

Projet 1 – Planification avec des robots

IFT608 / IFT702
Planification en intelligence artificielle
Hiver 2017

Professeur: Froduald Kabanza

Date de remise: 16 mars 2017 à 23:59 PM

L'objectif de ce projet est de se familiariser avec les algorithmes de planification de trajectoire et de planification de tâches en les appliquant à des robots simulés.

Le projet peut se faire par un groupe de 3 étudiants (pour un apprentissage optimal, il est recommandé de 1 à 2 étudiants). Il sera évalué sur 30 points comptant pour 30% de la note finale dans le cours IFT608. Il sera ramené à 25% pour IFT702.

Comme préalable, il faut installer les logiciels ROS, MoveIt et ROSPlan en suivant les instructions dans le document « [Installation de ROS, MoveIt et ROSPlan](#) ».

1 Tâche 1 (10 points) : Manipulations sans obstacles

La première partie du projet consiste à déplacer un robot et lui faire faire des manipulations simples dans un environnement sans obstacles.

Pour ce faire, il faut générer un package ROS de façon à pouvoir y exécuter des requêtes de planification pour le robot PR2 (Figure 1), un robot mobile fabriqué par Willow Garage¹.



Figure 1: Un robot PR2

¹ <http://wiki.ros.org/Robots/PR2>

1.1 Format URDF

URDF est un format XML permettant de représenter des modèles de robots. Les chaînes cinématiques y sont modélisées à l'aide entre autres de liaisons (*joints*) et de pièces (*links*). Un exemple est représenté sur la Figure 2.

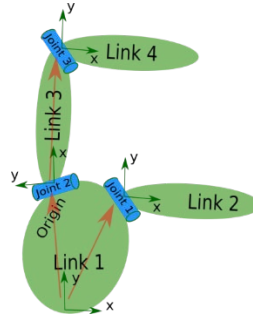


Figure 2: Exemple de chaîne cinématique²

1.2 Tâches à réaliser

1. Générer un package ROS pour le PR2 à partir du MoveIt! Setup Assistant : http://docs.ros.org/kinetic/api/moveit_tutorials/html/doc/setup_assistant/setup_assistant_tutorial.html

Sauvegarder le package créé par l'Assistant sous le nom *pr2_moveit_generated* dans le dossier *src/* de l'espace de travail créé avec catkin.

Une fois le package enregistré, la commande `roscd pr2_moveit_generated` devrait déplacer le dossier courant vers *pr2_moveit_generated*. Si ce n'est pas le cas, vérifier les étapes précédentes car une erreur de manipulation a eu lieu.

2. Pour modifier le package *pr2_moveit_generated* nouvellement créé il suffit de lancer de nouveau le MoveIt! Setup Assistant mais de cliquer cette fois sur *Edit Existing MoveIt Configuration Package* en spécifiant le chemin d'accès vers le package. Pour sauvegarder les modifications effectuées, il faut retourner dans l'onglet *Configuration Files* et cliquer de nouveau sur *Generate Package*.

Visualiser le robot dans RViz et lancer des requêtes de planification avec MoveIt. Lancer RViz à l'aide de la commande `roslaunch pr2_moveit_generated demo.launch`

Dans la fenêtre *Displays*, le groupe *Motion Planning* permet de lancer des requêtes de planification reliant un état initial à un état final désiré. **Ces états sont définis par rapport au Planning Group courant** (onglet *Planning Request*). Par exemple, si le *left_arm* est choisi, le bras gauche du robot pourra être déplacé.

² <http://wiki.ros.org/urdf/Tutorials/Create%20your%20own%20urdf%20file>

Les états initial et final (du *Planning Group* courant) peuvent être spécifiés de deux façons différentes :

- a. En déplaçant les marqueurs interactifs situés dans la fenêtre de visualisation du robot (disponibles seulement pour *left_arm*, *right_arm* et *base*).
- b. En changeant les *Start State* et *Goal State* dans la fenêtre *Motion Planning* (onglet *Planning*). En particulier, l'option *random* permet de spécifier un état aléatoire.

L'algorithme de planification utilisé (tiré de la librairie OMPL) peut être spécifié dans l'onglet *Context* de la fenêtre *Motion Planning*. Si aucun algorithme n'est spécifié, MoveIt! utilisera un algorithme par défaut.

