

Design 3

## Aperçu général sur le « Path Planning » en robotique

1

---

---

---

---

---

---

---

---

Design 3

## Plan

- ▶ La perception en robotique
- ▶ La notion de SLAM
- ▶ Planification pour le déplacement
  - ▶ La représentation de l'espace
  - ▶ La recherche d'un chemin dans l'espace
- ▶ Déplacement : le problème du contrôle

2

---

---

---

---

---

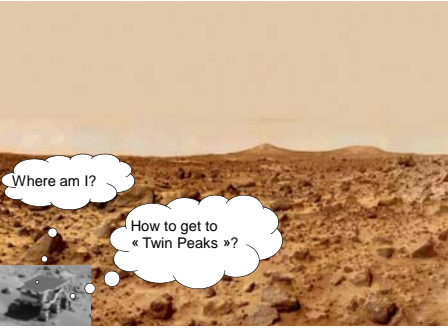
---

---

---

Design 3

## Perception



3

---

---

---

---

---

---

---

---

## Perception (2)

- ▶ Localisation et cartographie simultanées en robotique mobile [SLAM as Simultaneous Localization and Mapping]
- ▶ SLAM consiste à **estimer conjointement** la position du robot et les positions d'un ensemble d'objets remarquables de l'environnement : Ces objets sont immobiles et sont appelés amers.
- ▶ Pour le projet : Vision en vue de se localiser. La cartographie dépend de la représentation de l'environnement.

4

---

---

---

---

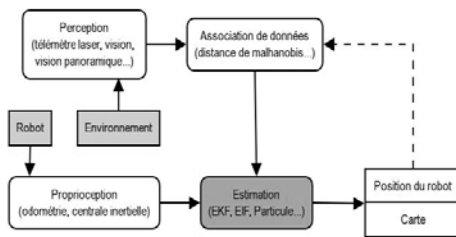
---

---

---

---

## SLAM



Algorithme générique pour SLAM

5

---

---

---

---

---

---

---

---

## SLAM (2)

- ▶ L'odométrie repose sur la mesure individuelle des déplacements des roues pour reconstituer le mouvement global du robot. En partant d'une position initiale connue et en intégrant les déplacements mesurés, on peut ainsi calculer à chaque instant la position courante du véhicule.
- ▶ Une centrale inertielle est un appareil de navigation de précision comportant des gyroscopes, des capteurs d'accélération et de vitesse angulaire et calculant en temps réel à partir de ces mesures l'évolution du vecteur vitesse et de la position du véhicule à bord duquel il est installé.
- ▶ EKF = Extended Kalman Filter

6

---

---

---

---

---

---

---

---

Design 3

## SLAM(2)

Étapes du SLAM : (A) = le robot observe la 1<sup>ère</sup> fois les trois amers (les angles des carrés); (B) = le robot se déplace, il connaît sa position avec incertitude; (C) = le robot réobserve les amers; (D) = la fusion de cette observation avec la carte construite précédemment permet de diminuer l'incertitude sur les 2 positions : celles des amers et celle du robot

---

---

---

---

---

---

---

---

Design 3

## Robot autonome

```

graph TD
    A[Environment Sensors] --> B[Task Planning]
    B --> C[Motion Planning  
Path Planning]
    C --> D[Controller]
    D --> E[Robot Actuators  
and Sensors]
    E --> F[Feedback]
    F --> D
  
```

---

---

---

---

---

---

---

---

Design 3

## Motion Planning

Continuous representation  
(configuration space and related spaces + constraints)

↓

Discretization  
(random sampling, criticality-based decomposition)

↓

Graph searching  
(blind, best-first, A\*)

---

---

---

---

---

---

---

---

# Représentation de l'espace de recherche

- Concept unificateur: L'espace de configuration Cspace qui se construit à partir de l'espace physique Workspace (W)
- Exemple:

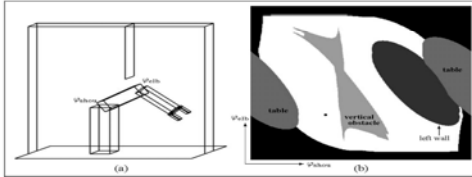


Figure 25.11 (a) Workspace representation of a robot arm with 2 DOF. The workspace is a box with a flat obstacle hanging from the ceiling. (b) Configuration space of the same robot. Only white regions in the space are configurations that are free of collisions. The dot in this diagram corresponds to the configuration of the robot shown on the left.

