

Localisation et navigation de robots

UPJV, Département EEA

M2 3EA, EC32, parcours RoVA

Année Universitaire 2022-2023

Fabio MORBIDI

Laboratoire MIS

Équipe Perception Robotique

E-mail: fabio.morbidi@u-picardie.fr

Mardi et mercredi 9h00-12h00

salle CURI 304: CM & TD

salle TP204: TP



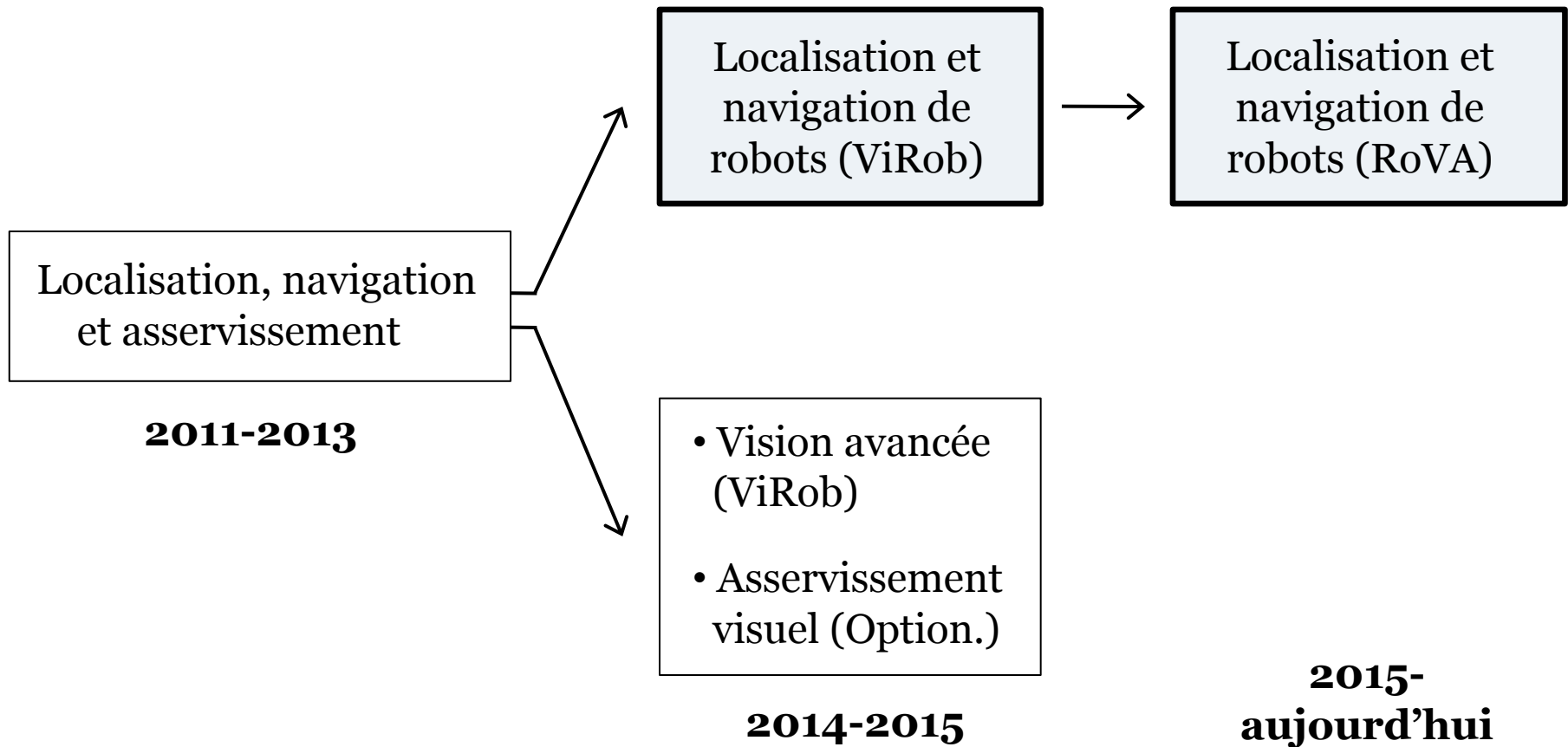
Electronique

Energie Electrique

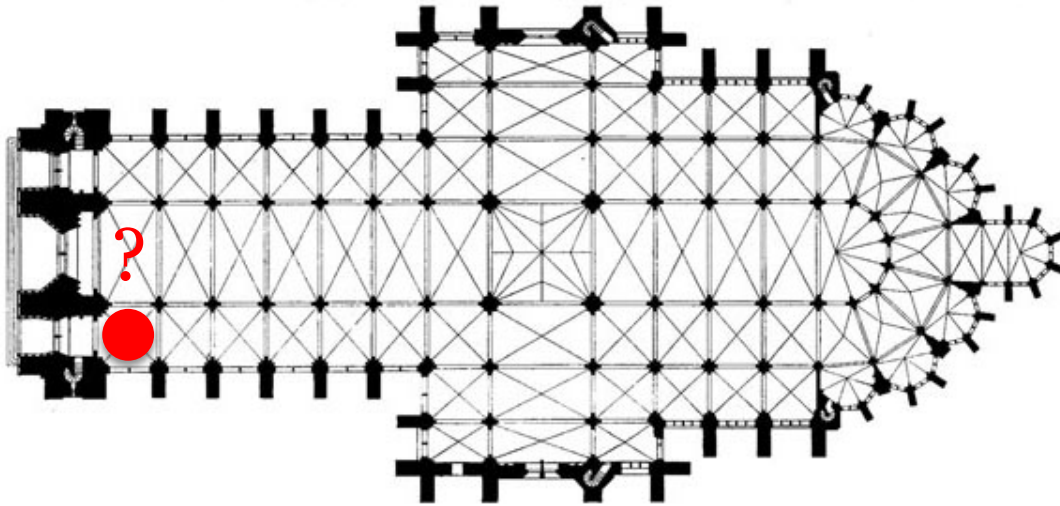
Automatique



Évolution du cours au fil du temps

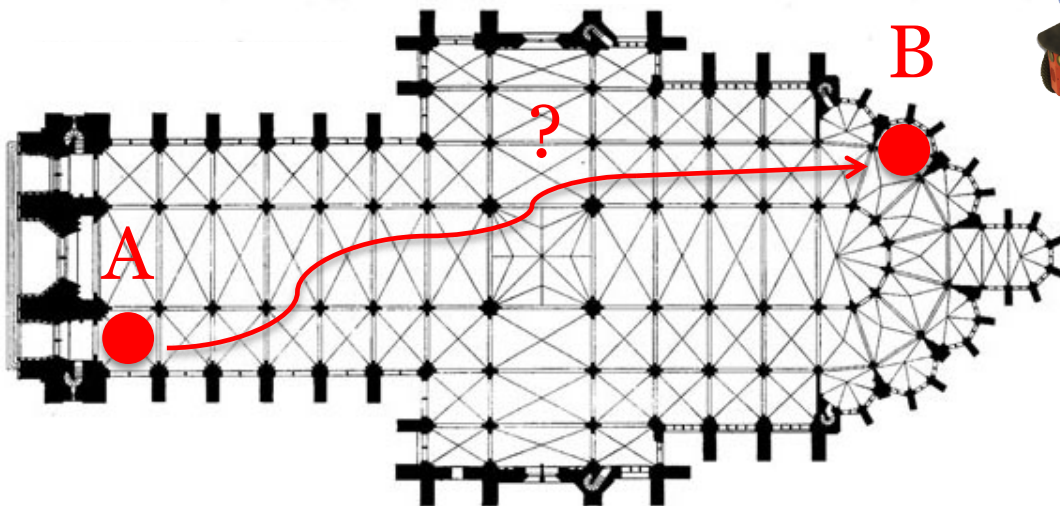


Deux problèmes cruciaux ... et fortement liés



Où suis-je?

Localisation !



Comment aller de A à B ?

Navigation !

et aussi ...

- Comment détecter B ?
- Comment éviter les obstacles ?

Plan du cours

Chapitre 1: Localisation

- 1.1 Introduction et défis
- 1.2 Odométrie
- 1.3 Localisation par filtre de Kalman
- 1.4 Autres techniques de localisation

Chapitre 2: Navigation

- 2.1 Stratégies de navigation
- 2.2 Architectures de contrôle
- 2.3 Navigation vers un but
- 2.4 Evitement d'obstacles

Bibliographie

- ***Introduction to Autonomous Mobile Robots***

R. Siegwart, I.R. Nourbakhsh, D. Scaramuzza, MIT press, 2^e éd., 2011 (chapitres 4, 5 et 6)

- ***Robotics: Modelling, Planning and Control***

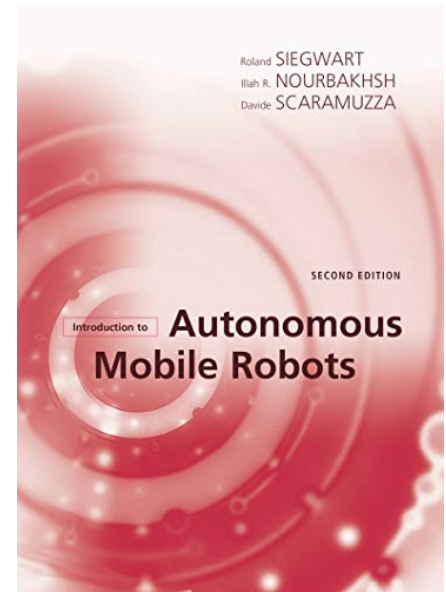
B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo, Springer, 1^{re} éd., 2009 (chapitres 4 et 12: planification de trajectoire)

- ***State Estimation for Robotics***

T.D. Barfoot, Cambridge University Press, 2017
(Partie I et III)

- ***Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software***

Y. Bar-Shalom, X.R. Li, T. Kirubarajan, John Wiley & Sons, 2001
(chapitres 5, 6 et 12)



Matériel de cours: http://home.mis.u-picardie.fr/~fabio/Teaching_LNR22-23.html

Connaissances préalables

On suppose que vous avez une bonne connaissance de:

- *Algèbre linéaire* (opérations sur matrices, valeurs et vecteurs propres, etc.)
- *Calcul différentiel* (gradient, matrice jacobienne, etc.)
- *Théorie des probabilités et de l'estimation, variables et processus aléatoires*
- *Notions de base de robotique mobile* (cours de PARM)

Structure du cours: CM (2 chapitres), 3 TD

Contrôle: 1 DS (exercices, questions de théorie), 3 TP (AmigoBots)



$$\text{Note finale} = \frac{1}{2} \left[\text{DS} + \left(\frac{\text{TP1} + \text{TP2} + \text{TP3}}{3} \right) \right]$$

