Trajectory Optimization with Dynamic Obstacles Avoidance

Philippe Weingertner and Minnie Ho
pweinger@stanford.edu minnieho@stanford.edu

Abstract

We study the problem of Trajectory Optimization.

1 Introduction

This project investigates trajectory optimization in the presence of obstacles [12, 3, 6, 13]. 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.

2 Related Work

Todo ...

3 Problem Formulation

We define a MPC problem over 20 time steps of 250 ms each with a Quadratic Cost function with $x \in \mathbb{R}^{60}$ and 160 constraints. We have 120 linear and nonlinear ($\|x_{\text{ego}} - x_{\text{obj}}\| \geq d_{\text{saf}}$) inequality constraints and 40 linear equality constraints (Dynamics Model).

$$\min_{u_0,\ldots,u_{T-1}} (x_T - x_{\text{ref}})\, Q_T\, (x_T - x_{\text{ref}}) + \sum_{k=0}^{T-1} (x_k - x_{\text{ref}})\, Q\, (x_k - x_{\text{ref}}) + u_k\ R\ u_k$$

$$\text{subject to } \begin{cases} x_{k,\text{min}} \leq x_k \leq x_{k,\text{max}} \\ u_{k,\text{min}} \leq u_k \leq u_{k,\text{max}} \\ x_{k+1} = A_d x_k + B_d u_k \\ x_0 = x_{\text{init}} \\ \forall\, (t_{\text{col}}, s_{\text{col}})_{i \in [1,10]} \qquad x_{t_{\text{col}}^{(i)}}\,[1] < s_{\text{col}}^{(i)} - \Delta_{\text{safety}} \text{ or } x_{t_{\text{col}}^{(i)}}\,[1] > s_{\text{col}}^{(i)} + \Delta_{\text{safety}} \end{cases}$$

Linear Dynamics with Constant Acceleration model in between 2 time steps

$$\begin{bmatrix} s \\ \dot{s} \end{bmatrix}_{k+1} = A_d \begin{bmatrix} s \\ \dot{s} \end{bmatrix}_k + B_d \left[\ddot{s}\right]_k \text{ with } A_d = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, B_d = \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix}$$

4 Methods

4.1 Collision Avoidance Model

We reformulate the collision avoidance model and later on demonstrate the improvements it provides in a set of benchmarks. An implementation of this model is available in mpc_mip.jl.

4.1.1 Disjunctive Constraints

In general the collision avoidance constraint is defined as $\left\| \text{pos}_{\text{ego}} - \text{pos}_{\text{obj}} \right\|_2 \geq \text{d}_{\text{safety}}$

When considering the evolution of an ego vehicle along a path denoted by $s\left(t\right)$ and a crossing-point for some other vehicle, at $\left(t_{\text{cross}}, s_{\text{cross}}\right)$, the collision avoidance constraint is reformulated as: $\left|s\left(t_{\text{cross}}\right) - s_{\text{cross}}\right| \geq \text{d}_{\text{safety}}$. Which is equivalent to a disjunctive constraint:

$$s\left(t_{\text{cross}}\right) \leq s_{\text{cross}} - d_{\text{safety}} \quad \vee \quad s\left(t_{\text{cross}}\right) \geq s_{\text{cross}} + d_{\text{safety}}$$

In practice the fundamental question we should anwser is wether we should proceed or yield the way w.r.t. this other vehicle. To handle this disjunctive constraint, we introduce a binary slack variable such that the OR constraint is replaced by an AND constraint

$$s\left(t_{\text{cross}}\right) \leq s_{\text{cross}} - d_{\text{safety}} + My \quad \wedge \quad s_{\text{cross}} + d_{\text{safety}} \leq s\left(t_{\text{cross}}\right) + M\left(1 - y\right)$$

with $y \in \{0, 1\}$ and $M \in \mathbb{R}^+$ some large value s.t. when y=1 the constraint is always true

This way, even if we have defined two constraints via a AND, which is required to apply optimization algorithms like Interior Point Methods or Simplex, only one or the other constraint will be active: the other one being always true. By using a binary slack variable, we have to use a Mixed Integer Programming solver.