---

---

---

---

---

---

---

---

---

---

# Espace de configurations

- Qu'est ce que c'est ?
  - un ensemble de zones atteignables construites à partir des connaissances robot/environnement.
- Comment la créer ?
  - Abstraire le robot comme un point objet; ensuite agrandir les obstacles de façon à rendre compte de l'empreinte et des degrés de liberté du robot.

---

---

---

---

---

---

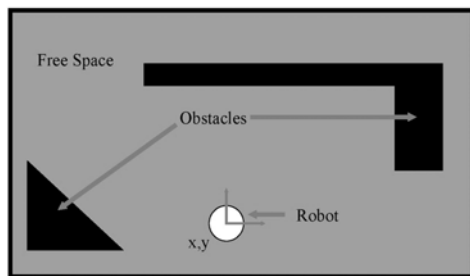
---

---

---

---

# Espace de configurations




---

---

---

---

---

---

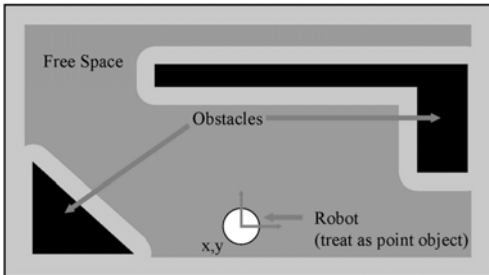
---

---

---

---

### Espace de configurations (2)




---

---

---

---

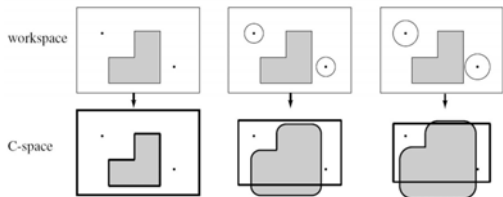
---

---

---

---

### Espace de configurations (3)




---

---

---

---

---

---

---

---

### Path Planning

- La planification de chemin peut être vue comme une recherche de séquences de configurations qui permettent au robot d'aller de START à GOAL.
- Un planificateur doit permettre de trouver la série de configurations qui minimise un critère donné tout en évitant les obstacles.

---

---

---

---

---

---

---

---

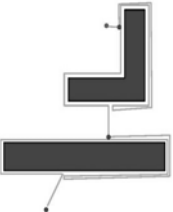
Design 3

## Path Planning: Exemple

But some computing power!

- known direction to goal
- otherwise local sensing

walls/obstacles & encoders



**"Bug 1" algorithm**

- 1) head toward goal
- 2) if an obstacle is encountered, circumnavigate it *and* remember how close you get to the goal
- 3) return to that closest point (by wall-following) and continue

Vladimir Lumelsky & Alexander Stepanov: *Algorithmica* 1984

16

---

---

---

---

---

---

---

---

---

---

Design 3

## Path Planning (suite)

- L'ensemble des configurations d'un robot forme un espace de recherche qui croit exponentiellement avec le nombre de degrés de libertés du robots (DOF)
- C'est la taille de l'espace de recherche qui crée la principale difficulté associée au path planning.
- On-line vs Offline Planificateurs

17

---

---

---

---

---

---

---

---

---

---

Design 3

## Off-Line vs Online Planificateurs

- Off-line : Considérer le problème au complet et lui trouver une solution complète avant de commencer à exécuter les actions dans le monde réel.
- Online : entrelacer computation et action : l'agent exécute une action, observe l'environnement et élabore une prochaine action.

18

---

---

---

---

---

---

---

---

---

---

## Un « bon » Path

- ▶ Il y a différentes manières de mesurer un path :
  - ▶ Temps
  - ▶ La distance parcourue
  - ▶ Coût
  - ▶ Distances relativement aux obstacles
  - ▶ Une combinaison de différents paramètres
  - ▶ Etc...

