coursera

Programming Assignment: Final Project: Vehicle State Estimation on a Roadway

You have not submitted. You must earn 16/20 points to pass.

Deadline Pass this assignment by April 7, 11:59 PM PDT

Instructions

My submission

Discussions

Project Overview

In this assignment, you will implement the Error-State Extended Kalman Filter (ES-EKF) to localize a vehicle using data from the CARLA simulator.

You will complete the following steps in order to pass this assignment:

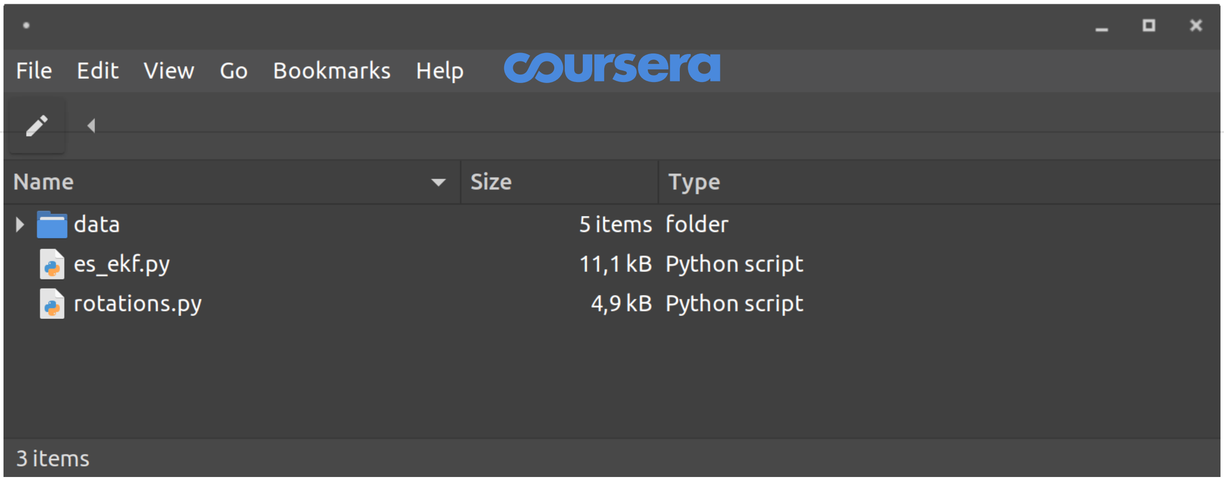
1. Download the relevant assignment files from the link below and examine the code to try and understand how it works.
2. Add code to the section marked “5. Main Filter Loop” to implement the ES-EKF. For details of the equations you should be coding up, see Lecture 2 (link to lecture 2).
3. To generate the required outputs, you will run the file with certain sections commented/uncommented as described in the “Getting outputs for submissions” section below.

Getting Started

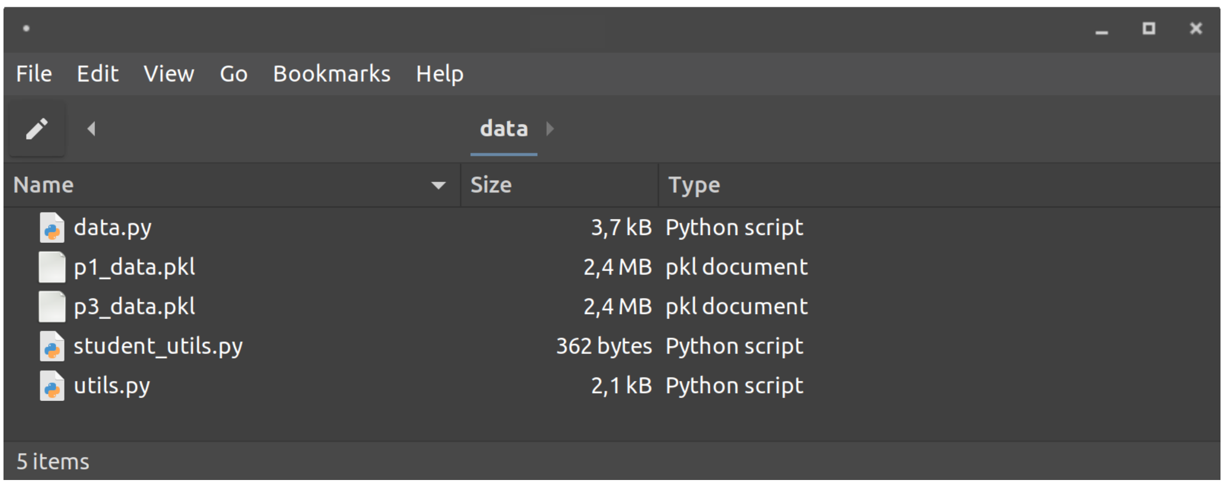
Start by downloading c2m5_assignment_files.tar.gz file from the link below.

