Two-stage MPC architecture for Safe Autonomous Driving

Martin Medina, Étienne Mitchell-Bouchard

Context & Methodology

Problem & Context

Autonomous vehicles in urban environments must both choose safe routes through a city road network and follow those routes safely in real time. The environment is dynamic and unpredictable, with other vehicles, pedestrians, and traffic lights. A system that can adapt to changing conditions continuously is needed.

Goal

This project implements a hierarchical control architecture for urban autonomous driving. Our goal is to enable a vehicle to safely track an offline-computed trajectory in real-time, while strictly respecting speed limits, handling static and dynamic obstacles, and adhering to safety and vehicle dynamic constraints.

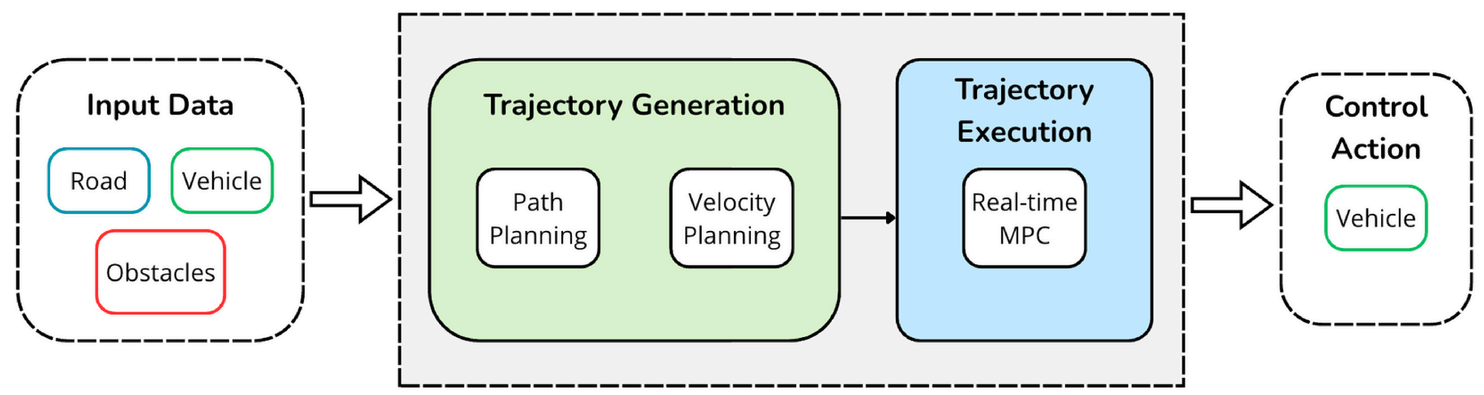
Model

We used a kinematic vehicle model to describe the vehicle's motion relative to the reference path.

- **State:** $x = [s, d, o, k, v]$ represents the traveled distance (s), lateral deviation (d), orientation error (o), trajectory curvature (k), and velocity (v).
- **Controls:** $u = [u_1, u_2]$ regulates the steering curvature rate (u_1) and acceleration (u_2).
- **Dynamics:** $\dot{x}(t) = f(x(t), u(t))$

Method : Multi-Stage Control Architecture

1. **Path Planning (Global Routing):**
   - Fetches real-world map data and speed limits. Processes raw waypoints into a continuous cubic spline.
2. **Offline Trajectory Planning (Non-Linear MPC):**
   - Generates an optimal velocity profile respecting road curvature and speed limits.
   - **Solver:** SLSQP.
   - **Discretization method:** Hermite-Simpson collocation.
   - **Objective:** Minimize travel time and deviation errors, respecting speed limits and maximizing user comfort.
3. **Online Real-Time Tracking (Linearized MPC):**
   - Tracks the reference trajectory in real-time, minimizing tracking errors, and safely handling static and dynamic obstacles.
   - **Linearization:** Dynamics are linearized around the operating point to ensure real-time solve rates.
   - **Solver:** SLSQP.
   - **Discretization method:** Explicit-Euler collocation.
   - **FSM Logic:** A Finite State Machine handles environment obstacles (traffic lights and vehicles).

