

Robotique Mobile

05 - Capteurs

David Filliat

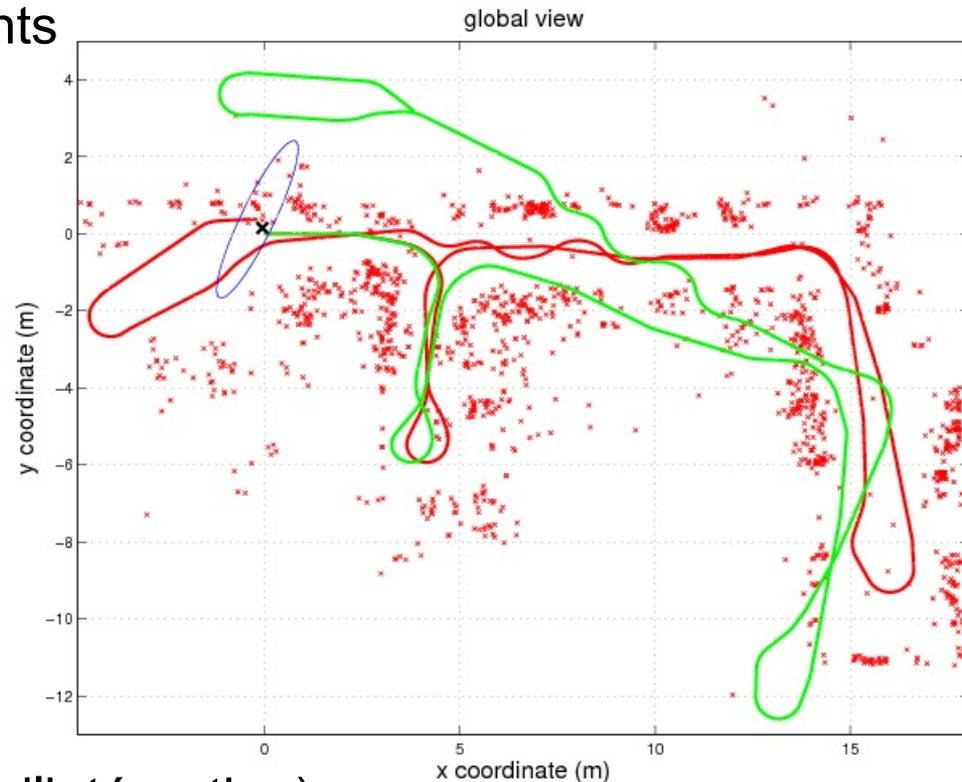
Alexandre Chapoutot

Goran Frehse

prenom.nom@ensta-paris.fr

Informations internes

- informations *proprioceptives* (ou idiothétiques)
- renseignent sur les déplacements



Ex : Odométrie, inertie

- **Erreur cumulative** (processus d'intégration)
- Inutilisable à long terme
- **Référence simple à utiliser, peu dépendante de la position**

Informations externes

- informations extéroceptives (ou allothétiques, *perceptions*)
- renseignement sur la position

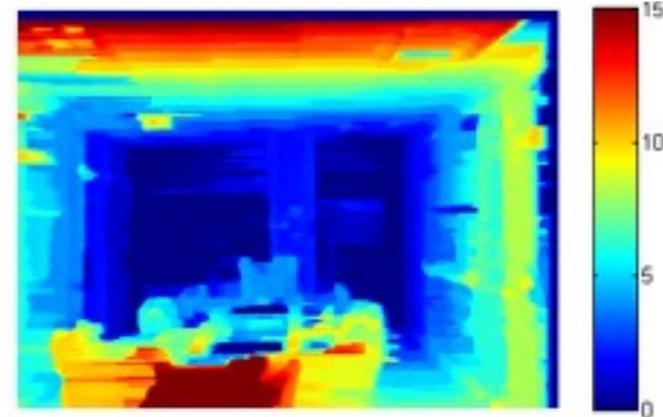
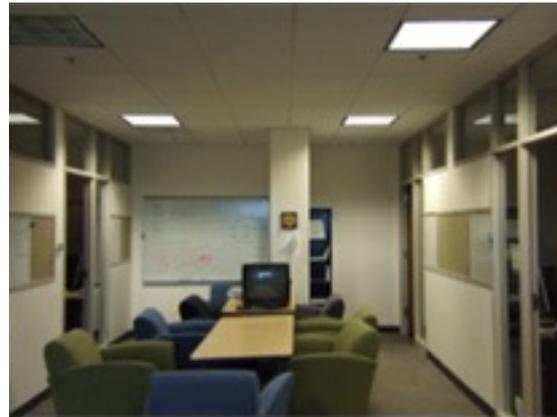
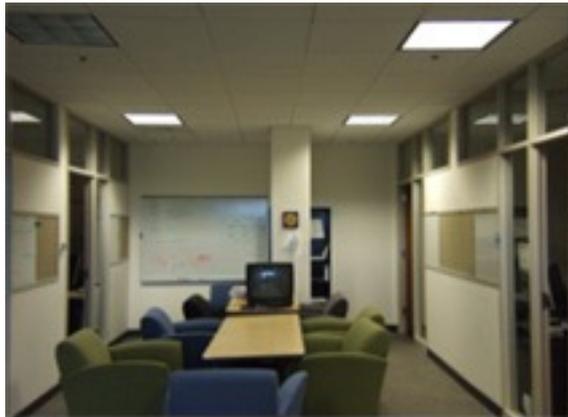
Ex : Capteurs de contact, télémètre, caméra

- **Erreur non cumulative**, mais :
 - **Ambiguïté perceptuelle**
 - **Variabilité perceptuelle**
- Difficilement utilisables seules



Modèle métrique pour les perceptions

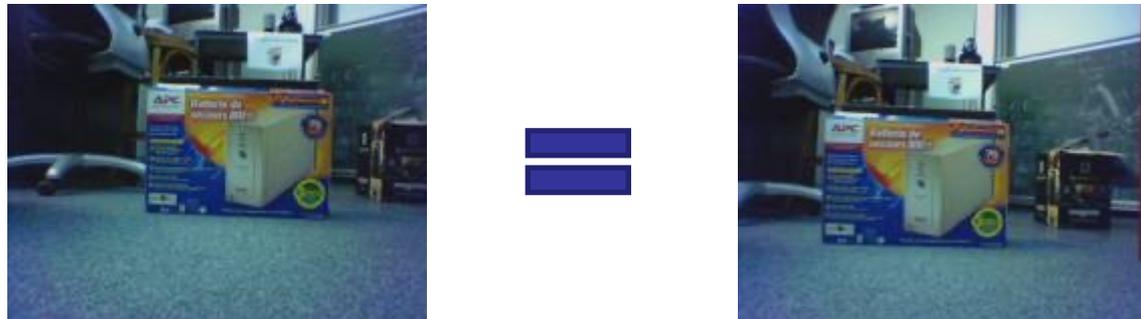
- Permet de retrouver la position métrique d'objets perçus à partir des perceptions (ex : stéréo-vision)



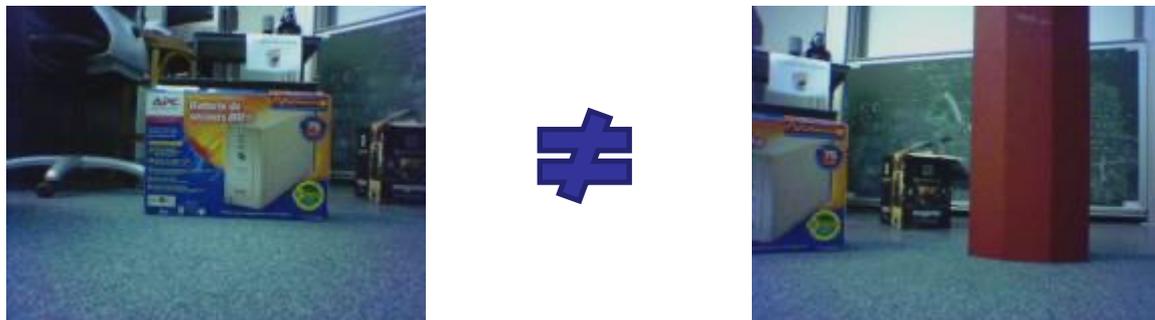
- Possible pour un télémètre, avec des caméras stéréoscopiques, ou une caméra en mouvement
- Impossible ou très difficile pour d'autres capteurs : température, odeur...
- Permet de faire abstraction du capteur utilisé
- Peut dépendre de l'environnement et être complexe à estimer

Perceptions *sans* modèle métrique

- Possibilité de mémoriser et reconnaître une position



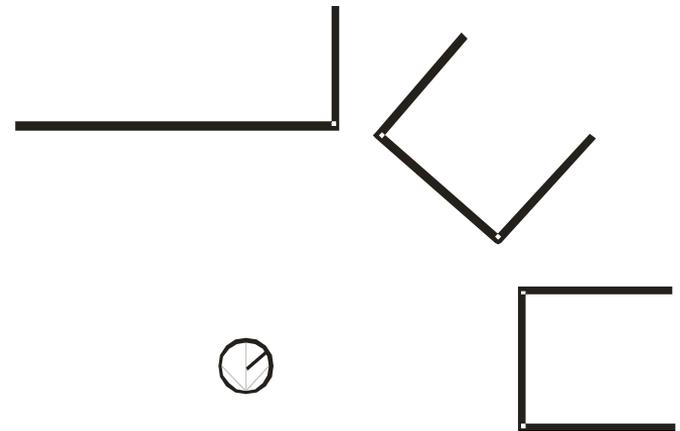
- Mais pas d'information autre que l'identité



-> pas d'info sur la position

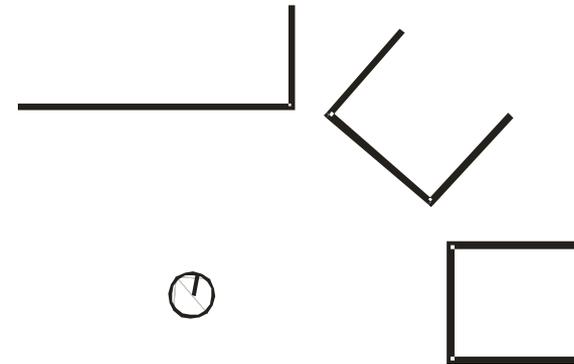
Perceptions *avec* modèle métrique

- Possibilité d'extraire des objets avec leurs positions (dans l'espace des données proprioceptives)
- Création d'une représentation objective

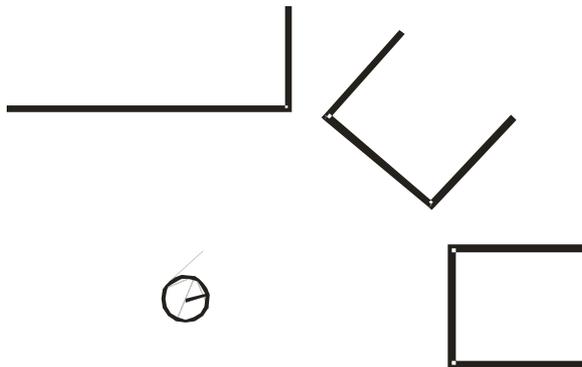


Perceptions *avec* modèle métrique

- Permet de calculer la position du robot par rapport à un objet perçu



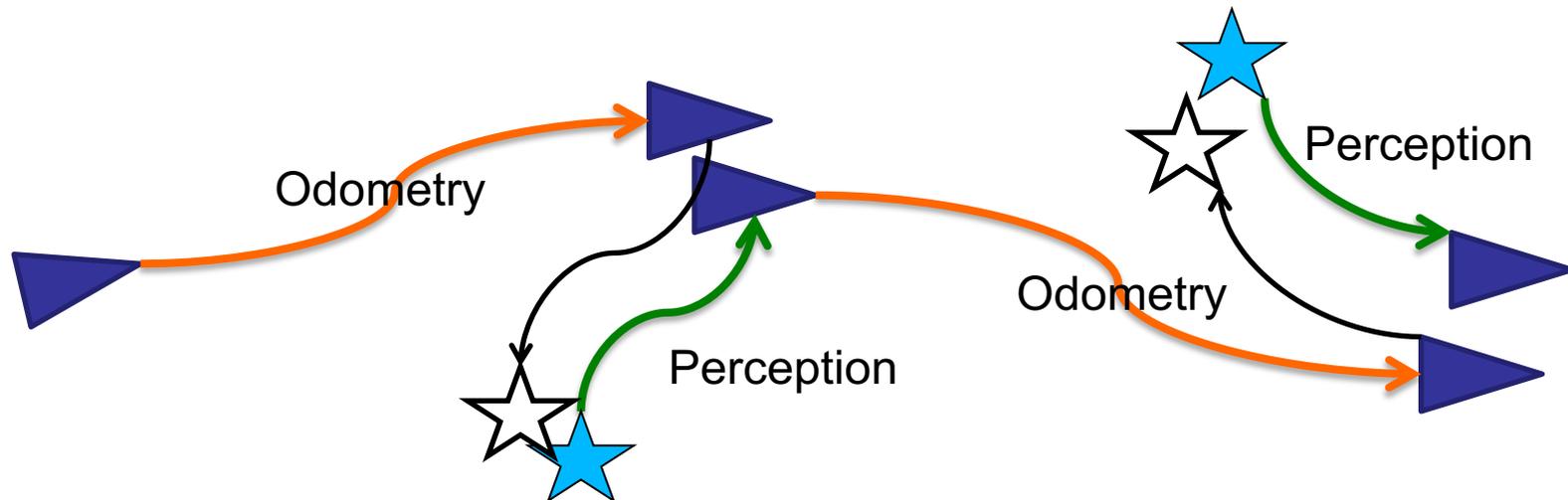
- Permet de calculer les perceptions pour une nouvelle position / un autre capteur



Utilisation conjointe des deux types d'information

- Compenser la dérive de la proprioception par les perceptions
- Lever les ambiguïtés des perceptions par la proprioception

Un bon système de navigation fusionne efficacement ces deux informations



Gestion des incertitudes

- Nombreux algorithmes utilisent de modèles probabilistes
- Information utile pour pondérer les mesures

Incertitude faible  Confiance accrue

Modèles direct

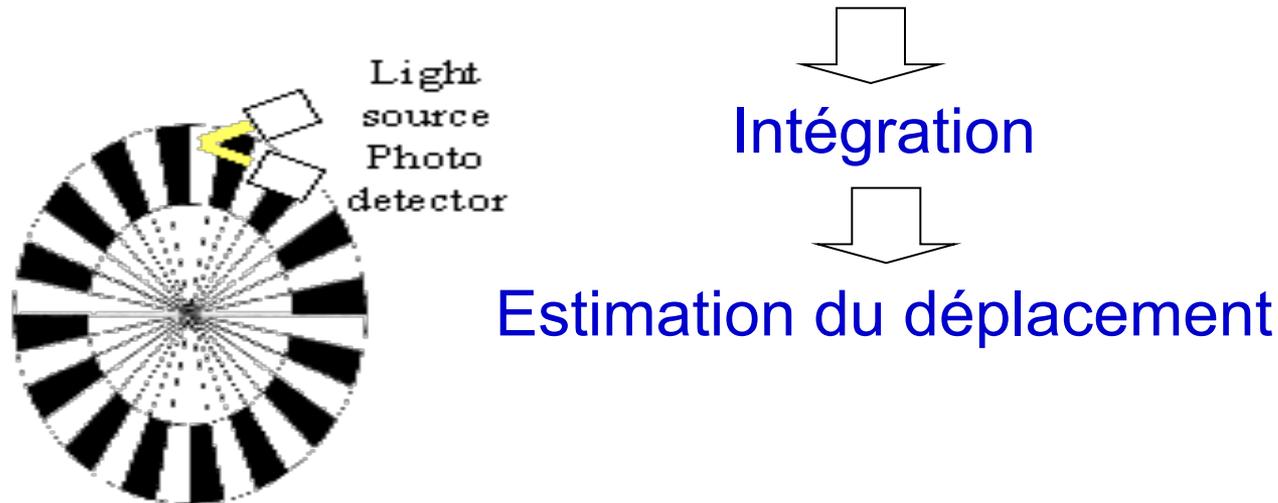
- $P(\text{mesure} \mid \text{valeur réelle})$
- Simple à obtenir, peut être dans la documentation capteur

Modèles inverses

- $P(\text{valeur réelle} \mid \text{mesure})$
- Généralement beaucoup plus complexe
- Mais c'est de qui nous intéresse

Capteurs proprioceptifs

Mesure de la rotation des roues (par ex codeur optique)
ou du déplacement des pattes



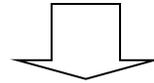
Dépend beaucoup du contact au sol

-> radar doppler / odomètre optique

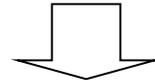
Estimation de la direction peu fiable



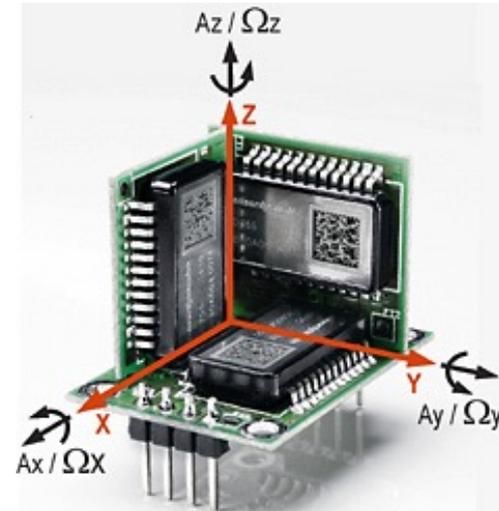
Mesure de l'accélération en translation/rotation



Intégration (2 fois)



Estimation du déplacement



- Les senseurs doivent être très précis : coût élevé
- Capteurs bas coût (précision faible)
 - Estimation de la position quasi impossible
 - Fusion avec accelero/gyro/magnéto pour bonne estimation de la direction

Gyroscopes / Gyromètres

- Direction par rapport à une direction de référence arbitraire
- Systèmes mécaniques, optiques ...

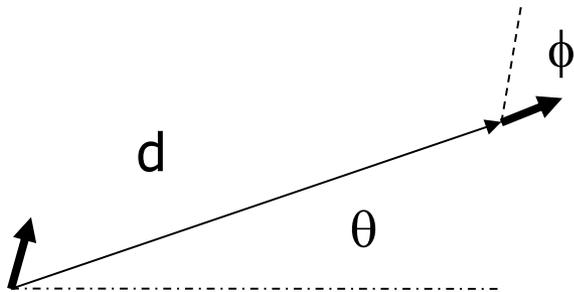
Magnétomètres : mesure le champ magnétique local

- En extérieur, donne le nord magnétique
- Difficile à utiliser en intérieur (perturbation par les masses métalliques)

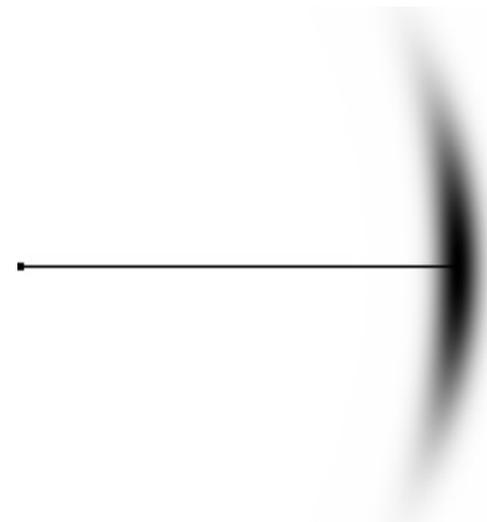
2 méthodes :

- Hypothèses de bruit sur la mesure puis utilisation du processus de calcul de l'odométrie
- Modèle directement sur la sortie : bruit gaussien sur distance, direction et angle

$$P(X|X_m) = \text{Gauss}(d-d_m) * \text{Gauss}(\theta-\theta_m) * \text{Gauss}(\phi-\phi_m)$$

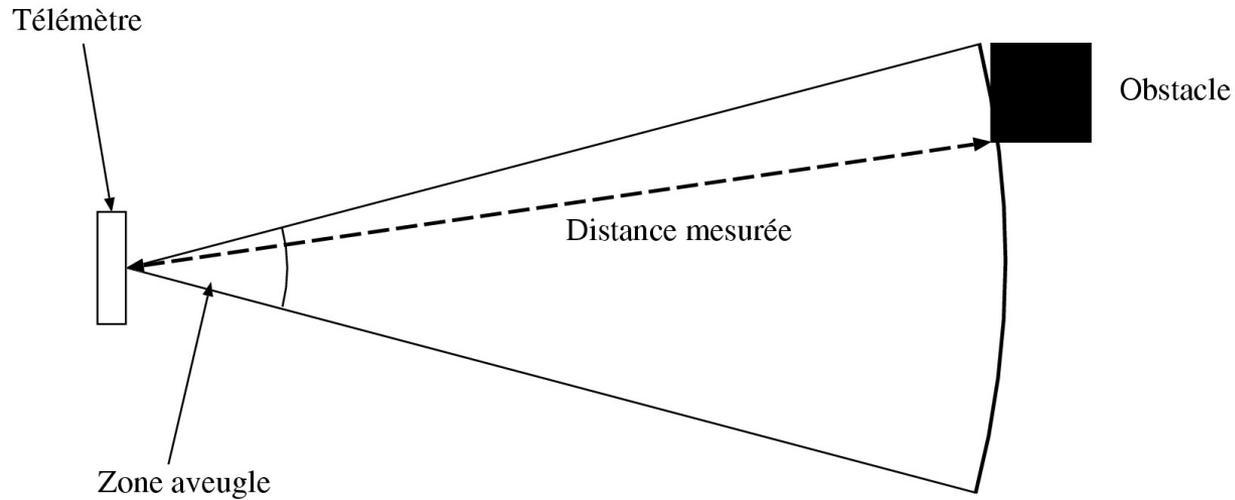


ex :



Capteurs extéroceptifs

Mesure du temps de vol d'une onde sonore

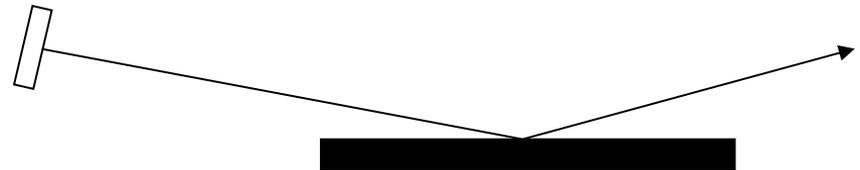


Avantages

- Peu cher
- Angle d'ouverture environ 30 degrés

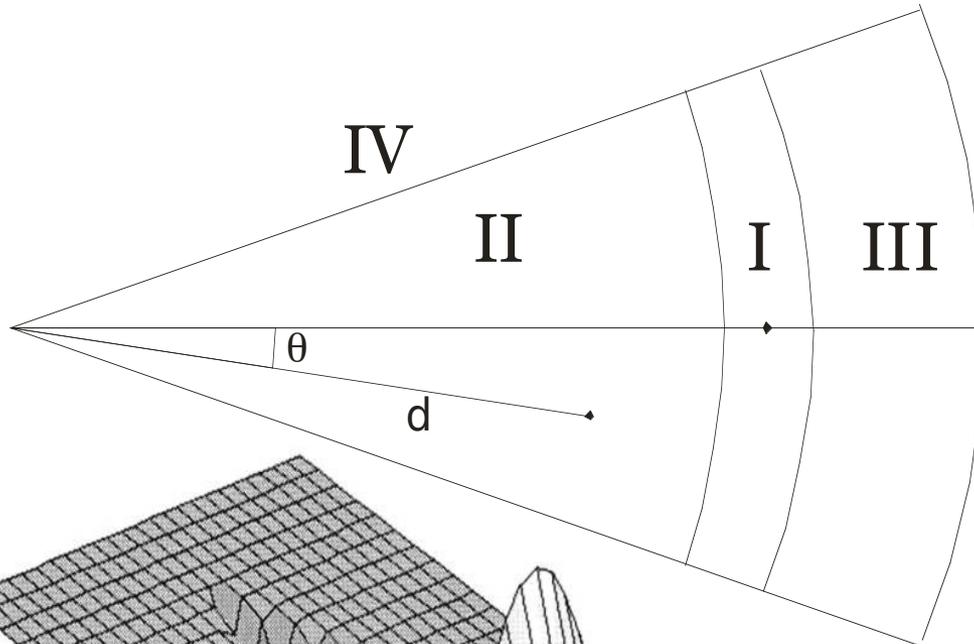
Limitations

- zone aveugle
- crosstalk (→ fréquence qq Hz)
- réflexions multiples
- réflexions spéculaires
- Réponse dépend du matériaux



