1 Lab Scope and Aim

The aim of this lab is to introduce the students to the basic functionalities in motion and control of a mobile robot. The robot that you will use in this lab is the turtlebot3 burger and you will be developing your control and motion algorithms within Simulink framework. This lab assignment involves (a) an intro to the simulated robot (b) an intro to basic usage through Simulink and (c) template code that we give you to make the bringup your robot. The main aim of this lab is to experiment with different control strategies so that the turtlebot follows the desired motion patterns.

Prerequisites: The students are suggested to have a good background in the following areas:
- *Automatic Control Systems*
- *Advanced Mathematics*
- *Completed Lab1 Tasks*
- *Simulink*

Lab Supplementary Material: This lab is accompanied with the supplementary material to support the assignments. The material is included in the lab2.zip file.

2 Simulink

The lab implementation is structured around Simulink. Simulink, developed by MathWorks, is a graphical programming environment for modeling, simulating and analyzing dynamical systems. Its primary interface is a graphical block diagramming tool and a customizable set of block libraries. More information regarding simulink can be find in `https://se.mathworks.com/products/simulink/getting-started.html`

3 TurtleBot Burger

TurtleBot3 (Figure 1) is a small, affordable, programmable, mobile robot to use in education, research, hobby, and product prototyping. The TurtleBot3 can be customized into various ways depending on how you reconstruct the mechanical parts and use optional parts such as the computer and sensor. In addition, TurtleBot3 is evolved with cost-effective and small-sized single board computer that is suitable for robust embedded system, 360 degree distance sensor and 3D printing technology. An advantage of this robotic educational platform is that it has a virtual simulated model as well for remote application development and will be the basis for this lab.

3.1 Turtlebot Bringup in Virtual Machine

This section provides instructions on how to connect to simulated Turtlebot from your local machine.

Configure simulated turtlebot in a virtual machine In this part, you will install a Virtual Machine Player, download and configure a new Virtual Machine file that includes the simulated Turtlebot executable files. To use the virtual machine file follow the steps below:

1. Install the VM player by following the instructions found VMWare Installation (**Do not download the VM**, just the VMWare player).

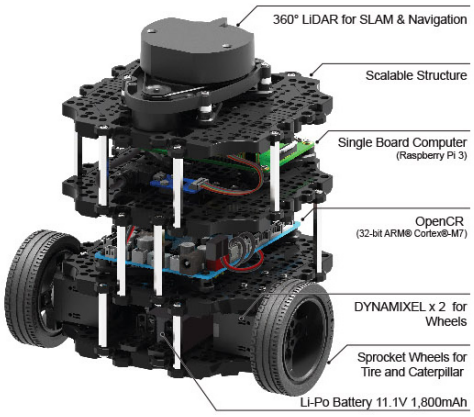

Figure 1: TurtleBot3

2. Download the modified Virtual Machine from Modified Virtual Machine.
3. Run the virtual machine using the Virtual Machine Player (VMWare) you installed previously.
   (a) Decompress the downloaded archive to a location on your hard drive.
   (b) Start VMware Player.
   (c) In VMware Player, press Open a Virtual Machine.
   (d) Browse to the location of the Ubuntu image, select the file and press OK.
   (e) The virtual machine is now added to your library.
   (f) In VMware Player, start the virtual machine.
4. The first time you start the virtual environment you will get a notification whether you moved or copied the file. Select the option "I copied it" as shown in Figure 2.

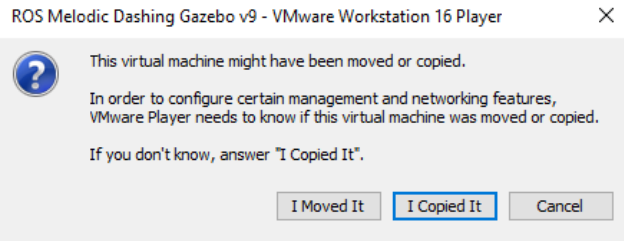

Figure 2: Virtual Machine first time message

Example to test simulated turtlebot Now that you have properly installed and loaded the virtual machine you can run an example program to test that the connection between the robot and Simulink has been established. In the virtual machine desktop select the Gazebo Empty icon as shown in Figure 3, which will initialize a virtual environment as shown in Figure 4.

