Summer School on Direct Methods for Optimal Control of Nonsmooth Systems
Albert-Ludwigs-Universität Freiburg – September 11 - September 15, 2023

Exercise 1: Introduction to CasADi and direct optimal control

Prof. Dr. Moritz Diehl, Armin Nurkanović, Jonathan Frey, Florian Messerer, Anton Pozharskiy

The goal of this exercise is to get familiar with the open-source tool `CasADi` and to solve a first optimal control problem via direct collocation. Templates are available for Matlab and Python.

About `CasADi`

`CasADi` is an open-source tool for nonlinear optimization and algorithmic differentiation.
In preparation of the summer school, please install `CasADi`: `https://web.casadi.org/get/`
Make sure you can use it succesfully, running a small test:

```
% Matlab
import casadi.*
x = SX.sym('x')
disp(jacobian(sin(x),x))
```

```
# Python
from casadi import *
x = SX.sym("x")
print(jacobian(sin(x),x))
```

If something is unclear, consult the `CasADi` documentation at `https://web.casadi.org/docs/`.
For Python users: The solution templates borrow some functionality from `nosnoc`, please clone `https://github.com/FreyJo/nosnoc_py/` and follow the installation instructions.

Pendulum on cart with friction

Consider a model of a pendulum on a cart with friction, with state $x = (p, \theta, v, \omega)$, where p describes the position of the cart, θ the angle of the pendulum mounted on the cart, v the velocity of the cart and ω the angular velocity of the pendulum. The control action u is a force which accelerates the cart.
The differential equations corresponding to this system can be derived using Lagrange mechanics, where $q = (p, \theta)$ describe the generalized coordinates and $\dot{q} = (v, \omega)$ are its derivative. The system has the following parameters, which are assumed to be fixed, the length of the pendulum $l = 1$, the mass of the cart $m_1 = 1$, the mass at the end of the pendulum $m_2 = 0.1$ and the gravitational constant $g = 9.81$.
The differential equations can be derived as follows. First, the derivatives of the generalized coordinates are simply $\dot{p} = v$ and $\dot{\theta} = \omega$. Second, the dynamics of the generalized velocities are

$$\ddot{q} = \begin{pmatrix} \dot{v} \\ \dot{\omega} \end{pmatrix} = M(q)^{-1} f_{\mathrm{all}}(q, \dot{q}, u) \tag{1}$$

where the inertia matrix $M(\cdot)$ of the system is given as

$$M(q) = \begin{pmatrix} m_1 + m_2 & m_2 l \cos(\theta) \\ m_2 l \cos(\theta) & m_2 l^2 \end{pmatrix} \tag{2}$$

and $f_{\mathrm{all}}(\cdot)$ gathers the gravity, control, Coriolis and friction forces, as follows:

$$f_{\mathrm{all}}(\cdot) = \begin{pmatrix} 0 \\ -m_2 g l \sin(\theta) \end{pmatrix} + \begin{pmatrix} u \\ 0 \end{pmatrix} - \begin{pmatrix} 0 & -m_2 l \cos(\theta) \\ 0 & 0 \end{pmatrix} \dot{q} + \begin{pmatrix} -F_{\mathrm{Friction}} \mathrm{sign}(v) \\ 0 \end{pmatrix}.$$

Here, F_{Friction} is a fixed parameter which we will change in this exercise to use the model, with or without friction. We will use values $F_{\mathrm{Friction}} \in \{0, 2\}$. For the following equations, we denote the explicit ODE as $\dot{x} = f_{\mathrm{ODE}}(x, u)$.

1. **Modelling and simulation with CasADi** The goal is to simulate the cart pole model with different settings.

   (a) Simulate the system without friction. Fill in the missing part in the `CasADi` model description.

   (b) Extend the model to include a smoothed friction model, by replacing $\text{sign}(z)$ with $\tanh(\frac{z}{\sigma})$. Model the smoothing parameter σ as a `CasADi` parameter, and run the simulation with $\sigma = 1.0$.

2. **Optimal Control with Direct collocation**
   Collocation methods belong to the class of implicit Runge-Kutta methods for solving initial value problems. To discretize an optimal control problem with *direct collocation* we replace the continuous-time dynamics

$$\dot{x}(t) = f_{\text{ODE}}(x(t), u(t)),$$

by the discrete-time collocation equations. Thereby, we split the control horizon $[0, T]$ into N control intervals with a uniform time discretization grid $t_n = nh$, $n = 0, \ldots, N$, where h is the step size and the corresponding state values are $x_n = x(t_n)$. For the control discretization we use $u(t) = u_n, t \in [t_n, t_{n+1}]$, $n = 1, \ldots, N$. On every control interval the state trajectory is approximated by polynomials $q_n(t)$, $n = 1, \ldots, N$. Note that in every control interval we may have multiple integration steps, but for simplicity we take only one integration step per control interval.

Next, on each control interval $[t_n, t_{n+1}]$, we compute the coefficients of these polynomials to ensure that the ODE is exactly satisfied at the *collocation points* $t_{n,i} = t_n + hc_i, i = 1, \ldots, n_\text{s}$, where, n_s is the number of stages. The choice of the points $0 = c_0 < c_1 < \ldots < c_{n_\text{s}} \leq 1$ determines the accuracy and stability properties of the resulting method. Popular choices for c_i are the Radau IIA or Gauss-Legendre points. In the lecture, we found the interpolating polynomial $\dot{q}_n(t)$ through the state derivatives $k_{n,1}, \ldots, k_{n,n_\text{s}}$. Here, we implement a collocation method by finding the interpolating polynomial $q_n(t)$ through the initial value x_n and *state values* $x_{n,1}, \ldots, x_{n,n_\text{s}}$ at the stage points.

