Online Workshop SPP 2364

Nonlinear Optimization and Direct Optimal Control for Practitioners

Katrin Baumgärtner, Florian Messerer, Prof. Dr. Moritz Diehl

General information

This online workshop consists of two parts:

- 26.06.2023, 12:00-13:30 – Part 1: Nonlinear Optimization
- 03.07.2023, 12:00-13:30 – Part 2: Direct Optimal Control and Model Predictive Control

The aim of this workshop is to give you some hands on experience on methods and in particular software for optimal control. The workshop exercises are based on `python`, `CasADi`, and `acados`.

About `CasADi`

The open-source tool `CasADi` implements algorithmic differentiation on user-defined symbolic expressions and provides standardized interfaces to a variety of numerical routines: simulation and optimization, and solution of linear and nonlinear equations. A key feature of these interfaces is that every user-defined `CasADi` function passed to a numerical solver automatically provides the necessary derivatives to this solver, without any additional user input. Often, the result of the numerical solver itself can be interpreted as a differentiable `CasADi` function, such that derivatives up to any order can be generated without actually differentiating the source code of the solver. Thus, concatenated and recursive calls to numerical solvers are possible and still result in differentiable `CasADi` functions. `CasADi` is written in `C++`, but allows user input to be provided from either `C++`, `python`, `Octave` or `MATLAB`. One particularly powerful optimization solver interfaced to CasADi is `IPOPT`, which is automatically provided in the standard `CasADi` installation.

About `acados`

`acados` is a software package for the efficient solution of optimal control and estimation problems. It provides a collection of computationally efficient building blocks tailored to optimal control and estimation problems. Among others, it implements: modules for the integration of ordinary differential equations (ODE) and differential-algebraic equations (DAE), interfaces to state-of-the-art QP solvers like `HPIPM`, `qpOASES`, `DAQP`, `qpDUNES` and `OSQP`, condensing routines and nonlinear programming solvers based on the real-time iteration framework. The back-end of `acados` uses the high-performance linear algebra package `BLASFEO`, in order to boost computational efficiency for small to medium scale matrices typical of embedded optimization applications. `MATLAB`, `Octave` and `python` interfaces can be used to conveniently describe optimal control problems and generate self-contained C code that can be readily deployed on embedded platforms.

Prerequisites and installation instructions

In preparation of the workshop, please install `python`, `CasADi` and `acados` and make sure the minimal examples run properly.

- `CasADi`: `https://web.casadi.org/get/`
- `acados`: `https://docs.acados.org/installation/index.html`
- `acados python`: In addition `acados` itself, which is written in `c`, the installation of the `acados python` interface is required:
  `https://docs.acados.org/python_interface/index.html#installation`
- In order to test whether the installation was successful, please run the minimal example in the examples folder of the `acados` repository:
  `<acados root folder>/examples/acados_python/getting_started/minimal_example_ocp.py`

Online Workshop SPP 2364

Nonlinear Optimization and Direct Optimal Control for Practitioners
Part 1: Nonlinear Optimization

Katrin Baumgärtner, Florian Messerer, Prof. Dr. Moritz Diehl

Within a production process, five spheres s_i with $i = 1, \ldots, 5$ shall be cut out from a quadratic plate with egde size $a = 10$cm. Three of those spheres shall be of radius R and two of radius $2R$. The objective is to maximize the radius R. The center of each sphere s_i can be expressed in Cartesian coordinates (x_i, y_i) on the plate, and are to be optimized in addition to the radius R. The spheres may not lie outside of the plate or overlap each other. To ensure this, the minimum distance between the centers of all spheres from each other as well as the edges of the plate must enter the constraints of the optimization problem. A depiction of a possible but suboptimal solution with $R = 1$ is given in Figure 1.

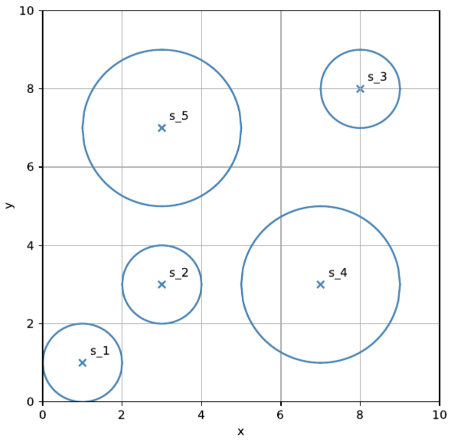

Figure 1: Graphical depiction of a possible, but suboptimal solution with $R = 1$.

