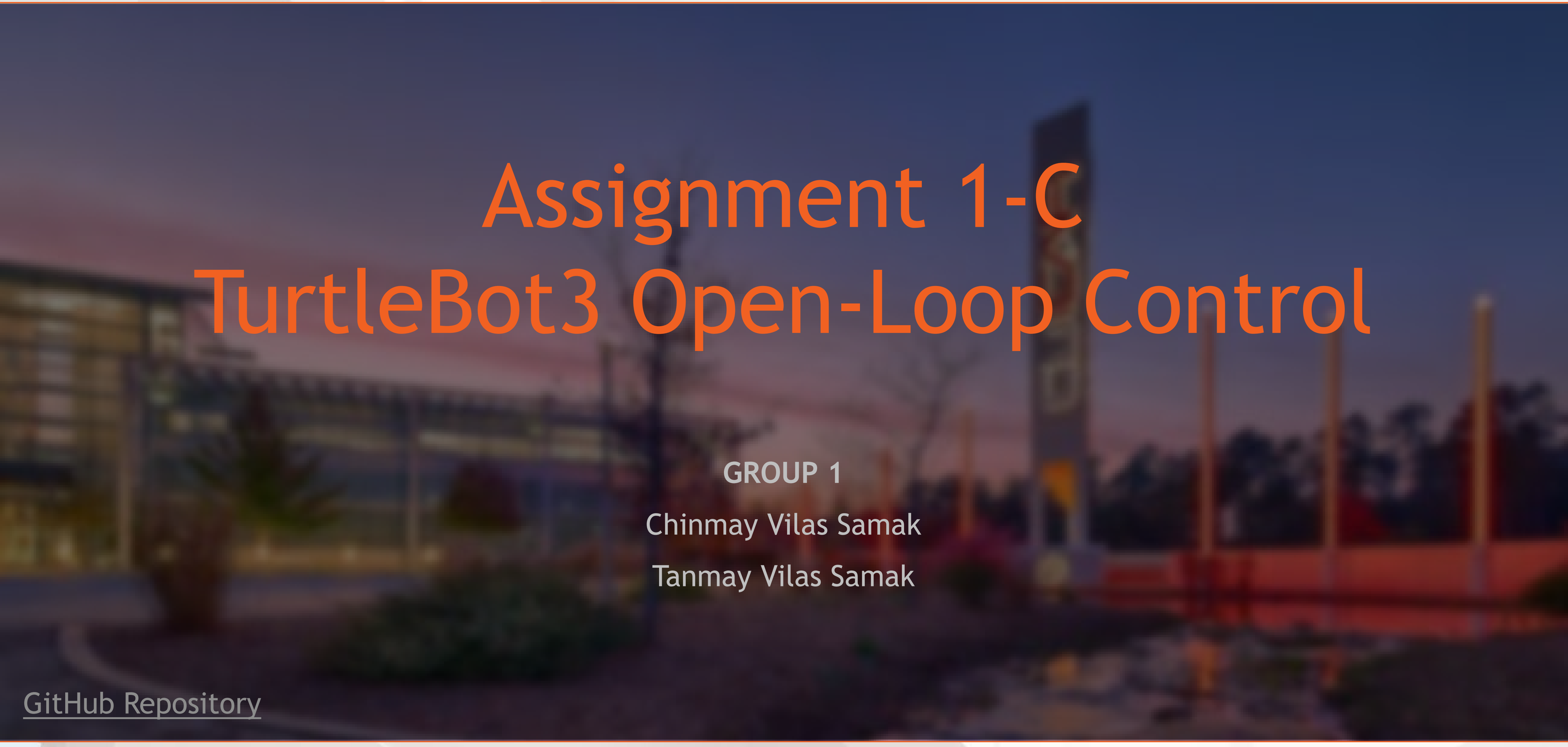


Assignment 1-C
TurtleBot3 Open-Loop Control

GROUP 1

Chinmay Vilas Samak

Tanmay Vilas Samak

GitHub Repository

Overview

- **Circle**
  - Move robot in circle with specified constant twist command
  - Radius of circle determined by ratio of linear and angular velocities (kinematic formulation)
  - Wait (delay) 4 seconds for simulation to initialize properly
  - Simulate for robot’s one complete traversal around the circle (kinematic formulation)
- **Square**
  - Move robot in square with specified constant twist command
  - Side length of square specified as a constant parameter (2.0 m) in script
  - Wait (delay) 4 seconds for simulation to initialize properly
  - Simulate for robot’s one complete traversal around the square (kinematic formulation)
- **Speed variation hypothesis**: as speed increases, trajectory profile will deviate further from ideal