Localisation et navigation de robots

UPJV, Département EEA

M2 3EA, EC32, parcours RoVA

Chapitre 1 : Localisation



Plan du chapitre

• Introduction et défis **Partie 1**

• Odométrie **Partie 2**

• Localisation par filtre de Kalman **Partie 3**

• Autres techniques de localisation **Partie 4**

Partie 1 : Introduction et défis

Introduction

Hypothèse: une carte de l'environnement est disponible



← Localisation facile

Localisation difficile →

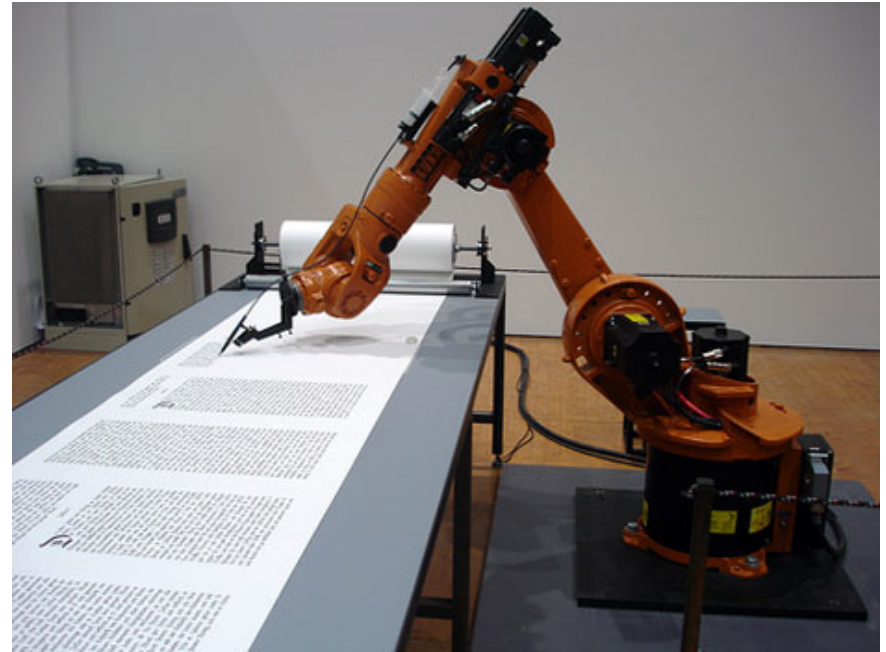


Introduction

- Robots manipulateurs

- Robot de type série:
où est l'effecteur ?

- Robot « fixe »



- À partir de la mesure des variables articulaires on peut déduire la pose de l'effecteur (modèle géométrique direct)
 - Connaissance des longueurs des segments nécessaire
 - Plus un problème de « *calibrage* » que de localisation

(cf. Cours M1 de “Robotique Industrielle”)

Introduction

- Robots manipulateurs
 - Robot delta:
où est l'effecteur ?



- Modélisation géométrique: plus complexe que pour les robots de type série ...
... mais volume de travail *limité* et facilement identifiable

Introduction

- Robots mobiles
 - Milieu ouvert
 - Pas de lien fixe avec l'environnement
 - Pas de mesures directes de la pose (position et orientation)
 - Zone de mouvement « sans limite »

[Vidéo ETHZ](#)

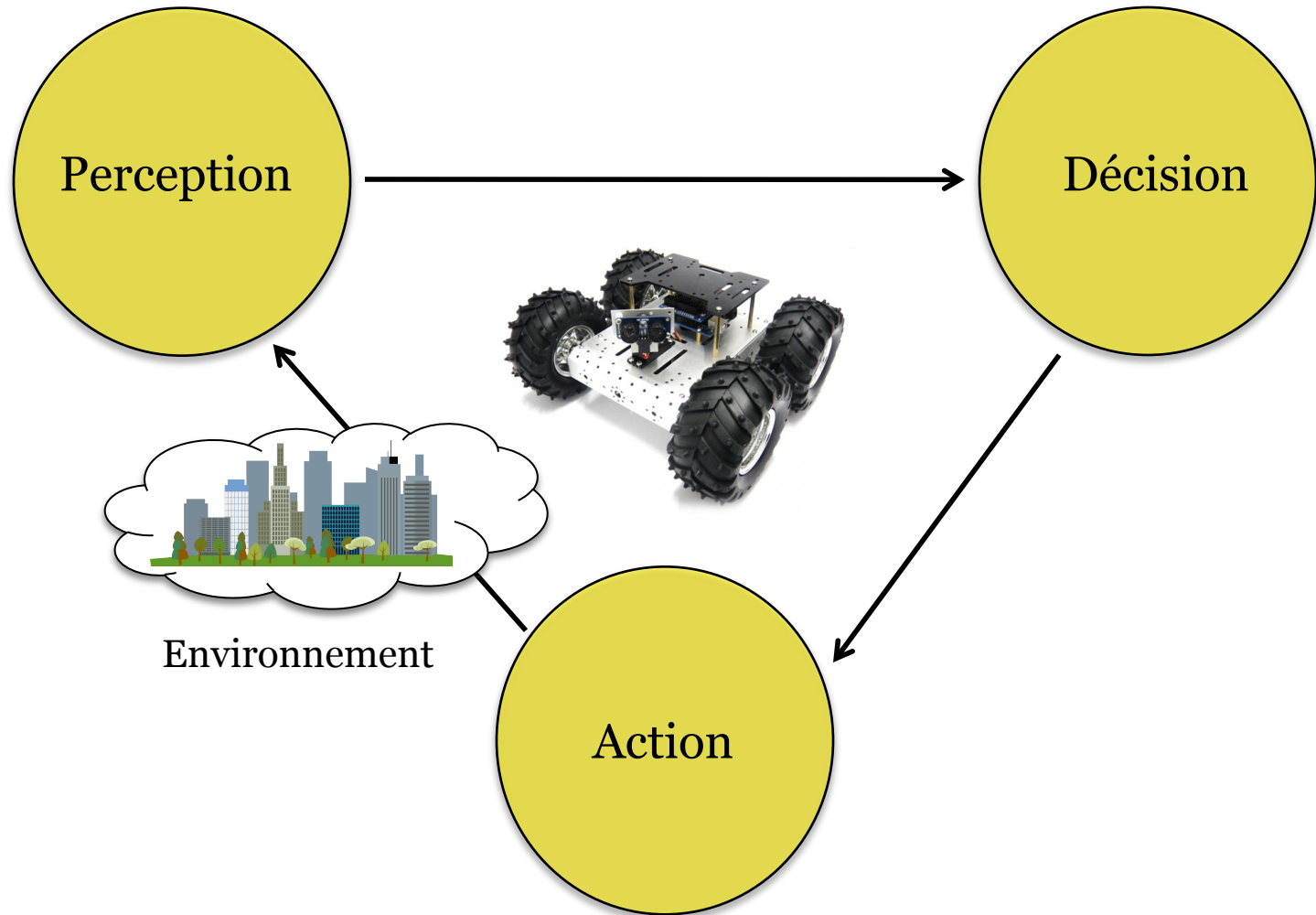


Introduction

- **Navigation**

- *À grande échelle*: on a besoin de la localisation (compétence importante et critique d'un robot mobile)
- En particulier, pour le succès de la navigation nous avons besoin de la:
 - **Perception**: le robot doit interpréter les mesures des capteurs pour extraire des informations pertinentes
 - **Localisation**: le robot doit déterminer sa position dans l'environnement
 - **Décision**: le robot doit décider de comment agir pour atteindre son but
 - **Commande du mouvement**: le robot doit actionner ses moteurs pour réaliser la trajectoire désirée

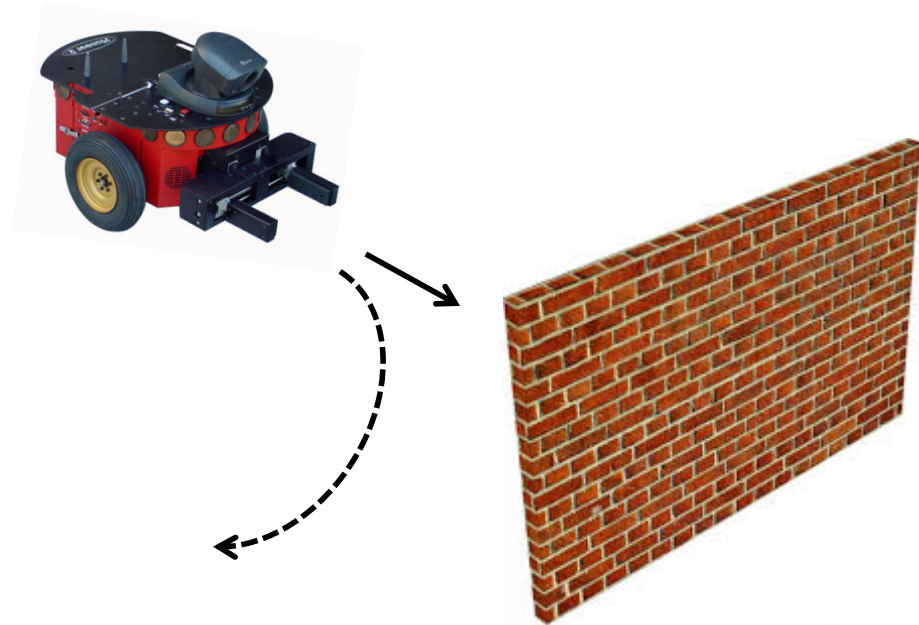
Introduction



Paradigme “See-Think-Act”

Introduction

Exemple ...



Perception: Il y a un mur en face !

