Time-Optimal Trajectory Generation for Dynamic Vehicles: A Bilevel Optimization Approach

Gao Tang*, Weidong Sun*, and Kris Hauser

***Abstract*— This contribution presents a framework to find time-optimal trajectories for dynamic vehicles like drones and autonomous cars. Hindered by its nonlinear objective and complex constraints, this problem is challenging even for state-of-the-art nonlinear programming (NLP) solvers. The proposed framework addresses the problem by bilevel optimization. Specifically, the original problem is divided into a lower-level problem, which computes a time-optimal velocity profile along a pre-specified geometric path, and an upper-level problem, which optimizes the geometric path by a Quasi-Newton method. The lower-level problem is convex and efficiently solved by interior-point methods using a customized KKT solver with variable reordering. Then, the gradients of the objective function can be derived from the Lagrange multipliers using sensitivity analysis. The method is guaranteed to return a feasible solution at any time, and numerical experiments on a ground vehicle with friction circle model show that the proposed method performs more robustly than general nonlinear solvers.**

I. INTRODUCTION

Time-optimal trajectory generation is an important topic in robotics to increase task efficiency. Existing techniques can be categorized into direct collocation or two-stage approaches. The direct collocation approach optimizes a discretized representation of the trajectory, both positions, velocities, and controls. However, this approach has to handle nonlinear dynamics and non-convex constraints, requiring solution of a Nonlinear Programming (NLP) problem. There is no guarantee that an optimal, or even feasible solution is obtained. The two-stage approach is an approximation technique where a geometric path is optimized separately from the speed along the path. Each of these optimization steps is more numerically stable, and time-optimal path parameterization (TOPP) approaches have been specialized to quickly and robustly optimize the speed along the path. However, in the two-stage approach the path is fixed after the first stage, and cannot be optimized further to reduce trajectory time.

We introduce a novel bilevel optimization approach that robustly and efficiently solves the time-optimal trajectory generation problem. Our approach hierarchically solves the path optimization and speed optimization subproblems : the lower-level problem solves time-optimal path parameterization (TOPP) which computes a time-optimal velocity profile along a geometric path by nonlinear convex optimization, the upper-level problem refines the geometric path using a Quasi-Newton method to achieve lower traversal time.

Our algorithm is applied to a ground racing vehicle whose dynamics obey a friction circle model. Experiments demonstrate the robustness and any-time feasibility of our approach against state-of-the-art NLP solvers IPOPT [1] and SNOPT [2].

G.Tang and W.Sun are with the Department of Mechanical Engineering and Materials Science, Duke University. Durham, NC, 27708 USA. e-mail: {weidong.sun, gao.tang}@duke.edu.

K.Hauser is with the Departments of Electrical and Computer Engineering and of Mechanical Engineering and Materials Science, Duke University. Durham, NC, 27708 USA. e-mail: kris.hauser@duke.edu.

* Denotes equal contribution

II. METHOD

A. Problem Formulation

We want to find a trajectory that travels from the start to the goal set in a minimum amount of time while being collision-free, respecting the dynamics and other system constraints like bounds on velocity and acceleration. The problem can be mathematically formulated as:

$$\begin{aligned} \underset{q(t),u(t)}{\text{minimize}} \quad & T \\ \text{subject to} \quad & q(t) \in \mathcal{X}_{\text{free}}, \quad \forall t \in [0,T] \\ & q(0) = q_s, \\ & q(T) \in \mathcal{X}_{\text{goal}}, \\ & h(q(t), \dot{q}(t), \ddot{q}(t), u(t)) = 0, \quad \forall t \in [0,T] \\ & f(q(t), \dot{q}^2(t), \ddot{q}(t), u(t)) \le 0, \quad \forall t \in [0,T] \end{aligned} \tag{1}$$

where T is the traversal time, $q \in \mathbb{R}^n$ is the generalized configuration of the robot parametrized by time t, $\mathcal{X}_{\text{free}}$ denotes all the collision-free configurations, q_s is the start configuration, $\mathcal{X}_{\text{goal}}$ denotes the goal region, $u(t)$ is the control input, function $h(\cdot)$ encodes the dynamics constraints and $f(\cdot)$ includes other system constraints.

B. Algorithm

In this work, we compute time-optimal trajectories for dynamic vehicles on a racing track. The track is decomposed into several convex regions and the trajectory is represented using Bézier spline. The path constraint can thus be written as linear constraints on the spline control points.

An algorithm for solving bilevel optimization is presented in Algorithm 1. The algorithm takes an initial guess of the geometric path p_0 and the linear constraints on the path encoded in $Gp_0 \le h$ and $Ap_0 = b$. We note that linear constraints are sufficient in many cases. Nonlinear constraints can also be handled at the cost of any-time feasibility.

In each iteration, the TOPP solver takes a path p and an fixed optimality parameter μ arbitrarily chosen to be 10^{-4}, then outputs a cost J, Lagrange multipliers λ and time parameterization $\{b_i\}_{i=0}^{N}$. The TOPP solver will solve the linear complementary condition to this value instead of 0. By doing this, TOPP takes fewer iterations to converge, and the upper optimization problem gets smoother gradient. Besides, this also enables warm start of TOPP solver. This parameter is equivalent to the coefficient in log barrier method, but the primal-dual framework does not require a feasible initial guess. Gradient g of the path p is computed using Lagrange multipliers in the Get-Gradient function, and later used to update the path. Results from sensitivity analysis of parametric NLPs are used to compute the gradient. Any gradient-based method can be used as the Take-A-Step function which updates x based on gradient g and possibly its history (in Quasi-Newton approaches). We use an off-the-shelf NLP solver SNOPT to perform Take-A-Step function. Even though we are using a NLP solver, constraints in upper-level optimization are linear so feasibility is always guaranteed. Optimality conditions are checked by the NLP solver.