$P(\text{mesure}|\text{valeur réelle})$: voir plus loin (télémètre laser)

$P(\text{valeur réelle}|\text{mesure})$:

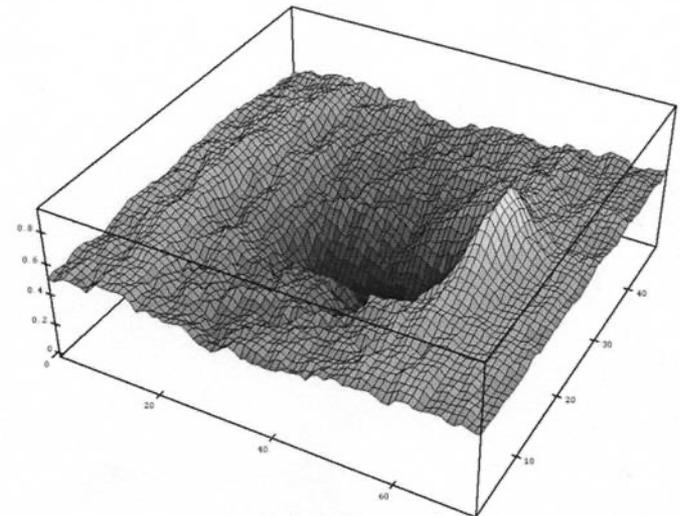
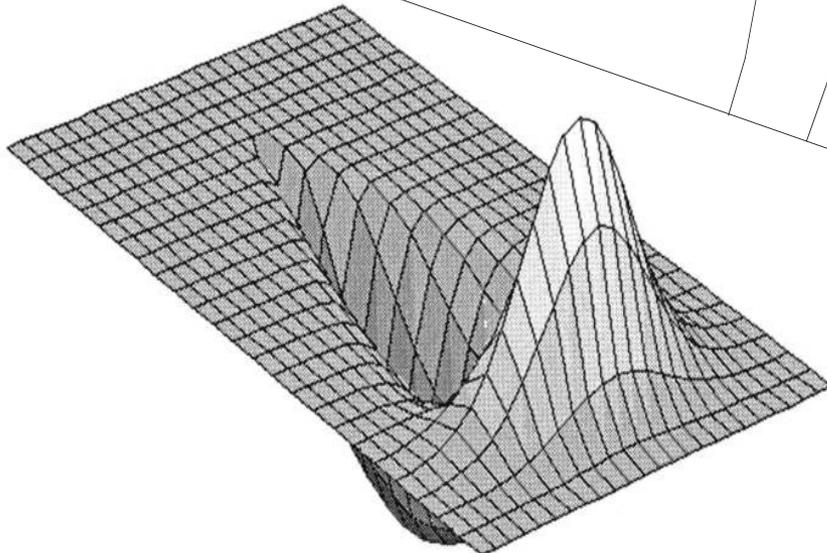


$$\text{I} : \text{Gauss}(\theta) * \text{Gauss}(d - d_m)$$

$$\text{II} : \exp(d) * \exp(-\theta)$$

$$\text{III} : \exp(d) * \exp(-\theta)$$

$$\text{IV} : 0$$



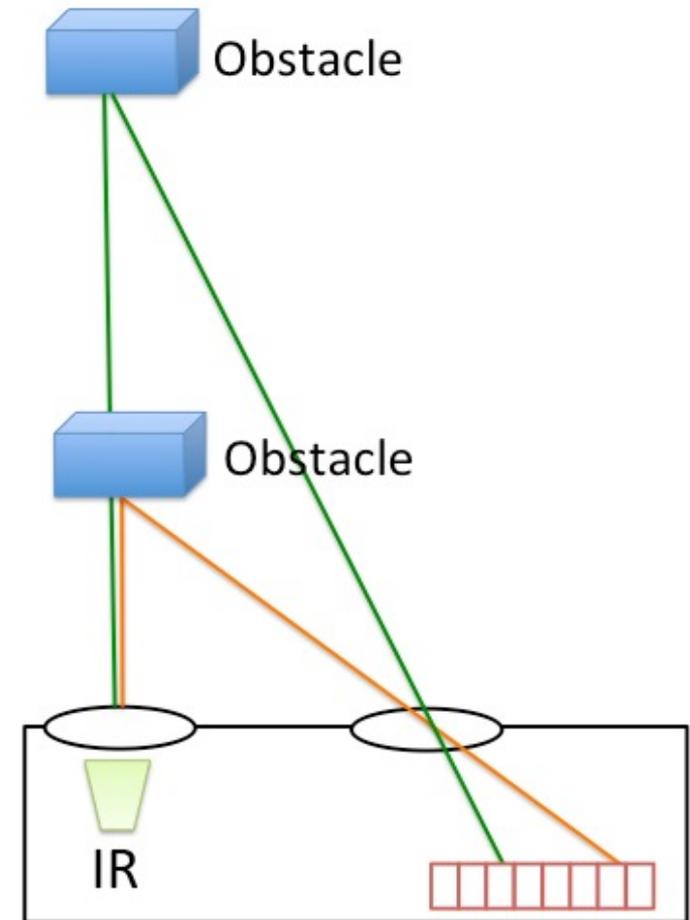
<http://www.cs.hmc.edu/courses/2003/spring/cs154/>

Triangulation d'un faisceau IR

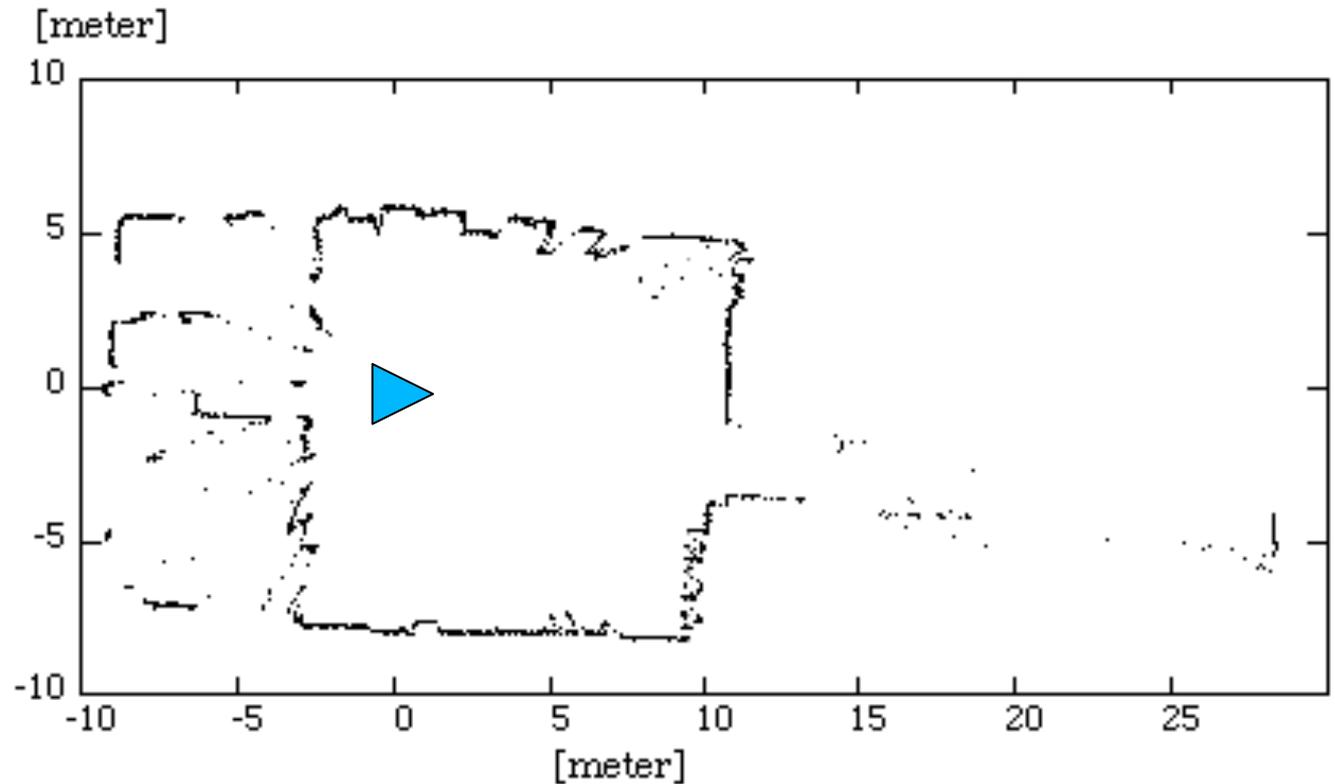
- Peu cher
- Faible consommation
- Peu d'interférences

Limitations

- Faible angle d'ouverture
- Sensible à la lumière extérieure



Mesure de distance grâce à un laser balayant un plan (LIDAR)



Avantages

- Couvre 180 à 360 degrés jusqu'à environ 50m
- Résolution $\frac{1}{2}$ ou 1 degré
- Fréquence : 10 -> 75 Hz
- Bruit de qq cm

Limitations

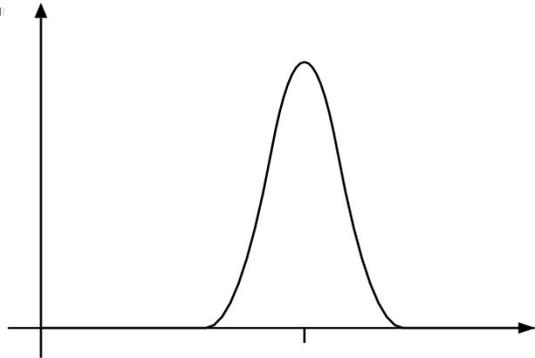
- Restreint à un plan → des obstacles non perçus
- Certains objets réfléchissants non détectés (réflexions spéculaires)
- Vitres (propres!) non détectées
 - > association avec sonars

Pour 1 faisceau

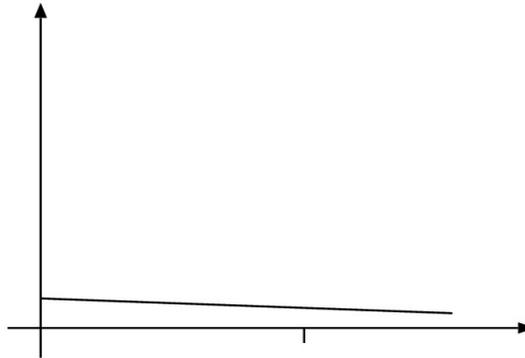
- Combinaison d'évènements
 - Perception de l'obstacle : bruit gaussien : $\text{gauss}(d-d_m)$
 - Perception d'un obstacle inattendu (personne ...) : bruit décroissant : $\exp(-d)$
 - Non perception des obstacles : bruit ponctuel de valeur max

Estimation des paramètres

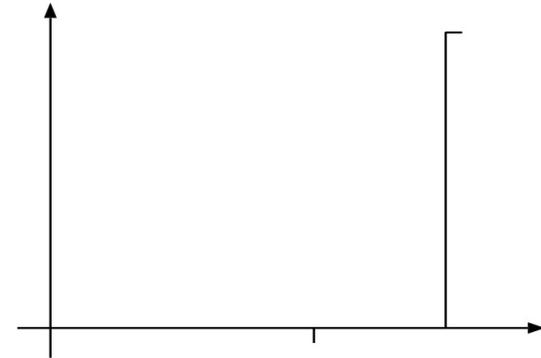
- Sur un ensemble de données
- Algorithme Expectation-Maximization
 - Associer les mesures à un évènement (plus probable)
 - Estimer paramètres des distributions élémentaires
 - Ré-associer



Perception de l'obstacle

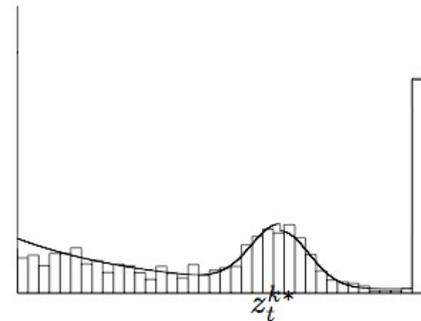
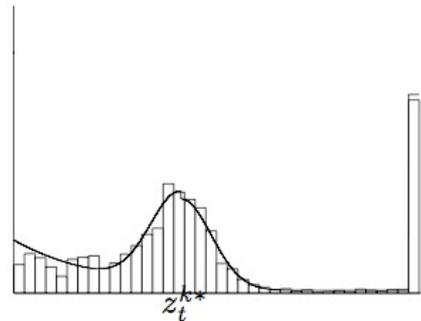


Perception d'un obstacle imprévu

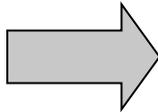
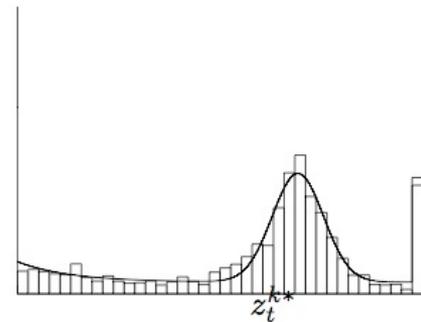
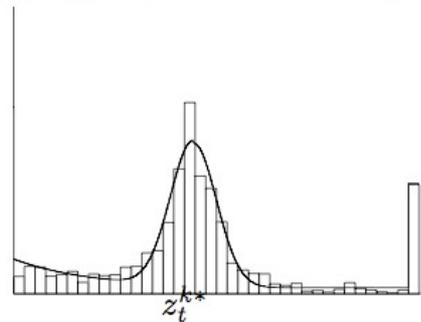


Retour de la valeur maximale

(a) Sonar data, plots for two different ranges



(b) Laser data, plots for two different ranges



Pour 1 scan

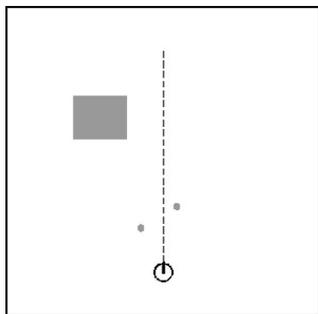
- Faisceaux supposés indépendants

$$P(S) = \prod_{i=0}^n P(s_i)$$

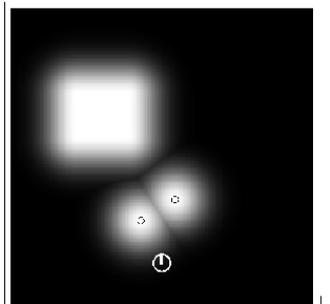
Rq : Approximation pour un faisceau

- Ne prendre en compte que le point mesuré
- probabilité = fct de la distance à l'obstacle le plus proche (pré-calcul)
- Modèle très approché, Ignore les occultations, mais correct près du max
- Utile si besoin de beaucoup d'évaluations (cf filtre particulière)

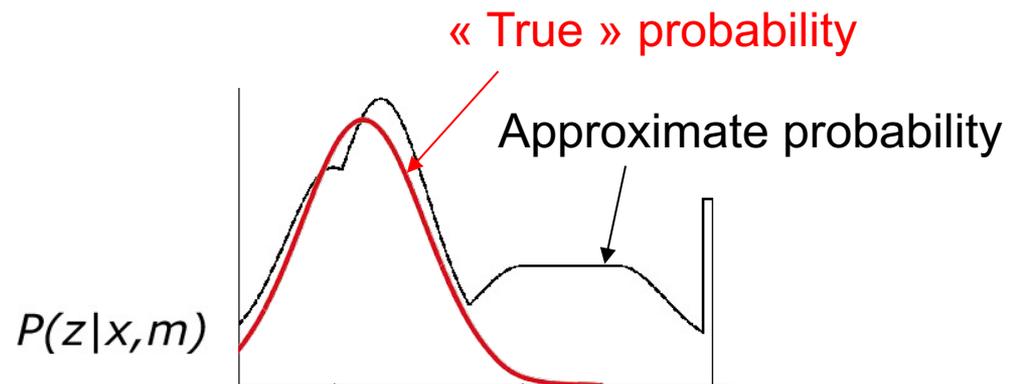
Example



Map m

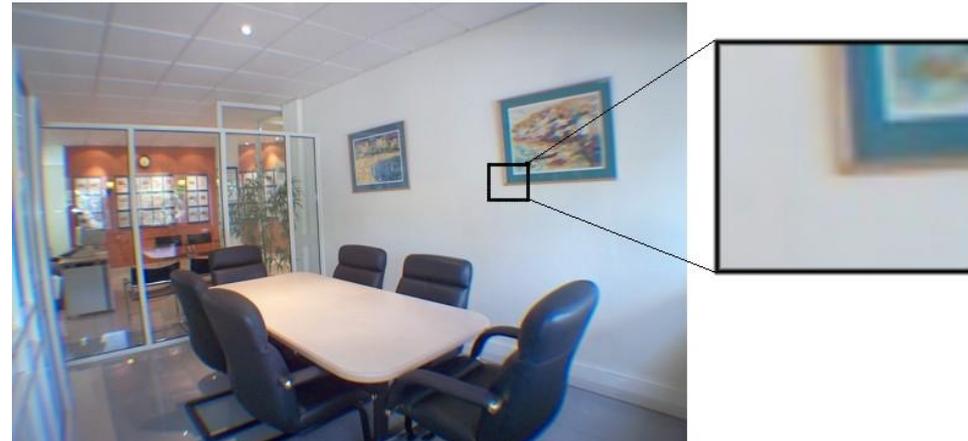


Likelihood field

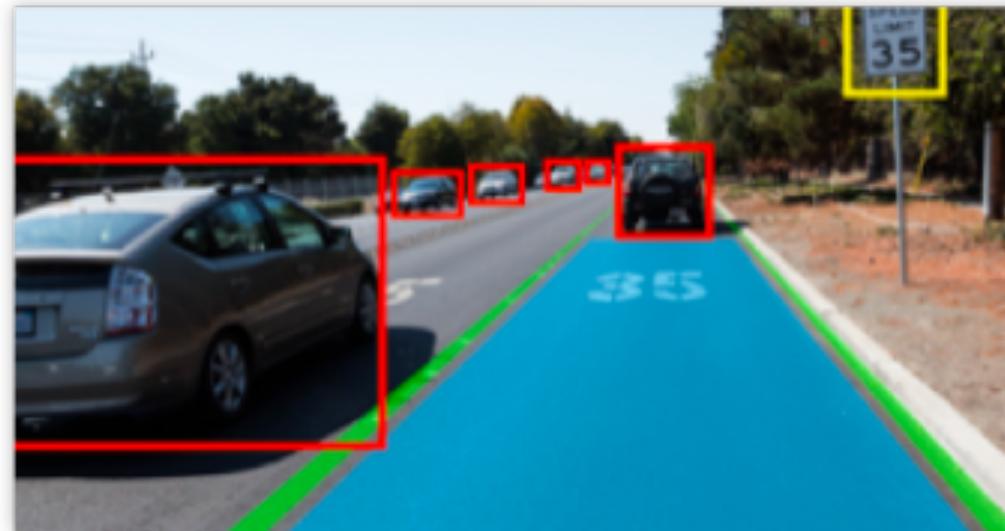
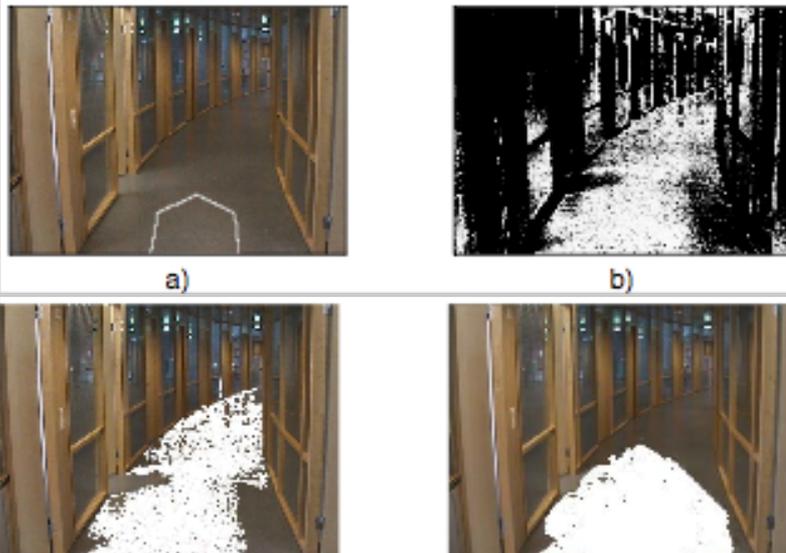


Caméra simple

- Détection d'amers 2d ou 3d (points, segments, imajettes, objets ...)



- Détection du sol, de la route, de l'espace navigable ...



Caméra simple

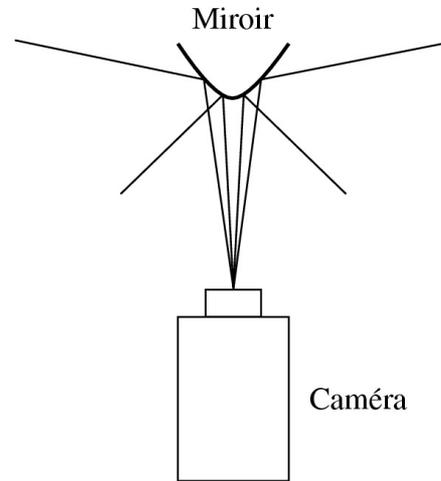
- Flot optique (odométrie, évitement d'obstacles)



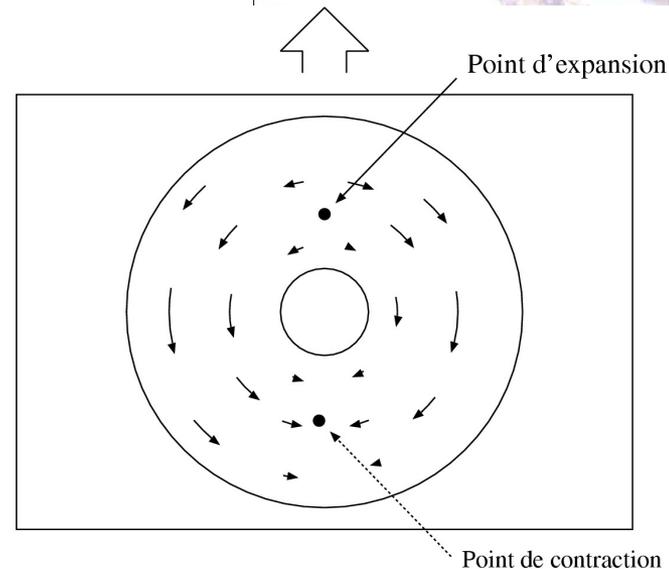
- Détection et suivi d'objets mobiles ...

