LAB 1a: INTRODUCTION TO ROS AND BAXTER

CS6310/5310 ME6220/5220

1. Introduction

This lab introduces students to the Robot Operating System (ROS) interface used by the Baxter robots. The goal of this lab is to achieve a basic understanding of the ROS environment and recognize how it integrates with a Baxter robot. By the end of this lab, students are expected to know the following:

- What a ROS Node is and how ROS Topics are used to communicate between nodes
- How to manually locate and monitor various ROS Topics and ROS Nodes using the rostopic and rosnode commands
- How to run Baxter demo programs.

2. Linux

This section is for students unfamiliar with the GNU command line in a Linux environment. If a student feels competent in this subject, they are free to skip this section.

To open the command line interface in Ubuntu 12.04, the version used on the computer controlling the Baxter robots, go to the search button at the top of the tool bar. Type Terminal and open the top result. This will open up the command line interface. A new window can be opened by pressing ctrl-shift-n, or a new tab can be opened by pressing ctrl-shift-t. Keep these shortcuts in mind because the ROS interface requires the use of multiple windows and tabs. The following tutorial will give you a brief overview of the function and use of common Unix commands. The following tutorial is to be completed before your assigned lab session and can be done on your own computer or on one of the Linux computers in the CADE lab.

Tutorial: http://freeengineer.org/learnUNIXin10minutes.html

2.1. Basic Commands

There are a few basic commands that you need to be familiar with for this lab and the following labs. Make sure you understand what they are and how to use them. You may want to keep this list as a reference.

- $cd

  Change Directory. This command is passed a string containing the path of the desired directory.

- $ls

  List Directory. This command lists all the contents of the current directory. This command may also be passed a path for a different directory to list.

- $pwd

  Print Working Directory. This command prints the absolute path of the current working directory.

- $mkdir

  Make Directory. This command creates a directory in the current working directory. This command is passed the name of the new directory.

- $chmod

  Change Mode. This command is used to change the permissions of a file or directory. See tutorial for use.

- $chown

  Change Owner. This command is used to change the owners of a file or directory. See tutorial for use.

3. Robot Operating System

The Robot Operating System (or ROS) is a piece of software written by Willow Garage. ROS allows users to break large problems down into more sizable chunks and the write programs to solve these smaller problems. These programs then communicate with each other forming a network that solves the larger problem.

The individual programs running in a ROS environment are called nodes. These nodes talk to each other using topics. A topic may contain anything from a single byte of information to any combination of integers, floating point numbers, and strings. Nodes may also communicate with each other via services. A service is a topic that sends a response packet when a new packet is received, typically after some action has been taken.

All of the communication between nodes is handled by the roscore program. You need one instance of this program running for each ROS network. A ROS network may span multiple computers via the internet. This is done by a computer linking to the roscore running on a remote machine.

3.1. Lab Activities

During this section of the lab, students are expected to become familiar with ROS basics.

This Portion of the lab may be done on a personal computer or the computer in the Large Robotics Lab (KENN 0112). The ROS server may also be installed on a personal computer with a Linux operating system using the tutorials given at http://wiki.ros.org/ROS/Installation.

For this section of the lab, students need to go to http://wiki.ros.org/ROS/Tutorials and complete the Understanding Nodes and Understanding Topics tutorials.

4. Baxter

The Baxter robot is a light industrial robot designed to be safe to work around humans. It has two arms and a head. Each arm has 7 degrees-of-freedom and has a variety of grippers that can be attached. In this lab, we have standard pincer grippers with variable size pincers. Each arm also has a camera as well as an IR range finder at the base of each wrist so the robot can see what it is picking up. The Baxter robot has a swivel head with a monitor on it as well as a camera and an array of sonar range finders for detecting nearby objects.

4.1. Lab Activities

For this part of the lab, students will run a few demo programs and get an idea of the Baxter robot's capability.

1. Setup the Baxter environment by running the baxter.sh script in the Baxter directory of the student account.

2. The first demo students should run is the keyboard demo. This demo allows you to manipulate the joints in the Baxter's arms manually. The following keys move the arms:

    - (1 4)-right shoulder 0
    - (2 3)-right shoulder 1
    - (q r)-right elbow 0
    - (w e)-right elbow 1
    - (a f)-right wrist 0
    - (s d)-right wrist 1
    - (z v)-right wrist 2
    - (x c)-right gripper
    - (9 6)-left shoulder 0
    - (7 8)-left shoulder 1
    - (y o)-left elbow 0
    - (u i)-left elbow 1
    - (h l)-left wrist 0
    - (j k)-left wrist 1
    - (n .)-left wrist 2
    - (m ,)-left gripper

Run this node by typing the following command:

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

Note that a roscore does not need to run on the computer as it is running on the Baxter. The ROS environment on the computer connects to the Baxter automatically.

3. The next demo students should run is the puppet demo. This demo will make one arm of the Baxter mimic the movements of the other. To run this node type the following command:

   ```
   $rosrun baxter_examples joint_velocity_puppet.py -l left
   ```

   This node allows the user to manipulate the left arm while the right arm will follow.

4. The final demo for this lab will involve two programs. First, run the joint_recorder.py node of the baxter_cs5310my_baxter package with the following commands: -l/-r –f <filename>. Here is an example command:

   ```
   $rosrun baxter_cs5310 joint_recorder -l -f my_data.txt
   ```

   This program will allow a user to move the robots arm to various positions and save them to be replayed later. The filename is the file where your arm positions are saved, the -l command is which limb you want to manipulate. Press c to save the starting position as position 0. Now move the arm to some other position. Press c again to save this position. Repeat this process until you have saved about 5 positions. Press e to exit the program.

   For the second part of this demo, run the joint_player node of the baxter_cs5310 package with the same limb and file commands given to the joint recorder.py node. This node will play back the positions recorded by the joint recorder node.

   ```
   $rosrun baxter_cs5310 joint_player -l -f my_data.txt
   ```