Intelligent Robotics - Homework 2
2021-2022

INTRODUCTION

To complete the homework, we have divided the different tasks in the 3 nodes as suggested, managed as Client/Server. NodeA is the client that calls the other nodes to complete the different tasks. NodeB is a server that takes care of the detection part. NodeC moves the arm in order to pick and place the different objects. We decided to implement the communication as an ActionClient/ActionServer infrastructure because of the easy access to goals and results of each node, and of the practical way to synchronize and coordinate the nodes' execution this option provides.

We applied the following changes to the `ias_lab_room_full_tables.world`:

- Made the cylindrical place tables static.
- Reduced the weight of the target objects from 2 kg to 0.1 kg and the weight of the tag from 1 kg to 10 gr.

We implemented the extrapoints part of transporting all three objects to the place tables.

NODE A - MASTER

NodeA is a client that acts like a master for the whole process, and its execution starts the actual execution of the task. It connects to the action servers moveTiagoServer, nodeB, nodeC and to human_node. The execution starts by fetching from human_node the sequence of object IDs for the task, then starting from the first object and cycling for all of them, the routine is started. First, the position in front of the table for the pick is assigned and sent as a goal to moveTiagoServer. The latter takes the robot to the desired position passing by a defined waypoint to facilitate the navigation. Once there, nodeA calls nodeB to execute the object detection, and the resulting poses are sent as goal in nodeC's call, together with the object's ID and the actionType value equal to 0 (actionType = 0 if the call is for the pick phase, actionType = 1 for the place phase). When the pick execution by C is over and the call is returned, A sends the coordinates of the point in front to the right cylindrical table as a goal to moveTiagoServer, which moves Tiago to that precise point passing from another waypoint. The last step of the routine is to call C with actionType = 1 to execute the object's place on the table. It should be mentioned that both the robot poses in front of the pick table and the place table, as well as the two waypoints, are defined in the "`map`" reference frame.

NODE B - DETECTION

NodeB is a server that takes care of the detection section of the pipeline. After Tiago arrives in front of the table, we slightly lift up the torso and move the head down, in order to have the target object and the highest number of obstacles all in sight for the best detection possible. Then we start the detection using the Apriltag plugin. In some cases it happened that, even if the target was on sight, it was not detected by Apriltag. Because of that, assuming that there is always a target object (ID <= 3) in sight, we added a loop to force the detection until the target is found. After the detection, we lower the torso and move the head in the starting position. The node returns as result a vector containing the poses of all the detected objects in the "`base_footprint`" frame that is the robot frame. In the first position of the vector it is always reported the pose of the target object.

NODE C - MOVE ARM

NodeC is the server that manages the correct movement of the arm to pick and place the objects. It takes in input the poses of the objects detected in the previous step, the object's ID and a flag that indicates the action (pick or place) that the robot has to perform.

PICK

First of all we add the table and all the detected objects as collision objects in the MoveIt scene. For the triangle and the hexagons, we load the given meshes while the cube and the table are created as primitives. All the obstacles are created a little larger than the real ones and placed with respect to the detected poses. Then the pick action starts. The complete movement to pick an object is composed by the following sequence of arm pose: Initial pose - Prepick pose - Pick pose - Prepick pose - Intermediate pose - Safe pose. We have placed `sleep(1)` after reaching every arm pose to stop the robot for one second, ensuring that the previous movement's transitory effects are over before starting the current movement. Pick pose and Prepick pose are given in the cartesian space based on the detected pose of the target object. The orientation of the end effector is obtained by rotating the orientation of the target using an adequate rotation matrix to have the EE ready to pick the object. We chose to implement this solution, given that we have all the information from the detection task, because it allows us to obtain complicated pick orientation (such as the one for the triangle) in a fast and efficient way. The position of the end effector for the Prepick pose is directly above the object, and the Pick position is lower, so that the gripper is around the target. Once the arm is in the Pick pose, we attach the collision object of the target to the end effector in MoveIt, passing the two links of the gripper as `nonCollisionLinks`, that is the set of links the object is allowed to touch without considering that as a collision. Then the object is attachedin Gazebo through the plugin and the gripper is closed.
The Initial pose and the Safe pose are defined in the Joint space and are the same for all the objects. The intermediate pose for both the cube and the hexagon is the

same as the Initial, for the triangle we have defined a different pose to make the motion planning easier. Once the robot is in the safe pose, all collision objects are removed, and Tiago moves towards the place area.

PLACE

Once the robot is in front of the correct cylinder table, we add it as a collision object then the place procedure can start. It works similarly to the pick procedure, as we divide the movement in the sequence Safe pose - Preplace pose - Place pose - Preplace pose - Safe pose. The Safe pose is the same at the beginning and at the end, and it is the one the arm is moved to at the last step of the pick procedure, and the Preplace pose is defined in the joint space uniquely for all the objects. The Place pose is defined in the joint space as well, but specifically for each object and its joint values have been determined from RViz in such a way that the end effector hangs some centimeters above the table. In that pose the gripper is opened, the target is detached from the robot both in Gazebo and MoveIt scene and dropped on the table. Then the arm returns in the safe pose and, if there are other objects to pick, Tiago moves towards the pick area.

COMMANDS FOR THE EXECUTION

All the necessary files for the correct execution of the tasks are contained in the tiago_iaslab_simulation package.

To execute the procedure, the commands to use are the following:

1) Clone the package in the src folder of the workspace.
2) Build the workspace.
3) Launch: `roslaunch tiago_iaslab_simulation start_simulation.launch world_name:=ias_lab_room_full`
4) Launch: `roslaunch tiago_iaslab_simulation navigation.launch`
5) Launch: `roslaunch tiago_iaslab_simulation apriltag.launch`
6) Run: `rosrun tiago_iaslab_simulation human_node`
7) Run: `rosrun tiago_iaslab_simulation moveTiago_server`
8) Run: `rosrun tiago_iaslab_simulation nodeB_Detection`
9) Run: `rosrun tiago_iaslab_simulation nodeC_moveArm`
10) Run: `rosrun tiago_iaslab_simulation nodeA`

Alternatively, it is possible to run the whole procedure from a single terminal using the following command, but all the information outputted by the nodes will not be visible in the terminal.

```
roslaunch tiago_iaslab_simulation hw2.launch
```