

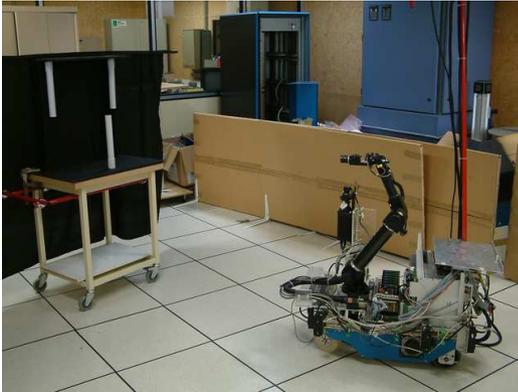
# Navigation autonome des robots mobiles: des problèmes de modélisation, perception et contrôle

Patrick Rives

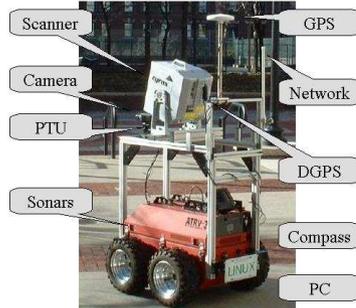
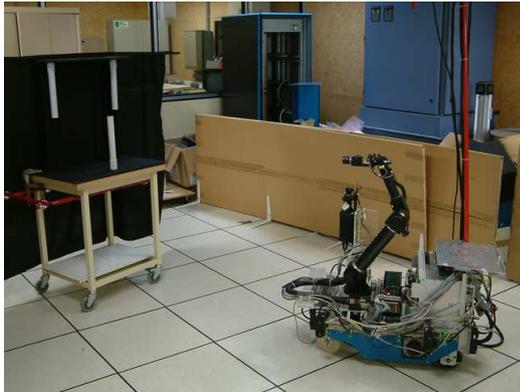
INRIA Projet ICARE  
2004 route des Lucioles -BP 93  
06902 Sophia Antipolis cedex  
e-mail : [Patrick.Rives@sophia.inria.fr](mailto:Patrick.Rives@sophia.inria.fr)

# C'est quoi aujourd'hui un robot mobile?

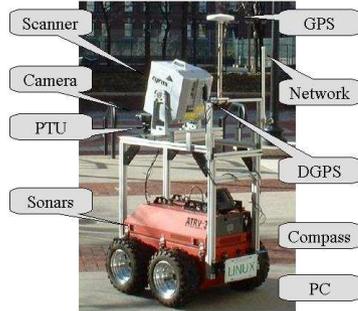
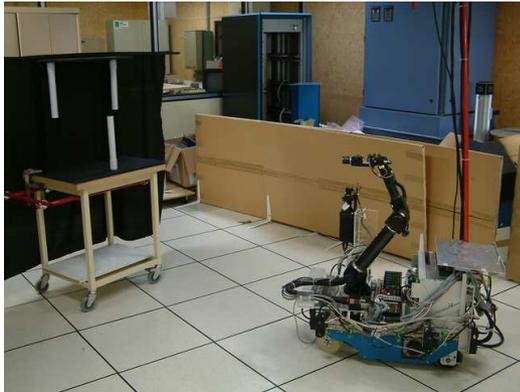
# C'est quoi aujourd'hui un robot mobile?



# C'est quoi aujourd'hui un robot mobile?



# C'est quoi aujourd'hui un robot mobile?



# Navigation autonome des robots mobiles

- Un domaine de recherche très actif depuis 1995 :
  - le problème du *passage à l'échelle*,
  - de nouvelles problématiques (par ex., traiter des environnements évolutifs, coopération de robots)

# Navigation autonome des robots mobiles

- Un domaine de recherche très actif depuis 1995 :
  - le problème du *passage à l'échelle*,
  - de nouvelles problématiques (par ex., traiter des environnements évolutifs, coopération de robots)
- Des enjeux économiques importants :
  - la réalisation de base de données géoréférencées (urbanisme, IGN, Mappy....),
  - la robotique autonome d'extérieur (militaire, urbaine, aérienne),
  - la surveillance environnementale et l'intervention après catastrophe
  - La robotique d'aide aux handicapés ( chaise roulante, assistance aux mal-voyants...)

# Vue d'ensemble

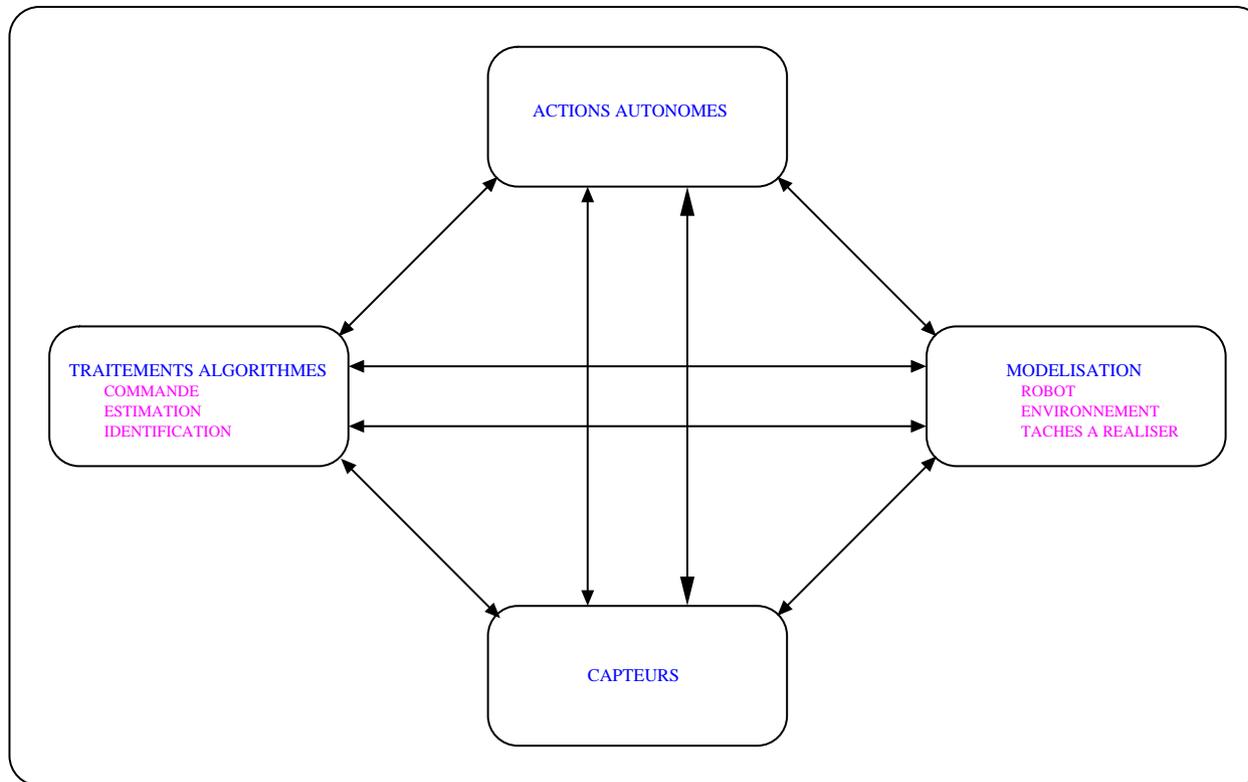
Les problèmes-clés de la robotique mobile autonome :

- Modéliser et contrôler les robots mobiles,
- Percevoir et interagir avec son environnement local,
- Explorer et représenter son environnement global.

# Vue d'ensemble

Les problèmes-clés de la robotique mobile autonome :

- Modéliser et contrôler les robots mobiles,
- Percevoir et interagir avec son environnement local,
- Explorer et représenter son environnement global.



# Plan de l'exposé

- **Modéliser et contrôler les robots mobiles,**
- Percevoir et interagir avec son environnement local,
- Explorer et représenter son environnement global.

# Introduction

- Un modèle générique de robot

$$\dot{x} = f(x, u) \quad , \quad f(0, 0) = 0, \quad f \text{ smooth}$$

localement **commandable** at  $(x, u) = (0, 0)$ .

# Introduction

- Un modèle générique de robot

$$\dot{x} = f(x, u) \quad , \quad f(0, 0) = 0, \quad f \text{ smooth}$$

localement **commandable** at  $(x, u) = (0, 0)$ .