---

---

---

---

---

---

---

---

## Représentation de l'ELC

- ▶ Méthodes cartes réutilisables (roadmap)
  - ▶ Connectivité de l'ensemble libre de configurations (ELC) sous la forme d'un graphe. Trouver ensuite le chemin « optimal » dans ce graphe.
- ▶ Méthodes de décomposition en cellules
  - ▶ Décomposer l'ensemble libre des configurations en un ensemble de régions (cellules) ne se recouvrant pas. L'adjacence de ces cellules forme un graphe. Trouver ensuite le chemin « optimal » dans ce graphe.
- ▶ Méthodes champ de potentiel
  - ▶ Utiliser des champs de potentiel répulsifs (obstacles) et attractifs (buts). Le gradient qui combine ces deux types de champs est utilisé pour guider le robot.

---

---

---

---

---

---

---

---

## Cartes réutilisables

- ▶ Méthode des sous-buts
- ▶ Graphe de visibilité
- ▶ Le diagramme de Voronoi
- ▶ Les cartes réutilisables probabilistes

---

---

---

---

---

---

---

---

# Méthode des sous-buts

- ▶ Générer une liste de configurations atteignables à partir de S;
- ▶ L'atteignabilité d'une configuration à partir de la prochaine est faite via un planificateur local, "local operator"
- ▶ Au début, le local opérateur ('Go Diagonal') cherche à atteindre G, s'il ne peut pas; une liste de configurations candidates est générée, ces listes forment les sous-buts.

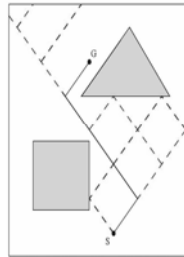


Figure 2.5: The Sub-goal method. Here, a 'go diagonal' local operator is used to generate sub-goals, eventually leading to the actual goal.

---

---

---

---

---

---

---

---

---

---

---

---

# Graphe de visibilité

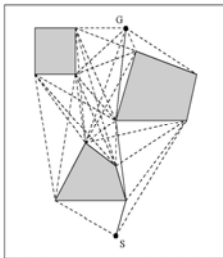


Figure 2.6: A Visibility graph with three obstacles.

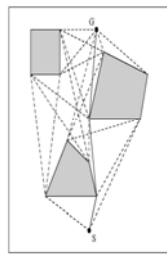


Figure 2.7: A Tangent graph with the same obstacles as Figure 2.6.

---

---

---

---

---

---

---

---

---

---

---

---

# Diagramme de Voronoi

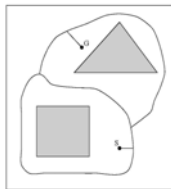


Figure 2.8: A Voronoi diagram. The curves are points of equidistance from the obstacles and the border. The start and goal are linked to the graph using some local operator.

[Applet pour Voronoi](#)

---

---

---

---

---

---

---

---

---

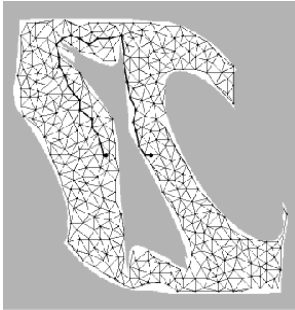
---

---

---



# Cartes réutilisables probabilistes



Le graphe est créé en générant de façon random un très grand nombre de configurations en ôtant celles qui ne font pas partie de l'ELC.

On joint ensuite les nœuds joignables facilement (par ex: en ligne droite).

---

---

---

---

---

---

---

---

# Décomposition en cellules

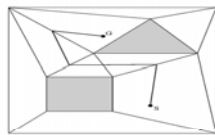


Figure 2.10: Exact cell decomposition example 1. Cells are generated as convex polyhedra with vertices at obstacle vertices.

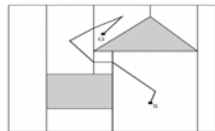


Figure 2.11: Exact cell decomposition example 2. Trapezoidal cells.

---

---

---

---

---

---

---

---

# Décomposition en cellules (suite)

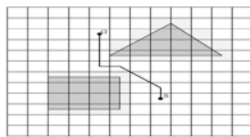


Figure 2.12: Approximate cell decomposition example 1. A regular grid of cells. Fully occupied cells are darkly shaded, partially occupied cells are lightly shaded and free cells are unshaded.

