2020

Le Robot Marcheur

PROJECT ROBOTIQUE INDUSTRIELLE

SUNGKUR SEEKCHA

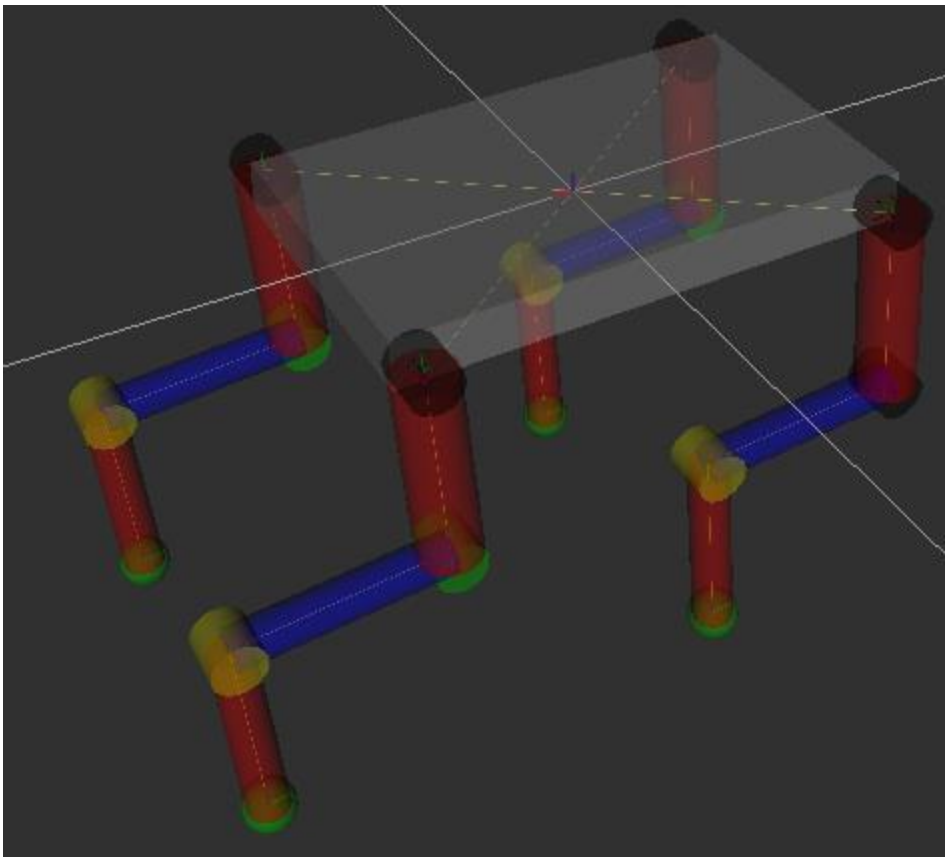
Table of Contents

Partie 2 Ajout des autres jambes

Question 1

1. A partir de l'urdf créé à la partie 1, créer un nouvelle urdf où vous ajouterez le reste des jambes de votre robot.

```
deeya@deeya-VirtualBox:~/catkin_ws$ cd src
deeya@deeya-VirtualBox:~/catkin_ws/src$ catkin_create_pkg udm_leg rospy roscpp u
rdf std_msgs geometry sensor_msgs
Created file udm_leg/CMakeLists.txt
Created file udm_leg/package.xml
Created folder udm_leg/include/udm_leg
Created folder udm_leg/src
Successfully created files in /home/deeya/catkin_ws/src/udm_leg. Please adjust t
he values in package.xml.
deeya@deeya-VirtualBox:~/catkin_ws/src$ ls
CMakeLists.txt    udm_leg            udm_project_control  urdf_quad
udm_hand_control  udm_oneleg_moveit  udm_urdf
deeya@deeya-VirtualBox:~/catkin_ws/src$ cd udm_leg
deeya@deeya-VirtualBox:~/catkin_ws/src/udm_leg$ ls
CMakeLists.txt  include  package.xml  src
deeya@deeya-VirtualBox:~/catkin_ws/src/udm_leg$ mkdir launch
deeya@deeya-VirtualBox:~/catkin_ws/src/udm_leg$ ls
CMakeLists.txt  include  launch  package.xml  src
deeya@deeya-VirtualBox:~/catkin_ws/src/udm_leg$ cd launch
deeya@deeya-VirtualBox:~/catkin_ws/src/udm_leg/launch$ touch check_urdf.launch
deeya@deeya-VirtualBox:~/catkin_ws/src/udm_leg/launch$ ls check_urdf.launch
check_urdf.launch
deeya@deeya-VirtualBox:~/catkin_ws/src/udm_leg/launch$ cd ..
deeya@deeya-VirtualBox:~/catkin_ws/src/udm_leg$ ls
CMakeLists.txt  include  launch  package.xml  src
deeya@deeya-VirtualBox:~/catkin_ws/src/udm_leg$ mkdir urdf
deeya@deeya-VirtualBox:~/catkin_ws/src/udm_leg$ ls
CMakeLists.txt  include  launch  package.xml  src  urdf
deeya@deeya-VirtualBox:~/catkin_ws/src/udm_leg$ cd urdf
deeya@deeya-VirtualBox:~/catkin_ws/src/udm_leg/urdf$ touch main.urdf
deeya@deeya-VirtualBox:~/catkin_ws/src/udm_leg/urdf$ ls
main.urdf
```