- Une tentative de classification....
  - **Systèmes holonomes**: (robots manipulateurs, certains robots mobiles d'intérieur,...)
  - **Systèmes critiques**:
    - **Systèmes non holonomes** (robots mobiles, voitures,...):  
⇒ sans dérive
    - **Systèmes sous actionnés** (glisseur, dirigeable,...):  
⇒ avec dérive

# Quels problèmes cherche t-on à résoudre?

# Quels problèmes cherche t-on à résoudre?

- Trajectoire de référence:

$$t \longmapsto x_r(t) \in M \quad t \in [0, T]$$

# Quels problèmes cherche t-on à résoudre?

- Trajectoire de référence:

$$t \longmapsto x_r(t) \in M \quad t \in [0, T]$$

- Problème de régulation: Déterminer  $u$  qui assure

# Quels problèmes cherche t-on à résoudre?

- Trajectoire de référence:

$$t \longmapsto x_r(t) \in M \quad t \in [0, T]$$

- **Problème de régulation:** Déterminer  $u$  qui assure
  - **Stabilité**

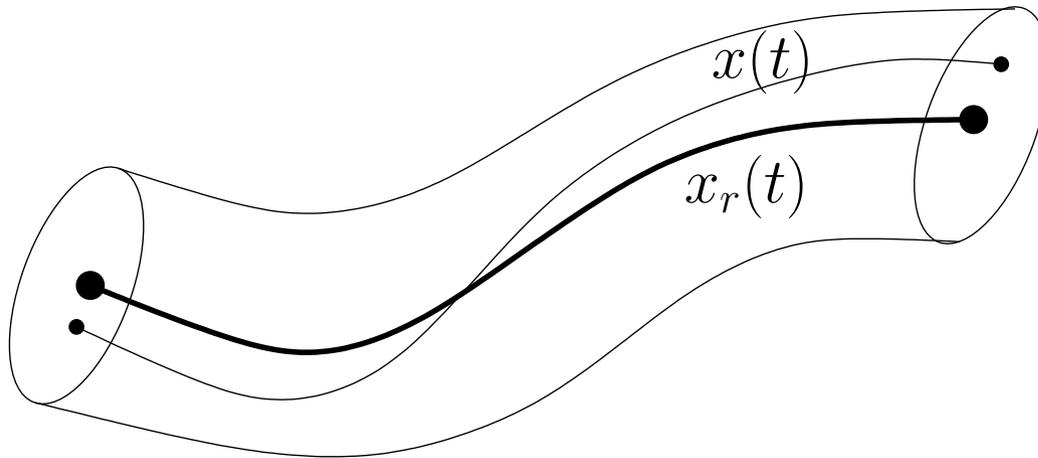
# Quels problèmes cherche t-on à résoudre?

- Trajectoire de référence:

$$t \longmapsto x_r(t) \in M \quad t \in [0, T]$$

- **Problème de régulation:** Déterminer  $u$  qui assure

- **Stabilité**



# Quels problèmes cherche t-on à résoudre?

- Trajectoire de référence:

$$t \longmapsto x_r(t) \in M \quad t \in [0, T]$$

- **Problème de régulation:** Déterminer  $u$  qui assure
  - Stabilité
  - Performance

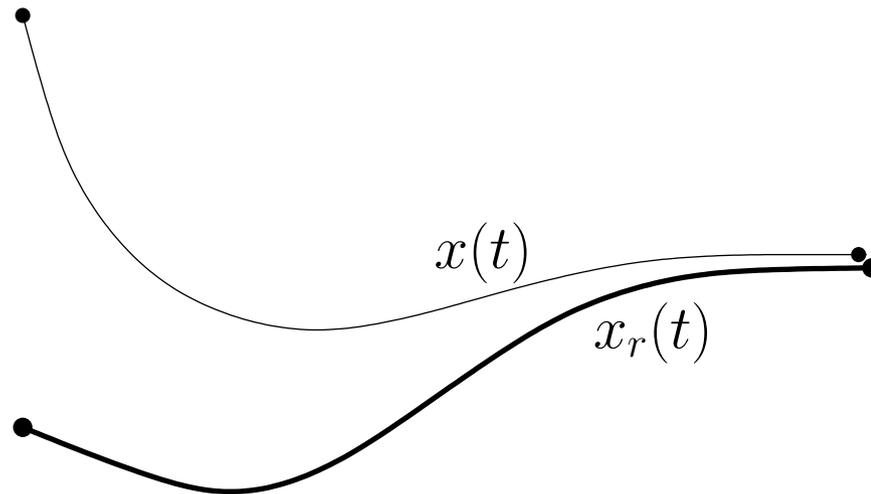
# Quels problèmes cherche t-on à résoudre?

- Trajectoire de référence:

$$t \longmapsto x_r(t) \in M \quad t \in [0, T]$$

- **Problème de régulation:** Déterminer  $u$  qui assure

- Stabilité
- Performance



# Quels problèmes cherche t-on à résoudre?

- Trajectoire de référence:

$$t \longmapsto x_r(t) \in M \quad t \in [0, T]$$

- **Problème de régulation:** Déterminer  $u$  qui assure

- Stabilité
- Performance
- Robustesse:  $\dot{x} = \hat{f}(x, u)$

# Quels problèmes cherche t-on à résoudre?

- Trajectoire de référence:

$$t \longmapsto x_r(t) \in M \quad t \in [0, T]$$

- **Problème de régulation:** Déterminer  $u$  qui assure

- Stabilité

- Performance

- Robustesse:  $\dot{x} = \hat{f}(x, u)$

- Quelques résultats:

# Quels problèmes cherche t-on à résoudre?

- Trajectoire de référence:

$$t \longmapsto x_r(t) \in M \quad t \in [0, T]$$

- **Problème de régulation:** Déterminer  $u$  qui assure

- Stabilité
- Performance
- Robustesse:  $\dot{x} = \hat{f}(x, u)$

- **Quelques résultats:**

- il **n'existe pas** de feedback  $u(x, x_r, u_r)$  qui stab. asymp. **toute** trajectoire  $x_r$  (Brockett 83)

# Quels problèmes cherche t-on à résoudre?

- Trajectoire de référence:

$$t \longmapsto x_r(t) \in M \quad t \in [0, T]$$

- **Problème de régulation:** Déterminer  $u$  qui assure

- Stabilité
- Performance
- Robustesse:  $\dot{x} = \hat{f}(x, u)$

- **Quelques résultats:**

- il **n'existe pas** de feedback  $u(x, x_r, u_r)$  qui stab. asymp. **toute** trajectoire  $x_r$  (Brockett 83)
- il **n'existe pas** de feedback  $u(x, x_r, u_r, t)$  qui stab. asymp. **toute** trajectoire  $x_r$  (Lizarraga 04)

# Quels problèmes cherche t-on à résoudre?

- Trajectoire de référence:

$$t \longmapsto x_r(t) \in M \quad t \in [0, T]$$

- **Problème de régulation:** Déterminer  $u$  qui assure

- **Stabilité**
- **Performance**
- **Robustesse:**  $\dot{x} = \hat{f}(x, u)$

- **Quelques résultats:**

- il **n'existe pas** de feedback  $u(x, x_r, u_r)$  qui stab. asymp. **toute trajectoire**  $x_r$  (Brockett 83)
- il **n'existe pas** de feedback  $u(x, x_r, u_r, t)$  qui stab. asymp. **toute trajectoire**  $x_r$  (Lizarraga 04)
- L'objectif de stab. **asymptotique** n'est pas atteignable en général!

# Quels problèmes cherche t-on à résoudre?

- Trajectoire de référence:

$$t \longmapsto x_r(t) \in M \quad t \in [0, T]$$

- **Problème de régulation:** Déterminer  $u$  qui assure

- **Stabilité**
- **Performance**
- **Robustesse:**  $\dot{x} = \hat{f}(x, u)$

- **Quelques résultats:**

- il **n'existe pas** de feedback  $u(x, x_r, u_r)$  qui stab. asymp. **toute** trajectoire  $x_r$  (Brockett 83)
- il **n'existe pas** de feedback  $u(x, x_r, u_r, t)$  qui stab. asymp. **toute** trajectoire  $x_r$  (Lizarraga 04)
- L'objectif de stab. **asymptotique** n'est pas atteignable en général!

⇒ **Une nouvelle approche :** La stabilisation pratique par fonctions transverses

# L'approche par fonction transverse

Plutôt que :

*stabilisation asymptotique d'un point d'équilibre*

# L'approche par fonction transverse

Plutôt que :

*stabilisation asymptotique d'un point d'équilibre*

considérer :

*stabilisation asymptotique d'un ensemble dans le voisinage de ce point*

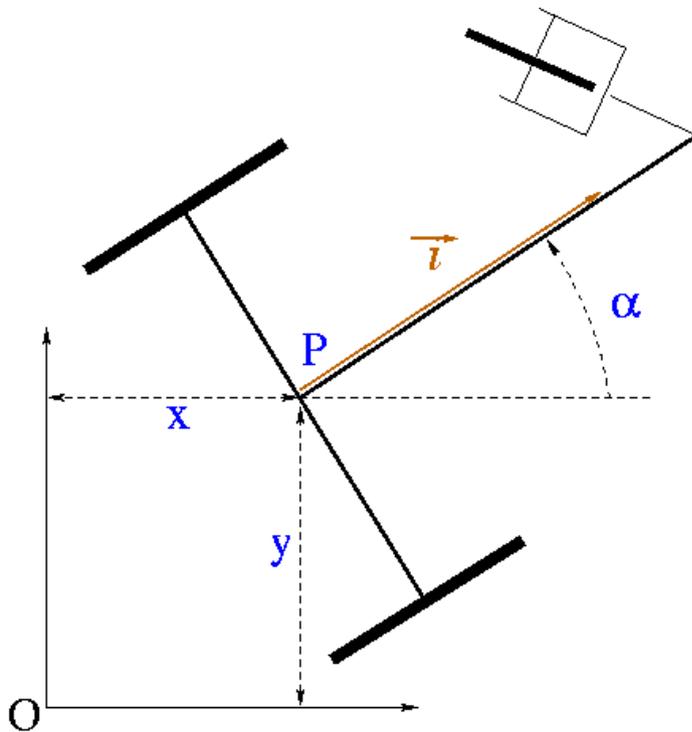
+

*réjection de perturbations additives*

(objectif de *stabilisation pratique*)

