

# Localisation et navigation de robots

UPJV, Département EEA

M2 3EA, EC32, parcours RoVA

Année Universitaire 2021-2022

**Fabio MORBIDI**

Laboratoire MIS

Équipe Perception Robotique

E-mail: [fabio.morbidi@u-picardie.fr](mailto:fabio.morbidi@u-picardie.fr)

**Mardi et mercredi 9h00-12h00**  
**salle CURI 305: CM & TD**  
**salle TP204: TP**



Electronique

Energie Electrique

Automatique



# Plan du chapitre

Stratégies de navigation

**Partie 1**

Architectures de contrôle

**Partie 2**

Navigation vers un but

**Partie 3**

Evitement d'obstacles

**Partie 4**

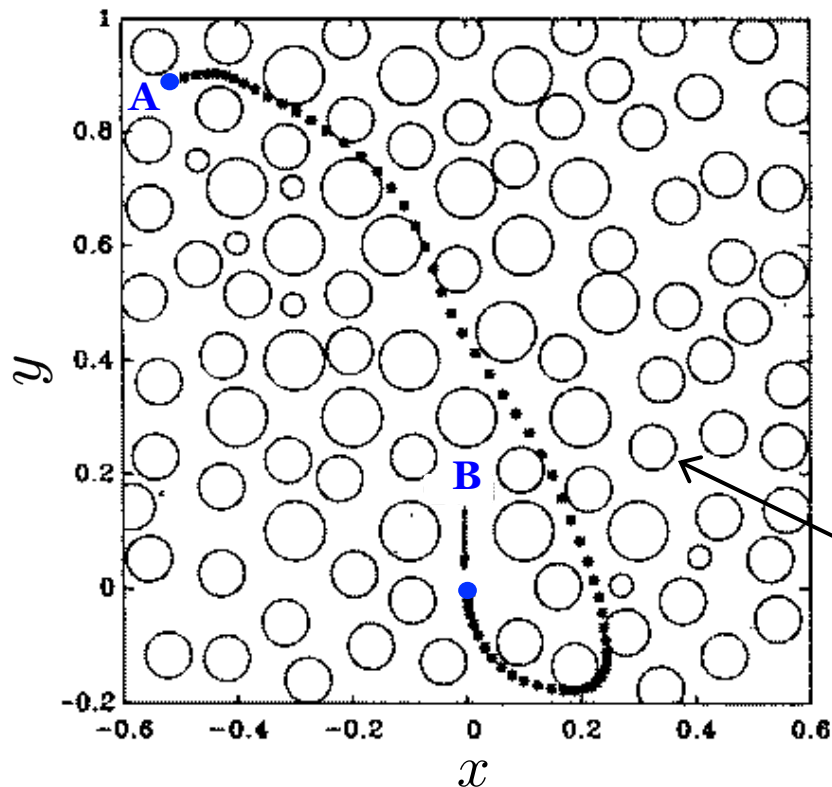
# Partie 4: Evitement d'obstacles

# Evitement d'obstacles: introduction

## Obstacle

- *Statique ou dynamique*

Exemples: poteau, mur, marche, humain, voiture, autres robots



**Objectif:** déterminer un parcours de A à B qui permet au robot d'éviter les collisions avec les obstacles

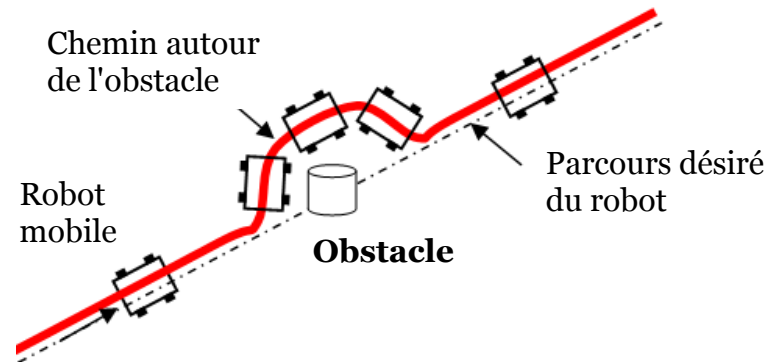
**Evitement d'obstacles:** géré normalement par un contrôleur de bas-niveau

# Evitement d'obstacles: introduction

- Comportement de base indispensable au bon fonctionnement d'un robot dans un *environnement dynamique*
- Il faut gérer les écarts entre le modèle interne et le monde réel

## Méthodes (pour des *obstacles statiques*):

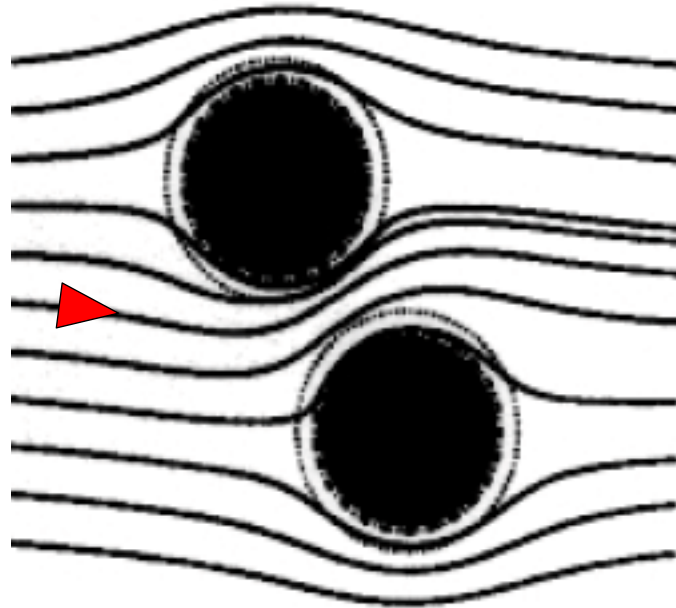
0. *Véhicule de Braitenberg* (approche réactif, pas de carte de l'environ.)
1. Champs de potentiel
2. Vector field histogram
3. Fenêtre dynamique
4. Graphe de Voronoï
5. Planification probabiliste: PRM, RRT



- Besoin de perceptions précises (ex. télémètres laser): avec sonars ou autres, une représentation *locale* de l'environnement, centrée sur le robot, peut être nécessaire
- Carte de l'environnement: *connue*
- Représentation de l'environnement: mtd 1, 3, 4, 5 *déterministe*, mtd 2 *probabiliste*

