Robot Laboratory

Department of Mechanical and Aerospace Engineering
Rutgers, The State University of New Jersey
Juan López Muro

Spring, 2024

1 Introduction

In this laboratory session, we aim to provide you with hands-on experience that complements the theoretical concepts covered in the classroom. Our focus will be on understanding, modeling, and controlling the dynamics a differential-drive robot shown in Figure 1 [1].

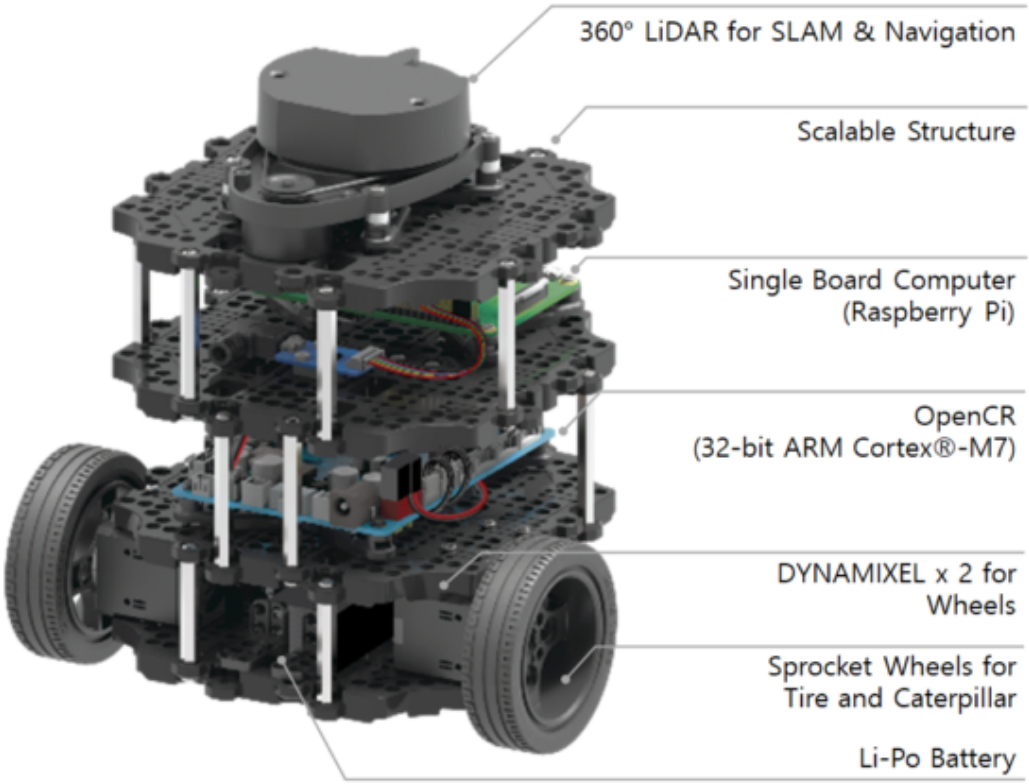

Figure 1: TurtleBot3 Burger: Components

2 Opening PowerShell in Windows

To open PowerShell in Windows, follow these steps:

1. Press `Win + S` to open the search bar.
2. Type `PowerShell` and press `enter`.

We will open five PowerShells during this lab.

3 Basic PowerShell Commands

3.1 Get Help

To get help about a specific command, use the `Get-Help` command. For example:

```
Get-Help Get-ChildItem
```

This will display information about the `Get-ChildItem` command.

3.2 List Files and Directories

To list files and directories in the current location, use the `Get-ChildItem` command or any of the aliases `gci`, `ls`, `dir`:

```
Get-ChildItem
```

3.3 Change Directory

To change the current directory, use the `Set-Location` command:

```
Set-Location C:\Path\To\Your\Directory
```

4 Secure Shell Command

The `ssh` command is a widely used command-line tool for securely connecting to remote servers or devices over a network. The acronym "ssh" stands for Secure Shell. It provides a secure encrypted communication protocol for secure access and management of network devices. The basic syntax for using the ssh command is as follows:

`ssh` **`[user]@[IP address]`**

where,

`[user]`: The username to log in with on the remote machine.

`[IP address]`: The address or IP of the remote machine.

