OCP/MPC Workshop 2024
Rockit tutorial: part 2

Alejandro Astudillo Vigoya, Wilm Decré, Louis Callens, Alex Gonzalez García, Dries Dirckx

July 17h, 2024

1 Assignment introduction

In this assignment, an optimal trajectory will be computed for a quadcopter drone. This trajectory serves as a reference for a simple controller that will track this trajectory. While simulating, disturbances will be present in the form of wind and a plant-model mismatch.

The optimal trajectory can be computed by solving an OCP using Rockit.

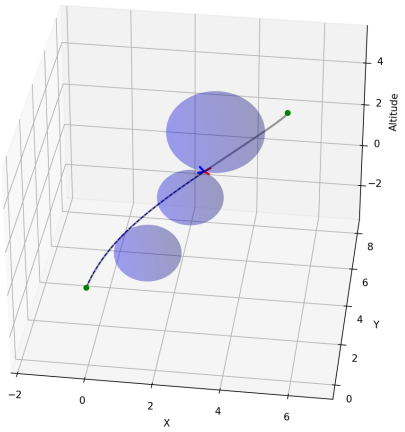

Figure 1: A 3D simulation of a drone tracking an optimal point-to-point trajectory, while avoiding obstacles.

2 The Optimal Control Problem

The goal of this assignment is to make the quadcopter fly from a starting position (with coordinates (0, 0, 0) [m]) to a destination(with coordinates (5, 8, -2) [m]), starting and ending in a hovering state, all while avoiding spherical obstacles.

The dynamics of the quadcopter used in the OCP are taken from [1]. This model makes some simplifications such that it can easily be used within an optimization problem. It considers 9 states, representing the position coordinates, velocities and euler angles. The control inputs are the angular

velocity of the roll, pitch and yaw angles and a total thrust of the drone. Note that this choice of control inputs implies that the angular velocity of the drone can change instantly. The model used to simulate the drone, does take into account rotational inertia meaning there is a plant-model mismatch.

3 Getting started

The code to simulate the drone, along with its controllers is taken from this github repository. Instead of using the class `Trajectory`, we've created a new class called `OptimalTrajectory`. When such an optimal trajectory is created, it calls a function defined within this class called `optimize`. We will add the computation of an optimal trajectory in there using rockit.

By running the file `run_3D_simulation.py`, a simulation is performed. Note that in the function `main()`, defined in `run_3D_simulation.py`, a `wind` object is created. Feel free to play around with these disturbances to put the controller to the test.

References

[1] Dries Dirckx et al. "Optimal and Reactive Control for Agile Drone Flight in Cluttered Environments". In: *IFAC-PapersOnLine* 56.2 (2023). 22nd IFAC World Congress, pp. 6273–6278. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2023.10.777. URL: https://www.sciencedirect.com/science/article/pii/S2405896323011540.