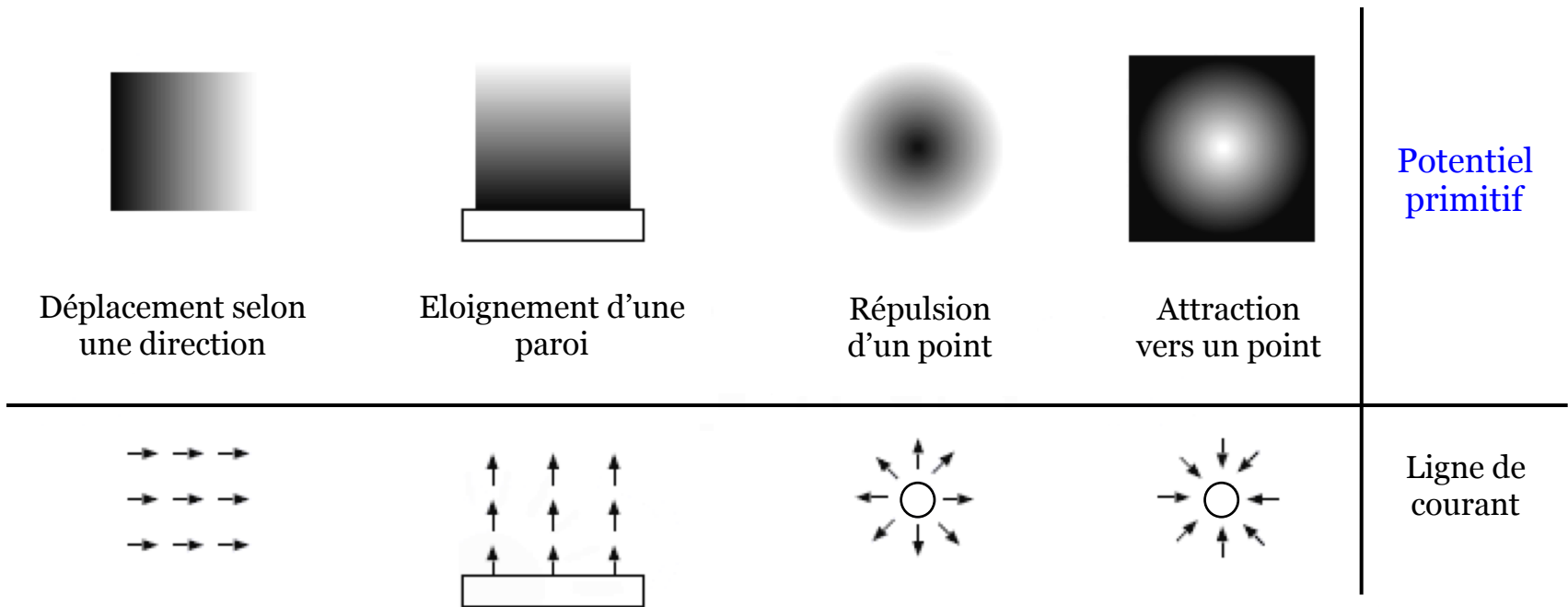
# Partie 4: Evitement d'obstacles

## 1. Champs de potentiel



# Champs de potentiel: introduction

- Robot: vu comme une *particule*
- Déplacement suivant les lignes de courant d'un potentiel obtenu par la perception de l'environnement



*“Real-time obstacle avoidance for manipulators and mobile robots”*,  
O. Khatib, Int. Journal of Robotics Research, vol. 5, n. 1, pp. 90-98, 1986

# Champs de potentiel: introduction

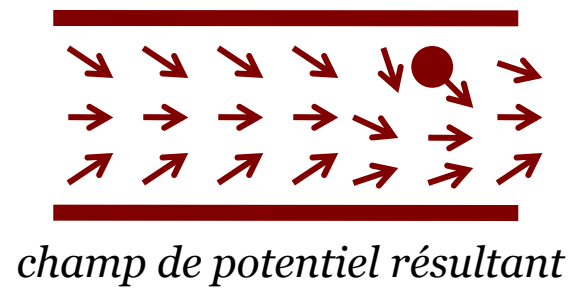
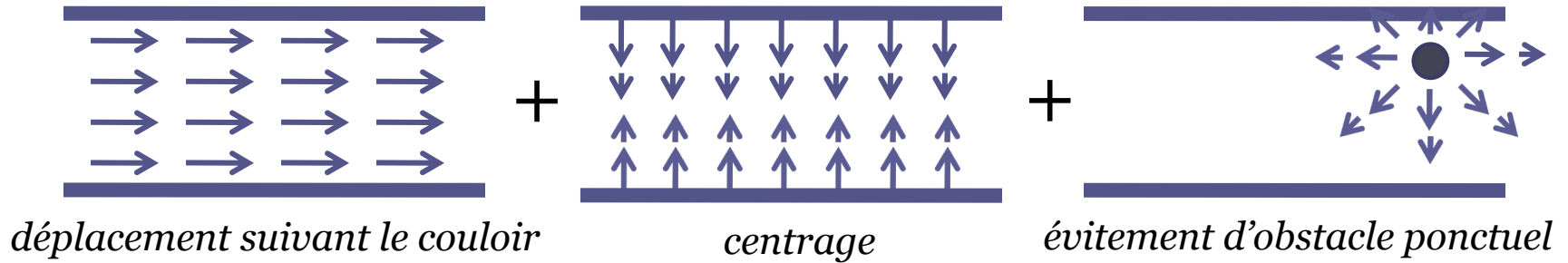
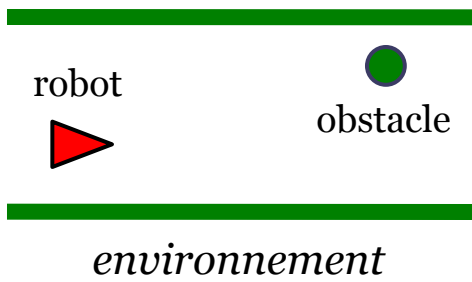
- Potentiel: différents objectifs
  - Évitement d'obstacles
  - Déplacement dans une direction préférée (vers le but)
- Comment le calculer ?
  - Sommation de potentiels primitifs
- Particularités des potentiels primitifs
  - Étendue spatiale limitée ou non
  - Intensité: fonction de la distance ou non

*“Robotics Modeling, Planning and Control”*, B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo, Springer 2009, Sect. 12.6



# Champs de potentiel: introduction

## Simple exemple



# Champs de potentiel

## Formulation mathématique:

$$\mathbf{F}(\mathbf{q}) = -\nabla U(\mathbf{q})$$

où

$\mathbf{F}(\mathbf{q})$  force agissant sur le robot dans la position  $\mathbf{q} = [x, y]^T$

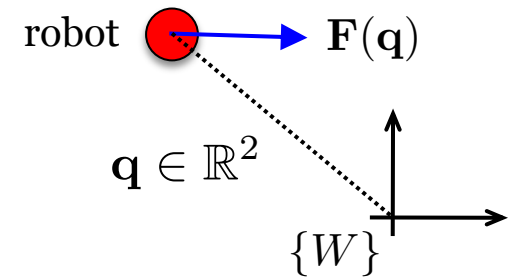
$U(\mathbf{q}) : \mathbb{R}^2 \rightarrow \mathbb{R}$  fonction potentiel de l'environnement

En d'autres termes, la force  $\mathbf{F}$  agissant sur le robot dans la position  $\mathbf{q}$  est égale au gradient négatif de la fonction potentielle  $U(\mathbf{q})$

- Nous étudierons deux types de potentiel  $U$ :

- 1) **Potentiel attractif** : pour guider le robot vers le **but**  $\mathbf{q}_b$

- 2) **Potentiel répulsif** : pour éviter la collision avec les obstacles



# Champs de potentiel

## Potentiels attractifs

1) *Fonction quadratique* de la distance du but  $\mathbf{q}_b$ :

$$U_a(\mathbf{q}) = \frac{1}{2} k_a \|\mathbf{q}_b - \mathbf{q}\|^2, \quad k_a > 0$$

$$\mathbf{F}_a(\mathbf{q}) = -\nabla U_a(\mathbf{q}) = k_a(\mathbf{q}_b - \mathbf{q})$$

Convergence vers zéro linéaire lorsque  $\mathbf{q}$  tend vers  $\mathbf{q}_b$ ;  $\mathbf{F}_a$  tend à augmenter indéfiniment lorsque la norme de l'erreur augmente

2) *Fonction linéaire* de la distance du but  $\mathbf{q}_b$ :

$$U_a(\mathbf{q}) = k_a \|\mathbf{q}_b - \mathbf{q}\|, \quad k_a > 0$$

$$\mathbf{F}_a(\mathbf{q}) = -\nabla U_a(\mathbf{q}) = k_a \frac{\mathbf{q}_b - \mathbf{q}}{\|\mathbf{q}_b - \mathbf{q}\|}$$