Caméra panoramique

- Caractérisation de position



- Flot optique
 - Point de contraction et expansion visibles



Caméra stéréoscopiques

- Info 3D
- Portée limitée
- Besoin de textures (lumière)

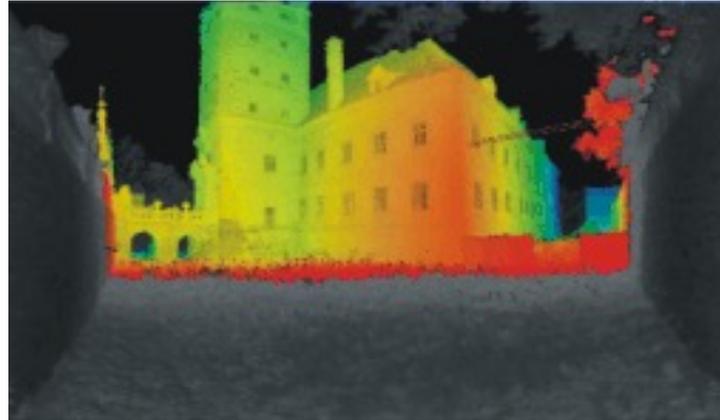


Caméra en mouvement

- ' Structure from motion '
- Approche complexe
- Appariements difficiles
- Similarités avec le SLAM

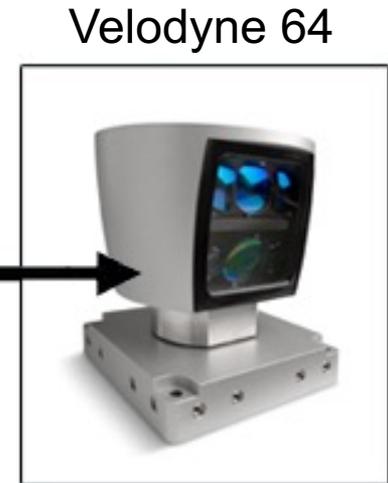


Laser balayant sur 2 axes



Très bonne info 3D
Fréquence d'acquisition faible
Mécanique importante (lourdeur, fragilité)

Laser multi-nappes



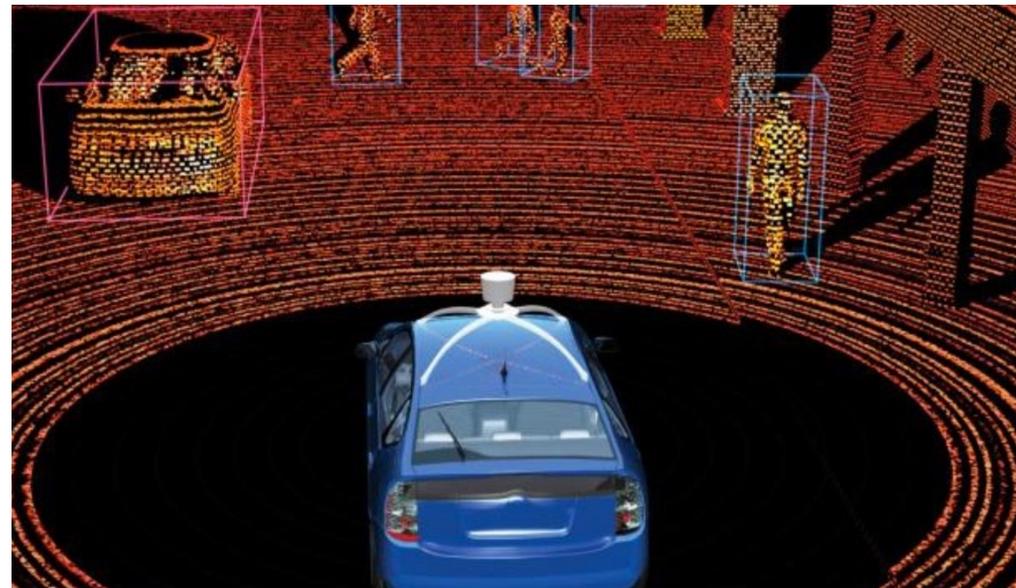
Faisceaux laser rotatifs
Vision à 360 degrés



32



16

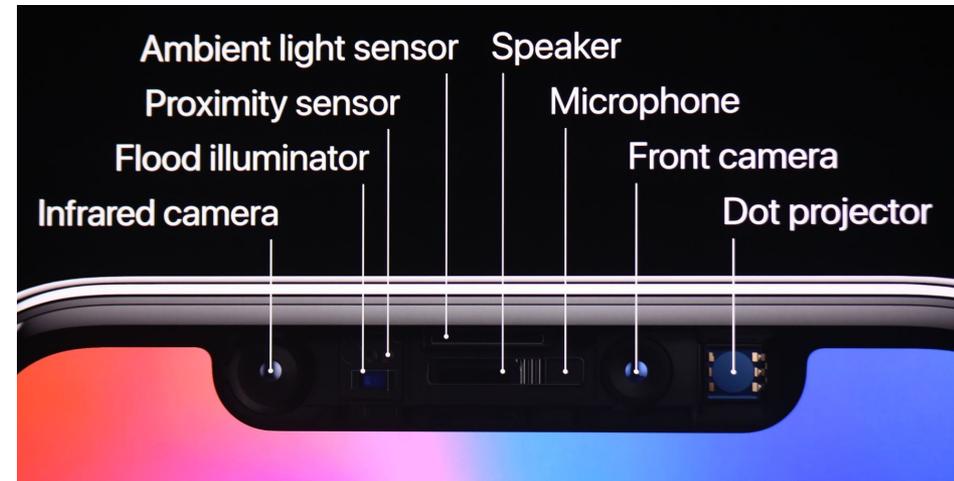


Triangulation en infrarouge + caméra couleur

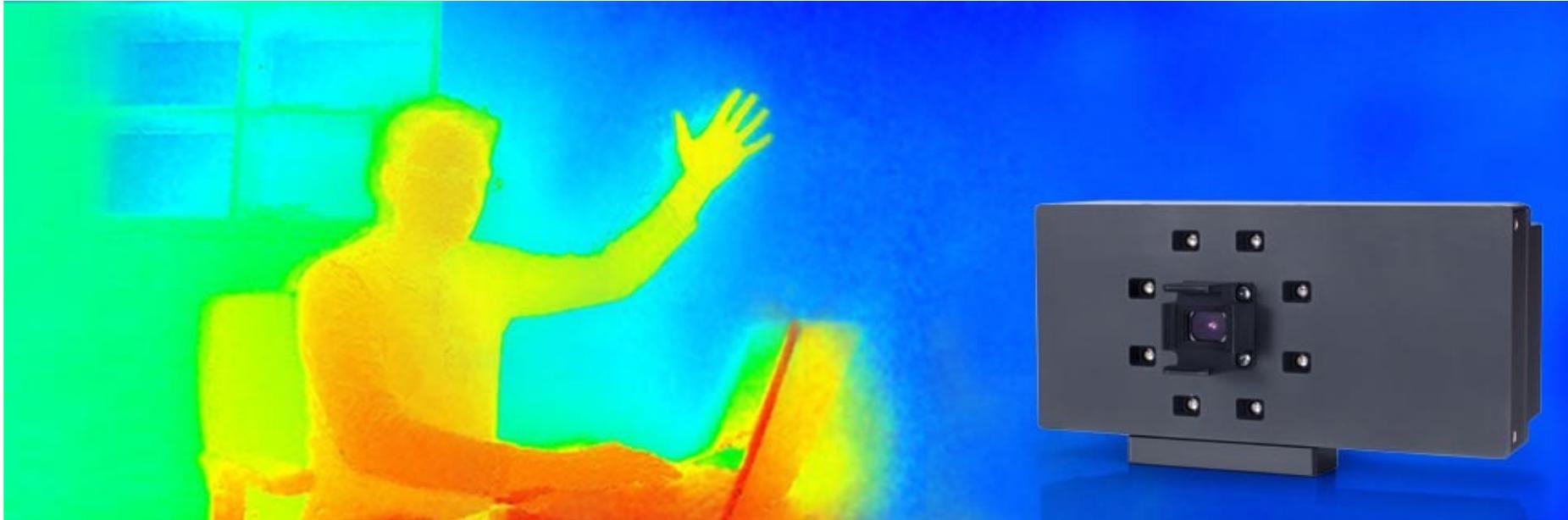


Acquisition dense à grande fréquence
Sensible aux conditions d'éclairage
Mesure bruitée

Kinect v1, Xtion, iPhone X;



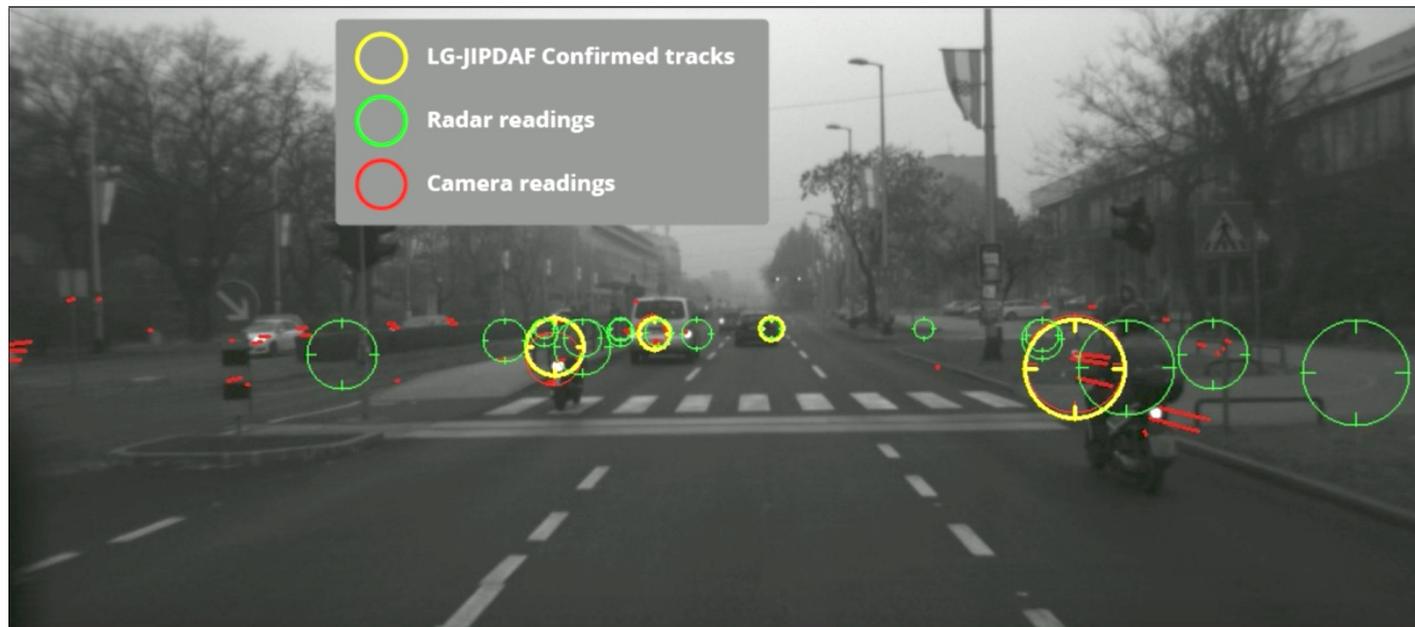
Caméra temps de vol



Kinect v2, Swiss Ranger
Acquisition dense à grande fréquence
Moins sensible aux conditions d'éclairage
Plus coûteux que la triangulation IR

Radar FMCW

- Frequency Modulated Continuous Wave
- Bonne mesure de distance, vitesse
- Portée élevée ($> 250\text{m}$), peu sensible à la météo
- Résolution angulaire limitée, beaucoup de bruit
- Difficulté pour les objets non métalliques (piétons...)



Capteurs de contact

- Arrêt d'urgence, évitement d'obstacles

Systèmes de balises

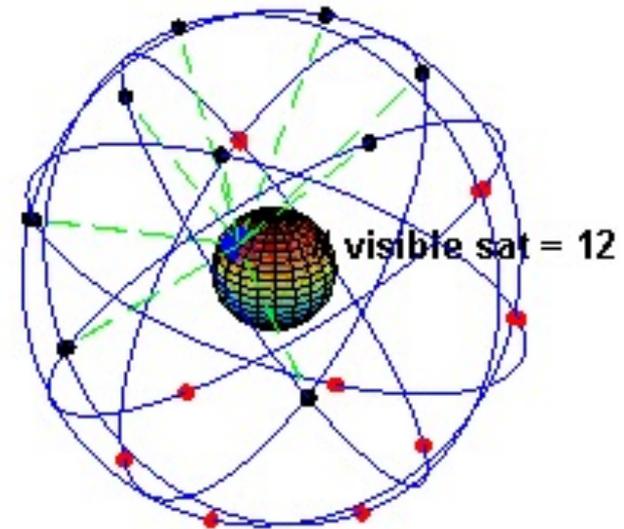
- **Préparation de l'environnement**
- Couleurs, codes barres, radio, infrarouge
- Utile dans certaines applications



Kiva Systems

GPS

- Système de balise global
- **Pas d'info sur l'environnement**
- Limitations en intérieur/urbain
- Précision typique
 - GPS ~ 6/8 m
 - DGPS ~ 1m

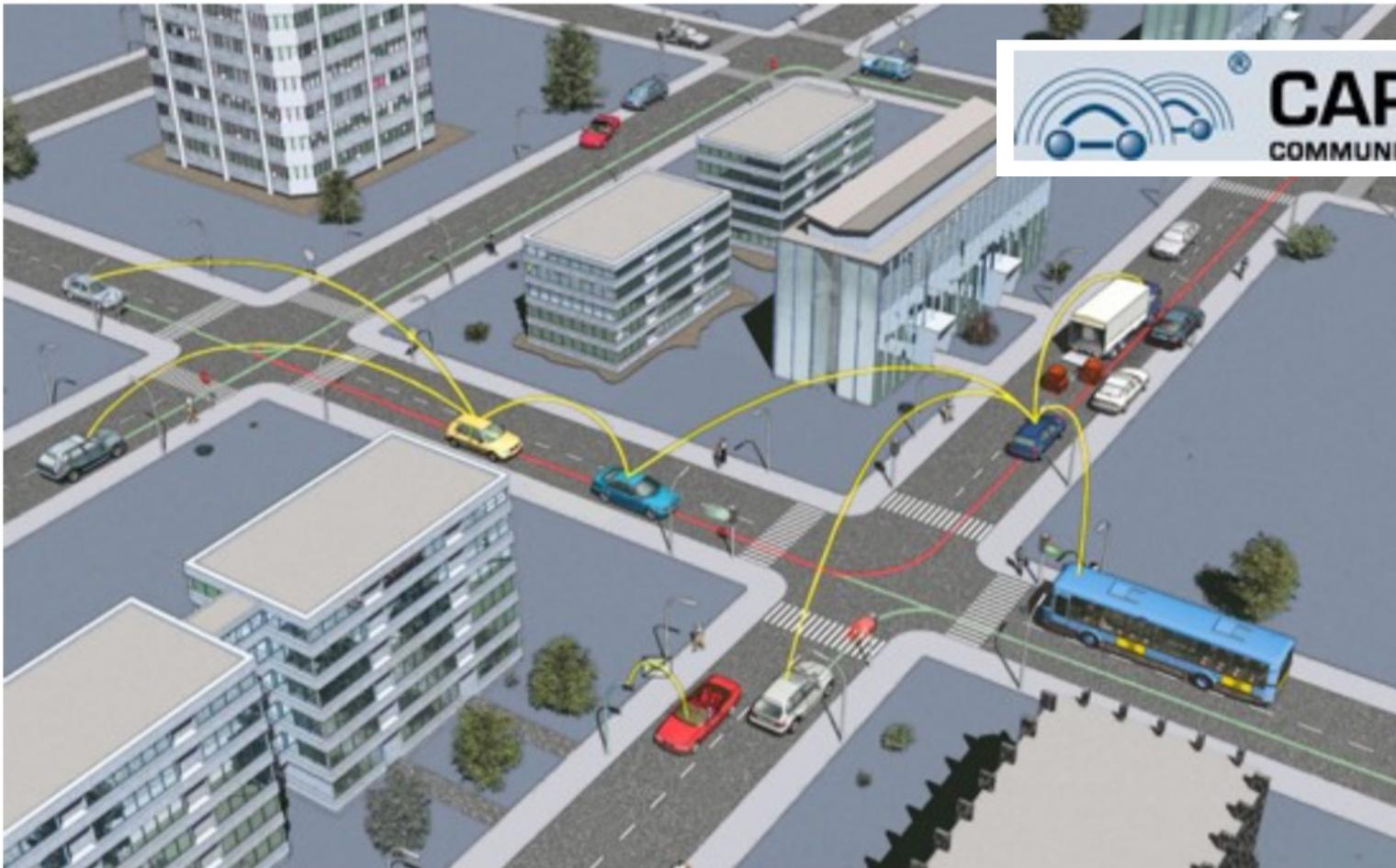


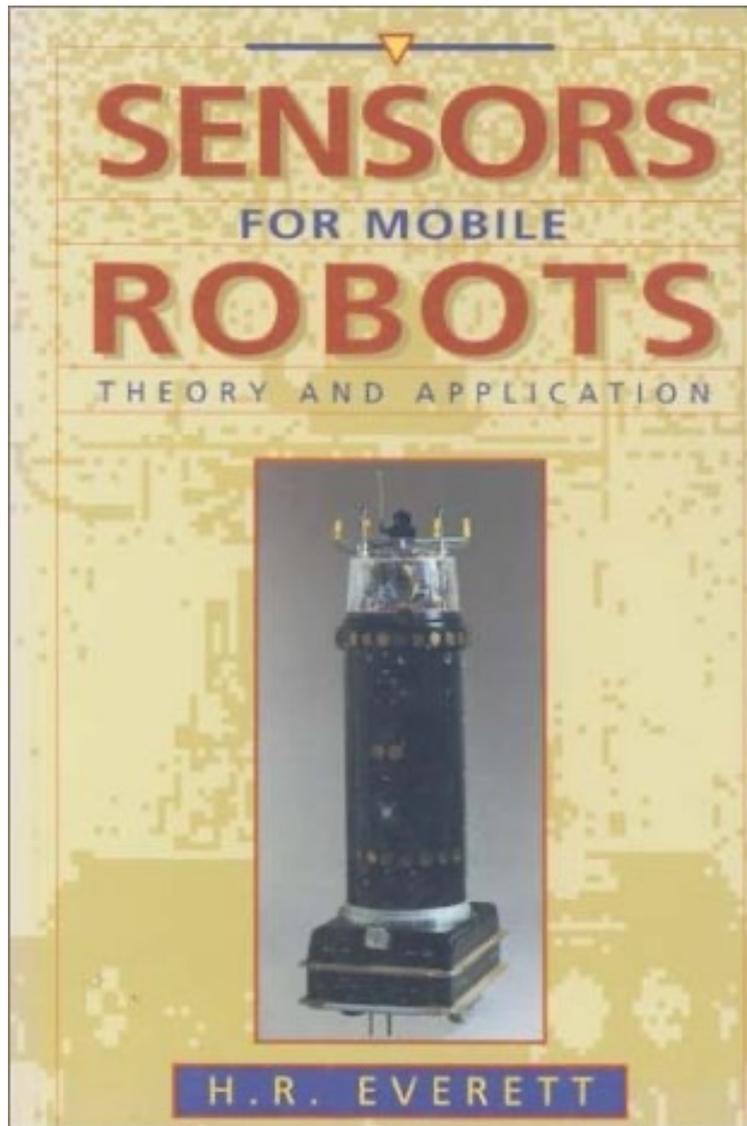
Nombreuses possibilités d'amélioration

- Corrélation avec carte
- Fusion avec inertie/vision...
- Inévitable en extérieur

V2X

- Informations venant de l'infrastructure / des autres usagers





**Sensors for Mobile Robots:
Theory and Application**
by [H. R. Everett](#)

Capteurs - En résumé

- Il existe deux grand types de capteurs:
 - Proprioceptifs qui mesurent les déplacement avec une erreur cumulative
 - Extéroceptifs qui mesurent la position avec des ambiguïtés indépendantes du temps
- Les centrales inertiellees donnent généralement une bonne orientation, mais sont chères pour obtenir des informations de position
- Les télémètres laser sont les capteurs les plus utilisés pour la navigation en robotique, avec un faisceau en intérieur, plusieurs faisceaux en extérieur
- Les différents types de caméras fournissent beaucoup d'information mais demandent des traitements assez complexes, très poussés par le deep learning

Robotique Mobile

06 - Plateformes

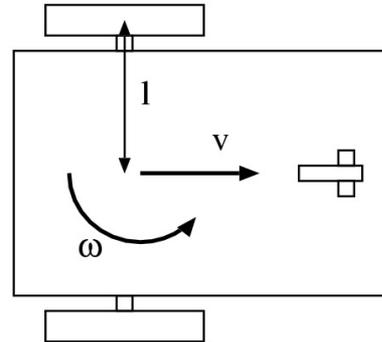
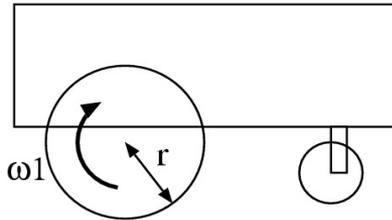
David Filliat

Alexandre Chapoutot

Goran Frehse

prenom.nom@ensta-paris.fr

Unicycle model



$$v = \frac{\omega_1 r + \omega_2 r}{2}$$

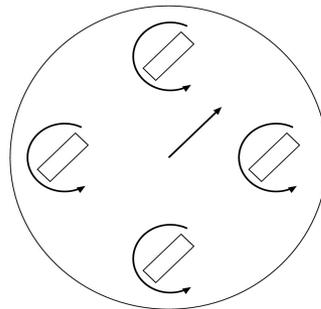
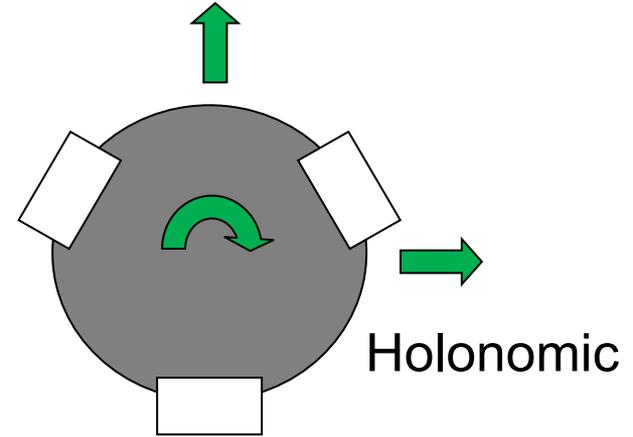
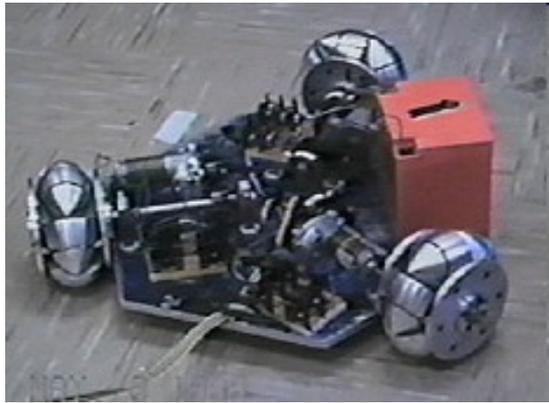
$$\omega = \frac{\omega_1 r - \omega_2 r}{2l}$$

- Two independent wheels/tracks
- Simple command
- Can decouple translation and rotation (~holonomy)
- Simple odometry, but noisy (in particular for tracks)

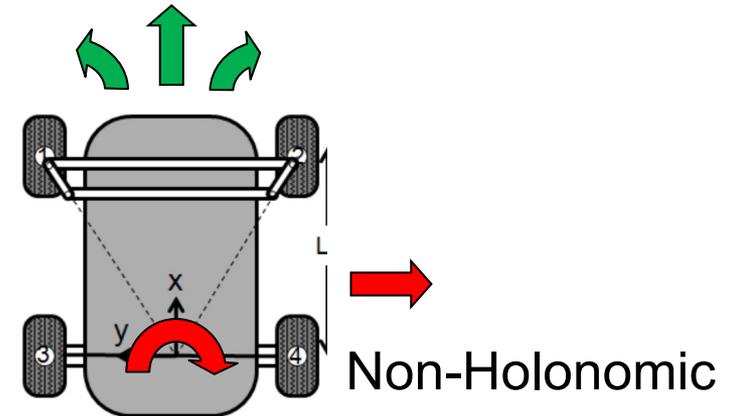


Holonomic platforms

- Can be controlled along any degree of freedom
- Wheels that can move sideways or Synchronized wheels

