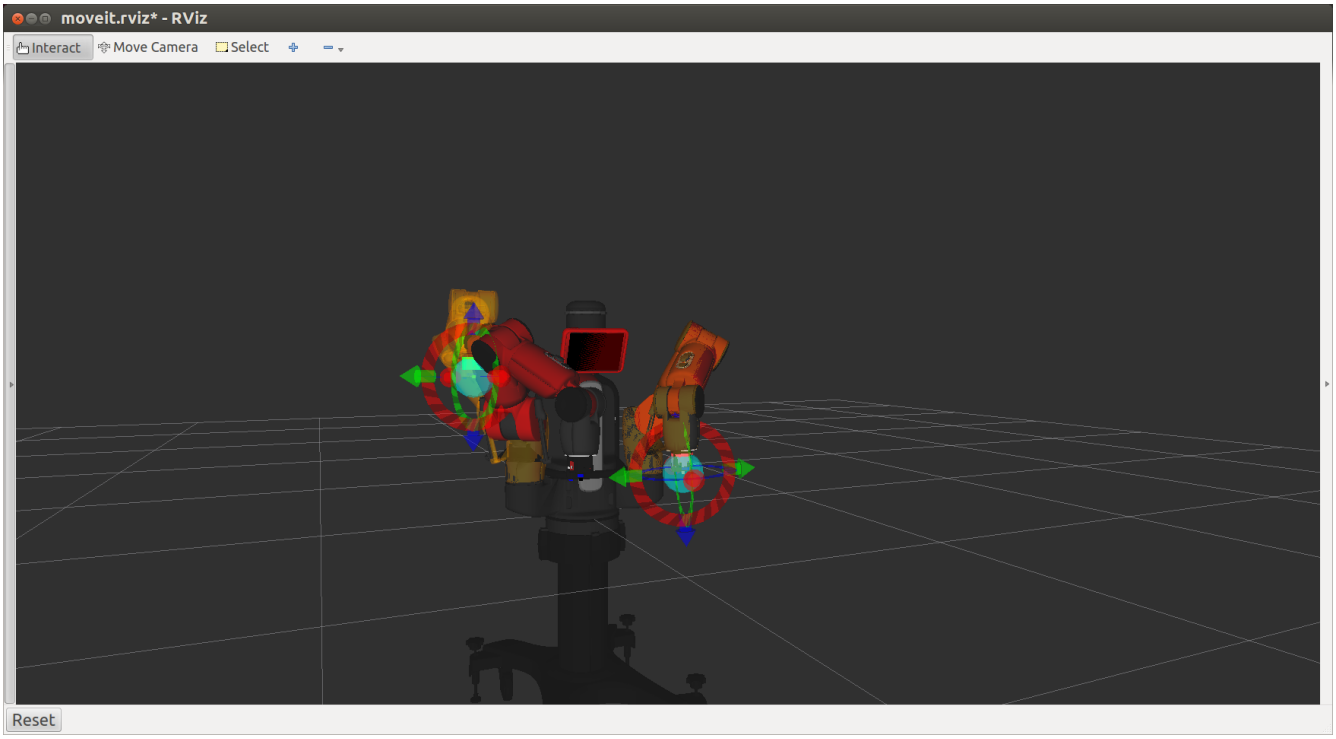
WORKING WITH BAXTER MOVEIT, PILLAR AND OTHER OBJECTS



Tom Harman, Carol Fairchild, Louise Li. 3/7/2015

Contents

FIRST, WATCH THIS VIDEO 3
See how to run MoveIt and Manipulate Baxter's arms. 3
TURN ON BAXTER AND WAIT FOR POWER UP- Assume Arms are Tucked 3
Untuck Arms 3
Run joint_trajectory_action_server.py 3
SECOND TERMINAL WINDOW: 3
Launch the baxter_moveit_config package and demo_baxter_launch 4
PLAN A MOVE 6
MOVE BAXTERS ARMS USING MOVEIT 7
ADD OBJECTS TO SCENE 9
Position of Objects 9
HAVE BAXTER AVOID AN OBJECT 11
Context tab for baxter to move around obstacles 11
Plan and Execute to move Left Arm Over Obstacle 13
Move Baxter's Arm Past Obstacle. 13
LAB WORK WITH BAXTER 14
CAUTION: Sometimes Baxter's real arms get stuck in odd positions. If so, move them apart and restart MoveIt. 14
SAVE OR LOAD STORED STATES 14
MOVEIT REFERENCES 16
MoveIt Packages 17
APPENDICES Do packages, topics , etc on Baxter 18

Figure 1 MoveIt Rviz Window with Display and Motion Planning Panes 4
Figure 2 MoveIt screen with Baxter in "Start State" 6
Figure 3 Baxter's Goal Arm Position in MoveIt 7
Figure 4 Baxter's Arms in Goal State 8
Figure 5 Tabletop and pillar added to scene 9
Figure 6 Context Tab 11
Figure 7 Baxter's simulated right arm moved out of the way 12
Figure 8 Baxter's Simulated arm moved to other side of obstacle 13
Figure 9 Stored States Tab 14

FIRST, WATCH THIS VIDEO

See how to run MoveIt and Manipulate Baxter's arms.
https://www.youtube.com/watch?feature=player_detailpage&v=1Zdkwym42P4

TURN ON BAXTER AND WAIT FOR POWER UP- Assume Arms are Tucked

I. First terminal window:

Use Alias in bachrc for CENG5931@D125
From ceng5931@d125
ceng5931@D125-43873:~$ **ros_ws**
ceng5931@D125-43873:/home/tlharmanphd/ros_ws$

tlharmanphd@D125-43873:~$ **cd ~/ros_ws**
tlharmanphd@D125-43873:~/ros_ws$ ***./baxter.sh***

Untuck Arms

This will Enable Baxter

[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ **. run_baxter untuck**
Today is Fri Feb 27 17:50:03 CST 2015
[INFO] [WallTime: 1425081004.255430] Untucking arms
[INFO] [WallTime: 1425081004.423053] Moving head to neutral position
[INFO] [WallTime: 1425081004.423393] Untucking: One or more arms Tucked; Disabling Collision Avoidance and untucking.

Run joint_trajectory_action_server.py
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ **rosrun baxter_interface joint_trajectory_action_server.py**
Initializing node...
Initializing joint trajectory action server...
Running. Ctrl-c to quit

SECOND TERMINAL WINDOW:

For CENG 5931 Log In
ceng5931@D125-43873:~$ **ros_ws**
ceng5931@D125-43873:/home/tlharmanphd/ros_ws$

OR
tlharmanphd@D125-43873:~$ **cd ~/ros_ws**
tlharmanphd@D125-43873:~/ros_ws$ **./baxter.sh**

[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$

Launch the baxter_moveit_config package and demo_baxter_launch

[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ **roslaunch baxter_moveit_config demo_baxter.launch**

... logging to /home/tlharmanphd/.ros/log/84c49790-bddb-11e4-937b-000af72ca0bb/roslaunch-D125-43873-21183.log

Checking log directory for disk usage. This may take awhile. (Lots of Information)

All is well! Everyone is happy! You can start planning now!

