

Commanding the UR5-e using MoveIt and the Dynamic Obstacle Avoidance Program

1. Power on the UR5e
2. Ensure all required repositories are sourced, if editing bash the following four lines should be present at the bottom of the file:

```
#source $HOME/ws_moveit/devel/setup.bash
#source $HOME/pyKinectAzure/devel/setup.bash
#source $HOME/UR_Driver/devel/setup.bash
#source $HOME/universal_robot/devel/setup.bash
```

3. Once robot is powered on, open a new terminal, and run the following command (found in terminal_commands.txt)

```
roslaunch ur_robot_driver ur5e_bringup.launch robot_ip:=192.168.12.100 \
  kinematics_config:=$HOME/my_robot_calibration.yaml
```

Depending on where your calibration file is located may need to change path, if working correctly terminal should display a message that the robot is powered off.

4. While the robot IP should be correct, if there are difficulties connecting to the lab computer or your own device, you will probably have to change the IP in the program settings to match your network.
5. On the UR5e load your program and make sure mode is in automatic
6. Open a new terminal and run the following command (found in terminal_commands.txt):

```
roslaunch ur5e_moveit_config moveit_planning_execution.launch
```

If working correctly terminal should display a message that MoveIt is ready to begin motion planning

7. Finally, only a third terminal and run the following command (found in terminal_commands.txt):

```
roslaunch ur5e_moveit_config moveit_rviz.launch
```

This will open an RVIZ window which can be used to move the robot, as well as visualizing planned trajectories.

To use the dynamic obstacle avoidance program:

1. Ensure that the file **'body_tracking_ros_publisher.py'** is executable before running it, made need to be recalibrated depending on orientation, file requires an azure Kinect sensor to be connected to your device.
2. Once the body tracking script is running a window should display showing the tracked skeleton of the operator
3. Ensure that **'UR5e_dynamic_obstacle_avoidance.py'** is executable before running, script will execute and perform dynamic collision avoidance. Target positions can be changed, and more joints added to suit application.
4. Run executables as usual through terminal, or alternatively, run and debug with visual code. A source build of MoveIt is useful for in-depth debugging.

Note on simulation.

1. To run using a simulator instead of the physical robot, run:
roslaunch ur_gazebo ur5e_bringup.launch and **roslaunch ur5e_moveit_config moveit_planning_execution.launch sim:=true** instead of the first two commands, RVIZ can still be launched in the same way, this will load a model of the UR5e into gazebo which can be commanded using MoveIt in the same fashion as the physical robot