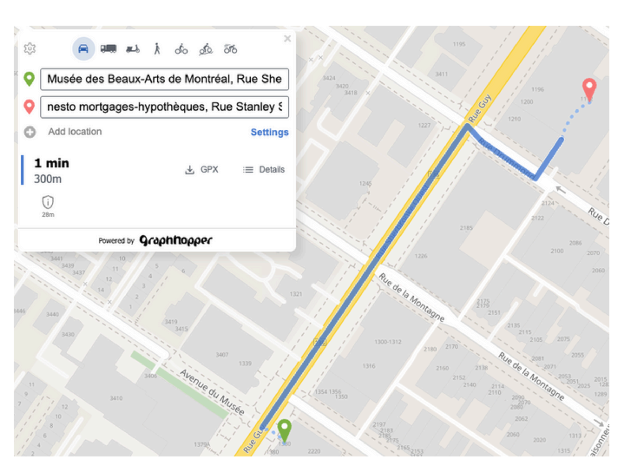


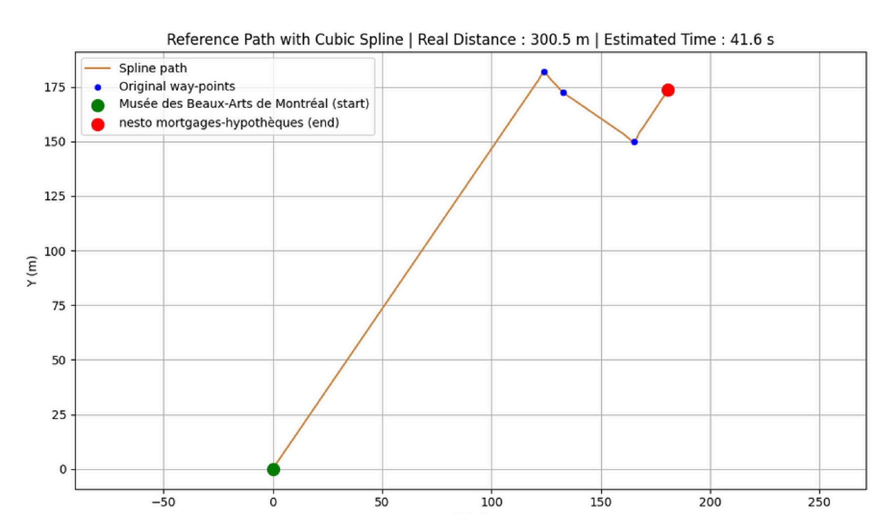
Baseline Verification

To validate the hierarchical control architecture, we demonstrate the generation and tracking of an optimal trajectory along a real-world 300m urban route in downtown Montreal.

Path Planning

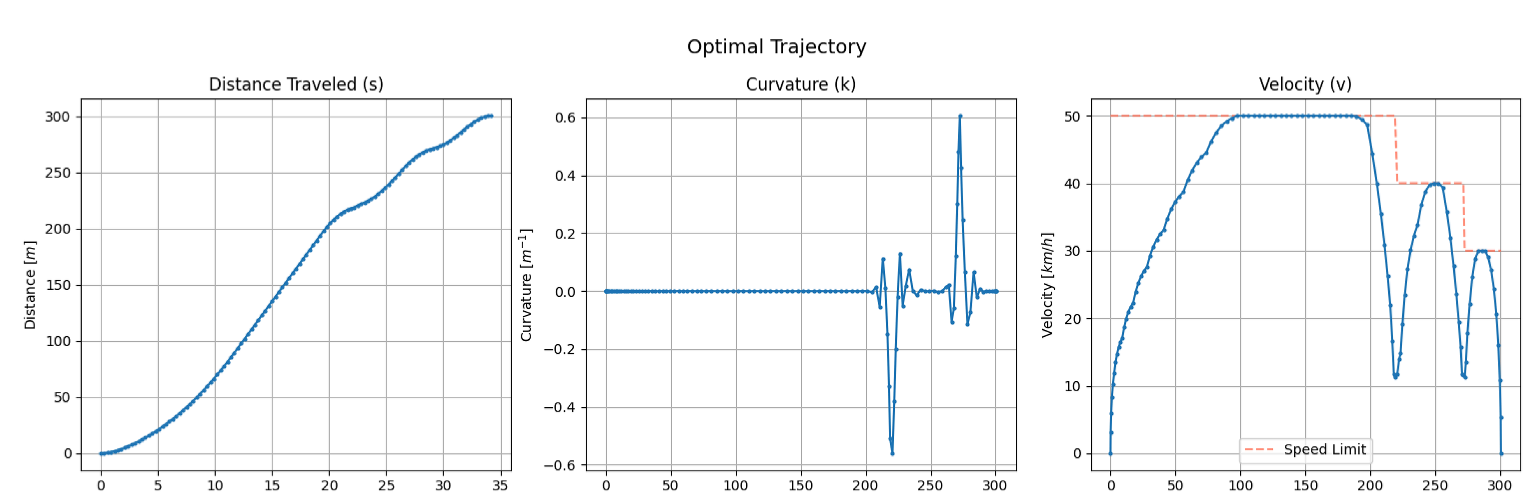
The system first ingests real-world map data (Musée des Beaux-Arts to Stanley St.) and processes raw waypoints into a continuous cubic spline. This creates a differentiable reference path for the solver to track.