# Application aux systèmes non-holonomes (1)

Véhicule de type "unicycle"



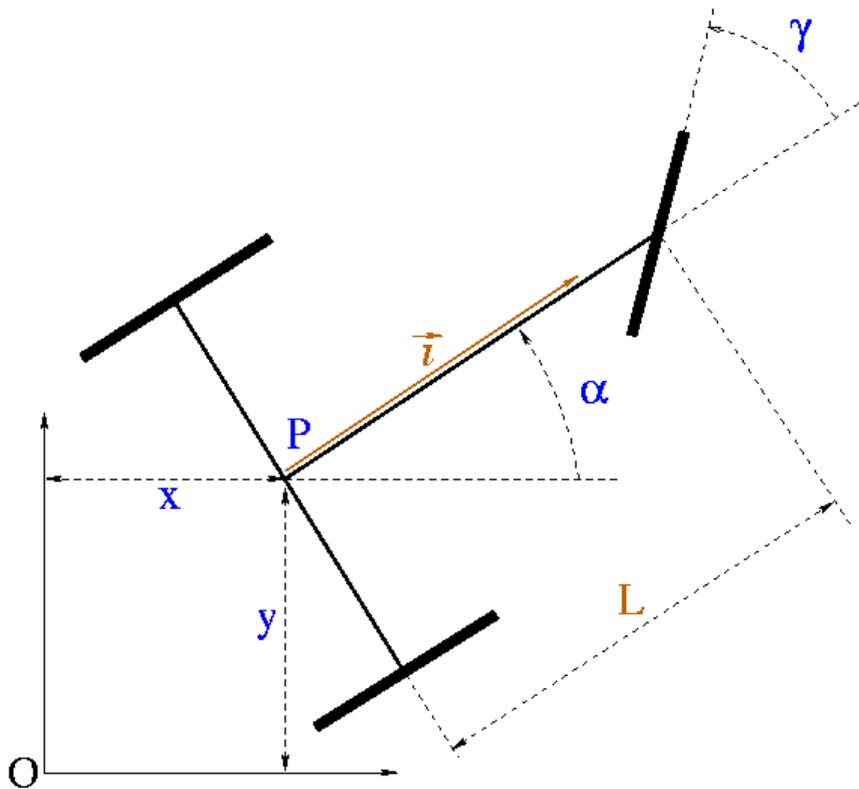
Etat :  $(x, y, \alpha)$

Entrée de commande :  $v_P, \dot{\alpha}$



# Application aux systèmes non-holonomes (2)

Véhicule de type "voiture"

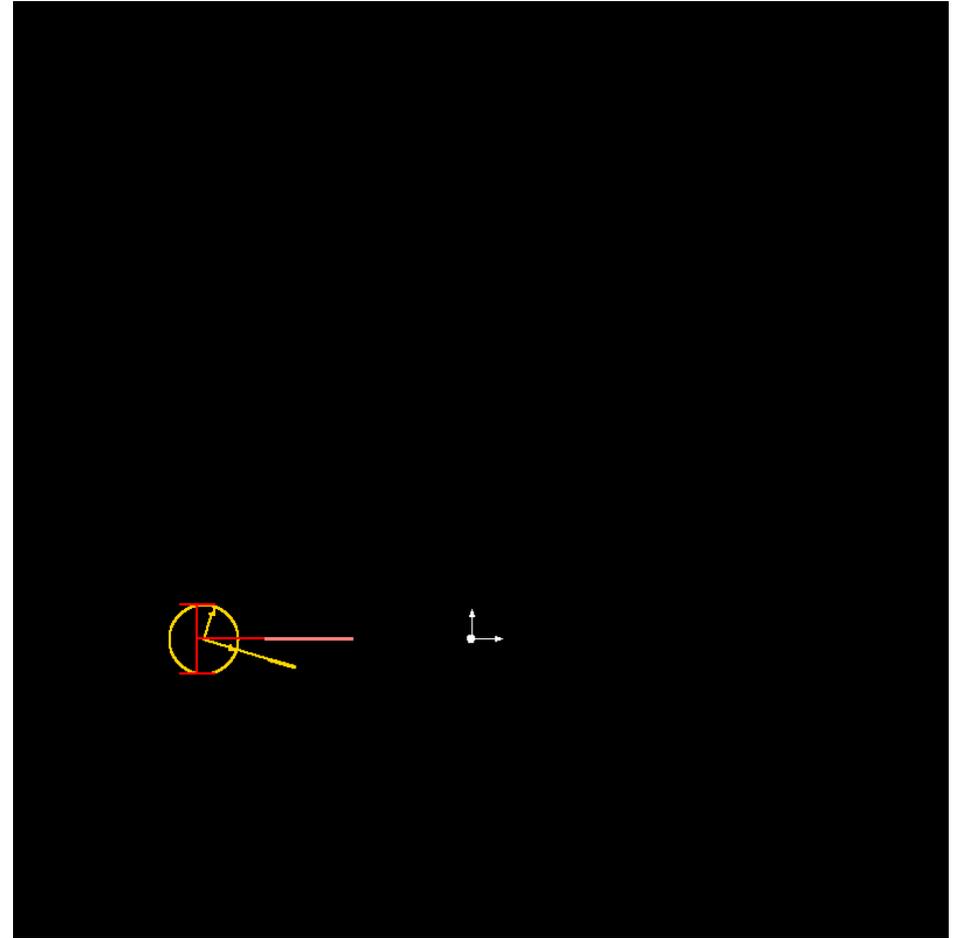
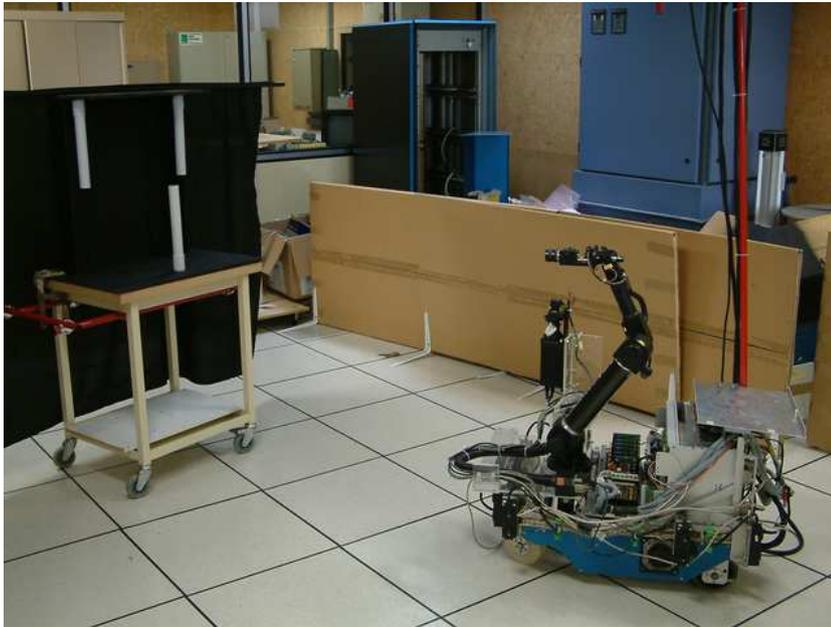


Etat :  $(x, y, \alpha, \gamma)$

Entrée de commande :  $v_P, \dot{\gamma}$

# Application aux systèmes non-holonomes (4)

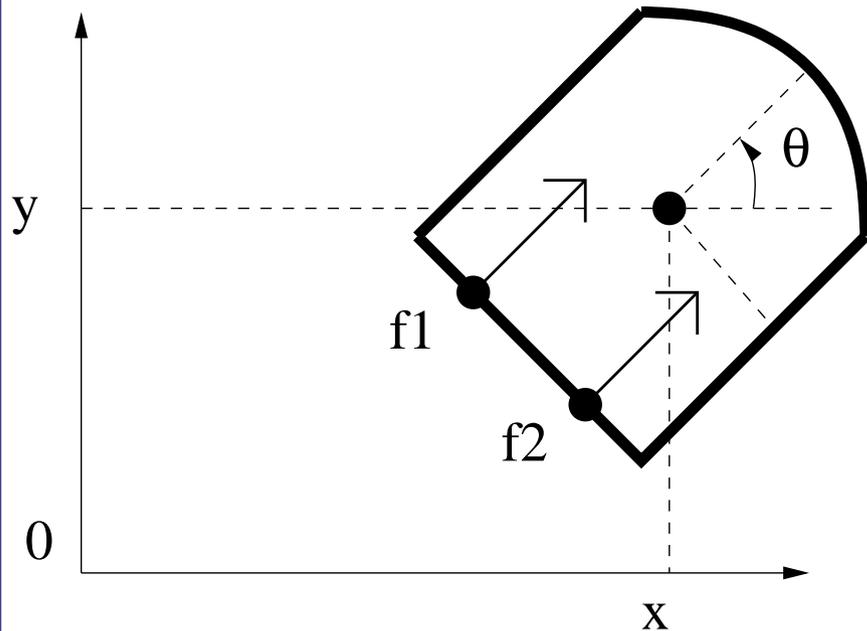
## Contrôle coordonné d'un manipulateur mobile