Le bouton *Plan* de l'onglet *Planning* appelle l'algorithme afin de trouver une solution. **La solution inclut seulement le *Planning Group* courant.**

- Si aucune solution n'est trouvée, la mention *Planning Failed* s'affichera. **De l'information supplémentaire sera affichée dans le terminal.**
 - Aucune solution n'est trouvée si l'état initial ou l'état final du *Planning Group* est en collision avec le robot lui-même ou avec l'environnement (des segments du robot sont alors affichés en rouge).
 - Aucune solution n'est trouvée si l'état initial ou l'état final se trouve en dehors des limites de l'environnement (*out of bounds*). Dans ce cas, il pourrait être avisé de rapprocher la base du robot du milieu de l'environnement.
 - Si une solution est trouvée, la case à cocher *Loop Animation* dans l'onglet *Planned Path* permet de répéter l'animation.
3. Il existe une version du package de PR2 qui intègre une définition plus complète du robot. Celle-ci permet notamment de déplacer la base du robot. Pour l'utiliser, suivre les instructions suivantes.

Télécharger le fichier zip suivant : https://github.com/ros-planning/moveit_pr2/archive/indigo-devel.zip

Extraire le répertoire *pr2_moveit_config* et le placer dans le répertoire *src* de l'espace de travail.

Visualiser le robot dans RViz en utilisant le package *pr2_moveit_config*. Pour ce faire, utiliser la commande suivante : *roslaunch pr2_moveit_config demo.launch*.

Dans la fenêtre *Motion Planning*, sous l'onglet *Planning*, étendre l'espace de travail à (10, 10, 10), ce qui correspond à la taille de la grille.

Sous cette nouvelle configuration, il est possible de choisir le *Planning Group* « *whole_body* » pour planifier la trajectoire du robot au complet.

1.3 Livrable

Pour cette tâche, il faut livrer une archive *tache1.zip* ayant le contenu suivant:

1. Package ROS *pr2_moveit_generated* où les opérations suivantes sont possibles dans RViz:
 - Manipuler les marqueurs interactifs pour les bras (*left_arm* et *right_arm*) et pour la *base*.
 - Sélectionner un *Planning Group* nommé *whole_body* contenant au moins les sous-groupes suivants : *left_arm*, *right_arm*, *left_gripper*, *right_gripper*, *base*.
 - Résoudre des requêtes de planification reliant un état initial à un état final. (Indice : tenter d'ouvrir le package *pr2_moveit_config* avec le MoveIt! Setup Assistant...)
2. Vidéo *tache1.avi* illustrant la résolution d'une requête de planification de trajectoire impliquant des changements dans les groupes suivants : *left_arm*, *right_arm*, *left_gripper*, *right_gripper*.

2 Tâche 2 (10 points): Évitement d'obstacles

La deuxième partie du devoir consiste à exécuter des requêtes de planification pour le PR2 dans un environnement parsemé d'obstacles.

2.1 Tâches à réaliser

1. Visualiser le robot dans RViz en utilisant le package *pr2_moveit_generated* généré dans la première partie du devoir.
2. Télécharger l'archive [Projet1-exemple.zip](#). Une fois décompressé, vous devriez avoir un répertoire *Projet-exemple1* contenant des fichiers dont vous aurez besoin dans cette tâche (voir ci-après).
3. Importer l'environnement *Projet1-exemple/example-scene/moveit.scene* dans RViz à partir de l'onglet *Scene Objects* de la fenêtre *Motion Planning*. La procédure est la suivante:
 - a. Dans l'onglet *Scene Objects* de la fenêtre *Motion Planning*, cliquer sur le bouton *Import From Text*.
 - b. Sélectionner le fichier text de la scene. Les objets importés devraient apparaître dans la section *Current Scene Objects*.
 - c. En sélectionnant un objet, il est possible de d'ajuster sa position, son orientation et sa taille dans la scene. Il est également possible de dupliquer les objets en utilisant les raccourcis *ctrl-c* et *ctrl-v*.
 - d. Le bouton *Export As Text* permet d'enregistrer la scene dans un fichier texte.

4. Modifier l'environnement de façon à reproduire celui apparaissant dans le vidéo *Projet1-exemple/pr2-scene-model.avi*, ou créer un environnement de complexité similaire.
5. Dans l'onglet *Context* de la fenêtre *Motion Planning*, cliquer sur *Publish Current Scene* après chaque modification de l'environnement. Ceci permet de notifier l'algorithme de planification afin qu'il tienne compte de ces changements lors des calculs de trajectoires.
6. Tester des requêtes de planification dans votre environnement et s'assurer que les chemins calculés évitent les obstacles.
7. Exporter l'environnement sous format texte sous le nom *pr2.scene*.

