TECHNISCHE UNIVERSITÄT WIEN

Autonomous Racing Cars
191.119 (VU 4,0) Semester: 2022S

Lab 7: Advanced Racing 1

2022-05-31

Preface

Read all the instructions below carefully before you start working on the assignment, and before you make a submission. All sources of material and ressources must be properly cited (this also includes datasheets).

- Completeness of solution: A complete solution of a task also includes knowledge about the theory behind.
- Exercises are to be solved in teams. All team members must be indicated on the submission protocol. However, every team member must be able to explain the handed in solution. Grading is on an individual basis. Upload your solution (one per team) in TUWEL until 2022-06-14 23:59.
- For this assignment there is no exercise interview. However if submissions are unclear, students might be invited for exercise interviews.
- Additionally short presentations of every team at the *Lab 7 presentation* are part of the assignment (details see below).

Learning outcomes

The following fundamentals should be understood by the students upon completion of this lab:

- Analyse and formulate project requirements, formulate goals, plan projects and identify potential future directions.
- Work with non-fully specified/formalized project descriptions.

Deliverables and Submission

- Write a exercise report and submit it as PDF file in TUWEL until 2022-06-14 23:59. Use the provided latex template and do not forget to fill in the parts marked with "TODO". The anonymous version of the lab report (all pages except the first one with personal data) will be shared with all the teams after the submission deadline. In any case the report must meet an adequate level for layout and readability that is appropriate for the academic context. It needs to be detailed enough, so that another team could reproduce your work without additional information.
- Submit a ZIP-Archive with all the relevant source code in TUWEL until the same deadline.

Transparency of Contribution

Describe in your submission protocol briefly how you worked together. How did you structure your work distribution and collaboration? Who contributed how much effort to which part of the work? (If one, or more team members, are not able to work on this assignment, you must also transparently state this here.)
(Please do not understand this preamble wrong to somehow exaggerate your contribution estimation: If you are working together well in your team, it should anyway be no problem to briefly describe how you worked together.)

1 Advanced Topics

In the previous labs you have worked with a ROS-based simulator and learned a lot about different algorithms for autonomous racing.

Now it is time to have a look at a bit more advanced topics to work towards an actually competitive race car. As already discussed at the last lab-info session, there are a lot of different parts that can be improved. By afterwards feeding all the team's outcomes together, we hope that you will all get the most advance and insights out of this lab.

For each of the topics the general lab structure should be the same, which is as follows:

a.) Based on the abstract that you submitted for lab 6 task (3.a.), formulate a more detailed goal. State your current basis (e.g. already available implementation from previous labs) to improve on. Describe what you want to achieve. Make plans how to do this. State how you are going to measure if you have achieved your goal [1].

b.) Implement or carry out the plan that you defined in task (1.a.).

c.) Add a structured documentation of your work from task (1.b.) in your PDF protocol. The level of detail should be s.t. another team of the ARC class could reproduce the same work based on the PDF.

d.) Perform the measurement as defined by yourself (refer to last sentence of task (1.a.)). Discuss the results.

e.) Formulate directions for future work. If you fully achieved your goal, describe in which directions even more improvement would be possible. If you did not achieve your goal by some margin, describe what alternative plan or modified approach could be more promising for better results. If your plan did not work at all, describe the reasons and give ideas for alternative approaches.

f.) Prepare a short (7 Minute) presentation giving an overview of your work. This should focus on:

- What was your plan to achieve the goal?
- How did you implement or carry out the plan?
- What are the results?
- Directions for future work.

Every teams should hold their presentations to share their work with all other students. (Instead of only one team presenting, as it was until now.)

[1] Since you define the goal yourself it might happen that your plan does not work at all to achieve this goal. For this lab this is of course okay. But you need in any case describe your ideas, thoughts, attempts, and implementations clearly and thoroughly. This also applies if you underestimated the workload of your attempt. Then you might show the partial work and progress by the end of lab 7. But please make sure to also have all further subtasks (1.c.) to (1.f.) written down for your partial work.

