AuE 823: Sp'23: Autonomy Science and Systems

Department of Automotive Engineering

Dhruv Mehta, Sumedh Sathe and Venkat Krovi

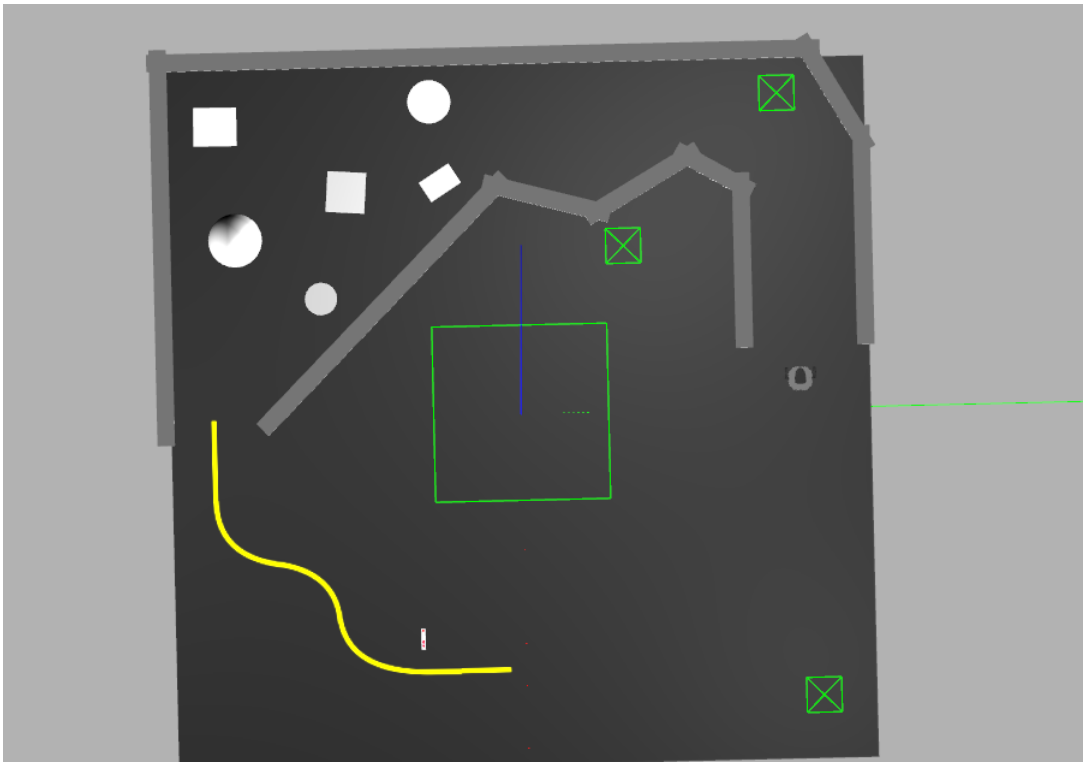
Final Project Instructions

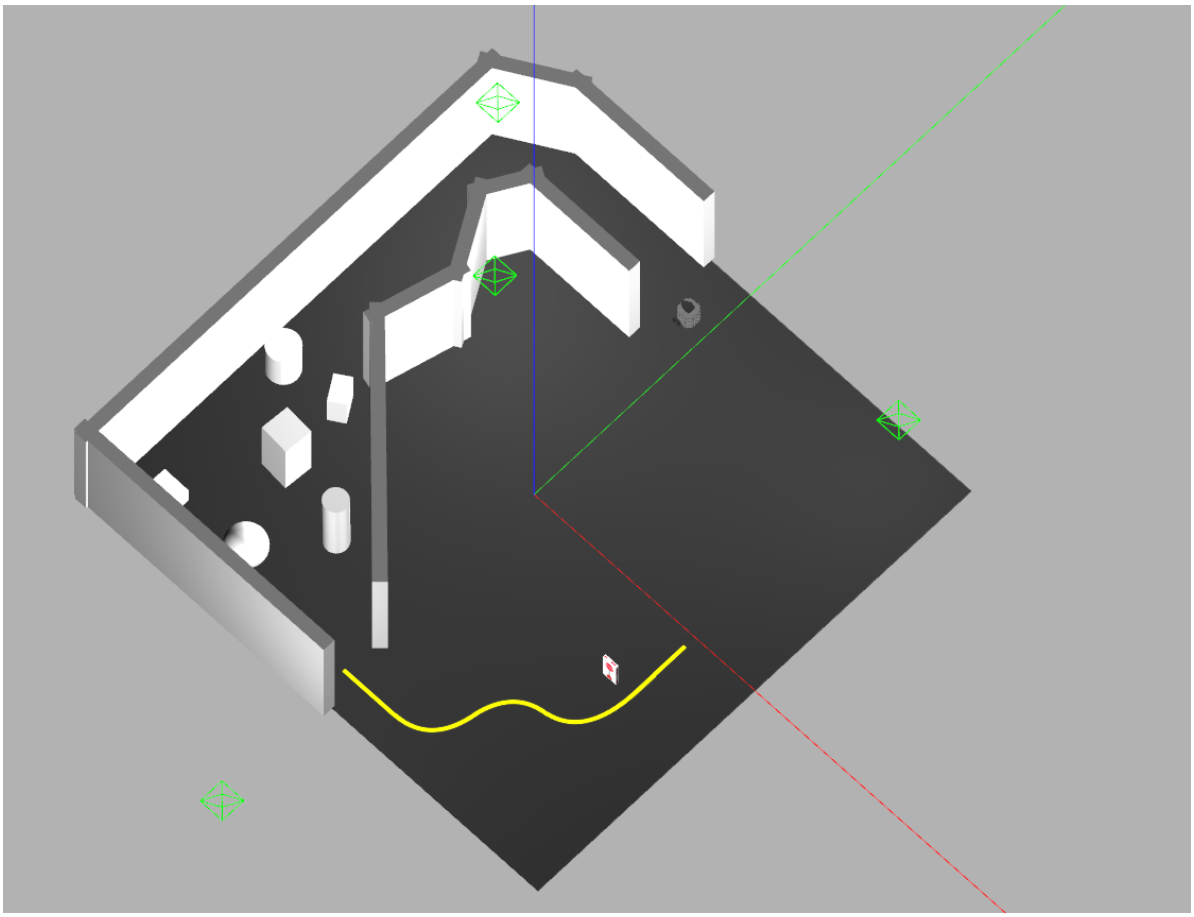
Due 25th April 2023

Please read this carefully and remember to ask questions in the TA hours. The Capstone Project has a simulation as well as sim2Real aspect, so please plan your activities accordingly.

THE ENVIRONMENT:

The final project will be an obstacle course which will contain elements of your past homework. The Turtlebot Burger will have to autonomously navigate through the Gazebo environment shown below:





The robot will have to complete the following tasks:

1. **Task 1: Wall following/Obstacle avoidance** - The Turtlebot starts here. It must successfully follow the wall and avoid the obstacles until it reaches the yellow line. Create a map of this corridor using a SLAM package of your choice. (You could also come up with your own solution for wall following and obstacle avoidance).
2. **Task 2: Line following & Stop Sign Detection** -
    - The Turtlebot must successfully follow the yellow line.
    - Stop Sign detection - While navigating the yellow line, the the Turtlebot should stop at the stop sign for 3 seconds before continuing. The stop-sign will be detected byTinyYOLO
3. **Task 3: AprilTag tracking** - For this task you will need to spawn another TB3 in the environment in the empty space past the yellow line and attach an AprilTag to the robot. The TB3 with the AprilTag will be teleoperated by the user and the preceding TB3 needs to track its motion.

    Note: There are several packages online to teleoperate robots in simulation (or you could also use the simple TB3 package). Make use of the concept of **namespaces** and **remap** to send different /cmd_vel commands to different robots.

KEY INSTRUCTIONS

1. Create a single launch file to bring up your entire code base. You may use a switch-case algorithm that accepts a keypress to indicate when one task is over and another begins, but that should be the only time you touch the keyboard.