The problem can be formulated as a nonlinear program in `CasADi` and solved using `IPOPT`, where the following sets of constraints must enters the optimization problem:

1. The radii of two of the spheres must be twice a big as the radii of the three other spheres, and must therefore fulfill the condition

$$r_i = R, \quad i = 1, \ldots, 3, \tag{1}$$
$$r_j = 2R, \quad j = 4, 5. \tag{2}$$
$$\tag{3}$$

2. The minimum distance of a sphere's x-coordinate from the left egde and the right edge of the plate must be greater or equal than its radius r_i, the same must hold for the distance of the y-coordinate from the top edge and bottom edge of the plate.

$$x_i - r_i \geq 0, \quad i = 1, \ldots, 5, \tag{4}$$
$$x_i + r_i \leq a, \quad i = 1, \ldots, 5, \tag{5}$$
$$y_i - r_i \geq 0, \quad i = 1, \ldots, 5, \tag{6}$$
$$y_i + r_i \leq a, \quad i = 1, \ldots, 5, \tag{7}$$
$$\tag{8}$$

3. The distance of two sphere's centers must be greater or equal to the sum of both sphere's radii, which can be expressed simply by using the Pythagorean theorem as

$$(x_i - x_j)^2 + (y_i - y_j)^2 - (r_i + r_j)^2 \geq 0, \quad i = 1, \ldots, 4, \quad j = i + 1, \ldots, 5. \tag{9}$$

Tasks

- Complete the template provided for this task with the information given above and run the script. On success, you should see a plot that depict the positioning of the spheres on the plate, and they should neither interlap nor lie outside the plate. How big is R if you use the initial guesses for the circles coordinates that are already contained in the template?

- Looking at the plot, could you think of a distribution for the spheres that might lead to even bigger values for R? Try setting different initial guesses for the spheres' center coordinates, and write down your best solution for R.

Online Workshop SPP 2364

Nonlinear Optimization and Direct Optimal Control for Practitioners Part 3: Direct Optimal Control and Model Predictive Control

Katrin Baumgärtner, Florian Messerer, Prof. Dr. Moritz Diehl

In this exercise, we consider the nonlinear continuous stirred-tank reactor (CSTR). For more information on the modelling of the system please check this lecture (slide 34 and following).

- Go through the file `cstr_model.py` to see how a dynamic model is formulated with `acados`.
- Go through the file `main.py` to see how a closed loop simulation can be implemented with `acados`.
  - To simulate the plant, we use an `acados` integrator, i.e. `AcadosSimSolver`
  - The controller is implemented using the `AcadosOcpSolver` class
- Fill in the gaps in the template regarding the cost and constraints.
- Run the main file, the result should look similar to Figure 2.

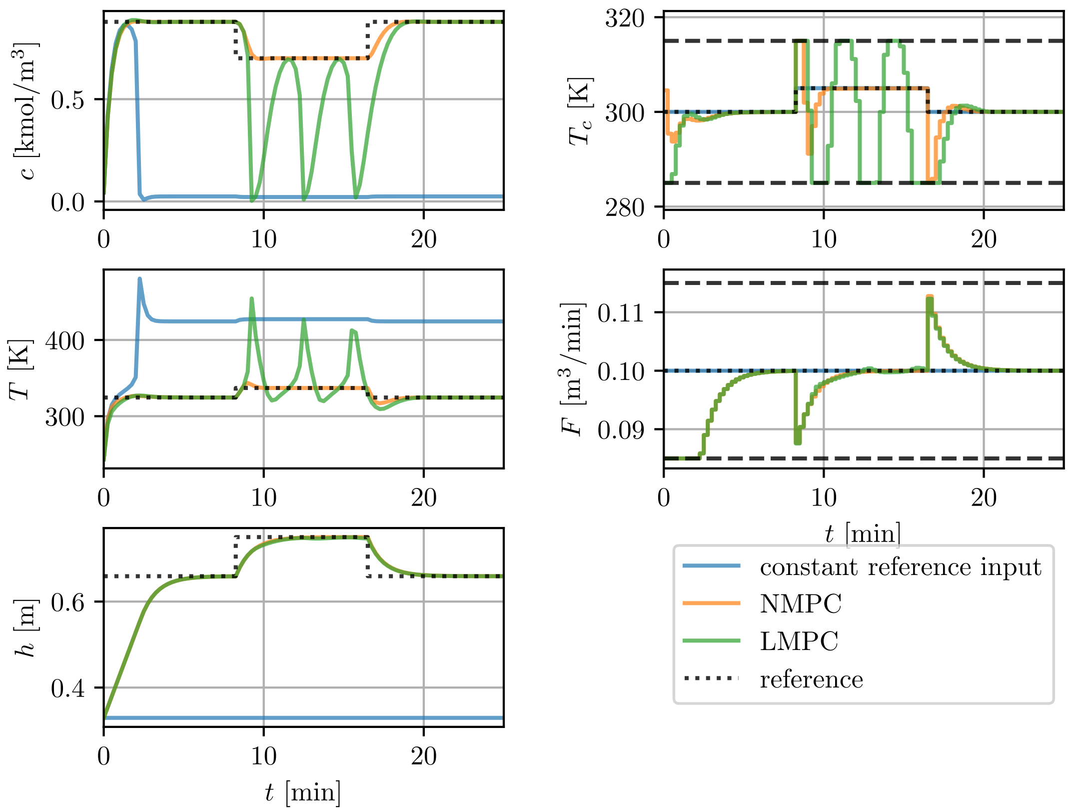

Figure 2: Linear MPC (LMPC) and nonlinear MPC (NMPC) for the CSTR example using `acados`.