Exercise: Time-optimal control

1 The effect of non-fixed time on problem structure.

In this section we will look at the dynamics of a point mass in a 2D $x - y$ plane, controllable by forces acting along both axes.
The point mass starts from standstill at $(0, 1)$ and needs to visit a fine grid of waypoints.
The point mass has limits on position, speed and acceleration constraints.

Tasks:

1. Locate and run the reference multiple-shooting implementation `template`. Interpret the code and the plots. Why does the OCP solution deviate so much from the reference trajectory?

2. Change the integration method to explicit Euler, and change the NLP solver the `sqpmethod`. Verify that the objective (=tracking error) is `2.530909e-01`. Explain why this takes only one SQP step.

3. Make the control horizon a decision variable, bounded between 0.5 s and 2 s, and initialize with 1. Solve with `ipopt` solver and verify that you obtain a much improved tracking error `1.972783e-02`.

   Switch again to `sqpmethod` solver. Why does the SQP method take more than 1 step?

   The solver might fail to converge unless you specify a strategy to deal with non-convex subproblems (default is to ignore convexity):

   ```
   opti.solver('sqpmethod',{"convexify_strategy": "eigen-clip"})
   ```

4. (extra) Explain the dense column in the constraint Jacobian. Make a re-implementation of 1.3 that does not have a dense column in the constraint Jacobian, but still yields the exact same tracking error. Compare the timing in the 'convexify' step with the previous exercise

2 Racing around obstacles

Tasks:

1. Start again from `template`. Introduce a circle in the scene, centered around $(1, 0.75)$ with a radius of 0.6. Remove the tracking objective. Find the time optimal trajectory that starts from standstill at $(0, 0)$ and ends at $(2, 1)$ without a prescribed final velocity.

   Verify that you obtain an optimal time of either `1.0437847e+00` or `8.4867232e-01` seconds.

2. (extra) Give the circle a radius of 0.601 and compute how much the optimal time $T^\star$ changed. Use a finite difference to approximate $\frac{\partial T^\star}{\partial r}$.

   Inspect the Lagrange multipliers (a.k.a dual variable variables) associated with the obstacle (see `https://web.casadi.org/docs/#dual-variables` for syntax). Explain why most multipliers are very small.

   Can you find the relationship between the multiplier of the active constraint and $\frac{\partial T^\star}{\partial r}$?

3. Increase the control horizon to $N = 40$. Find the time optimal trajectory that starts from standstill at $(0, 0)$, passes through transit point $(2, 1)$ at $k = 20$ without a prescribed velocity, and return to $(0, 0)$ without a prescribed final velocity.

   Find the clockwise solution. Verify that you obtain $T^\star =$`2.2065`

4. Instead of prescribing that the transit point should be encountered exactly in the middle of the total control time, consider a variant in which the encounter may happen at any time instance the solver chooses. Do you expect $T^\star$ to increase or decrease? Verify that $T^\star$ changes by `0.2208`.

5. (extra) Find a periodic trajectory around the obstacle that passes through $(0, 0)$ and $(2, 1)$ without prescribed velocity.

   Verify that you obtain $T^\star =$`1.9430`.

3 (extra) Time-optimal path following

1. Start again from `template`. Change to a CasADi built-in integrator object (`cvodes`). Verify that the solution stays very much the same.

2. Make the control horizon a decision variable again. Verify that you obtain a tracking error of `2.7382665e-03`

3. (extra) Change the problem as to require the end position to be $(\sin(2), \cos(2))$, and *add* $a * T$ to the objective to effectively create a trade-off between time-optimality and tracking error. Compare for $a = 1$ (leads to $T^\star =$ `0.96184`) and $a = 10$ (leads to $T^\star =$ `0.62242`).

4. (extra) Trade-offs like this are easy enough to suggest and solve, but difficult to tune. The factor a has little intuitive meaning. Change to a more intuitive scheme: optimize purely for time, and prescribe a maximum deviation (0.2) of any point w.r.t to the reference `ref`. Verify that you obtain $T^\star =$ `0.97318`.

5. (extra) In robotics, it is often required to follow a geometric path: ie a path without timing information, unlike our reference `ref` which speeds ahead at a constant pace. Adapt 3.4 such that the optimal trajectory stays within a limited bound to the geometric path $s \mapsto \begin{bmatrix}\sin(2s) & \cos(2s)\end{bmatrix}^T, s = 0..1$. Verify that you obtain $T^\star =$ `0.66866`.