The Robographer: Progress Review #8

Name: Rohit Dashrathi
Team: G (The Robographers)
Teammates:
Jimit Gandhi
Gauri Gandhi
Rohit Dashrathi
Sida Wang
Tiffany May

ILR No.: # 7
Submission Date: Feb 11, 2016

1. Individual Progress

Responsibilities: Project management
Learning Simulation using Gazebo

Softwares/tools Used: Google Drive, Google Calendar, Microsoft Excel, Gazebo, ROS

Task Description:

Following tasks were assigned to me and completed before the progress review 8:

a. Conducting team meeting and deciding goals for PR#8
b. Installation of Ubuntu 14.04 LTS along with ROS Indigo
c. Tele operate Turtlebot in the Gazebo environment
d. Creating the world for turtlebot movement

a. Setting the goals for PR#8:

Conducting team meeting and deciding goals for PR#8

1. Upgrade the turtlebot workstation computers to ROS Indigo and Ubuntu 14.04.
2. Implementation of the Robotics in Concert (ROCON) framework in Gazebo
3. Real-time ROCON implementation with the actual Turtlebots

b. Installation of Ubuntu 14.04 LTS along with ROS indigo:

I was assigned to learn the simulation module Gazebo which comes readily with ROS. As the ROS (Robot Operating System) works only on the Ubuntu operating system, I was compelled to install Ubuntu first on my laptop. Being the most updated stable version of Ubuntu, I installed Ubuntu 14.04 LTS on my laptop along with the existing Windows 7 partition. For this I created a bootable USB stick on windows[i] itself. To achieve this, firstly I downloaded ‘pen drive Linux’s USB installer’[ii]. This installer is known as UUI and is a Live Linux USB Creator that allows the user to choose from a selection of Linux Distributions to put on the USB Flash Drive. Further, I downloaded the Ubuntu 14.04 distribution in .iso[iii] format from the Ubuntu official website download section. Using the Ubuntu installer, I loaded the downloaded .iso file on a pen drive connected to my laptop. This created a bootable Ubuntu disk on the connected USB.

Furthermore, I proceeded to install the Ubuntu OS on the laptop[iv]. For this, I restarted the laptop while keeping the pen drive connected. While the laptop restarted, I pressed F12 to put it in the BIOS settings menu which is shown as follows:

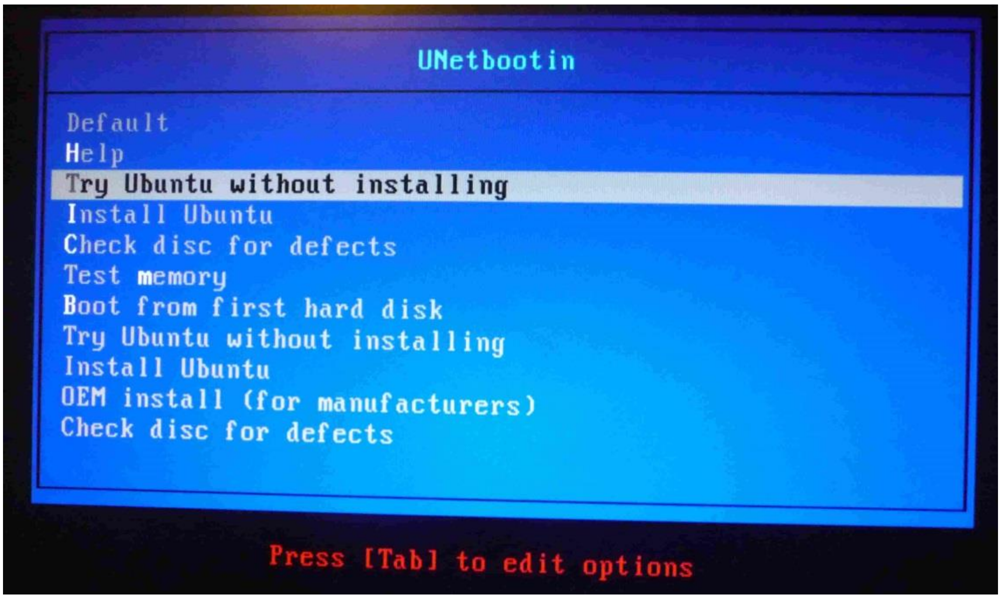
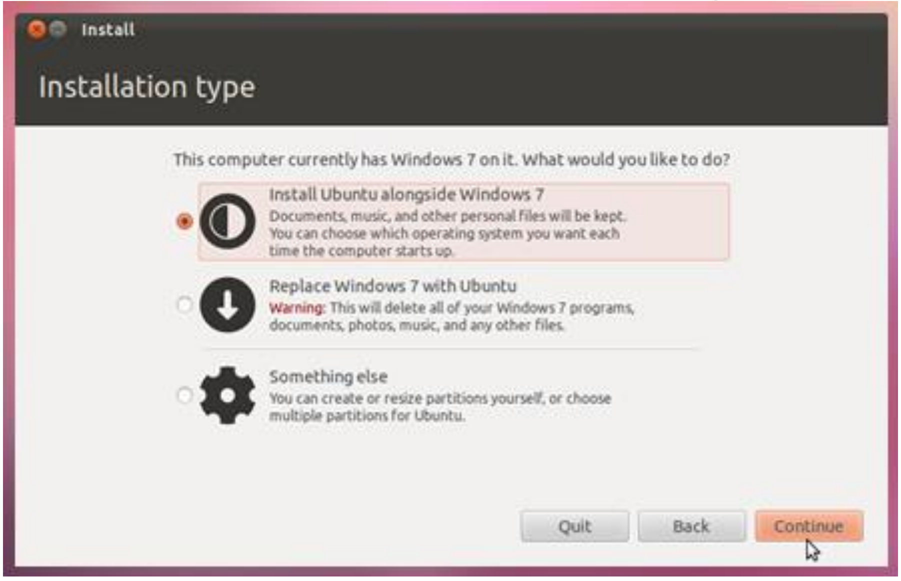

UNetbootin
Default
Help
Try Ubuntu without installing
Install Ubuntu
Check disc for defects
Test memory
Boot from first hard disk
Try Ubuntu without installing
Install Ubuntu
OEM install (for manufacturers)
Check disc for defects
Press [Tab] to edit options

Figure 1: Ubuntu Installation-Laptop put in BIOS menu

Further, I selected the ‘Install Ubuntu’ option from the BIOS menu which put the system in the Ubuntu installation process, as shown in the figure 2.



This computer currently has Windows 7 on it. What would you like to do?

● Install Ubuntu alongside Windows 7
Documents, music, and other personal files will be kept. You can choose which operating system you want each time the computer starts up.

○ Replace Windows 7 with Ubuntu
Warning: This will delete all of your Windows 7 programs, documents, photos, music, and any other files.

○ Something else
You can create or resize partitions yourself, or choose multiple partitions for Ubuntu.

Figure 2: Ubuntu Installation- Installing alongside windows

This was followed by the regular procedures of allotting Ubuntu the required space (min 60GB is recommended) and further details. Eventually, Ubuntu was installed successfully alongside windows. It should be noted here, the option of installing Ubuntu alongside Windows 7 should be chosen carefully. Often, the default option being selected by the system is ‘Replace Windows 7 with Ubuntu’ which causes the existing Windows OS files to be deleted. This leaves no space to recover the existing files. Hence, it is a good idea to always backup the files before performing this action.

After installing Ubuntu, I needed to install the ROS indigo package[v] on the same. Hence, I logged into Ubuntu, opened the terminal window and put the following command to setup the sources.list, which makes the computer accept the software from package.ros.org:

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc
/apt/sources.list.d/ros-latest.list'
```

This was followed by another command to set up the keys:

```
sudo apt-key adv --keyserver hkp://pool.sks-keyservers.net --recv-key 0xB01
FA116
```

Then, I updated my debian package index. Debian (deb) packages are nothing but the standard Unix archives. They are analogous to the archives made in Windows in .zip or .rar format.