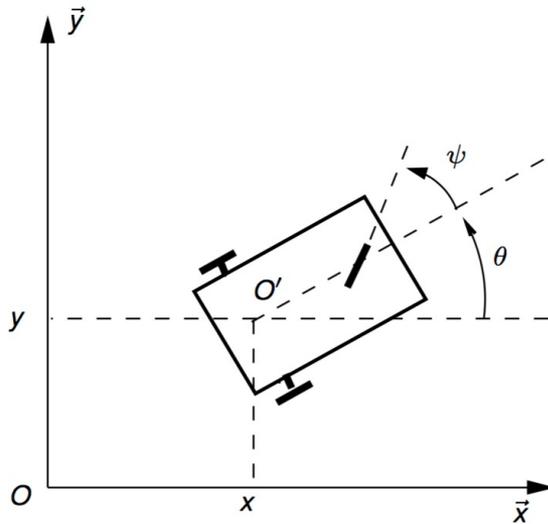


Holonomic bases



Bicycle / Tricycle model

- Control speed + front wheel angle



$$\dot{x} = v \cos \theta$$

$$\dot{y} = v \sin \theta$$

$$\dot{\theta} = \frac{v}{D} \tan \psi$$

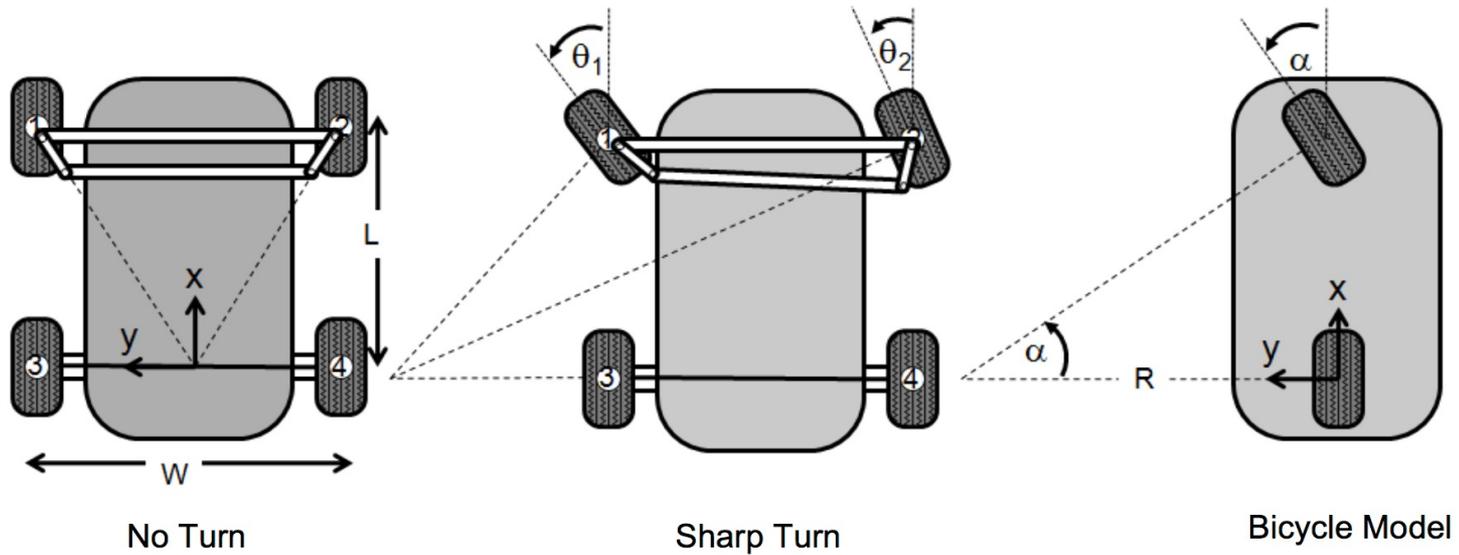
$$\dot{\psi} = \eta$$

- Non holonomic model
- Constraint motion more complex control

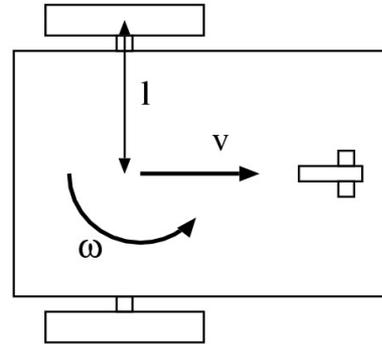
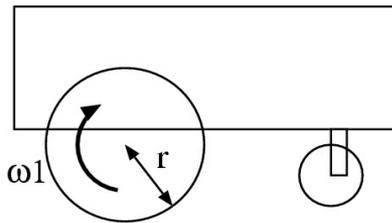
See « Robotique Mobile », Bernard Bayle,
Telecom Physique Strasbourg

Ackerman model

- Close to bicycle, with two front wheels
- Articulation to guarantee uniqueness of rotation center



Deux roues dont on commande les vitesses indépendamment



$$v = \frac{\omega_1 r + \omega_2 r}{2}$$

$$\omega = \frac{\omega_1 r - \omega_2 r}{2l}$$

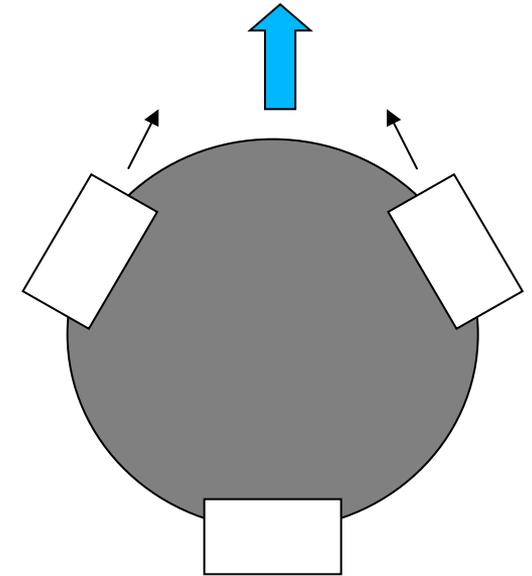
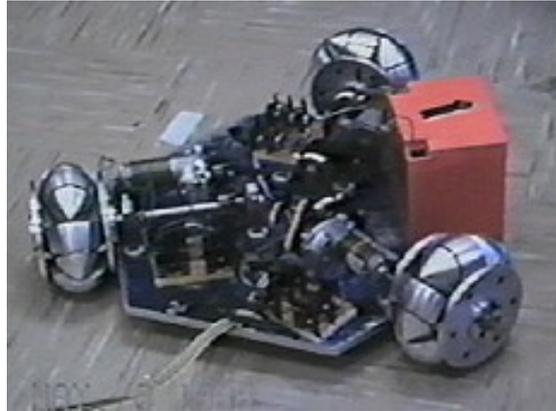
- Modèle **unicycle**
- Commande simple
- Permet de découpler translation et rotation
- Odométrie simple, mais bruitée (surtout pour des chenilles)



Découplage complet de la direction et de la translation



Roue omnidirectionnelle

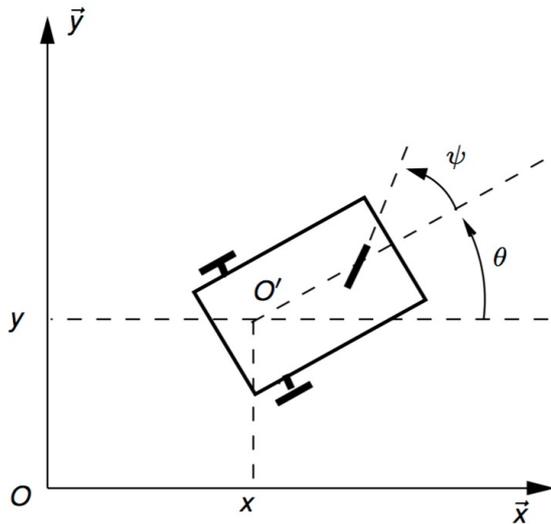


Base holonome



Modèle tricycle

- Commande vitesse + angle de roue



$$\dot{x} = v \cos \theta$$

$$\dot{y} = v \sin \theta$$

$$\dot{\theta} = \frac{v}{D} \tan \psi$$

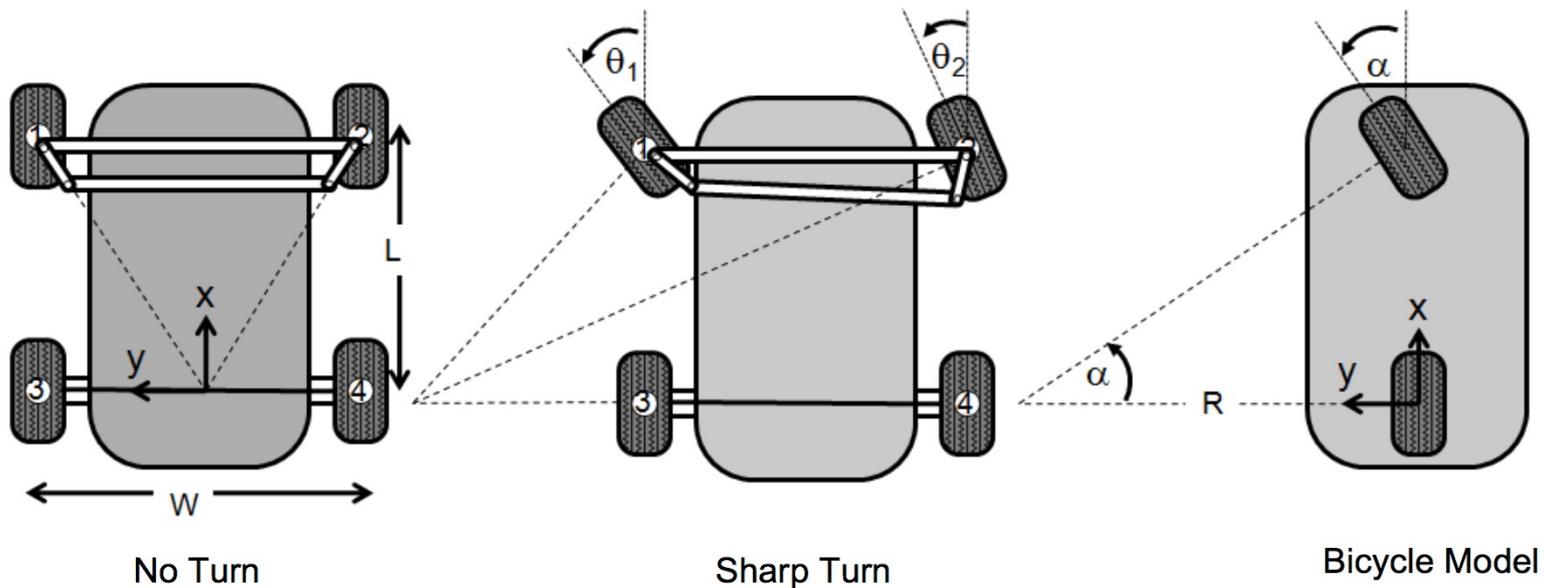
$$\dot{\psi} = \eta$$

- Modèle non holonome
- Déplacements contraints et commande un peu plus complexe

Voir cours « Robotique Mobile », Bernard Bayle,
Telecom Physique Strasbourg

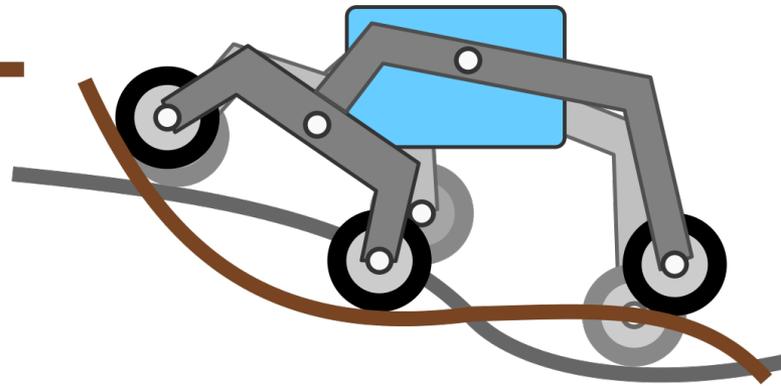
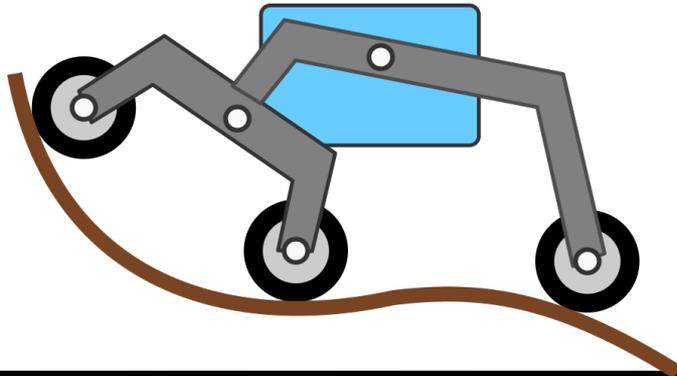
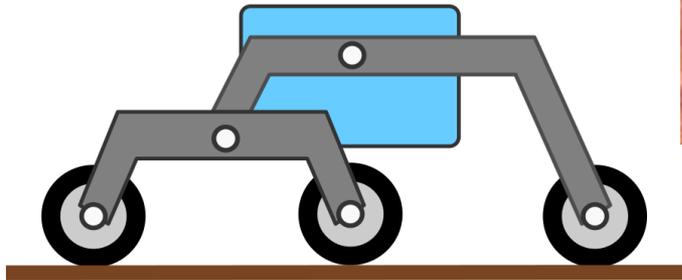
Modèle Ackerman

- Articulation des deux roues avant
- Proche du tricycle



Rocker-Bogie

- Degrés de liberté supplémentaires pour une plus grande mobilité



SHRIMP - EPFL



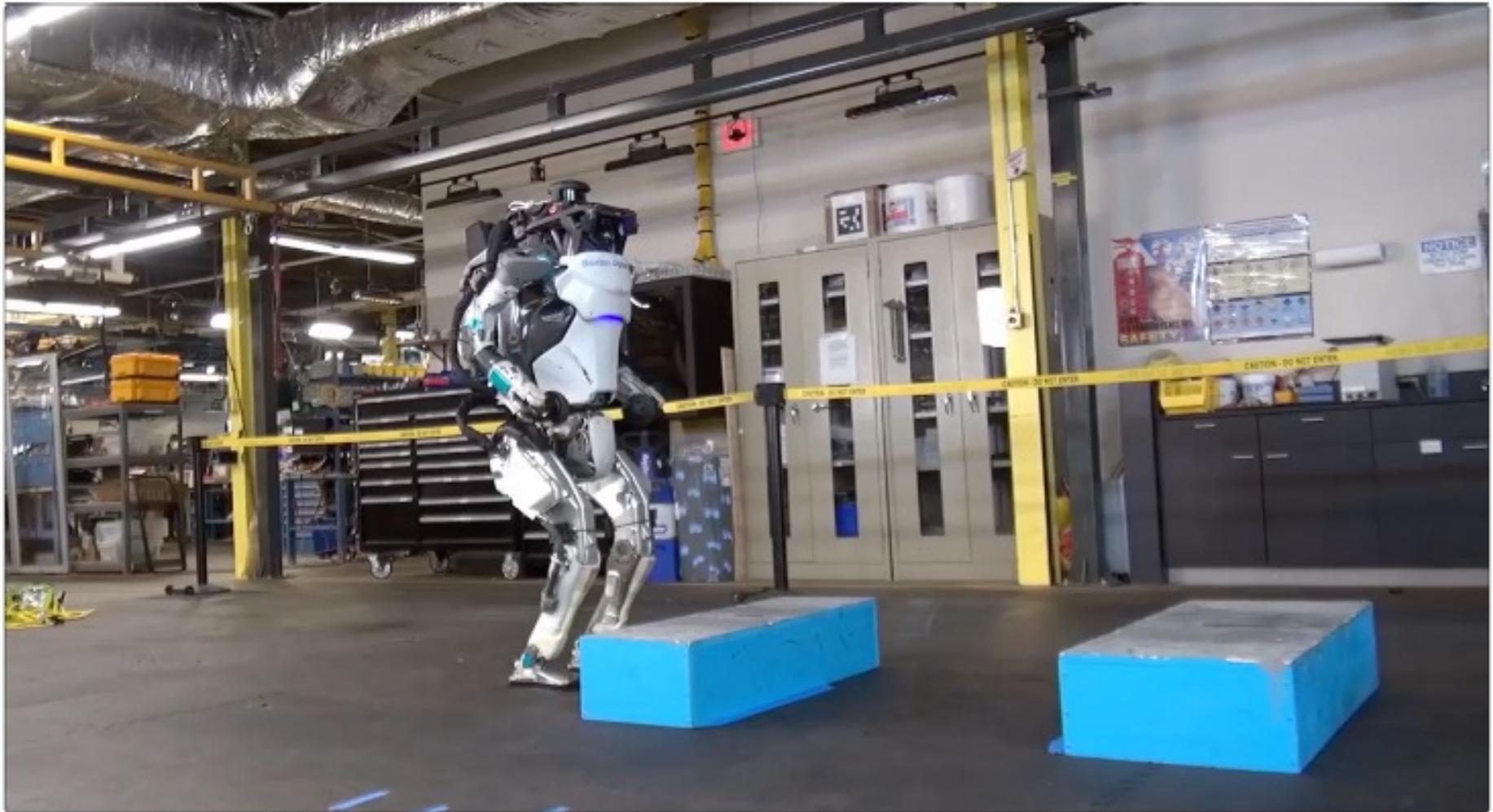
Mobilité en terrain quelconque

- Commande complexe



Boston Dynamics

Boston Dynamics



Boston Dynamics

Darpa Robotics Challenge (2015)

– Video du vainqueur (KAIST)



Darpa Robotics Challenge (2015)

– Behind the scene



Plateformes - En résumé

- Les plateformes **holonomes** permettent des déplacements dans toutes les directions, mais la plupart des plateformes ne le sont pas
- Les modèles les plus classiques sont les plateformes unicycle (2 roues différentielles) et tricycle (une roue avant orientable)
- Les voitures peuvent se représenter des manière approchée par un modèle tricycle
- Les plateformes à pattes sont potentiellement plus versatiles, mais restent encore peu matures

Robotique Mobile

07 – Navigation Réactive

David Filliat

Alexandre Chapoutot

Goran Frehse

prenom.nom@ensta-paris.fr

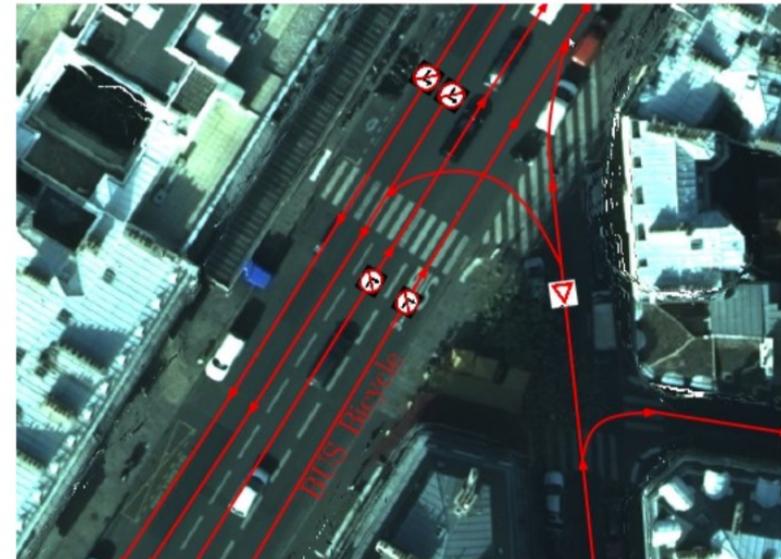
Navigation réactive

- Champ d'application restreint
- Très rapide
- Robuste (potentiellement ...)



Navigation utilisant une carte

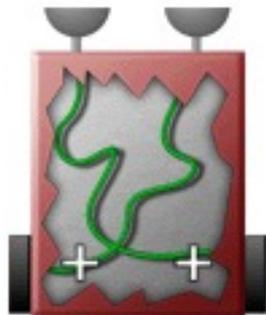
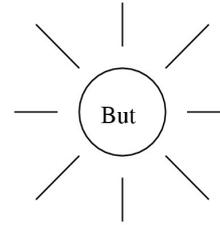
- Construction d'un modèle de l'env.
- Planification de déplacements à long terme



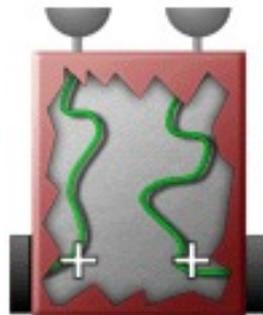
Comportement réactif : atteindre
un but

Expériences de pensée

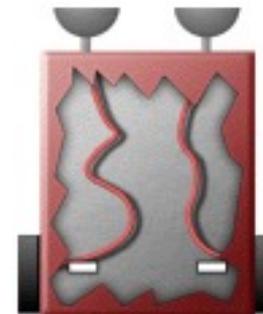
- Catégorie 1 de Trullier
- Architecture réactive triviale
- Liens capteurs de lumière/moteurs



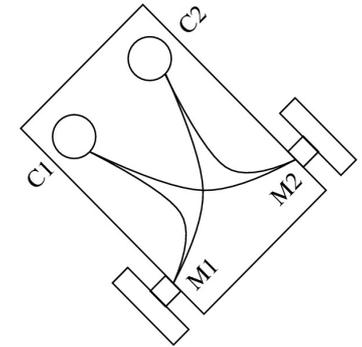
Aggressif



Peureux



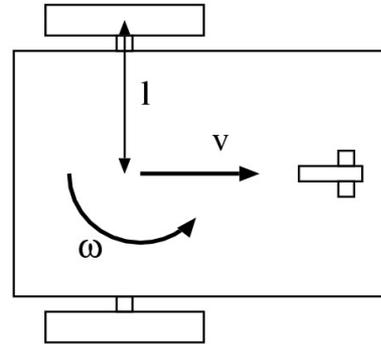
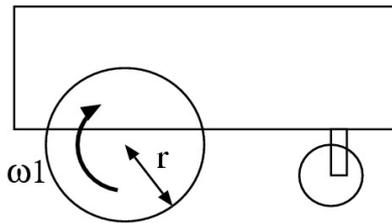
Amoureux



Comportements variés en fonction des paramètres reliant les capteurs aux moteurs

- **Difficilement utilisable sur un vrai robot** (équivalent à un simple contrôleur Proportionnel) \Rightarrow oscillations
- Utiliser au moins un contrôleur Proportionnel-Dérivée par exemple