Team abstracts and comments

Team 2

For lab 7 we aim at optimising our existing algorithms. We will base our work on our existing implementations of the disparity extender, the global planner, and the pure pursuit algorithm.

1. One crucial component for racing is the velocity calculation. We want to find a way to compute the fastest speed based on the available sensor information for both approaches. To assess the algorithms, we will extend the simulator with a lap time computation and plot the speed over time to look for optimisation potential. To minimise the simulation-to-reality gap we will try to measure or estimate the simulator parameters as good as possible.

2. For the disparity extender, we already have a velocity mapping based on the stretch length. We will try to improve this by finding a reasonable nonlinear mapping, potentially based on the expected (or measured) braking behaviour.

3. For the planning-based navigation we want to consider not only the distance towards an obstacle ahead, but mainly the knowledge about upcoming curve radii for velocity calculation.

4. After tuning both algorithms, we should be able to compare them on a variety of simulator maps (as well as on-site maps using SLAM). This helps us to decide how to proceed in lab 8 as we think disparity extender is already a very nice racing algorithm and we are not sure how much we can actually gain from global planning – especially in the light of mapping and localisation imperfections.

5. To assess these imperfections we want to try our path planning algorithms on the hardware car as fast as possible to be able to mitigate the simulation-to-reality gap quickly.

6. Furthermore, we think that better path planning can help to improve the race performance. We plan to extend our planner beyond the improvements outlined in Section 2.2.b. by considering not only the distance towards the finish line, but also maximising curve radii. We want to follow the gradient of the (weighted) superposition of both metrics to improve our paths. To do so, we need to compute the curve radius for each pixel after every position update in the planning phase. This can be based on lookahead trajectories (which have to be as long as an average curve on the map). At the moment, the trajectories are mainly smoothed out by the lookahead distance of the pure pursuit algorithm. However, this does not necessarily use the space on the map efficiently and might cause collisions. Instead, a better planner can create smoother paths (which make pure pursuit easier) and increase the possible velocities. To evaluate and tune this algorithm, we want to consider again the lap times as well as the path lengths.

7. In lab 8 we plan to develop methods that allow us to cope with map uncertainty and obstacles (mainly other cars). This can be either based on local replanning or combinations of reactive and planning-based algorithms. However, to define this work in further detail we need additional data from the hardware car as we cannot assess the performance of planning-based navigation on the real car at the moment

Comments

(1) Good metric and useful helper. If you can provide the lap-timer in a separete node and configurable (with respect to setting the finish line) it will be useful for other teams as well.
(2,3,4) Yes, tuning both algorithmns extensively and then comparing them is a good idea. You may also consider to already pre-calculate the speed profile for the raceline. Having the possibility to pre-calculate might be a significant benefit over reactive methods.
(5) Highly welcome.
(6) Yes, curve radius is essential. Several race-line optimization tools take this into account. Interesting aproach in 2.2.b.

Team 3

This subtask is to give an abstract of what further plan we would like to follow in the next two labs:

1. Adding race track optimization for path and speed profile based on road geometry. Currently the race-line is done with A*-Algorithm which tries to find the shortest path. However, the shortest path isn't always the fastest nor the one with best vehicle stability. Additionally, the speed profile is currently designed only based on curvature of the path and does not consider vehicle drive-train capabilities or road friction. Different optimisation based approaches are available which solve a nonlinear programming problem while considering vehicle dynamics and road geometry. These algorithms optimizes in time minimum fashion the race-line and speed simultaneously.

2. Combing speed and steering control: Pathfollow Controller. To enable good path following capabilities, predictive controllers with knowledge of vehicle dynamics and actuator dynamics should be considered. Additionally, stability constraints and actuator constraints should be taken into account. To tackle these problems, model predictive control (MPC) can be used.
– MPC based on linear time varying (LTV) bicycle model. At each discrete sample time, the plant model of the car is linearized along the offline optimised velocity trajectory and used to solve a quadratic optimisation problem (QP-Problem).
– For robust and fast solving of the QP-Problem, several commercial and open-source solvers are available. We want to use the OSQP solver from the Oxford university for the ease of use and good documentation as well as fast solver times.
– Already some small test were performed on laptops without ROS and an average execution time of 2ms were calculated. Given a sample time of 0.1s, the race-car hardware should be capable of performing the required calculations in time.