For the implementation, we make use of the Lagrangian polynomial basis. Using these time points, we define a basis for our polynomials:

$$\ell_i(\tau) = \prod_{j=0,\, i \neq j}^{n_\text{s}} \frac{\tau - c_j}{c_i - c_j}, \qquad\qquad i = 0, \ldots, n_\text{s}. \tag{3}$$

Note that, in contrast to the lecture, the counter starts from $i = 0$, as we include the point $c_0 = 0$, since we interpolate through x_n.

We approximate the state trajectory on $[t_n, t_{n+1}]$ by a linear combination of the basis functions:

$$q_n(t) = \sum_{j=0}^{n_\text{s}} \ell_j\left(\frac{t - t_n}{h}\right) x_{n,j}. \tag{4}$$

By differentiation, we obtain an approximation of the time derivative at each collocation point:

$$\dot{q}_n(t_{n,i}) = \frac{1}{h} \sum_{j=0}^{n_\text{s}} \dot{\ell}_j(c_i)\, x_{n,j} := \frac{1}{h} \sum_{j=0}^{n_\text{s}} {\color{red}C_{j,i}}\, x_{n,j}, \quad i = 0, \ldots, n_\text{s}. \tag{5}$$

The expression for the state at the end of an interval reads as:

$$x_{n+1} = \sum_{i=0}^{n_\text{s}} \ell_i(1)\, x_{n,i} := \sum_{i=0}^{n_\text{s}} {\color{red}D_i}\, x_{k,i} \tag{6}$$

Moreover, using the obtained approximation $q_n(t)$ we can integrate the *stage cost*

$$\int_0^T L(x(t), u(t))\mathrm{d}t,$$

over every control interval and obtain a formula for *quadratures*:

$$\int_{t_n}^{t_{n+1}} \sum_{j=0}^{n_\mathrm{s}} \ell_j \left(\frac{t-t_n}{h}\right) L(x_{n,j}, u_n)\, \mathrm{d}t = h \sum_{j=0}^{n_\mathrm{s}} \int_0^1 \ell_j(t)\, \mathrm{d}t\, L(x_{n,j}, u_n) := h \sum_{j=0}^{n_\mathrm{s}} B_j\, L(x_{n,j}, u_n). \quad (7)$$

Tasks:

(a) Using the derived formulae above, write down on paper the collocation equations for a single integration interval.

(b) We want to solve the continuous time optimal control problem (OCP)

$$\begin{aligned} \underset{x(\cdot),\, u(\cdot)}{\text{minimize}} \quad & \int_0^T f_q(x(\cdot), u(\cdot)) + f_{q,T}(x(T)) \\ \text{subject to} \quad & x(0) = \bar{x}_0, \\ & \dot{x}(t) = f_\mathrm{ODE}(x(t), u(t)),\, t \in [0, T), \\ & l_\mathrm{bu} \leq u(t) \leq u_\mathrm{bu}, \qquad t \in [0, T], \\ & l_\mathrm{bx} \leq x(t) \leq u_\mathrm{bx}, \qquad t \in [0, T] \end{aligned}$$

where the initial state is $\bar{x}_0 = [1, 0, 0, 0]$, the control bounds are $l_\mathrm{bx} = -30,\ u_\mathrm{bu} = 30$, the state bounds are $l_\mathrm{bx} = [-5, -\infty, -\infty, -\infty], u_\mathrm{bx} = [5, \infty, \infty, \infty]$ and the objective function terms are:

$$\begin{aligned} f_q(x, u) &= (x - x_\mathrm{ref})^\top Q (x - x_\mathrm{ref}) + u^\top R u \\ f_{q,T}(x, u) &= (x - x_\mathrm{ref})^\top Q_\mathrm{terminal} (x - x_\mathrm{ref}) \end{aligned}$$

with $Q = \mathrm{diag}(10, 100, 1, 1), R = 1, Q_\mathrm{terminal} = \mathrm{diag}(500, 100, 10, 10)$, and reference state $x_\mathrm{ref} = (0, \pi, 0, 0)$ describing the unstable equilibrium point.

(c) Define the ingredients listed above to specify the OCP and solve it with direct collocation in `CasADi`. Note: in Matlab, the OCP formulation is part of the model.
Complete the implementation of direct collocation to discretize the OCP problem. The matrices B, C and D are already provided, use your solution from (a).

(d) **Bonus**: Form the Jacobian of the constraints and inspect the sparsity pattern using the `spy` command. The following is a hint:

```
% MATLAB
J = jacobian(vertcat(g{:}), vertcat(w{:}));
spy(sparse(DM.ones(J.sparsity())));
```

```
# Python
J = jacobian(vertcat(g), vertcat(w))
import matplotlib.pylab as plt
plt.spy(DM.ones(J.sparsity()).sparse())
```

Repeat the same for the Hessian of the Lagrangian function $\mathcal{L}(w, \lambda) = f_\mathrm{objective}(w) + \lambda^\mathrm{T} g(w)$.