2.2 Livrable

Pour cette tâche, il faut livrer une archive *tache2.zip* ayant le contenu suivant :

1. Environnement *pr2.scene* pouvant être importé à partir de RViz.
 - L'environnement doit être de complexité similaire à celui apparaissant dans le vidéo *Devoir2-Fichiers/pr2-scene-model.avi*.
2. Vidéo *tache2.avi* illustrant la résolution d'une requête de planification de trajectoire impliquant des changements dans les groupes suivants : *left_arm*, *right_arm*, *left_gripper*, *right_gripper*, *base* au sein de l'environnement *pr2.scene*.

3 Tâche 3 (10 points) : Robonaut

La troisième partie du devoir consiste à exécuter des requêtes de planification pour le robot Robonaut (Figure 3) dans un environnement parsemé d'obstacles. Robonaut est un robot mobile fabriqué par la NASA et destiné à assister les astronautes de la station spatiale³.



Figure 3: Robonaut

3.1 Tâches à réaliser

1. Le tutoriel suivant peut servir de guide pour cette tâche :
http://picknik.io/moveit_wiki/index.php?title=Robonaut/Setup_Assistant/Quick_Start

³ http://www.nasa.gov/mission_pages/station/multimedia/robonaut_photos.html

Bien que ce tutoriel corresponde la distribution *groovy* de ROS plutôt que la distribution *kinetic* utilisée pour ce devoir, la démarche devrait rester sensiblement la même. Il se peut qu'il y ait quelques différences à prendre en considérations. Un package contenant la description plus récent du robot est disponible à l'URL

https://gitlab.com/nasa-jsc-robotics/r2_description.git

Le fichier *urdf* correspondant se trouve dans *r2_description/urdf/r2c5.urdf*.

À l'aide du *MoveIt! Setup Assistant*, générer un package nommé *r2_moveit_generated* dans le répertoire *src/* de l'espace de travail *catkin*.

2. Visualiser le robot dans RViz en utilisant le package *r2_moveit_generated* ainsi généré. Il est possible qu'aucun marqueur interactif ne soit affiché.
3. Modifier l'environnement de façon à reproduire celui apparaissant dans le vidéo *Projet1-exemple/r2-scene-model.avi*, ou créer un environnement similaire.

3.2 Livrable

Pour cette tâche, il faut livrer une archive *tache3.zip* ayant le contenu suivant:

1. Package ROS *r2_moveit_generated* où les opérations suivantes sont possibles:
 - Sélectionner un Planning Group nommé *whole_body* contenant au moins les sous-groupes suivants : *left_arm*, *right_arm*, *left_gripper*, *right_gripper*, *base*. Il n'est pas nécessaire de faire apparaître des marqueurs interactifs pour ces groupes.
 - Résoudre des requêtes de planification reliant un état initial à un état final.
2. Environnement *r2.scene* pouvant être importé à partir de RViz. L'environnement doit être similaire à celui apparaissant dans le vidéo *Devoir2-Fichiers/r2-scene-model.avi*
3. Vidéo *tache3.avi* illustrant la résolution d'une requête de planification de trajectoire impliquant des changements dans les groupes suivants : *left_arm*, *right_arm*, *left_gripper*, *right_gripper* au sein de l'environnement *r2.scene*

4 Soumission

Vous devez soumettre les trois livrables (*tache1.zip* à *tache3.zip*) ainsi qu'un fichier **etudiants.txt** contenant les noms des membres de l'équipe et leurs matricule par turnin sur le site opus.dinf.usherbrooke.ca au plus tard le 16 mars 2017 à 23:59 PM.

Une remise tardive sera pénalisée de 10% dans la première tranche de 3h, 20% dans la première trache de 6h, et ainsi de suite.

Les remises doivent respecter scrupuleusement les consignes données dans l'énoncée. Tout non respect des consigne qui complique la correction sera pénalisée en fonction de l'écart des consignes. Dans le pire cas, si une tâche ne s'exécute pas et qu'il devient difficile de déterminer en un temps raisonnable si c'est à cause du non respect des consignes ou à cause des erreurs, elle risque d'être évaluée à zéro.