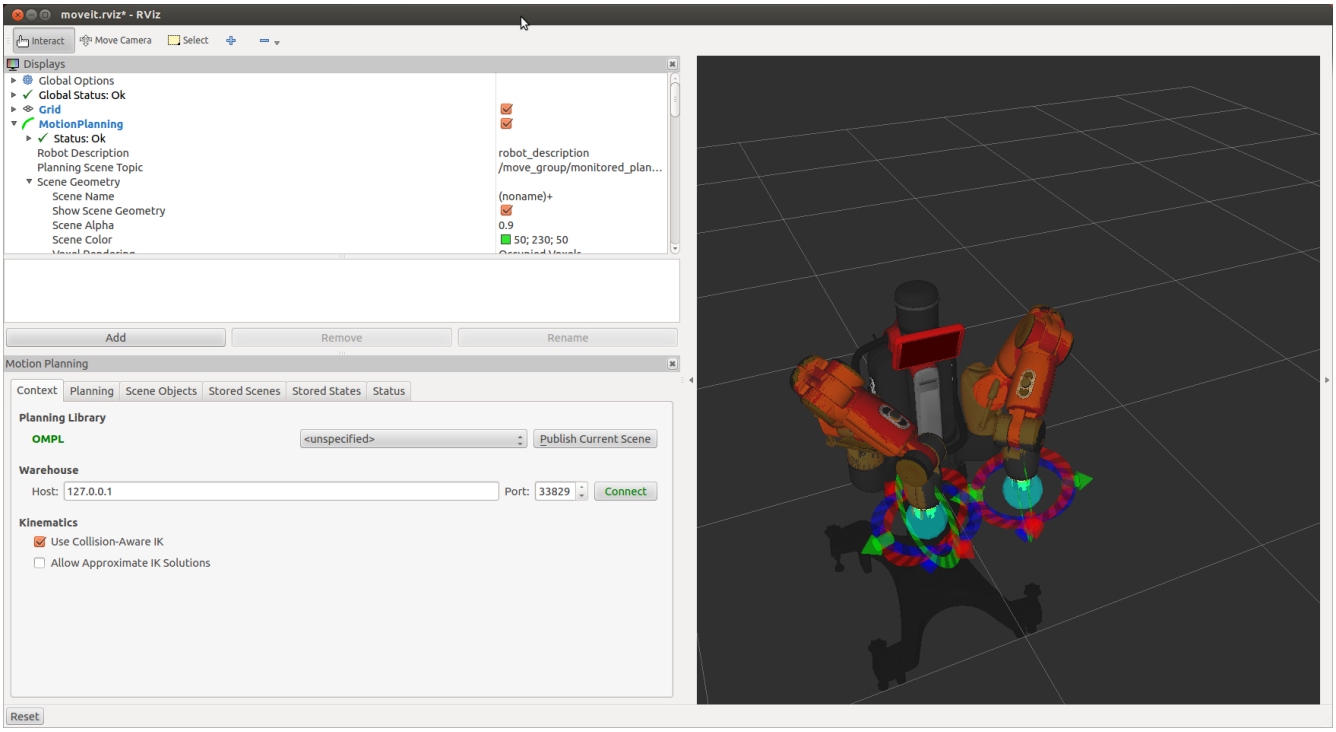


Figure 1 MoveIt Rviz Window with Display and Motion Planning Panes

In the Figure, the Displays and Motion Planning windows are shown on the left with the Context Tab information showing. On the right is simulated Baxter in the starting position of the real Baxter.

Click INTERACT and RightClick to choose Displays if you wish to change any parameters such as the color of the background or many other items. At first, just accept the default settings.

You can Click INTERACT and RightClick to choose Motion Planning and deselect Displays if desired.

Look at the tabs **Context/Planning/Scene Objects/Stored Scenes/Stored States/Status**

Context	Publish Current Scene AND Save scene to a database.
Planning	Set the start state, the goal state and plan and execute moves of Baxter's arms.
Scene Objects	Import or export scenes such as pillars or tabletops from a disk file.
Stored Scenes	Stored scenes on a database
Stored States	Store and load robot states.
Status	Status

Table 1 Tabs for Motion Planning

PLAN A MOVE

Make sure Baxter's real arms are separated and untucked.

Click INTERACT and RightClick to choose Motion Planning if necessary. Click the Plan tab.

Planning **Query** Select Start State

<current> Update

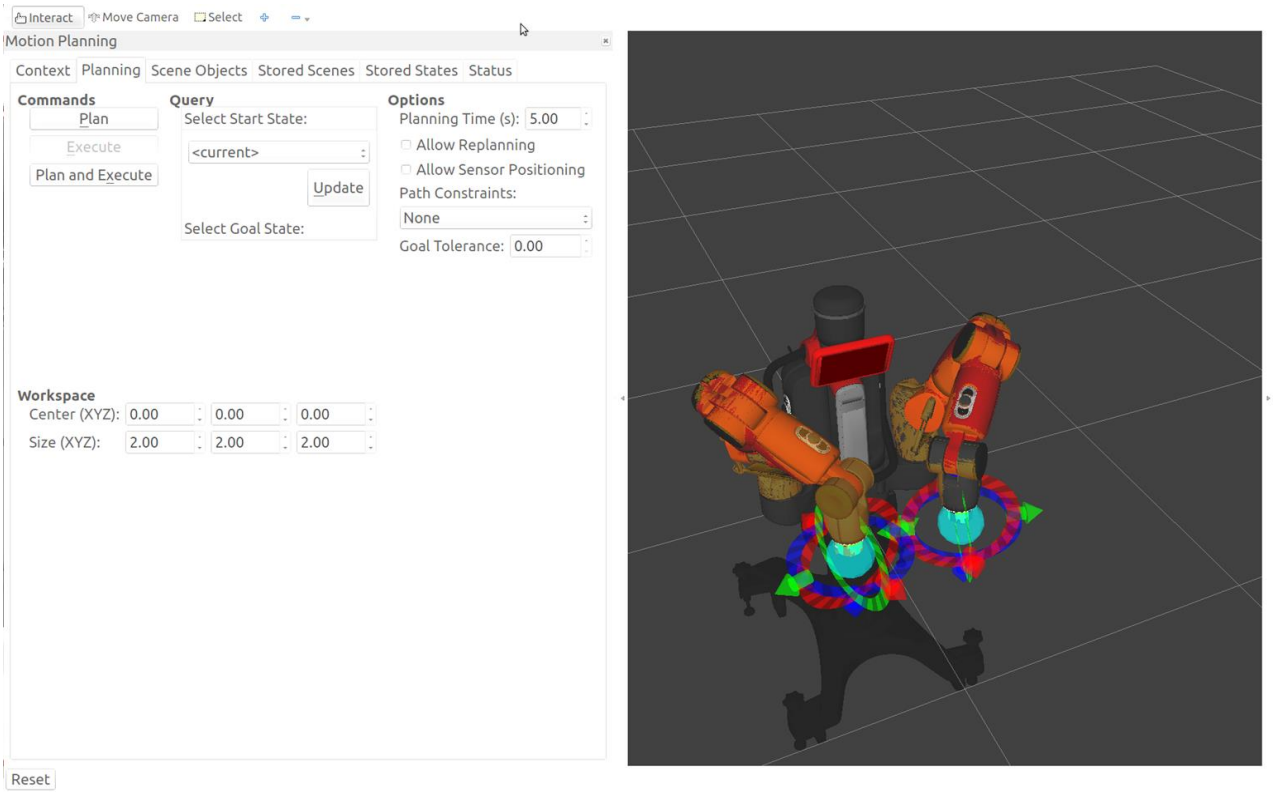


Figure 2 MoveIt screen with Baxter in "Start State"