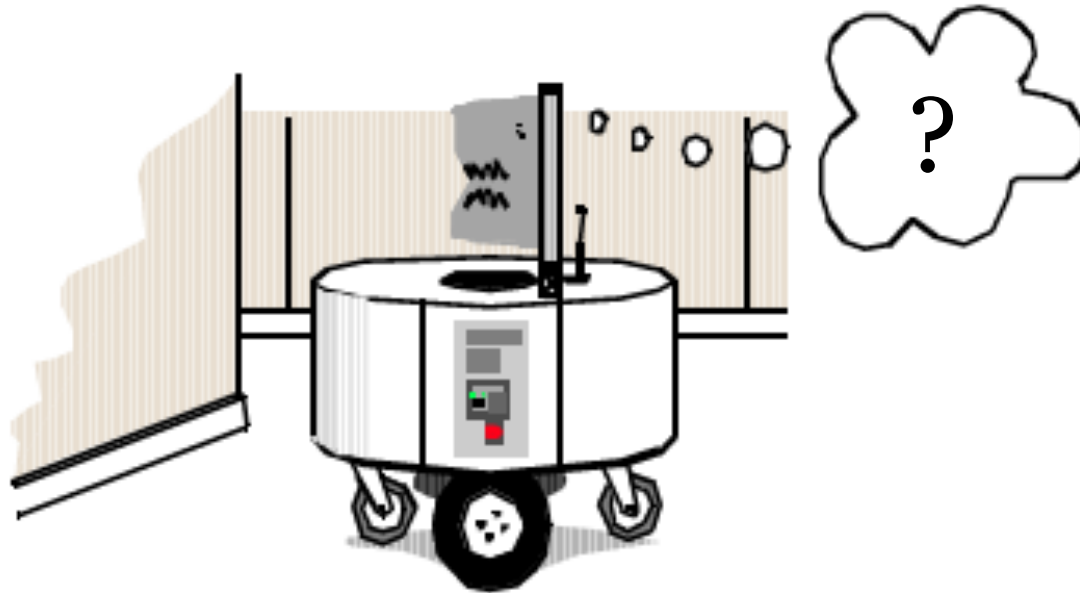
Décision: Il faut tourner à droite

Action: Régler la vitesse de la roue gauche et de la roue droite, par exemple:

$$\omega_G(t) = 0.1 \text{ rad/s}, \quad \omega_D(t) = 0.5 \text{ rad/s}, \quad t \in [t_a, t_b]$$

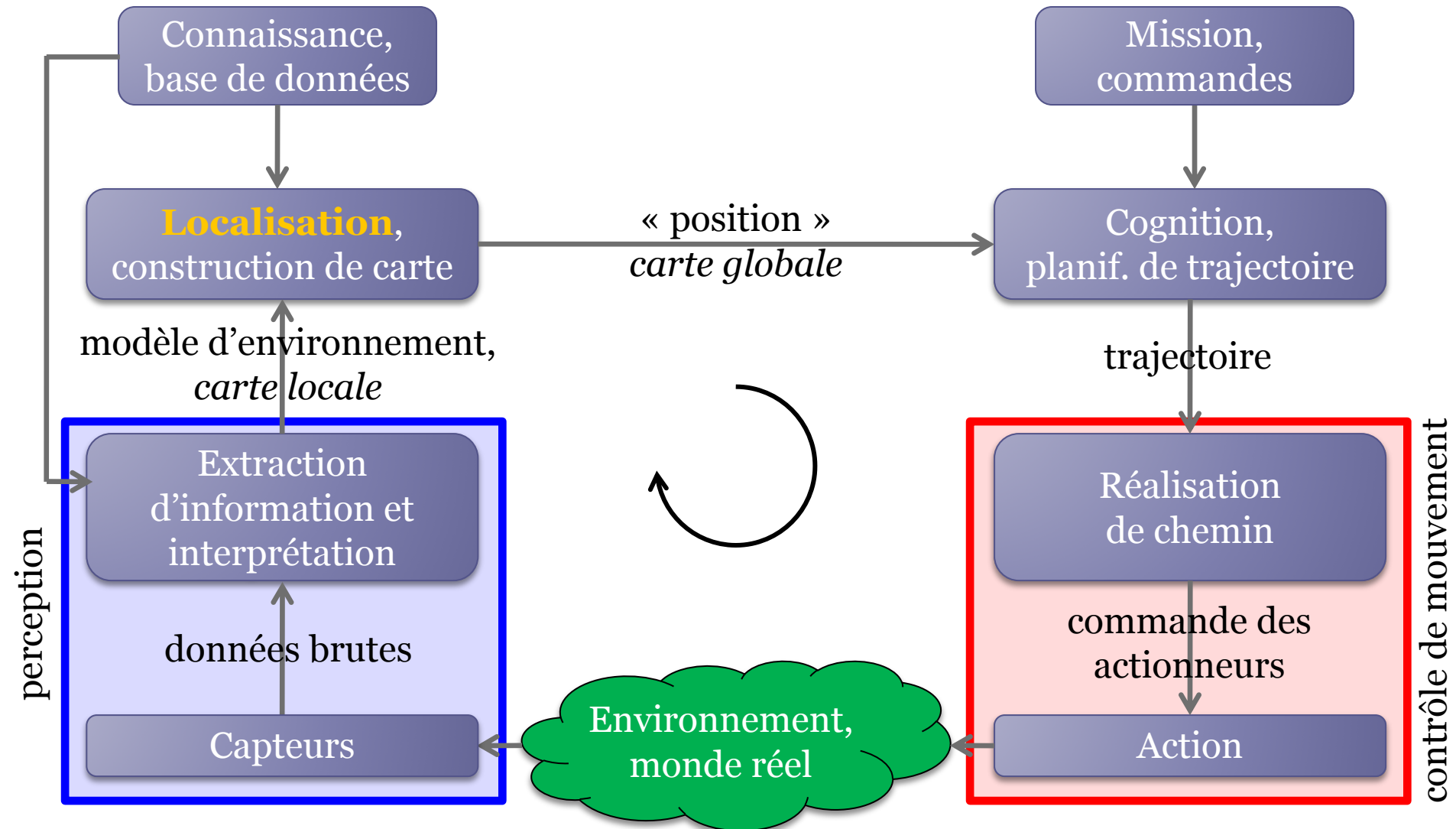
Introduction

- **Localisation:** il a reçu la plus grande attention de la part des chercheurs au cours des dernières décennies*

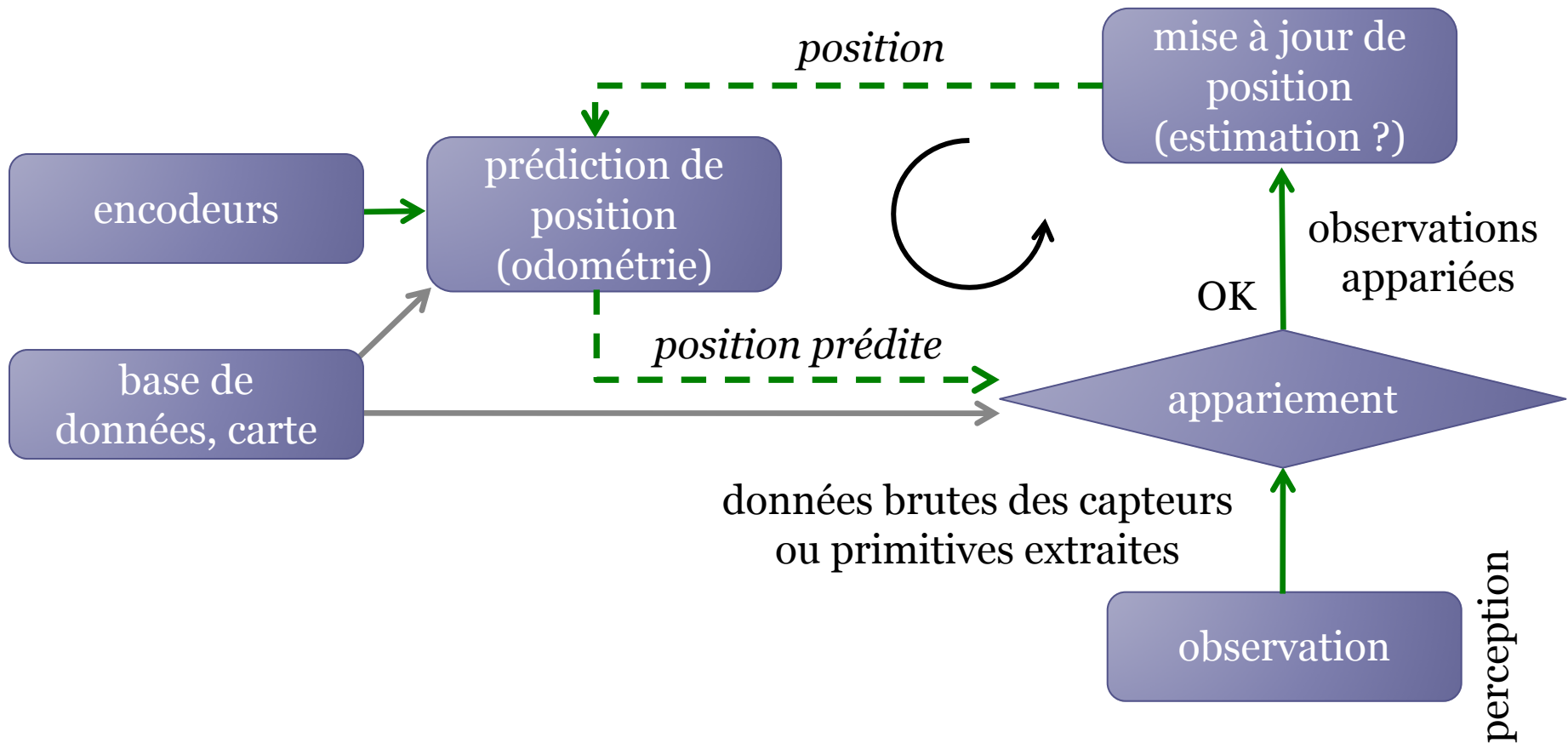


**Springer Handbook of Robotics*, B. Siciliano, O. Khatib (Éditeurs), 2^e éd., 2016, (2227 pages)

Où intervient la localisation ?



- Schéma général de localisation d'un robot mobile

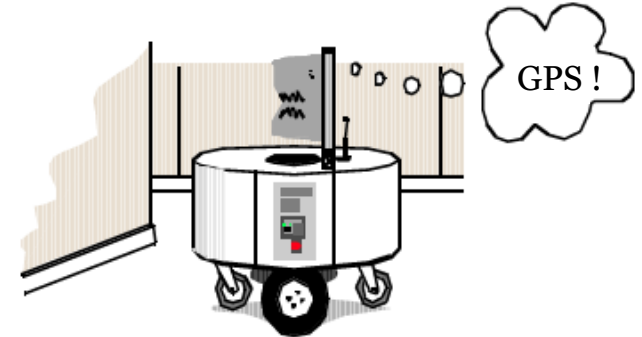


Les défis de la localisation

- Le bruit
 - de perception (capteurs)
 - d'actionnement (actionneurs)

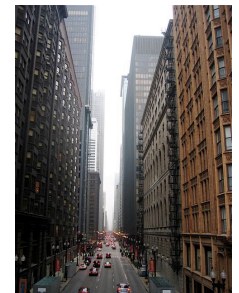
- L'aliasing perceptuel
 - Il définit le fait que deux lieux distincts peuvent avoir la même apparence

Les défis de la localisation



- Intuitivement

- « GPS précis » : solution ultime pour la localisation
 - Il informe le robot de sa position exacte
 - Intérieur et extérieur
 - Réponse immédiate à la question « où suis-je ? »
- Mais ... un tel capteur n'existe pas !
 - GPS précis à *quelques mètres* (GPS RTK plus précis, mais cher)
 - GPS est inadapté pour des environnements à taille humaine et encore moins pour les micro- et les nano-robots
 - GPS ne fonctionne pas à l'*intérieur* ou dans un environnement encombré (« canyons urbains »)