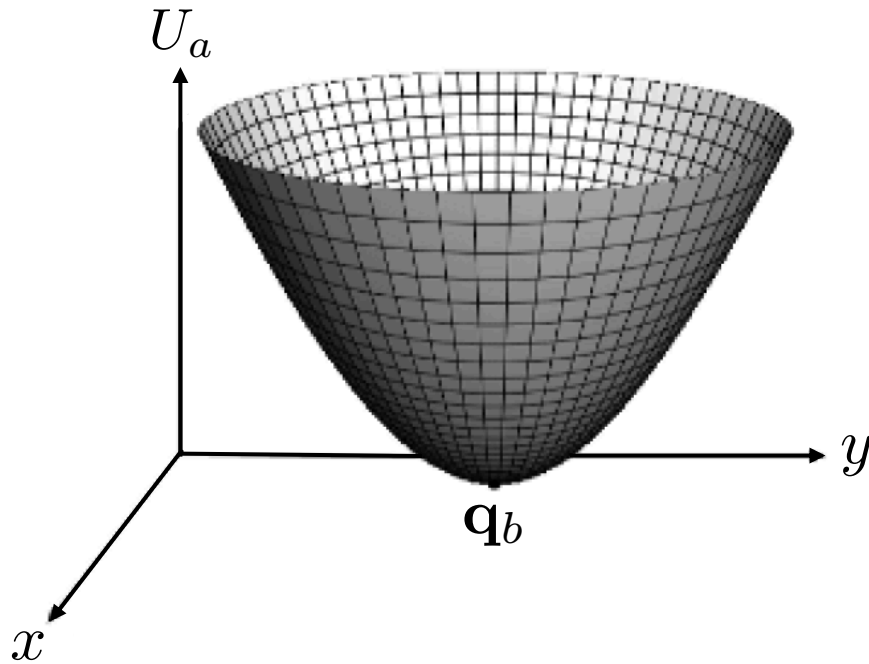
Module de la force constant;  $\mathbf{F}_a$  n'est pas définie en  $\mathbf{q}_b$

**Bon compromis:** utiliser la 2<sup>e</sup> fonction lorsque le robot est éloigné du but et la 1<sup>re</sup> près de  $\mathbf{q}_b$ . Pour avoir une *force continue*, passer d'une à l'autre lorsque  $\|\mathbf{q}_b - \mathbf{q}\| = 1$

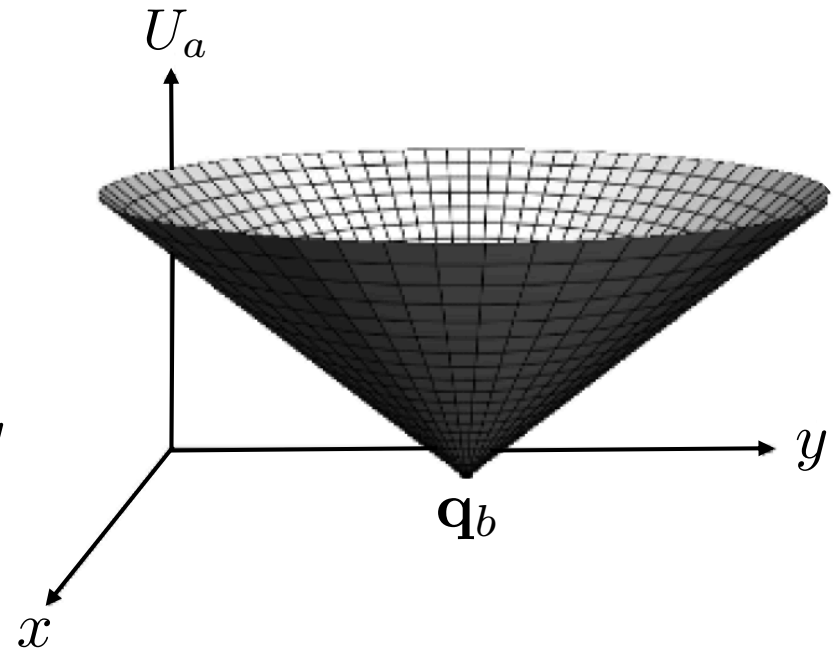
# Champs de potentiel

## Potentiels attractifs

1) *Fonction quadratique* : paraboloïde



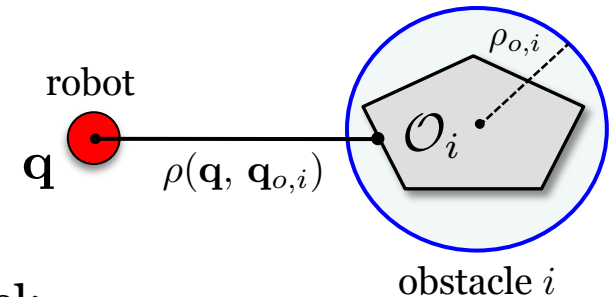
2) *Fonction linéaire* : cône



Les deux fonctions sont affichées pour  $k_a = 1$

# Champs de potentiel

## Potentiels répulsifs



Pour un obstacle convexe  $\mathcal{O}_i$ , on peut définir le potentiel:

$$U_{r,i}(\mathbf{q}) = \begin{cases} \frac{k_{r,i}}{\gamma} \left( \frac{1}{\rho(\mathbf{q}, \mathbf{q}_{o,i})} - \frac{1}{\rho_{o,i}} \right)^\gamma & \text{si } \rho(\mathbf{q}, \mathbf{q}_{o,i}) \leq \rho_{o,i} \\ 0 & \text{si } \rho(\mathbf{q}, \mathbf{q}_{o,i}) > \rho_{o,i} \end{cases}$$

avec  $i \in \{1, \dots, N\}$  ( $N$  est le nombre d'obstacles), étant:

$k_{r,i}$  : constante positive

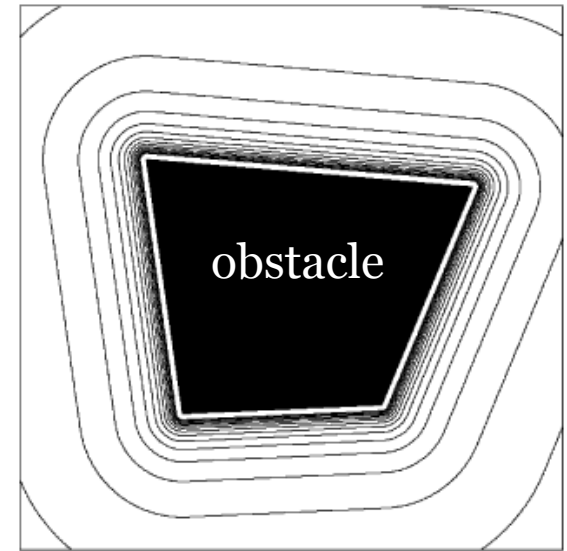
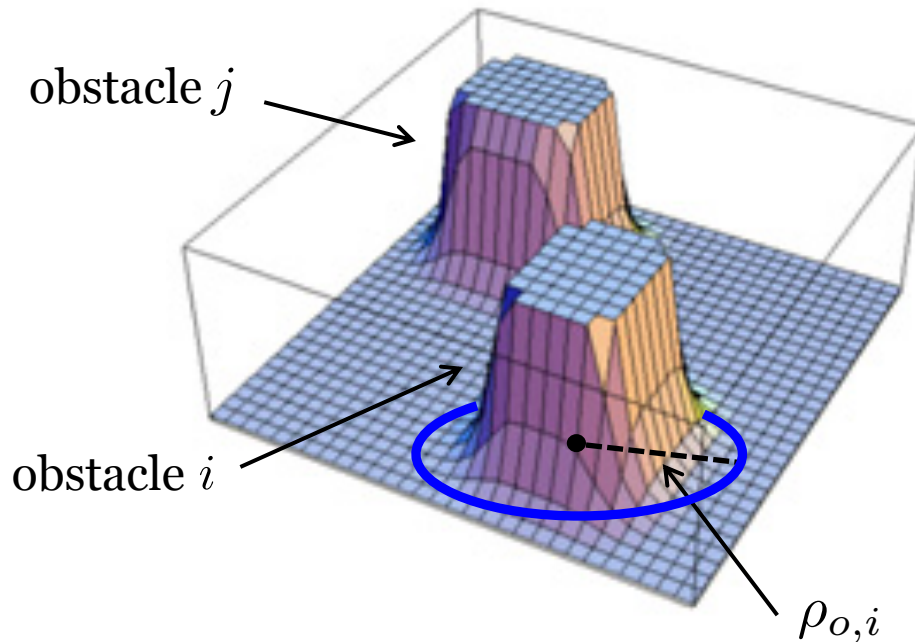
$\rho(\mathbf{q}, \mathbf{q}_{o,i}) = \min_{\mathbf{q}_{o,i} \in \mathcal{O}_i} \|\mathbf{q} - \mathbf{q}_{o,i}\|$  : distance minimale entre la position du robot  $\mathbf{q}$  et les points  $\mathbf{q}_{o,i}$  de l'obstacle  $i$