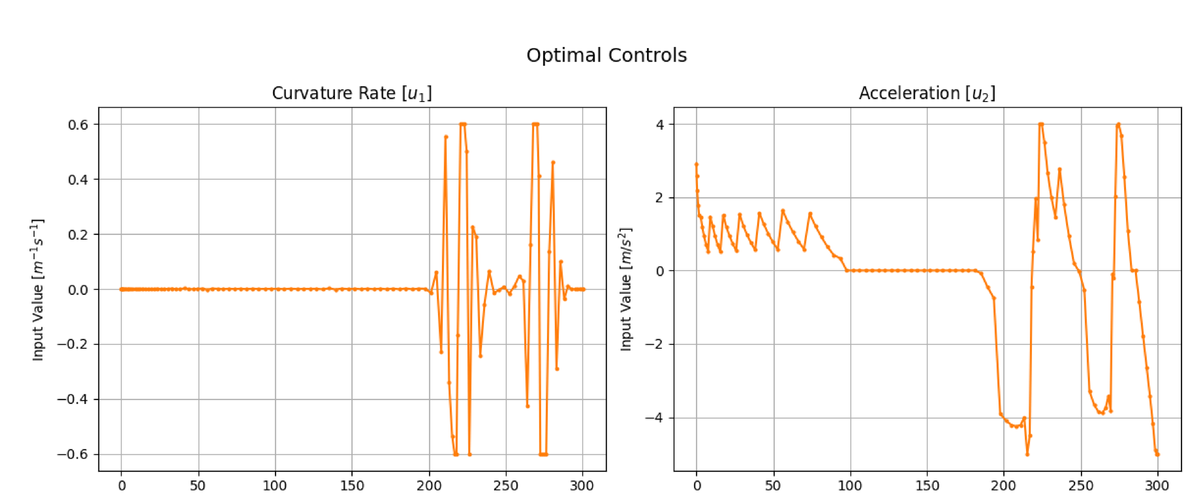


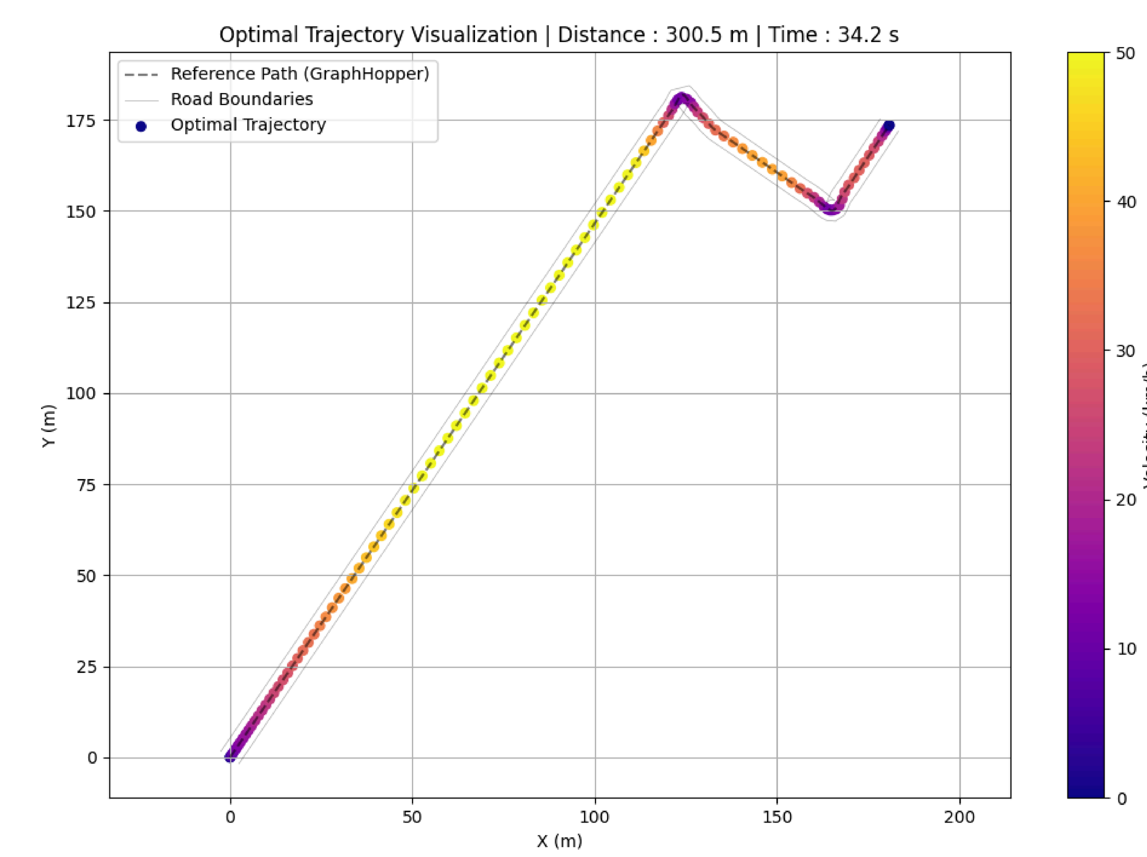


Trajectory Planning

The Non-Linear MPC utilizes a predictive horizon to generate a velocity profile that proactively adapts to road geometry, optimizing for time-efficiency while strictly maintaining vehicle stability and passenger comfort.

