# Practical Exercise

## Modeling and Control of Hybrid Systems (sc4160)

## 2005 – Version 1.1

Delft Center for Systems and Control, Delft University of Technology

---

Note: Changes or additions with respect to the previous version are indicated by a vertical bar in the right margin.

## 1 General remarks

- This practical exercise consists of several steps that are outlined in a road map. You should follow this road map and present the results in a clear and concise report. In this report you should clearly explain and motivate all the choices you have made while solving the practical exercise. In your report you should also add an evaluation and conclusions section of max. 1 page in which you briefly outline the main insights you have obtained while making this practical assignment. You should also add the MATLAB files you have written in an appendix to the report.
- The deliverables of this assignment are:
  - a written report about the assignment (to be emailed as a **pdf** file to `b.deschutter@dcsc.tudelft.nl`, or — in case you do not know how to make pdf files — to be delivered as a hardcopy to Bart De Schutter);
  - a zip file containing your MATLAB files (to be emailed to `b.deschutter@dcsc.tudelft.nl`).
- You will be graded on the contents and the presentation of the report, on the originality[1] of your answers, on the correctness, the *efficiency*, the readability of the MATLAB files (i.e., do not forget to include explanatory comments in your MATLAB files), and on your performance during the oral discussion about your report.
  We would like to put forward Tuesday, April 5, 2005 as the date for the oral discussion. The deadline for emailing the reports on the assignment would then be Monday, April 4, 2005 at 10.00 a.m.
- As already indicated during the lectures the current assignment is "experimental" in the sense that some of the questions may be changed or extended based on the progress the students make or the errors or difficulties they encounter. This also implies that, e.g., some of the parameter values may be changed during the course period. Hence, you should take care to keep the computations symbolic or analytic as long as possible and not to hardcode any of the parameters in your MATLAB programs (instead, write one separate MATLAB function or script that defines the parameters) so that you can easily take new parameter values into account.

[1] I.e., a correct answer that differs from the answers given by the other groups will be graded higher than a correct answer that is an almost literal copy of the answer of another group.

- We recommend that after each step of the road map you check your intermediate results with the teaching assistants Ion Necoara or Rudy Negenborn. They can be reached via email at `i.necoara@dcsc.tudelft.nl` or `r.r.negenborn@dcsc.tudelft.nl`.

## 2 Set-up

We consider an adaptive cruise control (ACC) application in which 2 cars are driving one after the other (see Figure 1).

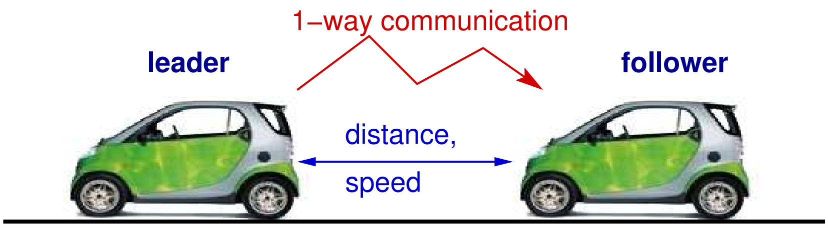


Figure 1: ACC set-up considered in the practical assignment.

In general, the aim of ACC is to ensure a minimal separation between the vehicles (i.e., distance keeping) and a speed adaptation (i.e., the speed differences between the vehicles should be kept as small as possible). In this exercise we will — for the sake of simplicity — only consider the speed adaption control and we assume that the leading vehicle communicates his speed to the following vehicle, which then has to track this speed as good as possible.

For the vehicle dynamics we consider a simplified model in which the following forces act on the vehicle (which has mass $m$) at time $t$:

- the "driving" force $F_{\text{drive}}(t)$, which is proportional to the throttle input $u(t)$: $F_{\text{drive}}(t) = bu(t)$,
- a dynamic friction force $F_{\text{friction}}(t)$, which is proportional to the square of the speed $v(t)$ of the vehicle: $F_{\text{friction}}(t) = cv^2(t)$.