$\rho_{o,i}$  : rayon d'influence de l'obstacle  $i$

$\gamma \in \{2, 3, \dots\}$

# Champs de potentiel

## Potentiels répulsifs



Contours équipotentiels  
d'un potentiel répulsif  $U_{r,i}$   
pour  $k_{r,i} = 1$ ,  $\gamma = 2$

Si  $\gamma$  est grande, la "pente" du potentiel est forte (typiquement,  $\gamma = 2$ )

# Champs de potentiel

## Potentiels répulsifs

Force repulsive résultant du potentiel  $U_{r,i}(\mathbf{q})$  :

$$\mathbf{F}_{r,i}(\mathbf{q}) = -\nabla U_{r,i}(\mathbf{q})$$

$$= \begin{cases} \frac{k_{r,i}}{\rho^2(\mathbf{q}, \mathbf{q}_{o,i})} \left( \frac{1}{\rho(\mathbf{q}, \mathbf{q}_{o,i})} - \frac{1}{\rho_{o,i}} \right)^{\gamma-1} \nabla \rho(\mathbf{q}, \mathbf{q}_{o,i}) & \text{si } \rho(\mathbf{q}, \mathbf{q}_{o,i}) \leq \rho_{o,i} \\ 0 & \text{si } \rho(\mathbf{q}, \mathbf{q}_{o,i}) > \rho_{o,i} \end{cases}$$

On peut définir le *potentiel répulsif global* (pour les  $N$  obstacles), comme:

$$U_r(\mathbf{q}) = \sum_{i=1}^N U_{r,i}(\mathbf{q})$$

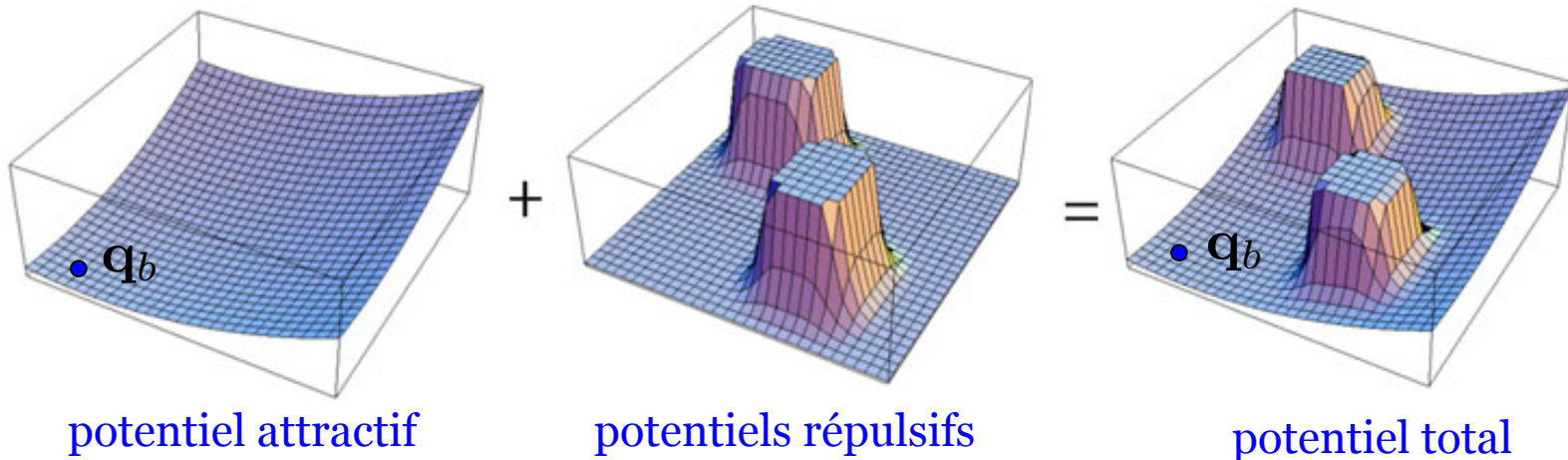
# Champs de potentiel

Potentiel total = potentiel attractif + potentiels répulsifs

$$U_t(\mathbf{q}) = U_a(\mathbf{q}) + U_r(\mathbf{q})$$

Ceci se traduit par la *force totale* agissant sur le robot:

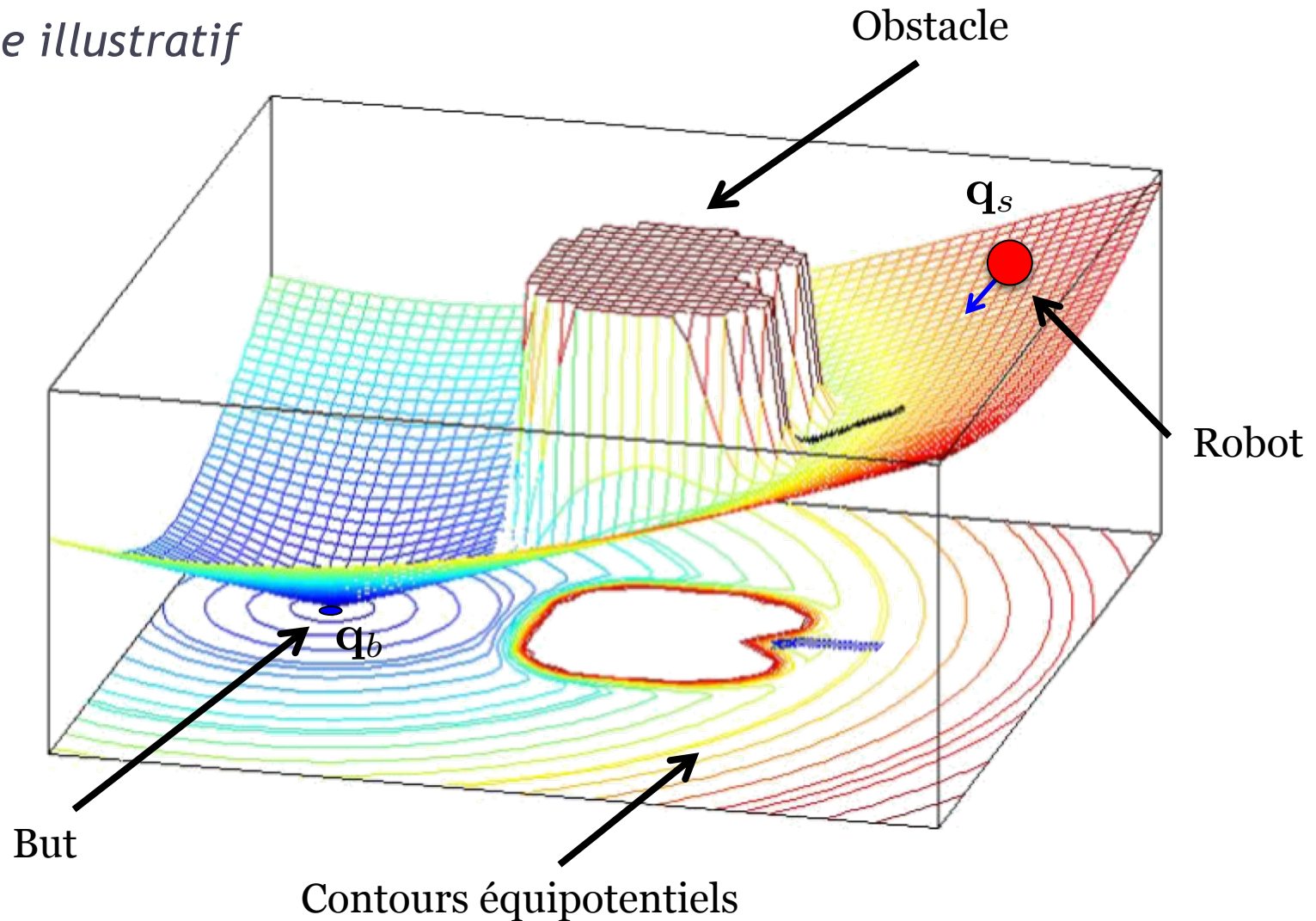
$$\mathbf{F}_t(\mathbf{q}) = -\nabla U_t(\mathbf{q}) = \mathbf{F}_a(\mathbf{q}) + \sum_{i=1}^N \mathbf{F}_{r,i}(\mathbf{q})$$





# Champs de potentiel

*Exemple illustratif*



# Champs de potentiel

Pour atteindre le but  $\mathbf{q}_b$ , on peut utiliser la *descente de gradient*.

À partir de:

$$\dot{\mathbf{q}} = -\nabla U_t(\mathbf{q})$$

si  $\mathbf{q}_0 = \mathbf{q}_s$  est la position initiale du robot, on peut mettre à jour sa position au fil du temps de la manière suivante:

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \alpha_k \frac{\mathbf{F}_t(\mathbf{q}_k)}{\|\mathbf{F}_t(\mathbf{q}_k)\|}, \quad \alpha_k > 0, \quad k \in \{0, 1, \dots\}$$

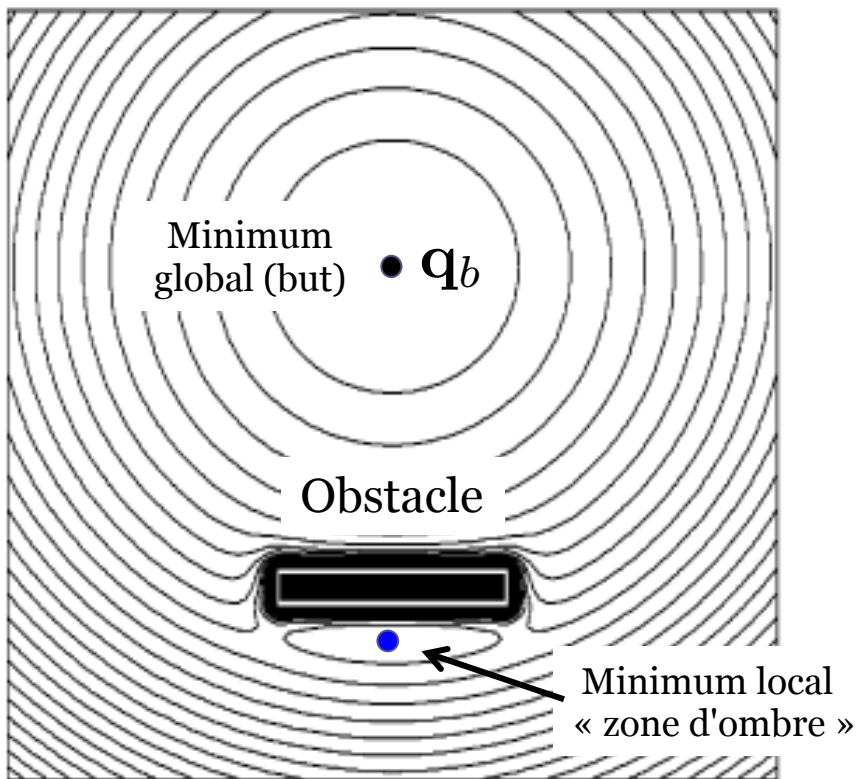
**Problème:** convergence vers tout *point critique*  $\mathbf{q}^*$  où  $\nabla U_t(\mathbf{q}^*) = \mathbf{0}$ .

Il faut vérifier la dérivée seconde de  $U_t$ : la matrice Hessienne de  $U_t$  doit être définie positive en  $\mathbf{q}^*$  pour avoir un *minimum local*

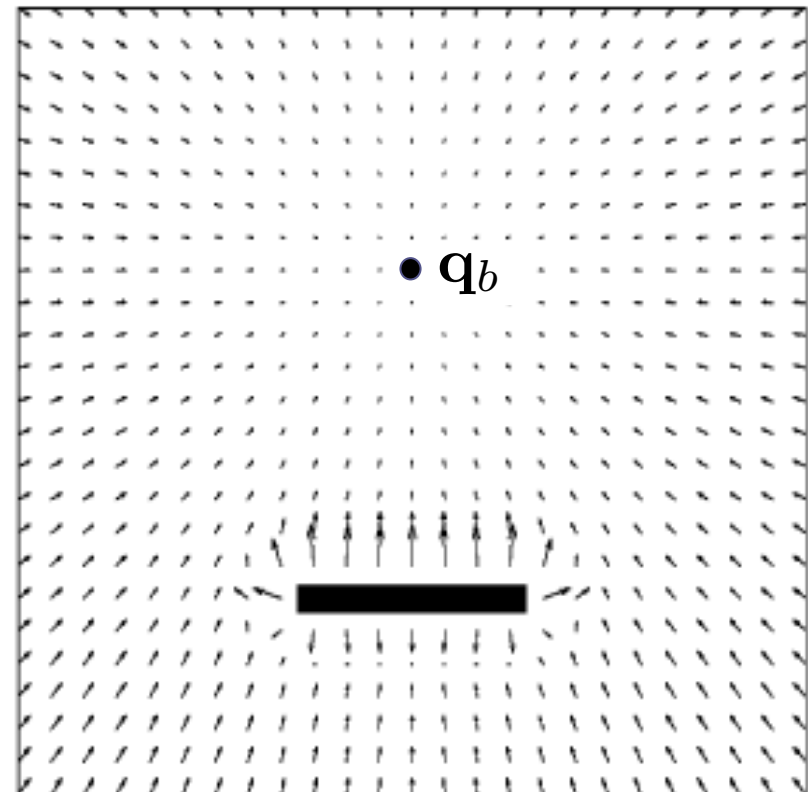
$$\mathbf{H}(\mathbf{q}^*) = \begin{bmatrix} \frac{\partial^2 U_t}{\partial x^2} & \frac{\partial^2 U_t}{\partial x \partial y} \\ \frac{\partial^2 U_t}{\partial y \partial x} & \frac{\partial^2 U_t}{\partial y^2} \end{bmatrix} \Bigg|_{\mathbf{q}=\mathbf{q}^*} \succ \mathbf{0}$$