c2m5_assignment_files.tar.gz

In it, you'll find es_ekf.py, rotations.py, and a folder labelled data, as shown below.

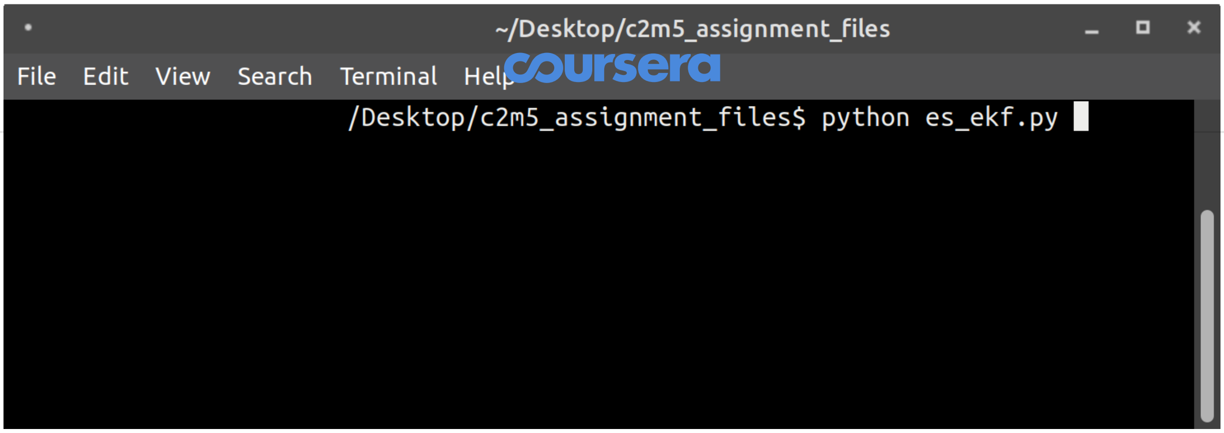


You will be adding code to the es_ekf.py to finish the sections that are labelled based on what you have learned in this course. The data folder contains the data you will use for the assignment, and the rotations file contains a Quaternion class and other rotation functions that are already implemented for you.



There are a large number of block comments throughout es_ekf.py. Take a look at the comments to try to understand what each section of the code is doing, and in particular to understand all of the data contained in the pkl files. As the code runs, some visualizations are already coded up for you, including a visualization of the ground truth trajectory, a visualization of the ground truth trajectory compared to your estimated trajectory, and error plots for all six degrees of freedom.

To run es_ekf.py, simply call 'python es_ekf.py' from the command line or 'run es_ekf.py' from within an interactive shell.



Getting outputs for Submissions

You will be submitting text files to coursera that contain the positions of five estimates from your solver. To get your submission txt files, you'll have to ensure the correct parts of the code are commented/uncommented out.

In the 'Data' section of the code, you'll see the following line:

```
with open('student_data/p1_data.pkl', 'rb') as file:
    data = pickle.load(file)
```

Ensure that you leave this as p1_data.pkl for parts 1 and 2. For part 3, switch it to p3_data.pkl.