1. Load urdf file in MoveIt.
2. Generate the collision matrix.
3. Add the virtual joint.

	Virtual Joint Name	Child Link	Parent Frame	Type
1	world_to_baseFoot	base_foot	world	fixed

4. Add new Planning groups
    a. Add group name
    b. Kniematic Solver : kdl_kinematics_plugin/KDLKinematicsPlugin
    c. Group Default Planner: RRT
    d. Add joint.
5. Add robot pose.

	Pose Name	Group Name
1	backward	front_left_leg
2	right	front_left_leg
3	forward	front_right_leg

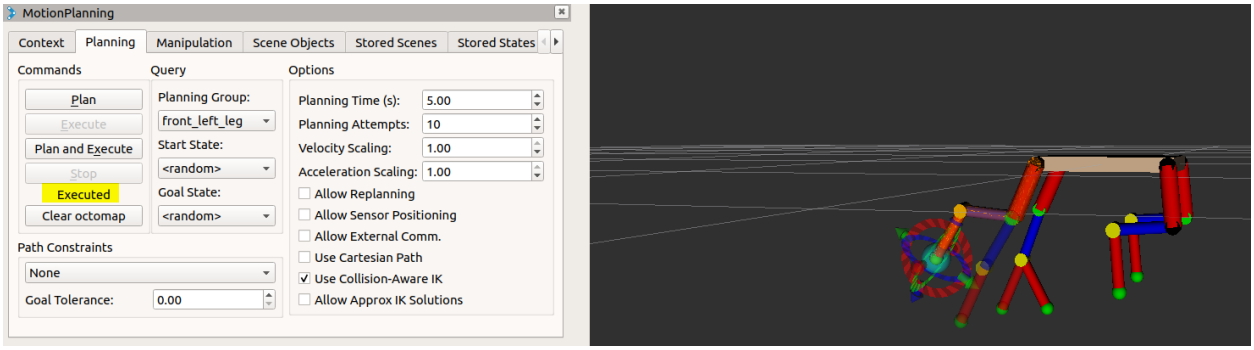
6. Add end effectors.

	End Effector Name	Group Name	Parent Link	Parent Grou
1	eff_front_left_leg	end_left_front_leg	base_foot	front_left_leg
2	eff_front_right_leg	end_front_right_leg	base_foot	front_right_le
3	eff_rear_left_leg	end_left_rear_leg	base_foot	rear_left_leg
4	eff_rear_right_leg	end_right_rear_leg	base_foot	rear-right_leg

7. Setup ROS Controllers > Auto Add_FollowJointsTrajectory
8. Simulate with gazebo > generate URDF
9. Add Author info
10. Generate Package
11. Open demo.launch

```
deeya@deeya-VirtualBox:~/catkin_ws$ roslaunch moveit2 demo.launch
... logging to /home/deeya/.ros/log/380a9832-0655-11eb-915d-080027eb1150/roslaun
ch-deeya-VirtualBox-31371.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://deeya-VirtualBox:38581/
```



Question 2

2. Dans udm_project_control créer un service permettant de déplacer le robot en ligne droite, à gauche, à droite ou en arrière.