# Champs de potentiel

Exemple: *obstacle rectangulaire*



Contours équipotentiels



Champ de force  $\mathbf{F}_t(\mathbf{q})$  résultant  
(lignes de courant)

# Champs de potentiel

- Problèmes
  1. *Minima locaux* du potentiel dans certaines configurations
  2. Pas de «décision» de la direction à prendre pour le robot
- Solutions au 1<sup>er</sup> problème:
  - Lorsque dans un minimum local, déclencher un comportement différent:
    - Par ex. déplacement aléatoire, suivi de murs
  - Hypothèse du « *monde sphérique* »: tous les obstacles ont une forme sphérique. Le potentiel total a des points-selles isolés (où le gradient est zéro), mais pas des minima locaux
  - *Fonctions de navigation* [Rimon & Koditschek, TRA92], champs rotationnels, potentiel imposé en *fonction harmonique*\*
    - Les fonctions harmoniques garantissent l'absence de minima locaux, mais complexité élevée (résultats en simulation uniquement)

\* Une *fonction harmonique* est une fonction  $f : \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}$  deux fois continûment dérivable qui satisfait l'équation de Laplace sur  $\Omega$ , à savoir  $\Delta f = 0$ .

# Champs de potentiel: sommaire

- Avantages:
  - 1) Capacité de générer « en temps réel » des trajectoires pour éviter les obstacles
  - 2) Facilité de mise en œuvre
- Formalisme du *schéma moteur*
  - Action sous forme de potentiel
  - Dépend des perceptions
- Utilisation
  - Bas-niveau dans une architecture hybride

# Partie 4: Evitement d'obstacles

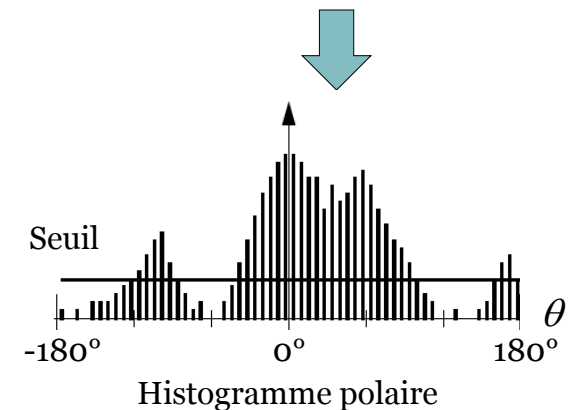
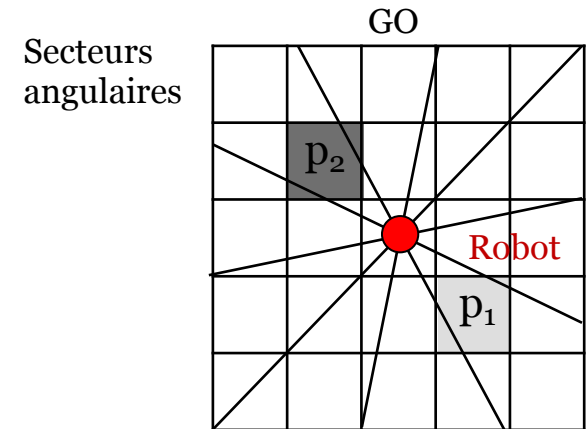
## 2. Vector field histogram (VFH)

# Vector field histogram

- Historiquement inventé pour les sonars (capteurs à ultra-sons)
- Robustesse aux incertitudes de modèle/mesure

## 1. Grille d'occupation (GO) locale

- Représentation statistique de l'environnement
- Discrétisation de l'espace en cellules de taille fixe
- Chaque cellule contient une probabilité  $p_i$ 
  - Haute si souvent perçue contenant un obstacle par le laser ou sonar
  - Faible sinon
- Mise à jour en continu (en temps réel)



## 2. Histogramme d'occupation

Représente l'occupation de l'espace autour du robot

*"The Vector Field Histogram - Fast Obstacle Avoidance for Mobile Robots"*,  
Y. Koren, J. Borenstein, IEEE Trans. Robot. Autom., vol. 7, n. 3, pp. 278-288, 1991

# Vector field histogram

## 1. Construction de la grille d'occupation (GO)

- Estimation de la position du robot  
(cf. le chapitre 1: localisation par odométrie)
- Perception
- Mise à jour des cellules de la GO
  - Modèle probabiliste:  $p(\text{occ}_i | s)$

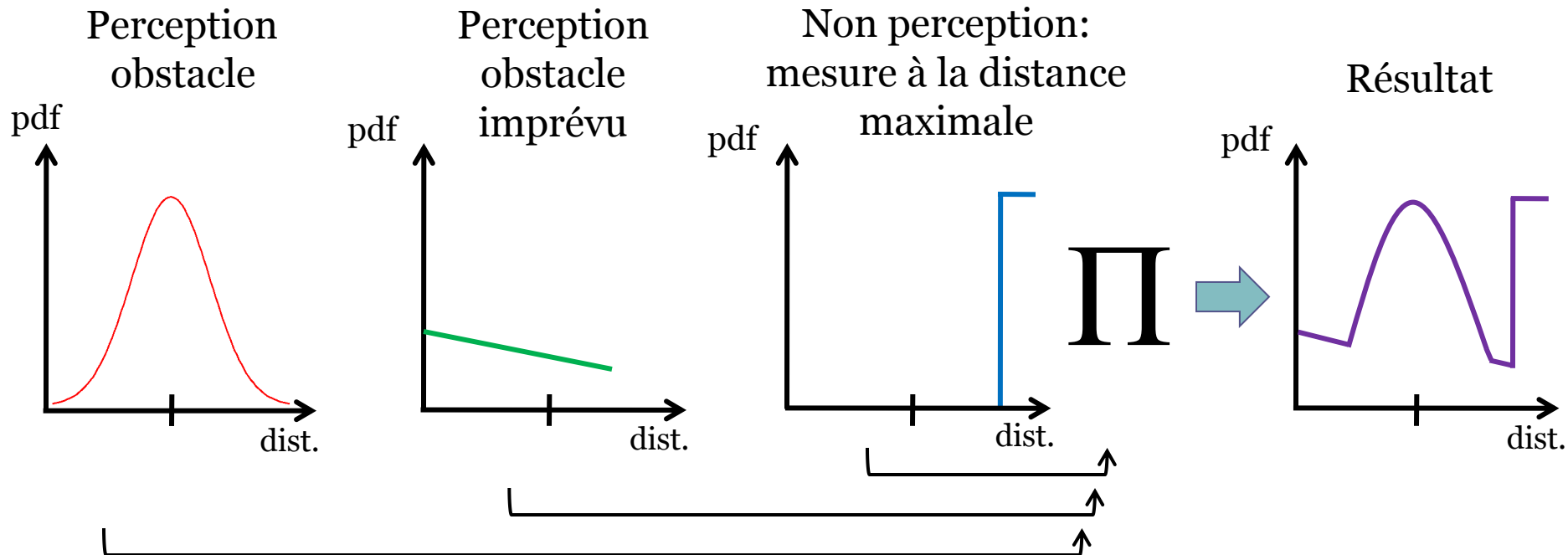
Pour une perception  $s$  donnée,  $p(\text{occ}_i | s)$  nous fournit la *probabilité d'occupation de la cellule  $i$*  dans le champ de vue du capteur du robot, en fonction de la valeur renvoyée par le capteur