MOVE BAXTERS ARMS USING MOVEIT

a. To Plan a goal state and move Baxter's real arms by Execute!

1. **Use arrows and rings to move Baxter's simulated arms to desired positions.**

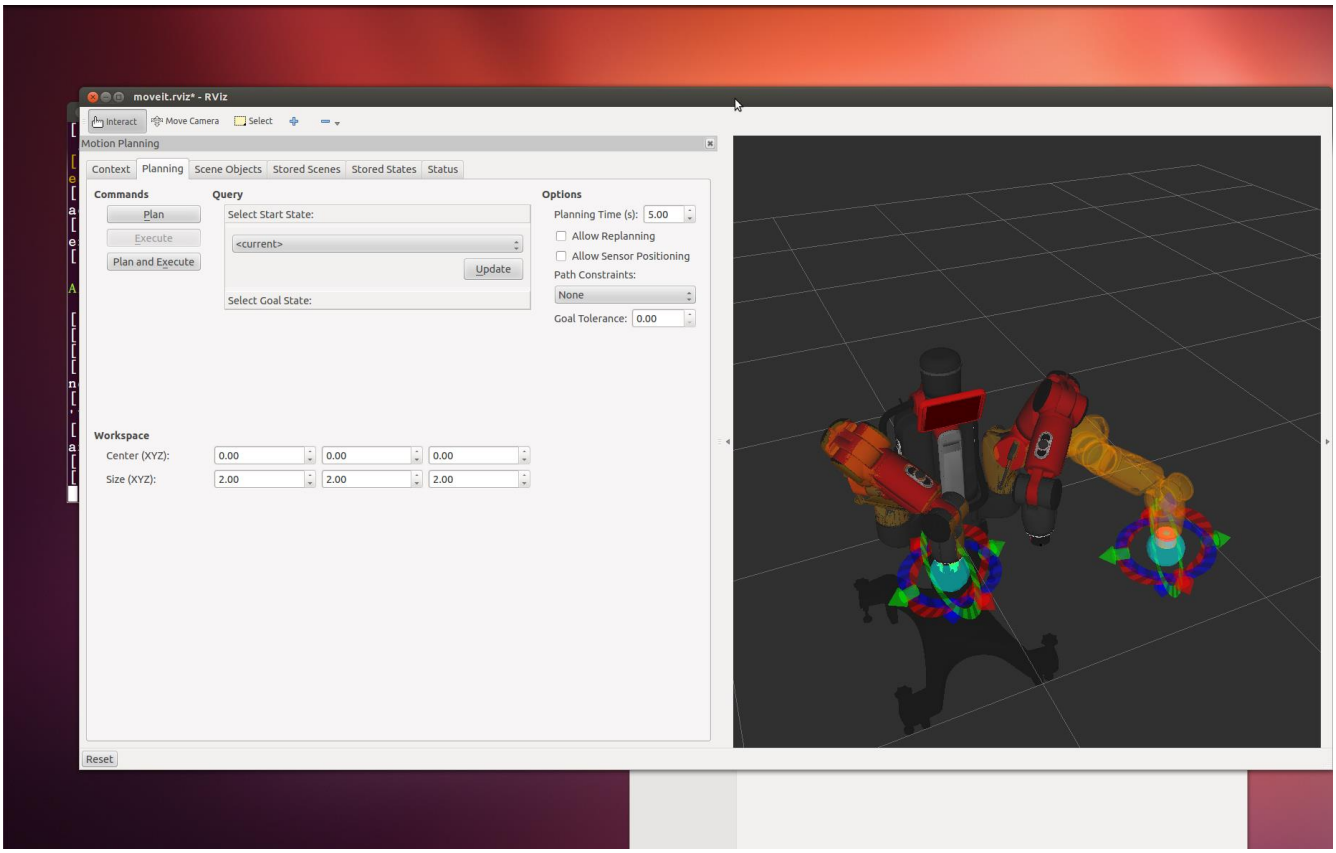


Figure 3 Baxter's Goal Arm Position in MoveIt

2. **Choose Plan to see the trajectory of Baxter's arms in MoveIt.**
3. You should see red arms move from the start state to the final (goal) state.

4. **Execute or Plan and Execute to move Baxter's "real" arms**

You should see Baxter's arms move to the desired goal state if the move was valid.

