Model Predictive Control - EL2700

Computer Exercise 1: CasADi for Linear and Quadratic Programs

2023

Automatic Control
School of Electrical Engineering and Computer Science
Kungliga Tekniska Högskolan

Operations with CasADi and Numpy

Dear student, welcome to the Computer Exercise lab 1! Before starting with tasks 1 and 2, we will do some warm-up exercises. The 5 warm-up questions are available on the file `operations.py`, and they cover most of the basics that you will need today.

Start by solving the questions in `operations.py` before proceeding. If you don't reach Question 5 by 15h40, please proceed to task 1 which covers the same topic.

As usual, a `install_deps.py` script is provided, which you can run with Python 3.6 or above to make sure that you have all the required libraries.

Task 1: A simple Linear Programming problem

In this first task we will do a warm-up exercise with Linear Programming. In linear programming, both the cost function and constraints are linear with respect to the optimization variable, taking the form

$$\min_{x} \quad g^T x \tag{1}$$

$$\text{subject to:} \quad x \geq x_{ub} \tag{2}$$

$$Ax \leq b \tag{3}$$

Note, however, that the solver we use is more generic (capable of solving Quadratic Programming problems), and therefore we might need to adjust our implementation to the form that the solver accepts. In our case, we will use CasADi's `conic` function, which solves optimization problems of the form

$$\min_{x} \quad \frac{1}{2}x^T H x + g^T x \tag{4}$$

$$\text{subject to:} \quad x_{lb} \leq x \leq x_{ub} \tag{5}$$

$$a_{lb} \leq Ax \leq a_{ub} \tag{6}$$

which is implemented as

```
qp = {'h': H.sparsity(), 'a': A.sparsity() }
S = conic('S','qpoases',qp) # 'qpoases' -> solver
r = S(h=H, g=g, a=A, lbx=x_lb, ubx=x_ub, lba=a_lb, uba=a_ub)
x_opt = r['x']
```

Taking this into account, lets take a look at the problem.

Airline Sitting Management MPC Airline (MPCA) is planning how many tickets should be sold from first class and second class to maximize profit. To be profitable, MPCA needs to sell at least 20 tickets from first class and 35 tickets from second class. Knowing that MPCA has a profit of 2000 SEK for second class tickets and 1500 SEK for first class tickets, and its airplanes can take a maximum of 130 passengers, how many tickets of each kind should be sold to maximize the profit?

Task 2: Hanging Chain[1]

In this problem we will formulate our first QP problem and solve it using CasADi's `conic` class. This problem deals with the hanging chain optimization problem. Our goal is to obtain the final position of each link of the chain by solving an optimization problem.

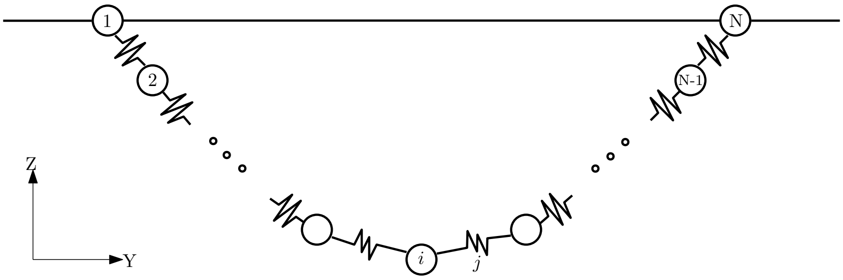

Figure 1: Hanging chain setup. Note that mass links 1 and N are fixed.

Consider the chain in Figure 1 . The chain is composed by N mass links coupled together via $N-1$ springs. Let the position of each mass element be denoted by $x_i, i = 1, \ldots, N$. Moreover, let m_i be the mass of each the ith element, k_i the spring constant of spring i, and g_c the gravity acceleration. Note that the mass elements 1 and N are fixed.

Q1: What is the kinetic and potential energy for each mass-spring system pair?
Q2: What is the total kinetic and potential energy for the whole chain?

To find out the final shape of the hanging chain, we will formulate a Quadratic Programming problem. To this end we will minimize the chain potential energy, while keeping in mind that the chain hangs by two fixed points.

First, lets take another look at the QP formulation in CasADi:

$$\min_{x} \quad \frac{1}{2}x^T H x + g^T x \tag{7}$$

$$\text{subject to:} \quad x_{lb} \leq x \leq x_{ub} \tag{8}$$

$$a_{lb} \leq Ax \leq a_{ub} \tag{9}$$

Q3: How would you translate the problem we want to solve into the QP problem in CasADi's framework? Consider at first $N = 3$ to solve it "by hand".
Q4: Consider. $m_i = 4[kg], k_i = 1000[N]$, $N = 10$, $g_c = 9.81[m/s^2]$, $x_1 = \begin{bmatrix}-2 & 1\end{bmatrix}$ and $x_N = \begin{bmatrix}2 & 1\end{bmatrix}$. Calculate the QP problem matrices using Numpy and CasADi variables. **Notes:**

- you should implement it as a function of N, such that it would be easy to add or remove links;

At this point we are ready to solve our problem. To that end, we will use CasADi's `conic` interface, which takes the following form

```
qp = {'h': H.sparsity(), 'a': A.sparsity() }
S = conic('S','qpoases',qp)
r = S(h=H, g=g, a=A, lbx=x_lb, ubx=x_ub, lba=a_lb, uba=a_ub)
x_opt = r['x']
```

Q6: We would like our chain to stop at the ground, where the height is $0.5[m]$. How should we do it?
Q7: Now we would like to consider a ground plane that is tilted, for instance $z - 0.1y >= 0.5$. How should we introduce this constraint?
Extra: Now, change the masses of certain links. What happens to the chain position?

Good Luck!

[1] Adapted from Joel Andersson, Joris Gillis and Moritz Diehl's "Equilibrium position for a hanging chain"