

Experimental Reeds and Shepp parking

BJM

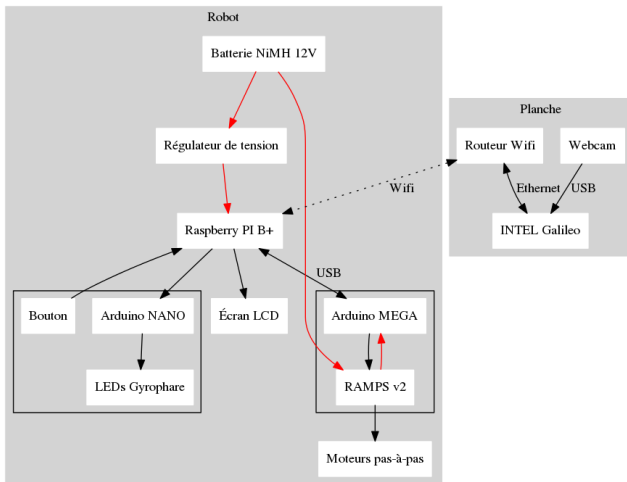
Outline

- 1 Robot
- 2 Computing the path...
- 3 Collision
- 4 Demo

Outline

- 1 Robot
- 2 Computing the path...
- 3 Collision
- 4 Demo

Engineering



Outline

- 1 Robot
- 2 Computing the path...**
- 3 Collision
- 4 Demo

General view

- **Visibility Probabilistic Roadmap**: search the free space
- **Dijkstra**: get the shortest 'found' path
- **Optimize**: get a shorter path

Reeds and Shepp

Configuration: x, y, θ

Tells us how to go from 'A' to 'B':

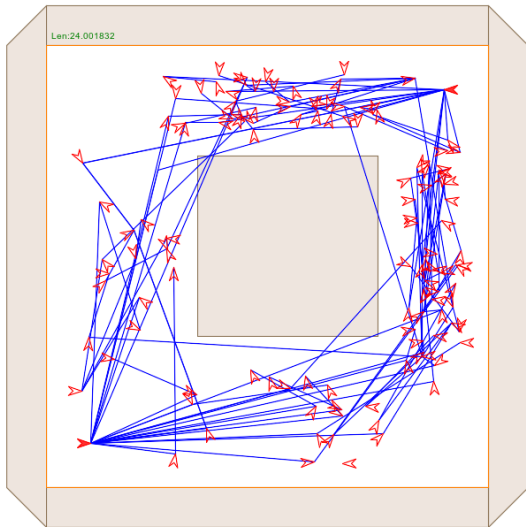
- optimally
- without obstacles
- with cups

Visibility PRM

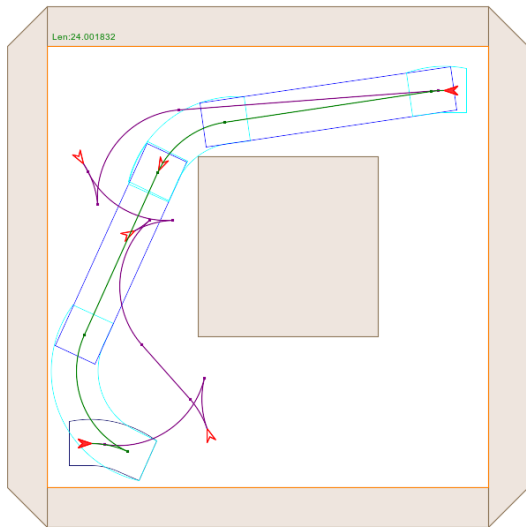
while(!stopcond)

- Get a point in the free space
- Find which **other points** can be attained from this point
 - Nowhere -> add it as a **'guard'** (create a new family)
 - To two different families -> add it as a **'connection'**
 - To a single family -> **do not** add it

