

## Lab 1: Setup and Lab Basics

### Exercise 1. (*Setup*)

Download the [provided virtual machine](#) and follow its readme.md file to finalize the setup.

Make sure the tag '2223' is checked out in the CPM Lab software repo. `git status` should return 'HEAD detached at 2223'. If it does, you can skip the steps of building the software. If it does not, check out the tag with `git checkout 2223` and rebuild the software.



More information on the CPM Lab can be found in the [documentation](#). The documentation will be your guide for most tasks for the remainder of the lab.

### Exercise 2. (*Git*)

If you are unfamiliar with git, go through the introduction to git provided through moodle.

### Exercise 3. (*Repository*)

- Create a private GitLab repository for your team. Invite your teammate and tutor and name the repo as follows: `cpnav_2223_group_<number><letter>`. For example the repository of group 2A should be named `cpnav_2223_group_2A`.
- 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.

- Initialize your repo with the code base given at [GitHub](#). Checkout the tag '2223'.

### Exercise 4. (*Controlling the vehicles*)

Open MATLAB. Read [how to run a high-level controller](#).

- Execute the example `CPMLAB/high_level_controller/01_direct_control/main.m` to see the vehicle moving in a circle.

- 
- b) The file CPMLAB/cpm\_lib/dds\_idl/VehicleCommandDirect.idl defines the message format of messages sent to control the vehicle in direct control mode. What are the contents of the interface definition language (IDL)-file? How is the message sent in MATLAB?
- c) The file CPMLAB/cpm\_lib/dds\_idl/VehicleStateList.idl defines the message format of messages sent to give information about the vehicle states. What are the contents of the IDL-file? How is the message read in MATLAB?
- d) Repeat Task b) and Task a) for the path tracking mode.
- e) Repeat Task b) and Task a) for the trajectory following mode.
- 

### Checkpoint 1

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

- run the path tracking example in your own setup
  - explain how communication between components is realized in the lab
  - explain the contents of a `VehicleStateList` message
  - show a GitLab repository with the code template, and your team and tutor as members
  - explain the git commands `pull`, `add`, `commit`, `push`
-