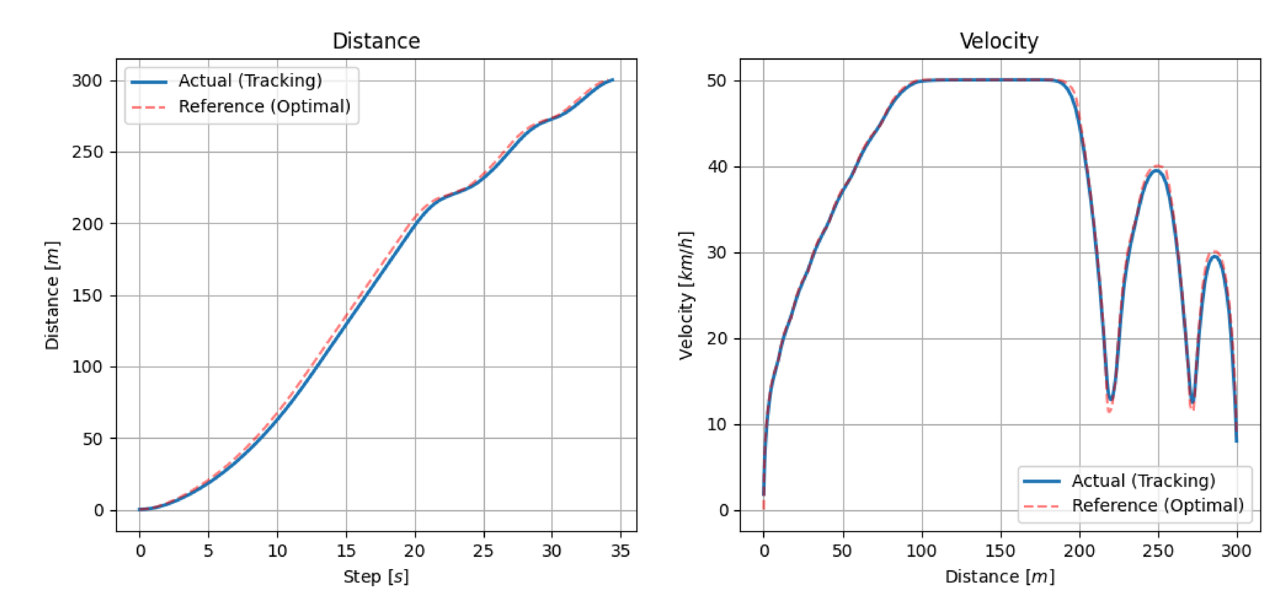






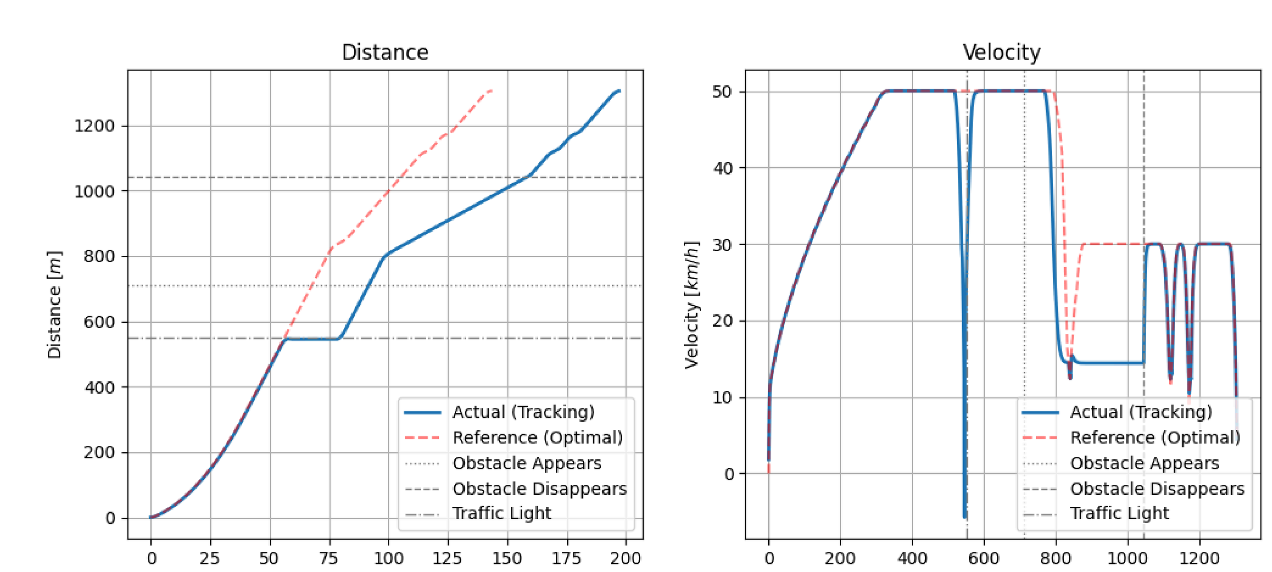
Trajectory Tracking

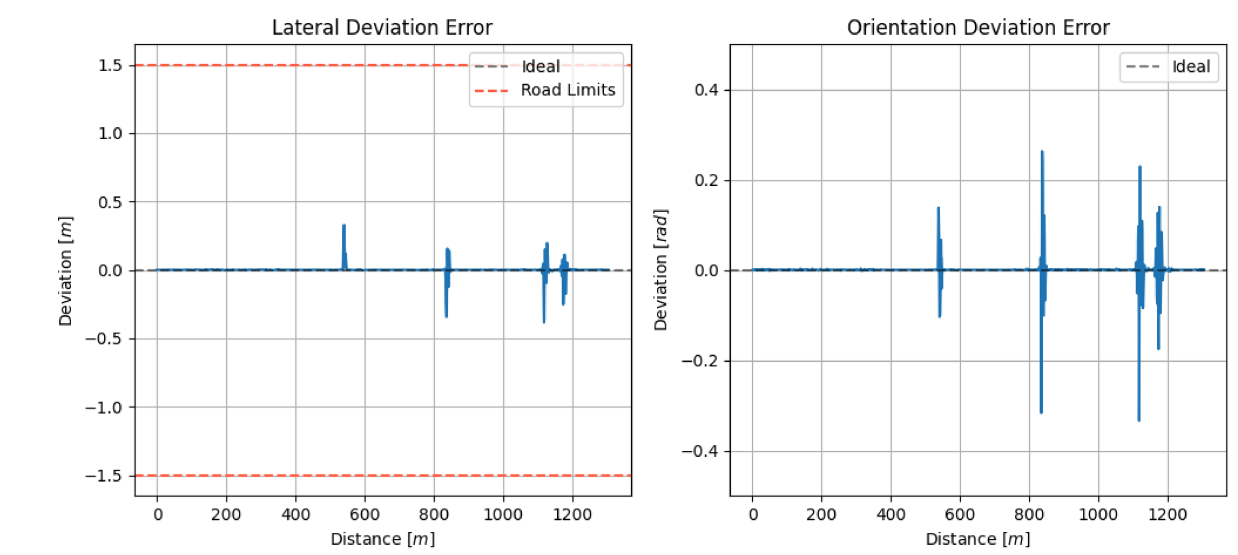
The linearized MPC operates in real-time to compute optimal control inputs (Curvature Rate and Acceleration) by minimizing tracking errors while ensuring the vehicle remains within its dynamic stability limits. This simulation is conducted in an obstacle-free environment to strictly isolate and establish a baseline for tracking performance.