Braking will be simulated by applying a negative throttle. We will assume that the vehicles drive in the forward direction, so the speed will always be nonnegative. For passenger comfort during the ACC operation we also include a maximal acceleration/deceleration: $|a(t)| \leqslant a_{\text{comf,max}}$. The parameters of the vehicle are given in Table 1.

| Parameter | Value | Units |
|---|---|---|
| $m$ | 800 | kg |
| $c$ | 0.5 | kg/m |
| $b$ | 3700 | N |
| $u_{\max}$ | 0.9 | — |
| $u_{\min}$ | $-1$ | — |
| $a_{\text{comf,max}}$ | 2.5 | m/s$^2$ |

Table 1: Parameters of the vehicle.

# 3 Tasks & Road map

**Step 1:** Note that as we are only considering the speed adaptation and as the leading vehicle communicates its speed to the follower, we only have to consider the following vehicle.
Write down the continuous-time model for the position $x(t)$ and speed $v(t)$ of the following vehicle.
Give the maximal throttle input $u_{\max}$ and the maximal braking input $u_{\min}$, determine the maximal speed $v_{\max}$ and the maximal acceleration $a_{\text{acc,max}}$ and deceleration $a_{\text{dec,max}}$ of the vehicle.

**Step 2:** Construct a piecewise affine (PWA) approximation $P$ with 2 regions of the friction force curve $V : [0, v_{\max}] \to \mathbb{R} : v \mapsto v^2$ as follows. We want a perfect match for $v = 0$ and $v = v_{\max}$. This implies that we still have two degrees of freedom, i.e, the coordinates $(\alpha, \beta)$ of the middle edge point of the PWA curve (see Figure 2). Now determine $\alpha$ and $\beta$ such that the *square* of the area between $V$ and $P$ (i.e., the hashed region in Figure 2) is minimized.

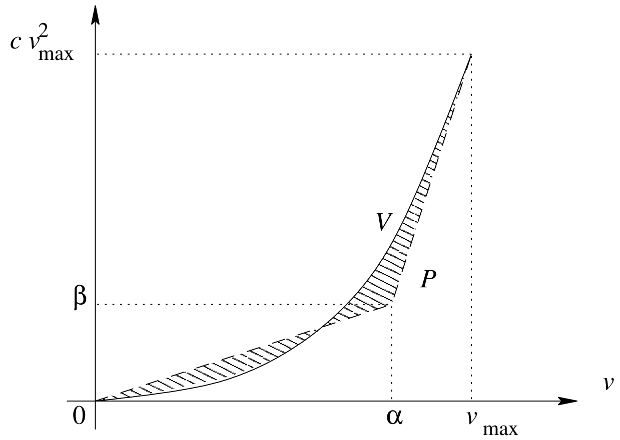


Figure 2: The quadratic function $V$ and its PWA approximation $P$.

**Step 3:** Now approximate the friction force using the PWA function $P$ instead of the quadratic function $V$. Compare the output of the resulting continuous-time PWA model with that of the original model for a sinusoidal throttle input, a white-noise input, and an arbitrary input of your choice. Can you explain where the differences — if any — come from?

**Step 4:** Now discretize the PWA model of the vehicle using a sample step $T$ with $T = 0.2\,\text{s}$ and a forward Euler rule for the discretization. Compare the discrete-time model with the two continuous-time models for the three input signals selected in Step 3.

**Step 5:** Transform the discrete-time PWA model of Step 4 into an MLD model.
As we are only considering speed tracking in this assignment and not distance keeping, the position of the car will not influence the performance (i.e., the control objective) in any way. Hence, for the transformation of the discrete-time PWA model into an MLD model, the only state variable that should be considered is the speed.

