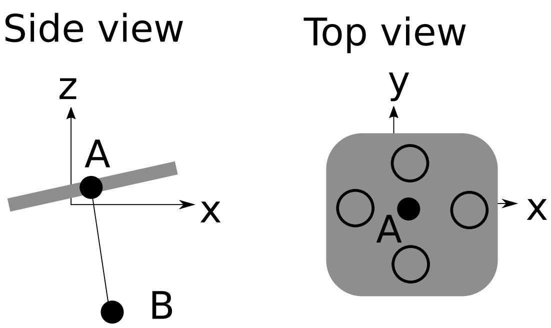
AWESCO Winter School on Numerical Optimal Control with DAE - University of Freiburg

Exercise 12: Optimal Control with DAEs

Joris Gillis, Rien Quirynen, Joel Andersson, Sebastien Gros, Moritz Diehl

Modeling



The model we will work with in this excercise and the next represents a quadcopter with a 3D pendulum attached at the centre of mass. This configuration allows for simple modeling: treating the translation modes and rotation modes independently. In the template you will find a multiple shooting code for a quadcopter system without pendulum. The file makes use of `casadi_struct2vec` and `casadi_vec2struct`. These helper functions convert between a structure and a flat vector. You can play in the terminal to see how these methods behave.

1.1 Run the given template as is: multiple shooting using a degree-4 'radau' scheme as integrator. Interpret the output of IPOPT. Which takes more time: the function evaluations (CasADi) or the solver time?

1.2 Change the solution method to a direct collocation method, using the same scheme and order as before, but now with the intermediate variables exposed to the NLP. You will need:

```
# Degree of interpolating polynomial
d = 4

# Get collocation points
tau_root = C.collocationPoints(d, 'radau')
```

Make use of the provided `simpleColl` method. You should be able to follow from the comments in its source code what it does. Check afterwards that your method delivers the same solution as the multiple shooting method.

1.3 Inspect the timings again. You should see a large difference with last time. Can you explain this?

1.4 Extra: compare the sparsity patterns of the constraint Jacobian for multiple shooting and direct collocation. Which is sparser in total? Which has the largest dense block?

In this second part, you will extend the DAE model with a pendulum.

2.1 Use the Lagrange formalism to add the pendulum to the dae. Add a point mass $m_b = 0.1\,\text{kg}$ at a fixed distance $L = 0.2\,\text{m}$ of the centre of mass of the quadcopter p_a. Make sure you add the appropriate invariants, and multiplier. Find (by hand) a `x0_guess`, `u_guess` and `z_guess` that makes the residual (`daefun`) zero. Try to use physical reasoning for this.

2.2 For the optimal control problem, choose

- `q_lb = [0,0,-inf,0,0,`L`_pendulum]` and `q_ub = [0,0,inf,0,0,`L`_pendulum]` for the initial time,
- `q_lb = [1,0,-inf,1,0,0.5+`L`_pendulum]` and `q_ub = [1,0,inf,1,0,0.5+`L`_pendulum]` for the final time.
- For both initial and final time constrain a zero velocity for p_b and put no constraints on the velocity of p_a.

Further, add a regularisation term for $p_a - p_b - [0; 0; L\_\text{pendulum}]$ to the objective, and solve the resulting optimal control problem.

2.3 The solution found should result in $p_{a,z}(t_0) = 0$ and $p_{a,z}(t_{final}) = 0.5$. What happens if you plug in this knowledge into the boundary conditions i.e. replace the above infinity?