Deux roues dont on commande les vitesses indépendamment



$$v = \frac{\omega_1 r + \omega_2 r}{2}$$

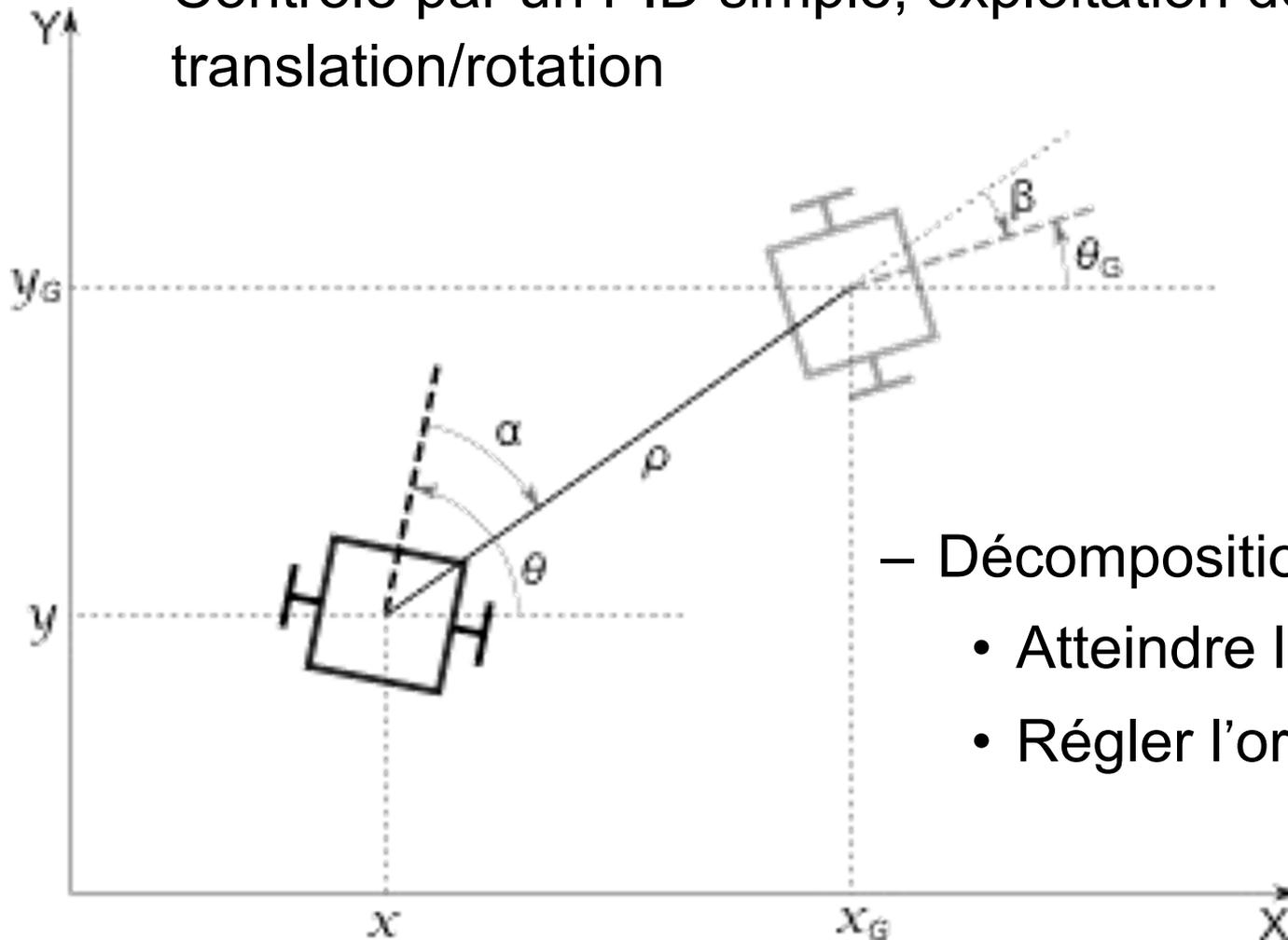
$$\omega = \frac{\omega_1 r - \omega_2 r}{2l}$$

- Modèle **unicycle**
- Commande simple
- Permet de découpler translation et rotation
- Odométrie simple, mais bruitée (surtout pour des chenilles)



Atteindre une position

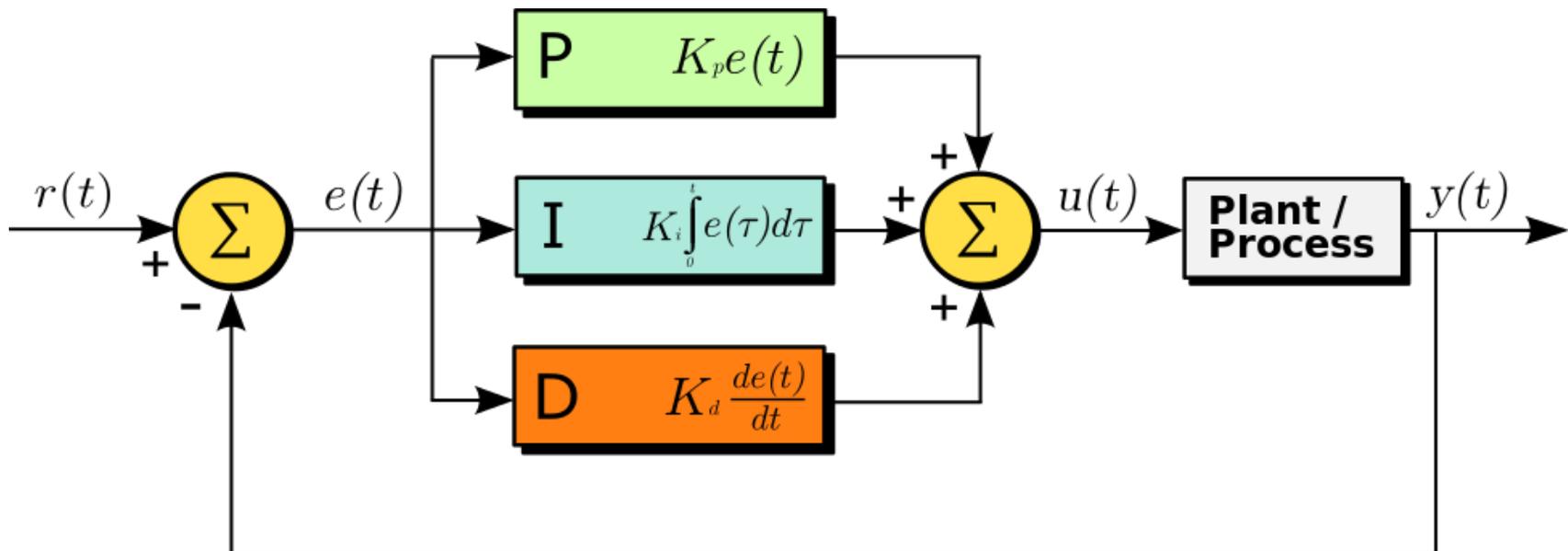
- Contrôle par un PID simple; exploitation de la séparation translation/rotation



- Décomposition du problème:
 - Atteindre la position
 - Régler l'orientation

Contrôleur PID

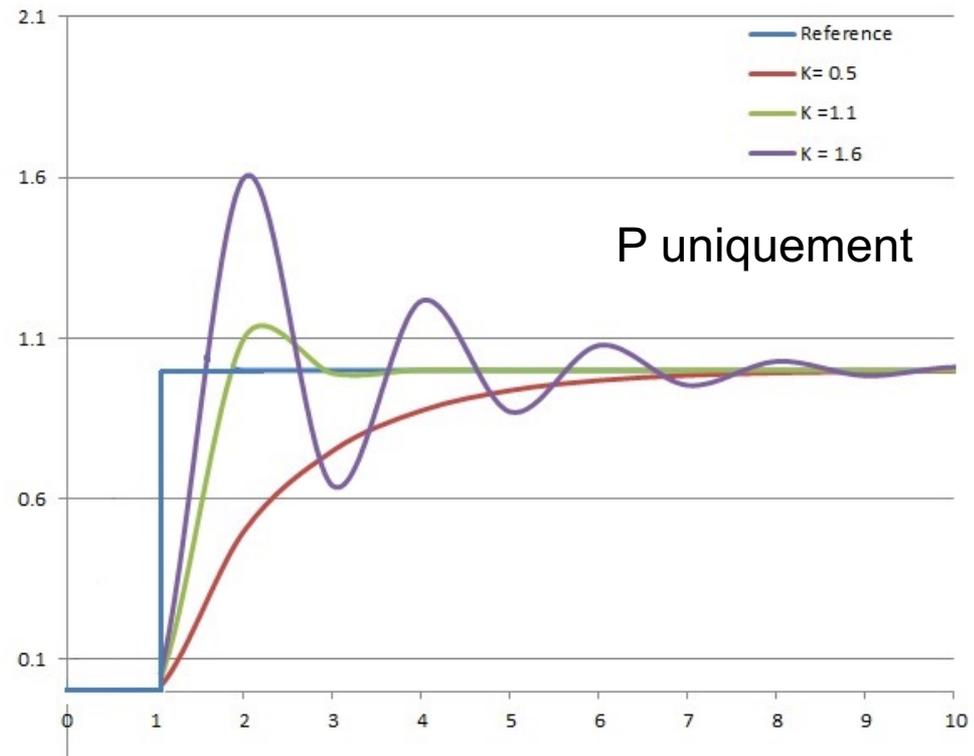
- Commande proportionnelle a
 - erreur
 - intégrale de l'erreur
 - dérivée de l'erreur



Réglage de PID

– Influence de P, I et D

	Précision	Stabilité	Rapidité
P	↗	↘	↗
I	↗	↘	↘
D	↘	↗	↗



- Réglage des paramètres : méthode de Ziegler-Nichols
 - Régler $P=I=D=0$
 - Augmenter P jusqu'à avoir des oscillations régulières
 - > K_u , fréquence des oscillations T_u

Contrôleur PID

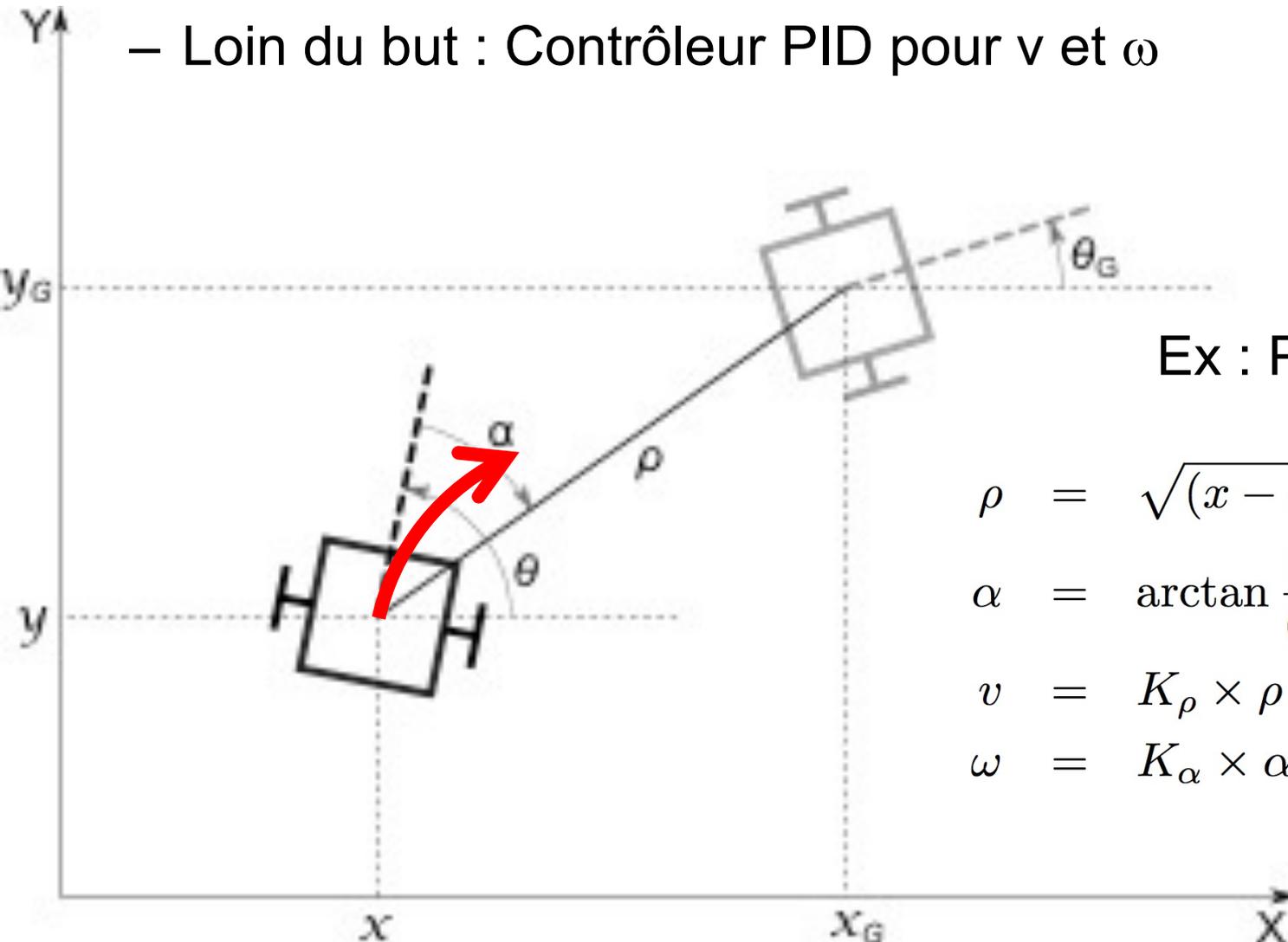
- Réglage des paramètres : méthode de Ziegler-Nichols
 - À partir de K_u et T_u , choisir:

Méthode de Ziegler-Nichols ¹			
Type de contrôle	K_p	T_i	T_d
P	$0.5K_u$	-	-
PI	$0.45K_u$	$T_u/1.2$	-
PD	$0.8K_u$	-	$T_u/8$
PID^2	$0.6K_u$	$T_u/2$	$T_u/8$

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right)$$

Atteindre une position

- Loin du but : Contrôleur PID pour v et ω



Ex : P uniquement

$$\rho = \sqrt{(x - x_G)^2 + (y - y_G)^2}$$

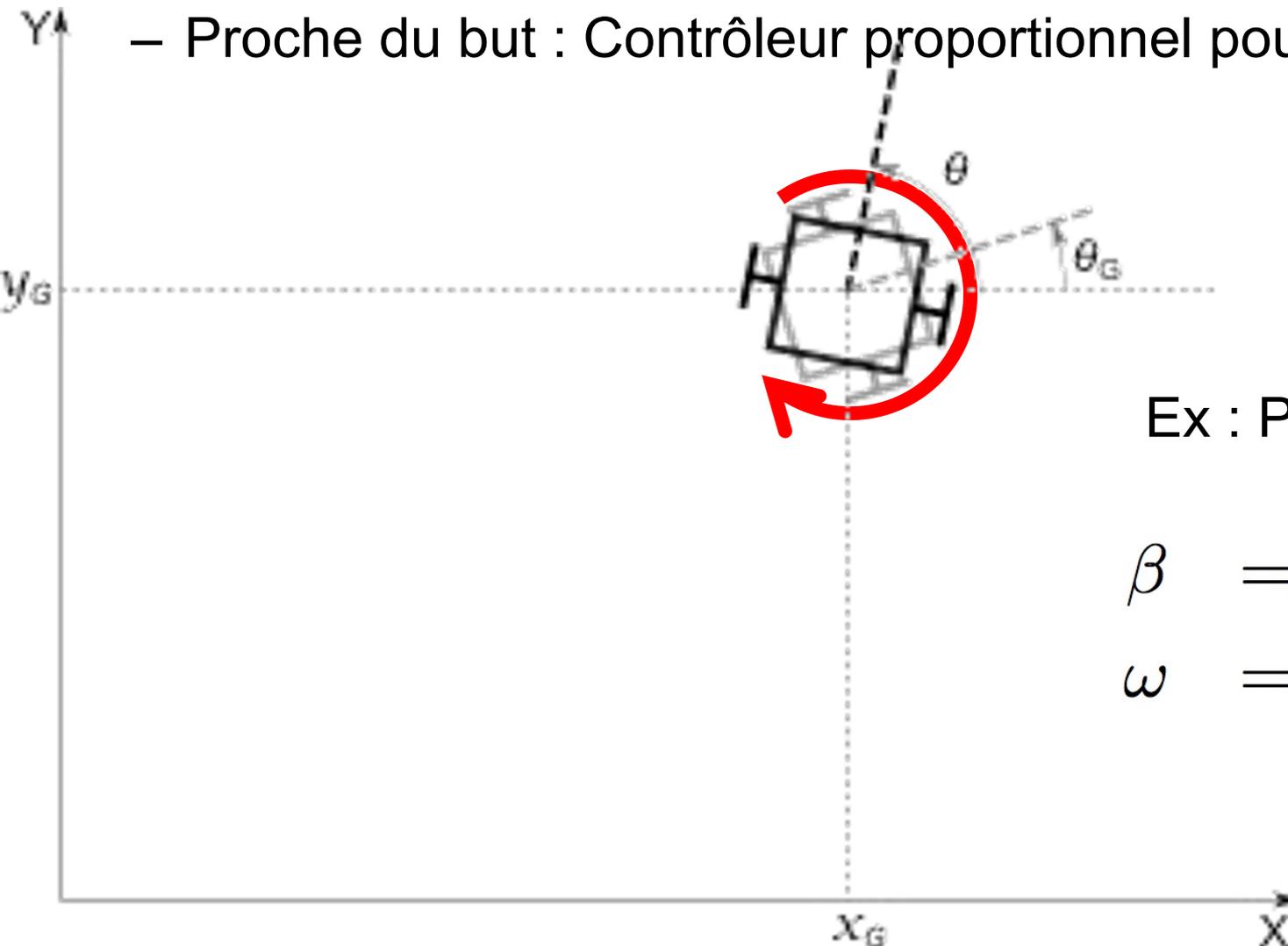
$$\alpha = \arctan \frac{(y - y_G)}{(x - x_G)} - \theta$$

$$v = K_\rho \times \rho$$

$$\omega = K_\alpha \times \alpha$$

Atteindre une position

- Proche du but : Contrôleur proportionnel pour ω



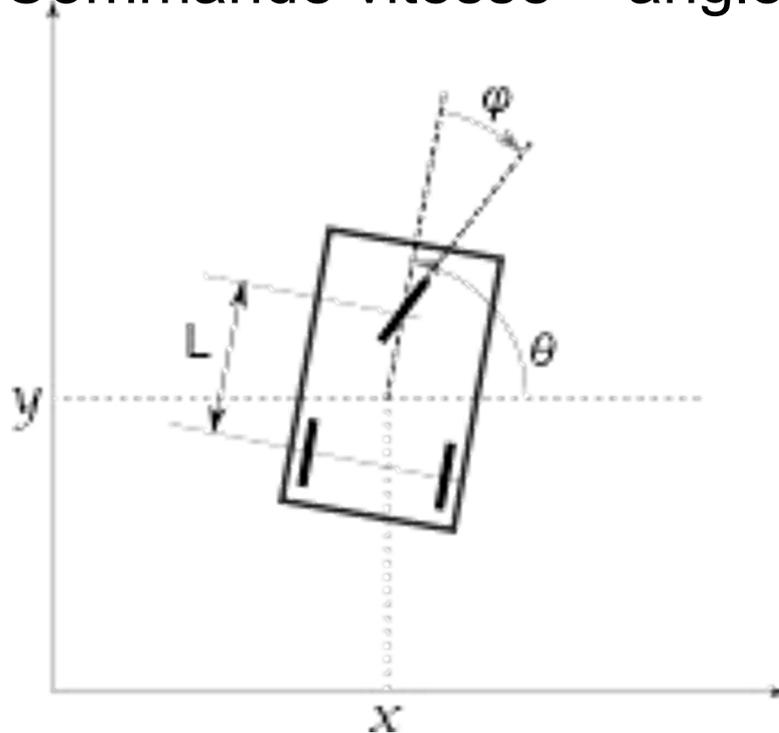
Ex : P uniquement

$$\beta = \theta - \theta_G$$

$$\omega = K_\beta \times \beta$$

Modèle tricycle (ou bicyclette)

- Commande vitesse + angle de roue

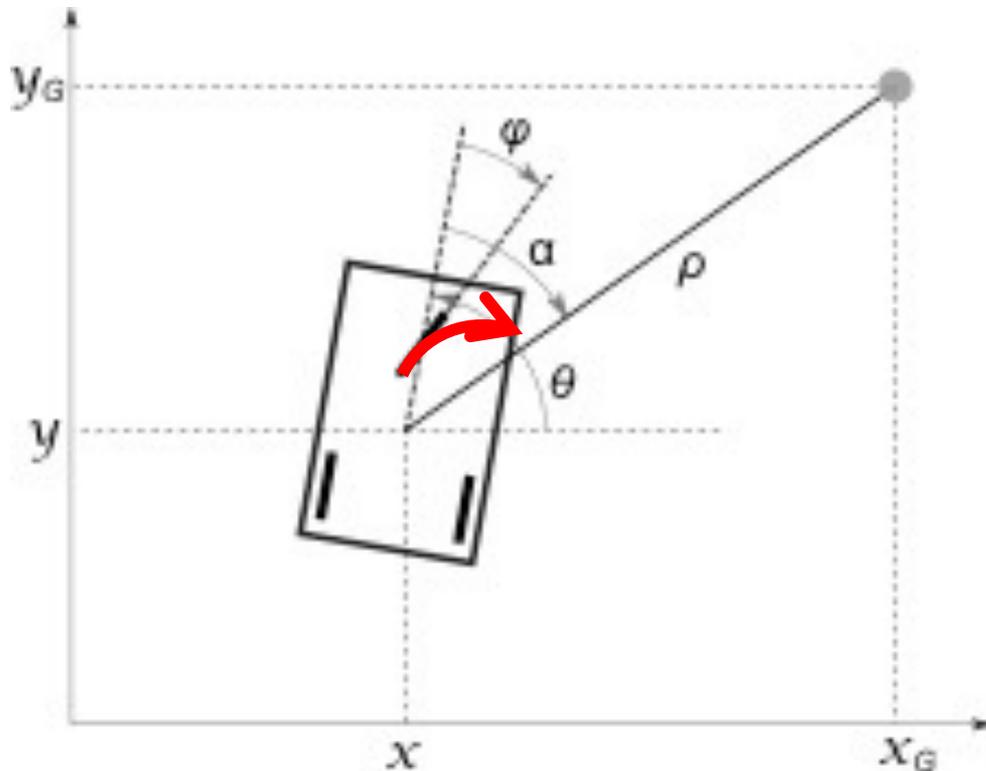


$$\begin{aligned}\dot{x} &= v \times \cos(\theta) \\ \dot{y} &= v \times \sin(\theta) \\ \dot{\theta} &= \frac{v}{L} \times \tan(\varphi)\end{aligned}$$

- Modèle non holonome
- Déplacements contraints et commande plus complexe

Atteindre un point (orientation libre)

- Simple PID sur la vitesse et la direction



Ex : P uniquement

$$\rho = \sqrt{(x - x_G)^2 + (y - y_G)^2}$$

$$\alpha = \arctan \frac{(y - y_G)}{(x - x_G)} - \theta$$

$$v = K_\rho \times \rho$$

$$\varphi = K_\alpha \times \alpha$$

Suivi de chemin (trajectoire sans vitesse)