The following block of code shows where the calibration rotation matrix is for determining the rotation between the LIDAR frame and the IMU frame.

```
# This is the correct calibration rotation matrix, corresponding to an euler rotation of 0.05, 0.05, .1.
C_li = np.array([
    [ 0.99376, -0.09722,  0.05466],
    [ 0.09971,  0.99401, -0.04475],
    [-0.04998,  0.04992,  0.9975 ]
])

# This is an incorrect calibration rotation matrix, corresponding to a rotation of 0.05, 0.05, 0.05
# C_li = np.array([
#     [ 0.9975 , -0.04742,  0.05235],
#     [ 0.04992,  0.99763, -0.04742],
#     [-0.04998,  0.04992,  0.9975 ]
# ])
```

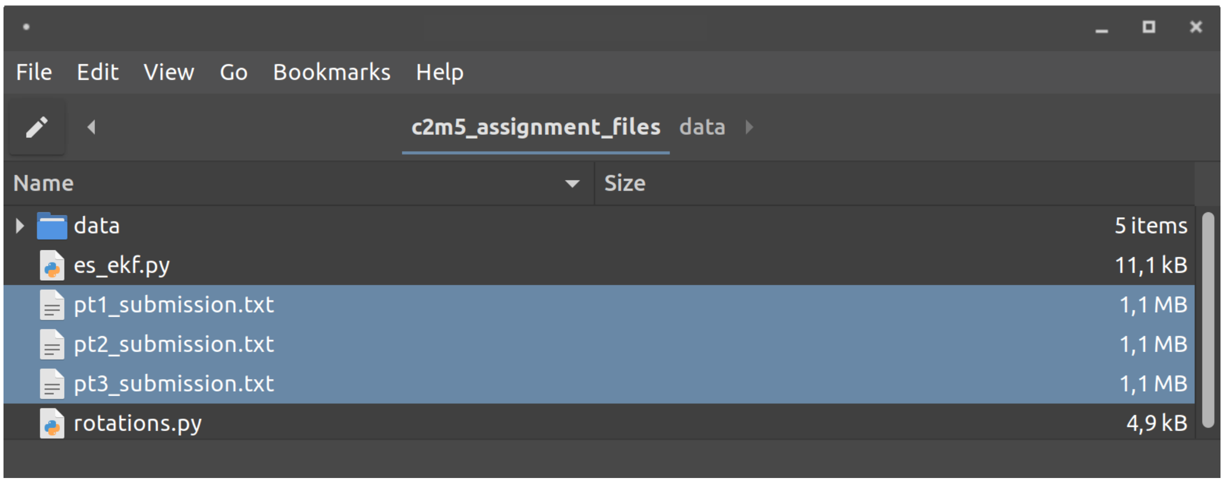
Ensure that you leave the top part uncommented for parts 1 and 3, but leave the bottom part uncommented for part 2.

As well, at the bottom of the code, you'll see three blocks that look like this:

```
# Pt. 1 submission
p1_indices = [9000, 9400, 9800, 10200, 10600]
p1_str = ''
for val in p1_indices:
    for i in range(3):
        p1_str += str(p_est[val, i]) + ' '
with open('pt1_submission.txt', 'w') as file:
    file.write(p1_str)
```

Ensure that the lines under Pt. 1 submission are uncommented for part 1, the lines under Pt. 2 submission for part 2, and the lines under Pt. 3 submission for part 3.

Once you're ready to submit, you should have three separate txt files titled ptx_submission.txt, where x is the part number.



You will receive 20% of your grade for each part for each position that is within a certain threshold away from the position given by our solution.T

Tricks and Tips

The final project is challenging! Here are a few tricks and tips to help you get started and to assist in debugging your code:

1. For Part 1 (and Parts 2 and 3), be sure to be *very* careful to properly treat the orientation states (i.e., states that involve rotations). As discussed in the course lessons, (sometimes small) errors when dealing with rotations can result in significant and difficult-to-diagnose bugs. Be sure to double check!
2. For Part 2, you will change the extrinsic calibration of the LIDAR sensor relative to the vehicle frame. This (purposefully) results in a systematic modelling error - what effect will this have on the estimator performance (recall the lesson on sensor calibration...)? To achieve better performance, you may need to adjust the variance of the LIDAR measurements in the filter (the variable 'var_lidar'). What change could you make to attempt to deal with a modelling error?

3. For Part 3, you will examine what happens when positioning information from the GPS receiver and the LIDAR 'drops out'. In this situation, the filter will propagate the motion model forward in time, using the IMU measurements to dead reckon. During the period where positioning updates are unavailable, how would you expect the variance (uncertainty) in the position states (x, y, z) to change? Should the filter remain consistent?

Good luck!

How to submit

When you're ready to submit, you can upload files for each part of the assignment on the "My submission" tab.