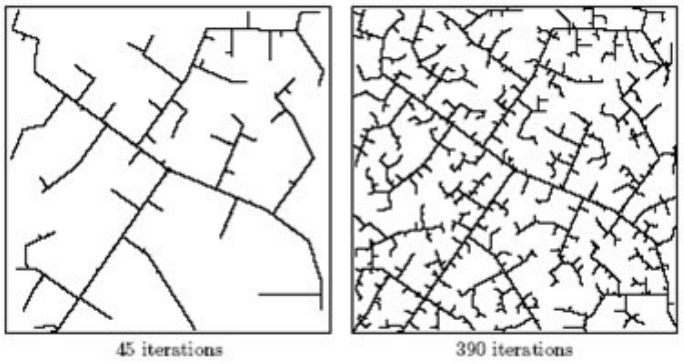
TP Robotique 4

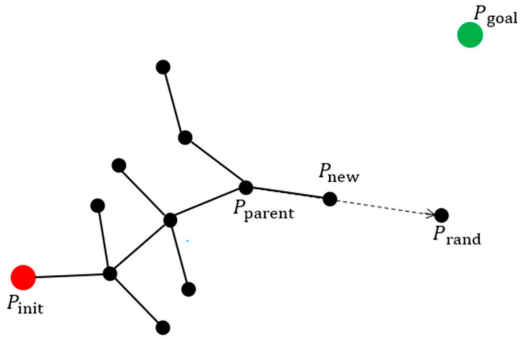
Rapidly Random Trees

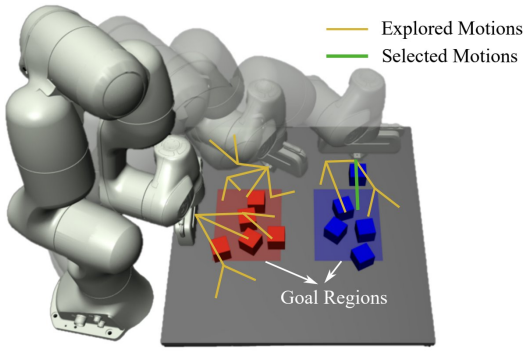
L'objectif du TP est de contrôler le robot Niryo avec l'algorithme RRT (Kuffner, Lavalle). L'algorithme RRT crée un arbre aléatoire pour la planification et le contrôle.

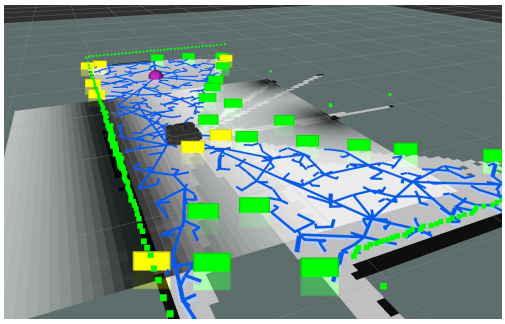
RRT utilise une discrétisation de l'espace cartésien ou de l'espace moteur pour créer un chemin.

https://en.wikipedia.org/wiki/Rapidly_exploring_random_tree









Les RRT peuvent être considérées comme une technique permettant de générer des trajectoires en boucle ouverte pour des systèmes non-linéaires avec des contraintes d'état. Un RRT peut également être considéré comme une méthode de Monte-Carlo pour biaiser la recherche dans les plus grandes régions de Voronoi d'un graphe dans un espace de configuration. Certaines variantes peuvent même être considérées comme des fractales stochastiques.

Pseudocode

```
Algorithm BuildRRT
    Input: Initial configuration q_init, number of vertices in RRT K, incremental distance Δq
    Output: RRT graph G

    G.init(q_init)
    for k = 1 to K do
        q_rand ← RAND_CONF()
        q_near ← NEAREST_VERTEX(q_rand, G)
        q_new ← NEW_CONF(q_near, q_rand, Δq)
        G.add_vertex(q_new)
        G.add_edge(q_near, q_new)
    return G
```

Le but du TP est de concevoir un contrôleur RRT

1) implémenter un système de graphe ou d'arbre d'états dans l'espace cartésien et faites un arbre RRT pour contrôler le robot et générer plusieurs trajectoires.
2) Affichez les trajectoires et trouvez celles minimales entre deux points. Quelle est la résolution spatiale en fonction du nombre d'itérations de l'algorithme
3) Mettez un obstacle et générez les chemins.

   Pour aller plus loin :

4) Testez une version dynamique : créez l'arbre en même temps que le robot se déplace en construisant de nouveaux noeuds et en supprimant les noeuds racines.

+ Pour le robot Niryo Ned

- Tutoriel bibliothèque Nyrio : Python API

  Ce package est capable de contrôler Ned physiquement mais également en simulation.

  https://archive-docs.niryo.com/dev/pyniryo/v1.1.2/fr/index.html

- Tutoriel bibliothèque Nyrio2 : Python+ROS

  https://archive-docs.niryo.com/dev/pyniryo2/v1.0.0/fr/index.html

- Tutoriel bibliothèque ROS

  https://archive-docs.niryo.com/dev/ros/v4.1.1/fr/index.html