

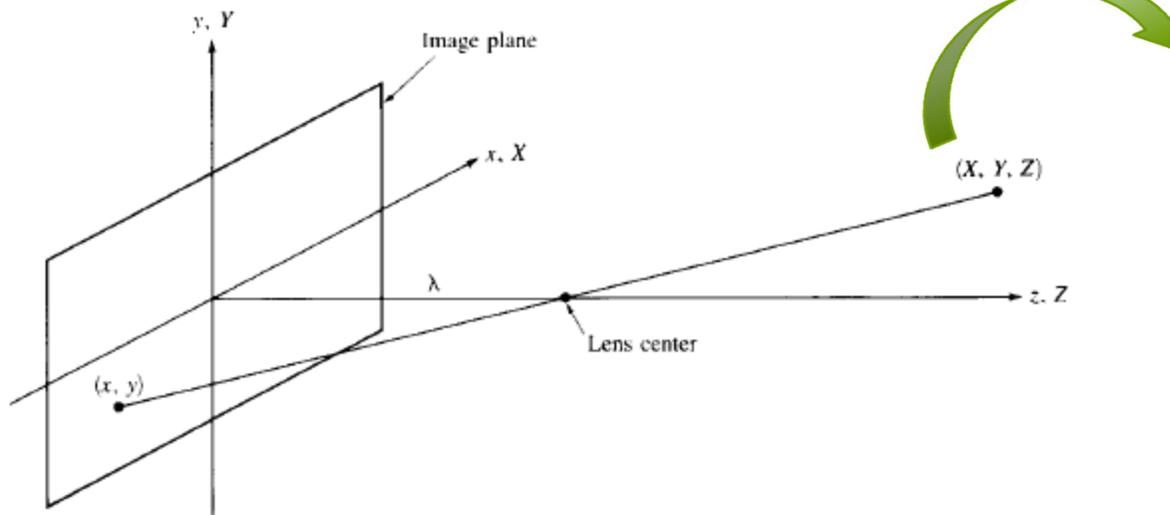


- ◆ Tous les points *visible* (considérant le champ de vision de la caméra) sont projetés dans le plan image.
- ◆ Cette image prend pour acquis que le repère universel est confondu avec le repère de la caméra → Nous allons voir plus tard comment faire lorsque ce n'est pas le cas.
- ◆  $\lambda$  est la longueur focale de la lentille.
- ◆ Notez que dans le plan image, l'image réelle est inversée.
- ◆ Le but sera premièrement de trouver la relation entre les coordonnées  $(x, y)$  d'un point du repère universel  $(X, Y, Z)$

# Rappel Cours #11

## Modèle de la caméra – Trou d'épingle (3)

- 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.**

# Rappel Cours #11

## Modèle de la caméra – Trou d'épingle (4)

- Étant donné cette particularité, la matrice de transformation s'exprime sous une forme à laquelle nous sommes moins habitués, 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*:

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

# Rappel Cours #11

## Modèle de la caméra – Trou d'épingle (5)

- Avant d'aller plus loin, rappelons la définition d'une coordonnée homogène (tel que présenté au cours #1):

En coordonnées homogènes, le point de coordonnées cartésiennes  $(x, y, z)$  est représenté par  $(x', y', z', w)$  où  $w \neq 0$ .

La relation entre les coordonnées cartésiennes et homogènes est donnée par

$$x = x'/w$$

$$y = y'/w$$

$$z = z'/w$$

- Donc,

Un point de l'espace cartésien:  $\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$

peut s'exprimer en coordonnées homogènes:

$$\begin{bmatrix} kX \\ kY \\ kZ \\ k \end{bmatrix}$$

avec  $k \neq 0$

Pour retrouver les coordonnées cartésiennes il faut diviser les coordonnées homogènes par le dernier élément du vecteur.

# Rappel Cours #11

## Modèle de la caméra – Trou d'épingle (6)

- Revenons à notre *transformation perspective*:

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

- Lorsque l'on applique une telle transformation à un point de l'espace cartésien, ce dernier sera transformé de cette façon:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-1}{\lambda} & 1 \end{bmatrix} \begin{bmatrix} kX \\ kY \\ kZ \\ k \end{bmatrix} = \begin{bmatrix} kX \\ kY \\ 0 \\ \frac{-kZ}{\lambda} + k \end{bmatrix}$$

- Pour retrouver les coordonnées cartésiennes, on divise par le dernier élément:

$$\underbrace{\begin{bmatrix} kX \\ kY \\ 0 \\ \frac{-kZ}{\lambda} + k \end{bmatrix}}_{\text{Coordonnées homogènes}} \Rightarrow \begin{bmatrix} \frac{kX}{\frac{-kZ}{\lambda} + k} \\ \frac{kY}{\frac{-kZ}{\lambda} + k} \\ \frac{-kZ}{\frac{-kZ}{\lambda} + k} \\ 0 \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{\lambda X}{\lambda - Z} \\ \frac{\lambda Y}{\lambda - Z} \\ 0 \end{bmatrix}}_{\text{Coordonnées cartésiennes}}$$

# Rappel Cours #11

## Modèle de la caméra – Trou d'épingle (7)

- Autre remarque concernant la matrice de *transformation perspective*:

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

- Remarquez que cette dernière matrice est non-inversible...

- Avez-vous une interprétation de ce fait mathématique?**

- Il est en effet impossible de retrouver les coordonnées 3D (X,Y,Z) d'un point de l'espace cartésien seulement à partir de ses coordonnées de l'espace image!
- Maintenant, tel que promis, visitons le cas où le repère universel n'est pas confondu avec le repère de la caméra.

- La transformation entre les points cartésiens et les points de l'image sera donnée par:**

$$\left( P \quad {}^cT_U \right) \quad \Rightarrow \text{où } {}^uT_c = \begin{bmatrix} {}^uR_c & {}^uP_c \\ 0 & 1 \end{bmatrix} \text{ est la matrice de transformation entre le repère de la caméra et le repère universel}$$

Remarque: Évidemment,  ${}^cT_U = ({}^uT_c)^{-1}$

# Rappel Cours #11

## Modèle de la caméra – Trou d'épingle (8)

- Autre remarque concernant la *transformation perspective*:

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-1}{\lambda} & 1 \end{bmatrix} \quad \text{et} \quad (P {}^c T_U)$$

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

$$P {}^c T_U = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ 0 & 0 & 0 & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

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

Ainsi,

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

# Rappel Cours #11

## Modèle de la caméra – Trou d'épingle (9)

- Continuant le développement *transformation perspective*:

$$\boxed{(P^c T_U)} \quad \begin{bmatrix} kx \\ ky \\ 0 \\ k \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ 0 & 0 & 0 & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

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

- 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:

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

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

# Rappel Cours #11

## Modèle de la caméra – Trou d'épingle (10)

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

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

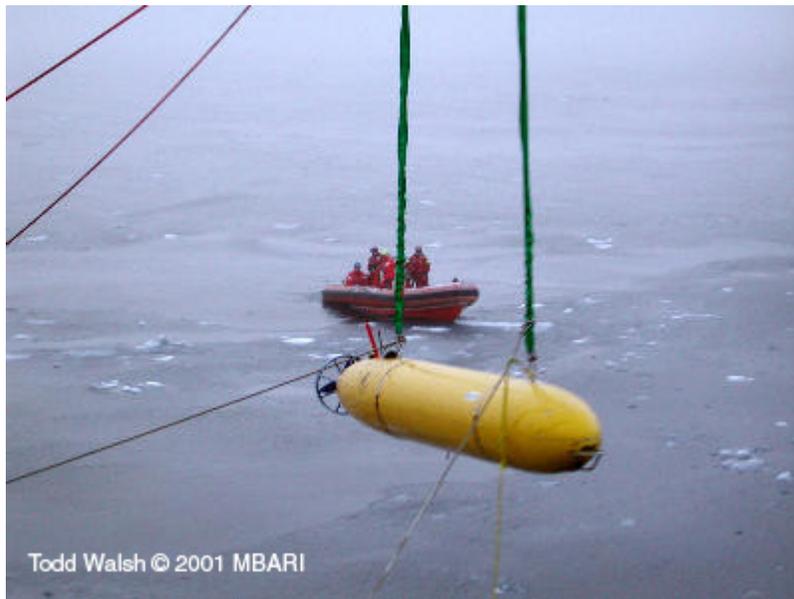
Si  $n \geq 6$  points sont disponibles, on peut solutionner au sens des moindres carrés l'équation :

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -x_1 X_1 & -x_1 Y_1 & -x_1 Z_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -y_1 X_1 & -y_1 Y_1 & -y_1 Z_1 \\ X_2 & Y_2 & Z_2 & 1 & 0 & 0 & 0 & 0 & -x_2 X_2 & -x_2 Y_2 & -x_2 Z_2 \\ 0 & 0 & 0 & 0 & X_2 & Y_2 & Z_2 & 1 & -y_2 X_2 & -y_2 Y_2 & -y_2 Z_2 \\ \vdots & \vdots \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -x_n X_n & -x_n Y_n & -x_n Z_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -y_n X_n & -y_n Y_n & -y_n Z_n \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{14} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{24} \\ a_{41} \\ a_{42} \\ a_{43} \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ \vdots \\ \vdots \\ x_n \\ y_n \end{bmatrix}$$

# Rappel Cours #11

## Robots mobiles – Raisons d’êtres (1)

- ◆ Pourquoi certaines applications requièrent-elles des robots dotés de mobilité?
  - ◆ Pour les environnements difficiles (même sur Terre)
    - ◆ Exemple: Robot qui inspecte Chernobyl, sous-marin autonomes



# Rappel Cours #11

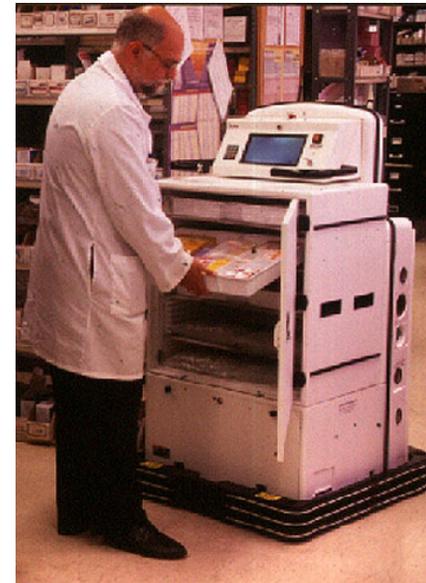
## Robots mobiles – Raisons d’êtres (2)

- ◆ Pourquoi certaines applications requièrent-elles des robots dotés de mobilité (suite)?
  - ◆ Pour les environnements difficiles (même sur Terre)
    - ◆ Pour remplacer les humains pour certaines tâches simples

Toyota’s tour-guide robot



HELPMATE: transport de nourriture en milieu hospitalier



# Rappel Cours #11

## Robots mobiles – Raisons d’êtres (3)

- ◆ Pourquoi certaines applications requièrent-elles des robots dotés de mobilité (suite)?
  - ◆ Pour les environnements difficiles (même sur Terre)
    - ◆ À des fins éducatives: Khepera



**K-Team**  **BEYOND MINIATURE TECHNOLOGY**  
MOBILE ROBOTICS

Home Products About Us Services Introduction Support Downloads Contact Us

► Newsletter  
email

► Products

-  K-Team
-  Khepera III
-  Khepera II
-  Khepera I
-  Khepera I
-  Khepera I
-  Khepera I

**What's New at K-Team?**

**New CPU power board for the K-Team!**  
This CPU board will expand the computing power and the programming capabilities of your K-Team!

**KHEPERA I ACCESS**

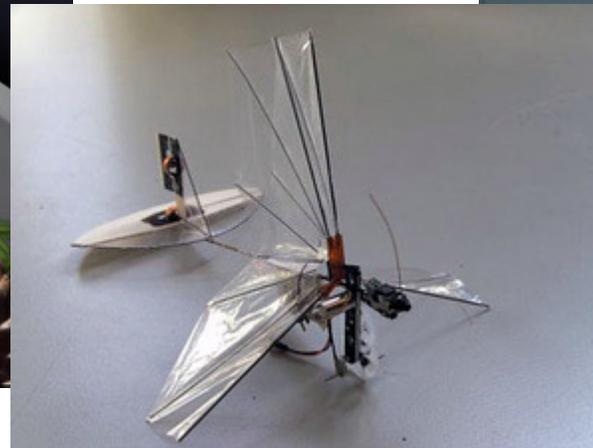
**More 3 gyros!**  
The new gyro for the Khepera I is now available!

**2 types of sensors for the Khepera I and Khepera II!**  
An ultra-compact sensor with 4x4 resolution!

# Rappel Cours #11

## Robots mobiles – Raisons d’êtres (4)

- ◆ Pourquoi certaines applications requièrent-elles des robots dotés de mobilité (suite)?
  - ◆ Pour les environnements difficiles (même sur Terre)
    - ◆ Pour la recherche et développement



# Rappel Cours #11

## Robots mobiles – Raisons d’êtres (5)

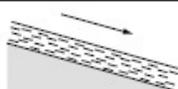
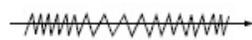
- ◆ Pourquoi certaines applications requièrent-elles des robots dotés de mobilité (suite)?
  - ◆ Pour les environnements difficiles (même sur Terre)
    - ◆ Applications militaires



# Rappel Cours #11

## Types de locomotions (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) 

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

**La roue!**

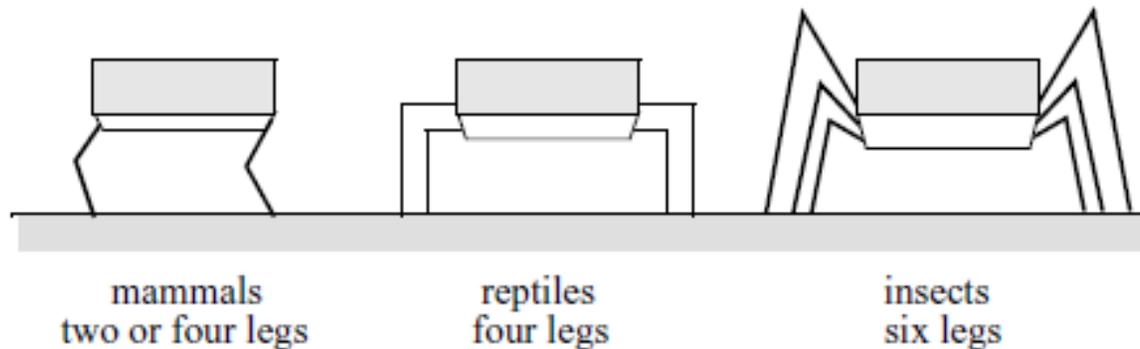
**(On ne la retrouve pas tel quel dans la nature!)**

Figure 2.1  
Locomotion mechanisms used in biological systems.

