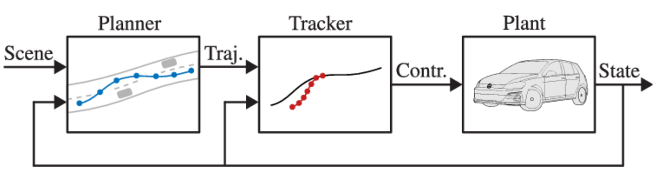
Trajectory Optimization - MPC

Leo Jaeeun Kim
University of Michigan
General Motors

ABSTRACT

Trajectory optimization plays a critical role in the performance, safety, and efficiency of autonomous systems. This research explores a Model Predictive Control (MPC) approach for generating optimal trajectories that satisfy nonlinear vehicle dynamics, obstacle avoidance constraints, and mission objectives. We implement a bicycle model for vehicle dynamics and leverage control barrier functions to ensure collision avoidance. Distinguishing between open-loop trajectory optimization and closed-loop control strategies, we examine how optimized trajectory sequences can be effectively integrated with tracking controllers. The optimization problem is formulated using CasADi's symbolic framework and solved with the IPOPT nonlinear programming solver. Our approach demonstrates effective navigation through environments with multiple obstacles while minimizing control effort and ensuring smoothness. The results highlight the potential of constrained optimization techniques for real-time trajectory planning in autonomous driving applications, balancing computational efficiency with solution quality, and bridging the gap between offline planning and online execution.



<Vehicle motion planning and control [3]>

INTRODUCTION

As autonomous vehicles and robotic systems become increasingly prevalent in complex environments, the ability to plan and execute optimal trajectories has emerged as a fundamental requirement. Trajectory optimization involves computing a time-parameterized path that guides the system from an initial state to a desired goal while minimizing a cost function and satisfying various constraints.

Trajectory optimization is fundamentally a collection of techniques used to find open-loop solutions to optimal control problems. The solution is a sequence of controls, given as a function of time, that moves a system from a single initial state to some final state. This sequence of controls, combined with the initial state, defines a single trajectory that the system takes through state space. An important distinction exists between open-loop and closed-loop solutions for trajectory optimization. While an optimal trajectory starting from any point in the state space can be recovered from a closed-loop solution by a simple simulation, open-loop solutions require additional mechanisms to handle disturbances and model uncertainties.

In autonomous driving, trajectory optimization enables smooth navigation and obstacle avoidance in dynamic scenarios. Despite advances in hardware and perception, achieving reliable and adaptable trajectory optimization remains challenging due to nonlinear dynamics, non-convex constraints from obstacles, and real-time computation requirements.

This paper presents a Model Predictive Control (MPC) formulation for trajectory optimization in autonomous vehicle applications. We employ a bicycle model to represent vehicle dynamics with states including position, heading, and velocity. The optimization framework incorporates control barrier functions to ensure obstacle avoidance while minimizing a cost function that balances goal-reaching, trajectory smoothness, and control effort. Our implementation uses CasADi for symbolic computation and IPOPT for efficient numerical optimization.

A key focus of this work is exploring the relationship between trajectory optimization and tracking controllers. While trajectory optimization generates an optimal reference path, tracking controllers like MPC aim to follow this reference while adapting to real-world conditions. Understanding this interplay is crucial for developing systems that maintain optimality while responding to environmental changes and disturbances.

The approach is demonstrated in a representative scenario where a vehicle must navigate from an initial state to a specified goal position while avoiding multiple obstacles. The results illustrate how constrained optimization can effectively generate trajectories that satisfy both dynamical constraints and safety requirements, contributing to the development of more capable and reliable autonomous systems.

METHOD

Trajectory optimization involves techniques used to find open-loop solutions for optimal control problems. The goal is to determine the control inputs as functions of time to drive a system from an initial state to a desired final state. These optimal trajectories are distinct from closed-loop control (like MPC or LQR), where feedback is employed continuously. Trajectory optimization methods include indirect and direct approaches. Indirect methods derive analytical conditions for optimality, while direct methods discretize the problem into a finite-dimensional optimization problem before solving it numerically. Direct methods (shooting or collocation) have become particularly popular due to their ease of handling complex constraints and dynamics. Direct collocation, specifically, discretizes the state and control trajectories using polynomial approximations to efficiently handle path constraints and complicated dynamics, making it suitable for practical applications like robotics and vehicle dynamics.

This research employs a direct collocation approach for trajectory optimization, recognized for its efficiency in handling complex dynamic systems and constraints commonly encountered in autonomous vehicle and robotic applications.