This problem reformulation corresponds to the Big-M reformulation of disjunctive constraints.

4.1.2 Elastic Model

We would like to have a convex formulation of the problem such that we can find as quickly as possible a guaranteed global minimum. The problem is that when defining a problem with such a collision avoidance constraint

$$\min_x \quad Q_{\text{uadratic}}\left(x\right)$$
$$\text{s.t. } a^T x \leq b \text{ (safety distance constraint)}$$

This might be causing infeasibility. In practice there may be no dynamically feasible motion plan to maintain a pre-defined safety distance. But we want to reveal by how much the constraint needs to be relaxed in order to become dynamically feasible. We are looking for a Motion Plan that is dynamically feasible and which violates at minimum our desired safety distance. In order to reveal this value, we introduce another slack variable, per collision avoidance constraint , such that the problem becomes:

$$\min_x \quad Q_{\text{uadratic}}\left(x\right) + y$$
$$\text{s.t. } a^T x \leq b\text{+y (safety distance constraint)}$$
$$\text{elastic slack variable: } y \in \mathbb{R}$$

If we do not use such an elastic slack variable, a convex solver would return an infeasibility verdict and Interior Point Methods would fail.

4.2 Optimization Algorithms

We use the following generic optimization formulation and notations:

$$\begin{array}{cc} \min\limits_{\mathbf{x}\in\mathbb{R}^n} & f(\mathbf{x}) \in \mathbb{R} \\ \text{subject to} & \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \in \mathbb{R}^k \\ & \mathbf{h}(\mathbf{x}) = \mathbf{0} \in \mathbb{R}^m \end{array}$$

4.2.1 Penalty Methods

We first consider Penalty and Augmented Lagrangian methods [7] which transform the constrained problem into an unconstrained one via the use of the following penalty methods:

$$p_{\text{quadratic}}(x) = \sum_i \max(g_i(x), 0)^2 + \sum_j h_j(x) \quad p_{\text{Lagrange}}(x) = \frac{1}{2}\rho \sum_i h_i(x)^2 - \sum_i \lambda_i h_i(x)$$

The issue with these methods is that even if they may end up approaching a minimum, it may be from an infeasible region. This is why Interior Point Methods are usually preferred.

4.2.2 Interior Point Method with Inequality and Equality Constraints

We build upon the work that was done for AA222 Project 2. An Interior Point Method based on a Quasi-Newton method, BFGS, with backtracking Line search was implemented. But here we deal with equality constraints, of the form $Ax = b$ (corresponding mainly to our Vehicle Dynamics Model), in addition to inequality constraints. We modify the way we compute the search direction. At every step, we do a second order approximation of our minimization function. We express the Lagrangian $\mathcal{L}(x, \lambda)$ and solve for $\nabla_x \mathcal{L} = 0$. The solution of the resulting system of equations provides the new search direction: $d = \Delta x_{\text{newton\_step}}$, everything else being unchanged.

$$\begin{array}{c} \min \\ \text{subject to} \end{array} \begin{cases} \hat{f}(x+v) = f(x) + \nabla f(x)^T v + \frac{1}{2} v^T \nabla^2 f(x) v & \text{Second order Taylor approximation} \\ A(x+v) = b \end{cases}$$

Via optimality conditions on $\mathcal{L}(x, \lambda)$ we get: $\begin{bmatrix} \Delta x_{\text{newton\_step}} \\ \lambda \end{bmatrix} = \begin{bmatrix} \nabla^2 f(x) & A^T \\ A & 0 \end{bmatrix}^{-1} \begin{bmatrix} -\nabla f(x) \\ -(Ax - b) \end{bmatrix}$

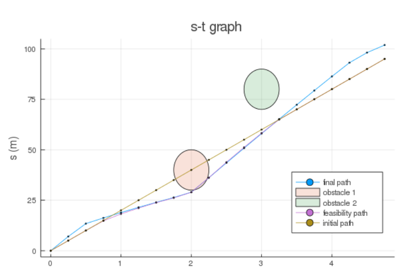
The details of the derivatoin can be found in [2] chapter 10 and our implementation in optimize.jl

4.2.3 Simplex Algorithm

Simplex is fast. We investigate how to bootstrap the feasibility search phase of an Interior Point method with a simplex algorithm.