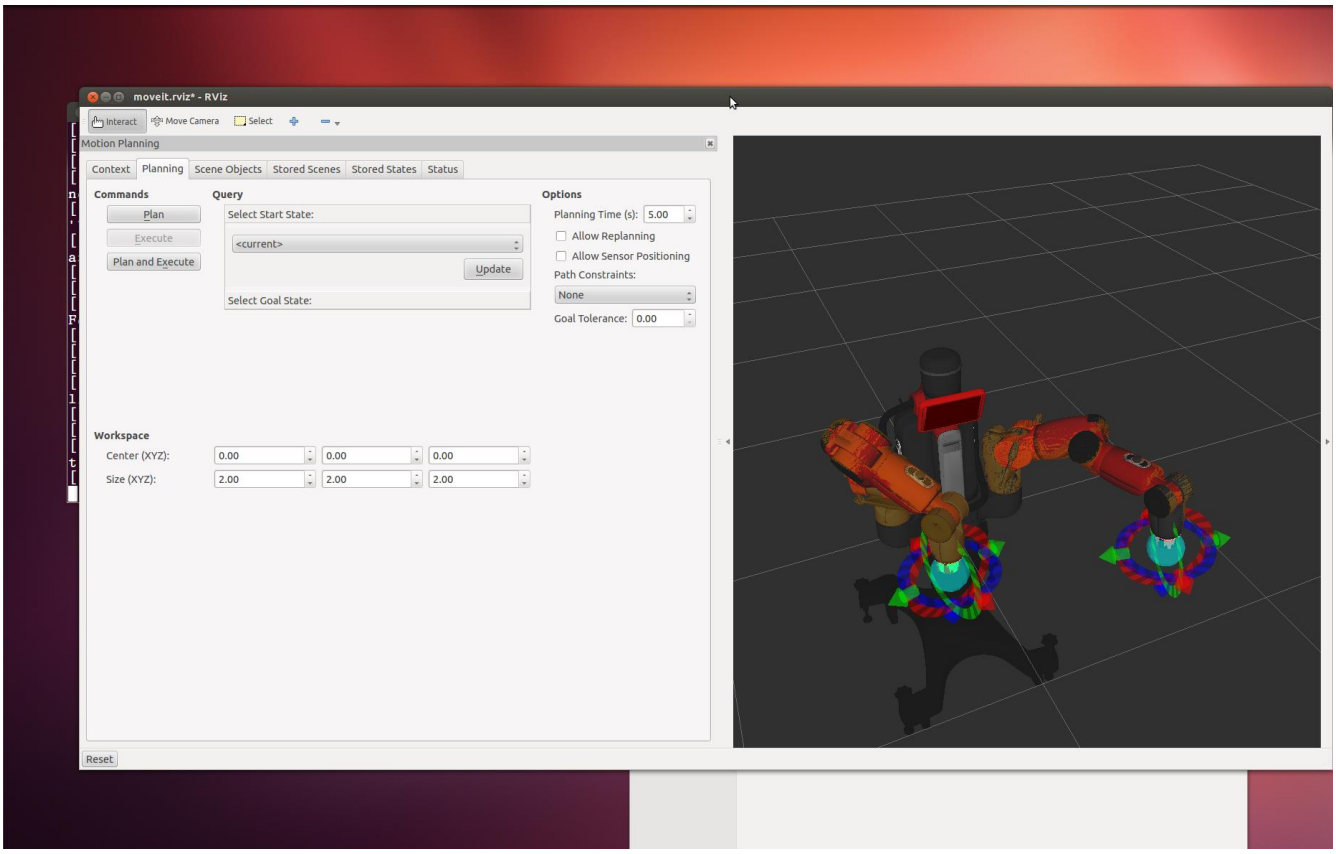


Figure 4 Baxter's Arms in Goal State

b. Move arms back to original start position –

Select Goal State:

<same as start> Update

Then, Plan and Execute again.

ADD OBJECTS TO SCENE

Select the *Scene Object* tab from the Motion Planning frame.

Make sure you know where to find the scenes in the computers directories.

You can now import this scene from the **Scene Geometry** field selecting *Import From Text*

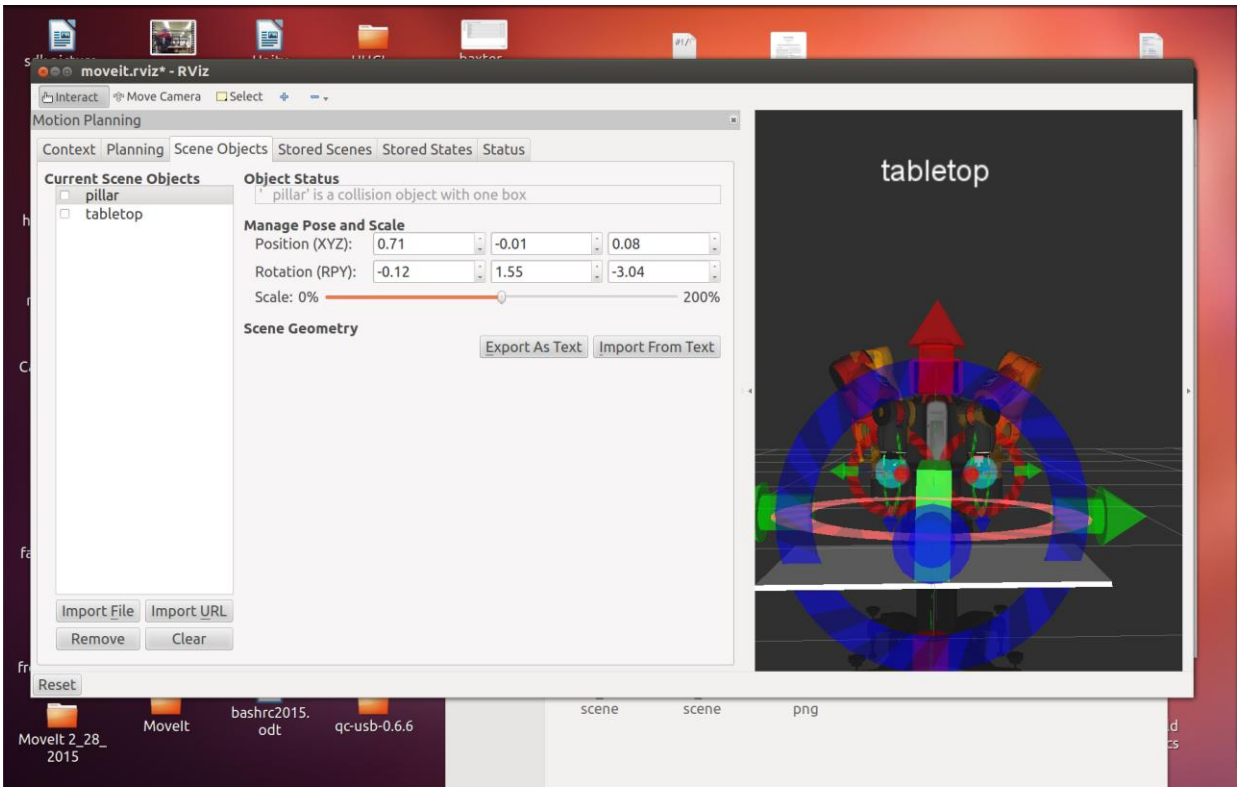


Figure 5 Tabletop and pillar added to scene

In the figure, the names of the objects are shown. In this figure, the file name used:

demo_scene3_3_2015.scene

To manipulate the objects by moving or rotating one, select the object name (not the checkbox) and the arrows and rings should appear. Change position with Green and Blue Arrows and rotate with ring. Moving the Scale: slider will change the size of the object. You can save the scene (Export) after you finish manipulating it. Move the mouse to rotate the scene and roll the mouse wheel to zoom the scene.

Position of Objects

In the figure, the x,y,z of the centroid of the pillar is shown with respect to Baxter's center of gravity. The x-axis extends outward toward the viewer. The positive y-axis is to the right in the view and the z-axis is upward. Thus, the green pillar is at x=0.71 meters out from Baxter's x=0 point. Baxter's z=0

point is at the base to which Baxter attaches. The reference points can be changed in the Display screen. Note that the Roll, Pitch, and Yaw (RPY) are about the x, y, and z axes respectively.