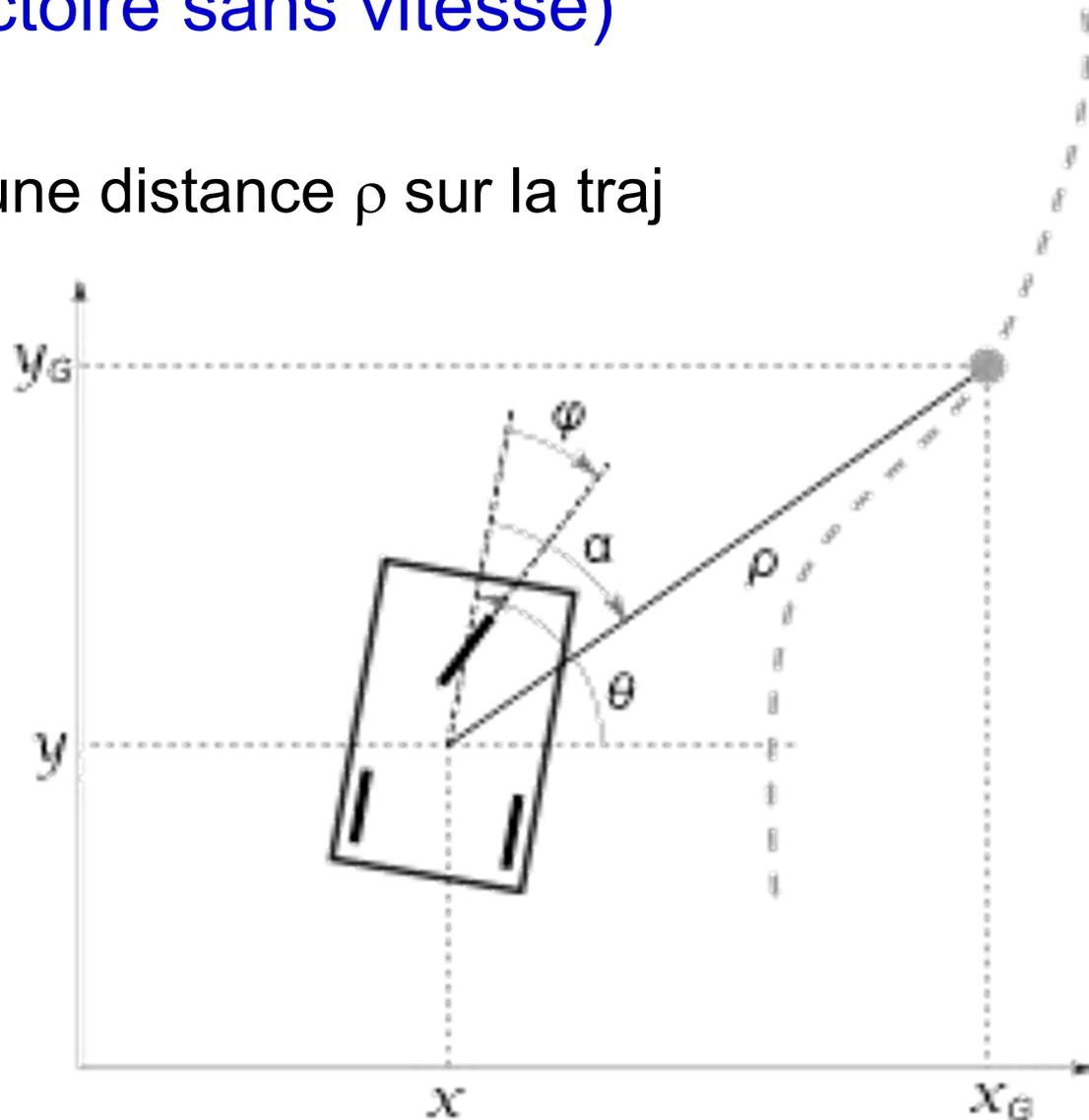
- Modèle de poursuite
- Chercher le point à une distance ρ sur la traj
- PID sur la direction

Ex : P uniquement

$$\alpha = \arctan \frac{(y - y_G)}{(x - x_G)} - \theta$$

$$v = v_0$$

$$\varphi = K_\alpha \times \alpha$$



Suivi de chemin (trajectoire avec vitesse)

- Modèle de poursuite
- PID sur la vitesse/direction avec but mobile

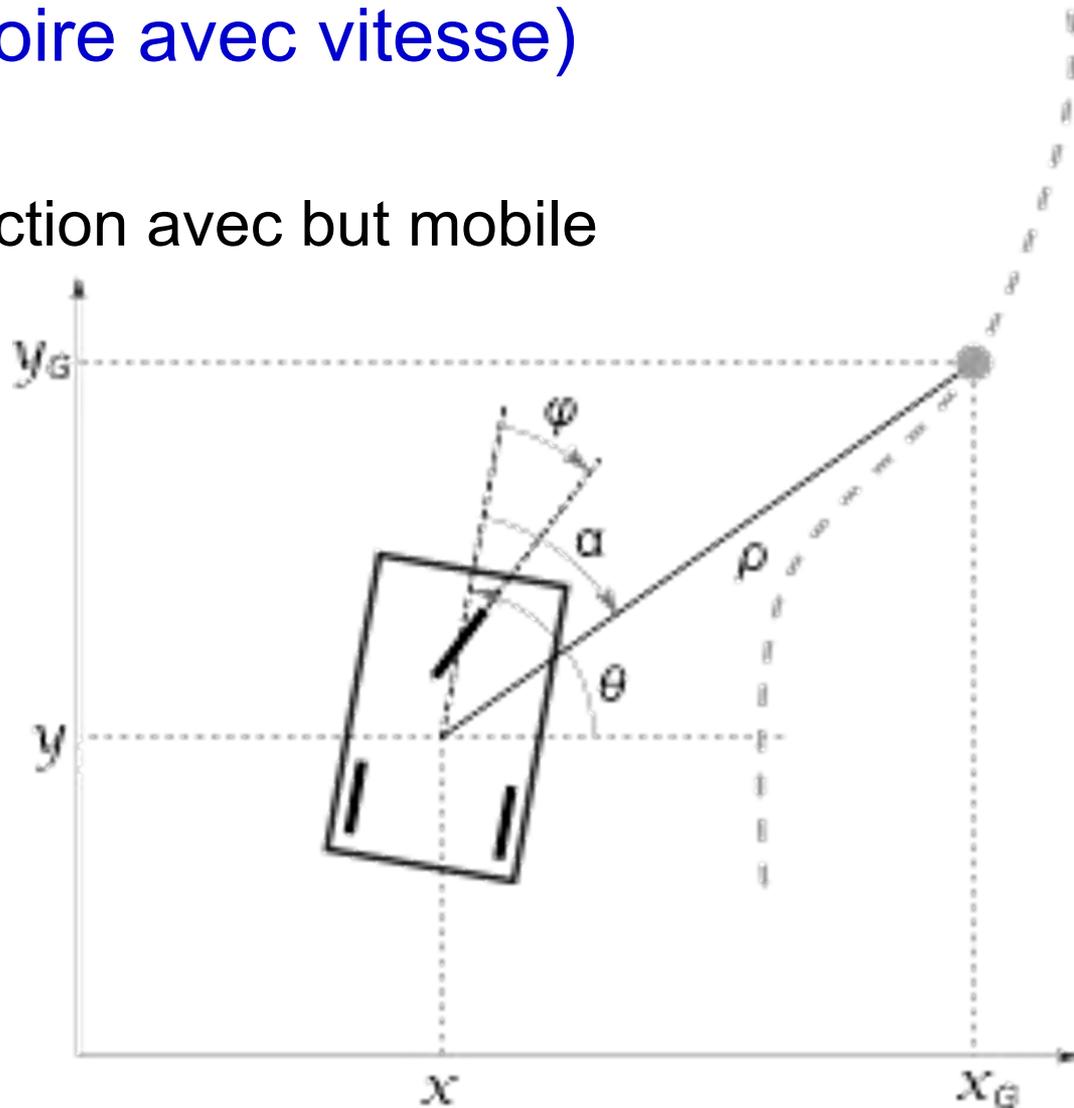
Ex : P uniquement

$$\rho = \sqrt{(x - x_G)^2 + (y - y_G)^2}$$

$$\alpha = \arctan \frac{(y - y_G)}{(x - x_G)} - \theta$$

$$v = K_\rho \times \rho$$

$$\varphi = K_\alpha \times \alpha$$



Inspiré des abeilles

- Catégorie 2 de Trullier
- Mémorisation du but : *Détection panoramique d'amers* (*snapshot*)

Cartwright, B. A., & Collett, T. S. (1983).
Landmark learning in bees. *Journal of comparative physiology*, 151(4), 521-543.

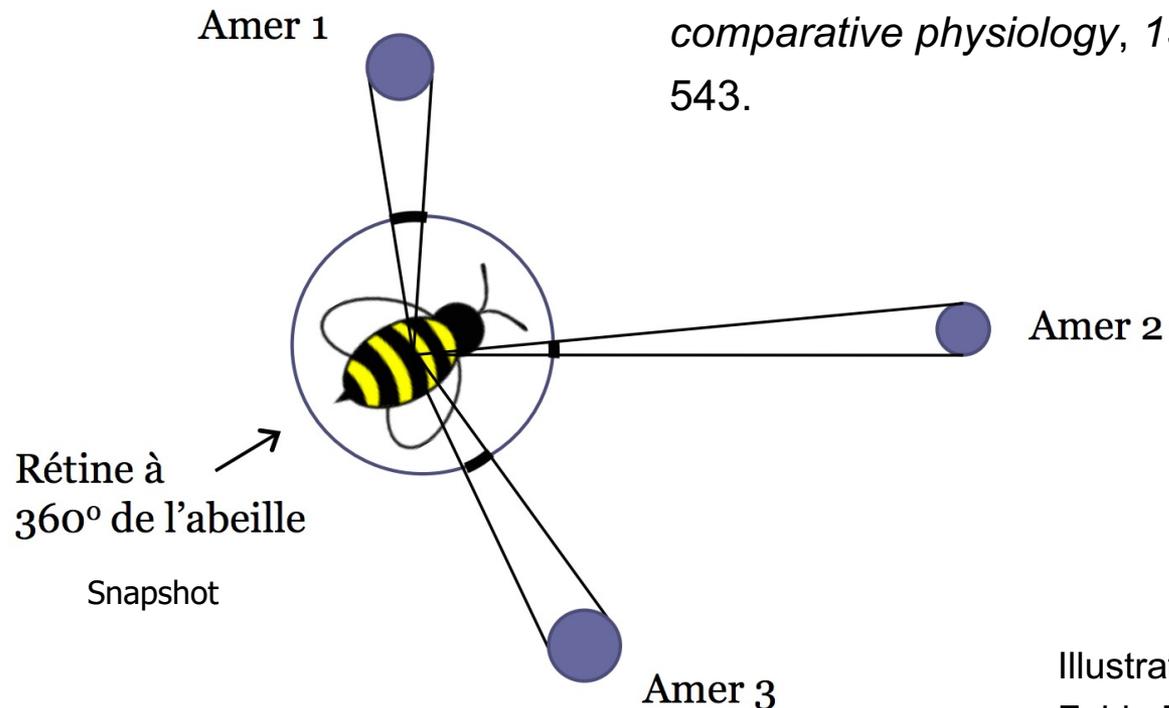
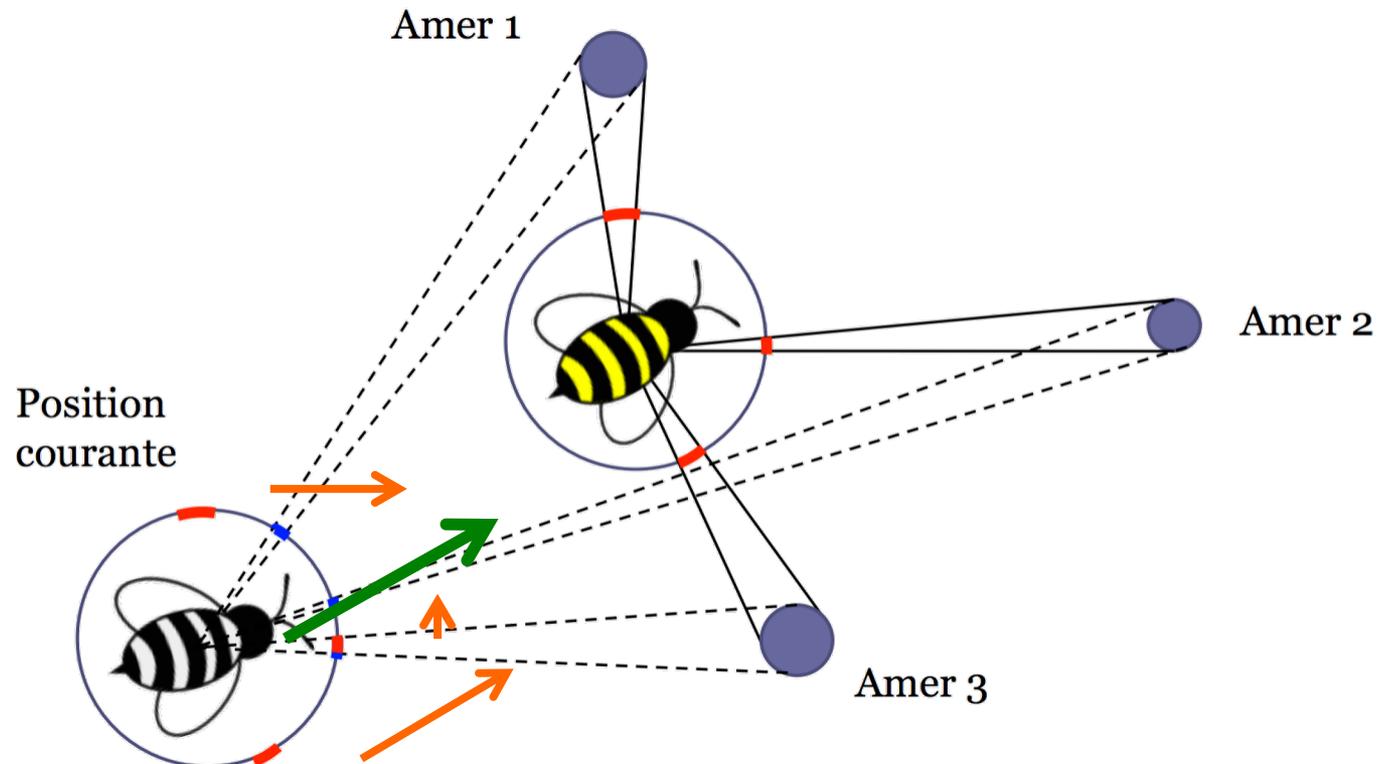


Illustration
Fabio Morbidi

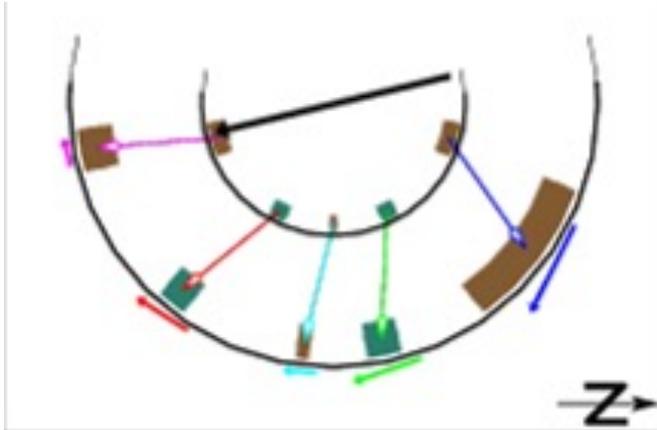
Déplacement vers le but :

- Association d'amers
- Estimation de la direction du but par sommation de contributions par amer

- **Association d'amers difficile,**
- **Nécessité de connaître l'orientation**



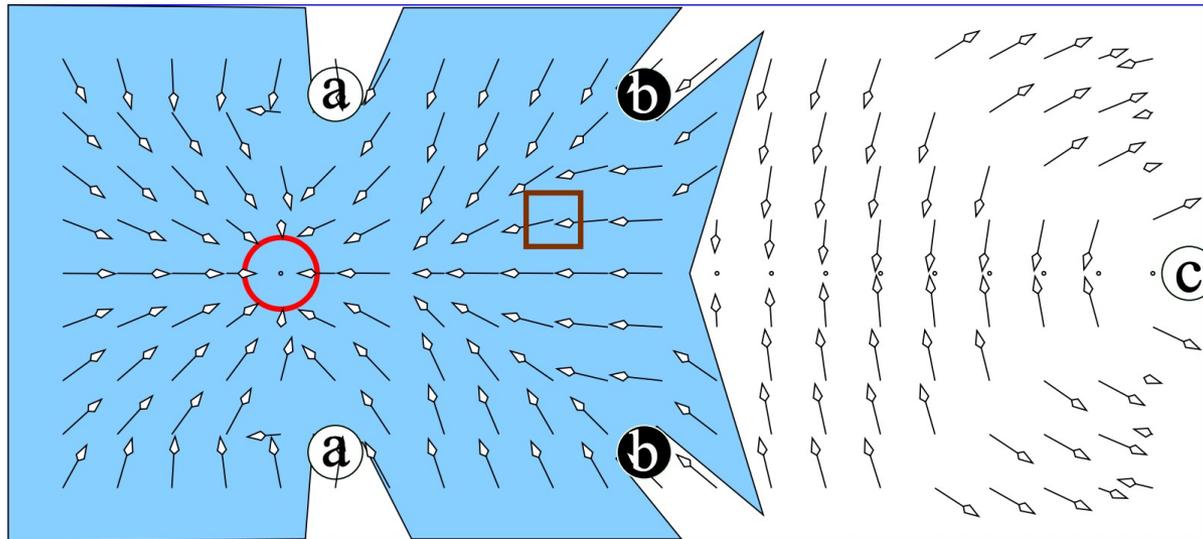
Utilisation de couleurs



- Améliore l'association d'amers
- Etend la zone de fonctionnement

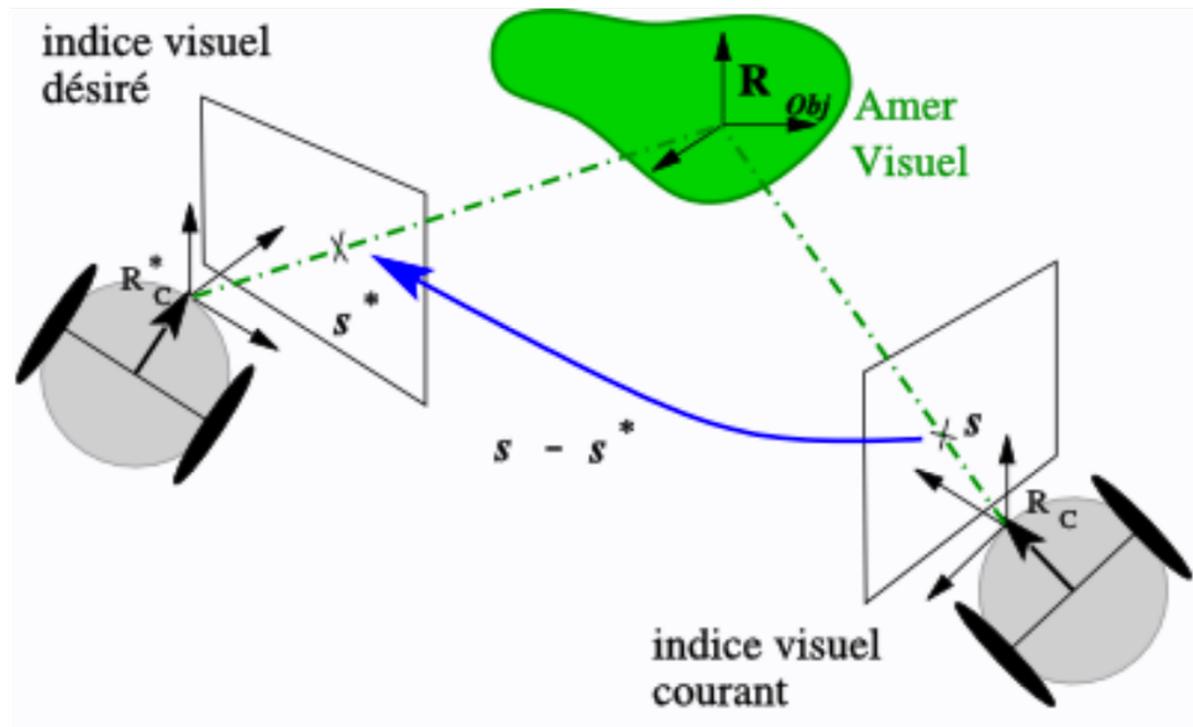
Gourichon, S., Meyer, J. A., & Pirim, P. (2002). Using coloured snapshots for short-range guidance in mobile robots. *International Journal of Robotics and Automation*, 17(4), 154-162.

Cette méthode reste locale



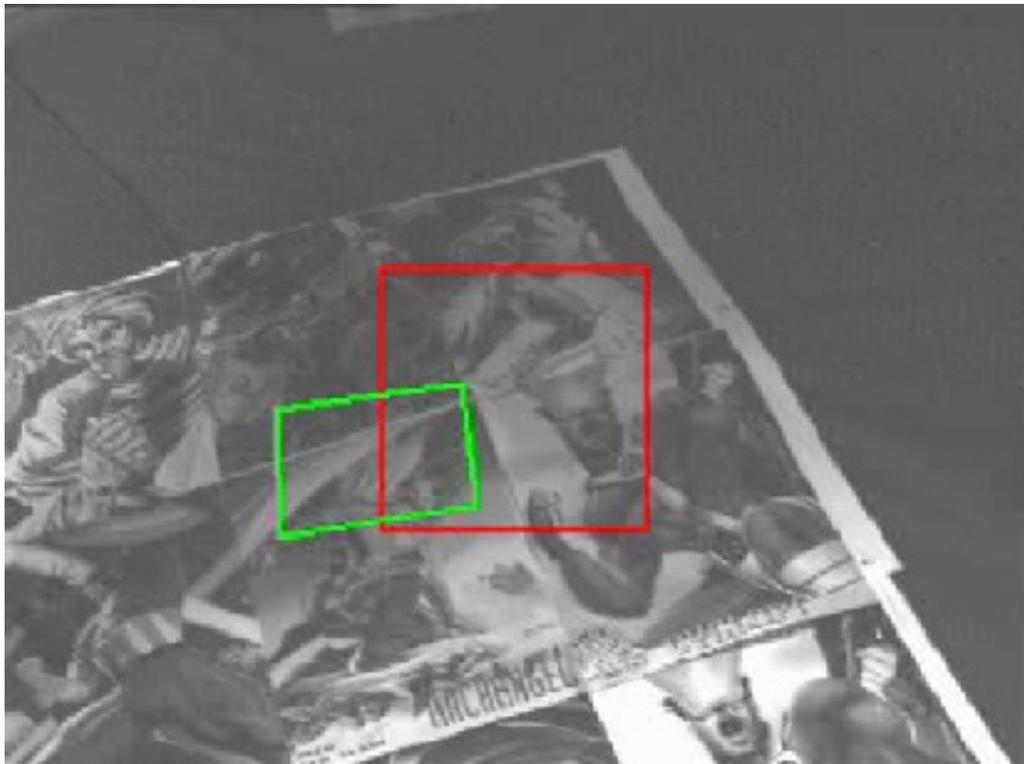
Contrôle d'un robot par la vision

- Asservissement de la commande du robot directement en fonction de l'erreur de vision
- Nombreux choix possible d'indices visuels
- Contrôle de robots complexes (bras manipulateurs)

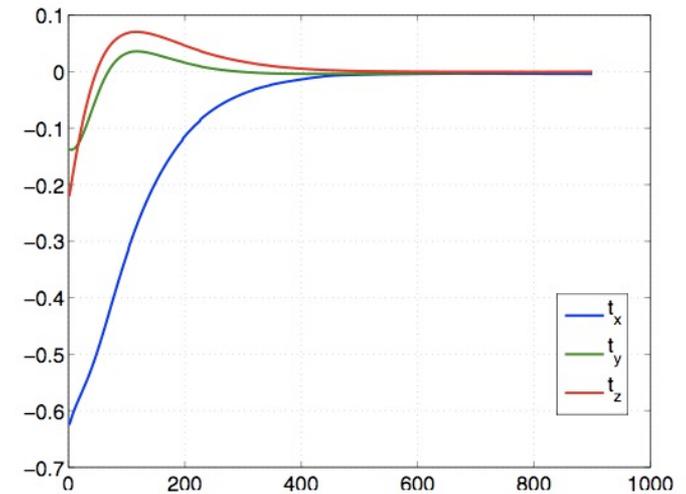


Contrôle d'un robot par la vision

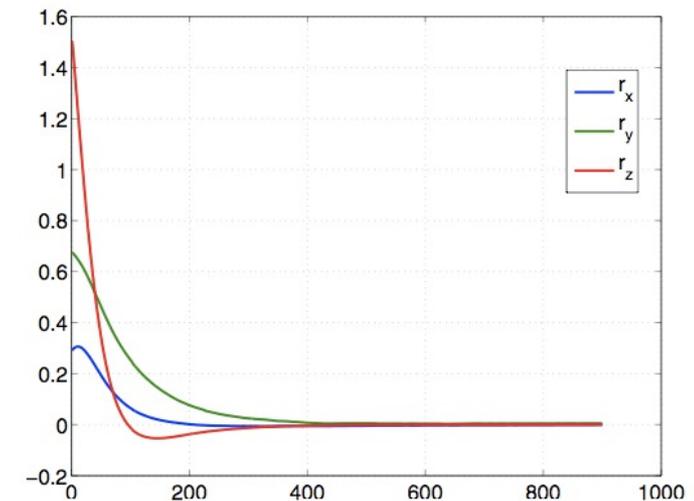
- Ex: Choix d'une image à atteindre
- Contrôle d'un bras portant une caméra