2. To properly launch the environment in Gazebo, set the `GAZEBO_MODEL_PATH` variable to point to models in your `aue_finals` package too. Use the following command to do so:

```
export
GAZEBO_MODEL_PATH=~/path/to/workspace/src/aue_finals/models:~/path/to/worksp
ace/src/turtlebot3_simulations/turtlebot3_gazebo/models
```

3. Download the environment, build your workspace and launch the world with the following command:

```
$ roslaunch aue_finals turtlebot3_autonomy_final.launch
```

4. There will be **CONTINOUSLY GRADED** assessments till the final deadline. Teams will need to report their progress as per the schedule. As mentioned in the beginning, this is a challenging assignment and will require efficient team management from all the teams.

BONUS:

Can you navigate the above course completely autonomously without the need to switch tasks with keyboard presses? What are the different ways you can implement this? Bonus marks for doing it successfully.

Hint: Here is one way, using AprilTags. In the Gazebo model given above, you can use the model for the stop sign as a reference for incorporating AprilTags into the environment. You can place the tags at the start of every new task to "switch" the code. You can also use this approach to attach AprilTags to your TB3. You could deploy the tags into the environment as a "visual only" obstacle at the beginning of every task. If you remove the collision component from the SDF file, the robot camera will be able to see the Tag, but the block will have no physical property -- the robot can pass through it like a curtain. Or you can place it on the side, like the stop sign.

SUBMISSION

1. Each team will have to submit their continuous assessments in the form of presentations every week (on Tuesdays and Thursdays, as per their team number).
2. The Final Project Presentation needs to be submitted by 25th April 2023.
3. The Final Project Report is due on 4th May 2023.
4. As usual, submit your code via Github.