Implementation

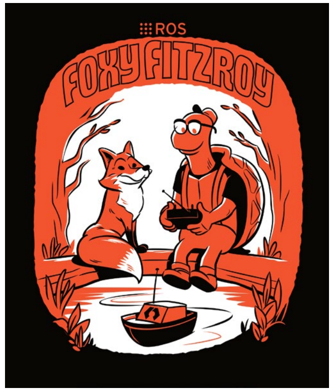
- **Circle** | argument defaults: `lin_vel = 0.1, ang_vel = 0.1`
  - Launch Gazebo with `empty_world` from `turtlebot3_gazebo` package
  - Launch `circle_node` from `assignment_1c` package
  - Launch `rviz2` from `rviz2` package and load configuration file from `assignment_1c` package
- **Square** | argument defaults: `lin_vel = 0.1, ang_vel = 0.1`
  - Launch Gazebo with `empty_world` from `turtlebot3_gazebo` package
  - Launch `square_node` from `assignment_1c` package
  - Launch `rviz2` from `rviz2` package and load configuration file from `assignment_1c` package
- **Move** | argument defaults: `maneuver = circle, lin_vel = 0.1, ang_vel = 0.1`
  - Launch Gazebo with `empty_world` from `turtlebot3_gazebo` package
  - Launch `circle_node` or `square_node` from `assignment_1c` package depending on `maneuver` argument
  - Launch `rviz2` from `rviz2` package and load configuration file from `assignment_1c` package

Dependencies

- ROS2 Foxy Fitzroy on Ubuntu 20.04 Focal Fossa
- TurtleBot3 Packages - Included with the assignment repository
- TurtleBot3 Simulations Packages - Included with the assignment repository
- TurtleBot3 Messages Package - Included with the assignment repository
- TurtleBot3 Dynamixel SDK Packages - Included with the assignment repository

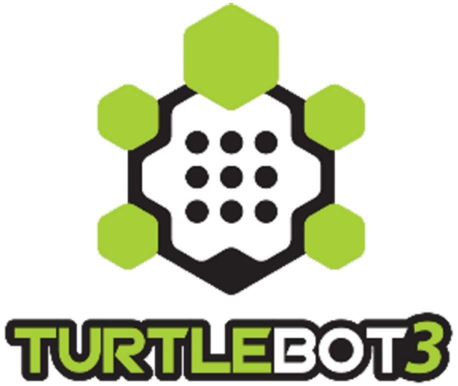
Source: Ubuntu

Source: ROS2 GitHub



Source: ROS2 Docs

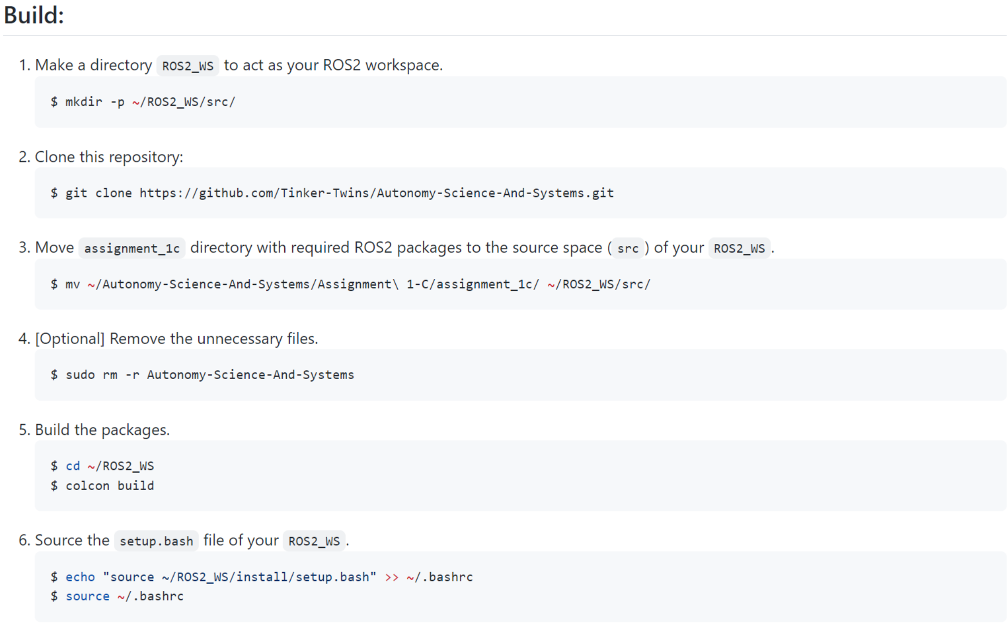
Source: GazeboSim



Source: ROBOTIS Inc.