Asservissement visuel



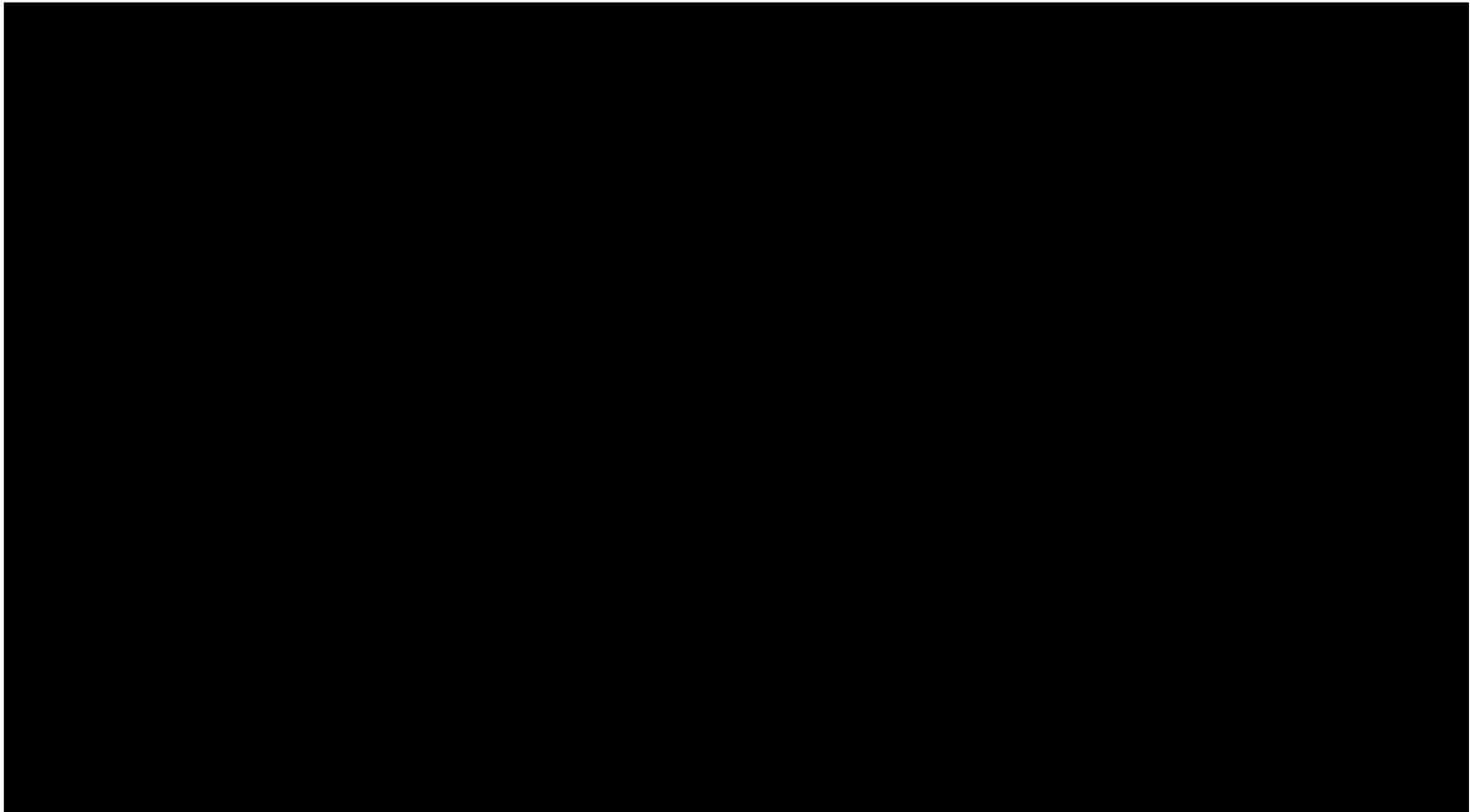
(e) Translation error function



(f) Rotation error

Application en robotique mobile

- Asservissement visuel pour le contrôle local
- Enchaînement de contrôles sur une séquence d'images

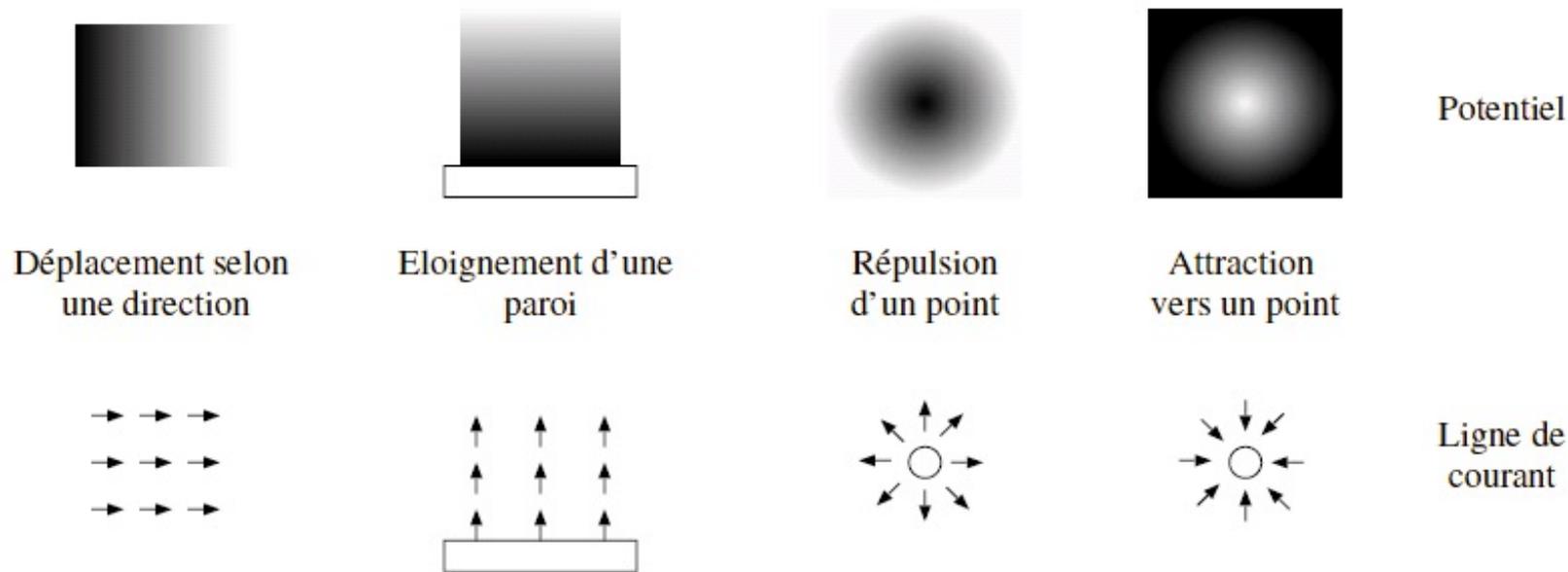


Visual Servoing on Cycab – INRIA Rennes

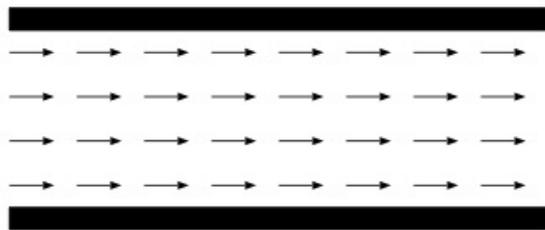
Comportement réactif : évitement d'obstacles

Méthode de combinaison de comportements

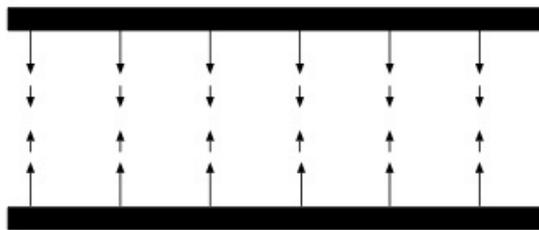
- Chaque objet de l'environnement est associé à un champ de forces
- Les forces peuvent varier avec la distance, peuvent agir sur une étendue limitée
- Le robot suit le champ de forces



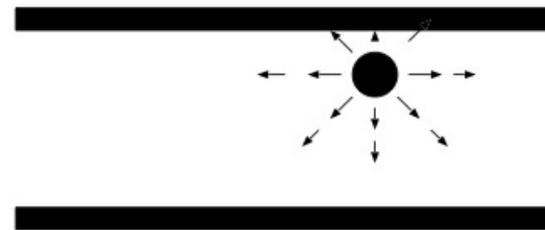
La combinaison donne la direction à prendre en chaque point de l'espace



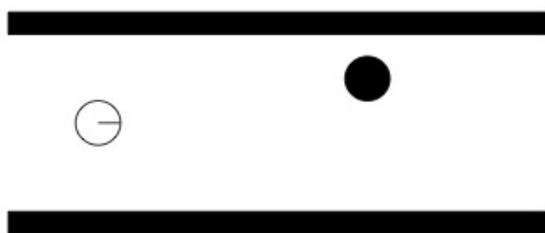
Suivi d'un couloir



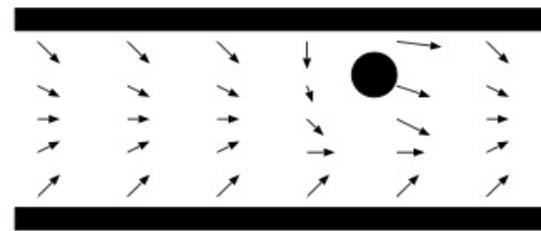
Centrage dans un couloir



Évitement d'un obstacle



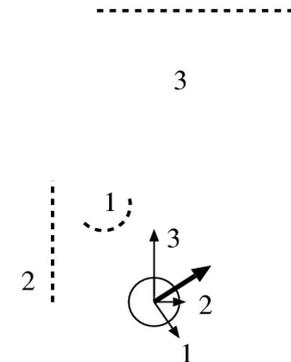
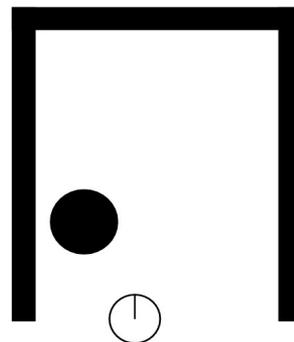
Environnement



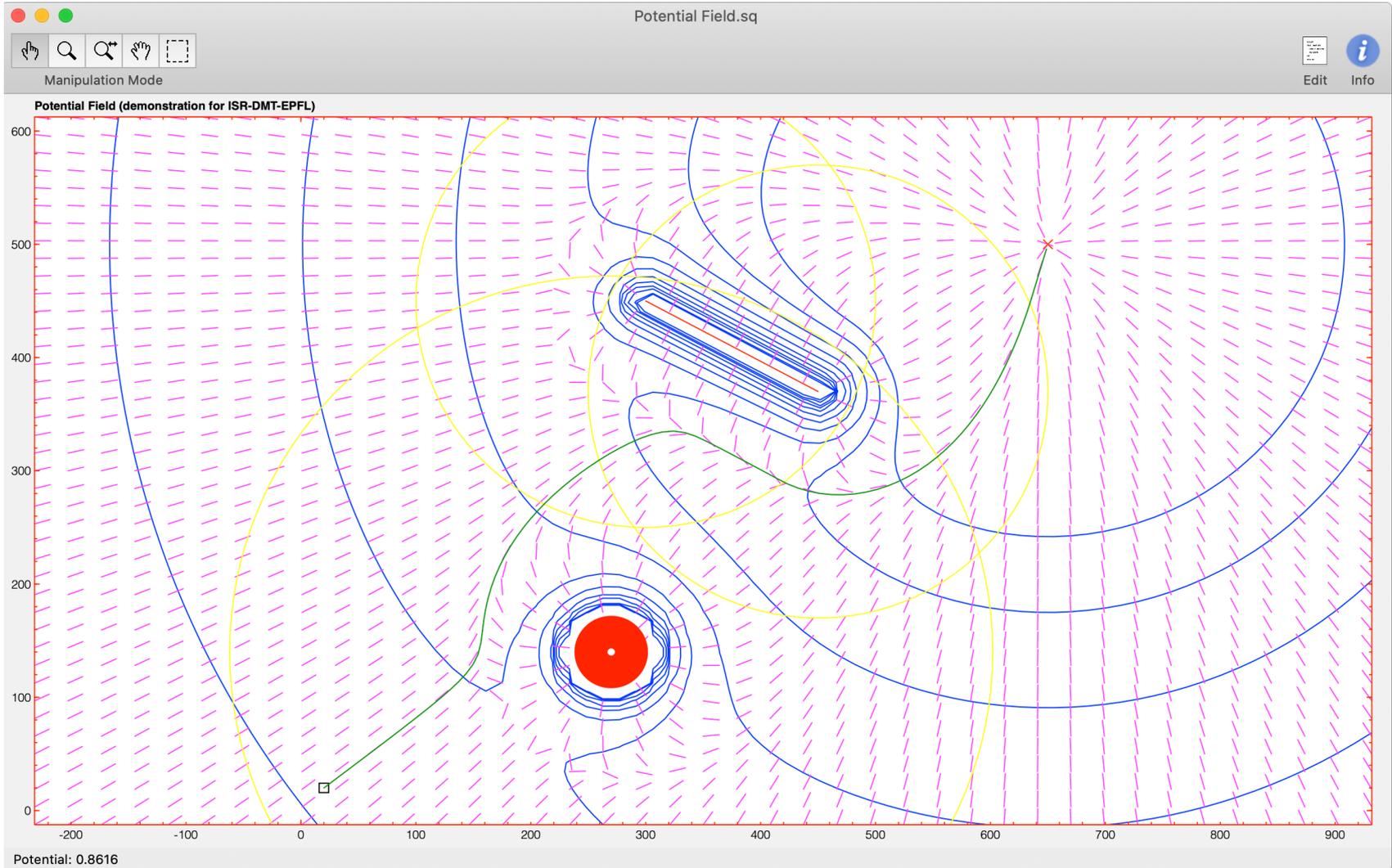
Potentiel résultant

Les calculs possibles dans l'espace relatif au robot

- pas de calcul global du champ de force

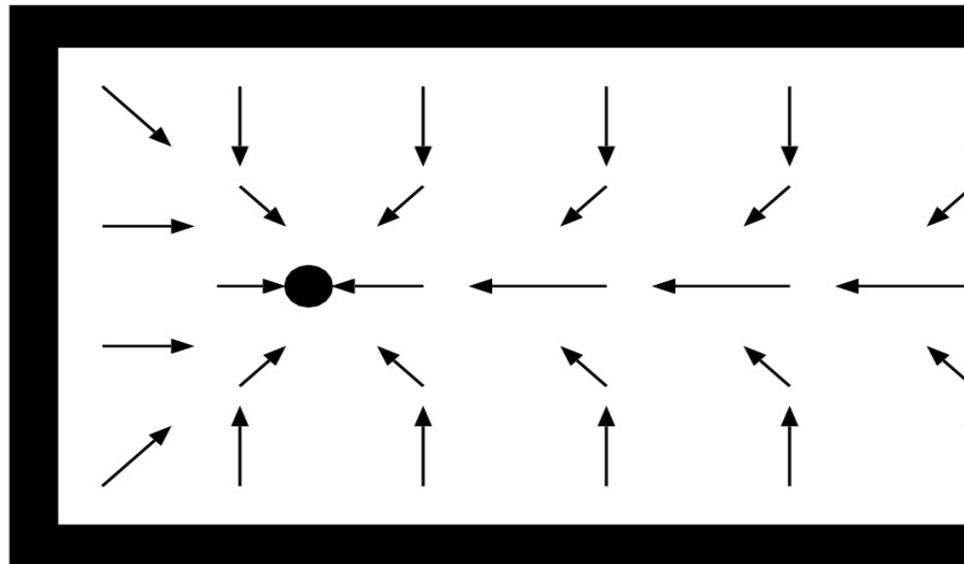


Démo



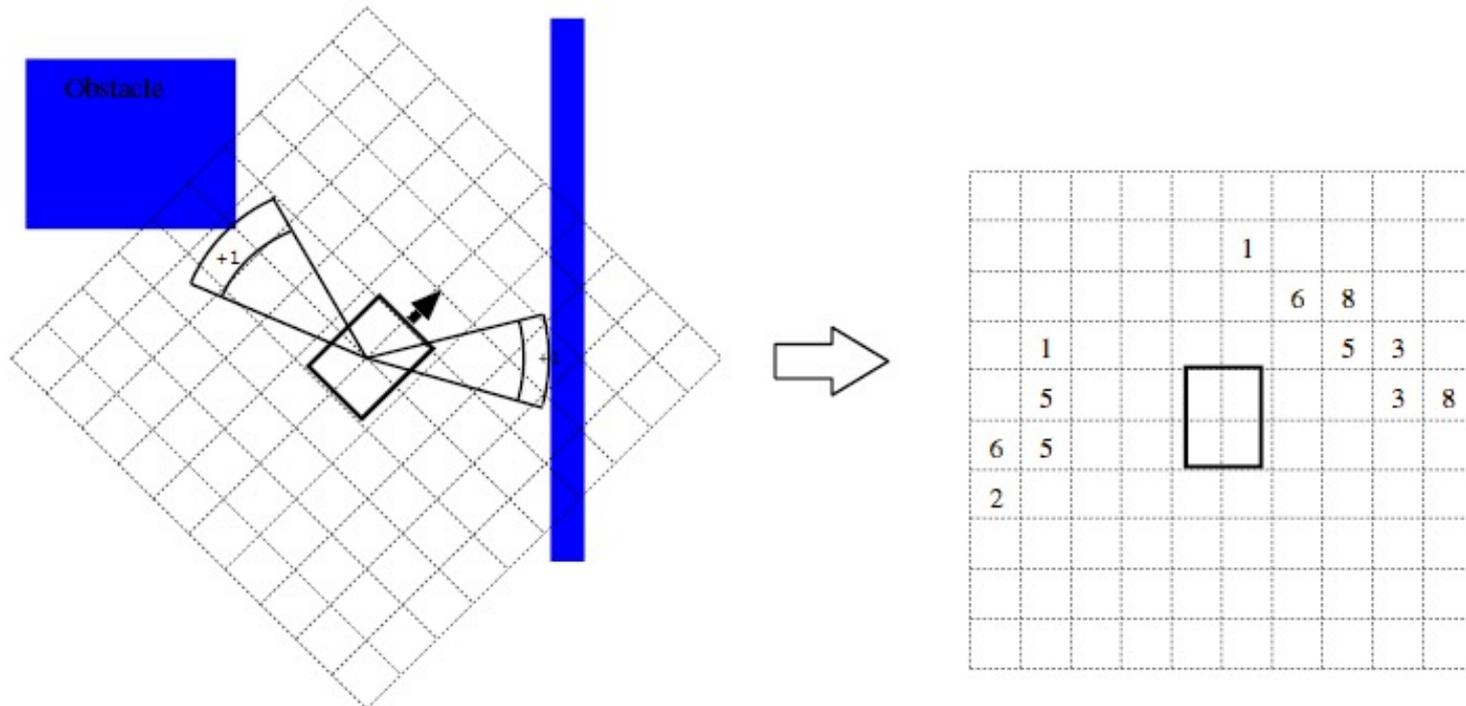
Problèmes de minimum locaux

- Ajout de valeur aléatoire
- Comportement spécifique (suivi de mur...)
- Fonction harmonique
- Mémorisation et évitement des zones déjà traversées

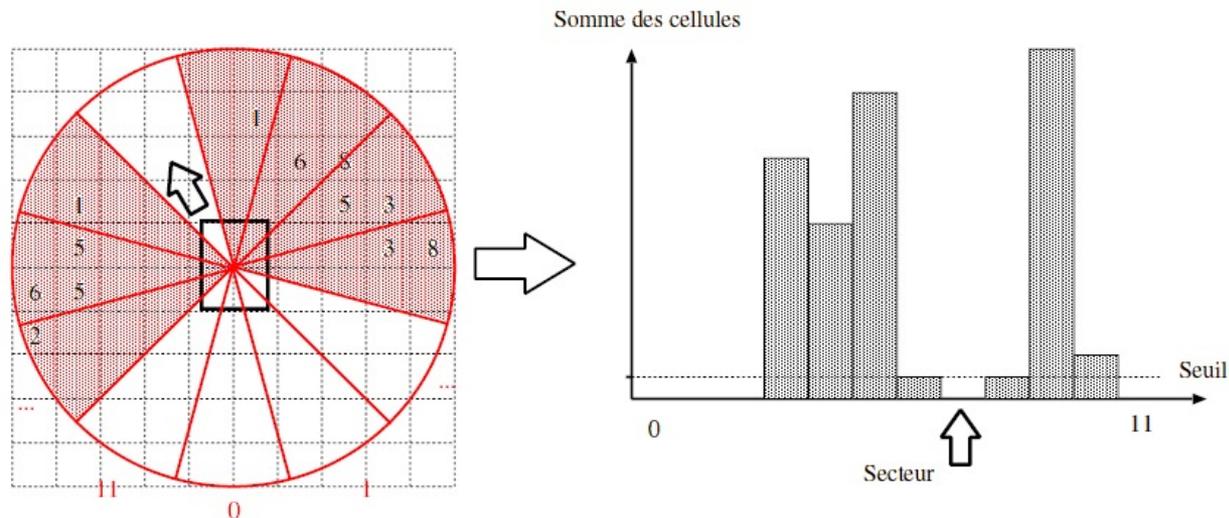


Utilise une représentation locale de l'environnement (grille d'occupation histogrammique)

- Utilisable avec des capteurs peu précis (sonars)



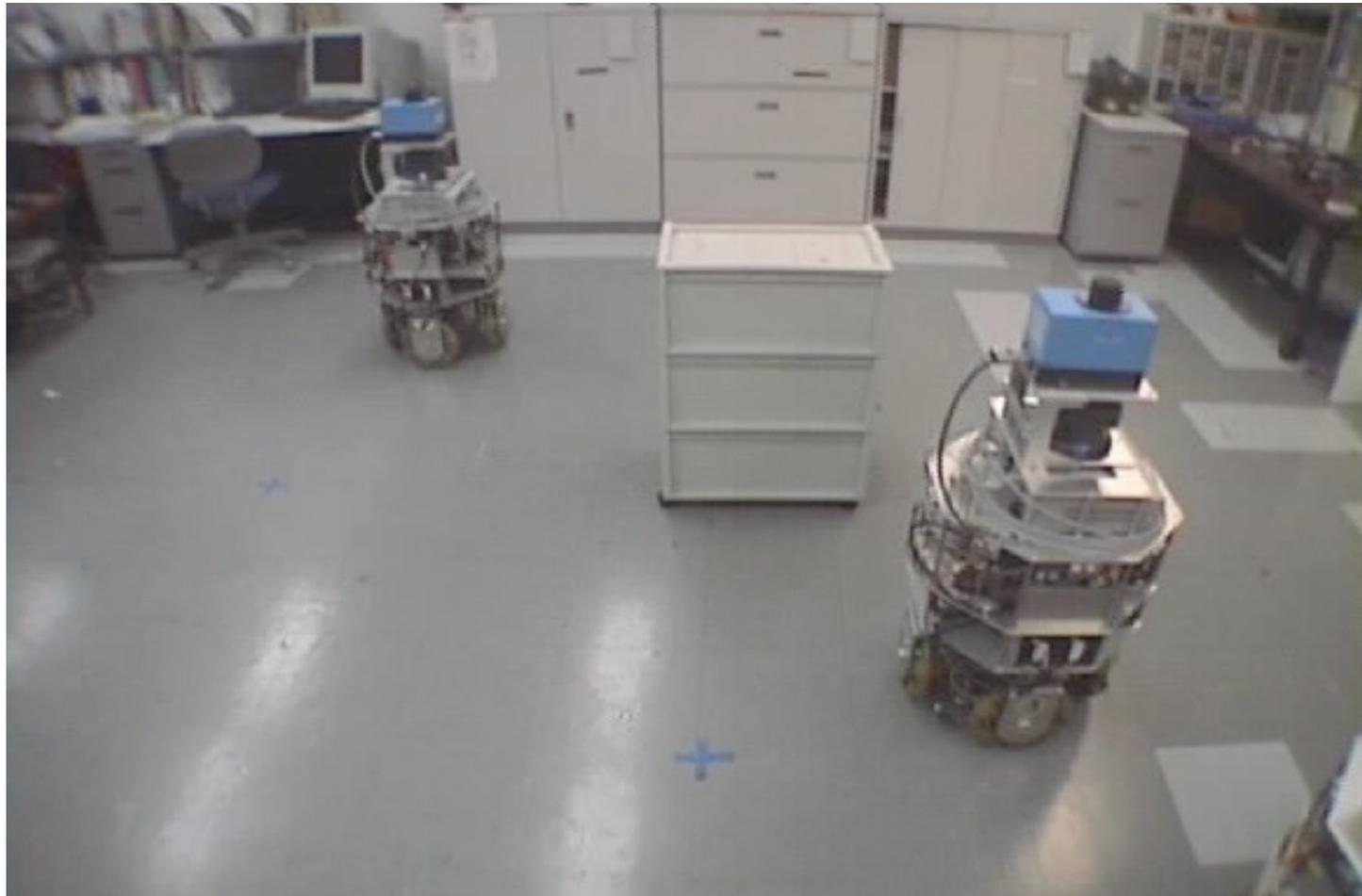
- Construction d'un histogramme d'occupation par secteur angulaire autour du robot
- Choix de la direction parmi les directions libres



Méthode très rapide

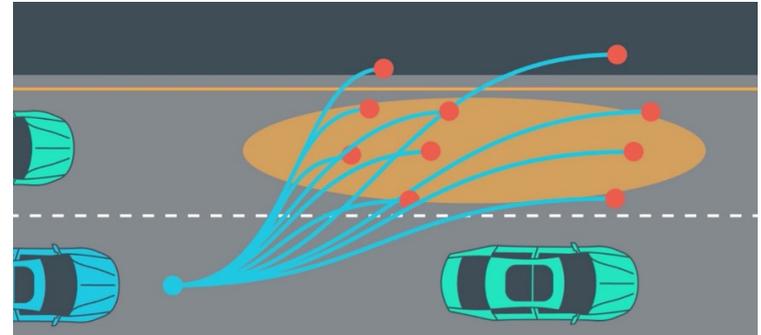
Permet des déplacements à vitesse élevée

Améliorations : - régler la vitesse en fonction de l'occupation de l'espace
- utiliser la dynamique du robot



Principe

- Paramétrer des trajectoires
 - Estimer des contraintes
 - Recherche des paramètres qui respectent différentes contraintes :
- Évitement d'obstacles (contrainte dure)
 - Direction préférentielle (souple)
 - Distance min aux obstacles (souple)
 - ...



