PATH PLANNING AND CONTROLS

Instructions

- Parts of the module highlighted in green are "checkpoints" and trainees are required to update the following task sheet once every checkpoint is completed.
- Documentation is a must for every checkpoint and trainees are required to create a google doc sheet where they document their learnings, errors encountered and doubts. This google doc must be made accessible to "anyone with link" and the link for the same must be updated on the task sheet.
  ~~Feel free~~ you are **encouraged** to ask doubts to JDEs if you feel stuck or want to understand a topic better
- Keep in mind, performance in modules is how the team will judge your abilities and effort to assign subsystems once the time comes
- Have a fun learning experience!

You should have developed an idea about Racing lines also after all you are in a *Racing* Team **:]**
And you must also be able to navigate the actual car without a driver through the track along those Racing lines. So let's dive into the subsystem responsible for it Path Planning and Controls !

So as the name suggest the subsystem is divided into two parts:**1)Path Planning:-**
Path planning involves generating the best possible track that our car can traverse for best possible lap time
2)Controls:-
This involves generating the inputs to car - steering, braking and acceleration, to follow the generated path
Our team focuses on participating in Formula Student UK/Germany/Netherlands etc and these competitions have their own specific problem statements.
In general, our race car is required to navigate through a racetrack defined by cones on either side (yellow cones on the left and blue

cones on the right). This being a time based, we need to push our car to its limits

Following are the main objectives of our system:-

- **First Lap -** In the first lap, we don't have the location, so we try to follow the midpoint at a relatively slower speed **:[** This is because there might be sudden turns ahead which can't be predicted without knowing the cones location. Keep in mind the car has never seen the environment it going to
- **Rest of the Laps -** Now when we have the positions of all the cones the actual path planning, involving our main "optimization" algorithm, to generate the path for best possible time **:]**
  Check this out to get an idea of our aim :**https://www.youtube.com/watch?v=FbKLE7uar9Y**
  (You might have seen this video already in the orientation but now you know what is happening behind the scenes)

- **Steering / Acceleration / Deceleration -** We get the position and pose our car by integrating perception and SLAM subsystems, and we use this to devise the inputs to the car

Now that we know what exactly needs to be done, we get

Fundamental theory

1. **Python (fundamentals):**
   Our entire codebase is currently on python (and we aim to switch to c++ a few years down the line since python is slower than c++) and this requires a firm grasp on python, its functionalities, data structures and working in general. Those confident in the same (since it has already been done in the software module) can skip learning python fundamentals and directly move on to the next task. Resources:
   - https://www.youtube.com/watch?v=_uQrJ0TkZlc watch till 3:50:47
   - https://www.w3schools.com/python/numpy/default.asp : Numpy - https://www.w3schools.com/python/pandas/default.asp : Pandas

- https://www.w3schools.com/python/matplotlib_intro.asp : Matplotlib
- You are free to use any other resource if you wish to learn python fundamentals Assignment:

Using matplotlib, plot the birth rate of India, Nepal and Bhutan Of the last 5 years in the same graph using different colors and try to use label for axis and different color for plotting Assumptions are most welcome (if you don't find birth rate values, just introduce some reasonable numbers)

Checkpoint-1!

2. Interpolation:

This is where the plan to plan a path actually starts. In the real world we can only get limited number of discrete points rather than a continuous set of paths that we need to follow.We need to connect those discrete points to extrapolate and obtain our trajectory.There are different way to "connect these dots", and we must study these to choose the best for us.You are free to explore these topics online for better understanding.

Note down the advantages/disadvantages(feasibility, use case etc, because finally we need to optimize this path) you think we have for different methods.

You are encouraged to explore on these topics

Resources:

A)Linear and quadratic Spline Interpolation

-https://www.youtube.com/watch?v=KLUr1A6vyzs -https://www.youtube.com/watch?v=j_jBK7zJ1vU

-https://www.youtube.com/watch?v=kCPMph3cPA8

B)Cubic and higher degree spline interpolation

- https://www.youtube.com/watch?v=wMMjF7kXnWA

Example:-https://www.youtube.com/watch?v=gT7F3TWihvk

(first try solving it yourself by understanding the problem and then cross check for any difficulties, now imagine doing this for hundreds of points :)

Assignment:
You are given four discrete points (in increasing order of x which you can assume (x1,y1),.,.). You need to interpolate the four points using cubic spline interpolation. You cannot use direct interpolation functions (can look up source code for reference). Feel free to search it up on google, but you should know exactly what the code is doing.

After that replace those points with some known points and check if you are getting the interpolated curve. When plotting a piecewise function you can use np.linspace to divide x-axis to make it look like a continuous line Checkpoint-2!

3. **Control systems:**

To keep our car on the desired path, we need to give appropriate inputs to the cars, namely-acceleration, deceleration and steering.
Basics of a controller:

- https://www.youtube.com/playlist?list=PLn8PRpmsu08pQBgjxYFXSsODEF3Jqmm-y(mainly used to control velocity) Assignment:

Just find some real-life applications of pid controllers and try to implement it on python. If you aren't able to code it, your progress will matter.

Checkpoint-3!

Overall Assignment- It will be given later

TIMELINE:

Checkpoint 1- 21/3

Checkpoint 2- 26/3

Checkpoint 3- 29/3