Build Instructions

- Make a directory **`ROS2_WS`** to act as your ROS2 workspace.
- Clone the assignment repository.
- Move **`assignment_1c`** directory to **`src`** of your **`ROS2_WS`**.
- [Optional] Remove the unnecessary files.
- Build the packages.
- Source the **`setup.bash`** file of your **`ROS2_WS`**.
- **NOTE:** Detailed instructions are available in README.md file of the assignment repository.



Build:

1. Make a directory `ROS2_WS` to act as your ROS2 workspace.

```
$ mkdir -p ~/ROS2_WS/src/
```

2. Clone this repository:

```
$ git clone https://github.com/Tinker-Twins/Autonomy-Science-And-Systems.git
```

3. Move `assignment_1c` directory with required ROS2 packages to the source space (`src`) of your `ROS2_WS` .

```
$ mv ~/Autonomy-Science-And-Systems/Assignment\ 1-C/assignment_1c/ ~/ROS2_WS/src/
```

4. [Optional] Remove the unnecessary files.

```
$ sudo rm -r Autonomy-Science-And-Systems
```

5. Build the packages.

```
$ cd ~/ROS2_WS
$ colcon build
```

6. Source the `setup.bash` file of your `ROS2_WS` .

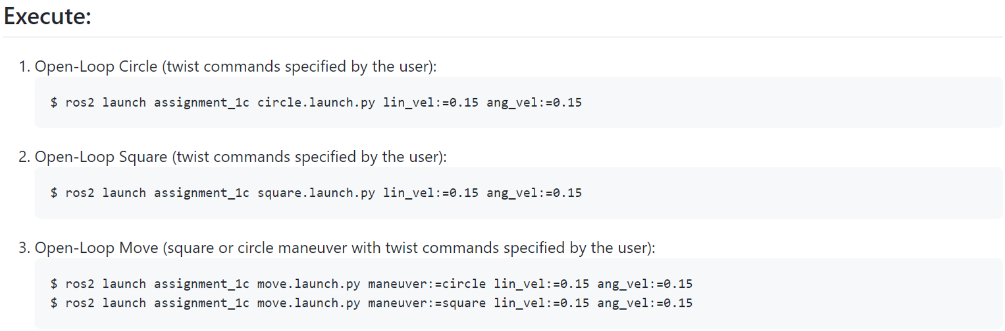
```
$ echo "source ~/ROS2_WS/install/setup.bash" >> ~/.bashrc
$ source ~/.bashrc
```

Source: Tinker Twins GitHub

Execution Instructions

- **Circle**
  - Launch `circle.launch.py` file from `assignment_1c` package
  - Provide arguments for `lin_vel` and `ang_vel` (float)
- **Square**
  - Launch `square.launch.py` file from `assignment_1c` package
  - Provide arguments for `lin_vel` and `ang_vel` (float)
- **Move**
  - Launch `move.launch.py` file from `assignment_1c` package
  - Provide argument for `maneuver` (string)
  - Provide arguments for `lin_vel` and `ang_vel` (float)
- **NOTE:** Detailed instructions are available

  in README.md file of the assignment repository.



Execute:

1. Open-Loop Circle (twist commands specified by the user):

```
$ ros2 launch assignment_1c circle.launch.py lin_vel:=0.15 ang_vel:=0.15
```

2. Open-Loop Square (twist commands specified by the user):

```
$ ros2 launch assignment_1c square.launch.py lin_vel:=0.15 ang_vel:=0.15
```

3. Open-Loop Move (square or circle maneuver with twist commands specified by the user):

```
$ ros2 launch assignment_1c move.launch.py maneuver:=circle lin_vel:=0.15 ang_vel:=0.15
$ ros2 launch assignment_1c move.launch.py maneuver:=square lin_vel:=0.15 ang_vel:=0.15
```