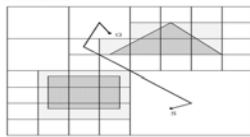


Figure 2.13: Approximate cell decomposition example 2. A quadtree decomposition of the scene. Fully occupied cells are darkly shaded, partially occupied cells are lightly shaded and free cells are unshaded.

---

---

---

---

---

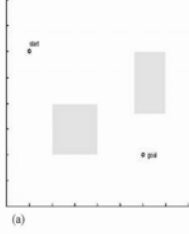
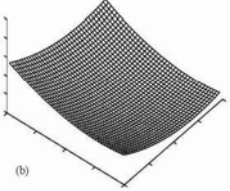
---

---

---

Design 3

## Champs de potentiel

(a) Deux obstacles, un start et un goal; (b) le potentiel attractif avec un minimum au point goal.

28

---

---

---

---

---

---

---

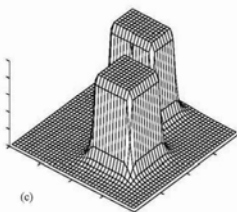
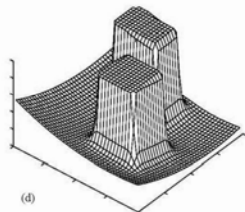
---

---

---

Design 3

## Champs de potentiel (2)

(c) Les potentiels répulsifs associés avec les obstacles; (d) le résultats des potentiels attractif et répulsif.

29

---

---

---

---

---

---

---

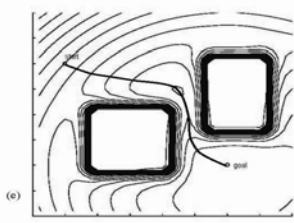
---

---

---

Design 3

## Champs de potentiel (3)



(e) Contournement selon le potentiel résultant et un path solution.

30

---

---

---

---

---

---

---

---

---

---

## Champ de potentiel (suite)

- ▶ Dans la configuration  $C$ , on a à faire à une particule soumise à un champ potentiel artificiel  $U(q)$ , avec  $q$  l'état du robot  $q = (x, y)$ 
  - ▶ Typiquement, pour un robot mobile

$$F(q) = -\nabla U(q)$$

A chaque itération, la force artificielle induite par le champ potentiel, indiquera alors la direction la plus prometteuse.

$$F(q) = -\nabla U(q) = (\partial U / \partial x, \partial U / \partial y)^T$$

31

---

---

---

---

---

---

---

---

---

---

## Champ de potentiel (suite)

Le champ potentiel est défini comme la somme d'un champ potentiel attracteur  $U_{att}(q)$ , poussant le robot vers la configuration finale, et un champ répulsif  $U_{rep}(q)$ , tenant le robot éloigné des obstacles.

$$U(q) = U_{att}(q) + U_{rep}(q)$$

Il convient de remarquer qu'avec le champ de potentiel on a la représentation et l'obtention du chemin cherché dans ladite représentation

32

---

---

---

---

---

---

---

---

---

---

## La recherche de chemin ou de plan

- ▶ Recherche dans un arbre :
  - ▶ On peut utiliser : Largeur/profondeur; Dijkstra; A\*, Meilleur d'abord; ID, IDA\*, LRTA\*, etc.
- ▶ Planification optimale dans un graphe quelconque.
  - ▶ Itération des valeurs (VI) (Déterministe/stochastique);
  - ▶ Itération de politique (PI) (Déterministe/stochastique).

33

---

---

---

---

---

---

---

---

---

---

## Itération des valeurs: cas déterministe

► Voir 2.3 Discrete Optimal Planning



### PLANNING ALGORITHMS

By **Steven M. LaValle**  
Copyright 2006  
Cambridge University Press  
ISBN-10: 0521862051 | ISBN-13:  
9780521862059  
Cover art by **James Kuffner**  
Art on the first page of each part by **Odrj**  
  
[The whole book in HTML](#)  
[BibTeX file for all 1005 references](#) (edited  
by **Jason O'Kane**)

---

---

---

---

---

---

---

---