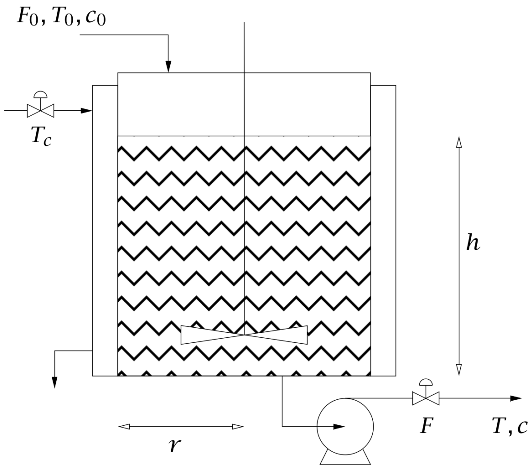
Summer School on Robust Model Predictive Control with CasADi
James B. Rawlings, Joel Andersson, Sergio Lucia, Moritz Diehl
University of Freiburg, September 15 to 19, 2025
`www.syscop.de/event/rmpc25`

Exercise 2: Numerical Optimal Control and NMPC with `CasADi` + `IPOPT` and `acados`

Jonathan Frey, Jingtao Xiong, Katrin Baumgärtner, Florian Messerer

In this exercise, we again consider the nonlinear continuous stirred-tank reactor (CSTR).[1]



An irreversible, first-order reaction $A \to B$ occurs in the liquid phase and the reactor temperature is regulated with external cooling. Mass and energy balances lead to the following nonlinear model:

$$\begin{aligned}
\dot{c} &= \frac{F_0(c_0 - c)}{\pi r^2 h} - k_0 \exp\left(-\frac{E}{RT}\right) c \\
\dot{T} &= \frac{F_0(T_0 - T)}{\pi r^2 h} - \frac{\Delta H}{\rho C_p} k_0 \exp\left(-\frac{E}{RT}\right) c + \frac{2U}{r \rho C_p}(T_c - T) \\
\dot{h} &= \frac{F_0 - F}{\pi r^2}
\end{aligned}$$

with states $x = (c, T, h)$ where c is the concentration of substance A, T is the reactor temperature and h is the height. The controls $u = (T_c, F)$ are the coolant liquid temperature T_c and the outlet flowrate F.

Description of closed-loop simulation environment

- The `main.py` file is prepared to simulate the closed-loop system with `acados` and plot the trajectories.
- The file `cstr_model.py` defines the model equations stated above with values for all the parameter values.
- The file `setup_acados_integrator.py` uses this model to generate an `AcadosSimSolver` instance which we use as our plant model.

[1] The example as well as the figure have been adopted from Example 1.11 in J. B. Rawlings, D. Q. Mayne, and M. M. Diehl. *Model Predictive Control: Theory, Computation, and Design.* Nob Hill, 2nd edition, 2017.

Exercises

Uncontrolled simulation:

- Simulate the system with a constant reference control input.

Exact NMPC with `acados`

- Set `with_nmpc = True` in `main.py` to create an `AcadosOcpSolver` and run it in a closed loop simulation.
  Note: check how the OCP is formulated using the `acados` interface using the `AcadosOcp` object. In comparison to the plain `CasADi` formulation from exercise 1, the multiple shooting discretization is done by the framework and does not need to be done by the user.

Exact NMPC with `CasADi` and `IPOPT` using the `AcadosCasadiOcpSolver`

`IPOPT` is a very reliable solver for nonlinear programming (NLP) formulations. It is the most widely used solver in `CasADi`, and known to be very robust. `IPOPT` tackles general sparse formulations. In contrast `acados` exploits the structure of optimal control problems.
The `AcadosCasadiOcpSolver` class allows one to formulate OCPs with the `acados` interface and use solvers interfaced with `CasADi`. Through the `AcadosCasadiOcpSolver`, one can interact with those solver, similar to the `AcadosOcpSolver`. This is very useful for prototyping, comparisons etc. See also Bonus exercise 2.

- Set `with_nmpc_ipopt = True` in `main.py` to also create an `AcadosCasadiOcpSolver`.
- Set `with_timevar_ref_nmpc = True` and `with_timevar_ref_nmpc_ipopt = True` to enable time varying references in a closed loop simulation. How is this done?
- Compare the trajectories without time-varying references to the one from the previous exercise, and also compare both trajectories (with and without time-varying references) along with the solvers' computation times.

Approximate and fast MPC

- Set `with_linear_mpc = True` in `main.py` to create an `AcadosOcpSolver` with a model linearized at the steady state.
- Set `with_nmpc_rti = True` in `main.py` to create an `AcadosOcpSolver` that uses the real-time iteration (RTI) algorithm. This algorithm performs one SQP iteration at each sampling time.
- Compare the resulting closed loop trajectories and the runtime of their controllers.

Bonus exercise 1: NMPC with model plant mismatch

- Note that the model parameter F_0 is implemented as a parameter in the `AcadosModel`.
- Task: Introduce a mismatch between the OCP model and the plant, by increasing F_0 in the OCP model by 5%. How well are the references tracked?

Bonus exercise 2: Differences between NLP solvers and warm starting

For this exercise the template `bonus_warm_start.py` is provided.

- Regard the first OCP of the closed-loop simulation. Solve it with the `AcadosCasadiOcpSolver` and `AcadosOcpSolver`.
- Compare the solutions. Can they be different if the problem is the same?

- Initialize the solver with the solution of the other. Is the IPOPT solution a solution for the `acados` solver?
- And vice versa?