3. Adding obstacle avoidance logic to either switch from MPC to follow the gap or modify optimized race track locally. The MPC can only follow a given race-line. When an obstacle suddenly appears on the race-line, it cannot avoid it. Simple replaning algorithms based on geometric objects (circles, splines) can be used to adapt the race-line.

4. (optional) Implementing an obstacle avoidance MPC or local graph based replaning.

Comments

(1) Your aproach for raceline optimiuation sounds good, go ahead.
(2) MPC for path-following is also a good idea. Note, that you should not forget to define a metric and show your improvements. E.g. you could compare MPC against your already implemented Pure Pursuit then.
(3) and (4) might be too much work for this lab. But worth considering for lab 8 if you are sill interested in this direction then.
Good idea to make use of public available tools and ressources.

Team 4

Primarily, we are interested in improving our path-finding methods. Currently the paths generated are far from optimal and also somewhat noisy, perhaps due to algorithmic problems. Once we are happy with our path-finding method and if there is time left, we could look at reacting to changes in the track on-the-fly, e.g. other cars. There are a few ideas for improving path-finding which we would explore:

1. Better cost function
   Currently our cost function is simply based on distance to the closest wall. It might be possible to come up with more sophisticated cost measures, e.g. taking better racing lines into account, preventing unnecessary distance from walls, etc.

2. Path post-processing
   Once the path-finding algorithm has come up with a path in pixel-space, we would explore various form of post-processing. This could be done while still in pixel-space using kernels, cellular automata, etc.. Or after moving into world-space by converting to line segments and vertices, vertices could be pushed around to optimize for measures such as curvature, length, among others.

3. Waypoint sampling
   Currently we simply take every nth pixel from the path as a waypoint. It would be sensible to instead take more waypoints in sections with a lot of curvature, and take less where the path is very straight.

4. Neural-Circuit-Policy based Reinforcement Learning
   We're still in the process of finding a use case scenario to employ NCP in RL and use it in a valid function. At the moment this step is tentative but not confirmed yet.

5. To react to changes in the track/other cars, we could implement a modified follow the gap algorithm that incorporates choosing gaps that are along the line of waypoints. With this, the car could briefly switch to a follow the gap algorithm when encountering obstacles that prevent a direct route to the next waypoint.

To assess if the path-finding optimizations were successful, we can simply look at how the laptime has changed on various tracks. Evaluating changes to reactivity will be more difficult. Depending on required effort, it might be of interest to extend the simulator to handle multiple cars on the track.

Comments

(1) Better cost function, (2) Path post-processing and (3) Waypoint sampling are good aproaches and work niceley together. Go ahead with this part.
(4) Neural-Circuit-Policy based Reinforcement Learning might be too much work for this lab, you can go on this direction for lab 8 if you are still interested in this.
(5) Yes, this is a good aproach. You can combine this with a improved raceline (1-3).
Metric of lap-time is ok.
Updating the simulator for multi-car scenarios would be actually very nice, but maybe too much work for this lab. Suggestion: Run the map-server twice and publish two maps to different topics. One of them is without obstacles, the other one has static obstacles. Then you can simulate obstacles by using the one for the simulator and the other for the raceline-generation.

Appendix

Grading

The following points can be achieved for each task of this exercise sheet:

Exercise	Points
1.a.	10
1.b.	43
1.c.	10
1.d.	7
1.e.	10
1.f.	20
Subtotal	100
Grand Total	100

Acknowledgments

This course is based on F1TENTH Autonomous Racing which has been developed by the Safe Autonomous Systems Lab at the University of Pennsylvania (Dr. Rahul Mangharam) and was published under CC-NC-SA 4.0 license.