ECE 470: Project Update 1

Team: Jeffery Zhou, Kenneth Tochihara, Charlie Ray

TA: Dhruv Mathur

Section: Monday 9AM

Team Name: GouBot

Brief Project Update

The goal of this project is to create a robotic dog that can fetch a ball and bring it back home. We envisioned this robot to be on wheels which could be designed using URDF like we learned in class or by importing a CAD image made separately using Solidworks. GouBot is derived from the word "Gou" translates to "dog" in Mandarin and is the reason behind the naming convention for this project.

Using the given Ubuntu Image, ECE470VM, a virtual machine (VM) was set up on a computer with VMware Fusion. This was done to minimize the amount of overhead in setup and to quickly establish a working environment to interact with the robot. To test the simulation environment, modifications were made to the `lab2andDriver` package to create an initial environment of interacting with the UR3 robot. Using the content from lab2, principles of Robot Operating System (ROS) was applied to verify connections between the Gazebo simulator and the Python controller. Nodes, publications, and subscriptions were established to verify the movements of the joint and the gripping end-effector in our executions.

After initializing these elements, it was essential to test the robot by writing some simple move commands. To test the robot and verify that the Gazebo environment was fully functional and compatible with ROS, several commands were sent to the robot. First it rotated each joint by 30° and rotated back to its home position using the following commands.

```
# loop through joints
for idx in range(6):
    # log index of joint
    rospy.loginfo("Sending offsets for joint " + str(idx + 1) +  "...")

    # offset joint by 30 degrees
    joint_angles[idx] += np.radians(30)
    move_arm(pub_command, loop_rate, joint_angles, 4.0, 4.0)

    # return joint back to original location
    joint_angles[idx] -= np.radians(30)
    move_arm(pub_command, loop_rate, joint_angles, 4.0, 4.0)
```

Other the gripper input and output values were tested by verifying the messages so that the user could see if the gripper was turned on or off at a given time. Furthermore, it tested the ability read gripper sensor inputs to further verify that sensor inputs were established from the simulator.

The tests in Gazebo using ROS were recorded in the YouTube playlist and in the codebase on GitHub.