

Baxter Simulator Exercise

Type in your Seminar Username and password.

When the Unity Desktop comes up, press Ctrl + Alt + T and bring up a terminal window.

Reference: These exercises outline the information and commands for the Baxter Simulator presented in ROS Robotics By Example, Chapter 6 –Wobbling Robot Arms Using Joint Control. Pages 255 – 290 in this book will provide a description of the commands and additional information about Baxter’s behavior.

Tip: Remember to use tab completion to see the fields of the message!

In order to perform these exercises, you should have completed the **Instructions for setting up Baxter Simulator in D158** and you should be able to bring up Baxter Simulator in Gazebo.

For these exercises, download any code needed from the following site:

https://github.com/FairchildC/ROS-Robotics-By-Example-2nd-Edition/tree/master/Chapter6_code

Starting at page 255 (*Bringing Baxter Simulator to life*), proceed through reading and understanding the text and performing the commands with Baxter Simulator. To start Gazebo and the Baxter simulator, first open a terminal window and type the commands:

```
$ cd ~/baxter_ws  
$ ./baxter.sh sim
```

Be sure that the command prompt begins with **[baxter - http://localhost:11311]** then enter the roslaunch command:

```
$ roslaunch baxter_gazebo baxter_world.launch
```

In a new terminal window, check the status of the robot then enable it with the commands:

```
$ cd ~/baxter_ws  
$ ./baxter.sh sim
```

```
$ rosrun baxter_tools enable_robot.py -s  
$ rosrun baxter_tools enable_robot.py -e  
$ rosrun baxter_tools enable_robot.py -s
```

(page 261) Display an image on Baxter’s head display with the command:

```
$ rosrun baxter_examples xdisplay_image.py --file=`rospack find  
baxter_examples`/share/images/baxterworking.png
```

Head wobbler will move Baxter’s head up and down then side to side:

```
$ rosrun baxter_examples head_wobbler.py
```

Use Ctrl+C to kill the process.

Move Baxter's arms to the "untuck" position with the command:

```
$ rosrun baxter_tools tuck_arms.py -u
```

Move Baxter's arms in a joint velocity wobble with the command:

```
$ rosrun baxter_examples joint_velocity_wobbler.py
```

Press Ctrl-C to stop...

(page 266) Use the keyboard to control Baxter's arms and grippers, enter this mode with the command:

```
$ rosrun baxter_examples joint_position_keyboard.py
```

Use Ctrl+C to kill the process.

(skip Controlling arms and grippers with a joystick)

(page 270) Using a Python script to "home arms" commanding Baxter's arms into a specific pose:

```
$ python home_arms.py
```

(skip Recording and replaying arm movements)

(page 273) Joints and joint state publisher

```
$ python home_arms.py
```

```
$ rostopic echo /robot/joint_states -n1
```

```
$ rostopic type /robot/joint_states
```

```
$ rostopic echo /robot/limb/left/endpoint_state/pose -n1
```

(page 279) Using a Python script to "zero angles" commanding Baxter's arm joints to zero angles:

```
$ python arms_to_zero_angles.py
```

To command Baxter's joint angles from the command line, first move Baxter's arms to another pose:

```
$ python home_arms.py
```

Then publish the joint_command topic to move Baxter's left arm:

```
$ rostopic pub /robot/limb/left/joint_command baxter_core_msgs/JointCommand "{mode: 1, command: [0.0, 0.0, 0.0, 0.0], names: ['left_w1', 'left_e1', 'left_s0', 'left_s1']}" -r 10
```

Use Ctrl+C to kill the process.

(page 283) View Baxter's tf tree using the following commands:

```
$ rosrun tf view_frames
```

```
$ evince frames.pdf
```

MoveIt!

(page 284) Restart Gazebo and the Baxter simulator with the following commands:

```
$ cd baxter_ws
$ ./baxter.sh sim
$ roslaunch baxter_gazebo baxter_world.launch
```

In a 2nd terminal window, untuck Baxter's arms and start the Python script that starts the joint_trajectory_action_server :

```
$ cd baxter_ws
$ ./baxter.sh sim
$ rosrn baxter_tools tuck_arms.py -u
$ rosrn baxter_interface joint_trajectory_action_server.py
```

In a 3rd terminal, start MoveIt! and wait for the response:

```
$ cd baxter_ws
$ ./baxter.sh sim
$ roslaunch baxter_moveit_config baxter_grippers.launch
```

Follow instructions from the book starting at page 285 and ending on page 292.

Using a state machine to perform YMCA

(page 303) Using a Python script with a implementation of a ROS state machine, move Baxter's arms to different poses to form the letters – Y M C and A:

```
$ python YMCAStateMach_for_Sim.py
```