(\*) Thèse de M. Fruchard

# Extension aux systèmes sous-actionnés (1)

Exemple: le glisseur



Etat :  $(x, y, \theta, v_1, v_2, \omega)$



Entrée de commande :  $f_1, f_2$

# Extension aux systèmes sous-actionnés (2)

Exemple: le dirigeable



Vu comme un système non holonome



Vu comme un glisseur

# Plan de l'exposé

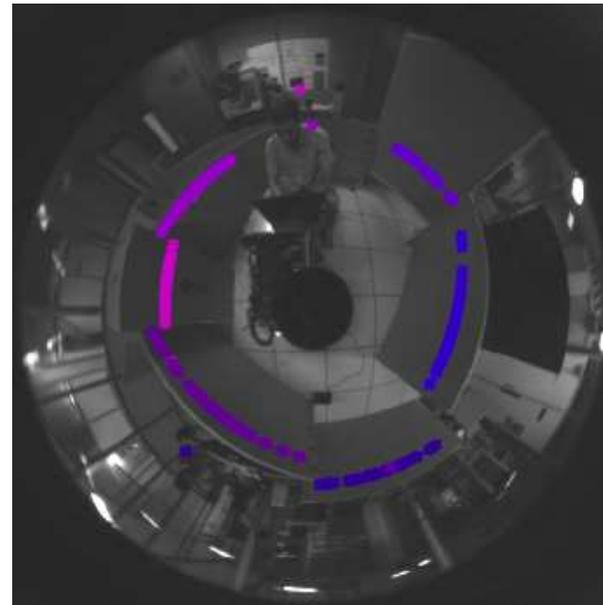
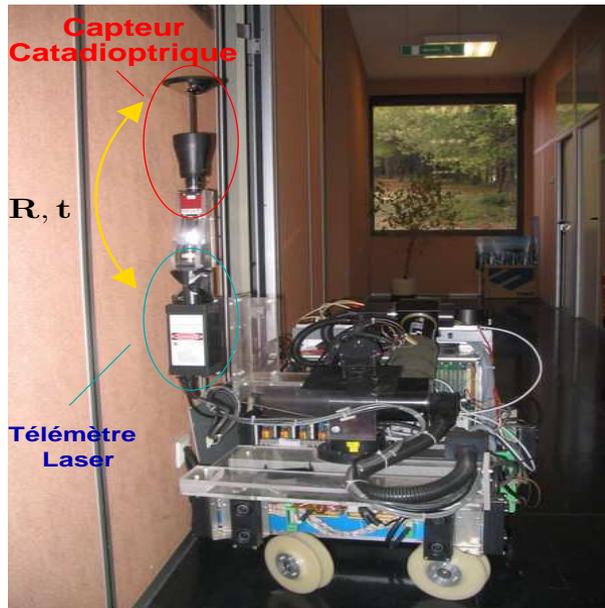
- Modéliser et contrôler les robots mobiles,
- **Percevoir et interagir avec son environnement local,**
- Explorer et représenter son environnement global.

# Percevoir son environnement (1/2)

## Les capteurs et les traitements...

- modélisation et calibration des capteurs (vision, vision omni, télémétrie, INS, odométrie, GPS...)
- techniques de fusion (lache, serrée, probabiliste, “voting”...)
- algorithmes temps-réel pour la commande
- algorithmes “robustes” aux erreurs de modélisation

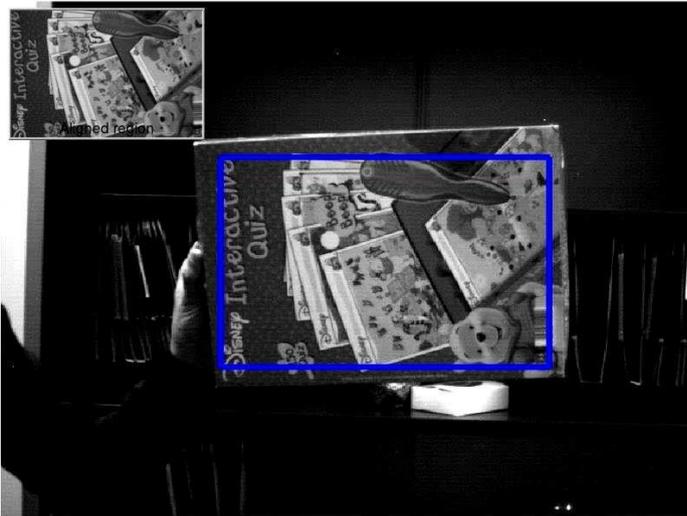
⇒ vers de nouveaux capteurs “intelligents”...



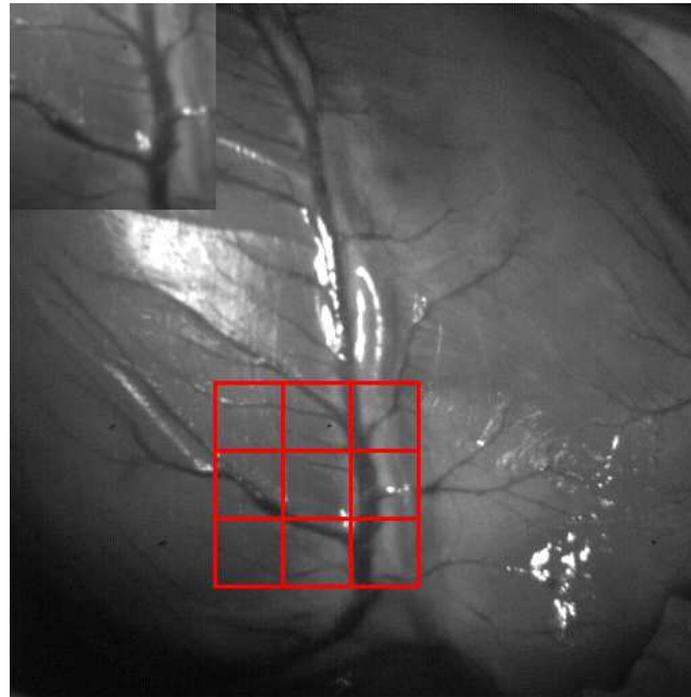
# Percevoir son environnement 2/2

## La complexité des scènes:

- objets déformables et / ou articulés
- changement de géométrie, d'éclairage et ombres

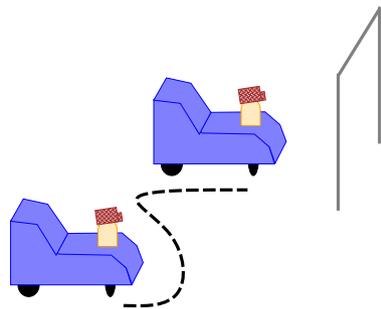
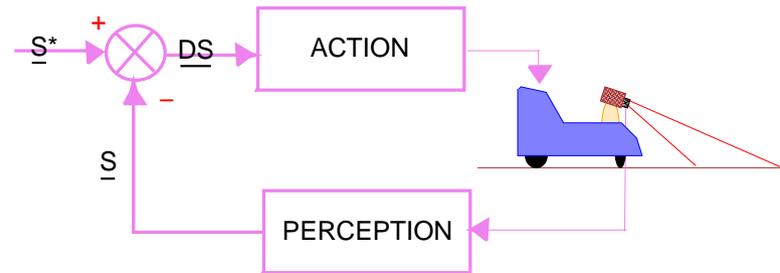
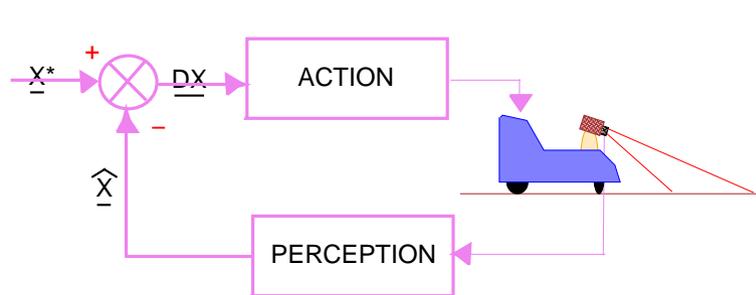


Changement d'illumination



objet déformable

# Se déplacer par rapport à son environnement

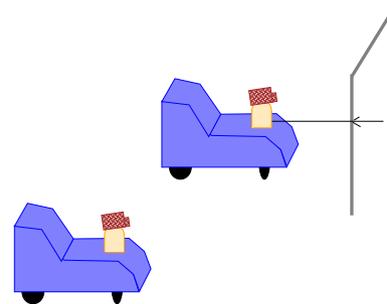


LOOK-AND-MOVE

3D ROBOT'S LOCALIZATION

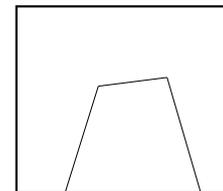
PATH PLANNING

PATH FOLLOWING

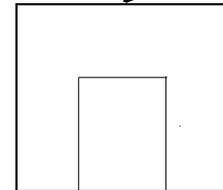


VISUAL SERVOING

IMAGE(t0)



IMAGE(tf)



