Optimal and Model Predictive Control (OMPC)
Albert-Ludwigs-Universität Freiburg – Summer Term 2016

Guidelines for project

Prof. Dr. Moritz Diehl and Dimitris Kouzoupis

General Rules and Guidelines:

1. As already announced, the project can be done by one or two people together.
2. The main result is a written report in LaTeX submitted as a PDF file. Please use the *report* template with font size 12pt. The report should be between 4 pages (excluding figures) and 15 pages (including all content).
3. The report must be a new and self-written document and may not contain any copy of other text or figures. Not a single one. The report must be solely written by the author(s).
4. The report must include a short, interesting title, the name(s) of the author(s) and an abstract. The content should be clearly structured in sections. It should start with an introduction and conclude with a short summary and critical discussion of the results.
5. Figures and tables should have a short caption and be referenced in the text properly, e.g., "the results are shown in Fig. 1". Use the latex commands `\caption`, `\label` and `\ref`.
6. Plots must contain physical units and axis descriptions.
7. The report must cite all external sources as references at the end and other people's contributions must be acknowledged. Using other people's ideas and help is allowed, even encouraged. But not citing or acknowledging them properly is a crime.
8. Mathematical or physical variables shall consist of one letter only and be printed in italics. This is automatic in Latex, e.g., a_i as `$a_i$`. Physical units and sub- or superscripts that refer to words are in normal roman letters (use `mathrm` when in Latex mathmode, e.g. x_{initial} as `$x_{\mathrm{initial}}$` or $\frac{\mathrm{kg}}{\mathrm{m}^3}$ as `$\frac{\mathrm{kg}}{\mathrm{m}^3}$`. Write, e.g., $m = 5$ kg (and not $m = 5kg$ or m= 5kg).
9. On July 19, 2016, during the lecture, a short presentation of 15 minutes (with maximum 10 slides) shall be given by the author(s) to the teacher and the class. The slides can be based on material taken from the report and may contain additional content, e.g., videos, if required.
10. The project grade is based on the form and content of the report, the originality and quality of the results, the quality of the slides, the oral presentation and the answers to any questions.
11. The content of report should comply with the instructions on the **next page**.

Content:

1. Main goal of the project is to design an MPC controller for a system of your choice with a tracking or economic MPC objective.

2. The simulation environment can be either MATLAB or Python. You can use optimization tools of your choice, including `fmincon, quadprog, ACADO, casadi`, etc.

3. The report should address to certain extent the following critical issues, related to MPC controller design:

   - Model-plant mismatch (e.g., parameter uncertainty)
   - State and measurement noise
   - State estimation (e.g., Kalman filter)
   - Offset-free performance
   - Nominal closed-loop stability (e.g. dependence on choice of horizon legth, terminal cost, etc)
   - Complete description of system model
   - Complete description of MPC optimization problem (model, objective, constraints)
     - *Tip:* You can try out the newly developed package `optidef` for Latex.

4. In the core of your code, there should be a closed-loop simulation of your chosen plant, under the commands of your MPC controller. The simulation of the plant should be completely decoupled with the controller and it must comply with the following instructions:

   - An initialization function `pars = plant_init()`, should return all parameters of your model in the structure `pars` (once, at the beginning of your script).
   - A simulation function `x_next = plant_step(x, u, Ts, k, pars)`, with `Ts` the sampling time of the plant and k the current time-step, should simulate your model of choice for one time step. Within this function you can use for instance a MATLAB integrator (see `ode45`), an ACADO exported object or your own Runge–Kutta scheme to discretize the ODE that describes your model.
     - *Tip:* Do not forget to assign `x = x_next` at the end of your simulation loop.
   - An output function `y = plant_output(x, u, Ts, k, pars)` that should calculate the output of your model at the current time step (e.g., a subset of its states with additive measurement noise).
   - Your controller is **NOT** allowed to access the structure `pars` **NOR** the system state `x`. It can use of course the plant output y as well as a copy of the parameters structure, which is not necessarily identical to the one used in the plant, see below) .

5. To test your MPC controller, we (or another team) will make slight changes to your *plant*, to ensure that most issues described in 3 are properly addressed. Therefore, you are encouraged to play around with such changes yourself beforehand.

Important dates:

1. Please send before **June 21, 12:00**, an email to *dimitris.kouzoupis@imtek.uni-freiburg.de* that contains the title of your project, the corresponding authors and a short description of your concept.

2. Upload your plant functions described in 4 to Google drive before **July 12, 12:00**.

3. Submit your final report and accompanying code to Google drive before **July 17, 2016, 23:59**.