GPS = *Global Positioning System* (constellation de 24 satellites orbitant à 20000 km)

Les défis de la localisation

- Au delà des limitations du GPS
 - Localisation est *plus* que la connaissance de position absolue



Exemple: *robot interactif*

- Éventuellement besoin d'une position absolue
- Position relative aux humains aussi importante
- La tâche de localisation doit inclure:
 - Perception (multi-capteurs) pour identifier les humains
 - Calcul de sa position relative aux humains

Les défis de la localisation

- Atteindre un lieu particulier demande plus qu'une « simple » localisation
 - Étape de décision
 - Sélection de stratégie pour atteindre le but
- Besoin d'acquérir ou de créer un modèle de son environnement: une carte (cf. SLAM)
 - Aide le robot dans sa planification de trajectoire
- En résumé, la localisation c'est plutôt:
 - Construire une carte de l'environnement
 - Déterminer la position du robot relativement à cette carte

Les défis de la localisation

- Éléments cruciaux de cette localisation:
 - Les capteurs du robot
 - Les actionneurs du robot
- Les capteurs et les actionneurs d'un robot sont *inexactes* et *incomplets*
 - Une mesure parfaite n'existe pas
 - Tout l'environnement ne peut être mesuré d'un coup

Le bruit de perception

- Perception
 - Capteurs: fondamentaux
 - Leur degré de discrimination de l'environnement est critique
- *Le bruit des capteurs*
 - Inconsistance de mesures dans un même environnement
 - Source:
 - Primitives de l'environnement non perçues (ex. droites)
 - Négligence de ces primitives

Le bruit de perception



- Exemple 1:
 - Système de vision pour localisation à l'intérieur
 - Utilisation des couleurs détectées par une caméra CCD
 - Soleil caché: illumination différente
 - Valeurs de teinte non constantes
 - Couleurs « bruitées » pour le robot
 - Teinte inutilisable sauf si:
 - Connaissance de la position du soleil
 - Connaissance de la forme et de la position des nuages
 - Autres: flou (mise au point, mouvement du robot)

Le bruit de perception

- Exemple 2:

- Sonar: télémètre à ultrasons

- Rapport signal sur bruit très pauvre

- Ultrasons émis vers une surface lisse, mal orientée

- Signal non reflété vers le capteur, pas d'écho de retour

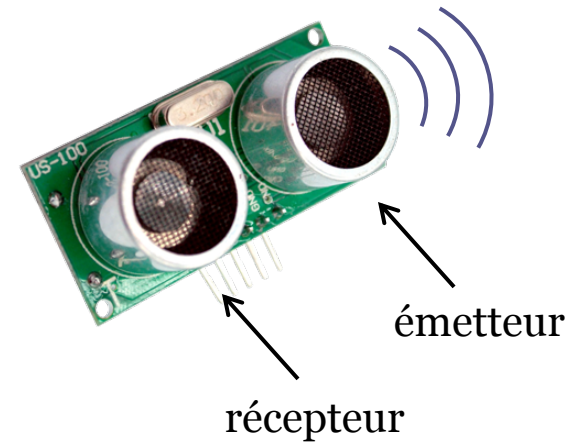
- Une petite quantité d'énergie peut revenir malgré tout

- Si le seuil sur l'amplitude d'énergie reçue en est proche

- Parfois le sonar permettra de détecter l'objet ... parfois non !

- Par conséquent :

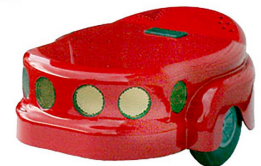
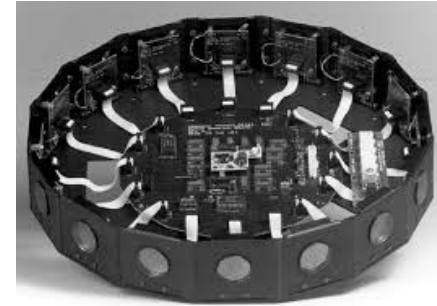
Pour le robot, un environnement inchangé peut donner deux lectures différentes pour un sonar !



Le bruit de perception

▫ Plusieurs sonars

- Sur les robots mobiles, en général:
 - Entre 8 et 48 sonars sur un même support (anneau)
- Interférences
 - de l'environnement (interférences « multi-chemins »)
 - entre émetteurs d'ultrasons (un signal émis par un sonar est reçu par un autre)
 - Elles peuvent engendrer d'importantes erreurs de mesure (sous-estimation), mais elles se produisent rarement, moins de 1% du temps



En résumé:

- Le bruit de perception réduit l'information utile des mesures d'un capteur
- La solution est de **prendre en compte plusieurs mesures**, en les fusionnant *au cours du temps* ou *avec d'autres capteurs* afin d'améliorer globalement l'information perceptuelle du robot

L'aliasing perceptuel

- Second inconvénient des capteurs
 - Le contenu informationnel des mesures peut être *très faible* (ça exacerbe les problèmes de perception et de localisation)
 - Problème peu intuitif pour les humains
 - Les humains ne rencontrent que rarement ce phénomène
 - La perception humaine, particulièrement la vision, tend à percevoir des *informations uniques dans chaque état local unique*
 - Chaque lieu différent semble différent
 - Mais pas facile de:
 - Se déplacer dans un bâtiment inconnu dans le noir
 - Se déplacer dans un labyrinthe à taille humaine (pas d'unicité visuelle)



L'aliasing perceptuel

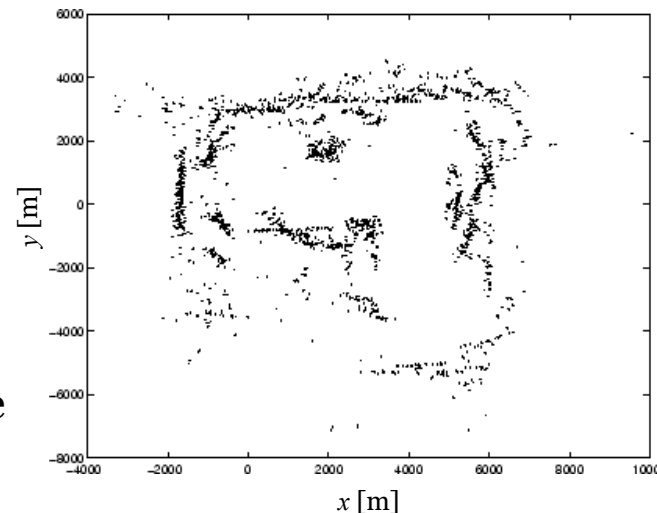
- En robotique
 - L'aliasing est la norme et non l'exception !

Exemple: sonars ou proximètres IR

- Champ de perception étroit
- Information de profondeur dans une seule direction
- Pas d'info comme couleur, texture, rigidité de surface
- Même avec plusieurs de ces capteurs, plusieurs états dans l'environnement donnent les mêmes mesures
- Formellement:

Il y a une association n à 1 des états de l'environnement vers la perception du robot

Carte basée
sonars



L'aliasing perceptuel

Un exemple concret:

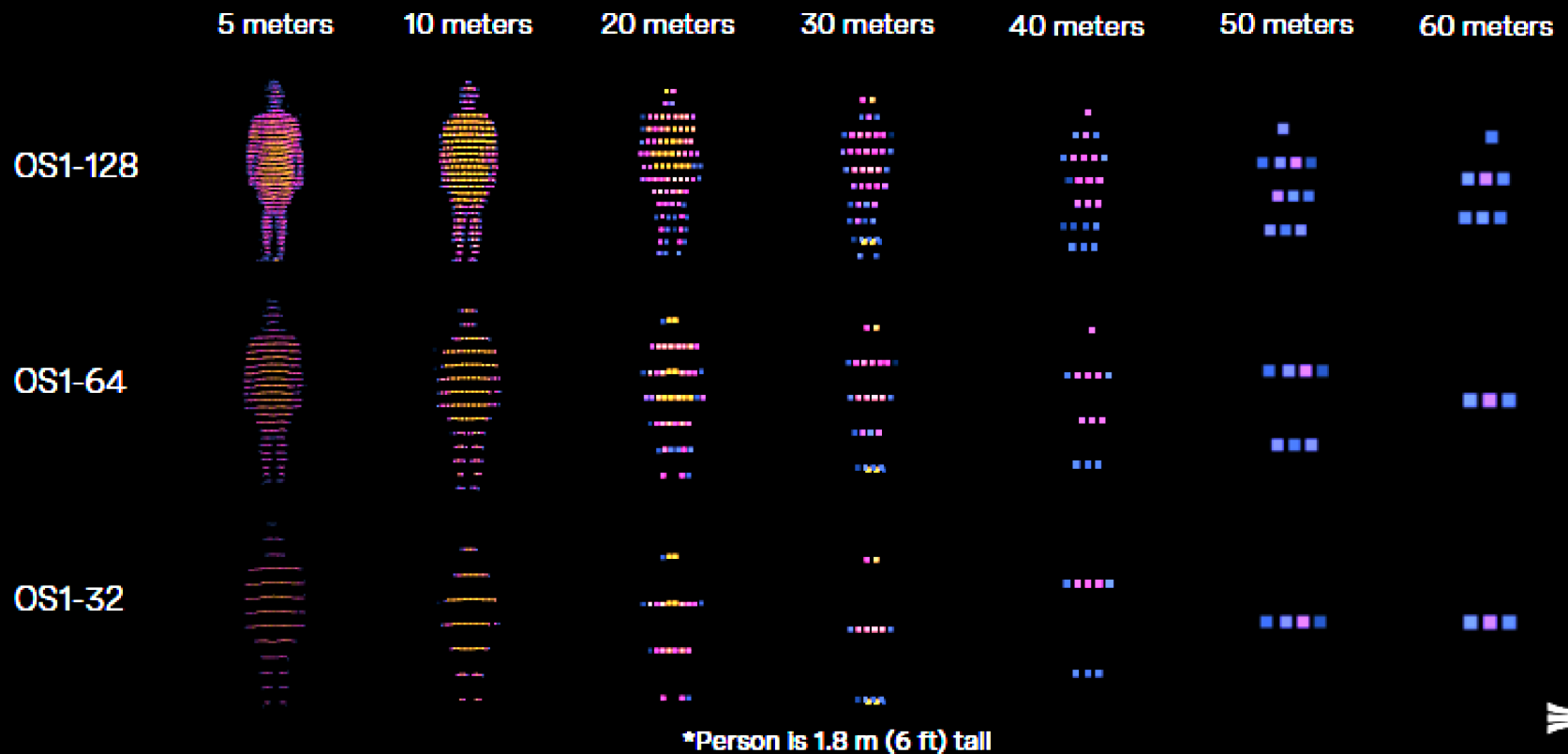
- Distinguer les humains des objets
- Le robot détecte un obstacle face à lui
 - Doit-il dire « excusez-moi » car l'obstacle peut être un humain en mouvement ?
... ou le robot doit-il planifier un chemin autour de l'obstacle car ce peut être un objet ?
- Pour le(s) sonar(s) seul(s), ces deux états souffrent d'aliasing et leur discrimination est impossible !