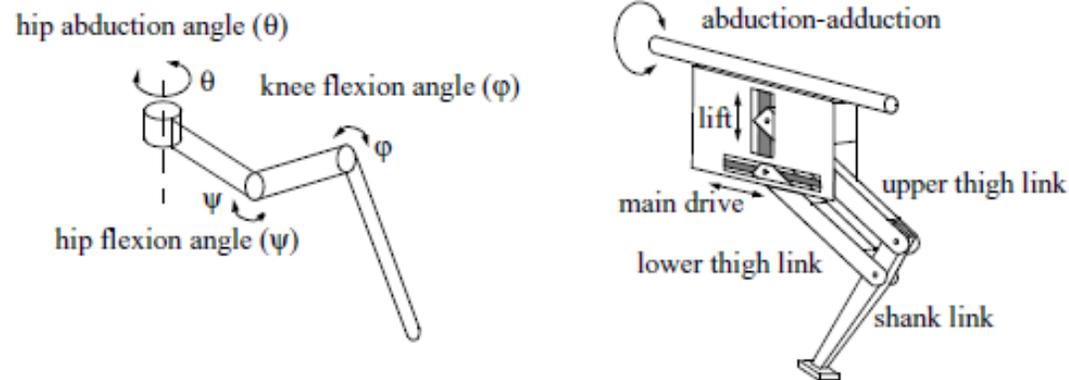
# Rappel Cours #11

## Types de locomotions – Configuration avec jambes (1)

- Dans la nature, on retrouve plusieurs animaux possédant des configurations de jambes desquelles on peut s'inspirer pour les robots mobiles:



- Différentes jambes à 3 D.D.L.:

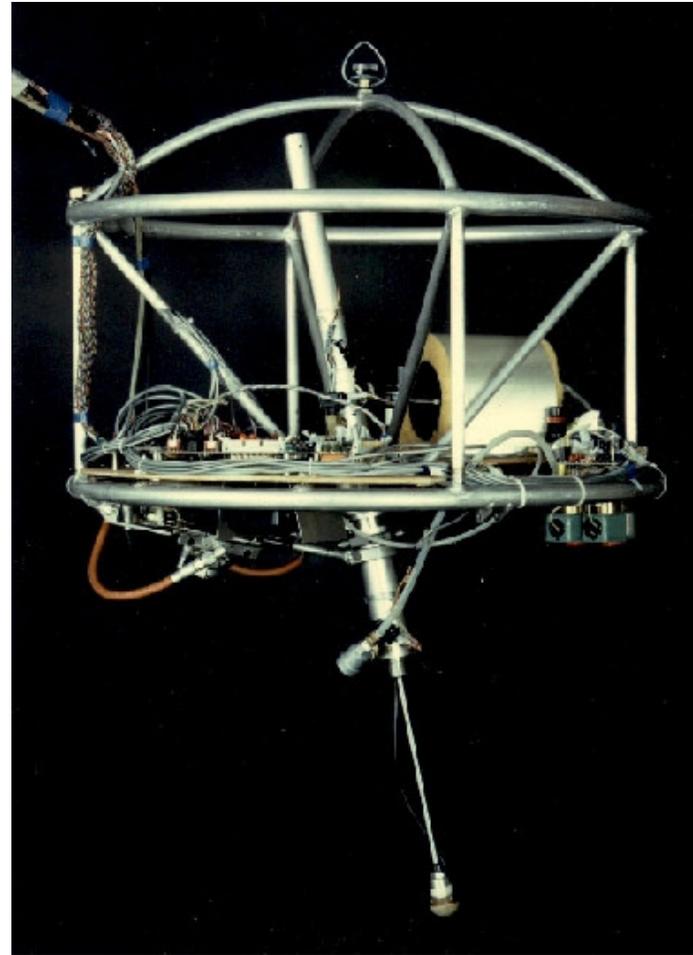
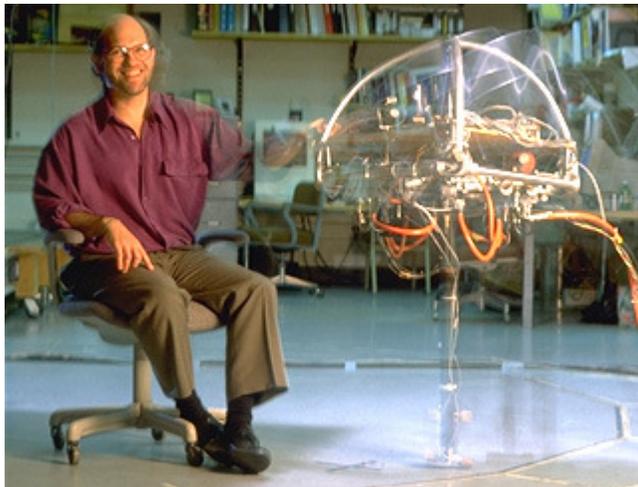


- Nombres de configurations possibles des jambes:  $N = (2k - 1)!$  où  $k$  est le nombre de jambes

# Rappel Cours #11

## Types de locomotions – Configuration avec jambes (2)

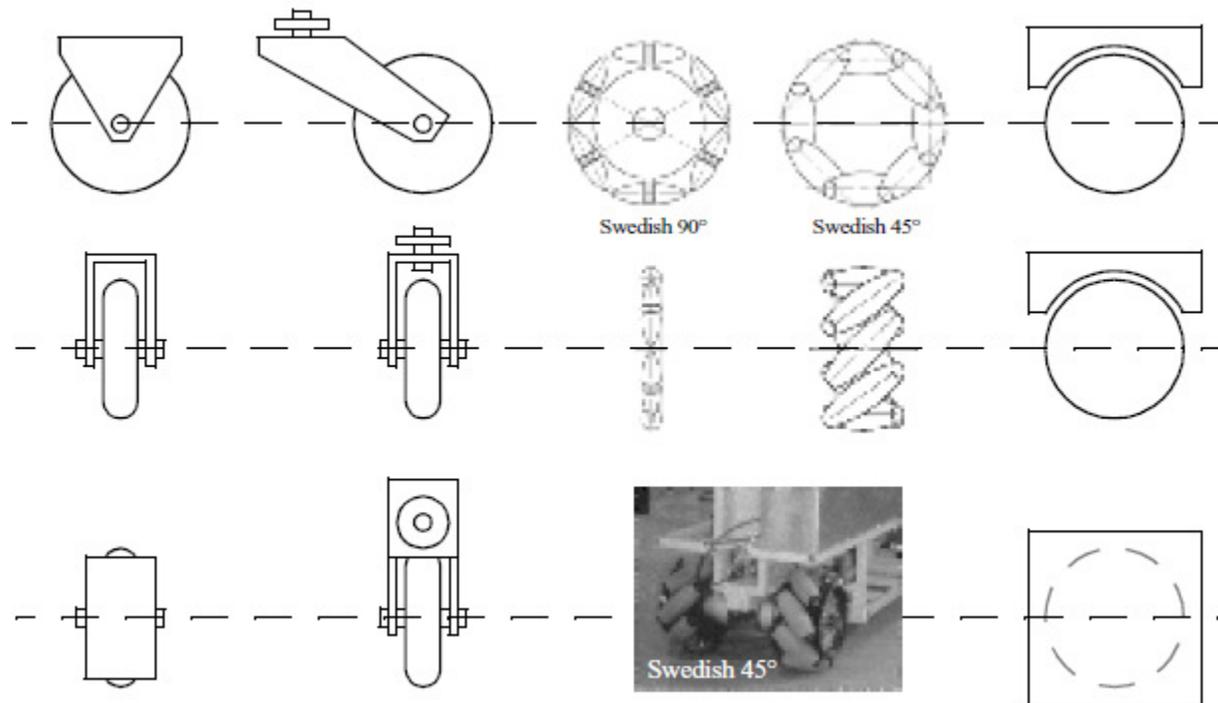
- ◆ Certains robots particuliers existent:



# Rappel Cours #11

## Types de locomotions – Configuration avec roues (1)

- ◆ 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:



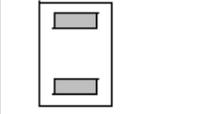
Aussi, tant qu'à parler de type de roue, pour vous ouvrir l'esprit:

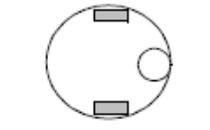
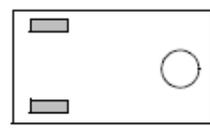
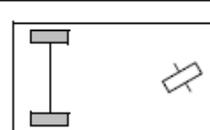
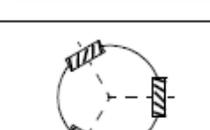
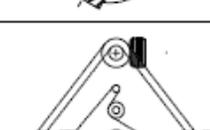
<http://www.radio-canada.ca/emissions/decouverte/2011-2012/Reportage.asp?idDoc=211681>

# Rappel Cours #11

## Types de locomotions – Configuration avec roues (2)

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

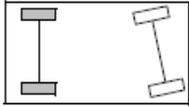
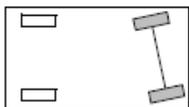
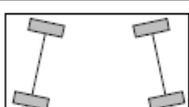
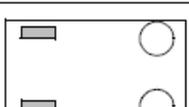
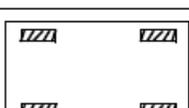
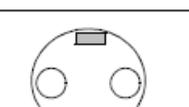
# of wheels	Arrangement	Description	Typical examples
2		One steering wheel in the front, one traction wheel in the rear	Bicycle, motorcycle
		Two-wheel differential drive with the center of mass (COM) below the axle	Cye personal robot

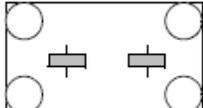
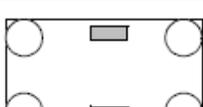
3		Two-wheel centered differential drive with a third point of contact	Nomad Scout, smartRob EPFL
		Two independently driven wheels in the rear/front, 1 unpowered omnidirectional wheel in the front/rear	Many indoor robots, including the EPFL robots Pygmalion and Alice
		Two connected traction wheels (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

# Rappel Cours #11

## Types de locomotions – Configuration avec roues (3)

### ◆ 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

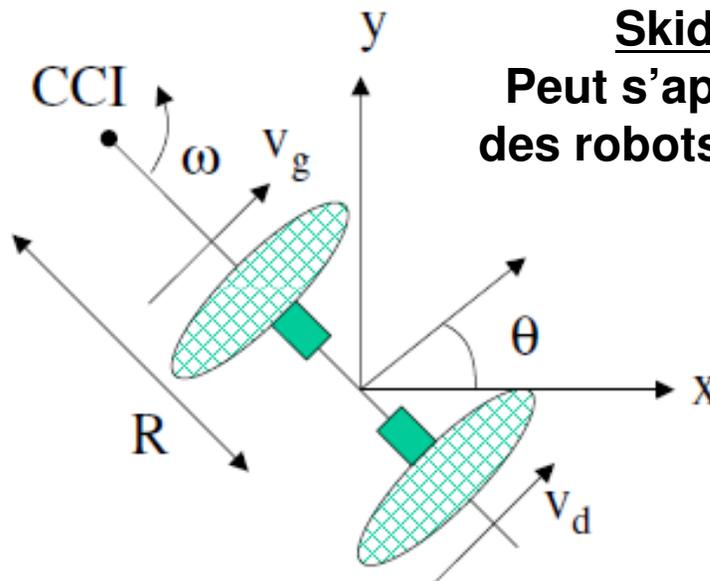
# of wheels	Arrangement	Description	Typical examples
6		Two motorized and steered wheels aligned in center, 1 omnidirectional wheel at each corner	First
		Two traction wheels (differential) in center, 1 omnidirectional wheel at each corner	Terregator (Carnegie Mellon University)
Icons for the each wheel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

# Rappel Cours #11

## Types de locomotions – Configuration avec roues (4)

- ◆ Géométries de roues fréquemment rencontrées (notes de cours):

### Robot à commande différentielle



**Skid steer robots:**  
Peut s'appliquer aussi pour  
des robots à 4 roues motrices  
2X2



<http://www.amigobot.com>

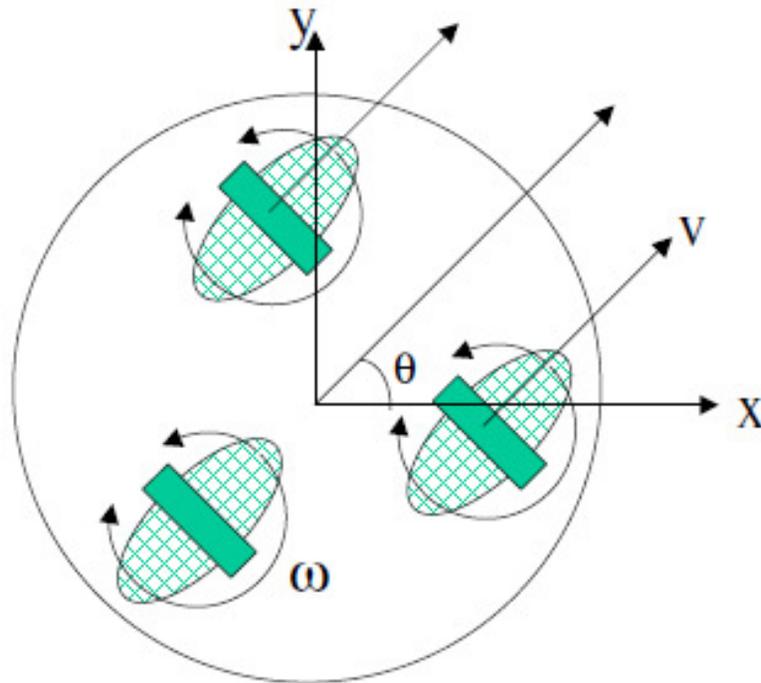
CCI: Centre de courbure instantané

# Rappel Cours #11

## Types de locomotions – Configuration avec roues (5)

- ◆ Géométries de roues fréquemment rencontrées (notes de cours):

### Robot à commande synchronisée



B21r  
<http://www.robosoft.fr>

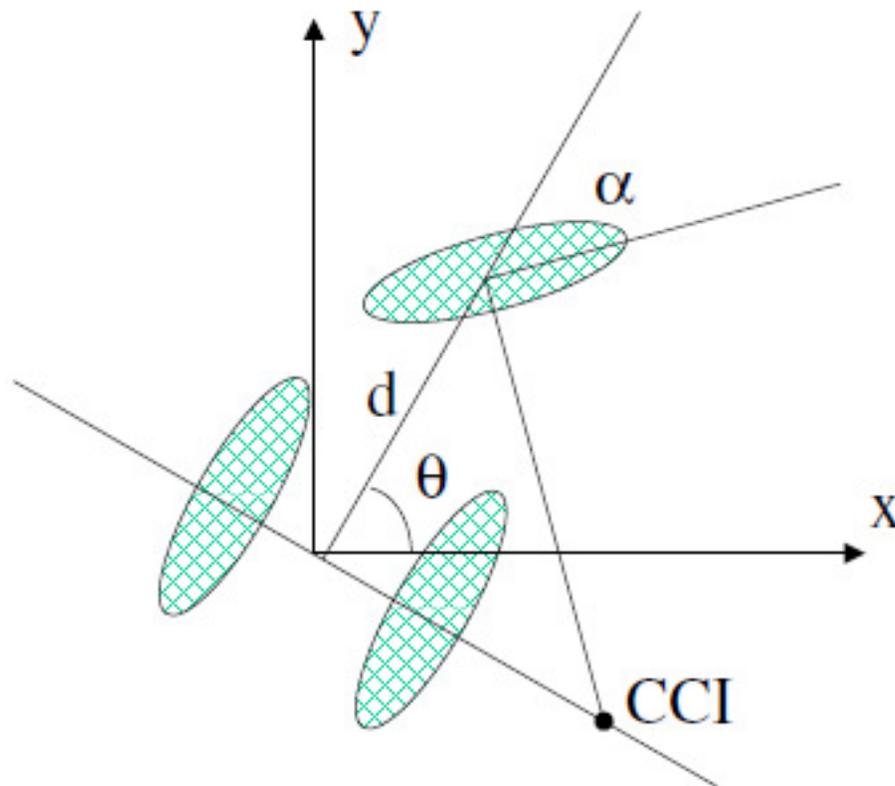
CCI à l'infini

# Rappel Cours #11

## Types de locomotions – Configuration avec roues (6)

- ◆ Géométries de roues fréquemment rencontrées (notes de cours):

### Tricycle - Bicyclette

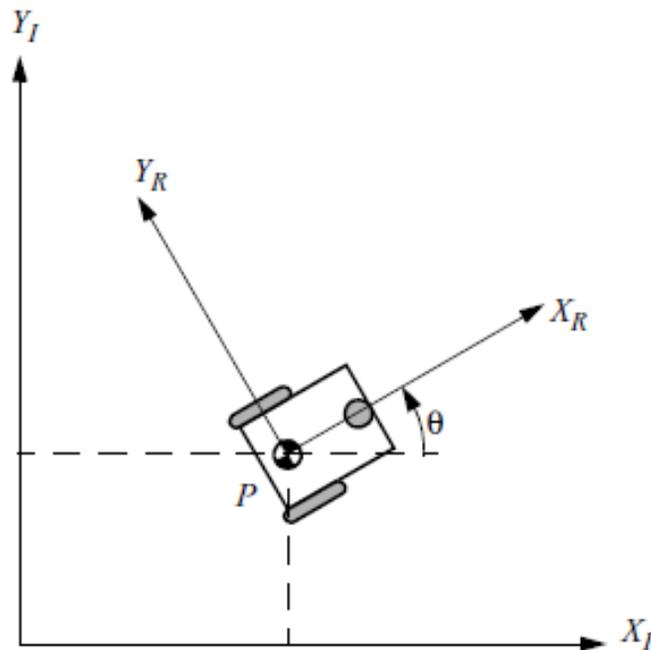


<http://www.robotcountryusa.com>

# Rappel Cours #11

## Robots mobiles : Premières notions de cinématique (1)

- ◆ Souvent, on utilise le modèle simple (planaire) suivant pour modéliser la pose d'un robot mobile exprimée dans le repère de référence (repère universel):



- ◆ La pose possède ici 3 degrés de liberté:

$$\xi_I = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

- ◆ Aussi:

$$\xi_R = {}^R F_I \xi_I$$

$$\xi_I = ({}^R F_I)^{-1} \xi_R = {}^I F_R \xi_R$$

- ◆ Souvent, même en sachant que le robot évolue dans un espace 3D (le robot évolue à une hauteur variable), on utilise des techniques de projection dans le plan, notamment pour le contrôle et le suivi de trajectoire.
- ◆ L'élaboration de la cinématique directe d'un robot mobile se fait moins *machinalement* que dans le cas d'un robot sériel statique.