Algorithm 1 Bilevel-Solver (p_0, G, h, A, b, μ)

$p \leftarrow p_0$
for $i \leftarrow 0$ to *max-iterations* **do**
    $J, \lambda, \{b_i\}_{i=0}^N \leftarrow \text{TOPP}(p, \mu)$
    $g \leftarrow \text{Get-Gradient}(\lambda)$
    $x \leftarrow \text{Take-A-Step}(x, J, g, G, h, A, b)$
    if optimality-conditions-satisfied **then**
        break
return $p, \{b_i\}_{i=0}^N$

C. Efficient TOPP

The goal of TOPP is to find a time parameterization s, with the geometric path p already given, so that the traversal time is minimized while satisfying dynamic and other system constraints. It can be formulated as a convex optimization problem under appropriate assumptions [3], [4]. We apply three speed-up strategies to solve TOPP more efficiently:

1) Nonlinear Objective: Verscheure *et al.* [3] formulated TOPP as a SOCP, which is inefficient because the size of the original problem might even be doubled through the introduction of slack variables. In this paper, the nonlinear objective is directly solved.

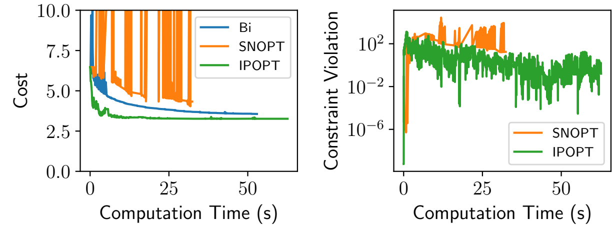
2) Eliminating Variables: To reduce the number of variables, we replace $\{a_i\}_{i=0}^{N-1}$ with a linear combination of $\{b_i\}_{i=0}^{N}$, i.e., $a_i = (b_{i+1} - b_i)/2\Delta s_i$.

3) Customized KKT solver: An efficient KKT system solver that exploits the structure of the problem allows significant acceleration. We apply a row and column permutation to the reduced KKT matrix to make it banded and use a LU decomposition implemented in SuperLU [5] to solve the system.

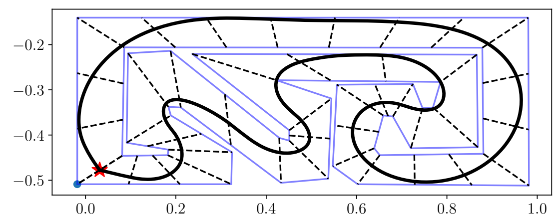
III. Experiments

We test our algorithm on a real-world racing track from the Tamiya Asia Cup Finals 2011[1], an RC car track with size 25 m by 11 m. We use a friction circle model [6], and the results are shown in Fig.1. Bilevel optimization achieves fast cost decrease in the early stage, but it converges slowly later. The NLP solvers fail to satisfy constraints throughout the optimization process. On the contrary, bilevel optimization is any-time and always feasible. The NLP solvers, especially IPOPT, decrease the cost function quickly, but as the right side of Fig. 1a shows, these solutions are indeed infeasible. This suggests high numerical sensitivity despite the existence of a feasible solution. In contrast, the stability of our method suggests that decoupling path optimization and time allocation results in better numerical stability.

[1]`http://quantumracing-rc.blogspot.com/2011/09/tamiya-asia-cup-finals-2011-in.html`, last accessed Apr. 8 2019



(a) History of cost and constraint violation



(b) Optimal geometric path and convex decomposition of the track

Fig. 1. (a): Cost and constraint are recorded whenever they are evaluated by the solver. The spikes are caused by line search with inappropriate step length. Nonlinear solvers fail to maintain feasibility even if starting from a feasible solution. (b): The black curve is the geometric path. The red star denotes the starting position.

References

[1] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, Mar 2006. [Online]. Available: https://doi.org/10.1007/s10107-004-0559-y

[2] P. E. Gill, W. Murray, and M. A. Saunders, "Snopt: An sqp algorithm for large-scale constrained optimization," *SIAM J. on Optimization*, vol. 12, no. 4, pp. 979–1006, Apr. 2002. [Online]. Available: http://dx.doi.org/10.1137/S1052623499350013

[3] D. Verscheure, B. Demeulenaere, J. Swevers, J. D. Schutter, and M. Diehl, "Time-optimal path tracking for robots: A convex optimization approach," *IEEE Transactions on Automatic Control*, vol. 54, no. 10, pp. 2318–2327, Oct 2009.

[4] K. Hauser, "Fast interpolation and time-optimization with contact," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1231–1250, 2014. [Online]. Available: https://doi.org/10.1177/0278364914527855

[5] X. Li, J. Demmel, J. Gilbert, iL. Grigori, M. Shao, and I. Yamazaki, "SuperLU Users' Guide," Lawrence Berkeley National Laboratory, Tech. Rep. LBNL-44289, September 1999, http://crd.lbl.gov/~xiaoye/SuperLU/. Last update: August 2011.

[6] T. Lipp and S. Boyd, "Minimum-time speed optimisation over a fixed path," *International Journal of Control*, vol. 87, no. 6, pp. 1297–1311, 2014. [Online]. Available: https://doi.org/10.1080/00207179.2013.875224