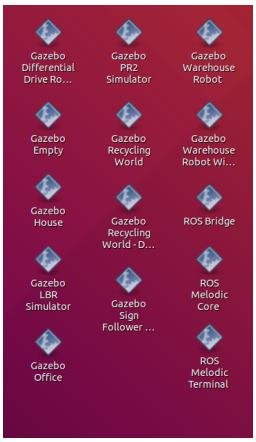

Figure 3: Various world start-up scripts

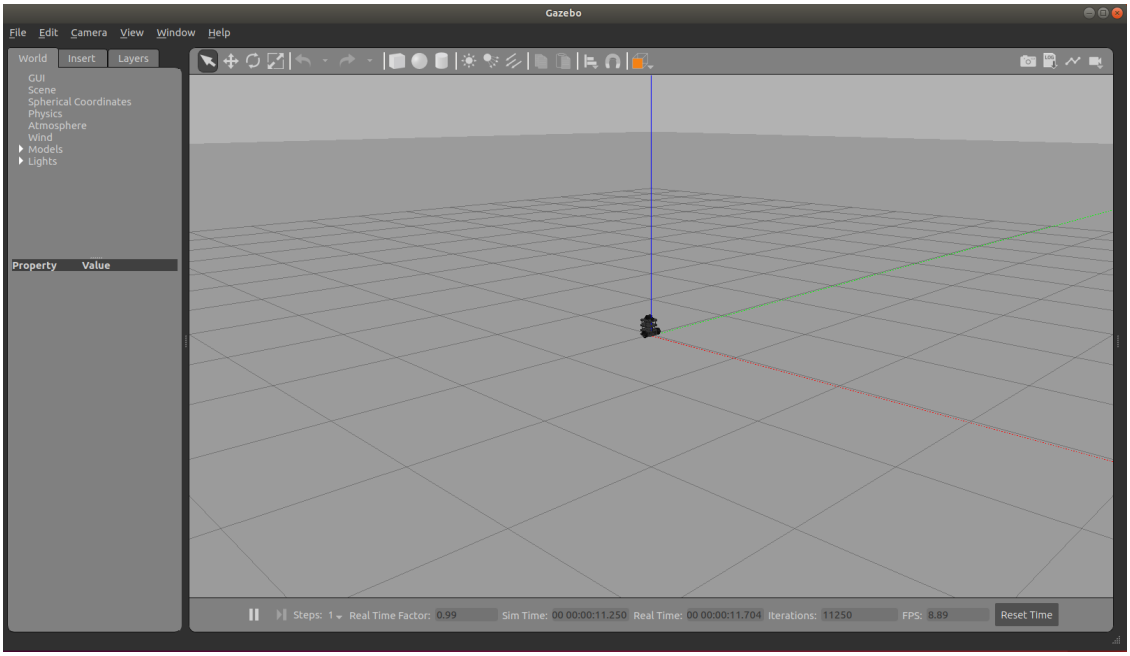

Figure 4: Virtual world initiated from the “Gazebo Empty” script

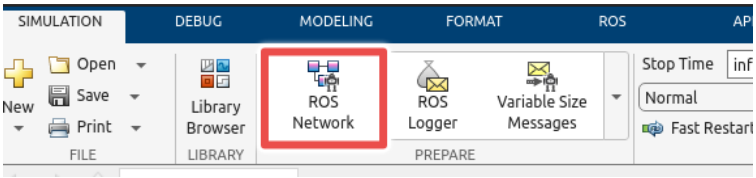

Figure 5: Bridge between turtlebot and Simulink.

Now open MATLAB in your computers and then open the Simulink file with the name "Lab2_Task_1234.slx". Follow Figure 5 steps to open Configure Network Addresses. Now, you must identify the IP depicted in the top right box of the virtual machine Desktop and type it in the window shown in Figure 6. Press test to ensure your connection to the robot. Finally, in the simulink model, assign

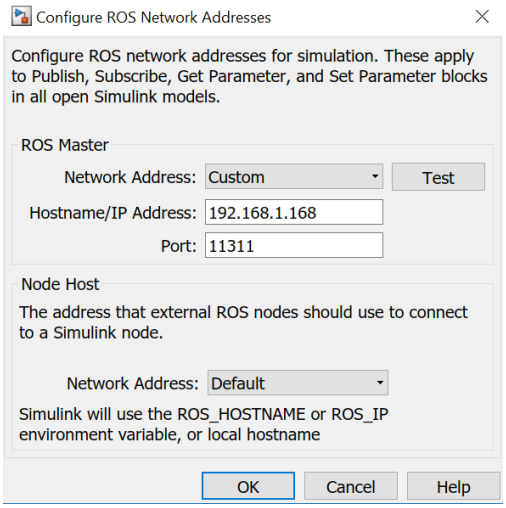

Figure 6: Example of configuring IP.

a constant linear velocity in the 'Proportional Controller' block and observe that the simulated turtlebot in the virtual machine should start moving. Please note that you should also flip the STOP switch (as shown in Figure 7). Now you are ready to move to the Assignments section.

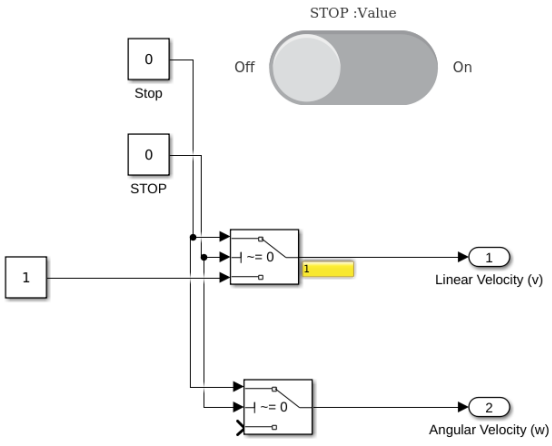

Figure 7: Simulink example testing code.