Trajectory optimization is formulated as an optimal control problem where the objective is to find the optimal sequence of state variables x_k and control inputs u_k that minimize a cost function *J*. This cost function typically includes both terminal state costs and accumulated costs over the entire trajectory.

$$J = N(x(t_N)) + \int_0^{t_N} L\big(x(t), u(t)\big)dt$$

Subject to dynamics constraints:

$$\dot{x} = f\big(x(t), u(t)\big)\,, \qquad x(0) = x_{init}\,, \qquad x(t_N) = x_{des}$$

and additional constraints, including path constraints (equality or inequality) and bounds on state and control inputs.

Direct Collocation Approach

The trajectory optimization problem is discretized by dividing the trajectory into N finite segments defined by discrete time points t_k. Within each segment, the state and control trajectories are approximated using polynomial interpolations. The continuous optimization problem thus transforms into a finite-dimensional nonlinear optimization problem of the form:

$$N(x_N) = \sum_{k=0}^{N-1} L_k(x_k, u_k)$$

subject to discretized dynamics constraints:

$$x_{k+1} = x_k + \int_{t_k}^{t_{k+1}} f\big(x(t), u(t)\big)dt \;\approx f_k(x_k, u_k)$$

with initial and terminal constraints:

$$x_0 = x_{init}\,, \qquad x(N) = x_{des}$$

and the inclusion of state and input constraints explicitly enforced at each discretization point:

$$g(x_k, u_k) \le 0\,, \qquad h(x_k, u_k) = 0$$

Numerical Implementation

The nonlinear optimization resulting from direct collocation is solved numerically using standard nonlinear programming solvers (e.g., IPOPT, SNOPT). The solver iteratively updates the decision variables (x_k, u_k) until convergence to a locally optimal solution is achieved. Sensitivity to initial guesses and computational efficiency considerations are carefully addressed.

Convex Approximation for Specific Cases

In scenarios where system dynamics and constraints are linear or affine, such as Linear Quadratic Regulator (LQR) problems, the optimization simplifies to a convex quadratic programming (QP) problem. This research employs convex formulations whenever applicable, leveraging their computational efficiency and guaranteed global optimality.

IMPLEMENTATION

Trajectory Optimization Formulation

This research implements a direct transcription approach to trajectory optimization for autonomous vehicle navigation. We formulate the problem as a nonlinear program (NLP) using the CasADi symbolic framework and solve it with the IPOPT (Interior Point OPTimizer) solver. The optimization determines a feasible and optimal trajectory from a starting point to a goal location while avoiding obstacles.

Vehicle Dynamics Model

Bicycle model to describe the vehicle dynamics with state vector comprising position coordinates, heading angle, and velocity:

$$x = [\,x \;\; y \;\; \theta \;\; v]^T$$

The control inputs are the steering rate (angular velocity) and acceleration:

$$u = [\omega, a]^T$$

The continuous-time dynamics are defined by the following differential equations:

$$\begin{aligned} \dot{x} &= v cos(\theta) \\ \dot{y} &= v sin(\theta) \\ \dot{\theta} &= \omega \\ \dot{v} &= a \end{aligned}$$

For numerical optimization, we discretize these dynamics using the forward Euler method with a time step Δt.

$$x_{k+1} = x_k + \,\Delta t * f(x_k, u_k)$$

where f(x, u)denotes the continuous dynamics function.

Problem Formulation

The trajectory optimization problem consists of the following components:

Decision variables:

$$\begin{aligned} &Control\ Inputs: u_0, u_1, + \cdots + u_{N-1} \\ &States: x_0, x_1, + \cdots + x_N \end{aligned}$$

where N is the number of discrete time steps.

Dynamics Constraints: For k = 0,1,⋯, N-1

$$x_{k+1} - \left(x_k + \Delta t * f(x_k, u_k)\right) = 0$$

Obstacle Avoidance Constraints

$$b(x_{k+1}) - \alpha * b(x_k) \leq 0$$

where the barrier function b(x):

$$b(x) = r^2 - [(x - x_{obs})^2 + (y - y_{obs})^2]$$

With r as obstacle radius ($(x_{obs,}y_{obs})$ as obstacle center, and $\alpha \in (0,1)$ as tuning parameter for barrier strictness.