# Rappel Cours #11

## Robots mobiles : Premières notions de cinématique (2)

- Très souvent, nous nous intéressons à la vitesse du robot dans le repère de référence (universel).

- Directement:

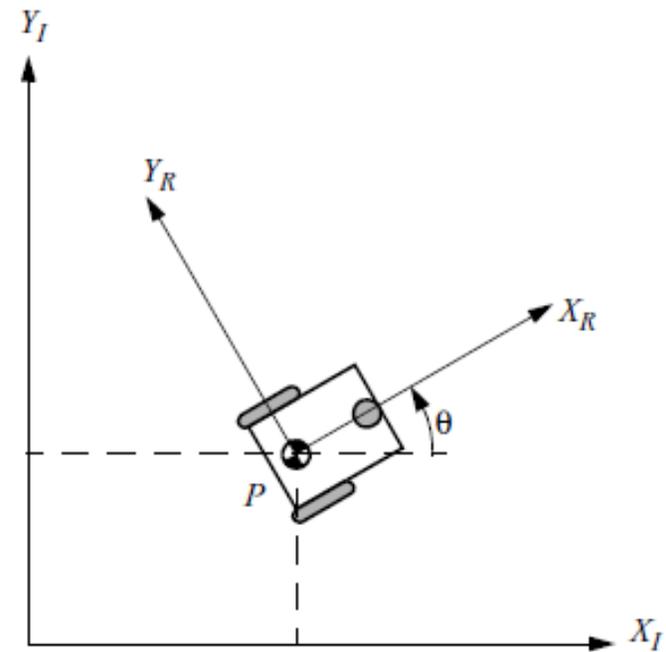
$$\dot{\xi}_R = {}^R T_I \dot{\xi}_I$$

- Où:

$${}^R T_I = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Donc:

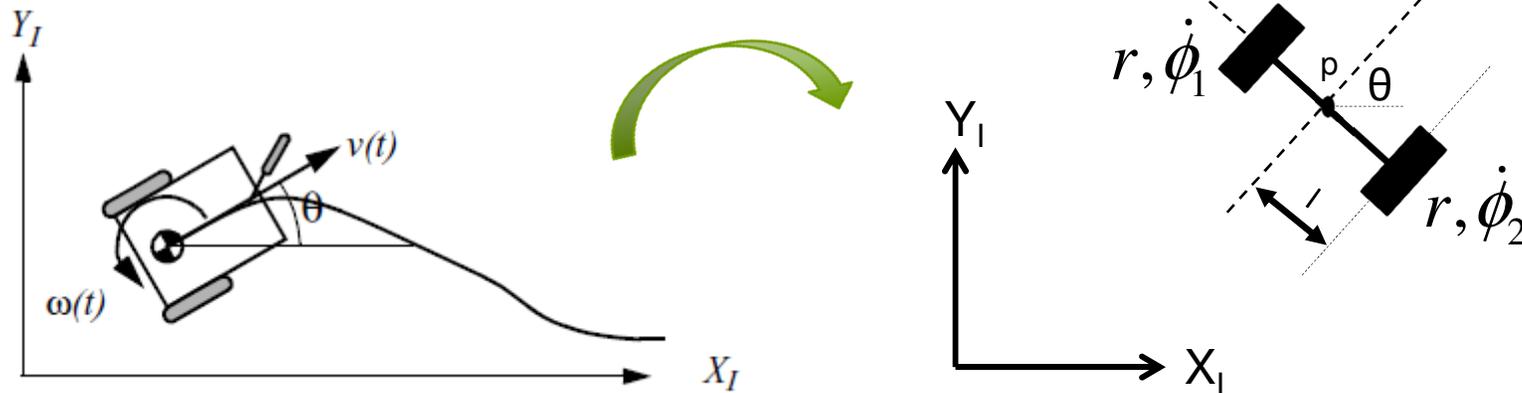
$$\dot{\xi}_I = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \dot{\xi}_R = ({}^R T_I)^T \dot{\xi}_R$$



# Rappel Cours #11

## Robots mobiles : Premières notions de cinématique (3)

- ◆ 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?

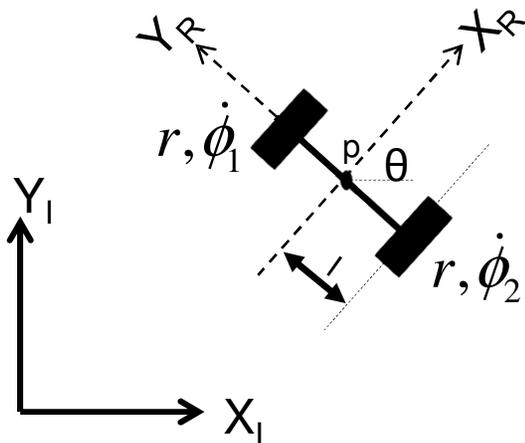
# Rappel Cours #11

## Robots mobiles : Premières notions de cinématique (4)

◆ Nous avons déjà mentionné que:

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

- ◆ Nous supposons ici que nous connaissons l'orientation initiale  $\theta$ .
  - ◆ 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  $\theta$ , 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  $\theta = 0$  et calculons la contribution de chacune des roues sur la vitesse de déplacement en  $x$ , en  $y$  et sur la vitesse de rotation.

# Rappel Cours #11

## Robots mobiles : Premières notions de cinématique (5)

- 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:

$$v_1 = r\dot{\phi}_1 \quad \text{où } r \text{ est le rayon de la roue en mètre et } \dot{\phi}_1 \text{ la vitesse angulaire de la roue en } \text{rad/sec}$$

- 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:

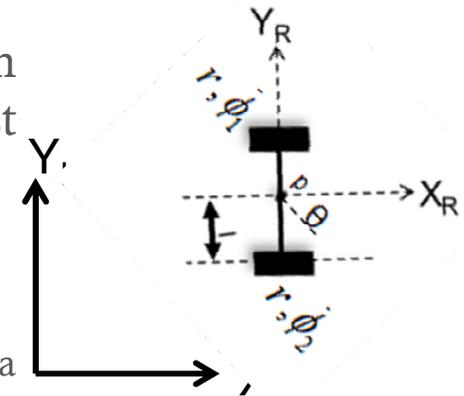
$$\dot{x}_{r1} = \frac{r\dot{\phi}_1}{2}$$

- 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:

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

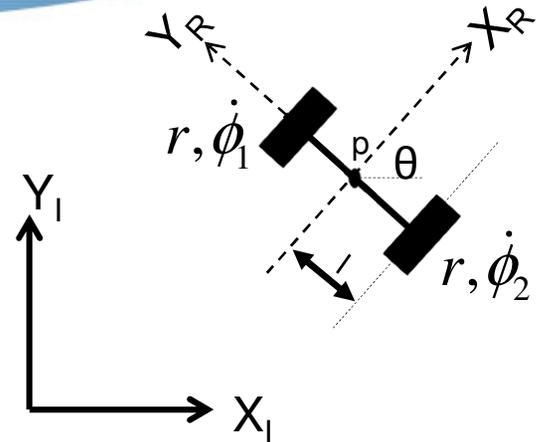


# Rappel Cours #11

## Robots mobiles : Premières notions de cinématique (6)

- Donc, en résumé:

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



- 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  $\theta$  simplement de cette façon:

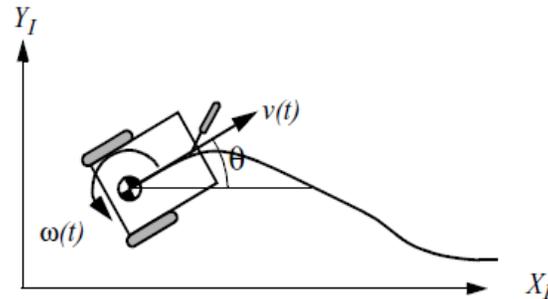
$$\theta(t) = \theta_0 + \int_{t_0}^t \dot{\theta}(t) dt$$

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

# Rappel Cours #11

## Robots mobiles : Premières notions de cinématique (7)

- ◆ *Remarque:* À la base, le robot que nous considérons était celui-ci:



- ◆ Dans l'élaboration de notre modèle, nous avons négligé la roue non-actionnée (celle qui pivote à l'avant du robot).

- ◆ Est-ce grave?



- ◆ *Autres types de roues qui n'amènent pas de contraintes sur les mouvements possibles du robot:*

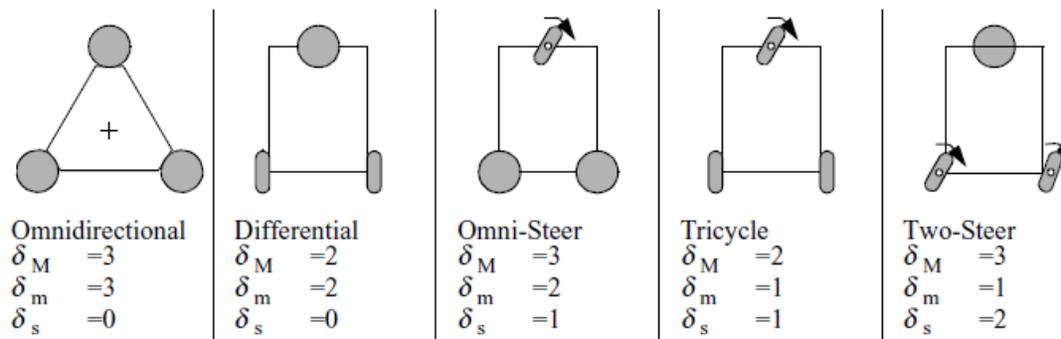
- ◆ Roues pivotantes standards, roues sphériques, roues suédoises...

# Rappel Cours #11

## Robots mobiles : Manoeuvrabilité

### Donnons ici quelques définitions:

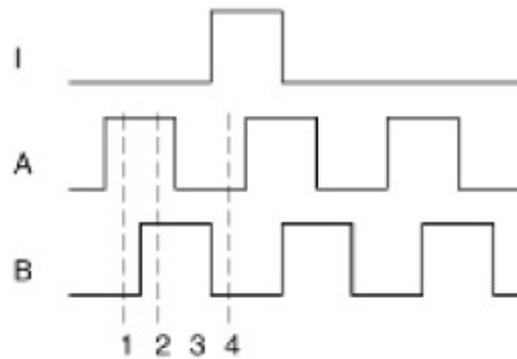
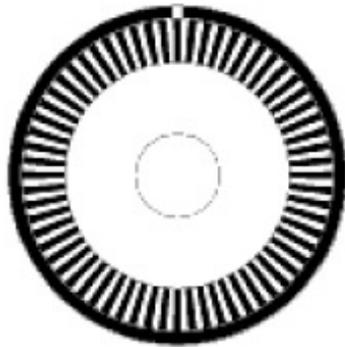
- Le **degré de mobilité** d'un robot mobile est le nombre de variables de la pose sur lesquelles il est possible d'agir simplement en changeant la vitesse des roues.
- Le **degré de directionnalité** (*steerability*) d'un robot mobile est le nombre de variables supplémentaires de la pose sur lesquelles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



# Rappel Cours #11

## Robots mobiles : Perceptions et capteurs (1)

- Capteurs de roues: Les encodeurs optiques:



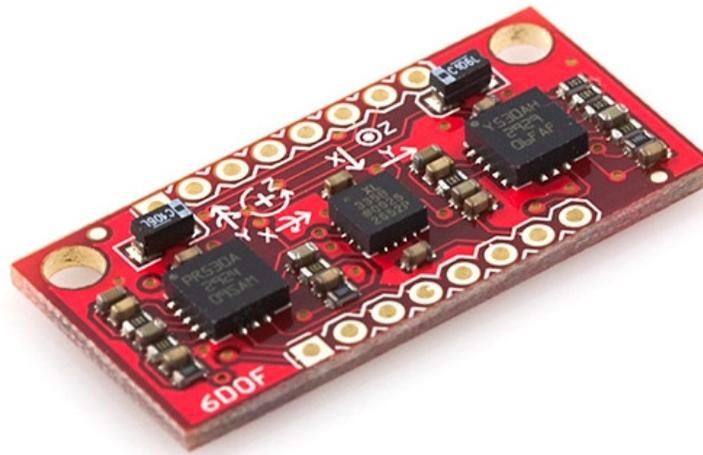
State	Ch A	Ch B
$s_1$	high	low
$s_2$	high	high
$s_3$	low	high
$s_4$	low	low

- Génèrent souvent des ondes en *quadratures*.

# Rappel Cours #11

## Robots mobiles : Perceptions et capteurs (2)

- La centrale inertielle ou IMU pour *inertial measurement unit*

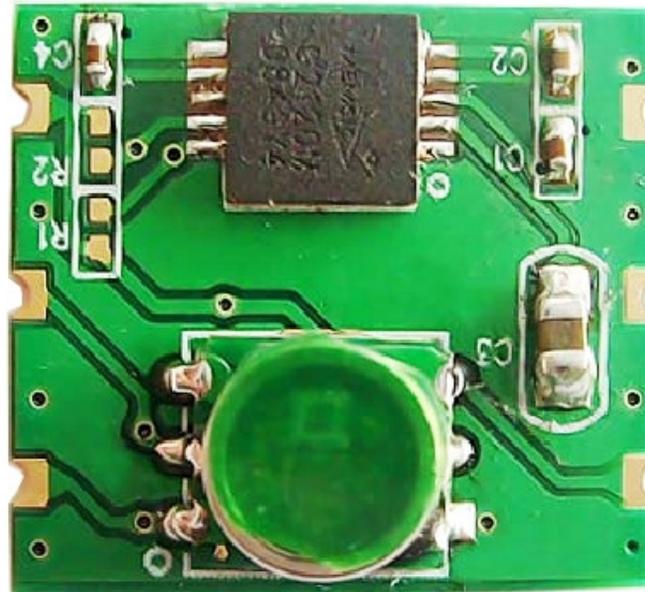


- Trois gyroscopes + trois accéléromètres pour connaître les accélérations et vitesses angulaires

# Rappel Cours #11

## Robots mobiles : Perceptions et capteurs (3)

- ◆ *La boussole*



- ◆ Trois gyroscopes + trois accéléromètres pour connaître les accélérations et vitesses angulaires

# Rappel Cours #11

## Robots mobiles : Perceptions et capteurs (4)

- Le GPS (lorsque c'est possible) pour *global positioning system*:



- Moyen très efficace pour se localiser dans un repère global.