Here is an example of the text file defining the tabletop and pillar:

```
(noname)+
*      pillar
1
box
0.508 0.13056 0.6528           (Height, Width, Length in m.)
0.71 -0.01 0.14                (Position x, y, z)
0.0104953 0.699637 0.0107159 0.714341  (Quaternion)
0 0 0 0
*      tabletop
1
box
0.7 1.3 0.02
0.7 0.04 -0.13
0 0 0 1
0.705882 0.705882 0.705882 1
```

You can change these numbers and save the file and reload the scene to see the effects.

To return to Baxter to move the arms, click on the Planning Tab.

HAVE BAXTER AVOID AN OBJECT

Context tab for baxter to move around obstacles

Selecting the *Context* tab from the Motion Planning frame. Under this tab you must click the *Publish Current Scene* Button under the **Planning Library** field. This tells MoveIt to plan around the obstacles in the modified environment.

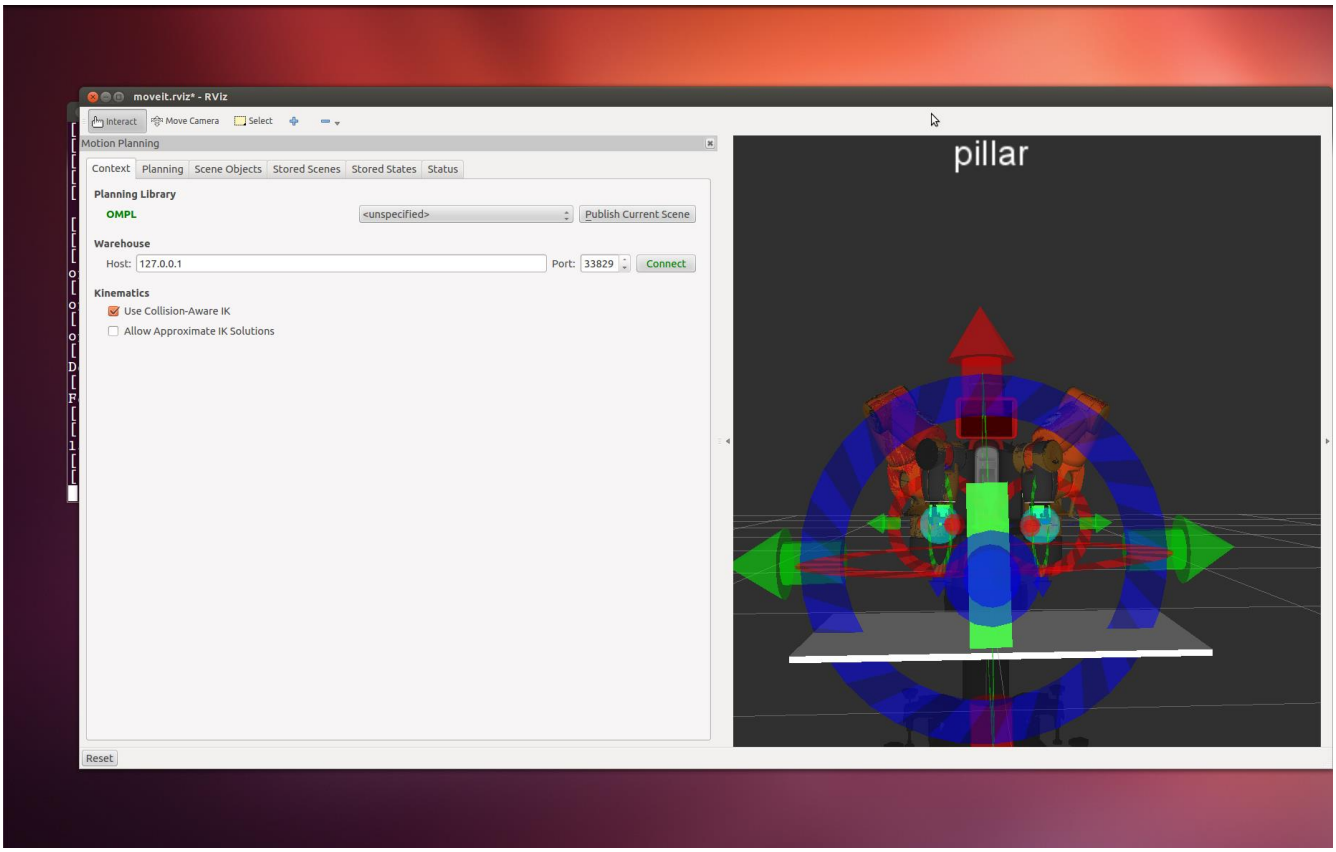


Figure 6 Context Tab

We can now drag our interactive markers for Baxter moving the goal state to a location on the opposite side of the pillar. Each time you hit the *Plan* Button a different arm trajectory path is shown on virtual Baxter. Each path avoids collision with the pillar.

In our example, the left arm is going to be moved to the other side of the obstacle. MoveIt will plan the trajectory so that Baxter's arm with not hit the obstacle.

CAUTION:

We move Baxter's other arm (Right arm in this case) out of the way to avoid any possible collisions.