Être humain
ou statue ?

L'aliasing perceptuel

Résolution de plusieurs Lidars *Ouster* : www.ouster.com

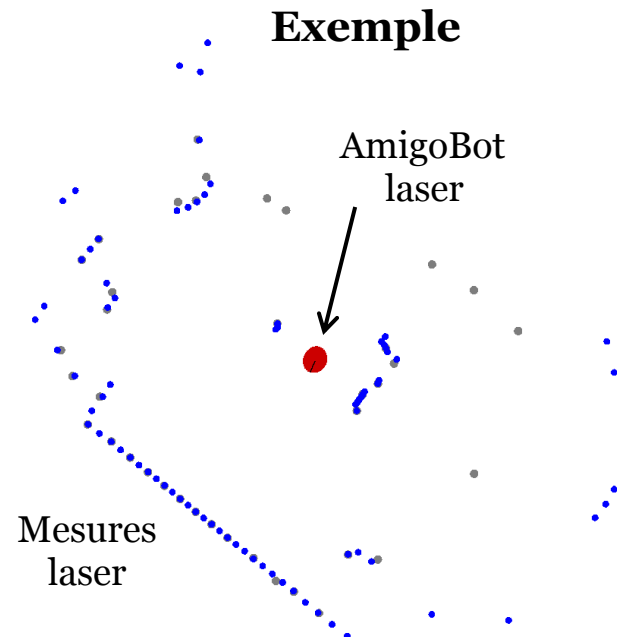


L'aliasing perceptuel

- En résumé l'aliasing perceptuel pose le problème suivant

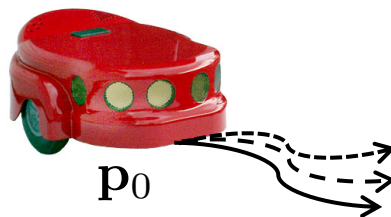
Même avec des mesures non bruitées, la quantité d'information qui vient des capteurs est *trop faible* pour identifier la pose du robot à partir d'un seul percept

- « **Solution** »
 - Localisation basée capteurs et mesures *multiples*



Le bruit d'actionnement

- Les difficultés de la localisation ne reposent pas uniquement sur les capteurs
- Les actionneurs sont aussi « bruités »
 - À cause de la perception ou non
 - Une même action ordonnée par le robot peut engendrer *différentes réalisations* ...
... même si du point de vue du robot, l'état initial avec l'action est parfaitement connu



Le bruit d'actionnement

- Les actionneurs d'un robot mobile induisent une incertitude sur l'état futur
- Pour un robot, se déplacer *accroît* son incertitude
- La *couche décisionnelle* peut minimiser cet effet
 - Planification et/ou interprétation adaptée
 - Retour des capteurs dans la boucle de commande

Le bruit d'actionnement

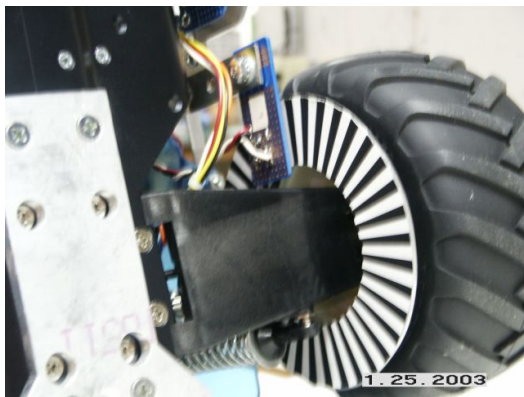
- Nature du bruit en robotique mobile
 - Du point de vue du robot
 - Impossibilité d'estimer sa propre position à partir de la connaissance de sa cinématique/dynamique
 - Vrai source d'erreur
 - Modèle incomplet de l'environnement
 - Type de surface non modélisé (rigidité, glissement, etc.)
 - Événements non envisagés (une personne pousse le robot, drone perturbé par une rafale de vent, etc.)

Le bruit d'actionnement

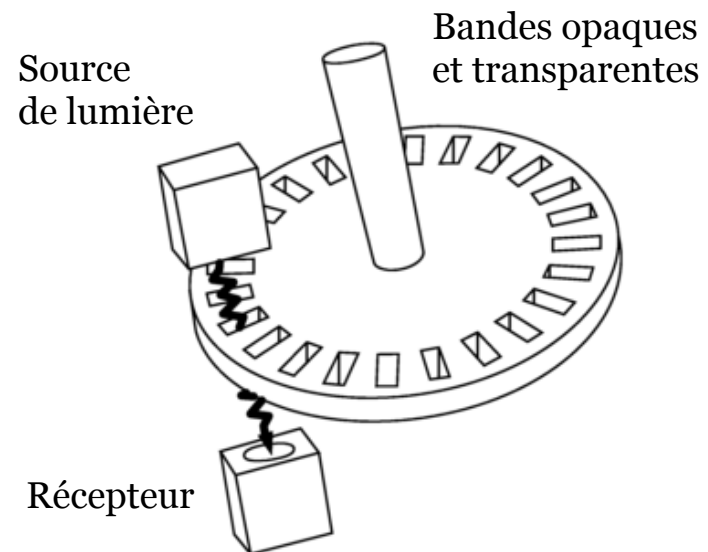
- Exemple : **l'odométrie**

L'*odométrie* est une technique permettant d'estimer la position et orientation d'un robot en mouvement. Le terme vient du grec *hodos* ("voyage") et *metron* ("mesure")

- Uniquement des capteurs sur les roues ...



Encodeur optique incrémental



Le bruit d'actionnement

- Exemple : **l'odométrie**

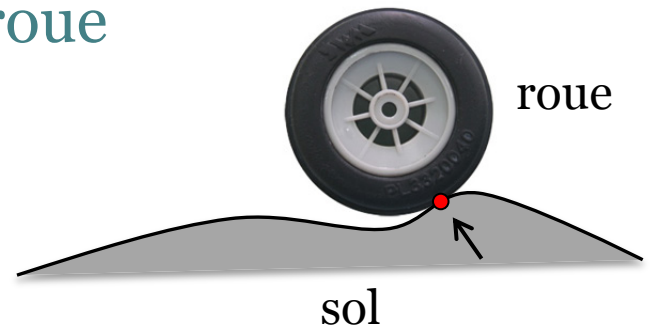
... ou couplés à un gyroscope/compas («dead reckoning»)

- Estimation de pose basée sur les capteurs proprioceptifs
- Mouvement intégré pour déduire la position
 - Intégration des erreurs de mesure
 - Accumulation d'erreur en position au cours du temps
- Besoin d'un autre mécanisme de localisation de temps en temps
 - ... sinon, la position estimée n'a vite plus de sens !

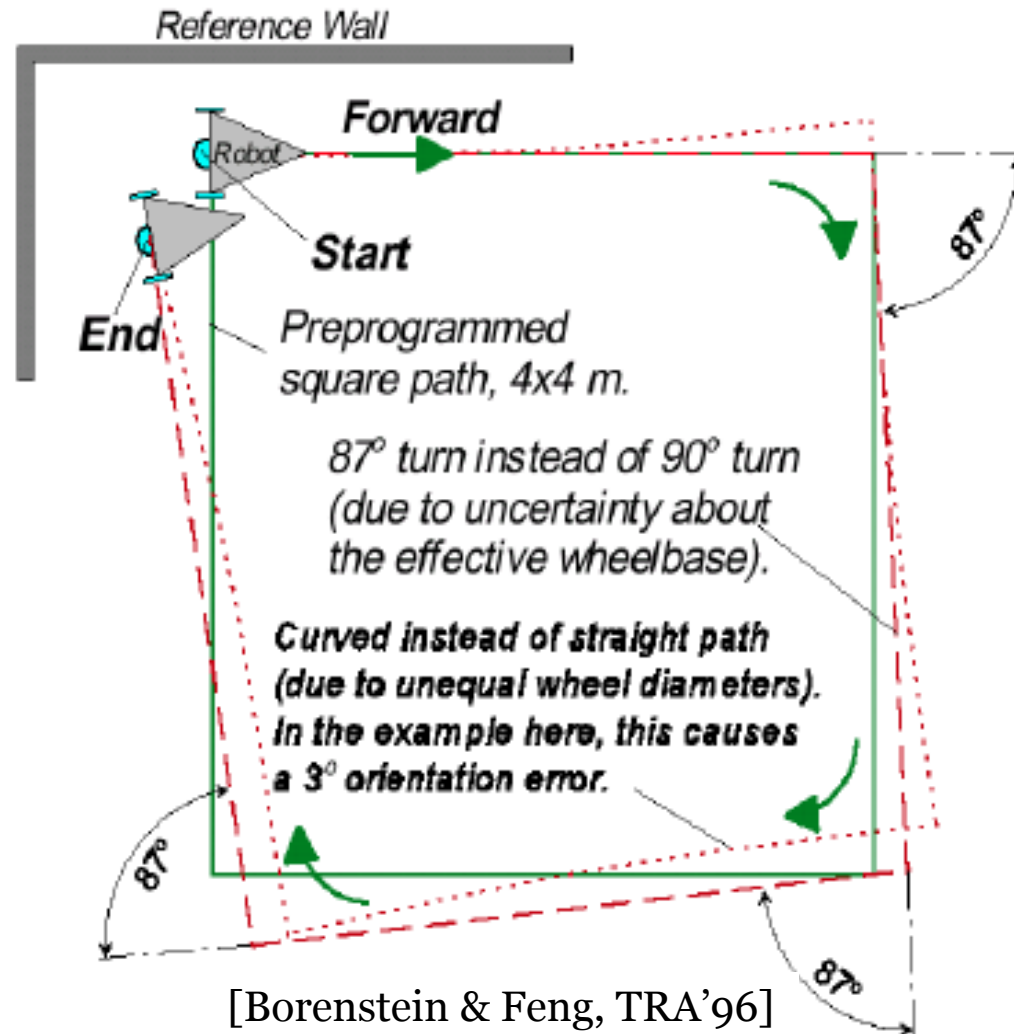
Le bruit d'actionnement



- Odométrie pure d'un *robot à conduite différentielle*
- Sources d'erreurs:
 - Résolution limitée pendant l'intégration (incréments de temps, résolution de mesure, etc.)
 - Mauvais alignement des roues (déterministe)
 - Incertitude sur le diamètre de la roue et/ou diamètre non constant (déterministe)
 - Variation du point de contact de la roue
 - Contact variable avec le sol (glisse, bosse, sol mou, etc.)



