

Exercise 1. (*Distributed platoon control*)

Your goal in this exercise is to realize a platoon using priority-based non-cooperative distributed model predictive control (DMPC). Use the folder TEAMREPO/31_distributed_platoon_control for this exercise.

- Choose a communication structure for the networked control system and reason about your choice. Draw the resulting graph representing the agent interaction you designed. Which agents can compute their control actions in parallel, which in sequence?
- Find a suitable state-space representation of each agent in the platoon. Draw a sketch of the coordinate system(s) you are using.
- Define a message format to exchange the predicted output between the high-level controllers in the interface definition language (IDL). Name it H1cPlan.idl and save it in TEAMREPO/0_common/. Convert the IDL file to MATLAB classes with

```
DDS.import('H1cPlan.idl', 'matlab', 'f')
```

Create a test program with a data writer and a data reader to test your communication (see CPMLAB/high_level_controller/examples/matlab/init_script.m for an example). You can create filters in the data distribution service, so a data reader will only read messages with specific fields, e.g., a vehicle ID. Read more about data readers with filters in the [DDS Blockset Pilot Support Package documentation](#).

- Control the vehicles to drive in a platoon using DMPC.

Communicate through the message format you created in item c). Your goal is to achieve the best performance, so use any information allowed by your control strategy and communication structure that helps you in that regard.

Aim to maintain a constant reference distance d_{ref} between the center of gravity of consecutive vehicles of

$$d_{ref} = 0.5 \text{ m.} \quad (1)$$

Apply the following state constraint on the distance and the input and change of input constraints:

$$\begin{aligned} d_{min} &= 0.3 \text{ m} & v_{min} &= 0 \text{ m/s} & a_{min} &= -1 \text{ m/s}^2 \\ v_{max} &= 1.5 \text{ m/s} & a_{max} &= 0.5 \text{ m/s}^2 \end{aligned} \quad (2)$$

If quadprog gets stuck in infeasible regions, think about possible reasons. Implement a slack variable for the distance in the class ModelPredictiveControl using the exact penalty method.



In order to implement the exact penalty method, have the slack variable appear in the linear term of the cost function with a high weight (e.g. 10^6) and constrain it to be positive.

e) Evaluate the DMPC developed in Task d) with the following scenario:

Vehicles with IDs 1, 2, 3, 5 and 7 form a platoon. Vehicle 7 is the leading vehicle, while the others follow in descending order. The vehicles start in distances unequal to their reference distance as given through the initial poses of the simulated vehicles. Use Equation 1 as the reference distance and Equation 2 as the constraints. The platoon should follow the speed profile given by

$$v_{\text{ref}} = \begin{cases} 0.5 \text{ m/s}, & 0 \text{ s} \leq t < 15 \text{ s} \\ 1.4 \text{ m/s}, & 15 \text{ s} \leq t < 25 \text{ s} \\ 0.8 \text{ m/s}, & 25 \text{ s} \leq t < 35 \text{ s} \\ 0.0 \text{ m/s}, & 35 \text{ s} \leq t. \end{cases} \quad (3)$$

All vehicles in the platoon should reach the same velocity and aim to reach the inter-vehicle reference distance d_{ref} . Stop the experiment at $t = 40 \text{ s}$.

Examine the controller's performance with plots of the inter-vehicle distances, the vehicles' velocities and accelerations. You can use the function `TEAMREPO/0_common/plot_platooning.m` to generate relevant plots after an experiment.

If the situation allows it, we will run your DMPC controller in the CPM Lab and provide you with plots and a video of the experiment. The first line of `TEAMREPO/31_distributed_platoon_control/main_dmhc.m` must be a comment that states the necessary sample time setting for the middleware. Push your code before Thursday, 6 pm.

Exercise 2. (*Ideas for advanced features*)

Have some fun and experiment with the tools you have now mastered. Here are some ideas on what you might want to work on, but feel free to try anything that tickles your fancy.

- a) Increase the number of vehicles in the platoon.
- b) Have vehicles join and leave the platoon.
- c) Have multiple platoons drive, consider collision avoidance between platoons.
- d) Have different constraints on agent dynamics drive in the same platoon.