Implémenté dans ROS :
package DWA

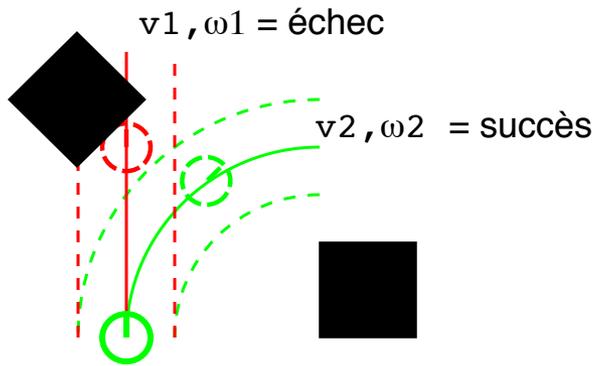
Application robot unicycle

- Tracé des contraintes sur un graphe des vitesses
- Recherche de vitesses (v, ω) qui respectent les contraintes

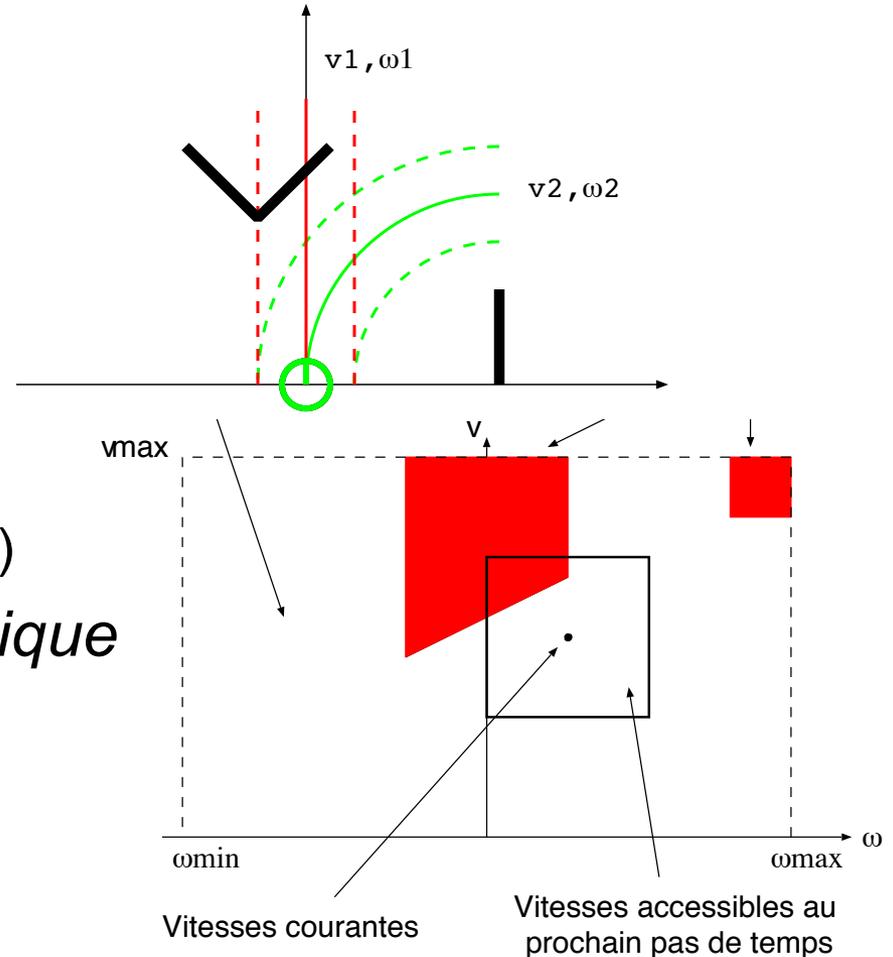
Contrainte d'évitement d'obstacles :

(estimé à un pas de temps donné dans le futur) :

Environnement réel



Perceptions du robot



Report dans le graphe (v, ω)

Tracé de la *fenêtre dynamique*
(*accel max*)

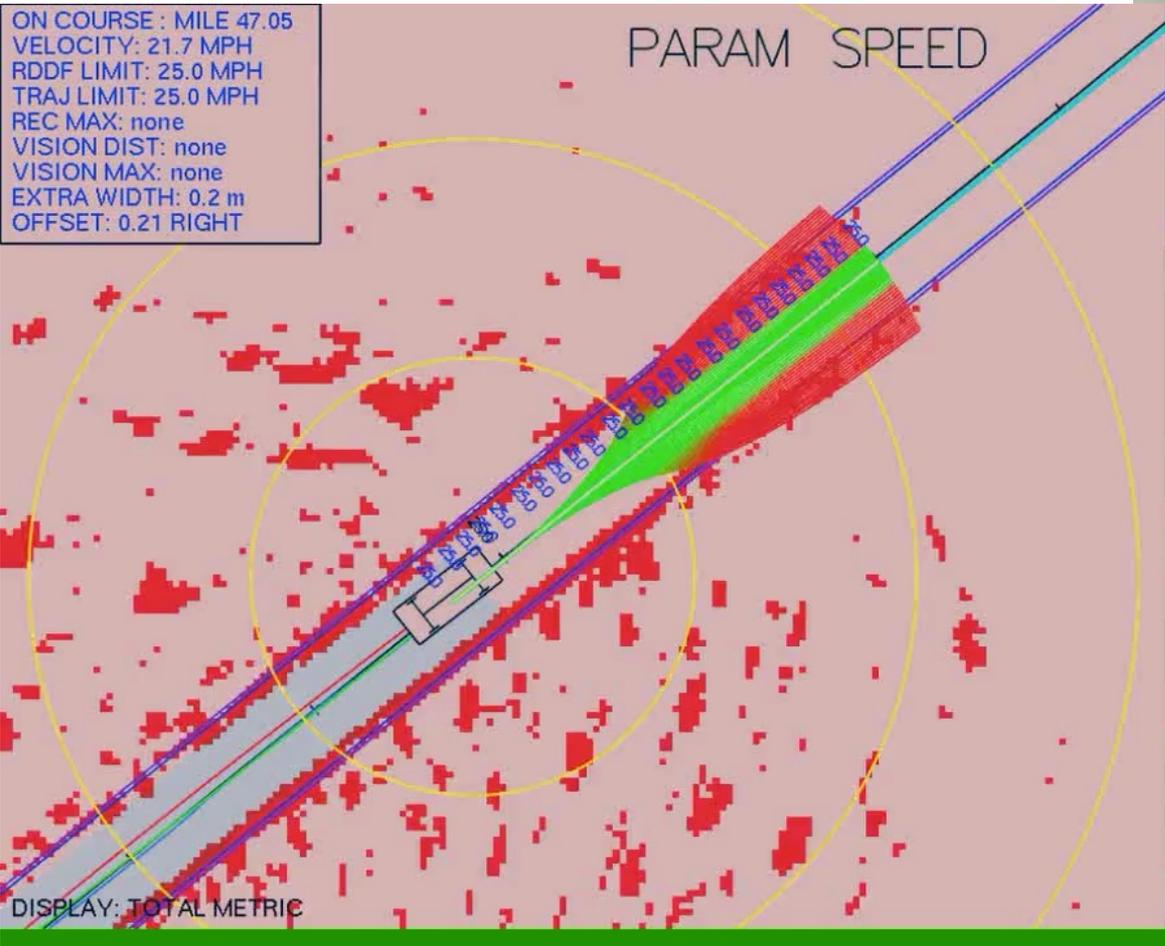
Exemple sur un véhicule autonome

- STANLEY, Stanford (2006)



ON COURSE : MILE 47.05
 VELOCITY: 21.7 MPH
 RDDF LIMIT: 25.0 MPH
 TRAJ LIMIT: 25.0 MPH
 REC MAX: none
 VISION DIST: none
 VISION MAX: none
 EXTRA WIDTH: 0.2 m
 OFFSET: 0.21 RIGHT

PARAM SPEED



DISPLAY: TOTAL METRIC

Apprentissage par renforcement

Apprentissage par l'interaction (essais/erreurs)

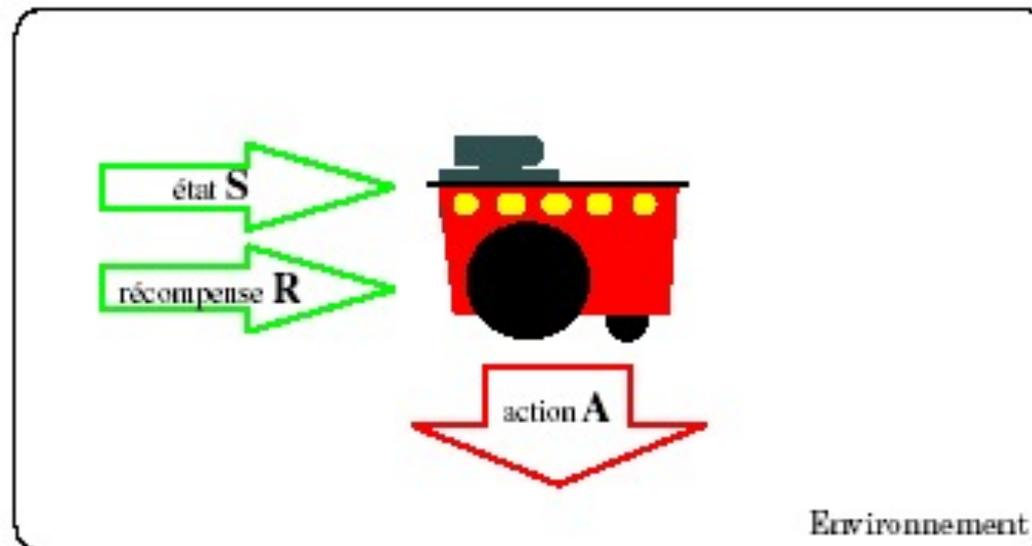
- ancrage dans l'environnement

Renforcement = « note » scalaire de la qualité du comportement

- Pas de connaissance de l'action « parfaite » :
apprentissage non supervisé

Cherche une stratégie pour maximiser la récompense

- A chaque état, on associe une action



Ensemble d'états (S)

Ensemble d'actions (A)

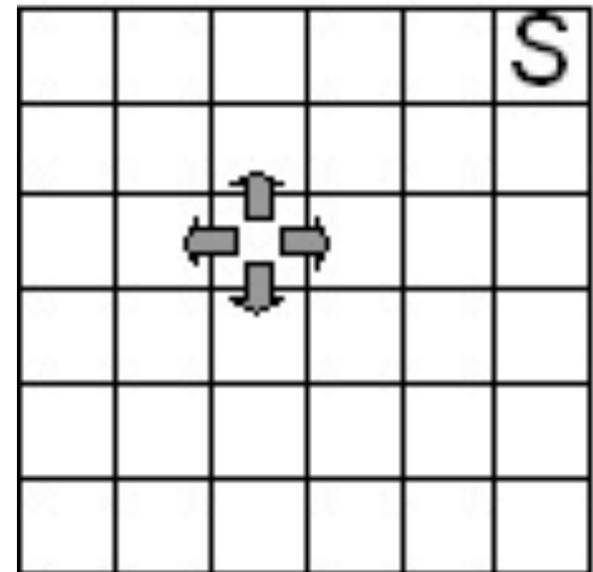
Fonction de Récompense (R) : $R(s,a,s')$

Environnement : Processus de décision markovien

- évolution de l'état et des récompenses
 - probabiliste
 - dépend des actions de l'agent

$$\mathcal{P}_{ss'}^a = P(s_{t+1} = s' | s_t = s, a_t = a)$$

$$\mathcal{R}_{ss'}^a = E(r_{t+1} | s_t = s, a_t = a, s_{t+1} = s')$$



But de l'apprentissage :

- trouver une politique $\pi : \{\mathcal{S}, \mathcal{A}\} \rightarrow [0, 1]$
- $\pi(s,a)$: probabilité de choisir l'action a dans l'état s

Maximisant la récompense à long terme :

$$R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

(ou la récompense par épisode)

Fonction de valeur (politique donnée):

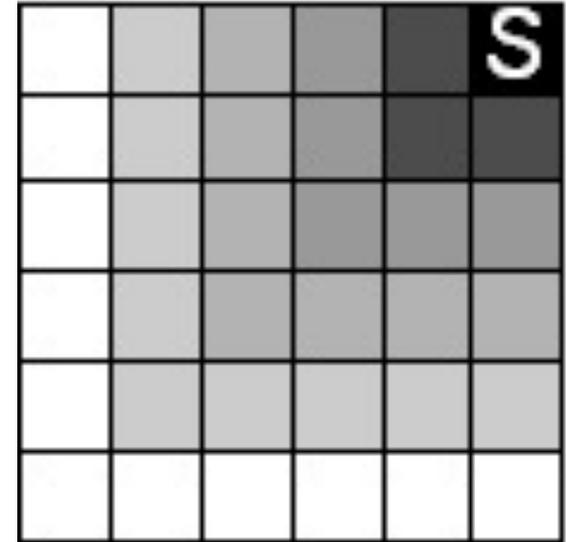
$$V^\pi(s) = E_\pi(R_t | s_t = s)$$

Autre forme :

$$Q^\pi(s, a) = E_\pi(R_t | s_t = s, a_t = a)$$

avec

$$V^\pi(s) = \sum_a \pi(s, a) Q^\pi(s, a)$$



caractérise la qualité d'une politique

Equation de d'optimalité de Bellman :

$$Q(s_t, a_t) = r_{t+1} + \gamma \max_a Q(s_{t+1}, a)$$

Un algorithme d'AR : Q Learning

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \right]$$

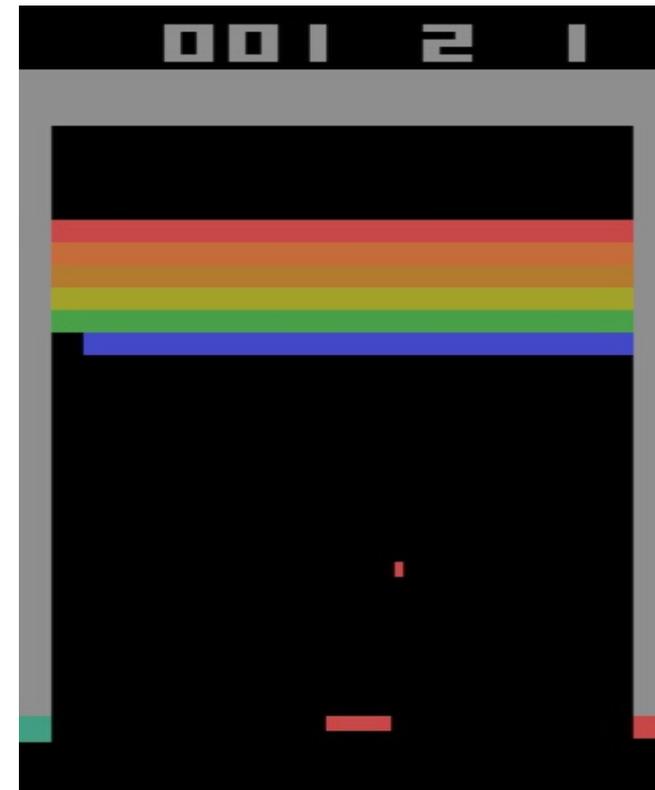
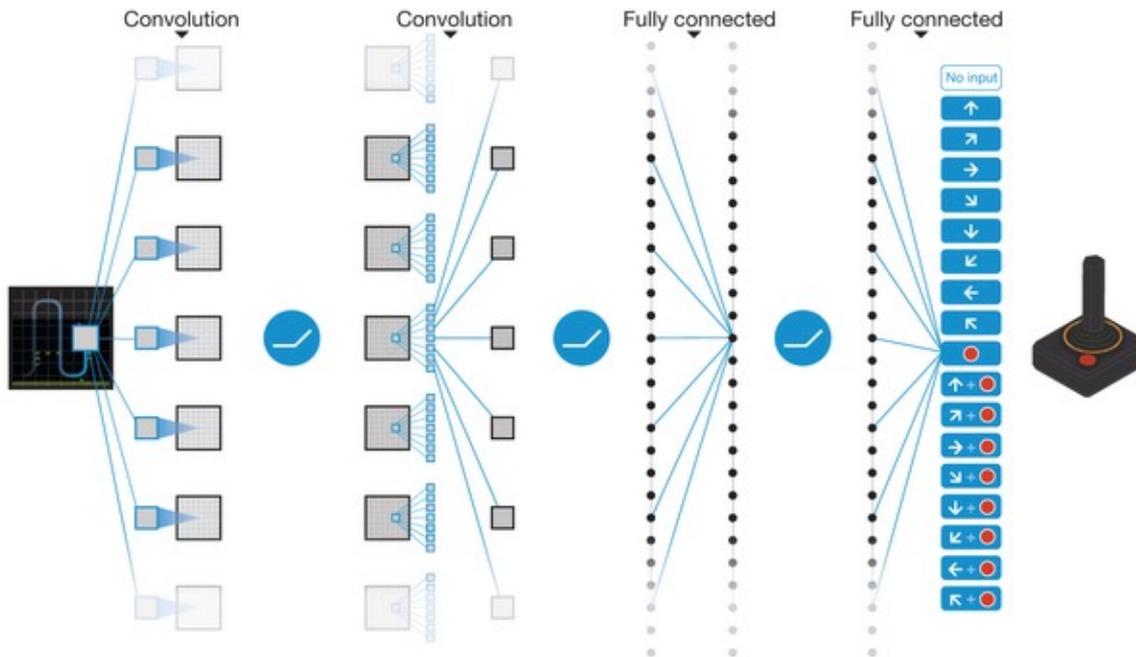
- Apprentissage *hors politique*
- Très utile en pratique

Algorithme:

- Initialiser $Q(s,a)$ aléatoirement
- Répéter
 - Déterminer état s_t
 - Choisir action a (ϵ -greedy)
 - Exécuter a , déterminer s_{t+1} et r
 - Evaluer $Q(s_t, a)$

Deep Learning pour l'apprentissage par renforcement

- Utilisation d'un réseau de neurones profond pour $Q(s,a)$
- Possibilité de prendre des images directement en entrée
- Apprend simplement en jouant et maximisant le score



Human-level Control through Deep Reinforcement Learning, Mnih et al., Nature 2015

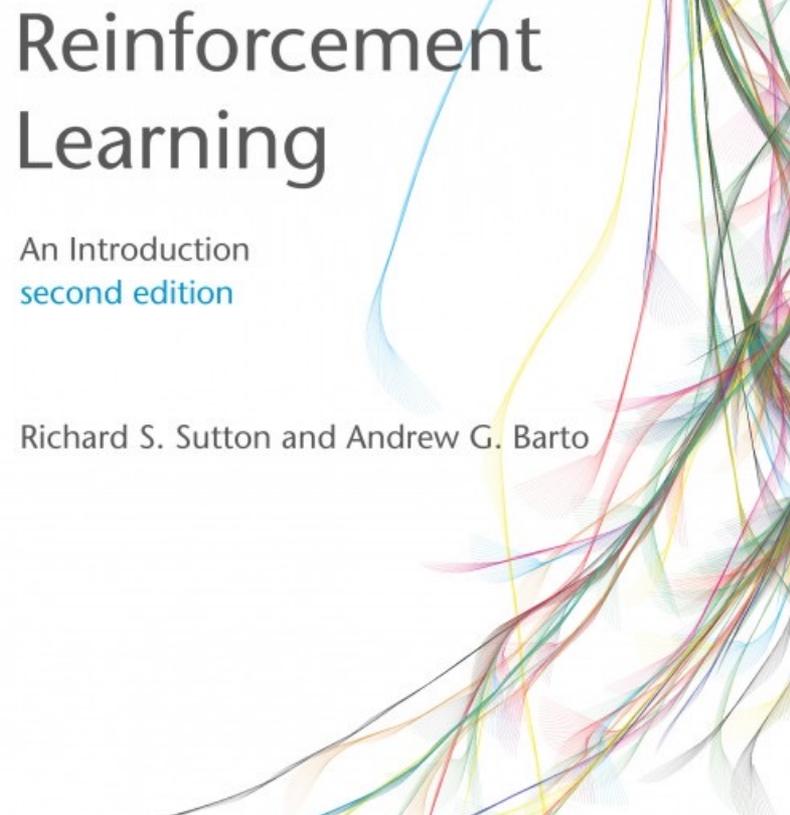
Exemple d'application à la conduite



Exemple d'application à la conduite



Kendall, A., et al. (2018). Learning to Drive in a Day. *arXiv:1807.00412*.



Reinforcement Learning

An Introduction
second edition

Richard S. Sutton and Andrew G. Barto

Reinforcement Learning: An Introduction

Richard S. Sutton, Andrew G. Barto

(disponible sur le web)

Navigation réactive - En résumé

- Stratégies de navigation utilisant uniquement les données capteurs pour prendre la décision : contrôle vers un but, évitement d'obstacles, ...
- Méthodes classiques :
 - contrôleurs PID pour atteindre un but/suivre une trajectoire
 - Asservissement visuel
 - Méthode de la fenêtre dynamique
- Apprentissage par renforcement : apprentissage de contrôleur réactif par essai/erreurs