# Rappel Cours #11

## Robots mobiles : Perceptions et capteurs (5)

- Les capteurs ultrasons:

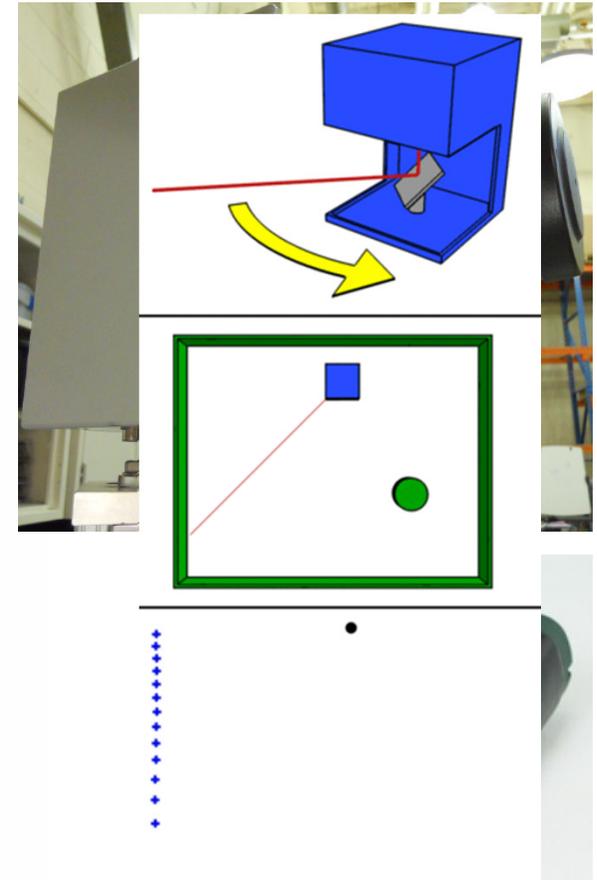


- Permettent de mesurer des distances en mesurant le temps de vol d'un ultrason et de son echo.

# Rappel Cours #11

## Robots mobiles : Perceptions et capteurs (6)

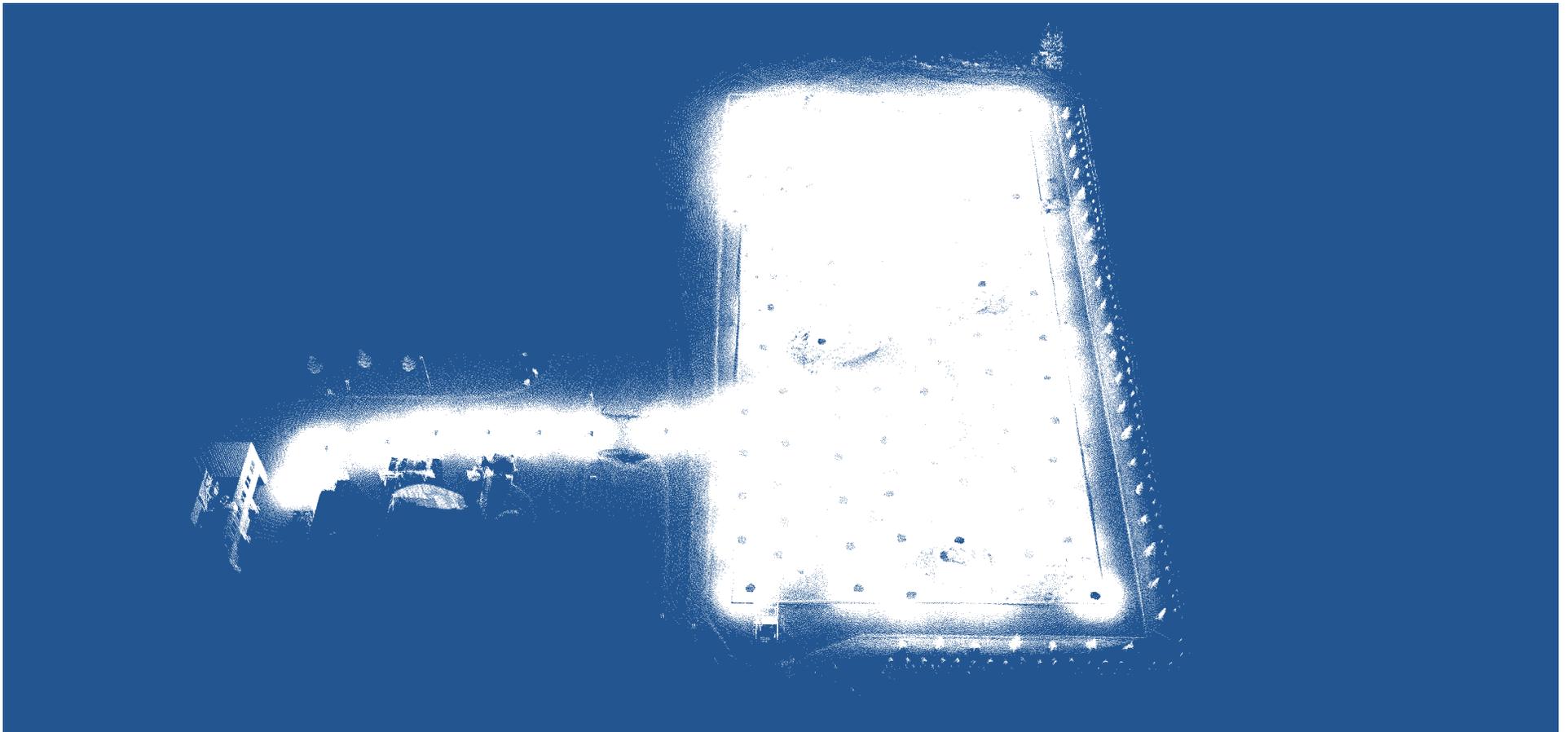
- Les télémètres laser:



- Permettent de mesurer des distances en mesurant le temps de vol d'un ou plusieurs rayon laser

# Rappel Cours #11

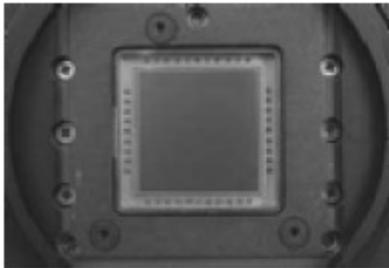
## Robots mobiles : Perceptions et capteurs (7)



# Rappel Cours #11

## Robots mobiles : Perceptions et capteurs (8)

### ◆ Caméras CCD:



2048 x 2048 CCD array



Orangemicro iBOT Firewire



Sony DFW-X700



Cannon IXUS 300

# Rappel Cours #11

## Robots mobiles : Perceptions et capteurs (9)

- ◆ Caméras stéréoscopiques:



- ◆ Permet la vision 3D, il faut résoudre le problème d'appariement

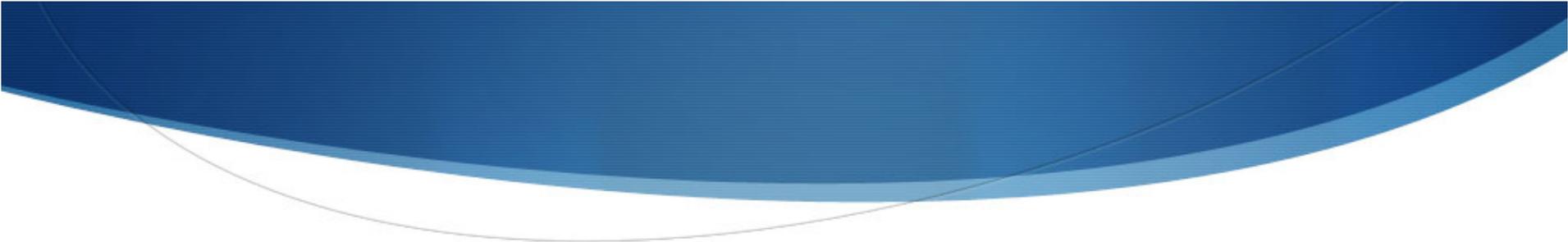
# Rappel Cours #11

## Robots mobiles : Perceptions et capteurs (10)

- ◆ Radars:



- ◆ Permet la détection d'obstacles à grandes distances. Peu sensible à la poussière.



*Le robot de la semaine:*

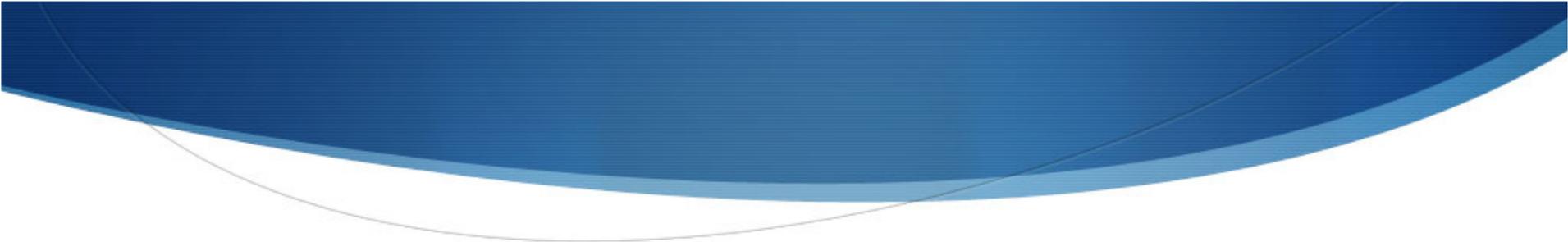
*ACE: The Autonomous City Explorer*

*Il s'agit d'une robot intéressant entres autres parce qu'il regroupe plusieurs de notions que nous avons vu jusqu'à présent au niveau de la robotique mobile:*



C:\Users\  
je\Desktop\ELE42C





# *Cours #12:*

*Localisation, contrôle et planification de trajectoires*

## Cours #12

# Cinématique des robots mobiles: Références supplémentaires

- Voici quelques références plus adaptées au cours d'aujourd'hui:

[1] Cook, Gerald, *Mobile Robots: Navigation, Control and Remote Sensing*. IEEE Press, Piscataway, NJ 2011.

[2] Astolfi, A., *Exponential stabilization of a wheeled mobile robot via discontinuous control*. Journal of Dynamic systems, measurement and control, ASME, 1999.

[3] Feng, L., Borenstein, J., Everett, H. R., *Where I am: Sensors and Methods for Autonomous Mobile Robot Positioning*, University of Michigan, 1994.

[4] Notes de cours usuelles: *Introduction – robots mobiles & Suivi de chemin pour un robot mobile à action différentielle*.

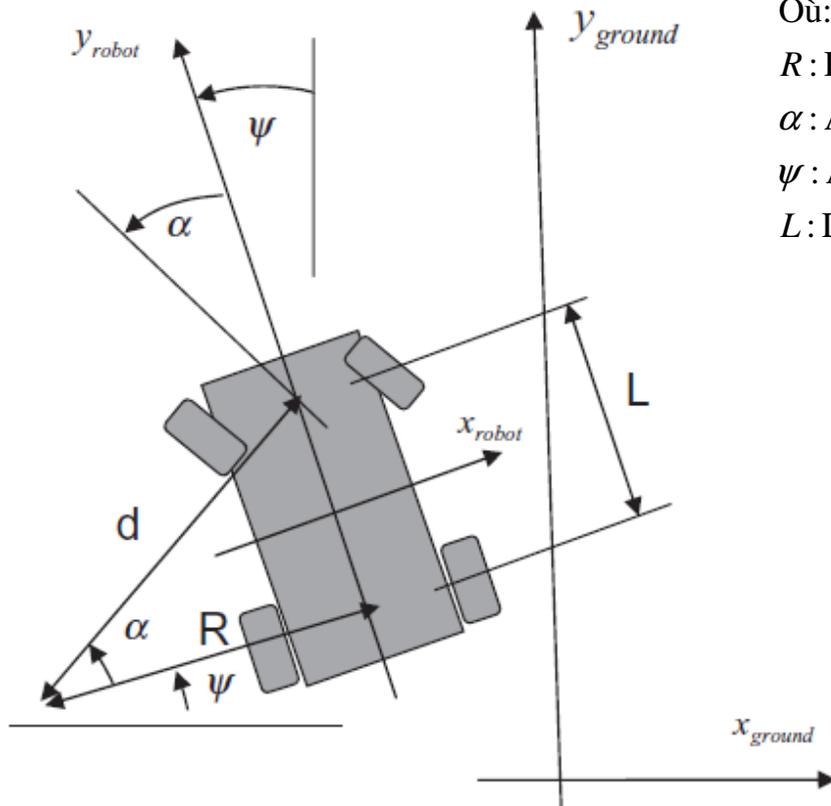
[5] Norwegian Space Institute, *Introduction to Inertial Navigation and Kalman filtering*, Forsvarets forskningsinstitutt, 2008.

## Cours #12

### Cinématique des robots mobiles: un autre exemple (1)

- Avant, d'aller plus loin, considérons un autre exemple de robot mobiles:

- \*\*Image tirée de [1]*



Où:

$R$  : Rayon de courbure instantané

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

$\psi$  : 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  $\alpha=0$

- Les roues arrières propulsent le véhicule, tandis que les roues avant le dirigent.