Note: Instead of udm_project_control used urdf_project_control.

```
deeya@deeya-VirtualBox:~/catkin_ws$ cd src
deeya@deeya-VirtualBox:~/catkin_ws/src$ mkdir urdf_project_control
```

1. Launch roscore

```
deeya@deeya-VirtualBox:~/catkin_ws$ roscore
... logging to /home/deeya/.ros/log/a4ae45b6-0672-11eb-b958-080027eb1150/roslaunch-deeya-VirtualB
ox-2868.log
```

2. Launch move_group.launch in moveit2.

```
deeya@deeya-VirtualBox:~/catkin_ws$ roslaunch moveit2 move_group.launch
... logging to /home/deeya/.ros/log/a4ae45b6-0672-11eb-b958-080027eb1150/roslaunch-deeya-VirtualB
ox-5854.log
```

3. Launch demo.launch in moveit2.

```
deeya@deeya-VirtualBox:~/catkin_ws$ roslaunch moveit2 demo.launch
... logging to /home/deeya/.ros/log/a4ae45b6-0672-11eb-b958-080027eb1150/roslaunch-deeya-VirtualB
ox-5667.log
```

4. Launch service.launch in urdf_project_control.

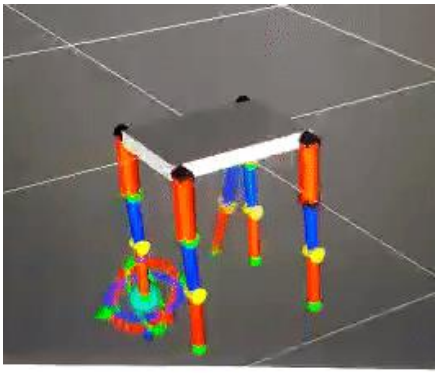
```
deeya@deeya-VirtualBox:~/catkin_ws$ roslaunch urdf_project_control service.launch
... logging to /home/deeya/.ros/log/a4ae45b6-0672-11eb-b958-080027eb1150/roslaunch-deeya-VirtualB
ox-10877.log
```

5. Call function ‘front’.

```
deeya@deeya-VirtualBox:~/catkin_ws$ rosservice call /kin_service "movement:
 data: 'front'"
res:
  data: True
message:
  data: "Success"
```

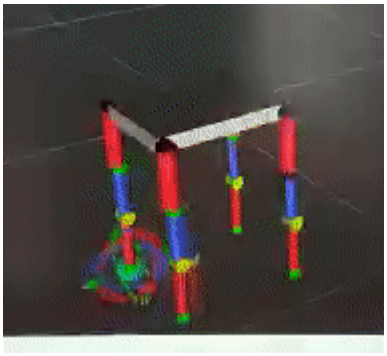
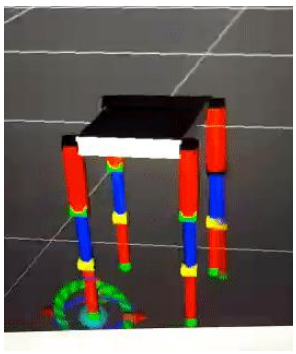
View front.git

Click on the image to see the movement



6. Call function ‘right’ or ‘left’ or ‘back’.

   Click on the image to see the movement

Annex

1. Error: Invalid Trajectory

```
ed from 2 to 2 states
[ INFO] [1601833882.324989485]: Execution request received
[ERROR] [1601833882.344039996]:
Invalid Trajectory: start point deviates from current robot state more than 0.01
joint 'joint_right_rear_1': expected: 0.7128, current: 1.21
[ INFO] [1601833882.344105437]: Execution completed: ABORTED
[ INFO] [1601833882.375912952]: ABORTED: Solution found but controller failed during execution
```

Solution : Change 1.21 to 0.7128

```
<group_state name="rr_left" group="rear_right_leg">
    <joint name="joint_right_rear_1" value="0.7901" />
    <joint name="joint_right_rear_2" value="0.1963" />
    <joint name="joint_right_rear_3" value="-1.5" />
    <joint name="joint_right_rear_4" value="-1.6" />
</group_state>
```