

---

# AA222 Final Project status: Trajectory Optimization with dynamic obstacles avoidance

---

Philippe Weingertner and Minnie Ho

## 1 Project description (reminder)

This project investigates trajectory optimization in the presence of obstacles. One such application for this class of problems is that of autonomous driving, where we have an ego vehicle and dynamic obstacles (vehicles, pedestrians) which may intersect our desired trajectory and which we wish to avoid using motion planning and control. Trajectory optimization problems minimize a cost function which takes into account start and terminal states, as well as cost along the trajectory path. The design space is subject to constraints on the states and control input at sampled time points and on a set of dynamic obstacles to avoid.

It is a multi-objective optimization problem: we have to optimize efficiency (time to goal), comfort (minimizing acceleration variations) and safety (avoiding collisions).

## 2 Project status

So we far we did:

1. Literature review and project objectives definition (cf below)
2. Github repository setup
3. In the continuity of a previous [project](#), we have a first baseline implementation. A Model Predictive Control (MPC) solution based on Julia JuMP.jl and Ipopt.jl which is fully functional.

We rely on a direct collocation method [3] where MPC is used: we plan over 20 time steps, for 20 commands, but apply only the first command before taking into account new observations for re-planning. What we plan to do next is:

1. Setup of our own optimizer, without relying on external packages like JuMP and Ipopt. We will leverage on project2 work where an Interior Point Method based on log barrier was implemented [4]. We implemented a first order (Conjugate Gradient) and second order (quasi-newton BFGS) optimizer with backtracking line search.
2. Handle an inconsistent constraint of the form  $abs(.)$  as per [1]:
  - (a) With slack variables
  - (b) With a smooth version of  $abs()$
3. Analyze the influence of various initialization strategies [1]:
  - (a) Initialization with a trajectory that is dynamically feasible but not safe
  - (b) Initialization with a Neural Network heuristic. Based on a previous project we already have a Neural Network that can propose a candidate solution. This solution is unsafe in 20% of the cases but could be used as a useful initializer.
4. If possible, depending on how fast we progress with the 3 previous objectives, we will compare optimization over station [5, 7, 2] with optimization over time [6] for 2D paths. In the first case we fix the time steps and derive positions at every time step, while in the second

case we fix the spatial positions and derive the time instants at which we go over these spatial positions. In the first case, when dealing with 2D paths, an analytical representation of the path is required, while in the second case, a sequence of waypoints is sufficient. However, in the second case, the constraints may be even more non-linear and non-convex.

The issue raised in Point 2) above is avoided in many publications by claiming that a higher level planner decides whether we have to proceed or yield the way. While this may be a reasonable strategy when dealing with a single object crossing our path, it does not scale to multiple crossing objects. An approach from basic principles is required to scale with the complexity of the scene. In this way, Point 3) could be a promising combination of techniques, leveraging on offline Neural Network training to speed up online optimization convergence. For Point 4), while some publications focus on one or the other approach, we would like to outline more precisely the pros and cons of these approaches and benchmark their accuracy and convergence speed.

## References

- [1] John T. Betts. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. 2nd. USA: Cambridge University Press, 2009. ISBN: 0898716888.
- [2] T. Gu, J. M. Dolan, and J. Lee. “Runtime-bounded tunable motion planning for autonomous driving”. In: *2016 IEEE Intelligent Vehicles Symposium (IV)*. 2016, pp. 1301–1306.
- [3] Matthew Kelly. “An Introduction to Trajectory Optimization: How to Do Your Own Direct Collocation”. In: *SIAM Review* 59 (Jan. 2017), pp. 849–904. DOI: [10.1137/16M1062569](https://doi.org/10.1137/16M1062569).
- [4] Mykel J. Kochenderfer and Tim A. Wheeler. *Algorithms for Optimization*. The MIT Press, 2019.
- [5] X. Li et al. “A practical trajectory planning framework for autonomous ground vehicles driving in urban environments”. In: *2015 IEEE Intelligent Vehicles Symposium (IV)*. 2015, pp. 1160–1166.
- [6] C. Liu, W. Zhan, and M. Tomizuka. “Speed profile planning in dynamic environments via temporal optimization”. In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. 2017, pp. 154–159.
- [7] Tianyu Gu et al. “Tunable and stable real-time trajectory planning for urban autonomous driving”. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2015, pp. 250–256.