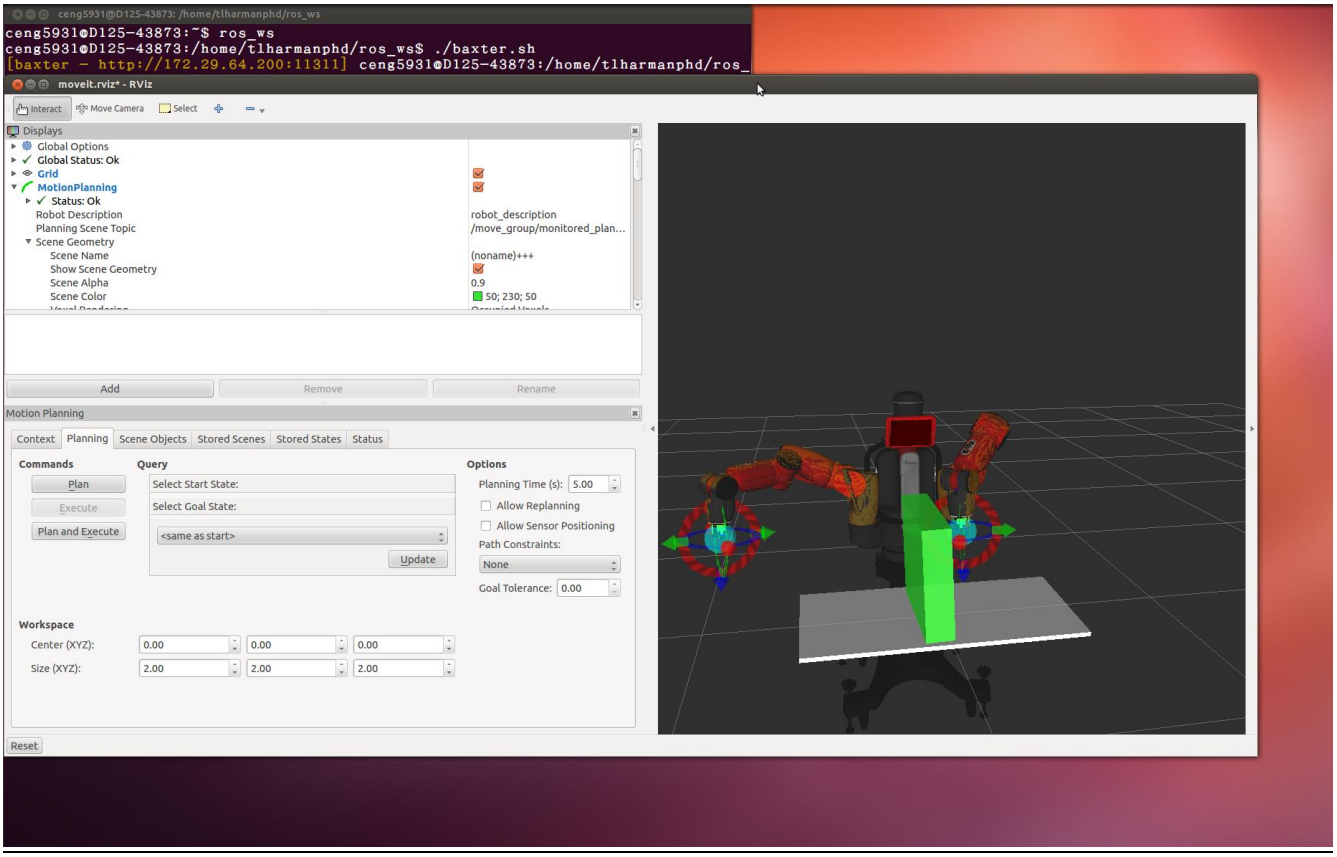


Figure 7 Baxter's simulated right arm moved out of the way

Plan and Execute to move Left Arm Over Obstacle
Move Baxter's Arm Past Obstacle.

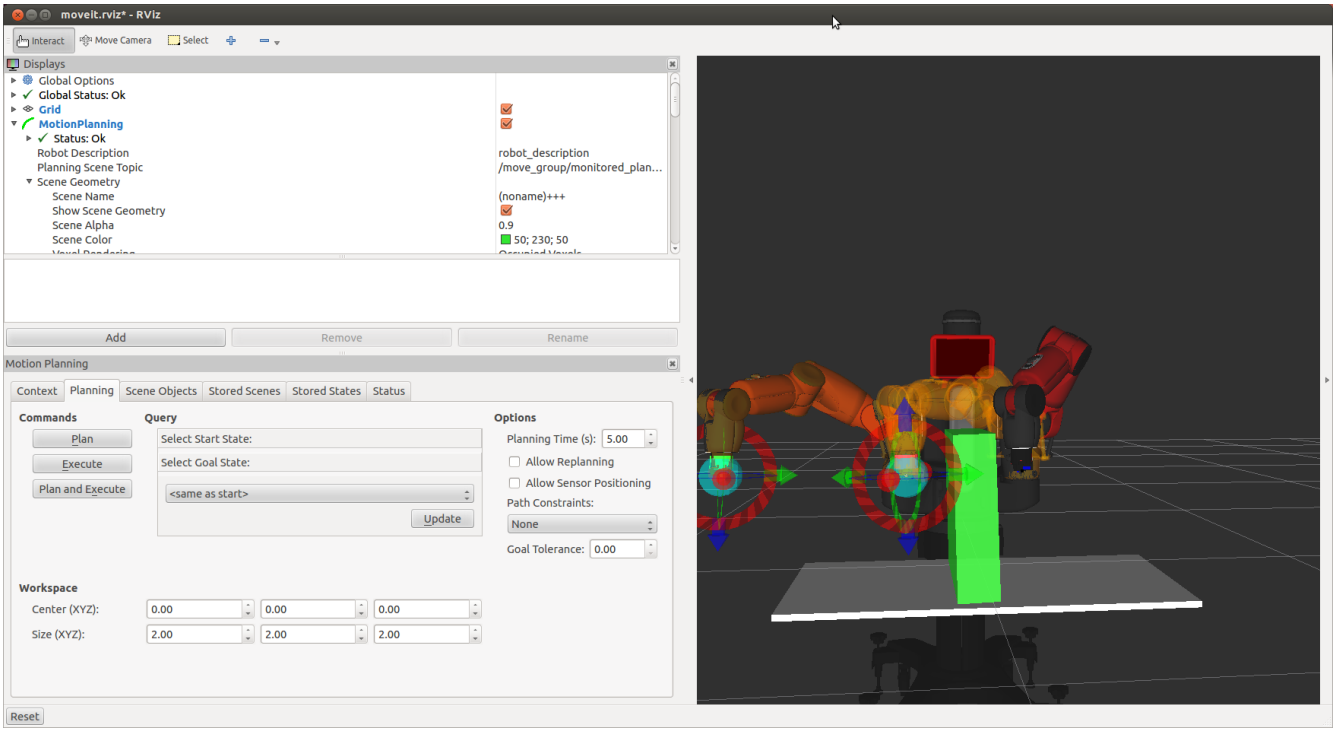


Figure 8 Baxter's Simulated arm moved to other side of obstacle

You should not have to move the arm to avoid the obstacle. MoveIt will plan a safe trajectory that avoids collision.

CAUTION: Sometimes Baxter's real arms get stuck in odd positions. If so, move them apart and restart MoveIt.

LAB WORK WITH BAXTER

Baxter's was setup with a table about at the height of the table in the simulation and with an object for Baxter to avoid.

Caution

The object in MoveIt (pillar) was made about 4 inches taller than the real object on the table because in MoveIt Baxter does not have the grippers installed. The electric gripper extends about 4 inches or so from Baxter's arm end in the simulation.

CAUTION: Sometimes Baxter's real arms get stuck in odd positions. If so, move them apart and restart MoveIt.

