

Photogrammetry & Robotics Lab

Techniques for Self-Driving Cars



Introduction to Control Assignment

Cyrill Stachniss, Jens Behley, Benedikt Mersch, Lucas Nunes

With contributions from Nived Chebrolu, Lasse Peters, Igor Bogoslavskyi,
Daniel Wilbers

First Assignment

- All material is on Ecampus (Instructions + Code)

  Assignment 1 (Mandatory)

Remaining Working Time: 27 Days, 2 Hours, 21 Minutes Edit Until: 29. Nov 2023, 12:00

Work Instructions

The goal of this assignment is to get hands-on experience in implementing controllers for following a trajectory. Please see the instructions in the assignment PDF.

Files

<i>ex1_control.pdf</i>	Download
<i>assignment_1.zip</i>	Download

Handing In

Schedule

Start Time	Today, 00:00
Edit Until	29. Nov 2023, 12:00
Remaining Working Time	27 Days, 2 Hours, 25 Minutes

Your Submission

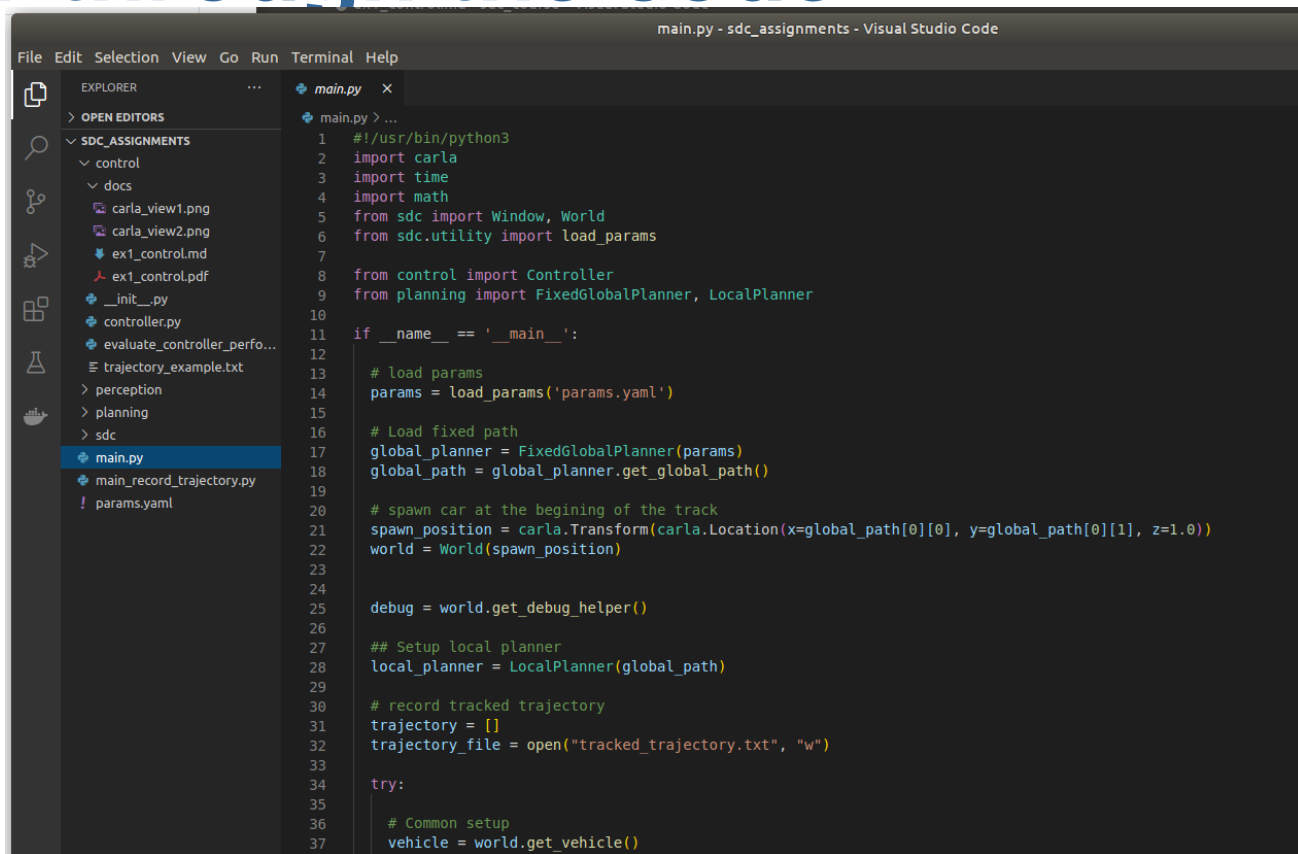
Team Members	[behley] <button>Delete Team</button>
	<button>Manage Team</button>
Submitted Files	You have not submitted any files.
	<button>Hand In</button>

- Submit zip file with `LastName1_LastName2.zip`

Tasks

1. Record Trajectory
2. Implement PID Controller
3. Implement Geometric Lateral Controller
4. Implement Model Predictive Controller (MPC)

A tour through the code



main.py - sdc_assignments - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

OPEN EDITORS

SDC_ASSIGNMENTS

- control
 - docs
 - carla_view1.png
 - carla_view2.png
 - ex1_control.md
 - ex1_control.pdf
 - __init__.py
 - controller.py
 - evaluate_controller_perfo...
 - trajectory_example.txt
- perception
- planning
- sdc
 - main.py
 - main_record_trajectory.py
 - params.yaml

```
1  #!/usr/bin/python3
2  import carla
3  import time
4  import math
5  from sdc import Window, World
6  from sdc.utility import load_params
7
8  from control import Controller
9  from planning import FixedGlobalPlanner, LocalPlanner
10
11 if __name__ == '__main__':
12
13     # load params
14     params = load_params('params.yaml')
15
16     # Load fixed path
17     global_planner = FixedGlobalPlanner(params)
18     global_path = global_planner.get_global_path()
19
20     # spawn car at the beginning of the track
21     spawn_position = carla.Transform(carla.Location(x=global_path[0][0], y=global_path[0][1], z=1.0))
22     world = World(spawn_position)
23
24
25     debug = world.get_debug_helper()
26
27     ## Setup local planner
28     local_planner = LocalPlanner(global_path)
29
30     # record tracked trajectory
31     trajectory = []
32     trajectory_file = open("tracked_trajectory.txt", "w")
33
34     try:
35
36         # Common setup
37         vehicle = world.get_vehicle()
```

Some additional hints

- Set sensible parameters in “`params.yaml`”
- Look in code/utils, etc. if we have already provided a helper function
- CARLA: if error appears that you cannot spawn at that location → restart CARLA/reload world

Questions?

Next Week: Q & A

- Discuss questions related to the task
- General questions

Thanks for your attention