Visibility PRM



Way and opt-way



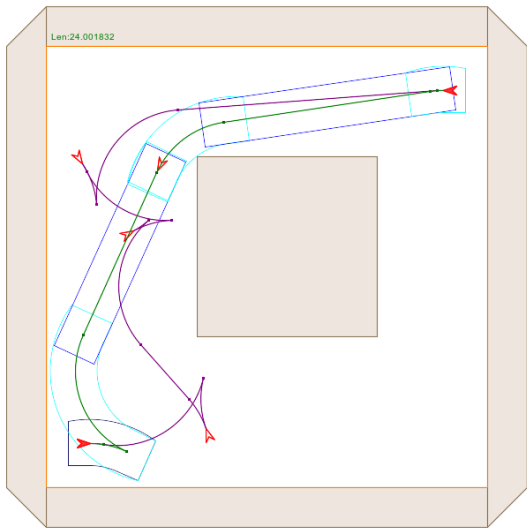
Optimisation

We have our way A_0, \dots, A_n .

- Can we go directly from A_i to A_{i+n} , $n \geq 2$?
- If yes, we get a shorter path!

- ϵ -move a point A_i ($i \neq 0$ and $i \neq n$)
- Is the path shorter?
- If yes, save it as the new path

<http://.fr/rob.png>



Outline

- 1 Robot
- 2 Computing the path...
- 3 Collision**
- 4 Demo

Basic geometry

- Scalar, cross product
- Distances, rotations
- Collinear lines

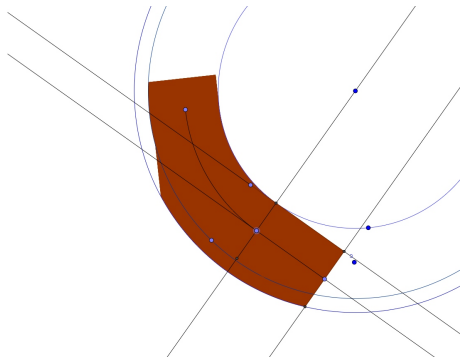
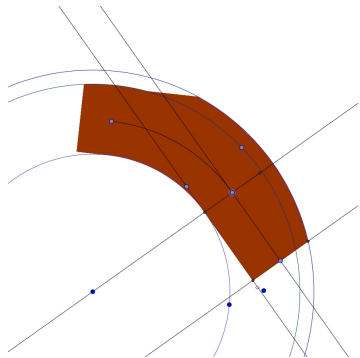
More advanced geometry

- Distance between line segment and point
- Intersection of segments
- Intersection of circle portion and segment

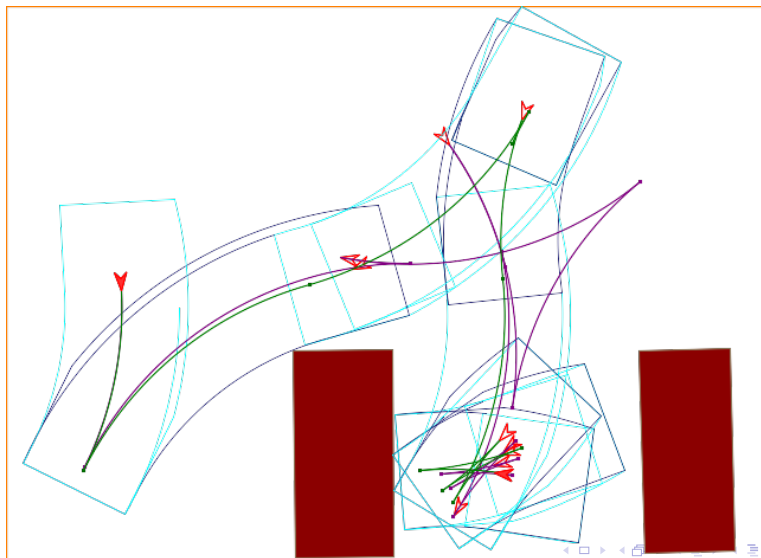
The general approach

- Test whether a configuration is admissible
inObst : testing if a corner of the robot is in an obstacle, and if a segment of the robot collides with a border of an obstacle.
- Test whether a path is admissible
Decomposing the path in segments and circle portions, and testing the collision with border of an obstacle.

The general approach



The general approach



Outline

- 1 Robot
- 2 Computing the path...
- 3 Collision
- 4 Demo**

Demo

<http://i.fr/rob/>