# Positionnement par asservissement visuel

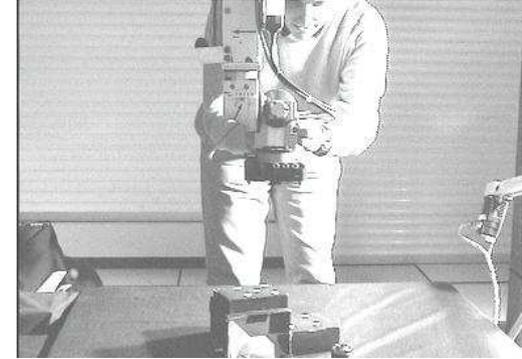
## Asservissement visuel invariant au modèle de caméra



Image de référence



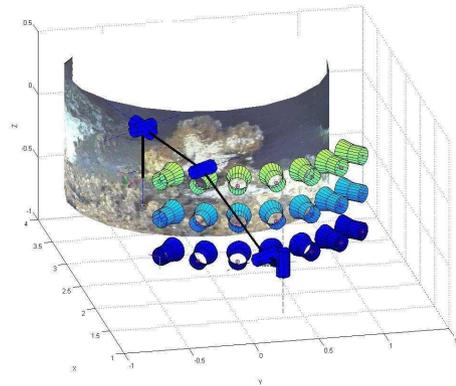
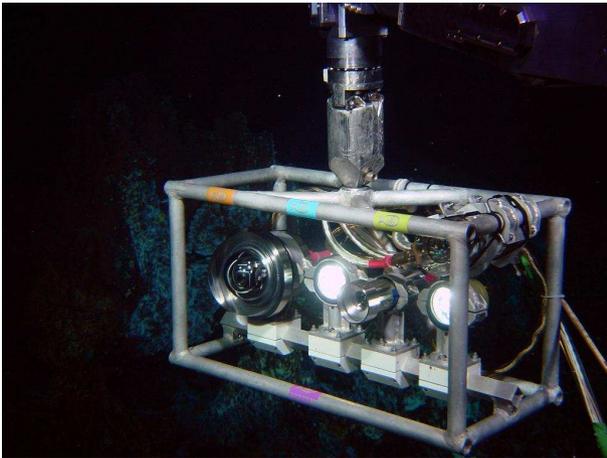
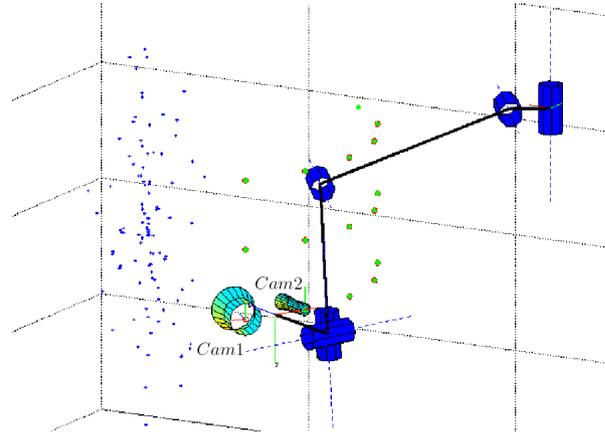
Image initiale



Vue du robot

- Apprentissage  $f = 12 \text{ mm}$
- Asservissement visuel  $f = 6 \text{ mm}$

# Application en robotique sous-marine.

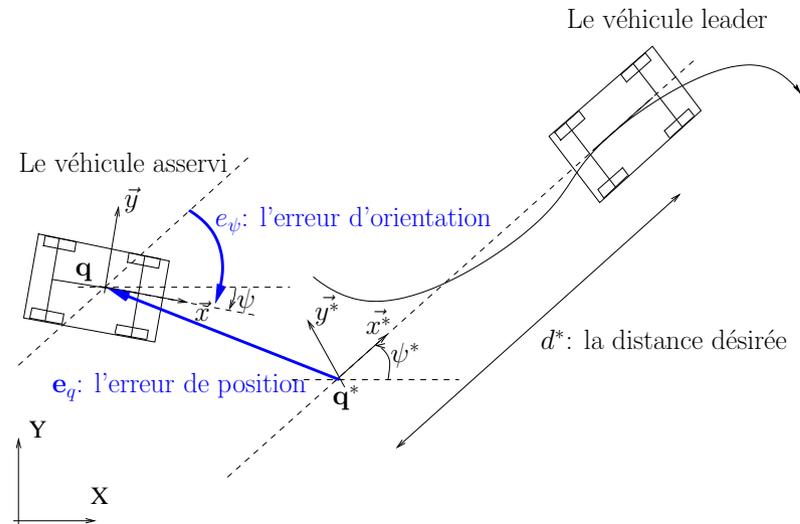


(\*) Travail de thèse de V. Brandou, projet IFREMER *THEMIS*

# Application au suivi de véhicules



Les deux CyCabs

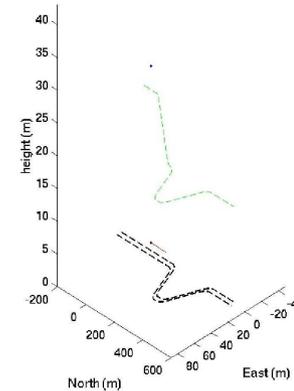
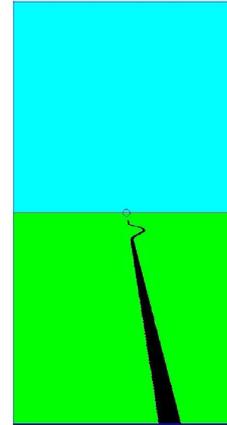


Objectif de commande

- Utilisation d'une tourelle Pan-and-tilt pour garder la cible dans l'image
- Reconstruction de la position du véhicule suiveur par rapport à la cible
- Asservissement sur le véhicule leader

(\*) Thèse de S. Benhimane, projet predict *MobiVIP*

# Application: asservissement visuel d'un dirigeable



- Dirigeable de 9m vectorisé, dynamique fortement non linéaire  
grande sensibilité aux perturbations
- Suivi de structures linéaires
  - Vitesse air désirée  $V_o = 8m/s$
  - Vent constant de  $V_w = 3m/s$  orientation 10deg Nord
  - Erreur initiale: Lateral 10m, Altitude 2m
  - Profil de descente de 25m à 15m

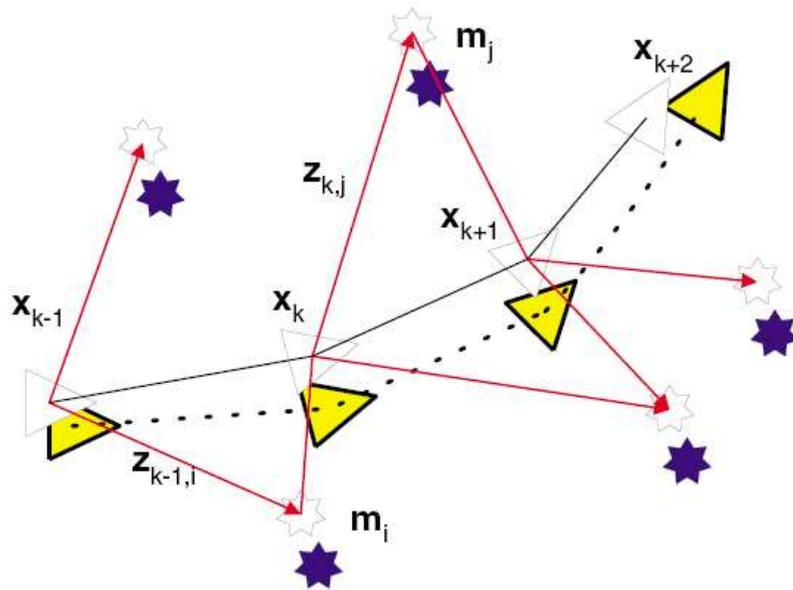
(\*) Collaboration avec le CenPRA (Campinas, Brazil) et IST (Lisbonne, Portugal)

# Plan de l'exposé

- Modéliser et contrôler les robots mobiles,
- Percevoir et interagir avec son environnement local,
- Explorer et représenter son environnement global.

# Un problème canonique : le SLAM (1)

**SLAM** (*Simultaneous Localization And Mapping*) : Comment, partant d'une position inconnue dans un environnement inconnu, reconstruire incrémentalement une carte de cet environnement et l'utiliser *simultanément* pour se localiser?