L'expérience du chemin carré unidirectionnel

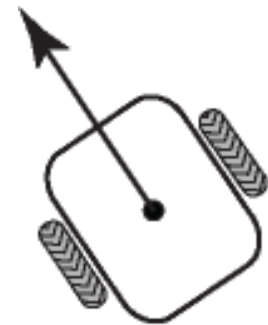


Le bruit d'actionnement

- Types de sources d'erreurs
 - *Déterministes* (systématiques)
 - Résolues (partiellement) par calibrage du système
 - *Non-déterministes* (aléatoires)
 - Erreurs résiduelles
 - Mènent à des incertitudes sur la pose au fur et à mesure

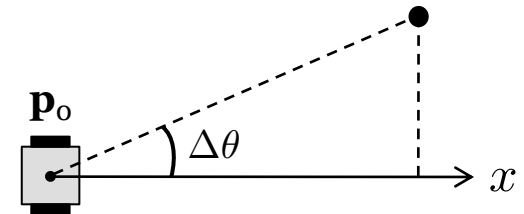
Le bruit d'actionnement

- Types géométriques d'erreurs (pour un robot à conduite différentielle)
 - Erreur en distance
 - Longueur du chemin intégré
 - Somme des mouvements de roue
 - Erreur en virage
 - Similaire à l'erreur en distance
 - Différence des mouvements de roue
 - Erreur de dérive (drift)
 - Une différence d'erreurs entre les roues mène à une erreur d'orientation du robot



Le bruit d'actionnement

- Sur de longues périodes
 - Erreurs de virage et de dérive plus importantes de l'erreur en distance
 - Contribution *non linéaire* à l'erreur de position
 - Soit un robot de pose initiale connue $\mathbf{p}_o = [x_o, y_o, \theta_o]^T$
 - Il se déplace en ligne droite selon l'axe x
 - Erreur en y pour un mouvement de d mètres: $d \sin(\Delta\theta)$
 - L'erreur devient très importante à mesure que l'erreur angulaire $\Delta\theta$ croît
 - Au cours du temps, l'erreur d'orientation par rapport au repère d'origine croît rapidement et ainsi l'incertitude sur la pose du robot



Objectif: établir un **modèle d'erreur** pour l'exactitude odométrique d'un robot et étudier la *propagation de l'erreur* au fil du temps

Plan du chapitre

• Introduction et défis **Partie 1**

• Odométrie **Partie 2**

• Localisation par filtre de Kalman **Partie 3**

• Autres techniques de localisation **Partie 4**

Partie 2 : Odométrie

Notation

$\alpha, \beta, M \in \mathbb{R}$ scalaires (nombres réels)

$\mathbf{u} \in \mathbb{R}^n$ vecteur colonne de dimension n

$\mathbf{A} \in \mathbb{R}^{n \times n}$ matrice carrée avec n lignes et n colonnes

$\lambda_i(\mathbf{A}) \in \mathbb{C}$ valeur propre i de la matrice $\mathbf{A} \in \mathbb{R}^{n \times n}$

$\mathbf{I}_n \in \mathbb{R}^{n \times n}$ matrice identité $n \times n$

$\mathbf{0}_{n \times m} \in \mathbb{R}^{n \times m}$ matrice de zéros $n \times m$

$\text{diag}(a_1, a_2, \dots, a_n) = \begin{bmatrix} a_1 & 0 & \cdots & 0 \\ 0 & a_2 & \cdots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & a_n \end{bmatrix}$ matrice diagonale $n \times n$

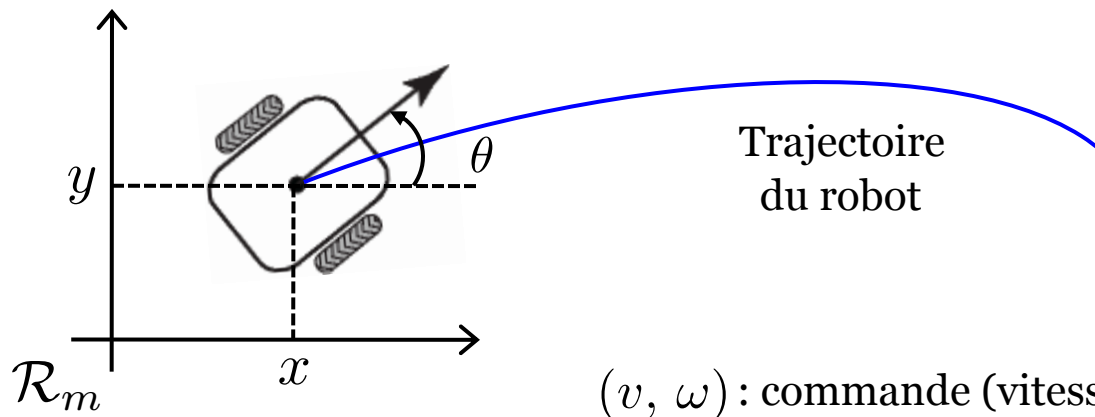
$\text{blkdiag}(\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_n)$ matrice par blocs diagonales (\mathbf{A}_i est une matrice carrée)

Modèle d'erreur pour l'exactitude odométrique

- Généralement la pose (position et orientation) d'un robot mobile est représentée par le vecteur:

$$\mathbf{p} = [x, y, \theta]^T$$

- Pour un robot de **type unicycle** (à conduite différentielle), la pose est estimée en partant d'une *pose connue* en intégrant le mouvement (somme des incréments des distances parcourues)



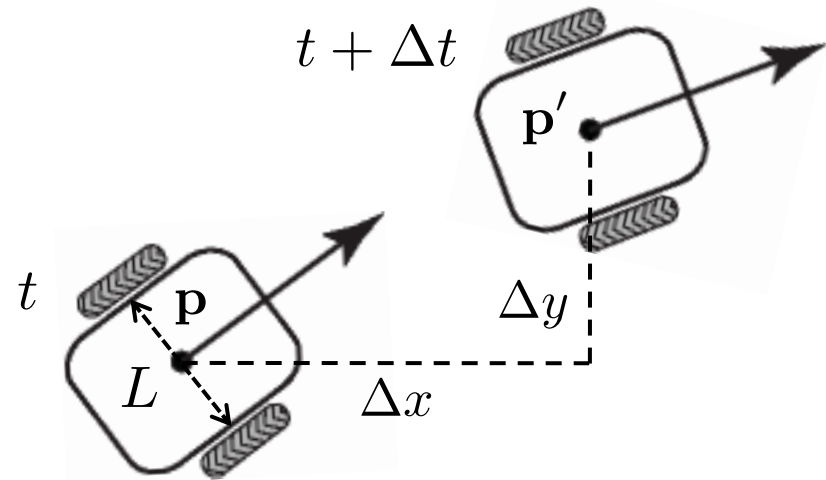
(v, ω) : commande (vitesse longitudinale et angulaire)

Modèle d'erreur pour l'exactitude odométrique

- Dans un système discret avec une *période d'échantillonnage* fixée, les incréments de distances parcourues sont:

$$\begin{cases} \Delta x = \Delta s \cos(\theta + \Delta\theta/2) \\ \Delta y = \Delta s \sin(\theta + \Delta\theta/2) \\ \Delta\theta = \frac{\Delta s_d - \Delta s_g}{L} \end{cases}$$

$$\text{où } \Delta s = \frac{\Delta s_d + \Delta s_g}{2}$$



$(\Delta x, \Delta y, \Delta\theta)$ Chemin parcouru pendant le dernier intervalle de temps $[t, t + \Delta t]$

$\Delta s_d, \Delta s_g$ Distance parcourue par la *roue droite* et *gauche*, respectivement

L Distance entre les deux roues du robot (largeur de l'essieu)

Modèle d'erreur pour l'exactitude odométrique

- La mise à jour de la pose en \mathbf{p}' se fait par:

$$\mathbf{p}' = \begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \mathbf{p} + \begin{bmatrix} \Delta s \cos(\theta + \Delta\theta/2) \\ \Delta s \sin(\theta + \Delta\theta/2) \\ \Delta\theta \end{bmatrix}$$

- En utilisant les relations pour Δs et $\Delta\theta$ vues précédemment, on obtient les équations de base de mise à jour de pose odométrique pour un robot unicycle:

$$\mathbf{p}' = \mathbf{f}(x, y, \theta, \Delta s_d, \Delta s_g) = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \frac{\Delta s_d + \Delta s_g}{2} \cos\left(\theta + \frac{\Delta s_d - \Delta s_g}{2L}\right) \\ \frac{\Delta s_d + \Delta s_g}{2} \sin\left(\theta + \frac{\Delta s_d - \Delta s_g}{2L}\right) \\ \frac{\Delta s_d - \Delta s_g}{L} \end{bmatrix}$$

Modèle d'erreur pour l'exactitude odométrique

- Ces équations n'ont d'intérêt que pour de *faibles déplacements*
- Plus le déplacement entre deux instants sera important, plus l'estimation de position sera *grossière* vis-à-vis de la réalité
- À cause de l'augmentation de l'erreur d'orientation et des erreurs de mouvement réalisées pendant l'incrément $(\Delta s_d, \Delta s_g)$, *l'erreur en position augmente avec le temps*

-
- Etablissons un modèle d'erreur pour la pose intégrée p' pour obtenir la *matrice de covariance*

$$\Sigma_{p'} \in \mathbb{R}^{3 \times 3}$$

de l'estimé odométrique de la pose

- Pour ce faire, on va supposer la connaissance d'une *matrice de covariance initiale*:

$$\Sigma_p \in \mathbb{R}^{3 \times 3}$$

Modèle d'erreur pour l'exactitude odométrique

- En admettant que \mathbf{p} et $\Delta_{dg} = [\Delta s_d, \Delta s_g]^T$ ne sont pas corrélés et que la fonction \mathbf{f} est bien approchée par la décomposition de Taylor au 1^{er} ordre, on peut appliquer *la loi de propagation de l'erreur**, obtenant:

$$\Sigma_{\mathbf{p}'} = \nabla_{\mathbf{p}} \mathbf{f} \cdot \Sigma_{\mathbf{p}} \cdot (\nabla_{\mathbf{p}} \mathbf{f})^T + \nabla_{\Delta_{dg}} \mathbf{f} \cdot \Sigma_{\Delta} \cdot (\nabla_{\Delta_{dg}} \mathbf{f})^T$$