- Par simple géométrie, nous savons que:

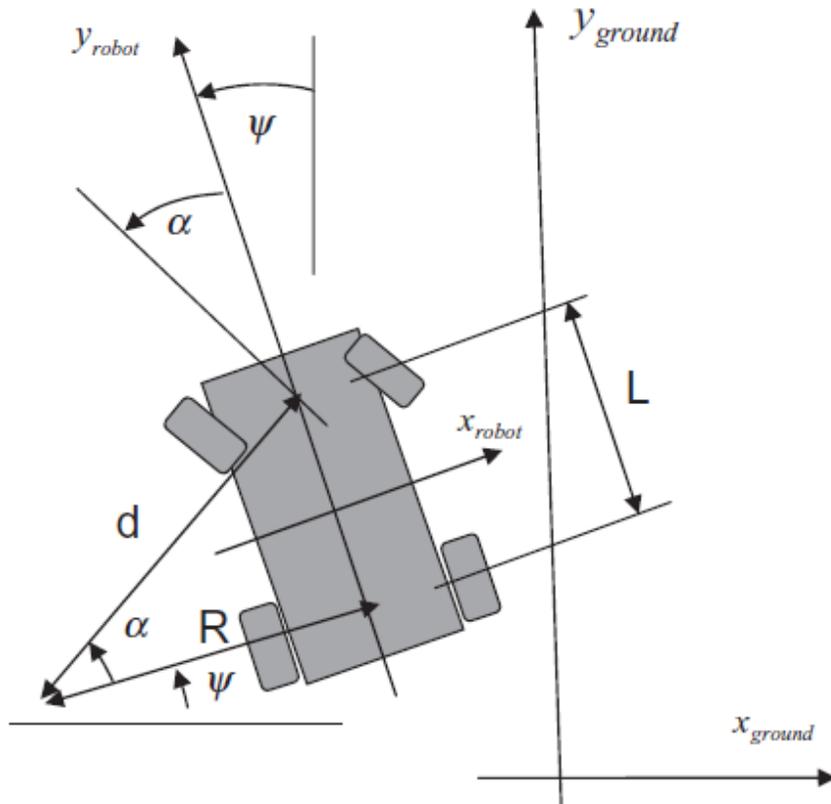
$$\tan(\alpha) = \frac{L}{R} \Rightarrow R = \frac{L}{\tan(\alpha)}$$

# Cours #12

## Cinématique des robots mobiles: un autre exemple (2)

- Suite:

- \*\*Image tirée de [1]



- Aussi par simple constatation:

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

- En substituant pour le rayon de courbure instantané:

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

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

$$\dot{x}_{robot} = 0$$

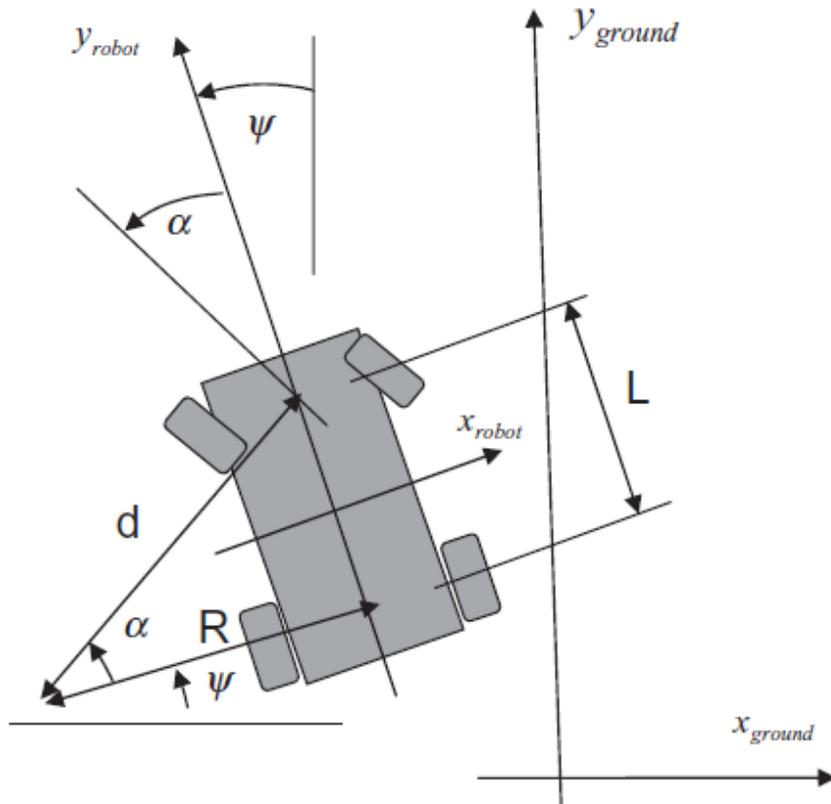
$$\dot{y}_{robot} = v_{roues\_arr}$$

## Cours #12

### Cinématique des robots mobiles: un autre exemple (3)

- Suite:

- \*\*Image tirée de [1]



- Aussi par simple constatation:

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

- En substituant pour le rayon de courbure instantané:

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

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

$$\dot{x}_{robot} = 0$$

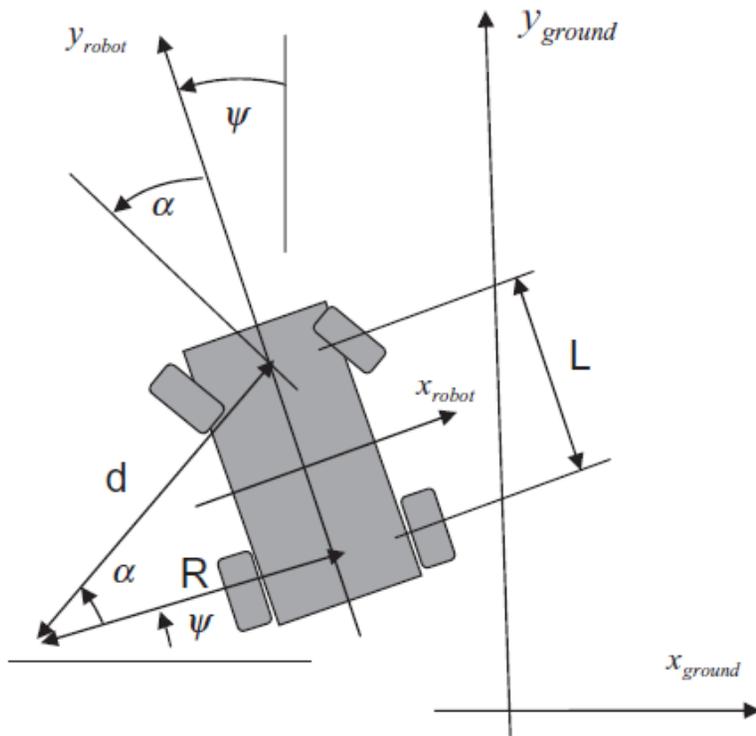
$$\dot{y}_{robot} = v_{roues\_arr}$$

# Cours #12

## Cinématique des robots mobiles: un autre exemple (4)

Suite:

\*\*Image tirée de [1]



On connaît 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):

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

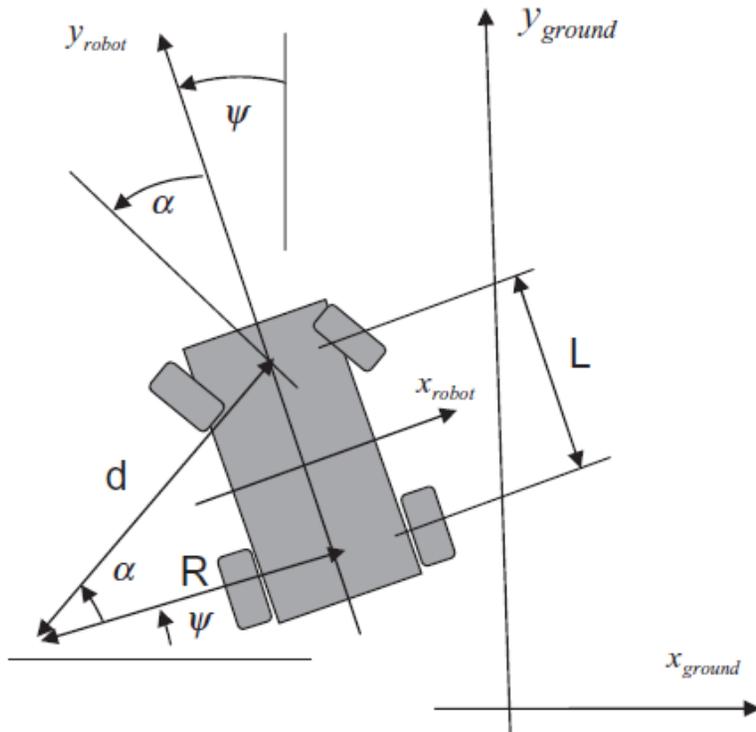
$$\dot{\xi}_{ground} = \underbrace{\begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{ground T_R} \begin{bmatrix} 0 \\ v_{roues\_arr} \\ \frac{v_{roues\_arr}}{L} \tan(\alpha) \end{bmatrix} = \begin{bmatrix} -\sin(\psi) v_{roues\_arr} \\ \cos(\psi)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

# 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:

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

- 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  $\psi$ . 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 (2)

### ◆ Rudolf E. Kálmán (1930-...)

- ◆ Ingénieur électrique de formation (MIT), mathématicien de la théorie des systèmes, automaticien.
- ◆ La contribution la plus reconnue de Kalman est le *filtre de Kalman*
  - ◆ Le filtre de Kalman, comme nous le verrons, est grandement utilisé dans les systèmes de contrôle, en avionique, en navigation en général (avion, robot mobile, sous-marin, satellite, guidage pour missile, GPS, etc...). Il est extrêmement répandu!
  - ◆ L'invention de ce filtre se fait durant une période très mouvementée de l'histoire américaine: le début des années 60 (*Space Race* qui est sur le point de culminer, tensions avec l'union soviétique).
- ◆ Lorsque le filtre fut inventé, Kalman dû faire face à beaucoup de scepticisme, son premier article ne fut presque pas remarqué! (Il a d'ailleurs dû publier en génie mécanique!)
- ◆ En 1960, il convainc par contre Stanley Schmidt de la NASA Ames Research Center de la valeur de son filtre. Résultat: le filtre est utilisé durant les missions Apollo!

# 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 entre 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 :

$$Y_K = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \end{bmatrix}$$

$$V_K = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_k \end{bmatrix}$$

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

$$\text{cov}(V_K) = R$$

## 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 meilleure estimation de  $x$ . Pour ce faire, on veut trouver l'estimation de  $x$  qui minimise cet indicateur:

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

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

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

- 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:

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

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

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

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

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

- Re-visitons ce dernier résultat:

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

- 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  $x$ :

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

- Le deuxième élément de  $J$  est encore l'erreur au carré 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?



```
1 - clc; clear all; close all;
2 - R = 0.04;
3 - P = R;
4 - RealValue=5;
5 - y(1) = RealValue + sqrt(R) * randn;
6 - xest(1) = y(1); K(1)=NaN;
7 - for i = 1:49
8 -     y(i + 1) = RealValue + sqrt(R) * randn;
9 -     P = 1 / ((1 / P) + (1 / R));
10 -    K(i+1) = P / (P + R);
11 -    xest(i + 1) = xest(i) + K(i+1) * (y(i + 1) - xest(i));
12 - end
13
14 %Tracer le graphique:
15 figure
16 plot(RealValue*ones(1,numel(y)),['k']);
17 hold on
18 plot(y,['r' '.'])
19 hold on
20 plot(xest,['b' '--'],'linewidth',2)
21 legend('Valeur réelle','Mesures','Estimation')
```

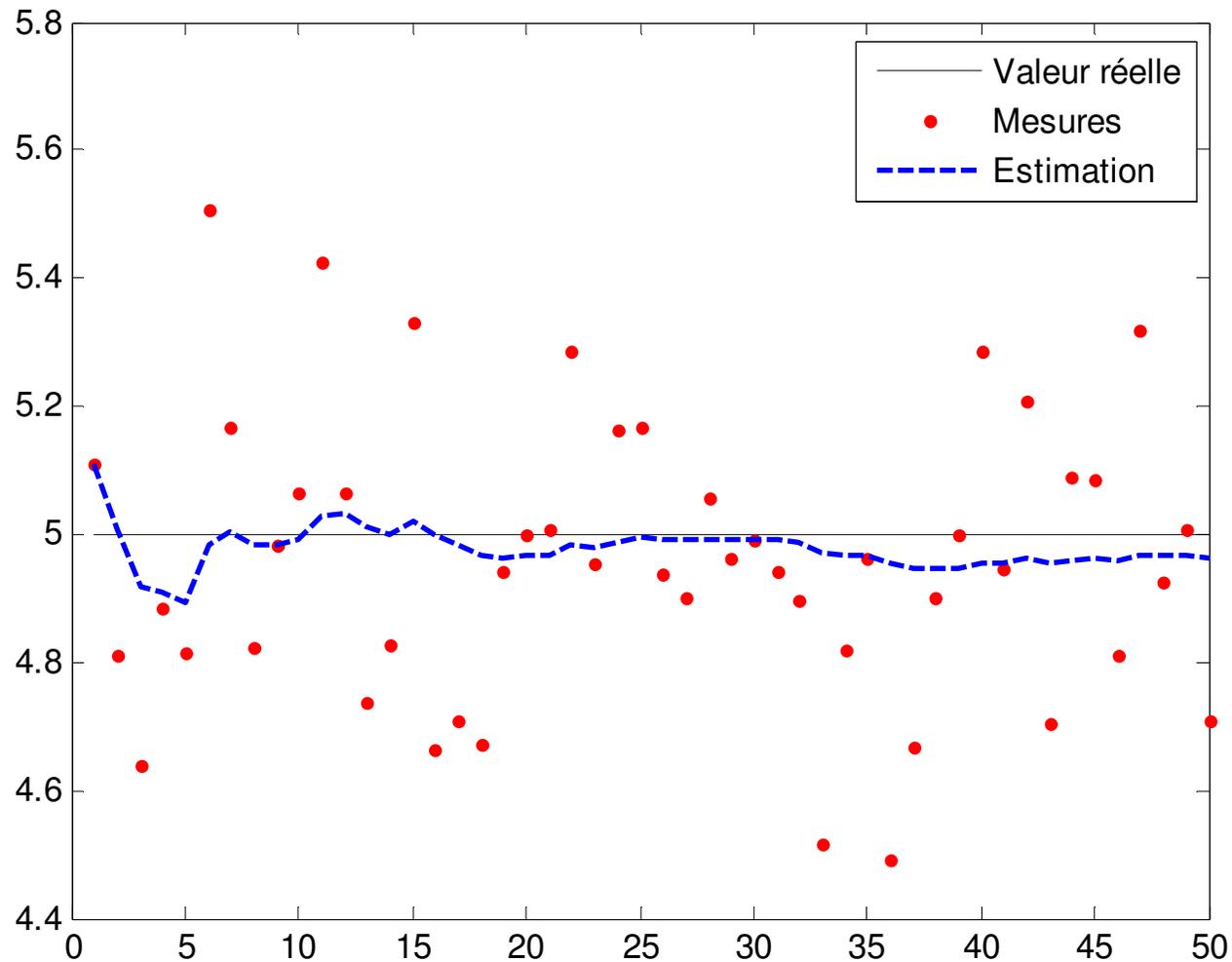
$$\hat{x}_{K+1} = \hat{x}_K + K_{K+1} [y_{K+1} - H_{K+1} \hat{x}_k]$$

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

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

# 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:

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

$$Y((k+1)T) = HX((k+1)T) + v((k+1)T)$$

- 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:

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

- Aussi, introduisons la notation suivante:

$Y_k$  signifie la sortie au temps  $Y(kT)$

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

$\hat{X}_{k/k}$  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*:

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

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

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

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

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

## 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 trouver le meilleur estimé de la position du véhicule **ET** de la vitesse. De plus, nous savons que la variance du capteur servant à mesurer 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'entrée  $u$ , seulement pour les dix premières périodes de temps  $T$  et que la valeur du signal d'entrée 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:

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

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

```
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;
```