4.3 Optimization under Uncertainty

We handle uncertainty as per the set-based minimax approach described in [7] chapter 17. We are dealing with imperfect observations of the surrounding vehicles and with even more uncertain driving models. As a consequence, the predicted crossing point $(t_{\text{cross}}, s_{\text{cross}})$ is uncertain. This uncertainty is represented by a random variable z and the crossing vehicle can be at any location within an uncertainty area represented as a circle in the (s, t) domain; the shadow area in the ST graph (longitudinal position S along the path vs Time).



We try to avoid the whole uncertainty area, complying to the minimax approach $\min_{x \in \mathcal{X}} \max_{z \in \mathcal{Z}} f(x, z)$. But if we can not avoid the full uncertainty area, we will remain as far as possible from its center, via the Elastic Collision Avoidance Model described previously. This Elastic Model strictly enforces dynamics constraints but relaxes the safety distance as little as possible.

5 Experiments

The gihub repo is AA222-project.

5.1 ST Graphs Analysis

- Runtime: ≤ 250 ms for real time applicability
- Feasibility constraints compliance: check safety & dynamics constraints
- Cost value: efficiency and comfort (lower cost function)

5.2 Anti Collision Tests Benchmarks

We use five metrics to evaluate the performance of our different approaches. (1) The main success metric is the percentage of cases where we reach a target state without collision. (2) The second metric is the agent runtime. (3) The third metric is a comfort metric: the number of hard braking decisions. (4) The fourth metric relates to efficiency: how fast we reach a target while complying to some speed limitation. (5) The last metric is a safety metric: for some of our randomly generated test cases, a collision is unavoidable. In these cases, we aim for a lower speed at collision.

6 Conclusion

It works even better than expected ...

References

[1] John T. Betts. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. Cambridge University Press, USA, 2nd edition, 2009.

[2] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, USA, 2004. 4.2.2

[3] Haoyang Fan, Fan Zhu, Changchun Liu, Liangliang Zhang, Li Zhuang, Dong Li, Weicheng Zhu, Jiangtao Hu, Hongye Li, and Qi Kong. Baidu apollo em motion planner. *ArXiv*, abs/1807.08048, 2018. 1

[4] T. Gu, J. M. Dolan, and J. Lee. Runtime-bounded tunable motion planning for autonomous driving. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pages 1301–1306, 2016.

[5] C. R. Hargraves and S. W. Paris. Direct trajectory optimization using nonlinear programming and collocation. In *Astrodynamics 1985*, pages 3–12, August 1986.

[6] Christos Katrakazas, Mohammed A. Quddus, Wen hua Chen, and Lipika Deka. Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions. 2015. 1

[7] Mykel J. Kochenderfer and Tim A. Wheeler. *Algorithms for Optimization*. The MIT Press, 2019. 4.2.1, 4.3

[8] X. Li, Z. Sun, Z. He, Q. Zhu, and D. Liu. A practical trajectory planning framework for autonomous ground vehicles driving in urban environments. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 1160–1166, 2015.

[9] C. Liu, W. Zhan, and M. Tomizuka. Speed profile planning in dynamic environments via temporal optimization. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 154–159, 2017.

[10] Changliu Liu, Chung-Yen Lin, and Masayoshi Tomizuka. The convex feasible set algorithm for real time optimization in motion planning. *SIAM J. Control and Optimization*, 56:2712–2733, 2017.

[11] Tianyu Gu, J. Atwood, Chiyu Dong, J. M. Dolan, and Jin-Woo Lee. Tunable and stable real-time trajectory planning for urban autonomous driving. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 250–256, 2015.

[12] Andreas Wachter and Lorenz Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106:25–57, 03 2006. 1

[13] Xiaojing Zhang, Alexander Liniger, and Francesco Borrelli. Optimization-based collision avoidance. *ArXiv*, abs/1711.03449, 2020. 1