The remote device is a small single-board computer on the robot, a "Rasberry Pi", running a Linux operating system. The user is the name of the Linux distribution, `ubuntu`. The IP address is a string of numbers separated by periods `192.168.1.X`. The last number, the last number `X`, will be variable and given in class.

On the first PowerShell, type the correct command to `ssh` into the robot is as follows:

```
ssh ubuntu@192.168.1.X
```

and press `enter`.

Automatically after, a password will be asked. The password is: `turtlebot`. Otherwise will also be provided in class.

5 ROS

ROS is an open-source, meta-operating system for the robot. It provides many services used in this lab, including hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management. It also provides tools and libraries for obtaining, building, writing, and running code across multiple computers. For more information, please read [2, 3].

5.1 `roslaunch`

`roslaunch` is a tool for easily launching multiple ROS nodes (a node is a process that performs computation) locally and remotely via `ssh`.

5.1.1 Bringup TurtleBot3

We will use `roslaunch` to bring up basic packages to start TurtleBot3 applications. On the second PowerShell, type the correct command to `ssh` into the robot. On the PowerShell, type `roslaunch` basic packages to start TurtleBot3 applications as follows:

```
roslaunch turtlebot3_bringup turtlebot3_robot.launch
```

and press `enter`.

5.1.2 Teleoperate TurtleBot3

We will use `roslaunch` for the teleoperation of the TurtleBot3 using a keyboard. On the second PowerShell, bring up basic packages for the teleoperation of the TurtleBot3 using a keyboard as follows:

```
roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch
```

and press `enter`.

Move around!
`ctrl-C` to quit.

5.1.3 Get TurtleBot3 pose data from VICON

Vicon is a solution for UAV and robotic systems that provides low-latency data that is easy to integrate into the control system. The Vicon solution provides the linear and angular position of the robot and is fully compatible with ROS.

Getting pose data on Onboard Processor Using ROS, get data from the VICON Machine

We will use `roslaunch` for getting pose data of the TurtleBot3 from VICON.

On the third PowerShell type the correct command to `ssh` into the robot. On the PowerShell, bring up basic packages for getting pose of the TurtleBot3 using VICON as follows:

```
roslaunch vrpn_client_ros sample.launch server:=192.168.1.10
```

and press `enter`.

5.2 `rostopic`

5.2.1 Display TurtleBot3 pose data from VICON

Topics are named buses over which nodes exchange messages. The `rostopic` command-line tool displays information about ROS topics. `rostopic`, like several other ROS tools, uses YAML-syntax at the command line to represent a message's contents. In particular, `rostopic echo` displays messages published to a topic on the screen. Use rostopic to display the pose of the turtlebot3.

On the fourth PowerShell type the correct command to `ssh` into the robot. On the PowerShell, display the pose of the turtlebot3 as follows (replace X by the number of the turtlebot):

```
rostopic echo /vrpn_client_node/turtlebot_X/pose
```

and press `enter`.
`ctrl-C` to quit.

6 Cyberduck

Cyberduck [4] is an open-source file transfer client for FTP (File Transfer Protocol) and SFTP (SSH Secure File Transfer). Cyberduck supports both FTP and SFTP protocols, allowing users to connect to and transfer files securely to remote servers.

The application has a user-friendly interface with drag-and-drop functionality, making it easy for users to upload, download, and manage files.
To open Cyberduck in Windows, follow these steps:

1. Press `Win + S` to open the search bar.
2. Type `Cyberduck` and press `enter`.

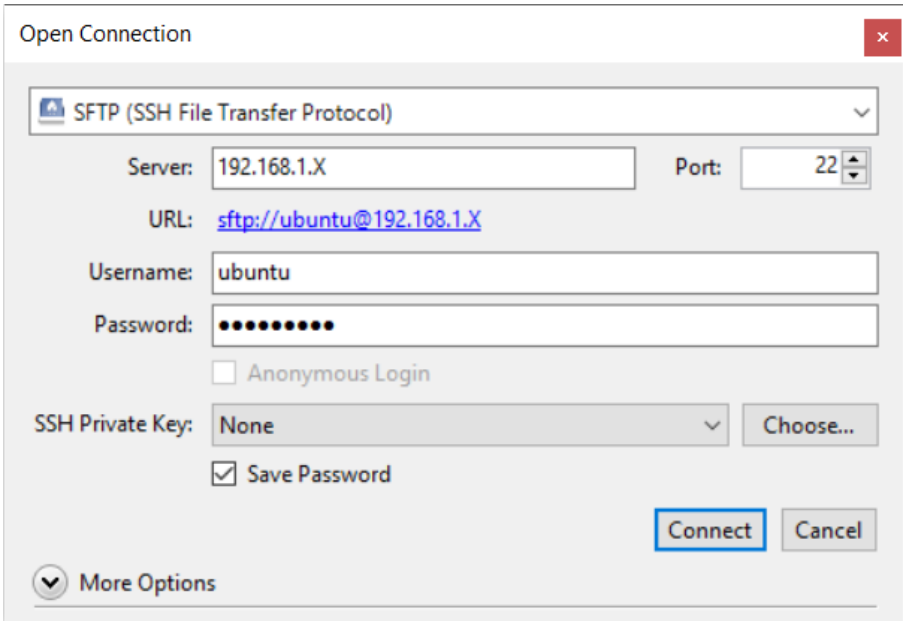

Figure 2: Cyberduck: open connection.

Open a new connection, select the SFTP (SSH File Transfer Protocol)
where,

[Protocol]: `SFTP (SSH File Transfer Protocol).`

[Server]: The address or IP of the remote machine: `192.168.1.X`, The last number [X], will be variable and given in class.

[Username]: The username to log in with on the remote machine: `ubuntu`

[Password]: The password is: `turtlebot`. Otherwise will also be provided in class.

7 Robot Laboratory

We will follow the same steps we did in the simulation.

TASK 0: Edit the mainStd.py file with Notepad++ [5]

0.1 The controller and the main section are the only methods that need to be modified. Both are at the end of this script.

TASK 1: Record a Manual Trajectory

1.1 Go to the main section, Task 1

1.2 Uncomment the `obj.record_traj("SecXX_LastName_Name.txt")` line, and comment out everything else.

1.3 Modify the string `SecXX_LastName_Name.txt`, using your section and name.

1.4 Save the File using `File` → `Save` or `ctrl-S`.

1.5 Place the robot in a known spot, position, and orientation.

1.6 Prepare to teleoperate the TurtleBot3 using the keyboard. To do that, on a PowerShell run:

```
roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch
```

1.7 Prepare to record the teleoperated trajectory. To do that, execute this Python script, on a PowerShell run:

```
python mainStd.py #use python3 depending on the version.
```

1.8 Select the former PowerShell and teleoperate the TurtleBot3 using the keyboard. When satisfied with the trajectory, select the later PowerShell and click ctrl-c to quit the recording.

1.9 Repeat 1.5 and 1.6 until satisfied.

1.10 Press `ctrl-C` on the former PowerShell to quit the teleoperation.

Task 2: Play Back the Recorded Manual Trajectory in an Open Loop

2.1 Go to the main section, Task 2

2.2 Uncomment the `obj.play_traj("SecXX_LastName_Name.txt")` line, and comment out everything else.

2.3 Modify the string `SecXX_LastName_Name.txt`, using your section and name.

2.4 Save the File using `File` $\rightarrow$ `Save` or `ctrl-S`.

2.5 Place the robot in a known spot, position, and orientation.

2.6 Be sure that the keyboard teleoperation process is killed. It can interfere.

2.7 Prepare to play back the teleoperated trajectory. To do that, execute this Python script, on a PowerShell run: `python mainStd.py #use python3 depending on the version.`

```
python mainStd.py #use python3 depending on the version.
```

2.8 Press `ctrl-C` to finish this task.

Task 3: Design and Code the Feedback Controller

3.1 Go to the main section, Task 3

3.2 Uncomment the `obj.control_traj("rec_SecXX_LastName_Name.txt", "log_ctrl_SecXX_LastName_Name.txt")` line, and comment everything else.

3.3 We will follow the same steps we did in the simulation.

3.4 Place the robot in an $x = 0$, $\psi = 0$ pose.

3.5 Implement a simple x-control using first-order error dynamics. Go to the controller function and follow all the X.0, X.1, X.2, X.3, X.4, and X.5. You only need to write one line of code at each step.

