In this lecture, we will finish the build of a simple pick and place pipeline with different MoveGroup APIs.

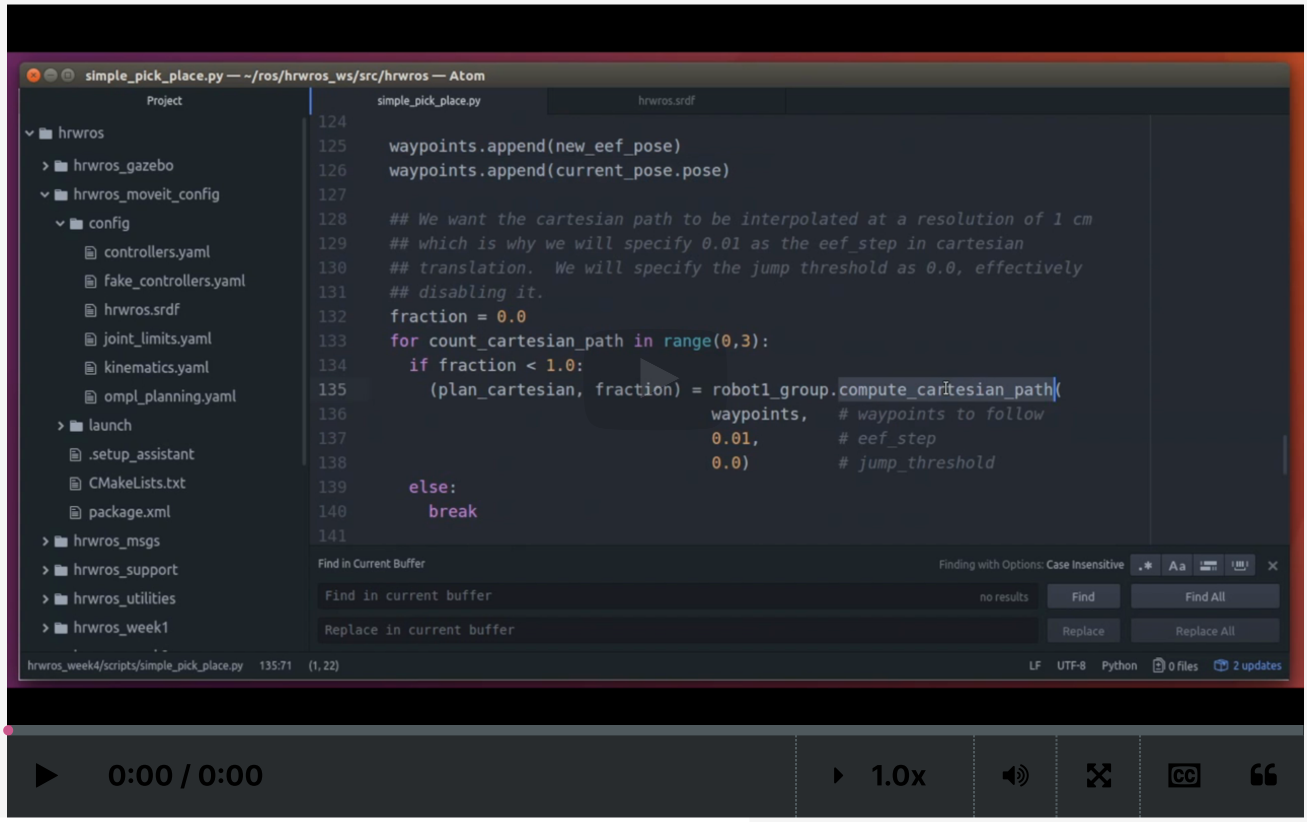
Important Note:

In the new version of ROS, the plan() function returns a tuple. Therefore, the old code:

`plan = robot1_group.plan(),` could no longer be used.

It has been replaced by the following code: `_, plan, _, _ = robot1_group.plan()`

Pick & Place: Part 3



```python

waypoints.append(new_eef_pose)
waypoints.append(current_pose.pose)

## We want the cartesian path to be interpolated at a resolution of 1 cm
## which is why we will specify 0.01 as the eef_step in cartesian
## translation.  We will specify the jump threshold as 0.0, effectively
## disabling it.
fraction = 0.0
for count_cartesian_path in range(0,3):
  if fraction < 1.0:
    (plan_cartesian, fraction) = robot1_group.compute_cartesian_path(
                                 waypoints,   # waypoints to follow
                                 0.01,        # eef_step
                                 0.0)         # jump_threshold

  else:
    break

```

Find in Current Buffer
Finding with Options: Case Insensitive
Find in current buffer
no results
Find
Find All
Replace in current buffer
Replace
Replace All
hrwros_week4/scripts/simple_pick_place.py 135:71 (1, 22)
LF UTF-8 Python 0 files 2 updates
0:00 / 0:00
1.0x

Video ▼ Subtitles ▼

Subtitles (captions) in other languages than provided can be viewed at YouTube. Select your language in the CC-button of YouTube

Computing the linear motion:

`compute_cartesian_path()` API

- Considers current Pose and the starting waypoint
- Uses the waypoints and two other arguments
- Checks for collision
- Outputs
    - Plan - the trajectory of joint angles and
    - A fraction of successful attempts

Update generated trajectory to the goal message and send it.

Lastly, use the `set_named_target("R1Place")` to move the robot to the object.

Run the script to see the robot in action. This concludes the Pick & Place code demonstration.

Question 1

1 point possible (ungraded)

The compute_cartesian_path API also checks for collisions of the linear paths it generates.

○ True

○ False

Submit