Summer School on Robust Model Predictive Control with CasADi
James B. Rawlings, Joel Andersson, Sergio Lucia, Moritz Diehl
University of Freiburg, September 15 to 19, 2025
`www.syscop.de/event/rmpc25`

Exercise 4: Zero-order robust NMPC with `acados`

Florian Messerer, Jonathan Frey, Katrin Baumgärtner, Moritz Heinlein

On this exercise sheet, we consider again the nonlinear ellipsoidal tube OCP from Exercise 3, for the example of the differential drive robot. We first learn how to solve it with the zoRO implementation of `acados`[1]. We then use it in the closed loop as an MPC solver.
Recall that the nonlinear ellipsoidal tube OCP is given as

$$\begin{aligned}
\min_{\bar{x},\bar{u},P} \quad & \sum_{k=0}^{N-1} l_k\left(\bar{x}_k,\bar{u}_k\right)+l_N\left(\bar{x}_N\right) && && \text{(1a)}\\
\text{s.t.} \quad & \bar{x}_0=\bar{\bar{x}}_0, && && \text{(1b)}\\
& P_0=0, && && \text{(1c)}\\
& \bar{x}_{k+1}=f_k(\bar{x}_k,\bar{u}_k,0), && k=0,\ldots,N-1, && \text{(1d)}\\
& P_{k+1}=\psi_k(\bar{x}_k,\bar{u}_k,P_k), && k=0,\ldots,N-1, && \text{(1e)}\\
& 0\geq h_k(\bar{x}_k,\bar{u}_k)+b_k(\bar{x}_k,\bar{u}_k,P_k), && k=0,\ldots,N-1, && \text{(1f)}\\
& 0\geq h_N(\bar{x}_N)+b_N(\bar{x}_N,P_N). && && \text{(1g)}
\end{aligned}$$

The main idea of the zero-order robust NMPC algorithm[2] (zoRO) is to fix the disturbance ellipsoids within one optimization iteration and only update the trajectory of disturbance ellipsoids outside the OCP.
In this exercise we only consider a very simple MPC problem, with a fixed reference point. For a more elaborate implementation, with time-varying reference tracking, including the generation of a reference trajectory alongside a given path, see `https://github.com/acados/acados/blob/main/examples/acados_python/zoRO_example/diff_drive`, which corresponds to the folder `acados/examples/acados_python/zoRO_example/diff_drive` in your local acados clone. The details of this implementation are described in Gao2023[3].
A side remark on the Generalized Gauss Newton (GGN) Hessian approximation (not necessary to read to solve the exercise, but helps to understand how the stage cost is set in acados): Recall that the stage and terminal cost functions are given as

$$\begin{aligned}
l(x,u) &= \gamma l_{\text{Huber}}(p-p_{\text{target}})+u^\top R u, && \text{(2a)}\\
l_N(x) &= \gamma_N l_{\text{Huber}}(p-p_{\text{target}})+\tau v^2. && \text{(2b)}
\end{aligned}$$

Both have a "convex-over-nonlinear" structure, $\psi(F(z))$, where we call $\psi(\cdot)$ the "outer convexity" and $F(\cdot)$ the "inner nonlinearity". The stage cost can be written in this form by considering $z=(x,u)$, $F(z)=Sz$, with $S\in\mathbb{R}^{4\times(n_x+n_u)}$ a selector matrix which selects the entries p_x, p_y, u_1, u_2 from z (i.e., the entries which actually enter the stage cost). Note that in this case, F is not actually nonlinear, but it could be in general. The outer convexity is then given as $\psi(y)=\gamma l_{\text{Huber}}((y_1,y_2)-p_{\text{target}})+(y_3,y_4)^\top R\,(y_3,y_4)$, with $y=F(z)$. For problems with this "convex-over-nonlinear" structure, we can use, e.g., the Generalized Gauss Newton (GGN) Hessian approximation. For more details, see[4].

[1] Jonathan Frey, Yunfan Gao, Florian Messerer, Amon Lahr, Melanie N Zeilinger, and Moritz Diehl. Efficient zero-order robust optimization for real-time model predictive control with acados. In *Proceedings of the European Control Conference (ECC)*, 2024

[2] Andrea Zanelli, Jonathan Frey, Florian Messerer, and Moritz Diehl. Zero-order robust nonlinear model predictive control with ellipsoidal uncertainty sets. *Proceedings of the IFAC Conference on Nonlinear Model Predictive Control (NMPC)*, 2021

[3] Yunfan Gao, Florian Messerer, Jonathan Frey, Niels van Duijkeren, and Moritz Diehl. Collision-free motion planning for mobile robots by zero-order robust optimization-based mpc. In *Proceedings of the European Control Conference (ECC)*, 2023

[4] Florian Messerer, Katrin Baumgärtner, and Moritz Diehl. Survey of sequential convex programming and generalized Gauss-Newton methods. *ESAIM: Proceedings and Surveys*, 71:64–88, 2021

Tasks

1. First consider the file `main_ocp.py`. Here, solve the initial OCP with both IPOPT and zoRO, in order to compare the two solutions. Complete the zoRO solver found in `solver_zoro_acados.py` such that you are able to run `main_ocp.py`. Note that we split the inequality constraints into box constraints on the states and controls as well as nonlinear constraints. Take a moment to compare the two resulting trajectories.

2. Now consider `main_mpc.py`. Take a moment to familiarize yourself with the code, and run it. How does the resulting trajectory compare to the initial OCP prediction?

3. In the previous simulation, all disturbance values were set to 0. Look for the `TODO` in `main_mpc.py` and play around with the disturbance scaling.

4. The above MPC setting was fairly simple. If you are interested, take a look at the more elaborate path following zoRO mentioned above.

Bonus Task without `acados` (Mini project)

Implement zoRO yourself. You need a nominal OCP with fixed backoffs, which you can modify after each call, as well as a routine for disturbance propagation and backoff computation along a given nominal trajectory. Then alternate between these two until convergence. As a starting point, you can use the given `SolverOcp` and `SolverOpenLoopRobustOcp` classes from Exercise 3.