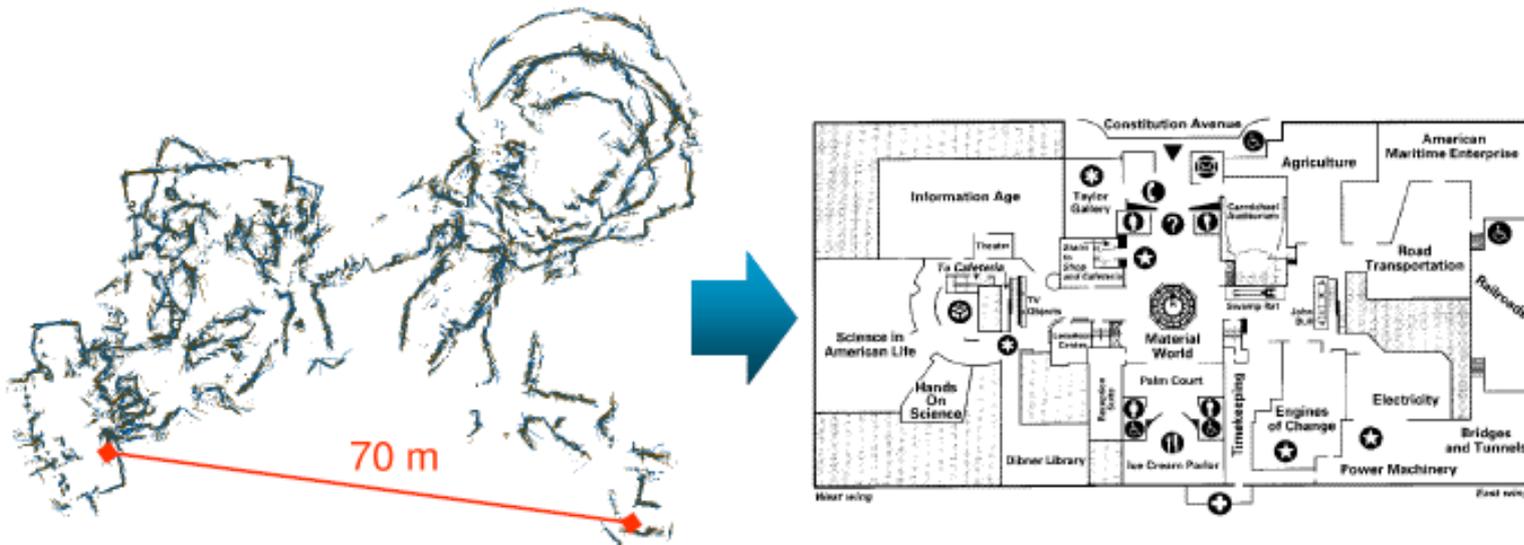
$$P(\mathbf{x}_k, \theta | \mathbf{Z}^k, \mathbf{U}^k) ?$$

avec :

$P(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k)$  le modèle d'évolution du robot,

$P(\mathbf{z}^k | \mathbf{x}_k, \theta)$  le modèle de perception

# Un problème canonique : le SLAM (2)



## Les difficultés intrinsèques au problème

- la corrélation entre position du robot et éléments de la carte  $\Rightarrow$  complexité en  $O(N^2)$ , où  $N$  est le nombre d'éléments de la carte.
- la non linéarité et la dérive des modèles d'évolution du véhicule,
- l'association des données (hypothèses sous-jacentes: scène statique, amers discernables, etc...).



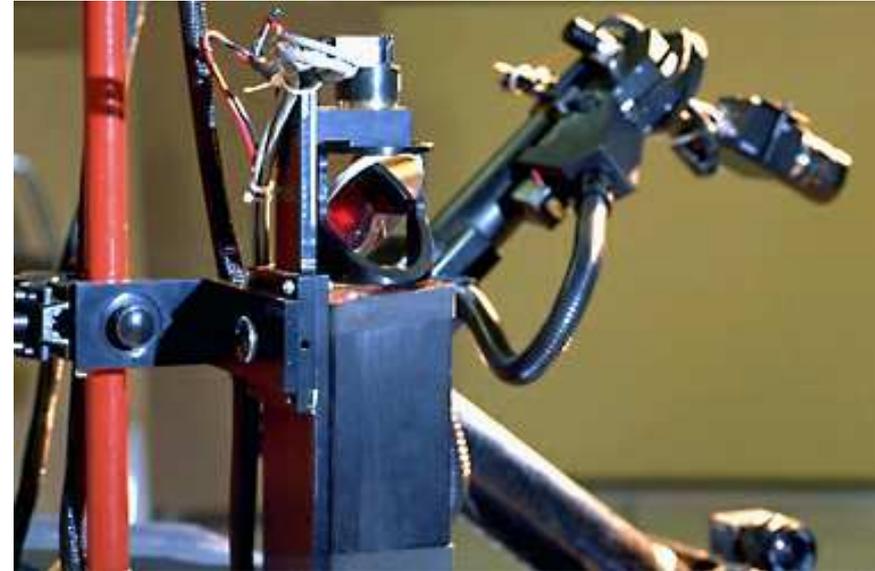
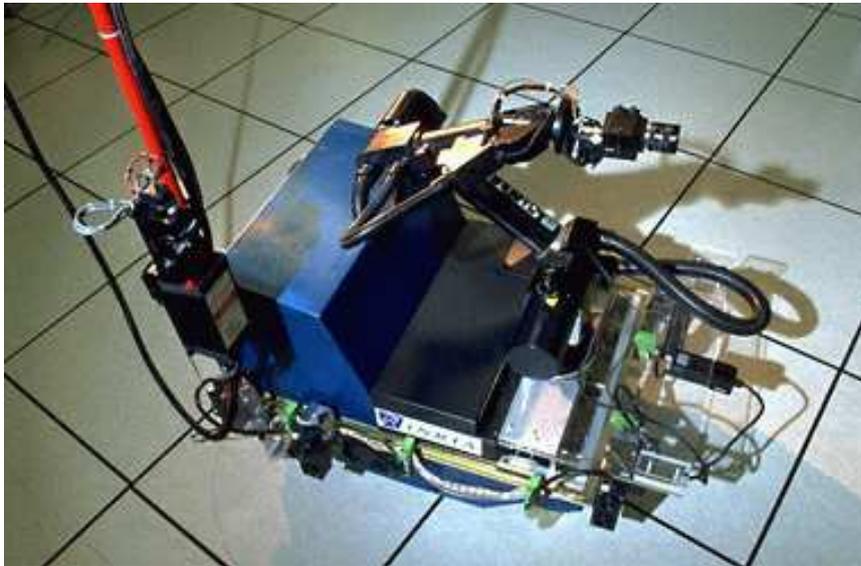
## SLAM dans des environnements d'intérieur

- ⇒ Navigation réactive et complétude des explorations
- ⇒ Compensation des dérives et construction des représentations globales

- SLAM dans des environnements d'intérieur
  - ⇒ Navigation réactive et complétude des explorations
  - ⇒ Compensation des dérives et construction des représentations globales
- SLAM dans des environnements d'extérieur
  - ⇒ Slam visuel monoculaire
  - ⇒ Slam visuel stéréo dans des scènes dynamiques

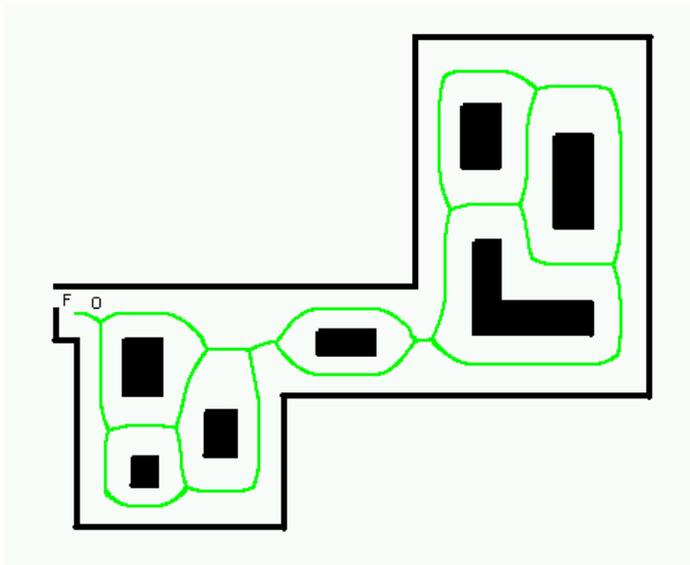
# Navigation réactive en environnement d'intérieur 1/3

**Objectif :** Explorer et cartographier un environnement d'intérieur **inconnu** en garantissant une navigation sûre.



## Navigation réactive en environnement d'intérieur 2/3

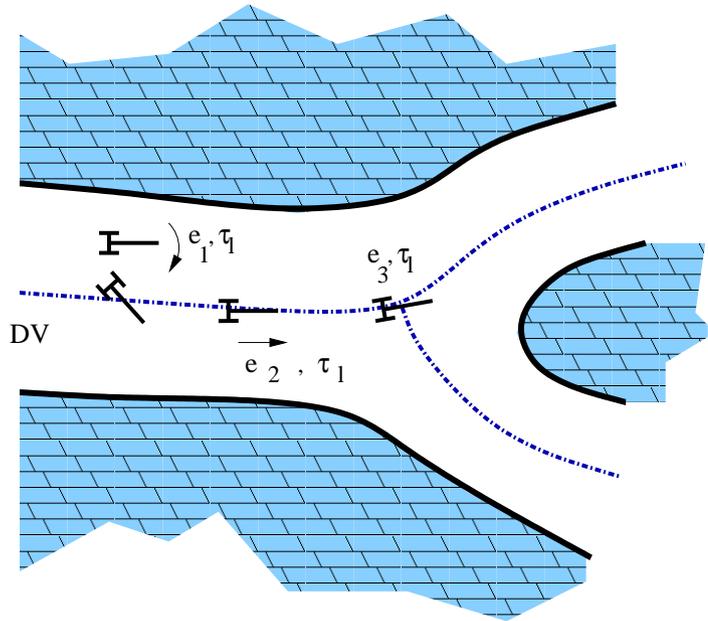
**Stratégie de navigation :** Un télémètre laser embarqué fournit à la cadence de 40ms une carte correspondant à un plan de coupe de l'environnement local. Cette carte est utilisée dans la commande pour contraindre le robot à se déplacer sur le **Diagramme de Voronoï** associé à l'environnement.