SAVE OR LOAD STORED STATES

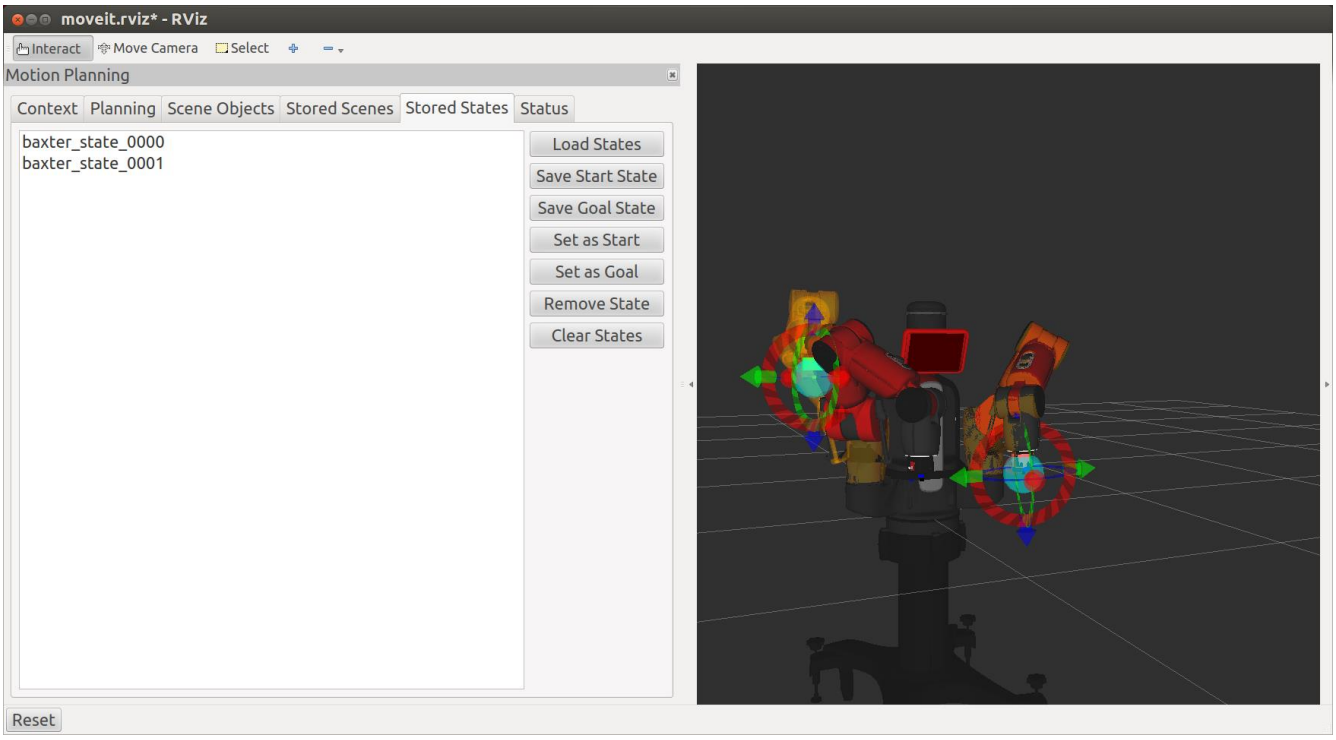


Figure 9 Stored States Tab

The figure shows several previously stored states. A stored state can be reloaded to use the Start or Goal state.

MOVEIT REFERENCES

Learn how to use MoveIt! with the Baxter Research Robot- Video
Published on May 9, 2014
https://www.youtube.com/watch?v=1Zdkwym42P4

MoveIt Tutorial
rethink-bwarzeski edited this page on Feb 12 20
https://github.com/RethinkRobotics/sdk-docs/wiki/MoveIt-Tutorial

MoveIt
http://moveit.ros.org/

MoveIt! Montage 2013
https://www.youtube.com/watch?v=dblCGZzeUqs&feature=player_embedded

https://www.youtube.com/watch?feature=player_embedded&v=dblCGZzeUqs

Various Videos:
https://www.youtube.com/user/moveitrobot

Environment Representation/Rviz
http://moveit.ros.org/wiki/Environment_Representation/Rviz

MoveIt Packages

Index of /packages.ros.org/ros/ubuntu/pool/main/r

https://mirror.umd.edu/packages.ros.org/ros/ubuntu/pool/main/r/

ros-groovy-moveit-commander/ 12-Sep-2014 23:00 -
ros-groovy-moveit-core/ 12-Sep-2014 23:06 -
ros-groovy-moveit-full/ 12-Sep-2014 23:02 -
ros-groovy-moveit-full-pr2/ 12-Sep-2014 23:05 -
ros-groovy-moveit-msgs/ 12-Sep-2014 23:02 -
ros-groovy-moveit-planners/ 12-Sep-2014 23:04 -
ros-groovy-moveit-planners-ompl/ 12-Sep-2014 23:01 -
ros-groovy-moveit-plugins/ 12-Sep-2014 23:00 -
ros-groovy-moveit-pr2/ 12-Sep-2014 23:04 -
ros-groovy-moveit-resources/ 12-Sep-2014 23:02 -
ros-groovy-moveit-ros/ 12-Sep-2014 23:04 -
ros-groovy-moveit-ros-benchmarks/ 12-Sep-2014 23:04 -
ros-groovy-moveit-ros-benchmarks-gui/ 12-Sep-2014 23:02 -
ros-groovy-moveit-ros-manipulation/ 12-Sep-2014 23:05 -
ros-groovy-moveit-ros-move-group/ 12-Sep-2014 23:06 -
ros-groovy-moveit-ros-perception/ 12-Sep-2014 23:02 -
ros-groovy-moveit-ros-planning/ 12-Sep-2014 23:01 -
ros-groovy-moveit-ros-planning-interface/ 12-Sep-2014 23:04 -
ros-groovy-moveit-ros-robot-interaction/ 12-Sep-2014 23:03 -
ros-groovy-moveit-ros-visualization/ 12-Sep-2014 23:01 -
ros-groovy-moveit-ros-warehouse/ 12-Sep-2014 23:00 -
ros-groovy-moveit-setup-assistant/ 12-Sep-2014 23:06 -
ros-groovy-moveit-simple-controller-manager/ 12-Sep-2014 23:02 -
ros-groovy-multi-level-map/

APPENDICES Do packages, topics , etc on Baxter

Previous Runs: In the Second terminal window:

[INFO] [1415819095.854055030]: Planning attempt 1 of at most 1
[INFO] [1415819095.862139351]: No planner specified. Using default.
[INFO] [1415819095.862412358]: Attempting to use default projection.
[INFO] [1415819095.863790959]: Starting with 1 states
[INFO] [1415819095.884643118]: Created 46 (18 start + 28 goal) states in 34 cells (18 start (18 on boundary) + 16 goal (16 on boundary))
[INFO] [1415819095.884699491]: Solution found in 0.022035 seconds
[INFO] [1415819095.886810868]: Path simplification took 0.002011 seconds
[INFO] [1415819293.007942946]: Loaded scene geometry from '/home/tlharmanphd/ros_ws/src/moveit_robots/baxter/baxter_moveit_config/baxter_scenes/baxter_pillar.scene'
[INFO] [1415819487.539798605]: Planning request received for MoveGroup action. Forwarding to planning pipeline.
[INFO] [1415819487.543723409]: No planner specified. Using default.
[INFO] [1415819487.543767585]: Attempting to use default projection.
[INFO] [1415819487.544253799]: Starting with 1 states
[INFO] [1415819487.686045847]: Created 701 (595 start + 106 goal) states in 410 cells (363 start (304 on boundary) + 47 goal (40 on boundary))
[INFO] [1415819487.686098847]: Solution found in 0.142275 seconds
[INFO] [1415819487.768547326]: Path simplification took 0.082403 seconds
[INFO] [1415819497.350501669]: Planning request received for MoveGroup action. Forwarding to planning pipeline.
[INFO] [1415819497.359955286]: Starting with 1 states
[INFO] [1415819497.480820067]: Created 566 (443 start + 123 goal) states in 289 cells (209 start (160 on boundary) + 80 goal (79 on boundary))
[INFO] [1415819497.480877989]: Solution found in 0.121620 seconds
[INFO] [1415819497.524113641]: Path simplification took 0.043184 seconds
[INFO] [1415819504.486034160]: Planning request received for MoveGroup action. Forwarding to planning pipeline.
[INFO] [1415819504.494148088]: Starting with 1 states
[INFO] [1415819504.571812341]: Created 845 (668 start + 177 goal) states in 465 cells (365 start (282 on boundary) + 100 goal (87 on boundary))
[INFO] [1415819504.571856863]: Solution found in 0.078298 seconds
[INFO] [1415819504.599893765]: Path simplification took 0.027991 seconds