- La matrice $\Sigma_{\mathbf{p}}$ est directement égale à la $\Sigma_{\mathbf{p}'}$ précédente (et elle peut être calculée en spécifiant une valeur initiale)
- Étant donnée \mathbf{f} on peut calculer les deux matrices jacobiennes $\nabla_{\mathbf{p}} \mathbf{f}$, $\nabla_{\Delta_{dg}} \mathbf{f}$:

$$\nabla_{\mathbf{p}} \mathbf{f} = \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial x} & \frac{\partial \mathbf{f}}{\partial y} & \frac{\partial \mathbf{f}}{\partial \theta} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\Delta s \sin(\theta + \Delta\theta/2) \\ 0 & 1 & \Delta s \cos(\theta + \Delta\theta/2) \\ 0 & 0 & 1 \end{bmatrix}$$

$$\nabla_{\Delta_{dg}} \mathbf{f} = \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial \Delta s_d} & \frac{\partial \mathbf{f}}{\partial \Delta s_g} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \cos(\theta + \Delta\theta/2) - \frac{\Delta s}{2L} \sin(\theta + \Delta\theta/2) & \frac{1}{2} \cos(\theta + \Delta\theta/2) + \frac{\Delta s}{2L} \sin(\theta + \Delta\theta/2) \\ \frac{1}{2} \sin(\theta + \Delta\theta/2) + \frac{\Delta s}{2L} \cos(\theta + \Delta\theta/2) & \frac{1}{2} \sin(\theta + \Delta\theta/2) - \frac{\Delta s}{2L} \cos(\theta + \Delta\theta/2) \\ 1/L & -1/L \end{bmatrix}$$

* On rappelle que: $\text{Var}[\mathbf{f}(\mathbf{X})] \simeq \mathbf{J}_{\mathbf{f}}(\mathbf{X}_0) \text{Var}[\mathbf{X}] \mathbf{J}_{\mathbf{f}}^T(\mathbf{X}_0)$, $\mathbf{X}_0 = \mathbb{E}[\mathbf{X}]$

Modèle d'erreur pour l'exactitude odométrique

- Pour l'incrément de mouvement des roues $(\Delta s_d, \Delta s_g)$, la matrice Σ_Δ doit être connue aussi:

$$\Sigma_\Delta = \text{COV}(\Delta s_d, \Delta s_g) = \begin{bmatrix} k_d |\Delta s_d| & 0 \\ 0 & k_g |\Delta s_g| \end{bmatrix}$$

où

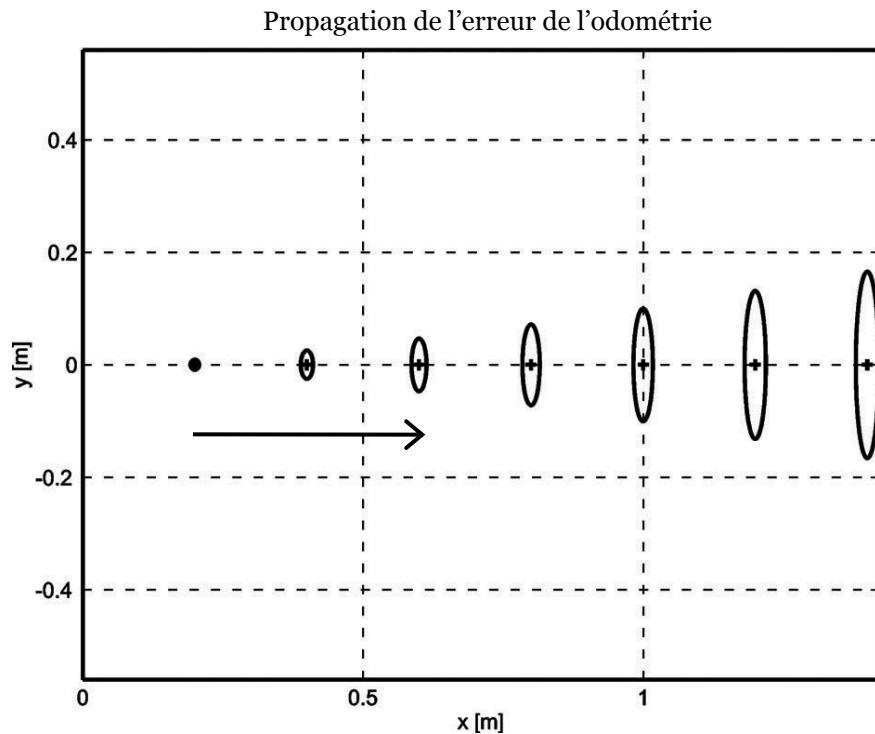
k_d, k_g sont des constantes positives représentant les paramètres non déterministes du *moteur* et de l'*interaction de la roue* (droite et gauche) *avec le sol*

Deux hypothèses sont englobées dans l'expression précédente:

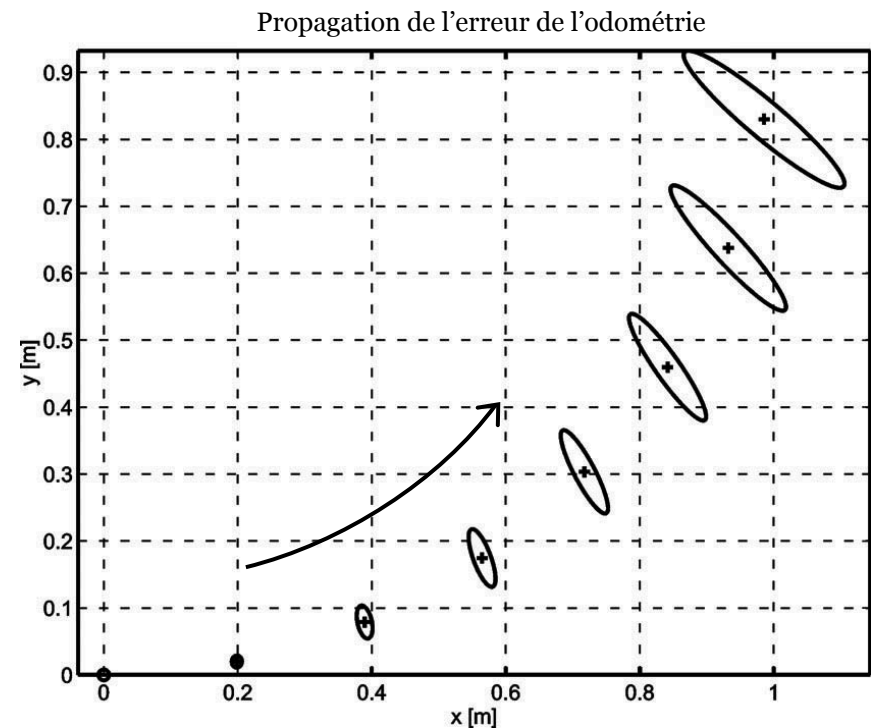
- Les erreurs de chaque roue sont *indépendantes*
- Les variances des erreurs sont *proportionnelles au module des distances parcourues*
- Ces hypothèses, bien que imparfaites, sont satisfaisantes dans la plupart des cas en environnement structuré. Les *sources d'erreur* sur le mouvement sont: roue déformée, glissement, sol non lisse, encodeurs imprécis, etc.
- Les valeurs des constantes k_d, k_g dépendent du robot et de l'environnement, et elles sont estimées *expérimentalement*

Modèle d'erreur pour l'exactitude odométrique

- Les figures montrent comment *croît* typiquement l'incertitude sur la position d'un robot au cours du temps (l'*incertitude de l'orientation* n'est pas représentée dans les figures, bien que son effet peut être observé indirectement)
- Les résultats ont été obtenus en appliquant le modèle de propagation de l'erreur précédent et en représentant la matrice de covariance par une *ellipse d'incertitude* (grâce à la décomposition spectrale de $\Sigma_{p'}$)



Mouvement rectiligne

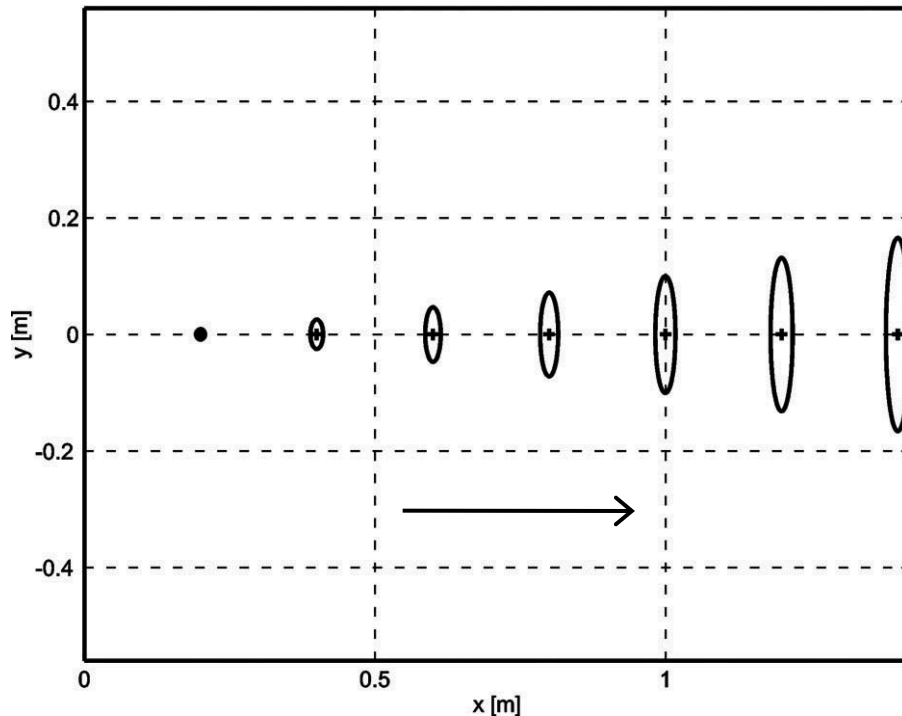


Mouvement circulaire (rayon const.)