**Step 6:** Now we design an MPC controller for the MLD model using the implicit MPC approach. The performance $J$ should be a trade-off between the tracking $J_{\text{track}}$ (i.e., the difference between the speed and the reference speed communicated by the leading vehicle) and the input energy

$J_{\text{input}}$ (for group 1), the smoothness of the throttle signal (for group 2), and the smoothness of the derivative of the throttle signal (for group 3): $J = J_{\text{track}} + \lambda J_{\text{input}}$. So if $\tilde{u}(k) = [u^{\text{T}}(k) \dots u^{\text{T}}(k + N_{\text{p}} - 1)]^{\text{T}}$, then we have $J_{\text{input}}(k) = \|\tilde{u}(k)\|_2^2$ for group 1, $J_{\text{input}}(k) = \|\Delta\tilde{u}(k)\|_2^2$ for group 2, and $J_{\text{input}}(k) = \|\Delta^2\tilde{u}(k)\|_2^2$ for group 3.
Write a MATLAB file that computes the optimal MPC input sequence for a given sample step $k$ for values of $N_{\text{p}}$ and $N_{\text{c}}$ up to 6, and for arbitrary values of $\lambda$. Note that (a discretized version of the comfort constraint $-a_{\text{comf,max}} \leqslant a(t) \leqslant a_{\text{comf,max}}$ should also be taken into account!
Also note that due to the approximation made in Step 2, which is only valid for nonnegative speeds, we should also *explicitly* add the constraint $v(t) \geqslant 0$.
In order to solve this problem you will need an MIQP solver. You can use the `miqp` toolbox for this, which can be downloaded from
`http://www.dcsc.tudelft.nl/~sc4160/miqp.zip` or from
`http://www.dii.unisi.it/~hybrid/tools/miqp/`
If the `miqp` returns "strange" results (e.g., if it claims that the MIQP is infeasible, whereas it is clear that the problem *is* feasible), it might be necessary to play around with the options of `miqp`. Sometimes also using the "obsolete" solver `qp` instead of the "new" solver `quadprog` might help. As a last resort, the enumerate solver `miqp_enum`, which can be found at
`http://www.dcsc.tudelft.nl/~sc4160/miqp_enum.m` might be used.

**Step 7:** Write a MATLAB file to simulate the closed-loop behavior of the system (i.e., apply the receding horizon approach in which at each step the optimal MPC control input is recomputed and applied to the system) using

a) the discrete-time PWA model,
b) the original continuous-time model.

The MATLAB file should allow the discrete-time PWA model or the original continuous-time model to be used as the simulation model.

**Step 8:** Select an appropriate value $\lambda$ based on the nominal values of $J_{\text{track}}$ and $J_{\text{input}}$ (this might require some tuning and iteration).
Select two different combinations $(N_{\text{p},1}, N_{\text{c},1})$ and $(N_{\text{p},2}, N_{\text{c},2})$ of $N_{\text{p}}$ and $N_{\text{c}}$ with $N_{\text{p},i} \in \{3, 4, 5, 6\}$ and $1 < N_{\text{c},i} < N_{\text{p},i}$ and for each combination run your program for the discrete-time PWA model and the original continuous-time model for the time interval $[0, T_{\text{end}}]$ with $T_{\text{end}} = 10$, for $v(0) = 1.2\alpha$ where $\alpha$ is the value found in Step 2, and for the speed reference signal $v_{\text{ref}}$ which defined as follows (see also Figure 3):

$$v_{\text{ref}}(t) = \begin{cases} 0.8\alpha & \text{for } 0 \leqslant t \leqslant 1 \\ 1.2\alpha & \text{for } 1 < t \leqslant 3 \\ 1.2\alpha - 0.25(t-3) & \text{for } 3 < t \leqslant 5 \\ 0.7\alpha & \text{for } 5 < t \leqslant 6 \\ 0.7\alpha + 0.8(t-6) & \text{for } 6 < t \leqslant 7 \\ 0.9\alpha & \text{for } 7 < t \leqslant 10 \end{cases} .$$

Make a plot of the evolution of the controlled closed-loop system in the $(x,v)$ phase plane and of the evolution of $x$, $v$, $v_{\text{ref}}$, $v - v_{\text{ref}}$, $u$ and $\Delta u$ over time. Compare the obtained trajectories and discuss the differences, if any.