The shell in which you launched demo_baxter.launch will provide information regarding which planner will be used, how long it took to find a solution, path simplification/smoothing time, and more. This will also display if your planner was unsuccessful in finding an allowable solution. This is often caused by collision with the environment during testing of the execution or invalid start/goal states. In very constrained or difficult motions, you may have to plan several times to find an acceptable solution for execution.

Upon execution the robot will avoid this 'virtual' object tracking the commanded trajectory. -- and it

did!!!!

TRYING TO ADD SCENES TO BAXTER MOVEIT

Repository of PR2 .scene files:
https://github.com/isucan/plannerarena/tree/master/problems/pr2_scenes

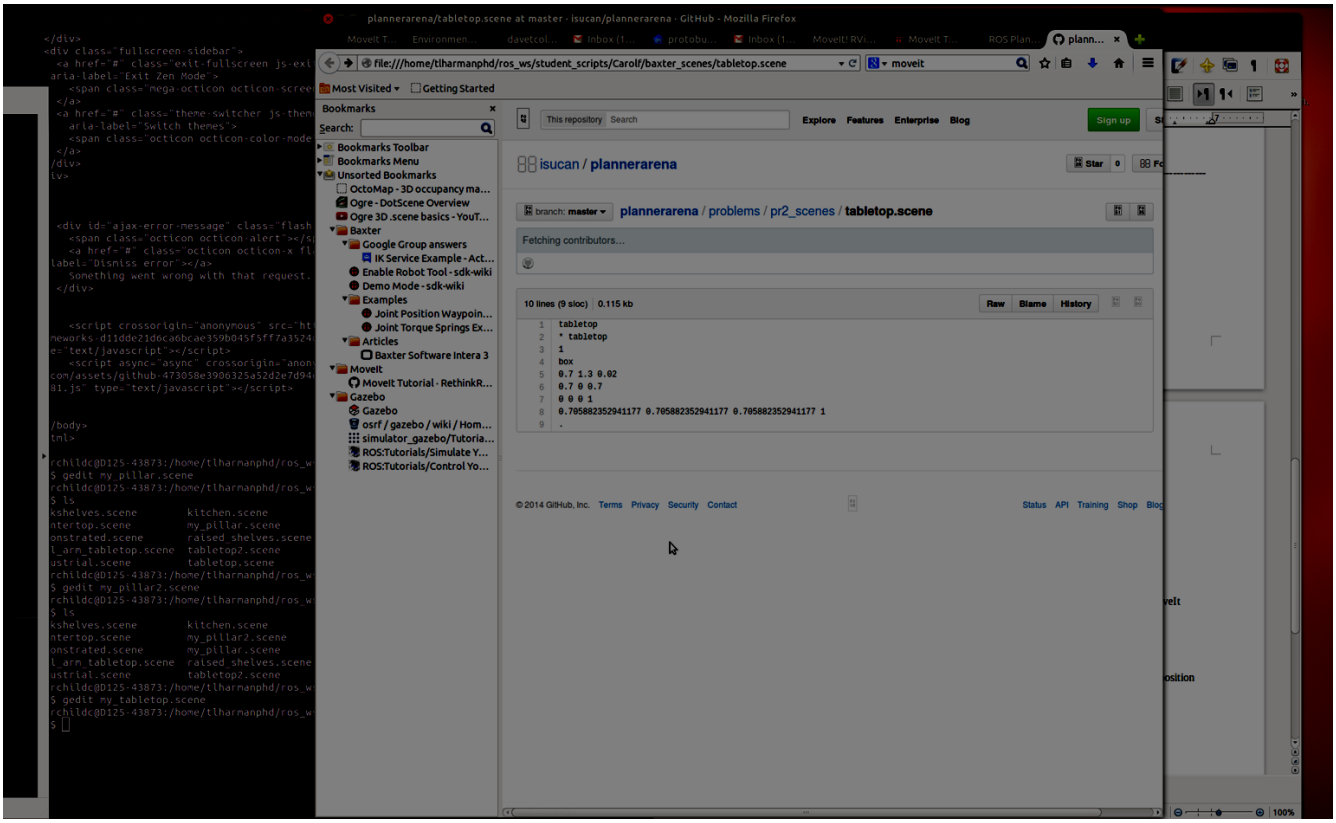
copied to /home/tlharmanphd/ros_ws/student_scripts/Carolf/baxter_scenes:

bookshelves.scene	kitchen.scene	video_3x2.scene
countertop.scene	raised_shelves.scene	video.scene
demonstrated.scene	tabletop2.scene	warehouse_demonstration.scene
dual_arm_tabletop.scene	tabletop.scene	
industrial.scene	tunnel.scene	

These .scene files were loaded but not displayed in MoveIt environment.
[INFO] [1415831411.921008627]: Loaded scene geometry from
'/home/tlharmanphd/ros_ws/student_scripts/Carolf/baxter_scenes/tabletop.scene'

I was able to create a my_pillar2.scene file in this directory by copying the data from the MoveIt tutorial. I saved it and used *Import From Text* to load it into the MoveIt environment.
[INFO] [1415830842.363200897]: Loaded scene geometry from
'/home/tlharmanphd/ros_ws/student_scripts/Carolf/baxter_scenes/my_pillar2.scene'
I selected it under **Current Scene Objects** (not using checkbox) and was able to change its position and orientation.

Since that worked for my_pillar2.scene, I tried it with tabletop.scene. Somehow I was able to get this:



I typed:
fairchildc@D125-43873:/home/tlharmanphd/ros_ws/student_scripts/Carolf/baxter_scenes$ gedit my_tabletop.scene
Then I copied the data from the screen to the file, saved it and exited. When I used *Import From Text* to load it into the MoveIt environment and was able to manipulate the tabletop like the pillar.

Attributes:
frame (required)
The base frame of the device, relative to the global frame
xyz (optional – defaults to ‘0 0 0’)
The position of the base frame, relative to the global frame
rpy (optional – defaults to ‘0 0 0’)
The rotation of the base frame, yaw/pitch/roll (in radians). Represented as an intrinsic rotation: first yaw (around z), then pitch (around y) and finally roll (around x).