3.6 Save the File `File` $\rightarrow$ `Save` or `ctrl-S`.

3.7 Prepare to test your x-position proportional controller. To do that, execute this Python script, on a PowerShell run:

```
python mainStd.py #use python3 depending on the version.
```

3.8 Press `ctrl-C` to finish this task.

3.9 Repeat 3.4 to 3.7 until satisfied.

3.10 Place the robot in an $x = 0$, $\psi = 0$ pose.

3.11 Implement a simple ψ-control using first-order error dynamics. Go to the controller function and follow all the P.0, P.1, P.2, P.3, P.4, and P.5. You only need to write one line of code at each step.

3.12 Save the File `File → Save` or `ctrl-S`.

3.13 Prepare to test your ψ-position proportional controller. To do that, execute this Python script, on a PowerShell run:

```
python mainStd.py #use python3 depending on the version.
```

3.14 Press `ctrl-C` to finish this task.

3.15 Place the robot in an $x = 0$, $y = 0$, $\psi = 0$ pose.

3.16 Implement a simple y-control using first-order error dynamics. Go to the controller function and follow all the Y.0, Y.1, Y.2, Y.3, Y.4, and Y.5. You only need to write one line of code at each step.

3.17 Save the File `File → Save` or `ctrl-S`.

3.18 Prepare to test your y-position proportional controller. To do that, execute this Python script, on a PowerShell run:

```
python mainStd.py #use python3 depending on the version.
```

3.19 Press `ctrl-C` to finish this task.

3.20 Implement the control signal saturation algorithm. Go to the controller function and follow the S.0 and S.1. You only need to write one line of code at each step.

3.21 Implement the "u" turn avoidance algorithm. Go to the controller function and follow U.0.

3.22 Save the File `File → Save` or `ctrl-S`.

3.23 Prepare to test your controller. To do that, execute this Python script, on a PowerShell run:

```
python mainStd.py #use python3 depending on the version.
```

3.24 Press `ctrl-C` to finish this task.

Task 4: Follow the Recorded Manual Trajectory in Closed-Loop

4.1 Go to the controller function and uncomment the `x_d = self.x_traj` and `y_d = self.y_traj` lines.

4.2 Go to the main section, Task 4

4.3 Uncomment the `obj.control_traj("rec_SecXX_LastName_Name.txt", "log_ctrl_SecXX_LastName_Name.txt")` line, and comment everything else.

4.4 Save the File using `File → Save` or `ctrl-S`.

4.5 Prepare to test your controller. To do that, execute this Python script, on a PowerShell run:

```
python mainStd.py #use python3 depending on the version.
```

4.6 Press `ctrl-C` to finish this task.

Task 5: Follow Your Desired Trajectory in a Closed Loop

5.1 Go to the main section, TASK5

5.2 Uncomment the `obj.control_traj("NameOfYourFile.txt", "log_NameOfYourFile.txt")` line, and comment everything else.

5.3 Replace `NameOfYourFile.txt` with the name of your File and `log_NameOfYourFile.txt` with the name you want.

5.4 Save the File using `File` → `Save` or `ctrl-S`.

5.5 Prepare to test your controller. To do that, execute this Python script, on a PowerShell run:

```
python mainStd.py #use python3 depending on the version.
```

5.6 Press `ctrl-C` to finish this task.

Task 6: RACE - Follow Proposed Trajectory in Closed Loop

6.1 Go to the main section, TASK6

6.2 Uncomment the `obj.control_traj("trajectoryRU_SOL.txt", "log_ctrl_SecXX_LastName_Name_RU.txt")` line, and comment everything else.

6.3 Replace `log_ctrl_SecXX_LastName_Name_RU.txt` using your section and name.

6.4 Save the File using `File` → `Save` or `ctrl-S`.

6.5 Prepare to test your controller and race. To do that, execute this Python script, on a PowerShell run:

```
python mainStd.py #use python3 depending on the version.
```

6.6 Press ctrl-c to finish this task.

Task 7: Post-Process and Prepare Presentation

7.1 Find all the log files on CANVAS. Post-process and prepare your presentation.

7.2 Follow the presentation general guidelines.

References

[1] robotis.com, "https://emanual.robotis.com/docs/en/platform/turtlebot3/features/," 2024.

[2] www.ros.org, "https://www.ros.org," 2024.

[3] wiki.ros.org, "http://wiki.ros.org/," 2024.

[4] cyberduck.io, "https://cyberduck.io/," 2024.

[5] notepad-plus plus.org, "https://notepad-plus-plus.org/," 2024.