

## Lab 2: Modeling Longitudinal Motion

### Exercise 1. (*Path tracking basics*)

In path tracking mode, the vehicles' steering is adjusted by the Stanley controller (Gabriel M. Hoffmann, Claire J. Tomlin, Michael Montemerlo and Sebastian Thrun. *Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing*. IEEE American Control Conference, 2007.).

- The file `CPMLAB/mid_level_controller/src/Controller.cxx` implements a speed controller and calls the path tracking controller. What kind of speed controller runs on the lab vehicles? What are the benefits of having the speed controller on the vehicle instead in the high-level controller (HLC)?
- Let a vehicle follow the path given by `TEAMREPO/+cmmn/outer_lane_path.m`. Understand how the path is defined and use this path for the remaining exercises.

### Exercise 2. (*Repository*)

- For smooth execution of your code in the lab, clone your repo such that you end up with the following folder structure (example for git repository TEAMREPO):

```
$HOME/dev
|-- software
    |-- cpm_lib
    |-- high_level_controller
        |-- examples
        |-- TEAMREPO
        |-- ...
    |-- indoor_positioning_system
    |-- ...
```



The path to your repository will be called `TEAMREPO` in the following. The path to the CPM Lab software will be called `CPMLAB`.

- Navigate to `TEAMREPO` and configure your git username and email with

```
git config user.name "FIRST_NAME LAST_NAME"
```

```
git config user.email "MY_NAME@example.com"
```

Please do not use the option `--global` as you are sharing a user account with others.

### Exercise 3. (*Code template*)

- Examine the file `TEAMREPO/main.m`.
- Examine the file `TEAMREPO/+mpmc/Hlc.m`. It is a code template for the position controller. Understand how a HLC is programmed using the `InterfaceHlc`-class.

#### Exercise 4. (*Experiment in the CPM Lab*)

The goal of this exercise is to make you familiar with the CPM Lab. Read [the introduction to running experiments in the CPM Lab](#), and follow the guide to run the path tracking example in the CPM Lab.

---

#### Checkpoint 1

Get a tutor to check your work. You should be able to

- submit a signed copy of the *Ordnung für Studierende* with the *Labrichtlinien und Sicherheitshinweise* attached (documents are handed out in the lab)
  - state the requirement on the first and last point of a closed path so that the interpolation along the path's distance works correctly
  - explain the `start` function of the `InterfaceHlc`-class
  - show a video of the path tracking example running in the CPM Lab
-

### Exercise 5. (*Longitudinal model*)

The goal of this exercise is to create a state-space model of the longitudinal vehicle motion along a path in path tracking mode with the input speed  $v_{in}$ , and the outputs  $s$  and  $v$ , which are the position and the speed, respectively.

- a) You need to determine the position along the given path from the measured `VehicleStateList`. Fill the missing parts in the function `measure_longitudinal` in `TEAMREPO/+cmmn/MeasurementTransformer.m` for this.
  - b) The function `measure_longitudinal` returns the one-dimensional position and speed of the longitudinal motion. Draw a sketch that relates the coordinate system in which the position is measured to the reference path on the map. Where is the origin of the one-dimensional coordinate system?
  - c) Decide on a model that could be used to describe the dynamics between input and output speed.
  - d) Draw a block diagram of your model for the longitudinal vehicle motion.
  - e) Write down the state-space model for the longitudinal vehicle motion.
- 

### Checkpoint 2

Get a tutor to check your work. You should be able to

- show a sketch of the coordinate system for your model
  - show a block diagram of your model
  - show the state-space model you created
-