```
sudo apt-get update
```

Also, to avoid the unknown dependency issues which occur frequently during the ROS indigo installation, I used following command:

```
sudo apt-get install libgl1-mesa-dev-lts-utopic
```

Finally, I installed all the ROS tools and libraries using:

```
sudo apt-get install ros-indigo-desktop-full
```

c. Tele operate the turtlebot in the Gazebo environment:[vi]

Our team has decided to accomplish the following tasks using Gazebo:
1. Simulating the currently developed single robot system
2. Simulating the SWARM robot system to be developed

I am working on the first task while Tiffany alongwith Jimit are working on the SWARM robot system simulation part. The purpose behind accomplishing the single robot system simulation is the ease in learning Gazebo-turtlebot simulation implementation. It will be easier to learn Gazebo with this task as the programming part has already been done in the Fall semester which can be directly implemented inside Gazebo. To achieve the completion of the first task, the first step was to tele-operate the turtlebot inside Gazebo environment. Firstly, I installed the turtlebot-gazebo software[vii] using following command inside the terminal:

```
$ sudo apt-get install ros-indigo-turtlebot-gazebo
```

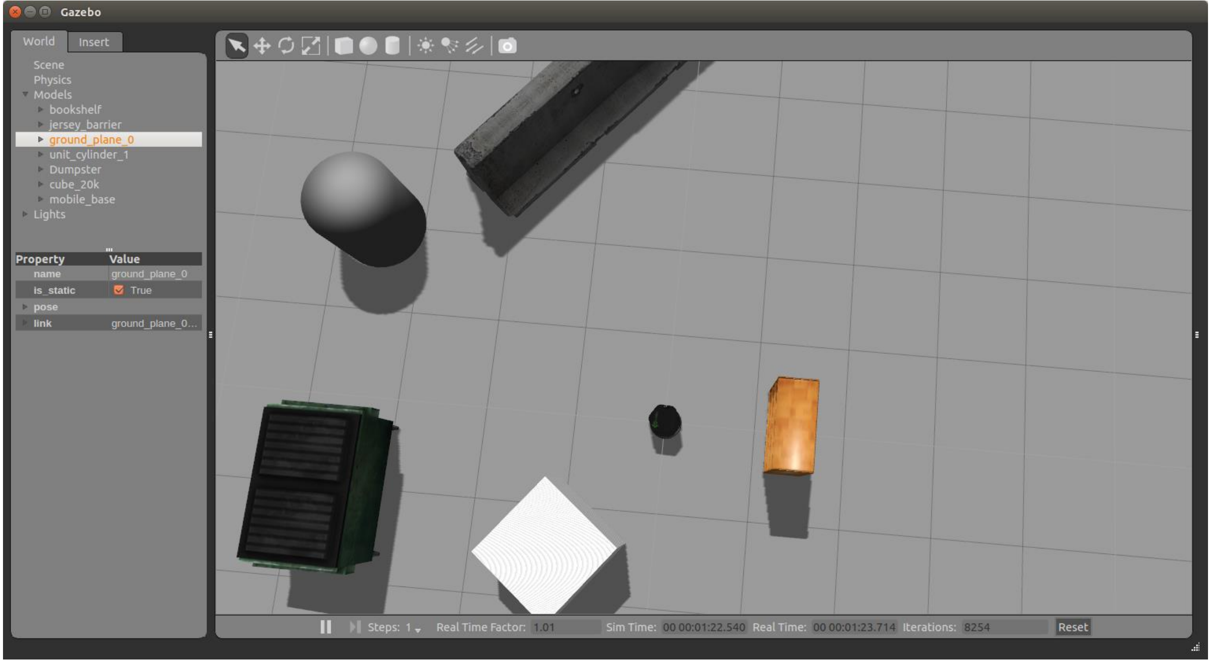
This was followed by sourcing the ROS directory:

```
$ source /opt/ros/indigo/setup.bash
```

Further, I entered the following command inside the terminal to open the Gazebo environment:[viii]

```
$ roslaunch turtlebot_gazebo turtlebot_world.launch
```

This started the turtlebot in the simulated environment as shown in the figure 3:



Property	Value
name	ground_plane_0
is_static	☑ True
pose	
link	ground_plane_0...

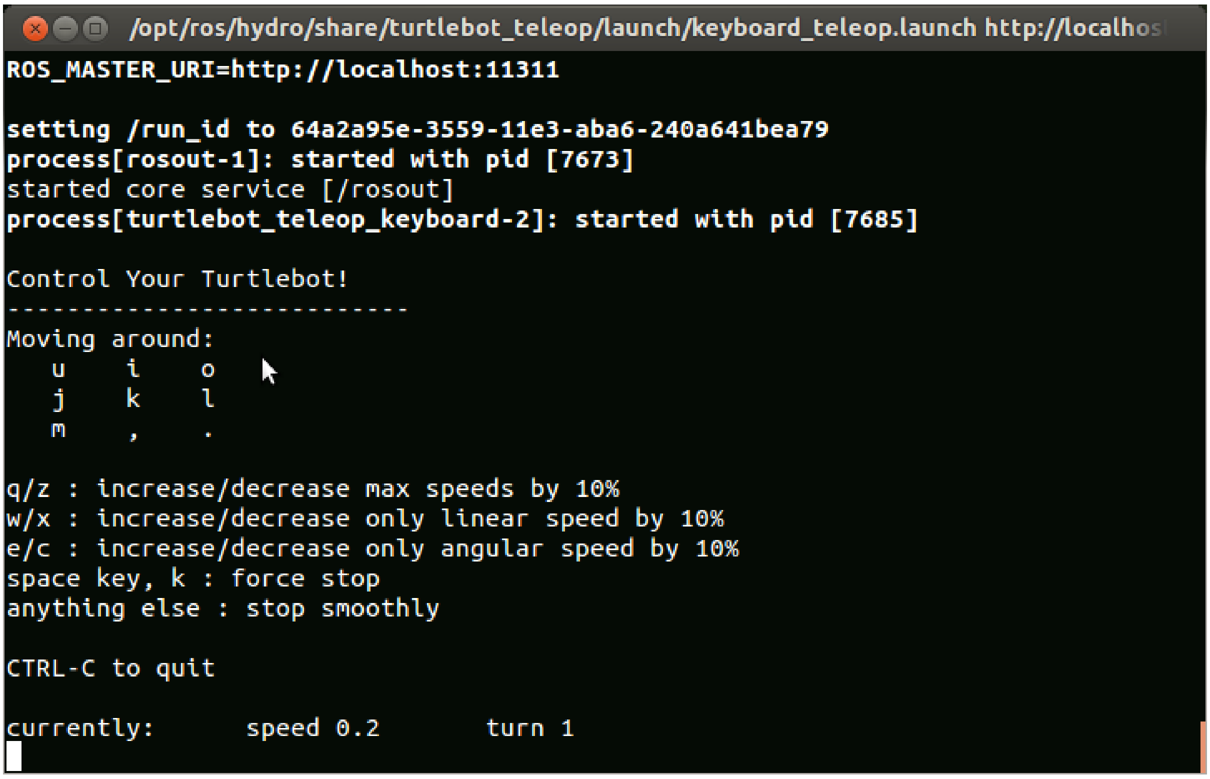
Steps: 1 Real Time Factor: 1.01 Sim Time: 00 00:01:22.540 Real Time: 00 00:01:23.714 Iterations: 8254 Reset

Figure 3: Bring up the turtlebot in Gazebo environment

This was followed by opening another terminal and entering the following command to put the turtlebot in the tele operation mode using the keyboard:

```
$ roslaunch turtlebot_teleop keyboard_teleop.launch
```

After entering this command, the terminal showed the teleop buttons as depicted by the following figure:



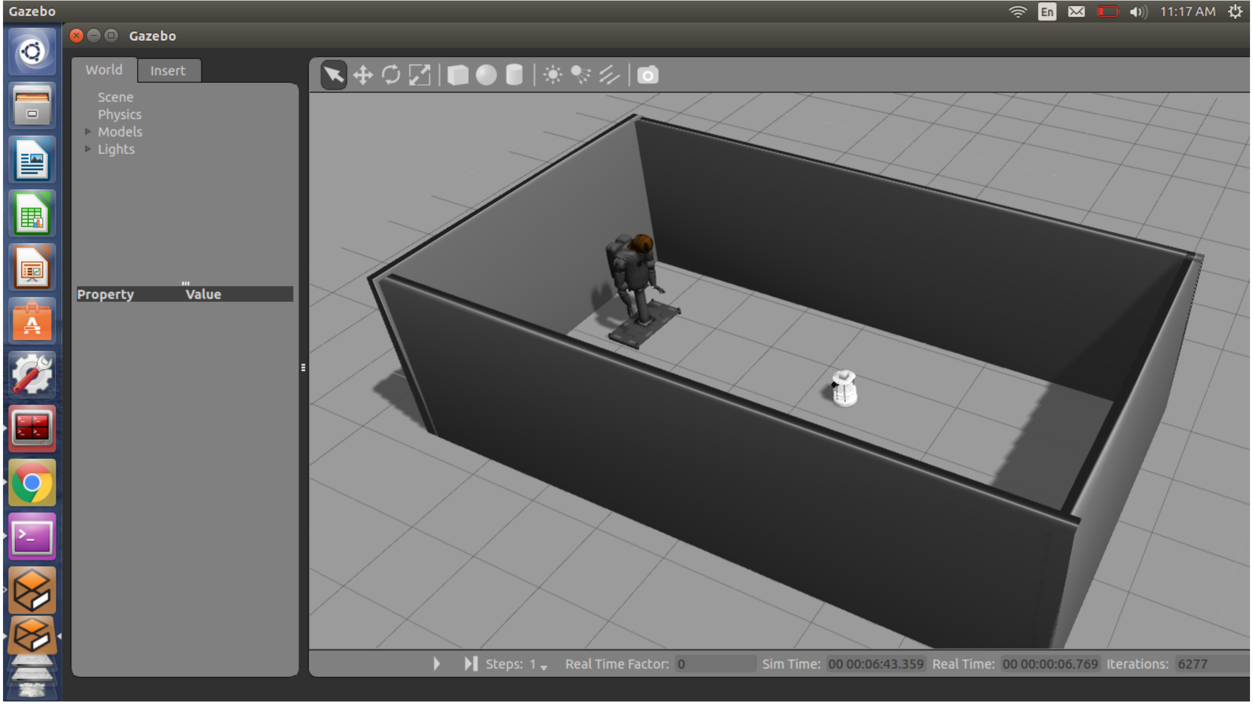
```
/opt/ros/hydro/share/turtlebot_teleop/launch/keyboard_teleop.launch http://localhos
ROS_MASTER_URI=http://localhost:11311

setting /run_id to 64a2a95e-3559-11e3-aba6-240a641bea79
process[rosout-1]: started with pid [7673]
started core service [/rosout]
process[turtlebot_teleop_keyboard-2]: started with pid [7685]

Control Your Turtlebot!
---------------------------
Moving around:
   u    i    o
   j    k    l
   m    ,    .

q/z : increase/decrease max speeds by 10%
w/x : increase/decrease only linear speed by 10%
e/c : increase/decrease only angular speed by 10%
space key, k : force stop
anything else : stop smoothly