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

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

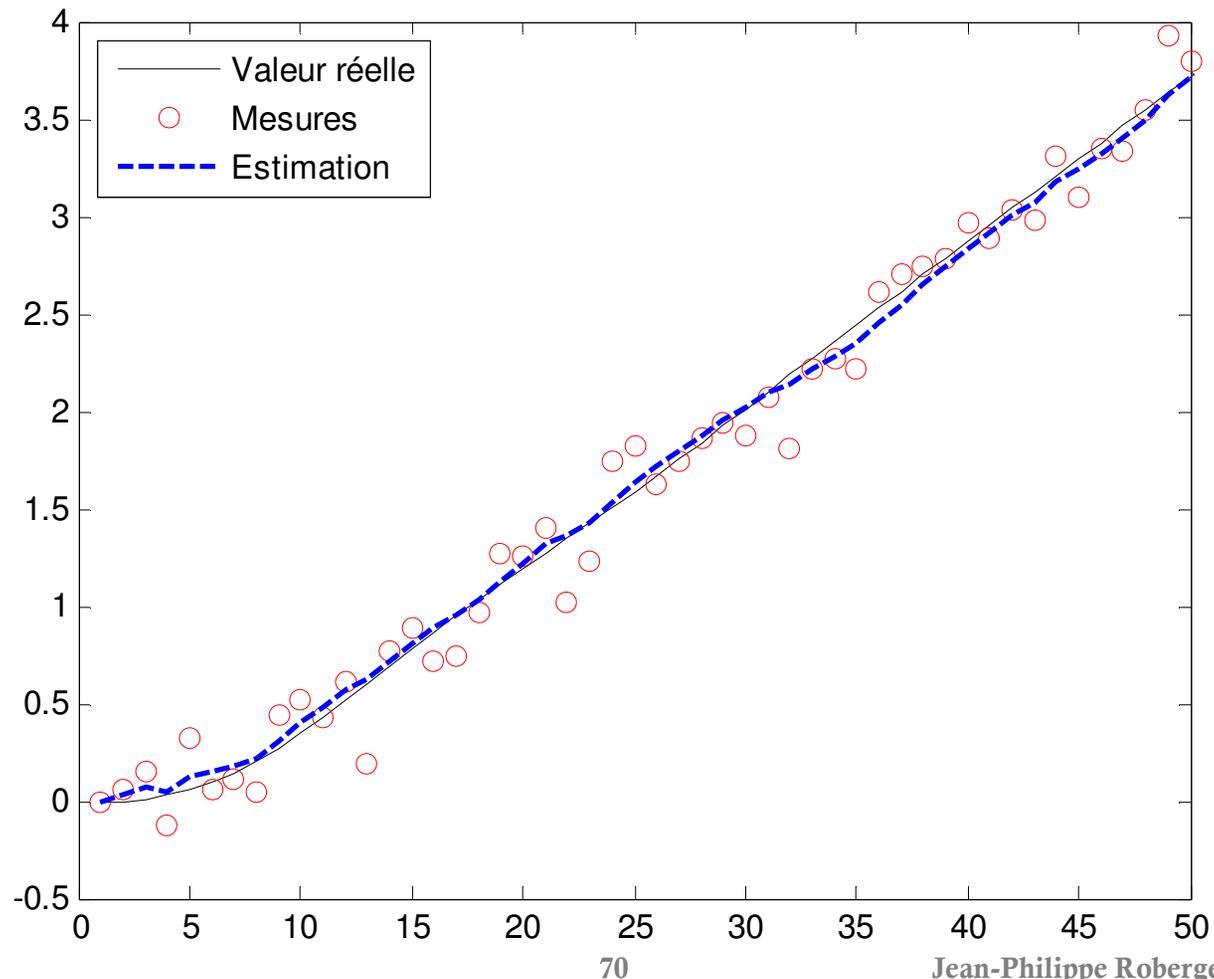
$$P_{k+1/k} = AP_{k/k} A^T + GQG^T$$

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

```
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
```

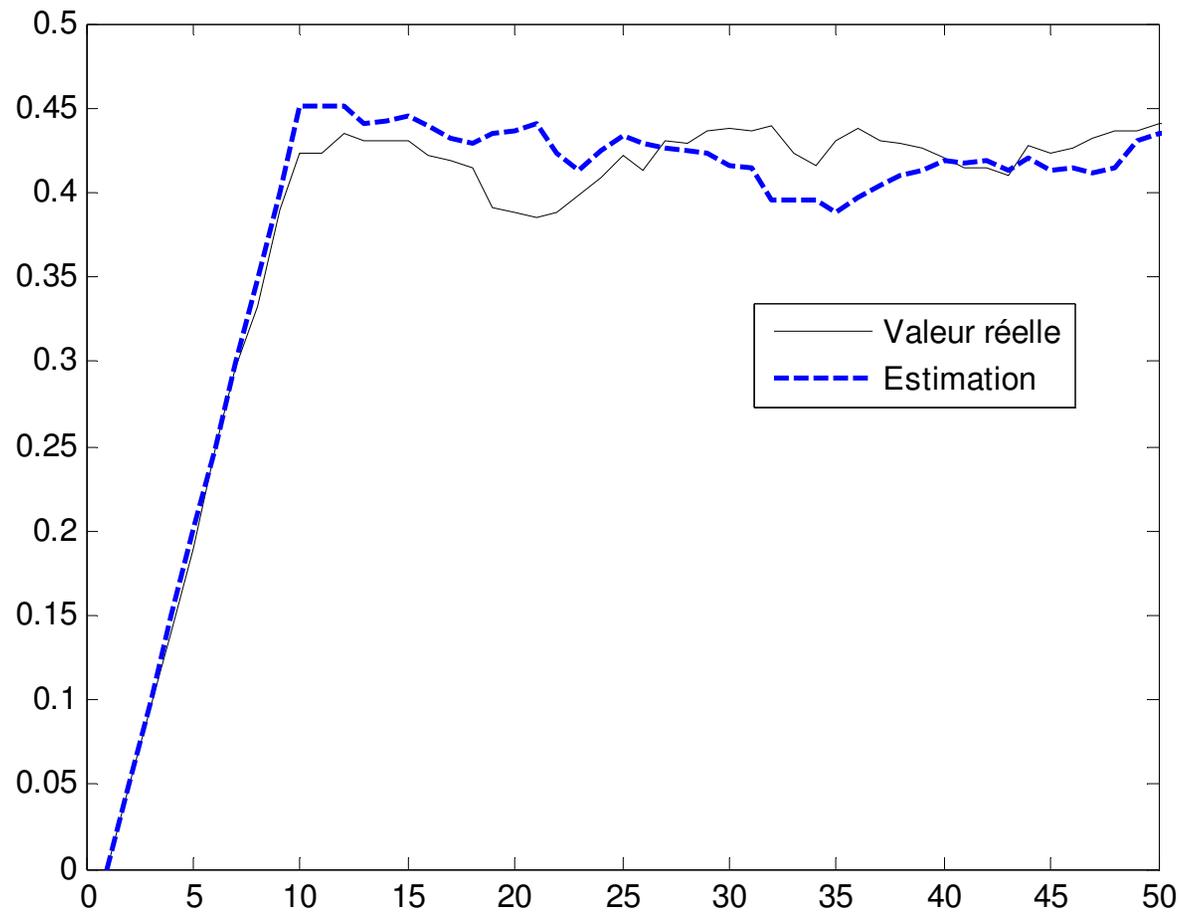
# Cours #12 – Filtre de Kalman (réf: [1]): Estimation des états d'un sys. dyn. par un processus récursif (5)

- ◆ L'état 1, c'est-à-dire la position (Rappel: nous avons 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'**avons **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:

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

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

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



Où:

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

# 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...

$$\dot{\Psi}_x = \Omega_z \Psi_y + \varepsilon_x$$

$$\dot{\Psi}_y = \Omega_x \Psi_z - \Omega_z \Psi_x + \varepsilon_y$$

$$\dot{\Psi}_z = -\Omega_x \Psi_y + \varepsilon_z$$

$\Psi_x$  : Erreur de longitude du système de navigation inertielle

$\Psi_y$  : Erreur de latitude du système de navigation inertielle

$\Psi_z$  : (Erreur d'azimut de la plateforme inertielle) – (Erreur de longitude) tan (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 (II)

## ◆ Ce que l'on connaît à priori:

- ◆ La latitude ( $45^\circ$  N supposée)
- ◆  $T=0.5$  heure
- ◆ Variance du processus de Wiener-Brown:  $(0.001 \text{ deg/heure})^2$
- ◆  $\Omega=0.2625161$  rad/heure,

◆ F, G

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$W = \begin{bmatrix} (0.001\pi/180)^2 & 0 & 0 \\ 0 & (0.001\pi/180)^2 & 0 \\ 0 & 0 & (0.001\pi/180)^2 \end{bmatrix}$$

Variance positionnement:  $\left(\frac{1000m}{R}\right)^2$

Variance Azimut et Longitude combinée:  $(0.1\text{deg})^2$

Variance du biais des gyroscopes:  $(0.02\text{deg})^2$

Variance des mesures de position:  $\left(\frac{100m}{R}\right)^2$

Variance de l'Azimut:  $(1 \text{ arc min})^2$

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

## 🔹 Filtre de Kalman – Calculs:

### 🔹 Matrice de transition et Q (Méthode de Van Loan) :

```

PHI =
    0.9957    0.0925    0.0043    0.4993    0.0232    0.0007    1.0e-007 *
   -0.0925    0.9914    0.0925   -0.0232    0.4986    0.0232
    0.0043   -0.0925    0.9957    0.0007   -0.0232    0.4993    0.0025    0    0
     0         0         0    1.0000     0         0         0    0.0025    0
     0         0         0         0    1.0000     0         0         0    0.8462
     0         0         0         0         0    1.0000
PZEROM =

R =
    1.0e-005 *
     0.0025    0    0    0    0    0
     0    0.0025    0    0    0    0
     0    0    0.3046    0    0    0
     0    0    0    0.0122    0    0
     0    0    0    0    0.0122    0
     0    0    0    0    0    0.0122
    0.1523    0    0    0    0    0.1523

Q =
    1.0e-009 *
     0.0127   -0.0000    0.0000    0.0380    0.0012    0.0000
    -0.0000    0.0127    0.0000   -0.0012    0.0380    0.0012
     0.0000    0.0000    0.0127    0.0000   -0.0012    0.0380
     0.0380   -0.0012    0.0000    0.1523     0         0
     0.0012    0.0380   -0.0012     0         0.1523     0
     0.0000    0.0012    0.0380     0         0         0.1523
  
```

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

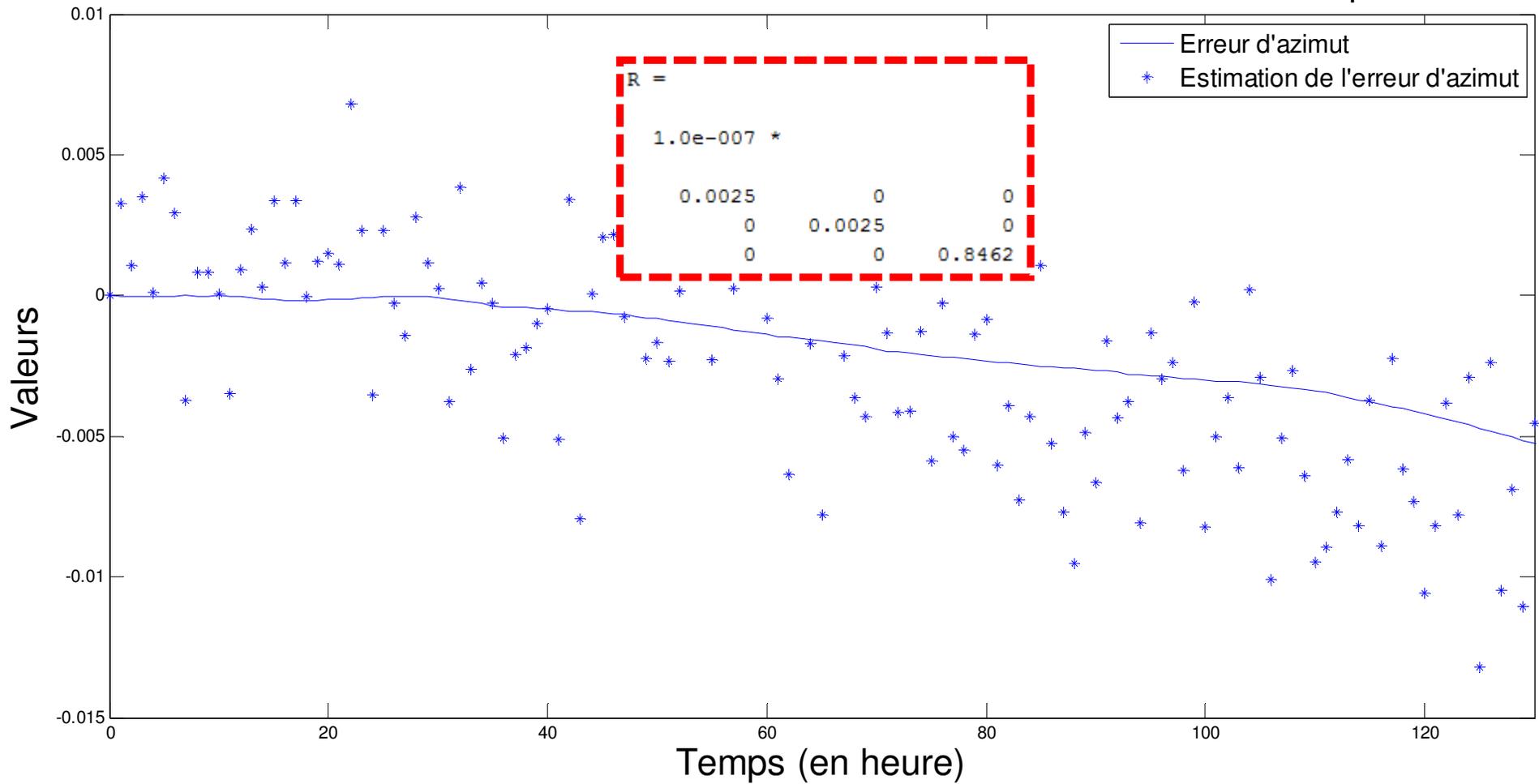
- ◆ **Solution** - Augmenter l'état du système de sorte à inclure le bruit dans l'état:

$$\begin{aligned}
 \dot{\mathcal{E}}_x &= f_x \\
 \dot{\mathcal{E}}_y &= f_y \\
 \dot{\mathcal{E}}_z &= f_z
 \end{aligned}
 \quad \longrightarrow \quad
 \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & \Omega_z & 0 & 1 & 0 & 0 \\ -\Omega_z & 0 & \Omega_x & 0 & 1 & 0 \\ 0 & -\Omega_x & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_F \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_G \begin{bmatrix} 0 \\ 0 \\ 0 \\ f_x \\ f_y \\ f_z \end{bmatrix}$$

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

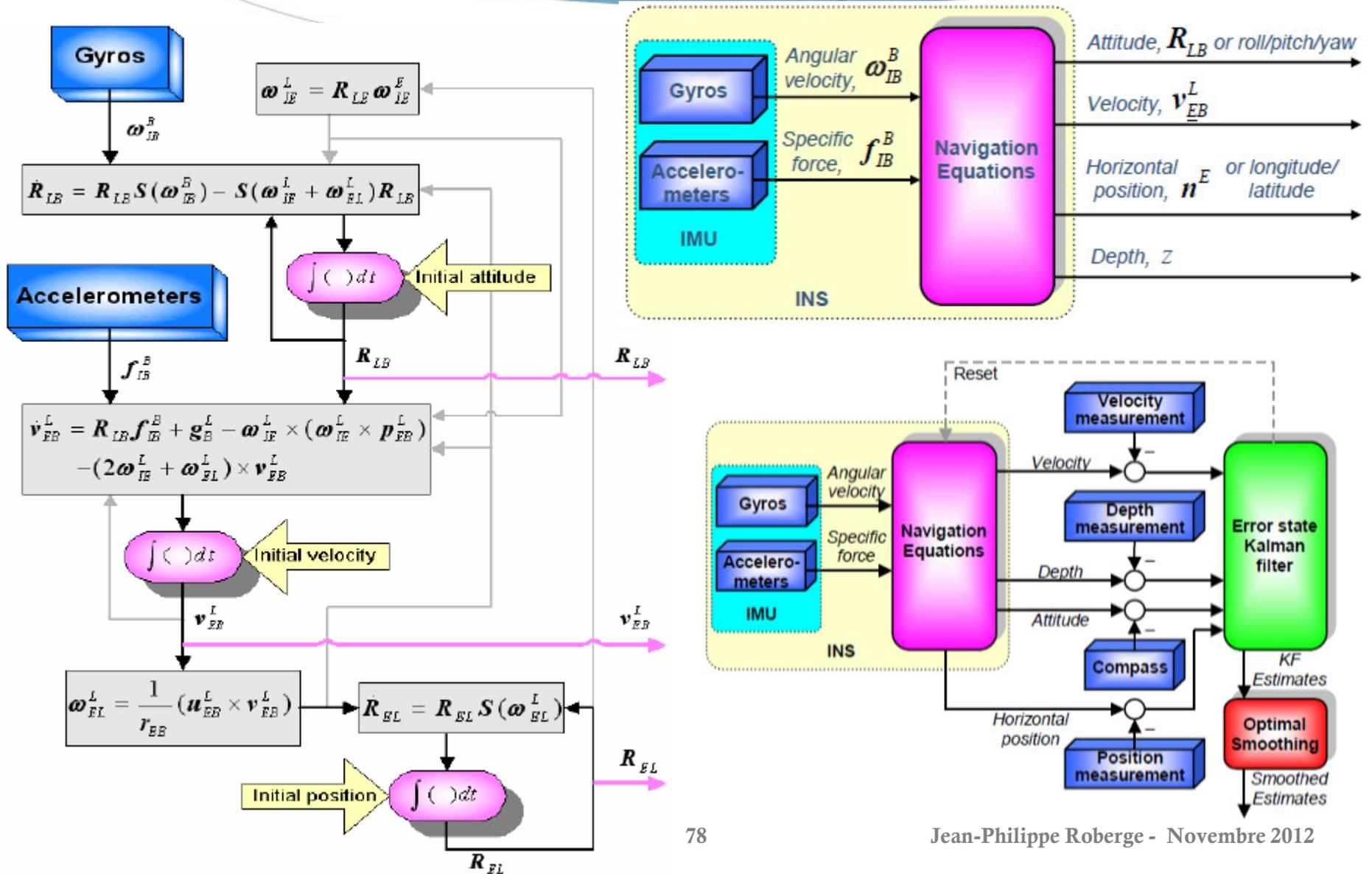
# Décalage des gyroscopes: Résultats (I)