Le **Diagramme de Voronoï** est défini comme l'ensemble de points équidistant à au moins deux objets  $O_i$  and  $O_j$ , tel que chaque point de l'ensemble est plus proche de  $O_i$  et  $O_j$  que de tout autre objet  $O_k$  dans l'environnement,  $k \neq i, j$ .

# Navigation réactive en environnement d'intérieur 3/3

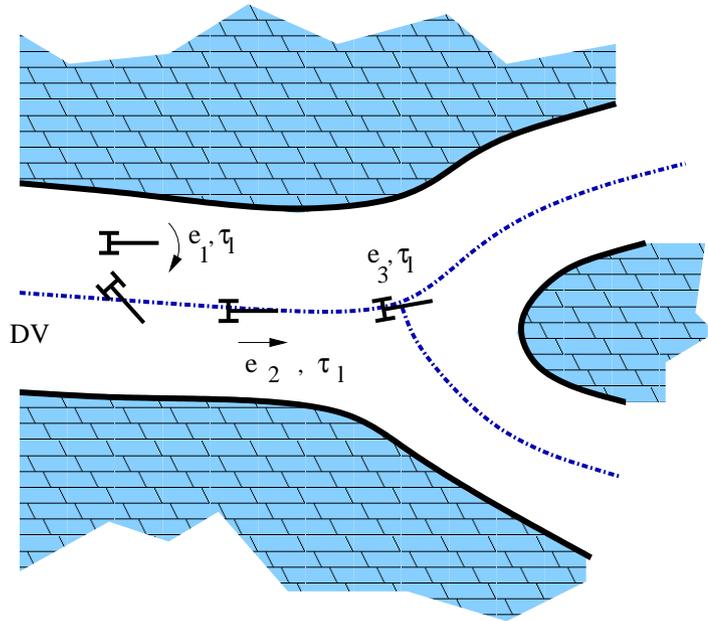
Fonctions de navigation :



→  $e_1$ , rejoindre la branche du Voronoï la plus proche,

# Navigation réactive en environnement d'intérieur 3/3

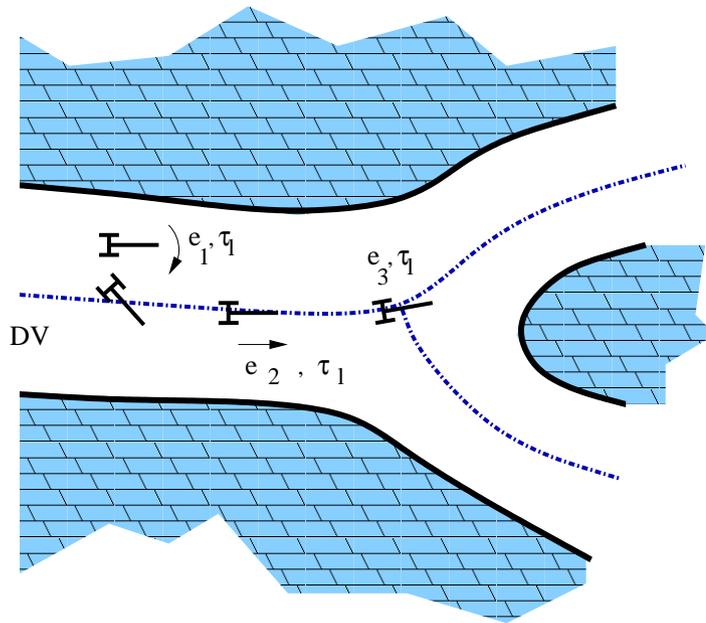
Fonctions de navigation :



- $e_1$ , rejoindre la branche du Voronoï la plus proche,
- $e_2$ , se déplacer le long de la branche du Voronoï

# Navigation réactive en environnement d'intérieur 3/3

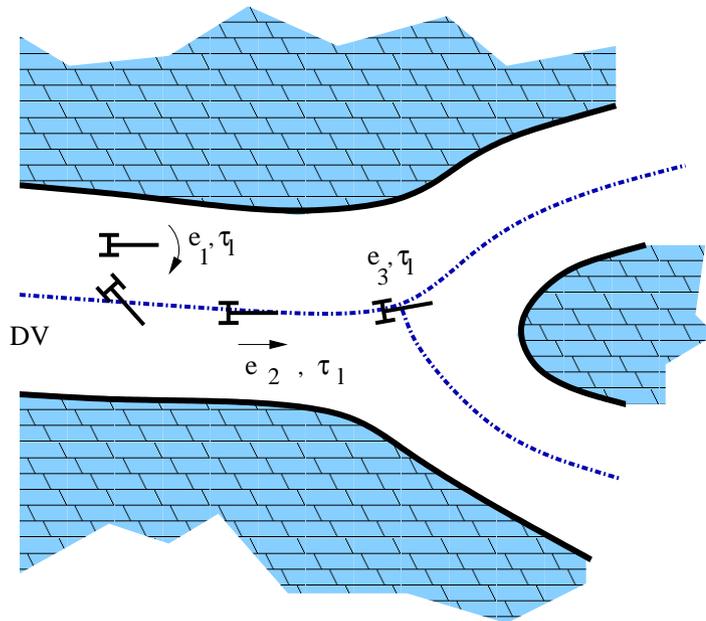
Fonctions de navigation :



- $e_1$ , rejoindre la branche du Voronoï la plus proche,
- $e_2$ , se déplacer le long de la branche du Voronoï
- $e_3$ , s'arrêter sur un point de bifurcation

# Navigation réactive en environnement d'intérieur 3/3

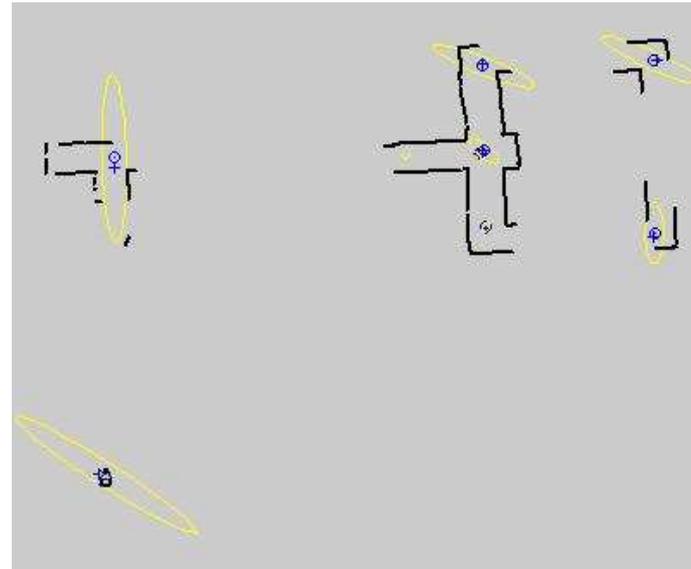
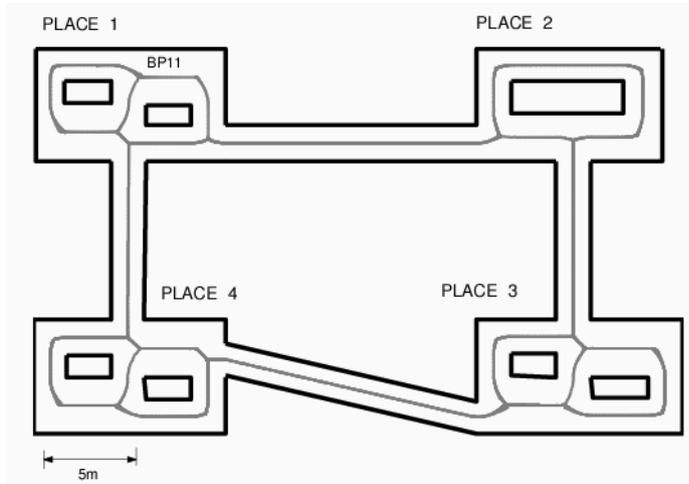
Fonctions de navigation :



- $e_1$ , rejoindre la branche du Voronoï la plus proche,
- $e_2$ , se déplacer le long de la branche du Voronoï
- $e_3$ , s'arrêter sur un point de bifurcation