# Construction d'une GO

- *Modèle probabiliste* d'un capteur à ultra-sons
- Probabilité de mesure en fonction de la *distance* de l'obstacle:

$$p(\text{obstacle} \mid \text{scan}) = \prod_{j=1}^M p(\text{mesure } j \mid \text{distObstacle})$$



# Construction d'une GO

- $p(\text{occ}_i | s)$ 
  - Similaire au cas précédent
  - Erreur gaussienne pour l'écart entre:
    - La direction de la cellule
    - La direction du capteur
- **Objectif:**
  - Accumuler les  $T$  mesures  $s_1, \dots, s_T$
  - Estimer la probabilité de la cellule  $i$  d'être occupée:

$$p(\text{occ}_i^T) = p(\text{occ}_i | s_1, \dots, s_T)$$

# Construction d'une GO

- Théorème de Bayes pour extraire la probabilité en fonction de la *dernière mesure*  $s_T$ . Rappel que:

$$p(x | y) = \frac{p(y | x) p(x)}{p(y)}$$

- Dans notre cas:

$$p(\text{occ}_i^T) = \frac{p(s_T | \text{occ}_i, \cancel{s_1, \dots, s_{T-1}}) p(\text{occ}_i^{T-1})}{p(s_T | s_1, \dots, s_{T-1})}$$

- Avec une hypothèse de « *monde statique* », on simplifie:

$$p(\text{occ}_i^T) = \frac{p(s_T | \text{occ}_i) p(\text{occ}_i^{T-1})}{p(s_T | s_1, \dots, s_{T-1})}$$

« *Monde statique* »: toutes les mesures sont *conditionnellement indépendantes* si on connaît la valeur d'une cellule de la GO (ce qui est *faux* en pratique et produit des limitations de la méthode)

# Construction d'une GO

- Notre modèle de capteur nous donne  $p(\text{occ}_i | s_T)$ , alors nous le faisons apparaître (à nouveau, Théor. de Bayes):

$$p(\text{occ}_i^T) = \frac{p(\text{occ}_i | s_T) p(s_T)}{p(\text{occ}_i)} \frac{p(\text{occ}_i^{T-1})}{p(s_T | s_1, \dots, s_{T-1})}$$

- Mais la probabilité d'«*inoccupation*»  $p(\overline{\text{occ}}_i) = 1 - p(\text{occ}_i)$
- La probabilité que la cellule  $i$  soit vide est donc:

$$p(\overline{\text{occ}}_i^T) = \frac{p(\overline{\text{occ}}_i | s_T) p(s_T)}{p(\overline{\text{occ}}_i)} \frac{p(\overline{\text{occ}}_i^{T-1})}{p(s_T | s_1, \dots, s_{T-1})}$$

# Construction d'une GO

- Le rapport des deux probabilités nous donne:

$$\frac{p(\text{occ}_i^T)}{p(\overline{\text{occ}}_i^T)} = \frac{p(\text{occ}_i | s_T)}{p(\overline{\text{occ}}_i | s_T)} \frac{p(\overline{\text{occ}}_i^{T-1})}{p(\text{occ}_i^{T-1})} \frac{p(\overline{\text{occ}}_i)}{p(\text{occ}_i)}$$

- En sachant que  $p(\overline{\text{occ}}_i) = 1 - p(\text{occ}_i)$ , tout est connu:

$$\frac{p(\text{occ}_i^T)}{1 - p(\text{occ}_i^T)} = \frac{p(\text{occ}_i | s_T)}{1 - p(\text{occ}_i | s_T)} \frac{1 - p(\text{occ}_i^{T-1})}{p(\text{occ}_i^{T-1})} \frac{1 - p(\text{occ}_i)}{p(\text{occ}_i)}$$

- Nous pourrions exprimer  $p(\text{occ}_i^T)$ , mais il est plus simple déterminer la quantité:

$$\ell_i^T = \log \left( \frac{p(\text{occ}_i^T)}{1 - p(\text{occ}_i^T)} \right) \quad \log(\cdot) : \text{logarithme naturel (de base } e)$$

# Construction d'une GO

- Ce qui se calcule simplement par (rappel:  $\log(x^b) = b \log(x)$  et  $\log(xy) = \log(x) + \log(y)$ )

$$\ell_i^T = \log\left(\frac{p(\text{occ}_i | s_T)}{1 - p(\text{occ}_i | s_T)}\right) - \ell_i^{T-1} + \log\left(\frac{1 - p(\text{occ}_i)}{p(\text{occ}_i)}\right)$$

- Mise à jour *incrémentale* de  $\ell_i^T$ . On utilise:
  - Les valeurs précédentes
  - Le modèle de capteur
- $p(\text{occ}_i)$  généralement initialisée à  $1/2$  (l'initialisation intègre l'*a priori* sur le fait que l'environnement contienne plus ou moins d'obstacles)
- $\ell_i^T$  permet de calculer la probabilité d'occupation  $p(\text{occ}_i^T)$ :

$$p(\text{occ}_i^T) = 1 - \frac{1}{1 + \exp(\ell_i^T)}$$

# Construction d'une GO

- Autres possibilités de mise à jour des GO
  - Calcul de  $p(\text{occ}_i^T)$  en fonction du nombre de perceptions de la cellule  $i$  (« approche fréquentiste vs approche bayésienne »)

$$p(\text{occ}_i^T) = \frac{\text{nbocc}_i^T}{\text{nbvide}_i^T + \text{nbocc}_i^T}$$

Nombre de fois  
que la cellule  $i$  a  
été détectée *vide*  
(sur les  $T$  mesures)

Nombre de fois que un  
obstacle a été détecté  
dans la cellule  $i$   
(sur les  $T$  mesures)

# Construction d'une GO

## ▫ Autres possibilités de mise à jour des GO

### 2. *Histogrammic in-motion mapping* (HIMM)

$$p(\text{occ}_i^T) = p(\text{occ}_i^{T-1}) + \lambda \kappa$$

$$\lambda = \begin{cases} 1 & \text{si la cellule } i \text{ présente un obstacle} \\ -1 & \text{sinon} \end{cases}$$

Valeur d'incrément  
(paramètre positif)

L'HIMM est très simple, mais:

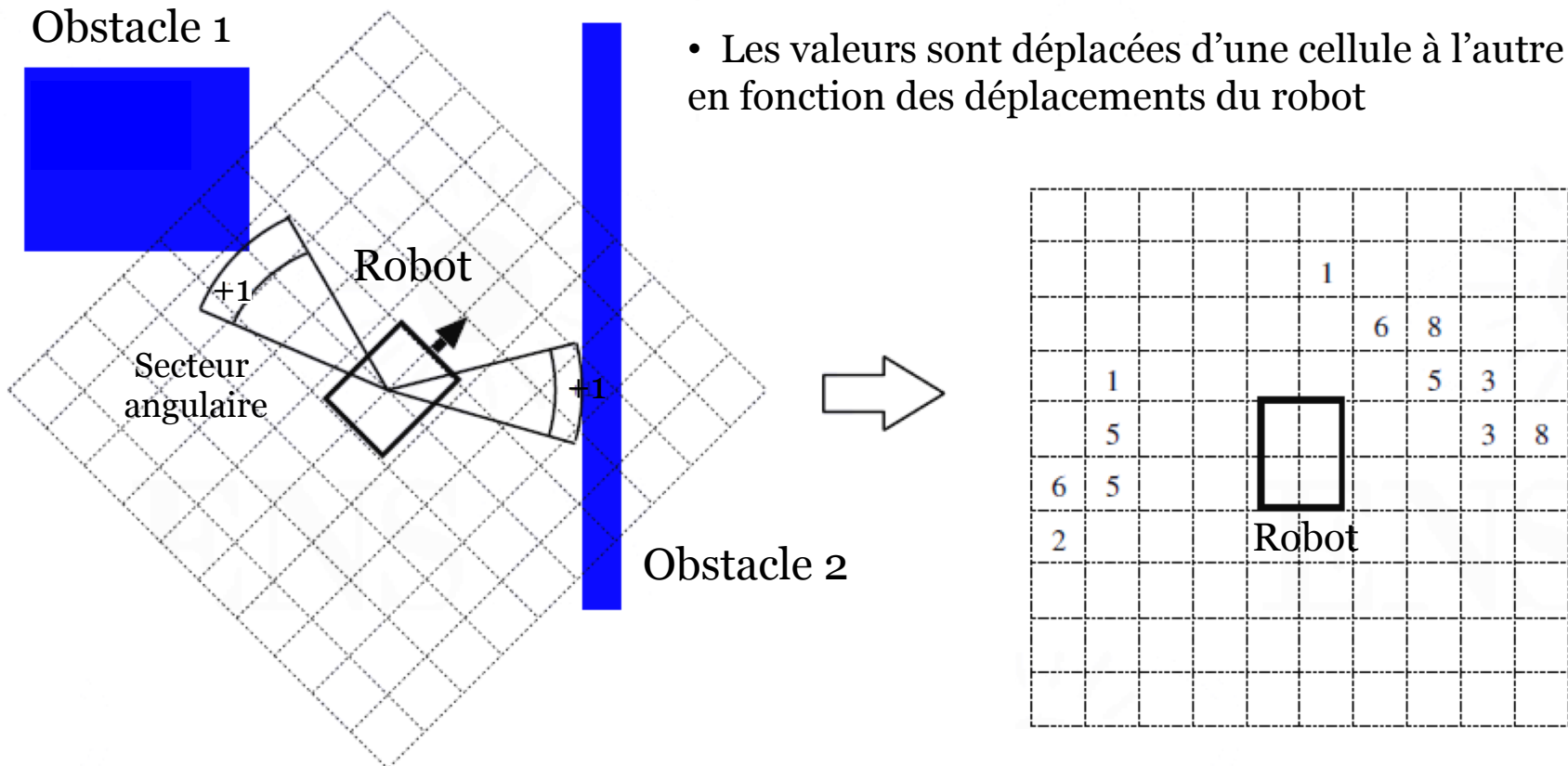
- Pas de convergence si le nombre de perceptions tend vers l'infini
- Sensible au bruit
- Réglage délicat des paramètres ( $\kappa$ ) pour être adapté à un robot

"*Histogrammic in-motion Mapping for Mobile Robot Obstacle Avoidance*", J. Borenstein, Y. Koren, IEEE Journal of Robotics and Automation, vol. 7, n. 4, pp. 535-539, 1991



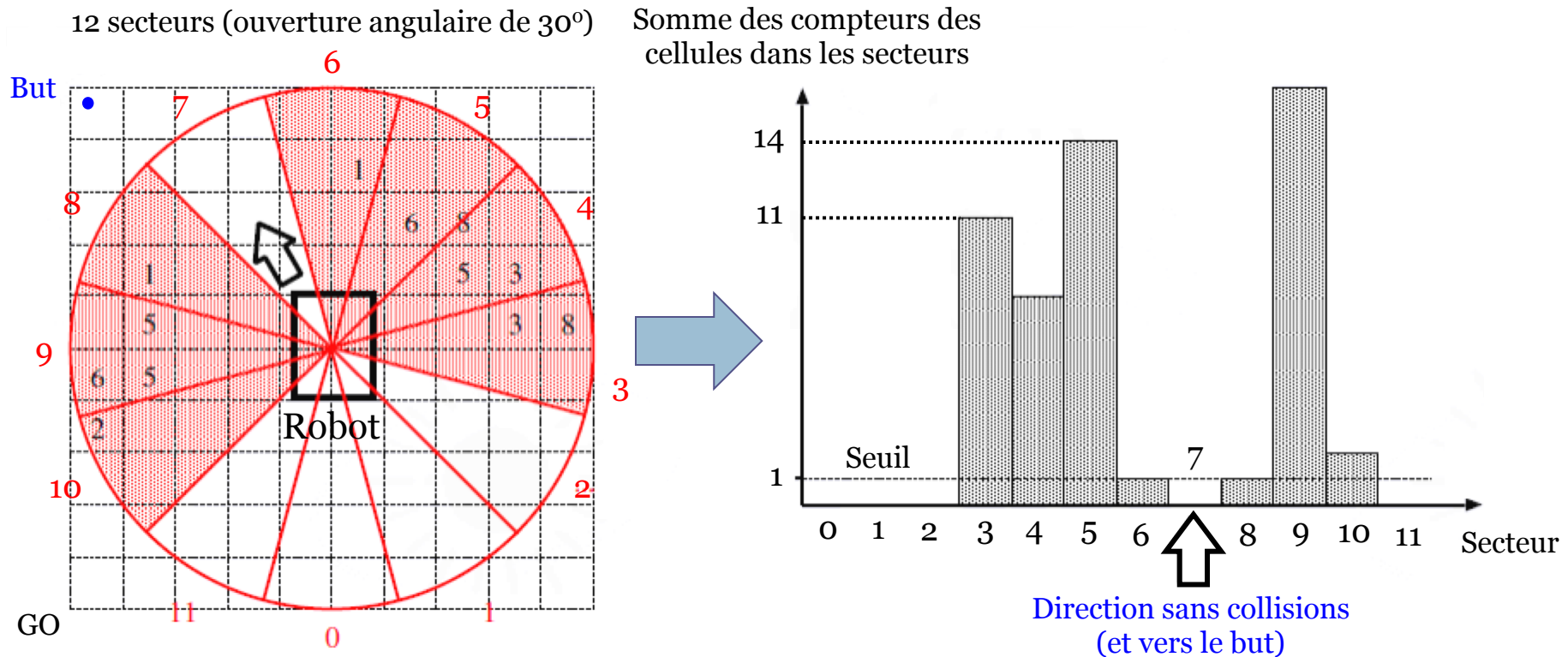
## 2. Histogramme d'occupation

- La GO est construite dans le *référentiel du robot*
- Un **compteur** est incrémenté pour chaque cellule appartenant au *secteur angulaire* dans lequel un obstacle a été détecté (ex. pour une probabilité  $> 1/2$ )
- Les valeurs sont déplacées d'une cellule à l'autre en fonction des déplacements du robot



## 2. Histogramme d'occupation

- Construction d'un *histogramme polaire* des obstacles à partir des valeurs des cellules (une barre verticale pour chaque secteur)
- On utilise l'histogramme pour déterminer la direction de déplacement du robot
- Seuillage pour éliminer une partie du bruit



## 2. Histogramme d'occupation

- Après seuillage
  - Ensemble de directions possibles (« vallées candidates »)
  - Choix selon un critère:
    - *Exemple*: direction la plus proche de celle du but
    - On peut prendre en considération le *modèle cinématique* et la *taille physique* du robot (VFH+ et VFH\*)

### Avantages du VFH

Très rapide, robuste aux incertitudes

Problème (« *local path planner* »):

Chemins non optimaux globalement  
(mais *quasi optimaux*, en pratique)

“VFH+: *Reliable obstacle avoidance for fast mobile robots*”, I. Ulrich, J. Borenstein, in Proc. Int. Conf. Robotics and Autom., pp. 1572-1577, 1998

“VFH\*: *Local Obstacle Avoidance with Look-Ahead Verification*”, I. Ulrich, J. Borenstein in Proc. IEEE Int. Conf. Robotics and Autom., pp. 2505-2511, 2000