Erreur d'azimut et estimation de l'erreur d'azimut en fonction du temps



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

*\*\*images tirées de [5]*



# Cours #12

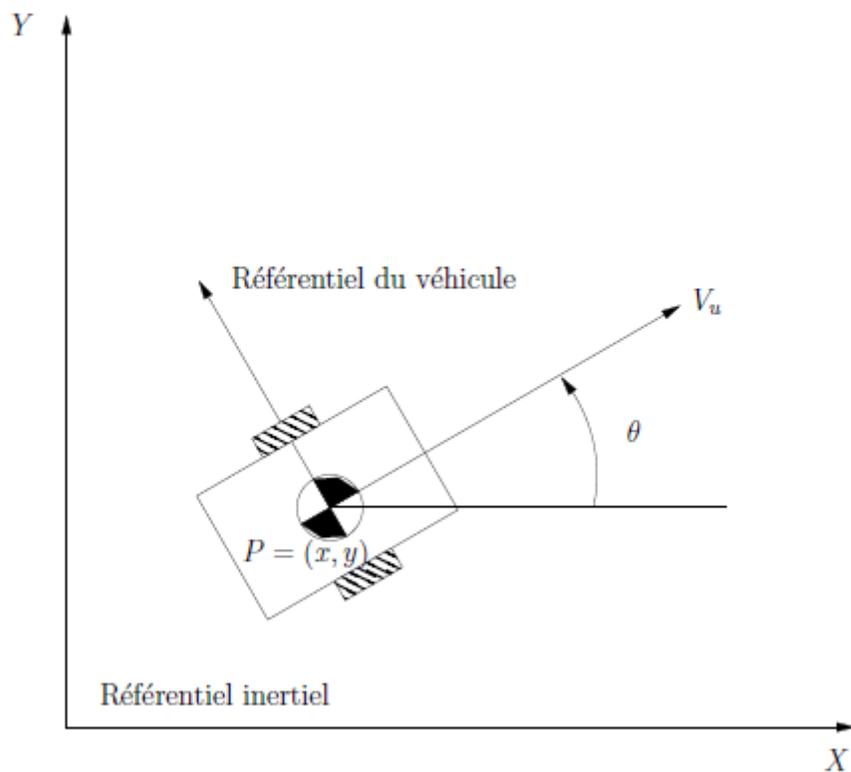
## Contrôleur classique de vitesse et rotation (1)

- ◆ 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.

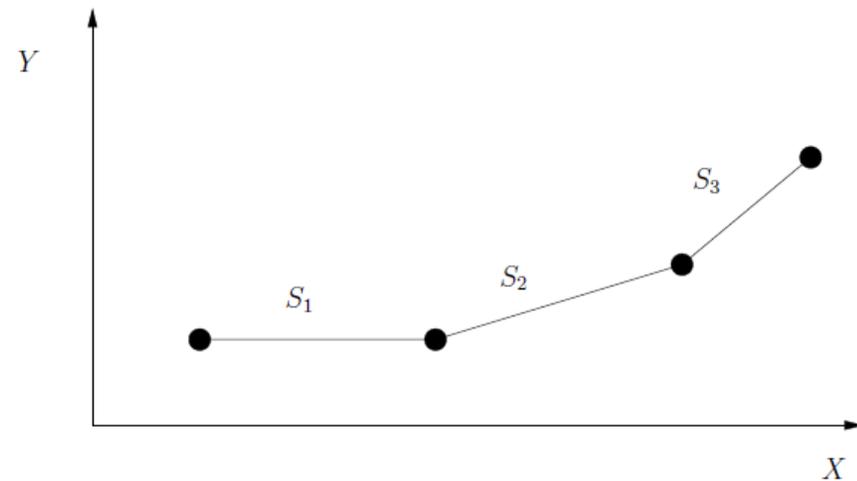
# Cours #12

## Contrôleur classique de vitesse et rotation (2)

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



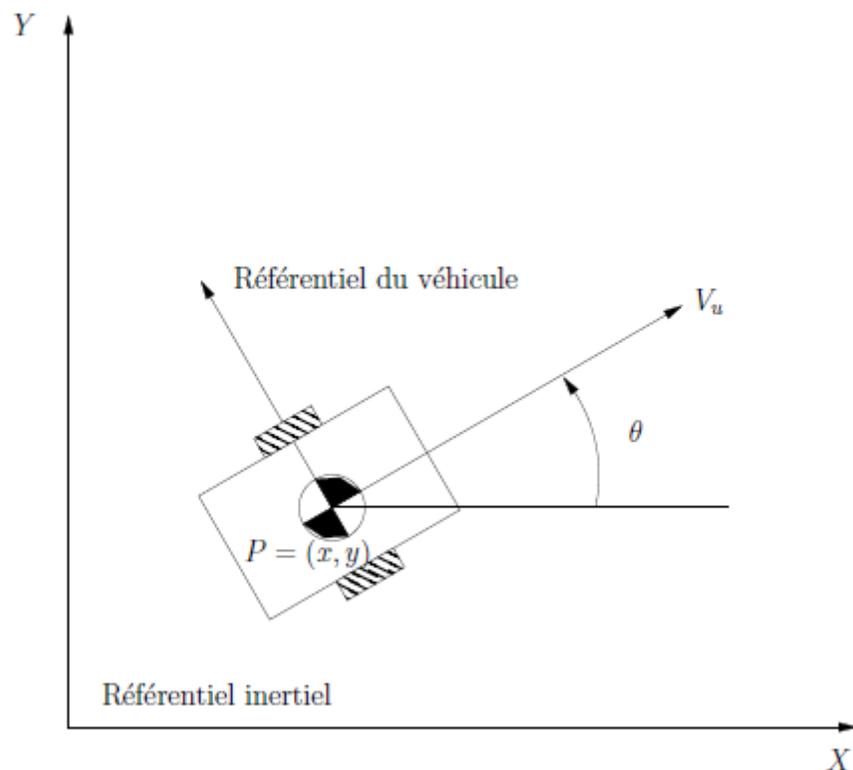
### Exemple de trajectoires à suivre:



# 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 prédéterminées 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:

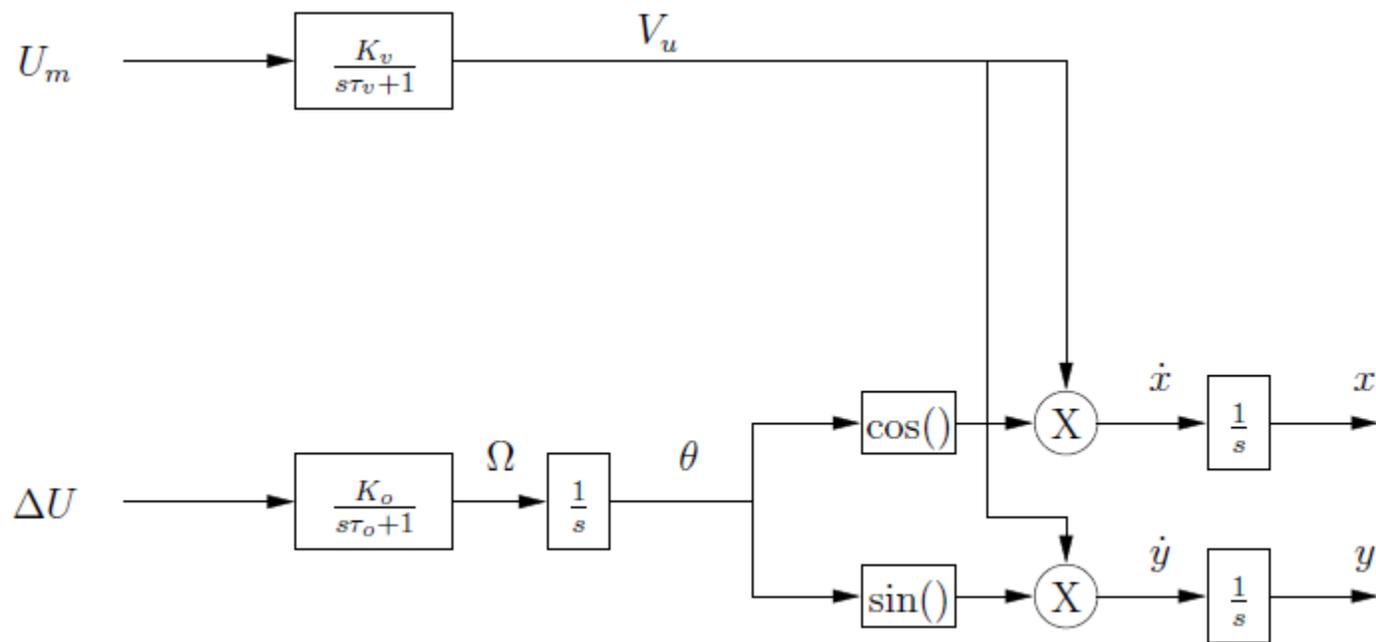
$$\frac{V_u}{U_m} = \frac{K_v}{s\tau_v + 1} \quad \text{et} \quad \frac{\Omega}{\Delta U} = \frac{K_o}{s\tau_o + 1}$$

$$\text{où: } U_m = U_1 + U_2 \quad \text{et} \quad \Delta U = U_1 - U_2$$

# Cours #12

## Contrôleur classique de vitesse et rotation (4)

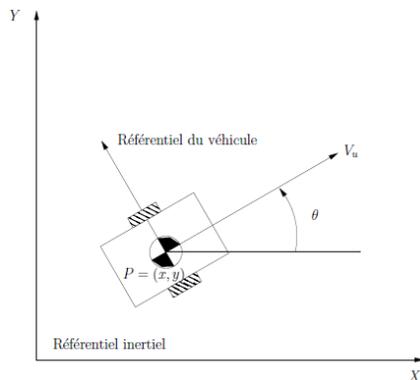
- ◆ Schéma du système en boucle ouverte:



# 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:

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

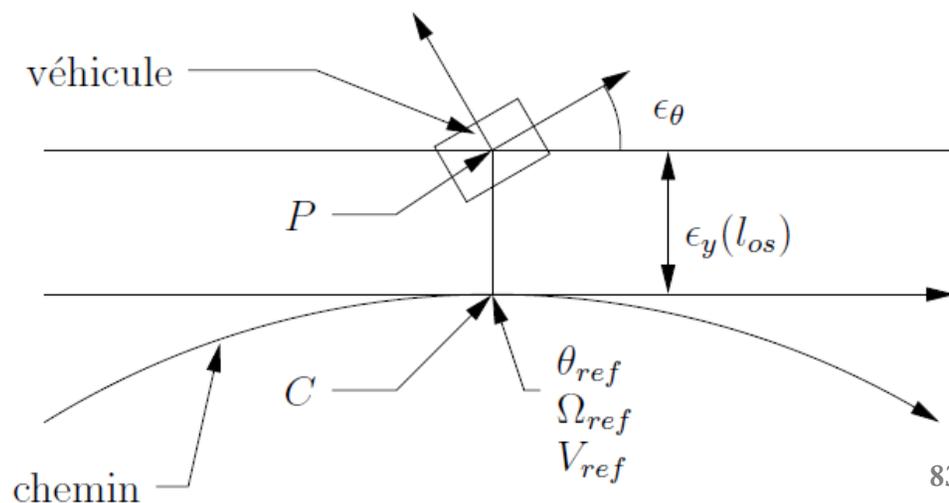
- En dérivant (voir vos notes de cours pour la démarche):

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

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

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

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

◆ 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  $\approx 1$ . 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  $\approx 0$  et cosinus  $\approx 1$  dans les équations ci-dessous:*

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

◆ 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:

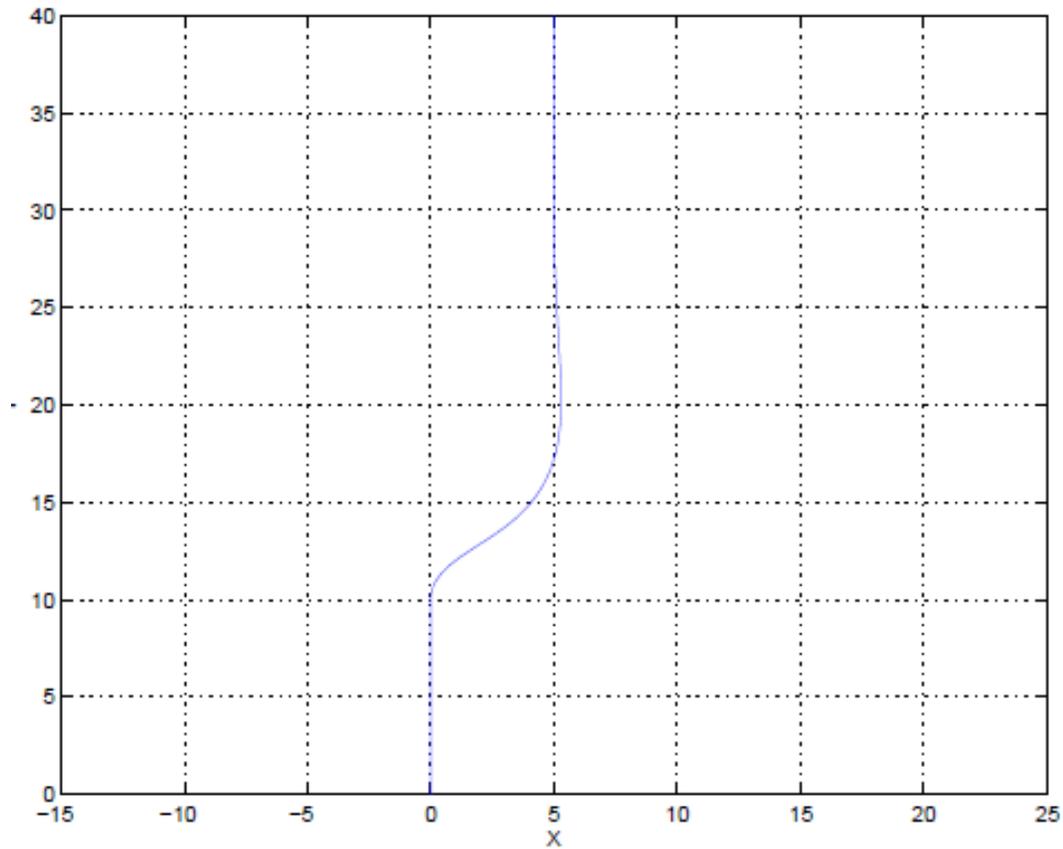
$$U_m = \frac{1}{K_v} \left[ V_u + K_p \tau_v (V_u^* - V_u) + K_I \tau_v \int (V_u^* - V_u) dt \right]$$