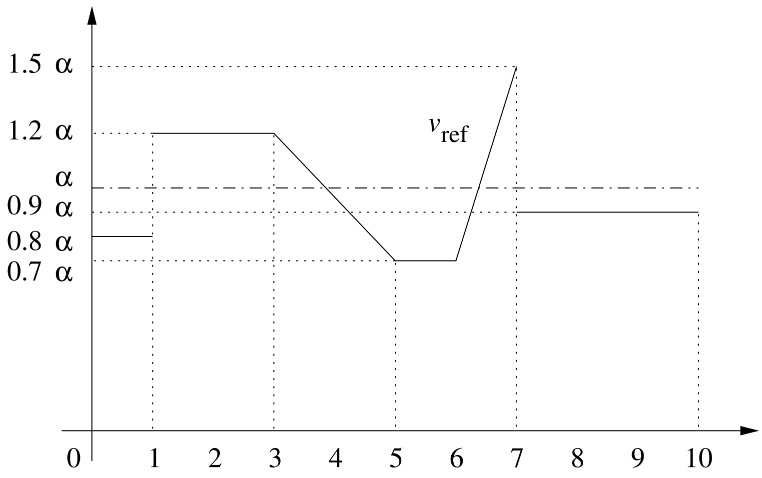


Figure 3: The reference speed signal to be used.

**Step 9:** You will have noticed that computing the optimal MPC input using the `miqp` program requires quite some computation time, especially for large $N_{\text{p}}$ and $N_{\text{c}}$. If this time is larger than the sampling time $T_{\text{c}}$ of the controller (in our case $T_{\text{c}} = T = 0.2$ s), then the (basic) on-line MPC optimization approach is not feasible. One of the possible solutions is then to use the explicit MPC approach in which for each possible current state $x(k)$ the optimal MPC input $u^*(k)$ is computed off-line and stored in a look-up table (cf. page 104 of the lecture notes and the references [19, 20, 23, 24, 32] of the lecture notes).

Apply the explicit MPC approach to the ACC example and repeat Steps 6–8 but now with explicit MPC instead of implicit MPC. In order to compute the explicit MPC solution you could use the Multi-Parametric Toolbox, which can be downloaded from

`http://control.ee.ethz.ch/~mpt/`

Compare the computation times required for each approach and for each of the combinations $(N_{\text{p},1}, N_{\text{c},1})$ and $(N_{\text{p},2}, N_{\text{c},2})$, and explain the results.

# A Possible additional tasks depending on the progress

- In Step 2 of the road map the approximation of the function $V$ by a PWA function was based on minimizing the area between the function and its PWA approximation. However, one could also minimize the error between the output of the original system model and the system model based on the PWA approximation over a given period for a given reference input.
  Determine $\alpha$ and $\beta$ using this alternative criterion. Do the values of $\alpha$ and $\beta$ coincide with the values obtained in Step 2? Explain.

- We could also consider a more detailed vehicle model with, e.g., static friction with a dead zone, a more involved dynamic friction model, or gear shifting (cf. the equations at the top of page 13 of the lecture notes) included, and/or consider additional constraints such as, e.g., a maximal speed.

- Now we examine the robustness of the MPC controller. We repeat the Steps 7 and 8 with a perturbed system, i.e., the "real" system equations will be given by the equations of Step 1 but with an extra noise term with some standard deviation $\sigma_e$. The prediction model used in the MPC controller is still the noise-free discrete-time PWA/MLD model.
  Write a MATLAB program to simulate the above perturbed system in combination with the MPC controller obtained in Step 6. For each of the two combinations $(N_{\mathrm{p},1}, N_{\mathrm{c},1})$ and $(N_{\mathrm{p},2}, N_{\mathrm{c},2})$ make a plot of the evolution of the controlled closed-loop system in the $(x,v)$ phase plane and of the evolution of $x$, $v$, $v_{\mathrm{ref}}$, $v - v_{\mathrm{ref}}$, $u$ and $\Delta u$ over time. Compare the obtained trajectories with the trajectories of Step 8 and discuss the differences, if any. If necessary, reduce or increase the noise level.

- Finally, we could also include the distance keeping component of ACC in our approach. This implies that we have to include a model of the leading vehicle and add a hard lower bound on the intervehicle distance and a penalty on the deviation from the reference intervehicle distance into the MPC problem.