Source: Tinker Twins GitHub

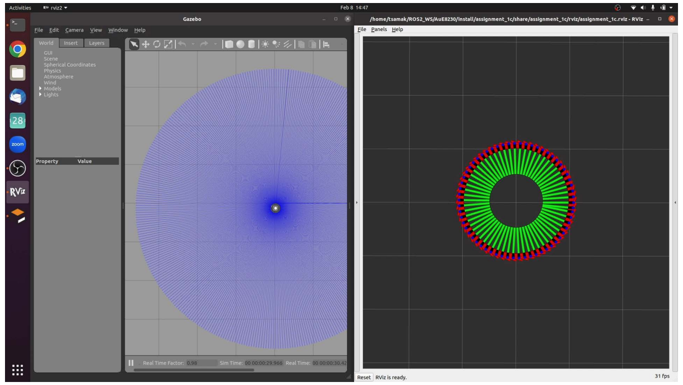
Results

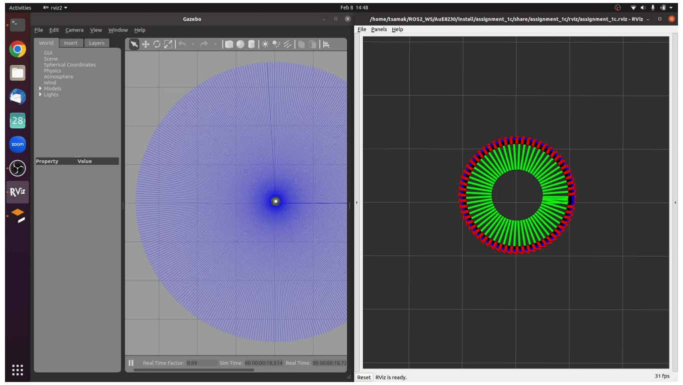
Slow

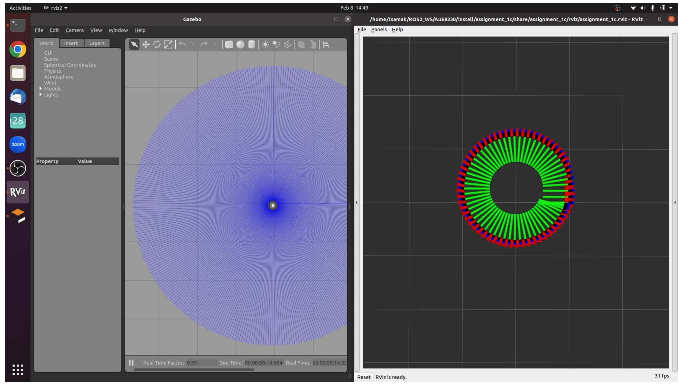
Medium

Fast

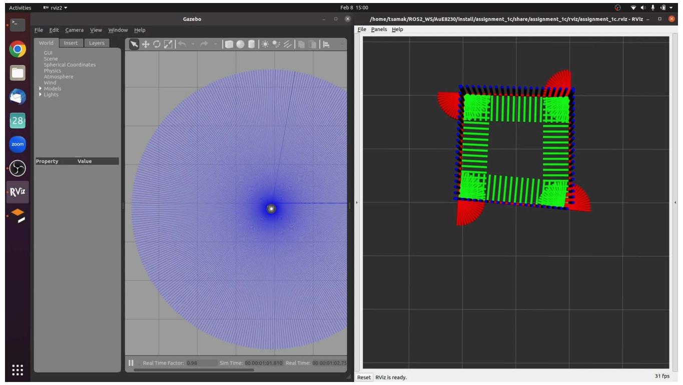
Circle

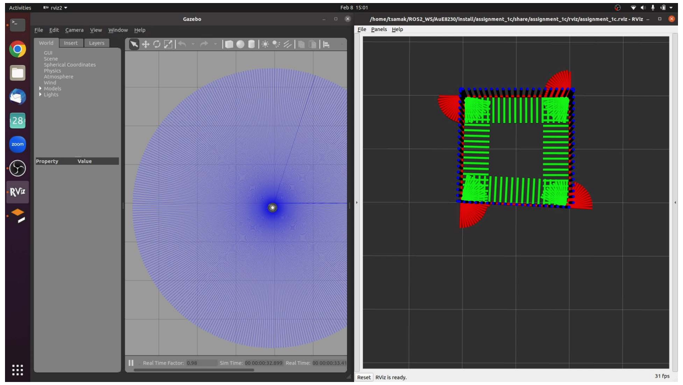

Activities
rviz2
Feb 8 14:47
Gazebo
/home/tsamak/ROS2_WS/AuE8230/install/assignment_1c/share/assignment_1c/rviz/assignment_1c.rviz - RViz
RViz is ready.