Initial State Constraint:

$$x_0 = x_{int}$$

Terminal State Constraints:

$$x_N = x_{goal}, y_N = y_{goal}, \theta_N = 0, v_N = 0$$

Objective Function

The cost function minimized by the optimization balances goal accuracy, trajectory smoothness, and control effort:

$$J = J_{terminal} + J_{running} + J_{control}$$

Terminal Cost:

$$J_{terminal} = \omega_1 \left[\left(x_N - x_{goal}\right)^2 + \left(y_N - y_{goal}\right)^2\right] + \omega_2[\theta_N^2 + v_N^2]$$

Running Cost:

$$J_{running} = \sum_{k=0}^{N-1} \{\omega_3 \left[\left(x_k - x_{goal}\right)^2 + \left(y_k - y_{goal}\right)^2\right] + \omega_4(\theta_k^2 + v_k^2)\}$$

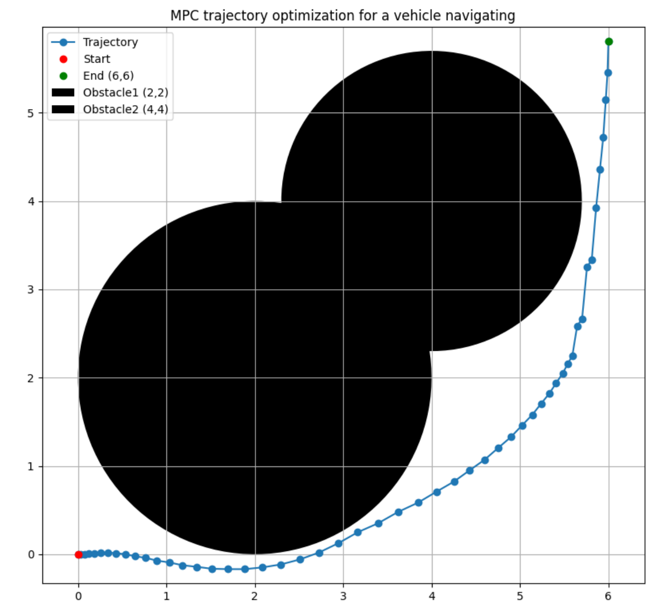
Control Effort Cost:

$$J_{control} = \sum_{k=0}^{N-1} \{\omega_5\left((\omega_k^2 + a_k^2)\right\}$$

Weights are selected as $\omega_1, \omega_2 = 100$, $\omega_3, \omega_4 = 10$, and $\omega_5 = 1$ to prioritize goal accuracy and maintain feasible control inputs.

Solution Approach and Implementation Details

The optimization problem is solved by constructing a symbolic representation in CasADi, defining appropriate bounds for states and controls, initializing the solver variables and multipliers, and finally solving the NLP using IPOPT's interior-point algorithm. The optimization is executed over a 5-second horizon with 50 discrete steps, and the control barrier function parameter is set to 0.9, achieving a suitable balance between safety and feasibility. Control inputs (steering rate and acceleration) are bounded within the interval [-1,1]. In the test scenario, two circular obstacles are placed at coordinates (2,2) with a radius of 2, and (4,4) with a radius of 1.7. The vehicle starts at the origin (0,0,0,0) and is required to reach the goal position at (6,6), thereby navigating a complex path to satisfy all imposed kinematic and dynamic constraints.



The example code for trajectory optimization can be found in https://github.com/woa0425/Leo_ADAS_script/blob/main/Trajectory_Optimization/Trajectory_Optimization_python/MPC_Trajectory_Optimization.py

CONCLUSION

This research presented a Model Predictive Control (MPC)-based trajectory optimization approach for autonomous vehicles, effectively addressing critical challenges such as nonlinear dynamics, obstacle avoidance, and real-time computation requirements. By employing a bicycle model coupled with control barrier functions, we ensured the generated trajectories-maintained safety constraints while achieving optimal performance criteria including goal accuracy, trajectory smoothness, and minimal control effort.

The direct collocation method facilitated the efficient transformation of the trajectory optimization problem into a numerically solvable nonlinear program. Utilizing CasADi's symbolic computation framework along with IPOPT's nonlinear solver allowed for robust and computationally efficient optimization, demonstrating real-time feasibility.

Results from simulation scenarios, involving dynamic navigation tasks with multiple obstacles, confirmed the effectiveness of the proposed MPC formulation. The optimized trajectories consistently balanced computational efficiency and solution quality, bridging the critical gap between offline trajectory planning and real-time execution.

Future work may explore extending this approach to more complex vehicle models, incorporating environmental uncertainties, and validating performance through real-world experiments. This advancement has the potential to significantly enhance the autonomy, safety, and adaptability of intelligent transportation systems and robotic applications.

REFERENCES

[1] University of Michigan, NAVARCH 565 – Lecture 11: Trajectory_Optimization

[2] University of Michigan, NAVARCH 565 – Lecture 12: VehicleDynamics TrajOpt Example

[3] V. A. Laurense and J. C. Gerdes, "Long-horizon vehicle motion planning and control through serially cascaded model complexity," *IEEE Trans. Control Syst. Technol.*, vol. 30,