Remarques

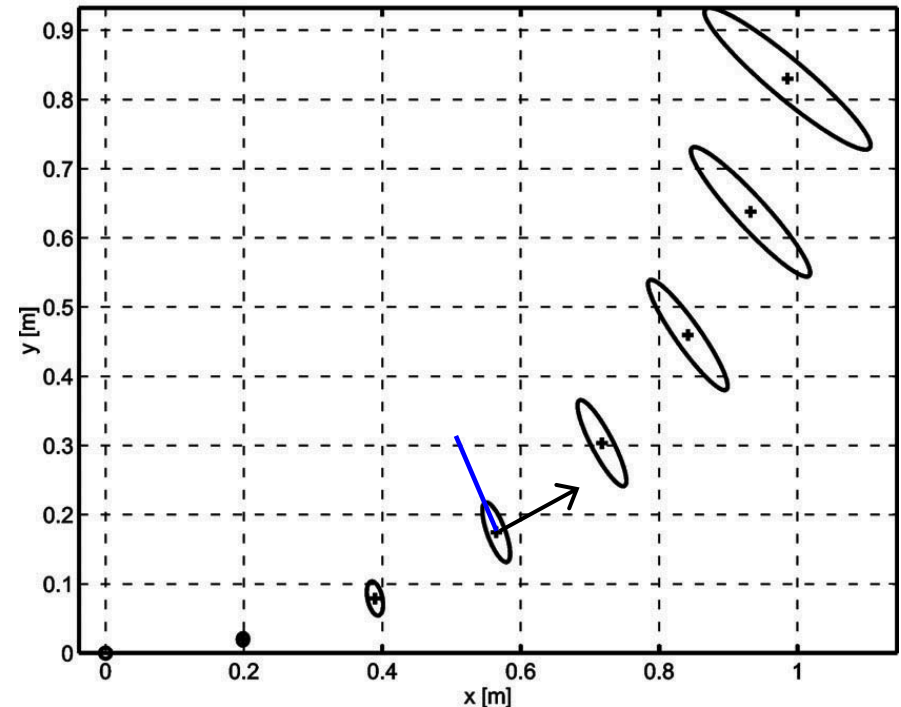
- L'incertitude dans la *direction perpendiculaire* au mouvement croît plus rapidement que dans le sens du mouvement. Cela résulte de l'intégration de l'erreur en orientation
- L'axe principal de l'ellipse d'incertitude *ne reste généralement pas* exactement perpendiculaire à la direction du mouvement

Propagation de l'erreur de l'odométrie



Mouvement rectiligne

Propagation de l'erreur de l'odométrie



Mouvement circulaire (rayon const.)

Calcul des ellipses d'incertitude

La *decomposition d'une matrice carrée en valeurs et vecteurs propres* est un outil de base de l'algèbre linéaire que l'on trouve dans de nombreuses disciplines

Définition

Un vecteur non nul $\mathbf{u} \in \mathbb{R}^n$ est un *vecteur propre* d'une matrice $\mathbf{A} \in \mathbb{R}^{n \times n}$ si et seulement si il existe un scalaire λ tel que :

$$\mathbf{A} \mathbf{u} = \lambda \mathbf{u}$$

où λ est appelé *valeur propre* associée à \mathbf{u} .

On détermine les valeurs propres et vecteurs propres en résolvant l'équation:

$$(\mathbf{A} - \lambda \mathbf{I}_n) \mathbf{u} = \mathbf{0}$$

Remarque

Les valeurs propres d'une *matrice symétrique réelle* $n \times n$ sont *réelles*. Toute matrice symétrique réelle admet n vecteurs propres *linéairement indépendants*. De plus, ces vecteurs peuvent être choisis de façon à être *orthogonaux* deux à deux et *unitaires*

Calcul des ellipses d'incertitude

Soit $\mathbf{A} \in \mathbb{R}^{n \times n}$ une matrice symétrique

En définissant les deux matrices $n \times n$ suivantes:

$$\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n), \quad \mathbf{U} = [\mathbf{u}_1 \ \mathbf{u}_2 \ \cdots \ \mathbf{u}_n]$$

où

$\lambda_1, \lambda_2, \dots, \lambda_n$ valeurs propres de la matrice \mathbf{A} triées par ordre *décroissant* (à savoir $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$)

\mathbf{u}_1 vecteur propre associé à la valeur propre de \mathbf{A} la plus forte (à savoir λ_1)

on peut écrire:

$$\mathbf{A} \mathbf{U} = \mathbf{U} \mathbf{\Lambda}$$

Calcul des ellipses d'incertitude

Exemple:

$$\mathbf{A} = \begin{bmatrix} 1 & -2 & 4 \\ -2 & 5 & 3 \\ 4 & 3 & -1 \end{bmatrix}$$

Valeurs propres triées par ordre décroissant: $\{6.2768, 4.0937, -5.3706\}$

$$\mathbf{\Lambda} = \begin{bmatrix} 6.2768 & 0 & 0 \\ 0 & 4.0937 & 0 \\ 0 & 0 & -5.3706 \end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix} -0.1070 & 0.8118 & -0.5740 \\ 0.9386 & -0.1081 & -0.3278 \\ 0.3281 & 0.5738 & 0.7504 \end{bmatrix}$$

Commande *Matlab*: $[\mathbf{U}, \mathbf{\Lambda}] = \text{eigs}(\mathbf{A})$

Attention:

- Les valeurs propres sont triées par ordre *croissant en module*
- Les vecteurs propres sont choisis *unitaires*

Calcul des ellipses d'incertitude

- Il est pratique courante de représenter l'incertitude de localisation en deux dimensions (plan du sol) mais à *3 degrés de liberté*, par une **ellipse d'incertitude** autour de la position estimée
- La *position réelle* du robot, si les paramètres du modèle d'erreur ont été bien choisis, doit se trouver dans cette ellipse
- En appliquant la décomposition en valeurs/vecteurs propres à $\Sigma_{p'}$, on obtient les deux matrices U et Λ d'où:

$$\Sigma_{p'} U = U \Lambda$$

- En prenant le vecteur propre u_1 associé à la plus forte valeur propre λ_1 de $\Sigma_{p'}$ on obtient des informations sur l'*orientation* de l'ellipse d'incertitude

Remarque

Par définition, une matrice de covariance est une *matrice symétrique* et *semi-définie positive*:

$$\Sigma_{p'} = \Sigma_{p'}^T, \quad \Sigma_{p'} \succeq \mathbf{0}$$

donc, ses valeurs propres sont **réelles** et **non négatives**

Calcul des ellipses d'incertitude

- Les deux premières coordonnées $u_{1,x}$ et $u_{1,y}$ du vecteur \mathbf{u}_1 permettent de déduire l'orientation de l'ellipse dans le plan du sol:

$$\theta_e = \arctan\left(\frac{u_{1,y}}{u_{1,x}}\right)$$

- En pratique, on préférera l'utilisation de la fonction `atan2`, qui teste dans quel quart se trouve la position $(u_{1,x}, u_{1,y})$ pour que θ_e ait un signe cohérent
- La longueur des demi-axes de l'ellipse est définie par la racine carrée des deux valeurs propres de $\Sigma_{\mathbf{p}}$, *les plus fortes*, à savoir λ_1 et λ_2
- Il faut noter que λ_1 et λ_2 peuvent être interprétées comme des *variances*.
 - Par exemple, si les paramètres du modèle d'erreur sont corrects, pour assurer que la position réelle se situe dans l'ellipse d'incertitude avec une confiance de 99.7%, les demi-axes de celle-ci auront pour longueurs, $3\sqrt{\lambda_1}$ et $3\sqrt{\lambda_2}$, respectivement

En résumé ...

TD1

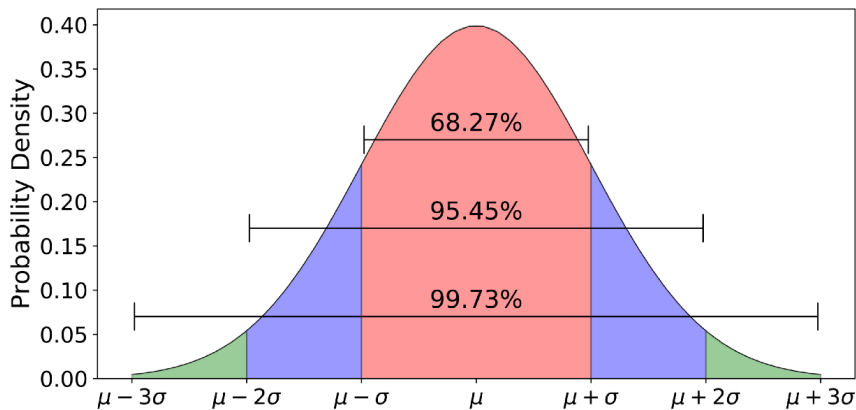
Ellipse d'incertitude 3σ

$$\theta_e = \arctan\left(\frac{u_{1,y}}{u_{1,x}}\right)$$

Remarque:

Si X est une variable aléatoire gaussienne,
 μ est l'espérance de la loi et σ est l'écart type:

$$X \sim \mathcal{N}(\mu, \sigma^2)$$



$$\Pr(\mu - \sigma \leq X \leq \mu + \sigma) \simeq 0.68269$$

$$\Pr(\mu - 2\sigma \leq X \leq \mu + 2\sigma) \simeq 0.95450$$

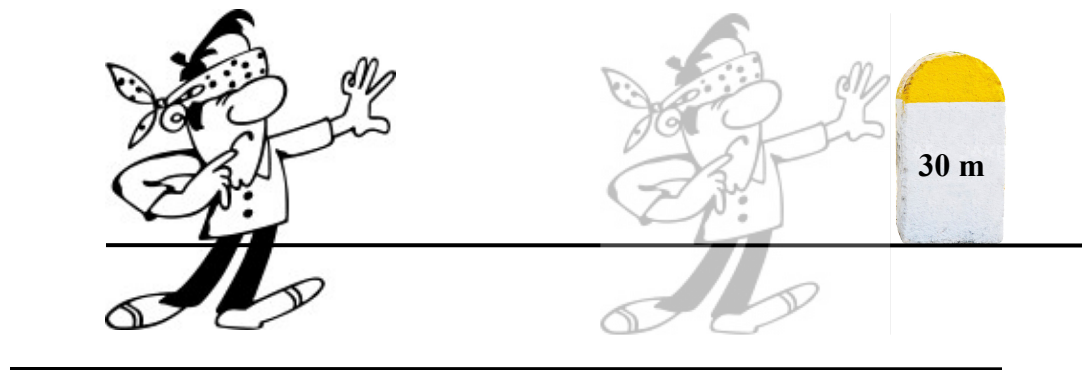
$$\Pr(\mu - 3\sigma \leq X \leq \mu + 3\sigma) \simeq 0.99730$$

$$\Pr(\mu - 4\sigma \leq X \leq \mu + 4\sigma) \simeq 0.99993$$

$$\Pr(\mu - 5\sigma \leq X \leq \mu + 5\sigma) \simeq 0.99999$$

Odométrie ... et les êtres humains ?

- Nous “connaissons” la longueur de nos jambes et les butées mécaniques des nos articulations depuis notre enfance
- La longueur de nos pas est donc “codée” dans notre cerveau (bien que nous ne sommes pas totalement conscients)
- Nous pouvons estimer assez précisément la distance parcourue
 - Même *sans retour visuel* (et *sans compter les pas*), la précision est plutôt élevée avec des distances jusqu'à 27 m [Andre & Rogers, 2006]



“Using verbal and blind-walking distance estimates to investigate the two visual systems hypothesis”, J. Andre, S. Rogers, *Percept. Psychophys.*, vol. 68, pp. 353–361, 2006