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

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

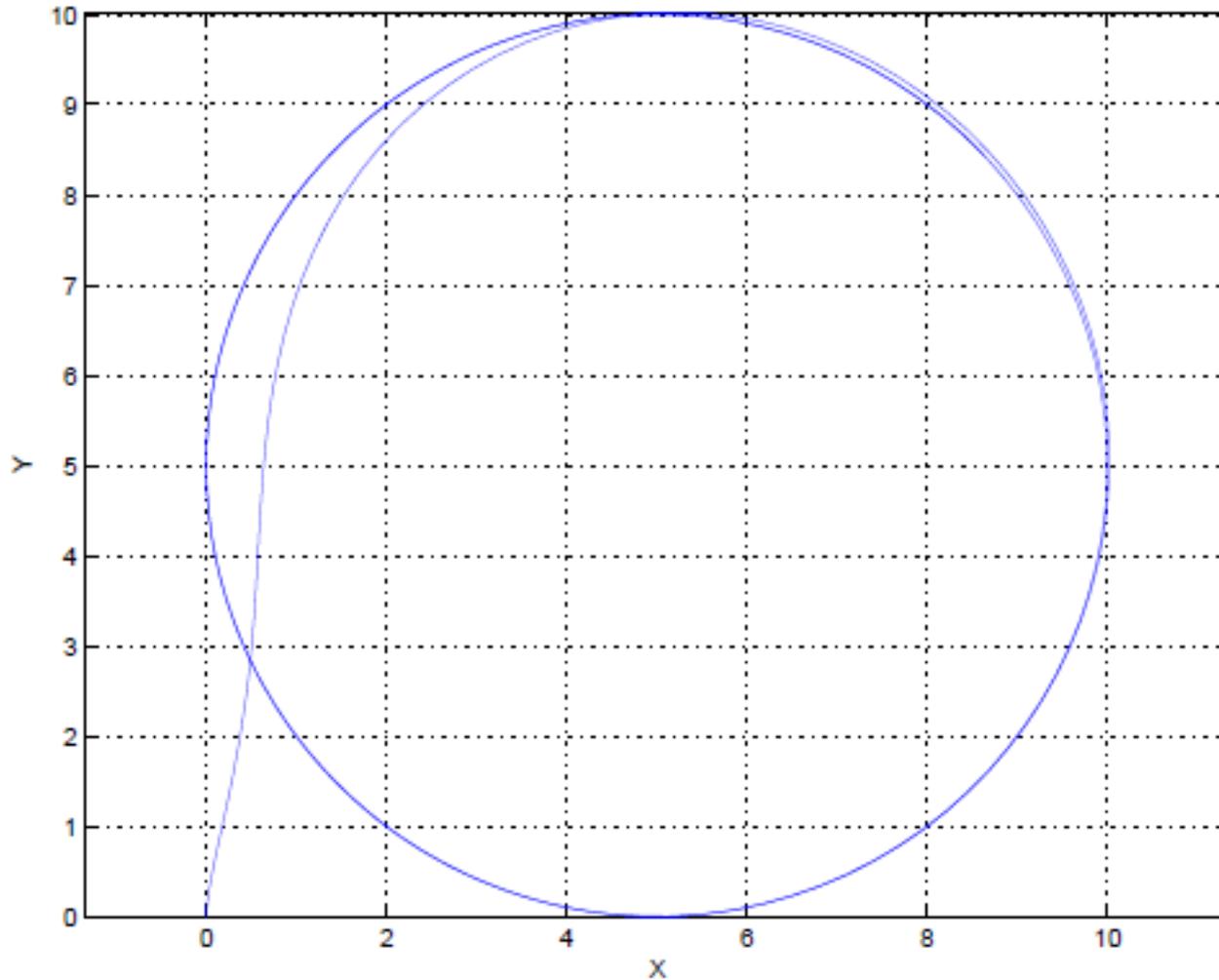
# Cours #12

## Contrôleur classique de vitesse et rotation (10)



# Cours #12

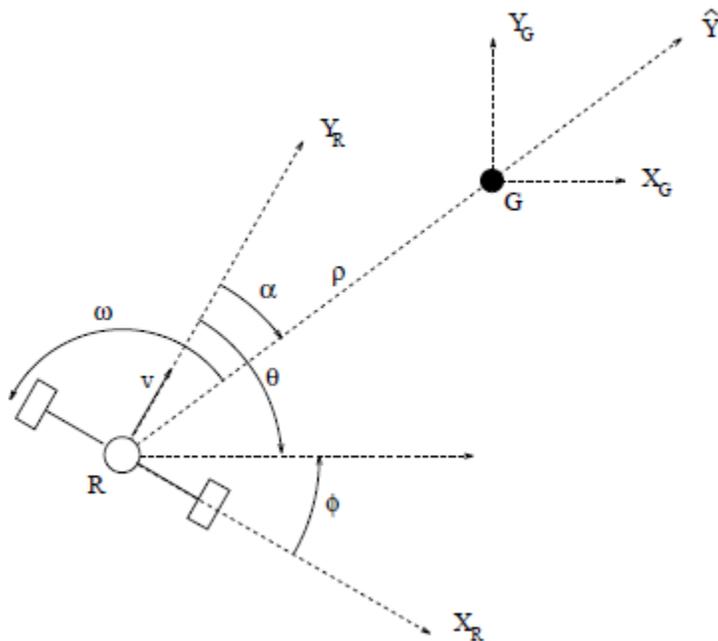
## Contrôleur classique de vitesse et rotation (11)



# 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:

$$\rho = \sqrt{x^2 + y^2}$$

$$\alpha = -\theta + \arctan\left(\frac{-y}{-x}\right) \bmod\left(\frac{\pi}{2}\right)$$

$$\phi = \frac{\pi}{2} - \theta$$

# 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:

$$\dot{\rho} = -\cos(\alpha)v$$

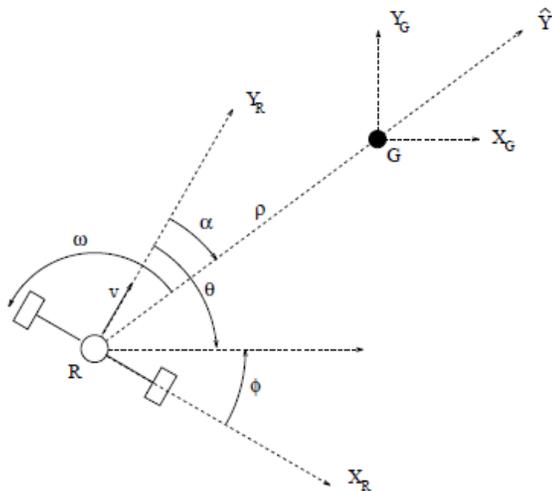
$$\dot{\alpha} = \frac{\sin \alpha}{\rho}v - \omega$$

$$\dot{\phi} = -\omega$$

- 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:

$$\dot{\rho} = -k_p \rho \cos(\alpha)$$

$$\dot{\alpha} = k_p \sin \alpha - k_\alpha \alpha - k_\phi \phi$$

$$\dot{\phi} = -k_\alpha \alpha - k_\phi \phi$$

# 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 appelée 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:

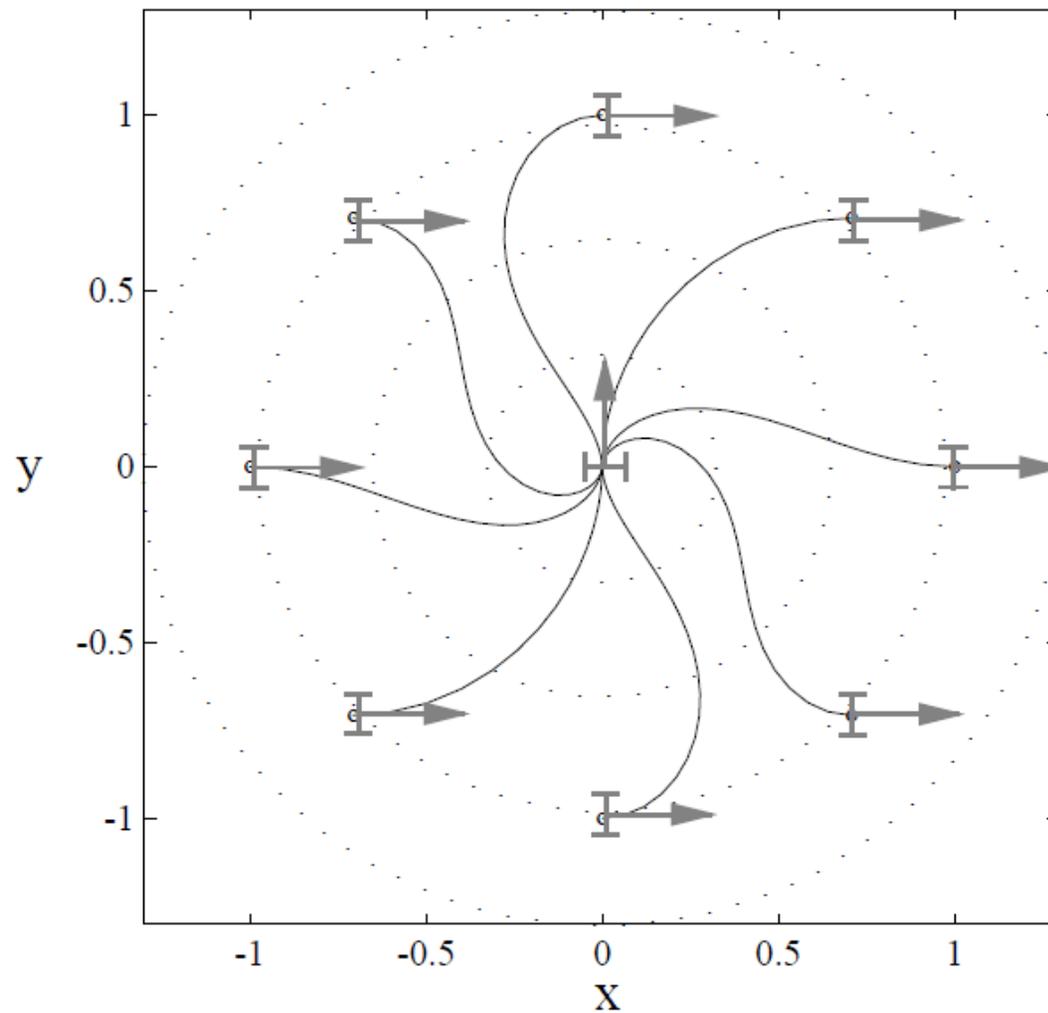
$$P(\lambda) = \det(\lambda I - A) = (\lambda + k_\rho)(\lambda^2 + \lambda(k_\alpha + k_\phi - k_\rho) - k_\rho k_\phi)$$

- ◆ 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

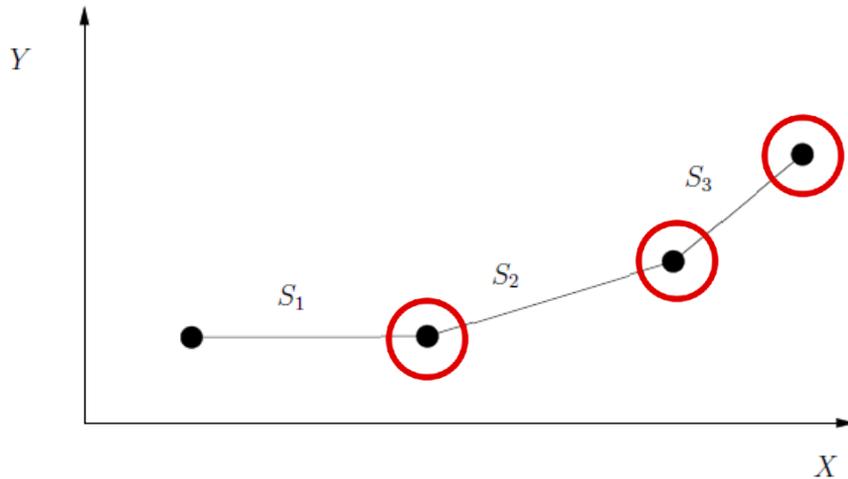
### Contrôleur d'Astolfi: un autre contrôleur qui lui ressemble (1)

- 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 décroît linéairement en fonction de la distance euclidienne:

$$v = k_p \rho$$

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

- 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 décroît/croît drastiquement/décroit/croît drastiquement, etc...
- Pour régler ce problème, il faut pousser plus loin notre étude du contrôleur d'Astolfi.**

## 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 connaît 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.  $\zeta \geq 1$ )

## 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:

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

- On a donc:

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

- En substituant pour trouver “zeta”:

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

- 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:

$$\begin{array}{l}
 v = k_\rho \rho \\
 \omega = k_\alpha \alpha + k_\phi \phi
 \end{array}
 \quad
 \begin{array}{l}
 k_\rho > 0 \\
 k_\phi < 0 \\
 k_\alpha + k_\phi - k_\rho > 0.
 \end{array}
 \quad
 \begin{array}{l}
 \text{critère supplémentaire:} \\
 \frac{k_\alpha + k_\phi - k_\rho}{2\sqrt{-k_\rho k_\phi}} \geq 1
 \end{array}$$

← 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  $K_p$  élevé approprié, par exemple, pour une vitesse maximale en pratique de 1,5m/s si on choisi  $K_p=3$ :

$$v = k_\rho \rho$$

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

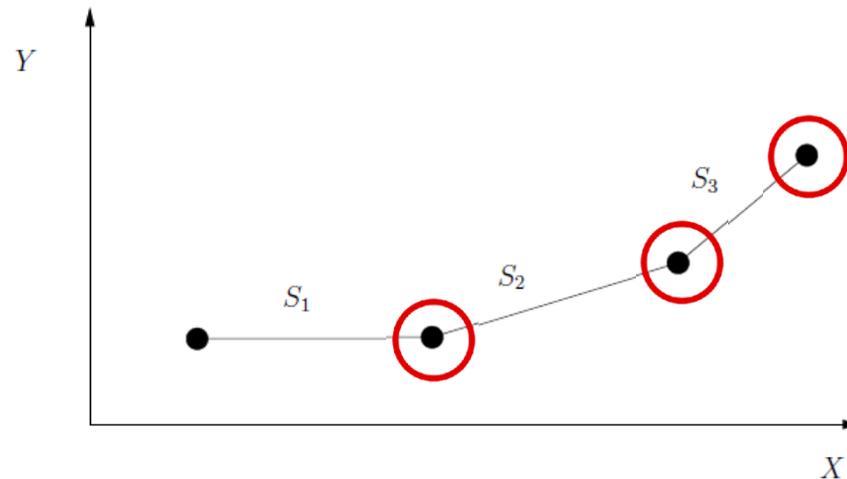
- En gros, vous pouvez choisir le gain de cette façon:

$$97 \quad k_\rho = \frac{v_{\max}}{\rho_{\text{limite}}}$$

## 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?

### 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:

$$\begin{aligned}
 k_\rho &> 0 \\
 k_\phi &< 0 \\
 k_\alpha + k_\phi - k_\rho &> 0.
 \end{aligned}
 \quad \text{critère supplémentaire: } \frac{k_\alpha + k_\phi - k_\rho}{2\sqrt{-k_\rho k_\phi}} \geq 1$$

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

$$\begin{aligned}
 v &= k_\rho \rho - k_1 \alpha - k_2 \dot{\phi} \quad \text{avec } k_{1,2} > 0 \\
 \omega &= k_\alpha \alpha + k_\phi \dot{\phi}
 \end{aligned}$$

- 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.

# 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., Everett, 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, Springer-Verlag, 1990.
- [4] Siegwart, R. Nourbakhsh, I., *Introduction to Autonomous Mobile Robots*. MIT Press, 2004.