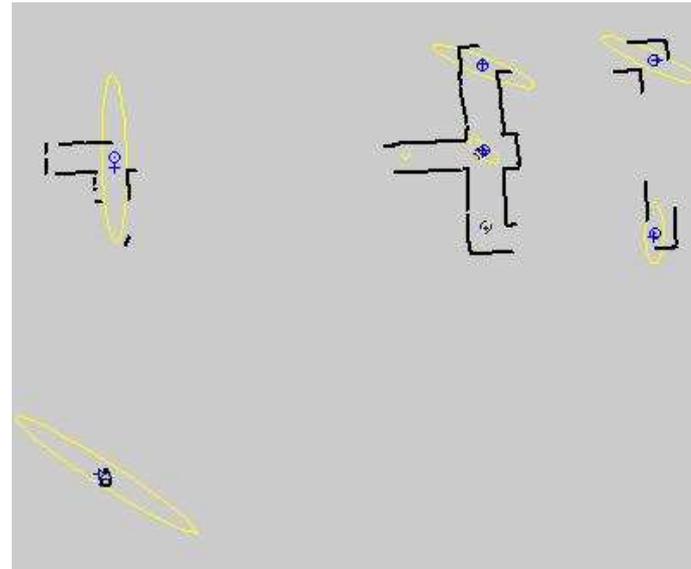
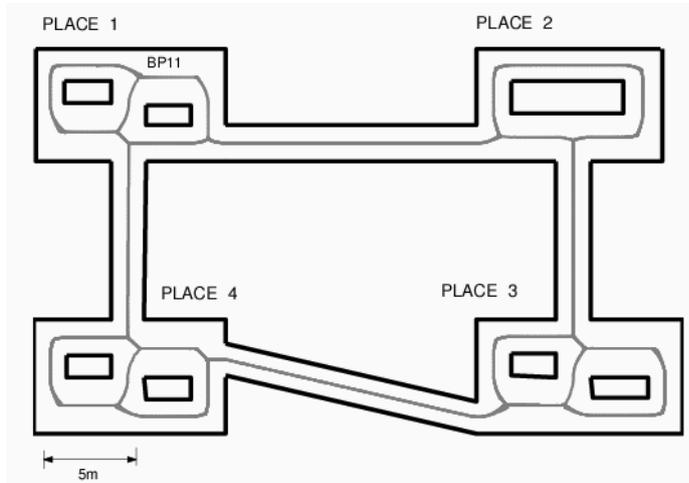
→ Les tâches  $e_1$ ,  $e_2$  et  $e_3$  sont implémentées par des tâches élémentaires de commande référencée capteur.

# Cartographie et compensation des dérives



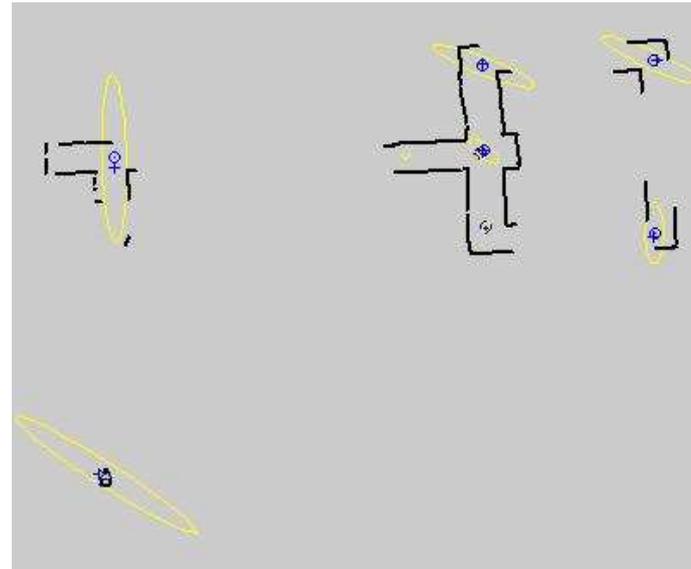
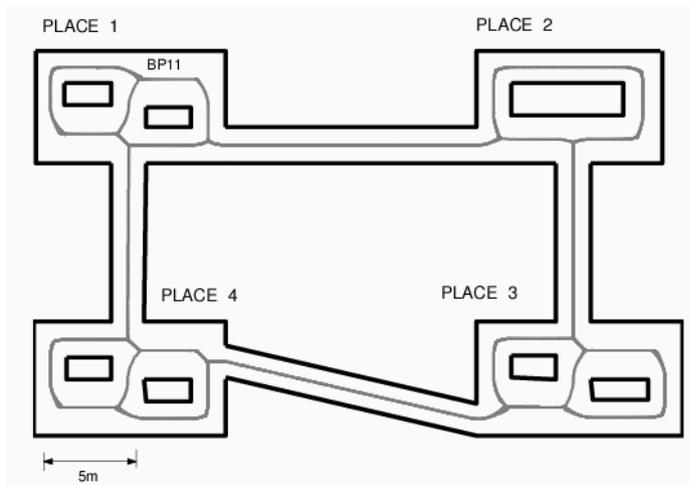
⇒ Structuration en lieux,

# Cartographie et compensation des dérives



- ⇒ Structuration en lieux,
- ⇒ Identification des lieux et fermeture de boucle,

# Cartographie et compensation des dérives



- ⇒ Structuration en lieux,
- ⇒ Identification des lieux et fermeture de boucle,
- ⇒ Optimisation sur un modèle semi-rigide contraint par l'observabilité

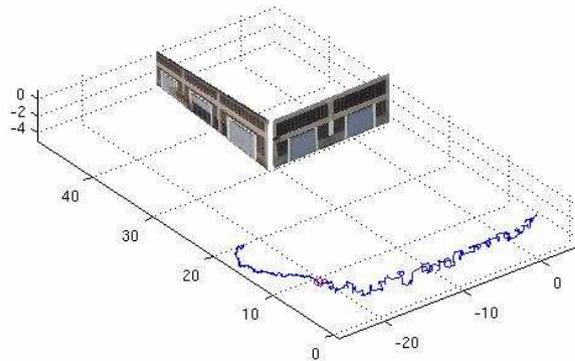
# SLAM monoculaire en environnement structuré

## Objectifs:

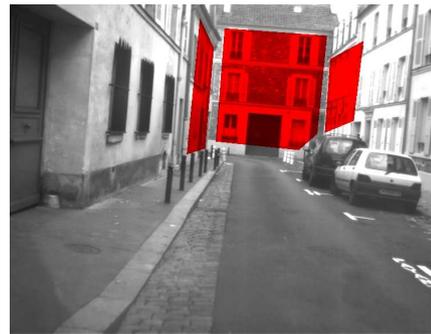
- Identifier et suivre les structures planes de l'environnement,
- Estimer la trajectoire de la caméra en 6D,
- Estimer les plans de la scène 3D,

## Difficultés:

- Une seule caméra,
- Robustesse aux changements d'illumination,
- Performances Temps réel



Détection de plan

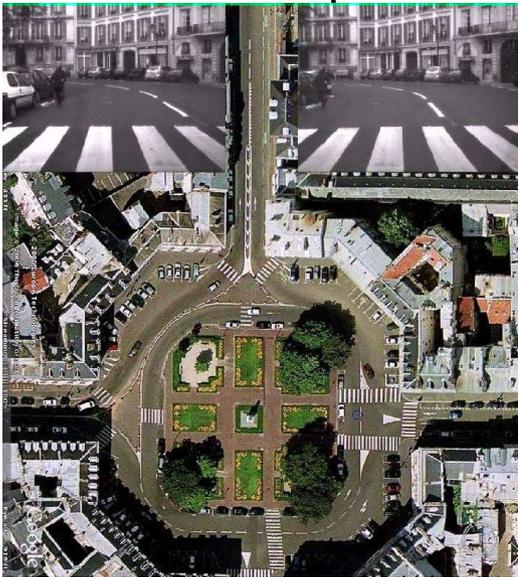


SLAM monoculaire

(\*) Thèse de G. Silveira

# SLAM stéréovision en environnement complexe et dynamique

- Objectifs:
  - Localisation précise et reconstruction dense sans modèle a priori,
  - Aide à la conduite et conduite automatique en environnement urbain.
- Difficultés:
  - Scènes très complexes et dynamiques,
  - Changement d'illumination et ombres portées,
  - Grand déplacement dans les images



Trajectographie par stéréo dense



Approche robuste

(\*) Postdoc de A. Comport

# Quelques perspectives...

- **Contrôler des systèmes complexes**
  - Synthèse de contrôleurs robustes aux perturbations en robotique aérienne,
  - Optimisation énergétique pour les missions de longue durée.
  - Contrôle de formation de robots.
- **Percevoir et interagir avec l'environnement**
  - Intégrer la connaissance a priori (modélisation, apprentissage...),
  - Prendre en compte les erreurs et les incertitudes (Bayésien, méthodes à erreur bornée, analyse par intervalle...).
- **Exploration et navigation autonome**
  - Manipuler de grande base de données (SIG, géoréférencement...),
  - Problème des initialisations (bootstrap) des méthodes itératives,
  - Une stratégie de coopération de robot pour l'exploration et le SLAM (filtrage particulière).

# Quelques perspectives...

- **Contrôler des systèmes complexes**
  - Synthèse de contrôleurs robustes aux perturbations en robotique aérienne,
  - Optimisation énergétique pour les missions de longue durée.
  - Contrôle de formation de robots.
- **Percevoir et interagir avec l'environnement**
  - Intégrer la connaissance a priori (modélisation, apprentissage...),
  - Prendre en compte les erreurs et les incertitudes (Bayésien, méthodes à erreur bornée, analyse par intervalle...).
- **Exploration et navigation autonome**
  - Manipuler de grande base de données (SIG, géoréférencement...),
  - Problème des initialisations (bootstrap) des méthodes itératives,
  - Une stratégie de coopération de robot pour l'exploration et le SLAM (filtrage particulière).