3.2 Notes

Note1: You should always run your algorithms in your local machine following the instructions provided in section 3
Note2: For Assignment Tasks 1 to 4 use "Lab2_Task_1234.slx"
Note3: For Assignment Task 5 use "Lab2_Task_5678.slx"
Note4: For Assignment Tasks 1 to 4 use "Gazebo Empty" virtual world
Note5: For Assignment Task 5,7 use "Gazebo House" virtual world
Note6: For Assignment Tasks 6 use the "Turtlebot_Corridor"

4 Assignments

Task 1 (Moving to a point) In this task you should develop your controller to move turtlebot from Point A $[x_A, y_A]$ to Point B $[x_B, y_B]$. Repeat the experiment with different points B.

Task 2 (Moving to a pose) In this task you should develop your controller to move turtlebot from initial pose A $[x_A, y_A, \theta_A]$ to pose B $[x_B, y_B, \theta_B]$. Repeat the experiment with different poses B.

Task 3 (Following a trajectory) In this task you should develop your controller to move turtlebot along a trajectory. For this task your turtlebot should be able to follow the trajectories defined below:

- square → square pattern of your choice. Observe your robot repeating this square motion
- letter → the first letter of your last name
- curved path → e.g. sinusoid of your choice

Task 4 (Following a line) In this task you should develop your controller to follow a line on the plane defined by $ax + by + cx = 0$. For this task your turtlebot should be able to follow the lines with the coefficients defined below:

- $a = 1, b = -1, c = 1$
- $a = 1, b = -2, c = 4$

Task 5 (Lidar data plotting) In this task, you should plot the surroundings of your robot using ranges from lidar. Move your turtlebot close and far from objects to evaluate the min-max ranges of the lidar and report the limitations.

Task 6 (Keep distance from surfaces) In this task you should develop your controller for turtlebot to keep a distance from a surface that can be either straight wall or rotated surface. In this case you should use measurements from the lidar sensor which is placed on top of the robot. The schematic of the task is shown in Figure 8. The desired behavior of Turtlebot is shown in Figure 9. In this task you can select how many and which beams you want to use from the lidar. Try your algorithms with different beam configurations and select the one that performs better, providing an overview in your report.

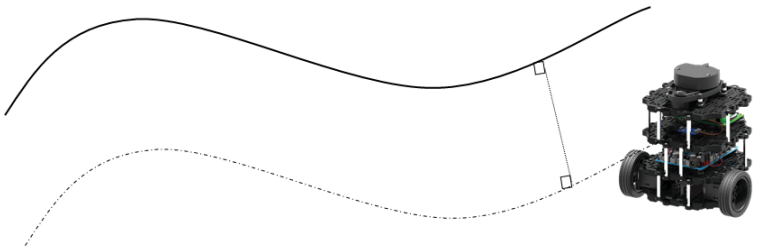

Figure 8: Example of path with constant distant to the uneven wall.

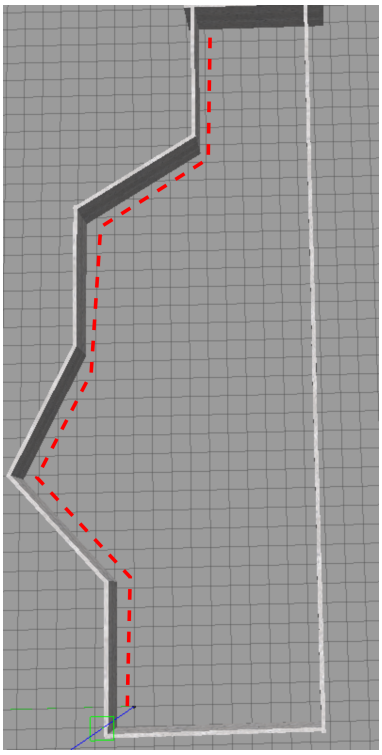

Figure 9: Desired Turtlebot path in the distance to wall task

Task 7 (Lidar data registration) In this task, you should plot a map of your robot surroundings (Figure 10) using ranges from lidar, while following a straight path. More specifically you should provide a plot that contains all lidar measurements registered during the motion of turtlebot. Hint: Convert ranges to x, y points and associate them with the robot pose in the local coordinate frame.

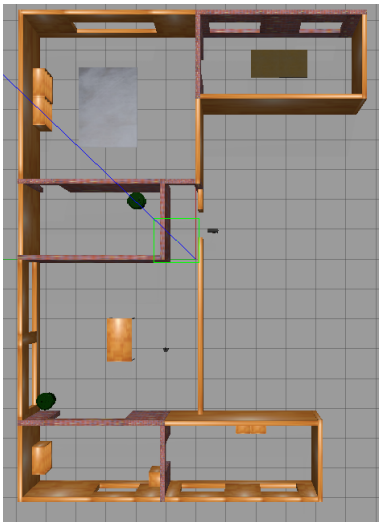

Figure 10: Simulated house environment for Turtlebot