

Exercise 6: Optimal Control Formulations

Prof. Dr. Moritz Diehl, Andrea Zanelli, Dimitris Kouzoupis

In this exercise we will have a look at the special structures of the optimal control formulations discussed in the class. To this end, we will consider the following optimal control problem:

$$\begin{aligned} \min_{x, u} \quad & \sum_{i=0}^{N-1} (\|x_i\|_2^2 + 2u_i^2) + 10 \|x_N\|_2^2 \\ \text{s.t.} \quad & x_0 = \bar{x}_0 \\ & x_{i+1} = f(x_i, u_i), \quad i = 0, \dots, N-1, \end{aligned} \tag{1}$$

where f describes the discretized dynamics obtained by applying the explicit RK4 integration scheme to the following continuous-time dynamics:

$$\begin{aligned} \dot{\phi} &= \omega \\ \dot{\omega} &= \sin(\phi) + \tau, \end{aligned} \tag{2}$$

where ϕ is the angle describing the orientation of an inverted pendulum, ω is its angular velocity and τ is the input torque. The goal is to swing up the pendulum starting from $\phi = \pi$. We will consider different reformulations of problem (1) and analyze their structure.

1. **Sequential approach:** The state variables in (1) can be eliminated by means of a forward simulation such that a more compact optimization problem is obtained:

$$\min_u \Phi(u). \tag{3}$$

Using CasADi, formulate $\Phi(u)$ and set up problem (3) using an RK4 integrator with 10 intermediate steps and a step size $h = 0.1$. Use $N = 50$ nodes (horizon $T = 5$) and $\bar{x}_0 = [\pi, 0]^T$ as initial state. Solve the formulated problem using IPOPT and plot state and control trajectories in two separate plots. Use $u_i = 0.1$, $i = 0, \dots, N-1$ as initial guess. Before solving the problem, compute the Hessian of $\Phi(u)$ and plot its sparsity pattern using the MATLAB command `spy`.

(2 points)

2. **Simultaneous approach:** If both states and controls are kept as decision variables, a larger problem is obtained with a special structure that can be exploited by tailored algorithms.

- (a) Using CasADi, formulate problem (1) without eliminating the state variables, solve it with IPOPT and plot the state and control trajectories. Make sure that you obtain the same result as with the sequential approach. Eliminate the initial state, by fixing it to $x_0 = \bar{x}_0$ and use $u_i = 0.1$, $i = 0, \dots, N-1$ and $x_i = [0.1, 0.1]^T$, $i = 1, \dots, N$ as initial guess. Before solving the problem compute the Hessian of the Lagrangian and plot its sparsity pattern using the MATLAB command `spy`. In order to highlight the block-banded structure of the Hessian, reorder the variables as $[u_0, \lambda_1, x_1, u_1, \lambda_2, \dots]$. For this last part, you can initialize the equality multipliers to any arbitrary value since we are only interested in the sparsity pattern here.

(3 points)

- (b) Implement now your own full-step Newton method and use it to solve problem (1). Initialize both primal and dual variables to 10 element-wise. Terminate the algorithm when the 2-norm of the right-hand-side is less than 10^{-6} . Plot the resulting trajectories and make sure you obtain the same results as with the previous implementations. *Hint: you already have the Lagrangian of the problem from point a) which can be used to compute the right-hand-side of the linear system associated with the Newton steps. Analogously, the matrix coefficient of the linear system can be obtained by computing the Hessian of the Lagrangian.*

(3 points)

- (c) Make a copy of your script from point a) and modify the formulation in order to include box constraints on the inputs:

$$-1 \leq u_i \leq 1, \forall i = 0, \dots, N-1. \quad (4)$$

Solve the modified problem with IPOPT using the same initial guess as before and plot the obtained trajectories.

(2 points)

- (d) **[Bonus]** Make a copy of your script from point a) and modify the formulation this time by including logarithmic barrier terms in the cost that approximate the constraints introduced in c):

$$\sum_{i=0}^{N-1} (\|x_i\|_2^2 + 2u_i^2 - \rho (\log(-u_i + 1) + \log(u_i + 1))) + 10 \|x_N\|_2^2, \quad (5)$$

where $\rho = 0.01$ is a fixed parameter. Solve the modified problem with IPOPT using the same initial guess as before and plot the obtained results.

(1 bonus point)

- (e) **[Bonus]** Solve the formulation at point d) with your implementation of the Newton method. In order to make sure that the arguments of the logarithms in the cost are always positive, implement a linesearch: starting from a full step, check whether any of the constraints are violated and half the step size until the arguments of the logarithms are positive. Use the same initial guess as at point b).

(2 bonus points)

This sheet gives in total 10 points and 3 bonus points