CTRL-C to quit

currently:      speed 0.2       turn 1
```

Figure 4: Turtlebot Teleop mode

Using the buttons shown in the above figure, I was able to operate turtlebot and move it in the simulation environment.

d. Creating the world for turtlebot movement:

After controlling the turtlebot using teleop mode, the next step for creating the single robot system simulation was the creation of a turtlebot world similar to the existing room where the FVE was carried out. I followed the ‘Gazebo-building a world’[ix] tutorial and created a simple world with the same dimensions as that of the actual room of FVE performance. This looked like as shown in the figure 5:



Steps: 1 Real Time Factor: 0 Sim Time: 00 00:06:43.359 Real Time: 00 00:00:06.769 Iterations: 6277

Figure 5: Creation of the Gazebo world

2. Challenges

I faced the following challenges during the preparation for the PR#8:

- **Finding out the correct Gazebo version :**
  The ROS indigo package already has a dependent Gazebo 2.x version with it. However, Gazebo7 is the current and updated version which comes as standalone product and has no dependency on ROS. Gazebo7 has better functionalities and is more user friendly than any of the previous versions of Gazebo. Hence, I installed Gazebo7 from its website. However, it did not work with any of the ROS commands. Later with some research on the Gazebo website, I acknowledged that the only Gazebo 2.x series versions are supported by ROS indigo[x]. Hence, I had to install the Gazebo 2.x version again. Due to confusing documentations on the Gazebo website, a lot of time was wasted before starting the actual handling of Gazebo. One good thing was that there was no need to first uninstall the previously installed Gazebo7 as it was directly replaced during the Gazebo 2.x version installation.

- **Absence of the 'Undo-Redo' functionalities in Gazebo:**
  During the creation of world in Gazebo, I noticed that there was no presence of the regular undo (control+z) -redo (control+y) functionalities. Hence, during the world creation, extra care had to be taken. I made mistakes for several times during this activity as any wrongly entered dimensions/models in the world could not be un-done and I had to either delete the particular dimension/model or start designing an altogether new world. This bug is still present in the latest Gazebo7 as well.

3. Teamwork

Since the start of the project, I have taken the responsibility for the completion of the mechanical work subsystem which has been completed now. Hence, I decided to try my hand over the Gazebo for simulating the turtlebot behavior. I, Tiffany and Jimit worked on the Gazebo on different modules. I worked on developing the existing single robot system environment in Gazebo, while Tiffany worked on developing a world model in Gazebo, similar to the real world environment used currently (NSH 1602). Jimit worked on implementing the ROSCON in Gazebo. Gauri and Sida both worked together on setting up the ROCON multi master environment. I must mention that Gauri along with Jimit did very well and lead from the start throughout the project work before PR#8. Gauri also did a great job as a presenter for this PR.

4. Future Plans

Individual future plans before the PR#9 for the Robographer Project:

1. Order remaining quantities of the components
2. Make necessary modifications in the pan tilt elevate unit design
3. Fabricate another pan tilt elevate unit
4. Integrate it with another turtlebot
5. Achieve the AprilTag detection in the Gazebo world

References

[i] http://www.ubuntu.com/download/desktop/create-a-usb-stick-on-windows

[ii] http://www.pendrivelinux.com/universal-usb-installer-easy-as-1-2-3/#button

[iii] http://www.ubuntu.com/download/desktop

[iv]https://help.ubuntu.com/community/WindowsDualBoot

[v] http://wiki.ros.org/indigo/Installation/Ubuntu

[vi] http://wiki.ros.org/turtlebot_gazebo/Tutorials

[vii] http://wiki.ros.org/turtlebot_gazebo

[viii] http://wiki.ros.org/turtlebot_gazebo/Tutorials/indigo/Gazebo%20Bringup%20Guide

[ix] http://gazebosim.org/tutorials?tut=build_world

[x] http://gazebosim.org/tutorials/?tut=ros_wrapper_versions