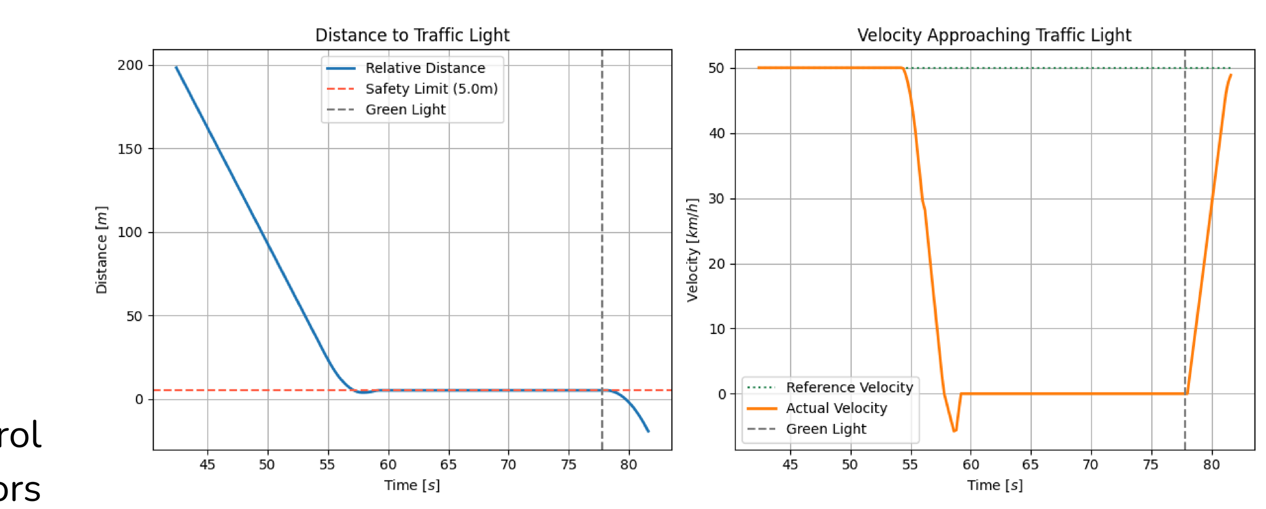


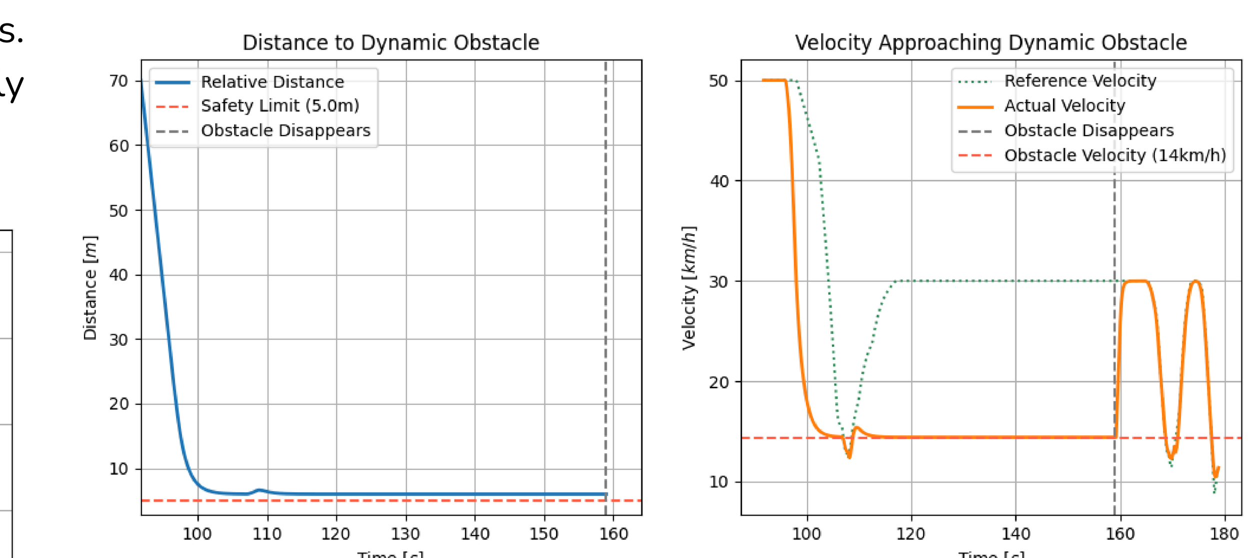
Complex Evaluation

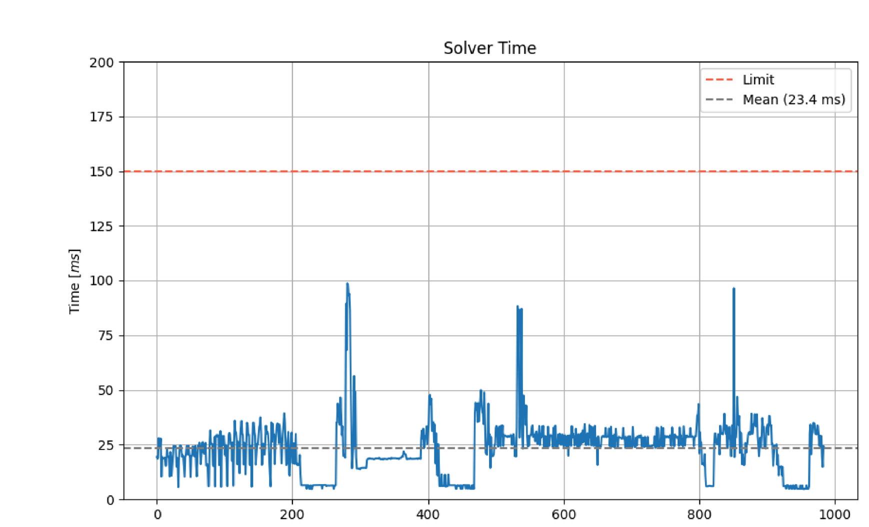
To assess system robustness, we extended the validation to a longer 1300m urban route. This scenario introduces static (traffic light) and dynamic (lead vehicle) obstacles, forcing the system to dynamically switch between trajectory tracking, precision braking, and adaptive car-following modes in real-time.











Scan the QR code to view animations of this tracking simulation and an additional complex scenario:

Conclusion & Limitations

This project validated a hierarchical control architecture that effectively balances computational speed with safety, successfully handling obstacles like traffic lights and external vehicles. However, this real-time implementation limits the vehicle's ability to recover from large deviations or extreme maneuvers. Additionally, the simulation currently assumes perfect perception and deterministic obstacle behavior. Future work must incorporate sensor noise and stochastic prediction to handle real-world uncertainty.

Main inspiration paper: Emam, M., & Gerdts, M. (2024). MPC-Based Routing and Tracking Architecture for Safe Autonomous Driving in Urban Traffic. SN Computer Science, 5:375.