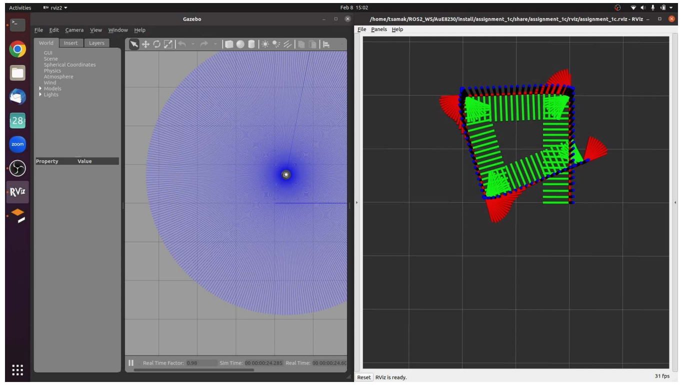

Feb 8 14:48
Gazebo
/home/tsamak/ROS2_WS/AuE8230/install/assignment_1c/share/assignment_1c/rviz/assignment_1c.rviz - RViz


Feb 8 14:49
Gazebo
/home/tsamak/ROS2_WS/AuE8230/install/assignment_1c/share/assignment_1c/rviz/assignment_1c.rviz - RViz

Square


Feb 8 15:00
Gazebo
/home/tsamak/ROS2_WS/AuE8230/install/assignment_1c/share/assignment_1c/rviz/assignment_1c.rviz - RViz


Feb 8 15:01
Gazebo
/home/tsamak/ROS2_WS/AuE8230/install/assignment_1c/share/assignment_1c/rviz/assignment_1c.rviz - RViz


Feb 8 15:02
Gazebo
/home/tsamak/ROS2_WS/AuE8230/install/assignment_1c/share/assignment_1c/rviz/assignment_1c.rviz - RViz

$v = 0.25, \omega = 0.25$

$v = 0.50, \omega = 0.50$

$v = 0.75, \omega = 0.75$

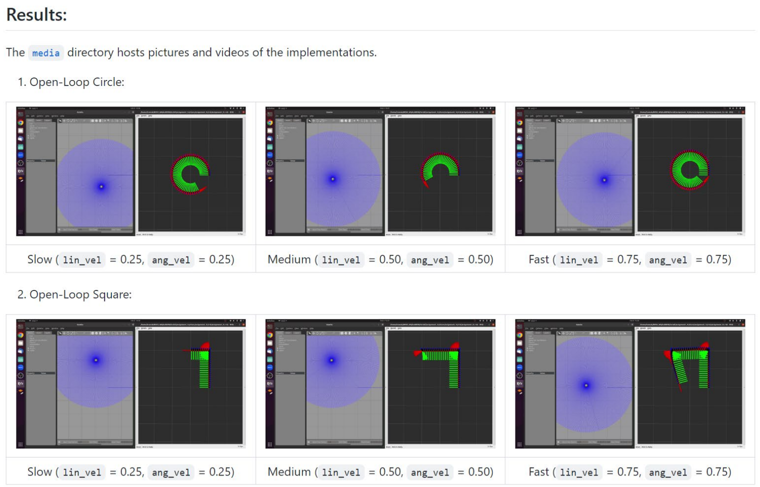
Analysis

Parameter/Metric	Slow Circle	Slow Square	Medium Circle	Medium Square	Fast Circle	Fast Square
Resemblance to ideal trajectory	Very High	Low	High	Medium	Medium	Extremely Low
Kinematic behavior	Very High	High	High	Medium	Medium	Very Low
Time for 1 complete traversal	High	Very High	Medium	Medium	Very Low	Low

Note 1: The analysis presented above is qualitative.
Although a detailed quantitative analysis can be accomplished, it was beyond the scope of this assignment.

Note 2: The physical TurtleBot3 has a maximum travel speed of 0.22 m/s and maximum rotational speed of 2.84 rad/s.
The simulated TurtleBot3 was operated at higher limits to analyze the effect of simulation dynamics on the performance of the open-loop controllers designed.

Note 3: Deployment videos as well as pictures are available in README.md file as well as in the media directory of the assignment repository.



Results:

The `media` directory hosts pictures and videos of the implementations.

1. Open-Loop Circle:

Slow (`lin_vel` = 0.25, `ang_vel` = 0.25) | Medium (`lin_vel` = 0.50, `ang_vel` = 0.50) | Fast (`lin_vel` = 0.75, `ang_vel` = 0.75)

2. Open-Loop Square:

Slow (`lin_vel` = 0.25, `ang_vel` = 0.25) | Medium (`lin_vel` = 0.50, `ang_vel` = 0.50) | Fast (`lin_vel` = 0.75, `ang_vel` = 0.75)

Source: Tinker Twins GitHub

Thank You!