AuE 893: Sp'20: Autonomy Science and Systems

Department of Automotive Engineering, Clemson University

Adhiti Raman, Mugdha Basu Thakur, Ardashir Bulsara and Venkat Krovi

Final Project Instructions

The last 1.5 months of the Autonomy course have been restructured due to the online-only deployment, so this document is intended to give you an overview of what the final project will look like.

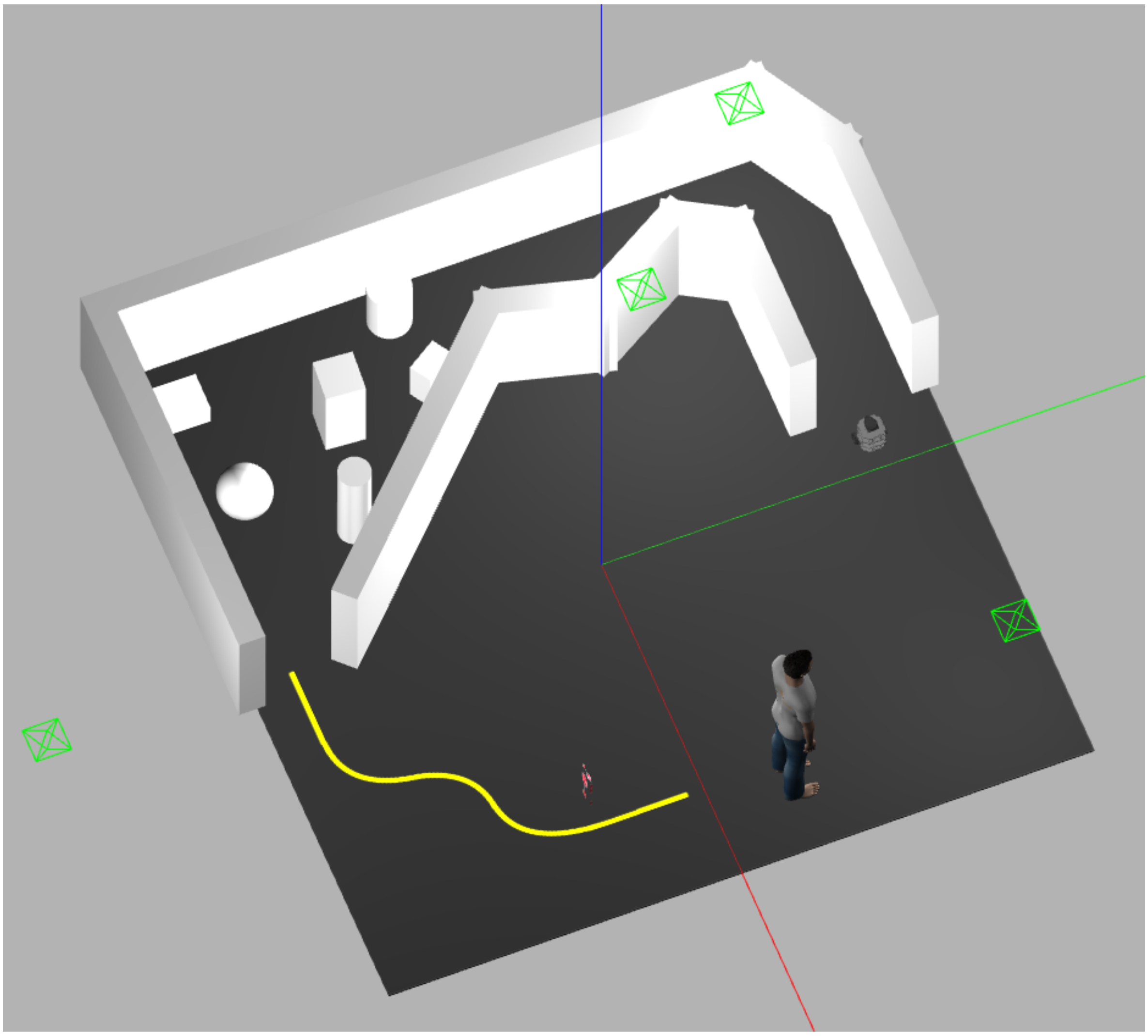
Since we cannot have on-site in-person demos, the entire final project will be deployed only on Gazebo. You may deploy your code on the physical Turtlebot at home for each of the tasks individually to the best of your ability to do so -- **but it will not be a "graded deliverable"**.

In the interest of social distancing, please **do not** meet your team to work together on the physical robot outside of ICAR.

Teams should work and coordinate to complete the project online.

The Gazebo 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.
2. **Task 2: Line following** - 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 by TinyYOLO (handout pending)
3. **Task 3: Human tracking** - The Turtlebot must use a trained DL network to identify the human in the environment and follow it around (handout pending). The human in Gazebo can be teleoperated around using the keyboard. This teleoperation is already part of the given Gazebo environment.

Key instructions:

1. To get started, you will need to incorporate a camera into your Turtlebot Burger Gazebo model.
2. 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.
3. For the final exam, you will demonstrate the working live over WebEx/Zoom through screen-share.
4. There will also be a separate PPT-only component, so be sure to take many good videos for your PPT.

The environment is provided in a zipped file on Canvas. Instructions to run it are included in the README.

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 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.

As always:

- Publish code on Github, submit link to Canvas.
- Update README to describe how to run your code and the contributions of each team member.

NOTE: This document will be subject to modifications based on your comments, feedback and progress.