AWESCO Winter School on Numerical Optimal Control with DAE - University of Freiburg

Exercise 1: Introduction to CasADi and Quadratic Programming

Andrea Zanelli, Joel Andersson, Joris Gillis, Sebastien Gros, Moritz Diehl

CasADi is an open-source software tool for solving optimization problems in general and optimal control problems in particular. In its most typical usage, it leaves it to the user to formulate the problem as a standard form constrained optimization problem of the form:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & \underline{x} \le x \le \overline{x} \\ & \underline{g} \le g(x) \le \overline{g}, \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^{n_x}$ is the decision variable, $f : \mathbb{R}^{n_x} \to \mathbb{R}$ is the objective function, and $g : \mathbb{R}^{n_x} \to \mathbb{R}^{n_g}$ is the constraint function. For equality constraints, the upper and lower bounds are equal.

In this exercise, f is a convex quadratic function and g is a linear function, in which case we refer to problem (1) as a (convex) quadratic program (QP). To solve a QP with CasADi, start by creating a struct containing expressions for x, f and g:

```
x = SX.sym('x',n)
f = (some expression of x)
g = (some expression of x)
prob = {'x':x,'f':f,'g':g}
```

This symbolic representation of the problem is then used to construct a QP solver as follows:

```
solver = qpsol('solver','qpoases',prob)
```

where the arguments are, respectively, the *display name* of the solver `s`, the solver *plugin* – here the open-source QP solver qpOASES – and and the above symbolic problem formulation. A set of algorithmic options can be passed as an optional forth argument. Optimization solvers are *functions* in CasADi that are evaluated to get the solution:

```
arg={'x0':x0,'lbx':lbx,'ubx':ubx,'lbg':lbg,'ubg':ubg}
res = solver(arg)
```

Where `lbx`, `ubx`, `lbg` and `ubg` are the bounds of x and $g(x)$ and `x0` is an initial guess for x (less important for convex QPs, since the solution is unique).

Equilibrium position for a hanging chain

We want to model a chain attached to two supports and hanging in between. Let us discretize it with N mass points connected by $N-1$ springs. Each mass i has position $(y_i, z_i),\ i = 1, \ldots, N$. The equilibrium point of the system minimises the potential energy. The potential energy of each spring is

$$V_{\text{el}}^i = \frac{1}{2} D_i \left((y_i - y_{i+1})^2 + (z_i - z_{i+1})^2 \right).$$

The gravitational potential energy of each mass is

$$V_{\text{g}}^i = m_i\, g_0\, z_i.$$

The total potential energy is thus given by:

$$V_{\text{chain}}(y, z) = \frac{1}{2} \sum_{i=1}^{N-1} D_i \left((y_i - y_{i+1})^2 + (z_i - z_{i+1})^2 \right) + g_0 \sum_{i=1}^{N} m_i\, z_i, \tag{2}$$

where $y = [y_1, \cdots, y_N]^T$ and $z = [z_1, \cdots, z_N]^T$.

We wish to solve

$$\underset{y,z}{\text{minimize}} \qquad V_{\text{chain}}(y, z) \tag{3}$$

subject to constraints modeling the ground, to be introduced below.

Tasks:

1.1 Go to the CasADi website and locate the user guide. Make sure the version of the user guide matches the version of CasADi used in the course (3.0.0-rc2). Then, with a Python or MATLAB interpreter in front of you, read Chapter 3 as well as Sections 4.1-4.3 in Chapter 4 of the user guide.

1.2 From the course website, you will find solution scripts for Python and MATLAB that solve the unconstrained problem using $N = 40,\ m_i = 40/N$ kg, $D_i = 70N$ N/m, $g_0 = 9.81\ \text{m/s}^2$ with the first and last mass point fixed to $(-2, 1)$ and $(2, 1)$, respectively. Go through the script and make sure you understand the steps.

1.3 Introduce ground constraints: $z_i \geq 0.5$ and $z_i - 0.1\, y_i \geq 0.5$, for $i = 2, \cdots, N-2$. Solve your QP again, plot the result and compare it with the previous one.

1.4 **Extra**: What would happen if you add instead of the linear ground constraints, the nonlinear ground constraints $z_i \geq 0.5 + 0.1\, y_i^2$ to your problem? The resulting problem is no longer a QP, but is it convex?

1.5 **Extra**: What would happen if you would add instead the nonlinear ground constraints $z_i \geq 0.5 - 0.1\, y_i^2$ to your problem? Is the problem convex?