Platforms and Algorithms for Autonomous Driving Planning and Control Module

Assignment 2: Motion Control

AA 2024/2025

Deadline: 13/01/2025, 23:59.
Bonus point if submitted by 23/12/2024 23:59

Introduction

The purpose of this assignment is to make in practice the control algorithms seen during the lessons. By completing this task, you will gain experience implementing and/or tuning the controllers to track a path and a velocity.

It is highly recommended to read/study the paper Kinematic and dynamic vehicle models for autonomous driving control design to have potentially some inspirations if you want to use the kinematic model at high speed instead of a different, and more suitable, model.

Instructions

- The assignment must be solved individually.
- A short report must be written for each assignment, describing the results, including motivations, observations from the simulations, and any conclusions.
- The report should be concise and clear, with all figures properly labeled, including axes and legends.
- The code should be well-commented to ensure clarity.

- Plagiarism will not be tolerated and will result in points being deducted.
- Submit your work sharing a private GitHub repository or as a zip file.
- Use the template shared. It is sufficient to complete the code with the requests below for exercise 1 and 2, not totally for exercise 3. You are free to extend it to improve the management of the plots, use different libraries for the control algorithms, save the results in an external file and use the data for plots or something else in a different environment, etc.

Vehicle Parameters

Use the same simulation.py script produced for the Assignment 1, using the non-linear single track, RK4 and integration time of **0.05s** as it is already defined in the main.py template.

Outputs

For each exercise, you are required to plot the following outputs:

- Trajectory to track and actual trajectory: x, y
- Longitudinal velocity: v_x
- Lateral velocity: v_y
- Front slip angle: α_f
- Rear slip angle: α_r
- Lateral tire force: F_y as a function of slip angle
- Steering angle δ
- Side slip angle β
- Lateral error
- Velocity error
- Longitudinal acceleration

as well as create a report in which for each exercise include the most useful plots and describe the results.

Exercises

Exercise 1: Longitudinal Control (Points: 3)

In this exercise, you will complete the template code in order to track a certain velocity, in particular:

- Tune the PID gains in order to track a speed of $15 m/s$
- Tune the PID gains in order to track a speed of $25 m/s$
- Keep steering angle at zero
- Implement the antiwindup as done in slide
- Set a limit of -2 and +2 for the PID output

Plot the requested outputs for both velocity conditions and indicate what are the settling time and eventually the overshoot, as well as other things you could consider useful.

Exercise 2: Low-speed Lateral Control (Points: 7)

In this exercise, you have to calculate the steering angle to apply in order to:

- Track the *ovaltrj.txt* path, first at $10 m/s$ then at $20 m/s$.
- Use the PurePursuit algorithm.
- Use the Stanley algorithm.
- The longitudinal speed error should be less than 5% of the target speed after the settling time.
- The lateral error should be less than $1 m$
- Make the simulation long enough to complete the whole path once
- Bonus: extend the PurePursuit to make the lookahead depended on the curvature. You can use the calc_curvature() function of the Spline.

Exercise 3: High-speed Lateral Control (Points: 5)

In this exercise, you have to calculate the steering angle to apply in order to:

- Track the *ovaltrj.txt* path, first at $23 m/s$ then at $25 m/s$. Respecting the lateral and speed error indicated in the previous exercise.
- Use the PurePursuit algorithm.
- Use the Stanley algorithm.
- Use LQR or MPC (**both controllers for MUNER students or MPC with also another model such as linear and nonlinear, and as an additional bonus point if done for CS students**).
- Bonus: Try to track the path at the highest speed you can respecting the lateral and speed error requirements using one of the algorithms.

At $25 m/s$ it could not be **easily** feasible to track the path at the requested requirements. In that case, relax the lateral error requirement and show your best result.

Deliverables

For the submission, you should provide the following:

- A clear and concise report in PDF format, including considerations and explanation of the plots you may consider useful.
- A well-documented code repository (via GitHub) or a zip file with all the source code.
- Any additional files necessary for the simulation (e.g., external files for saving results).