FrontSeat Summer School on *Optimization-based Control Systems: Theory & Applications*

Exercise 1: Nonlinear Optimization and Optimal Control Problem with `CasADi` and `IPOPT`

Jonathan Frey, Marco Vaccari

Note: This exercise PDF and all accompanying Python template code can be downloaded or cloned from the course repository https://github.com/CPCLAB-UNIPI/FrontSeatSummerSchool (please follow the instructions in README.md).

Exercise 1.1

The aim of this exercise is to formulate and solve a constrained nonlinear optimization problem with 11 variables in `CasADi`, described in the following.

Within a production process, five spheres s_i with $i = 1, \dots, 5$ shall be cut out from a quadratic plate with edge 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.

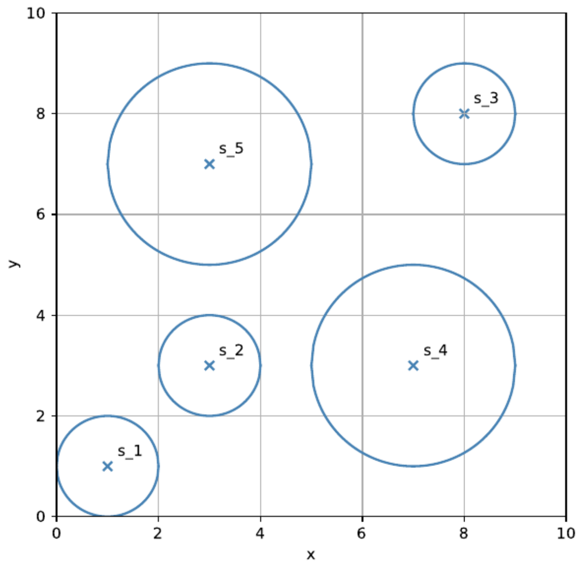


Figure 1: Graphical depiction of a possible, but suboptimal solution with $R = 1$.

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

(a) 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

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

(b) The minimum distance of the x-coordinate of any sphere from the left edge 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.

$$x_i - r_i \geq 0, \quad i = 1, \ldots, 5, \tag{3}$$
$$x_i + r_i \leq a, \quad i = 1, \ldots, 5, \tag{4}$$
$$y_i - r_i \geq 0, \quad i = 1, \ldots, 5, \tag{5}$$
$$y_i + r_i \leq a, \quad i = 1, \ldots, 5, \tag{6}$$

(c) The distance of the centers of two spheres must be greater or equal to the sum of both spheres' radii, which can be expressed simply by using the Pythagorean theorem as

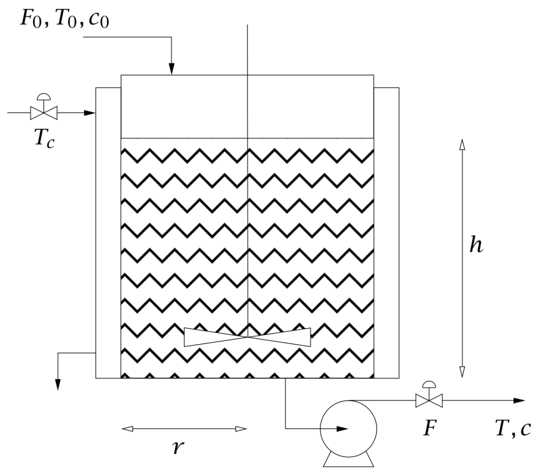
$$(x_i - x_j)^2 + (y_i - y_j)^2 - (r_i + r_j)^2 \geq 0, \quad i = 1, \ldots, 4, \quad j = i + 1, \ldots, 5. \tag{7}$$

Tasks

1. If you have never before used `CasADi`, take some moments reading its documentation at https://web.casadi.org/docs/
2. Complete the template provided for this task with the information given above and run the script. On success, you should see a plot that depicts the positioning of the spheres on the plate, and they should neither overlap 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?
3. 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.
4. Check the output of the NLP solver `IPOPT` to assess if the NLP was solved successfully.

Exercise 1.2

As a second example, we solve a first optimal control problem (OCP) using the single shooting formulation. To this end, we consider a continuous-stirred tank reactor (CSTR).[1]



[1] The example as well as the figure have been adopted from Example 1.11 in Rawlings, J.B., Mayne, D.Q. and Diehl, M., 2017. *Model predictive control: theory, computation, and design (Vol. 2)*. Madison, WI: Nob Hill Publishing.

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.

Using a sampling time of 1 min, a linearized discrete state space model is obtained and, assuming that all the states are measured, the state space variables are

$$
x \triangleq \begin{bmatrix} c - c_s \\ T - T_s \\ h - h_s \end{bmatrix} \qquad u \triangleq \begin{bmatrix} T_c - T_s \\ F - F_s \end{bmatrix}
$$

The corresponding linear model is

$$
x(k+1) = A_p x(k) + B_p u(k) \tag{8}
$$

in which

$$
A_p = \begin{bmatrix} 0.2681 & -0.00338 & -0.00728 \\ 9.703 & 0.3279 & -25.44 \\ 0 & 0 & 1 \end{bmatrix} \qquad B_p = \begin{bmatrix} -0.00537 & 0.1655 \\ 1.297 & 97.91 \\ 0 & -6.637 \end{bmatrix} \tag{9}
$$

The OCP is given by the following NLP:

$$
\begin{aligned}
\min_{u} \quad & \sum_{k=0}^{N-1} l_k\left(x_k(\bar{x}_0, u), u\right) + l_N(x_N(\bar{x}_0, u)) && \text{(10a)} \\
\text{s.t.} \quad & 0 \geq h(u_k), \ k = 0, \dots, N-1 && \text{(10b)}
\end{aligned}
$$

with control trajectory $u = (u_0, \dots, u_{N-1})$. The initial state $\bar{x}_0$ enters the problem as a parameter. The constraint h encode lower and upper bounds on the control inputs. The control aim is to follow a given reference trajectory, x^{ref}, u^{ref}, which is implemented via the cost

$$
l(x_k, u_k) = \frac{1}{2}\left\| x_k - x^{\text{ref}} \right\|_Q^2 + \frac{1}{2}\left\| u_k - u^{\text{ref}} \right\|_R^2, \qquad l_N(x_N) = \frac{1}{2}\| x_k - x^{\text{ref}} \|_P^2. \tag{11}
$$

Tasks

1. Fill in the template adding CasADi expressions for the linear model, stage and terminal cost.
2. The lower input bound serves as tentative for initial trajectory to be supplied to the OCP for the iteration. Does the performance of the solver change if you modify u_{min}?
3. Check how the OCP is created in `OCPsolver`. Specifically, how is the trajectory of states created as an expression of $\bar{x}_0, u$?