

On-Road Trajectory Planning for General Autonomous Driving with Enhanced Tunability

Tianyu Gu¹, John M. Dolan², and Jin-Woo Lee³

¹ Electrical & Computer Engineering (ECE), Carnegie Mellon University,
tianyu@cmu.edu

² ECE and Robotics Institute (CS), Carnegie Mellon University

³ Research & Development, General Motors*

Abstract. In order to achieve smooth autonomous driving in real-life urban and highway environments, a motion planner must generate trajectories that are locally smooth and responsive (reactive), and at the same time, far-sighted and intelligent (deliberative). Prior approaches achieved both planning qualities for full-speed-range operations at a high computational cost. Moreover, the planning formulations were mostly a trajectory search problem based on a single weighted cost, which became hard to tune and highly scenario-constrained due to overfitting. In this paper, a pipelined (phased) framework with tunable planning modules is proposed for general on-road motion planning to reduce the computational overhead and improve the tunability of the planner.

Keywords. On-Road Motion planning, Autonomous Passenger Vehicle

1 Introduction

The development of autonomous passenger vehicles made substantial progress in the 2007 DARPA Urban Challenge. Vehicle intelligence was achieved with various planning schemes, but a common three-layer planning architecture underlay the six entries that finished the competition [1]. **Route Planning** generated a task-level plan in the form of a sequence of global checkpoints to be reached to finish a task. **Behavior Planning** was responsible for making high-level maneuver decisions considering the complex traffic conditions, like lane change, yield to traffic, etc. **Motion planning** generated dynamically-feasible trajectories containing both spatial and temporal information for tracking control.

1.1 Prior work

Elastic band [8] and other algorithms based on numerical optimization have been widely used in other motion planning problems. For on-road autonomous driving, however, the road structure provides strong heuristics, hence sampling-based methods (with a suitable sampling pattern at reasonable density) are sufficient to produce a feasible solution. Moreover, a sampling-based method also gives us control over the scale of the search space, which is useful in pre-determining the run-time.

* This work was supported by General Motors.

The on-road motion planner of CMU’s Boss [2] spawned short-horizon ($\leq 30m$) trajectories by laterally shifting from the lane centerline and picking the closest-to-center collision-free trajectory for execution. This scheme could efficiently handle low-speed navigation in the simplified competition environment. However, its myopic nature could cause difficulties in realistic high-speed navigation on urban/highway roads. Several subsequent on-road planners adapted the spatial state-lattice concept originally used for unstructured environments to road structure [10][7] and constructed a spatiotemporal lattice augmented with time and velocity dimensions. The main idea was to exploit lattice search to increase the planning horizon and to introduce a deliberative quality into the formerly purely reactive planning.

In [7], the authors exhaustively generated and evaluated dynamically-feasible trajectories connecting from the vehicle’s current position to fixed longitudinal stations along the road over long spatial look-aheads with abundant speed variations. The planning was formulated as a trajectory ranking and search problem. The primary drawback was the need to tune multiple incommensurable cost terms to achieve a certain behavior, which is generically difficult. Meanwhile, costs tend to overfit the particular scenario for which they are tuned. The planner alone was therefore not able to perform general on-road driving. It required a higher-level decision maker to select the right cost set for different situations. Another practical disadvantage of this scheme was that, in order to retain reactivity, the trajectories were exhaustively sampled and evaluated in a high-dimensional space, which was computationally expensive.

In [3], the authors used a two-step planning approach that generated a “coarse” trajectory to capture the desired maneuver first, then generated a “fine” local trajectory to follow. The use of incommensurable costs posed a similar tuning and overfitting problem to that in [7]. In [4], the authors used optimization methods along with discrete search algorithms to generate a reference that can deal with different road geometry and static objects. The approach was limited to handling static objects, and the optimization routine could potentially cause undesirable reference jittering due to cycle-to-cycle execution. This paper extends the two-step planning work [3, 4] to adapt to more complex scenarios encountered in reality and to improve the formulation for tuning purpose.

1.2 Motivation

Prior work demonstrated reactive (short planning horizon in Boss [2]) and deliberative (long-horizon lattice-based exhaustive trajectory sampling [7]) planning schemes. The challenge is to achieve both planning qualities in a computationally efficient planning framework. Second, prior planning was performed by a single trajectory search process (i.e., ranking after sampling and evaluation of candidates). Ranking was based on an overall cost, obtained by the summation of multiple cost terms manually defined and tuned. Tunability was difficult to achieve because the tuning multiple incommensurable costs is nontrivial, and parameter overfitting (toward the scenario used for tuning) became inevitable.

We propose a planning framework (Fig. 1) for general autonomous on-road navigation. It transforms the single trajectory search problem into a pipelined (phased) planning process that adapts to a broad range of navigation situations:

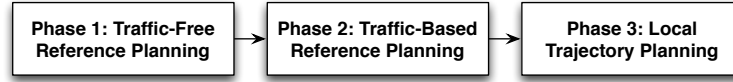


Fig. 1: Cascaded Planning Framework

Phase 1: Traffic-Free Reference Planning generates a traffic-free reference trajectory for each drivable lane assuming that the vehicle will follow the reference without any traffic interference. Only the geometry of the map is taken into account at this phase of planning.

Phase 2: Traffic⁴-Based Reference Planning makes use of three *maneuver planners* to generate a traffic-based reference by performing reference variation to respond to static and moving objects through lateral swerve, longitudinal speed variation or lane change.

Phase 3: Local Trajectory Planning generates a parametric trajectory for tracking control to carry out the planned reference. A model is used to evaluate vehicle response for dynamic feasibility and collision safety.

This framework exploits the independence between the stationary environment factors (traffic-free: e.g., road geometry) and the changing environment factors (traffic-based: e.g., obstacles) in shaping the overall plan. Decoupled planning phases are devised. At each planning phase, planning is further decomposed into smaller tunable planning problems. Parameters with clear physical interpretations are preferred over arbitrarily designed cost terms. Incommensurable cost parameters, if used, are carefully designed such that the effect of tuning and the planning output are well understood. The remainder of this paper is organized as follows: Sections 2, 3 and 4 describe the details of traffic-free, traffic-based and local planning respectively. Section 5 demonstrates a few testing scenarios to highlight the planning capabilities of our approach. Section 6 discusses the contributions and limitations of our work and future work.

2 Traffic-Free Reference Planning

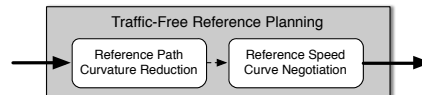


Fig. 2: Traffic-Free Reference Planning Sequence

Based on the map, which contains road waypoints and speed limit information, a continuous path curve and speed profile are generated to provide a smooth reference whenever the route plan is updated. To achieve this, two planning modules are executed sequentially, as illustrated by Fig. 2.

2.1 Reference Path Curvature Reduction

The map waypoints are first interpolated to generate lane reference centerlines L^{cl} . The goal is to reduce the maximum curvature of L^{cl} for both comfort and

⁴ In our terminology, traffic may refer to static obstacles and moving objects like pedestrians, bicyclists or surrounding vehicles.

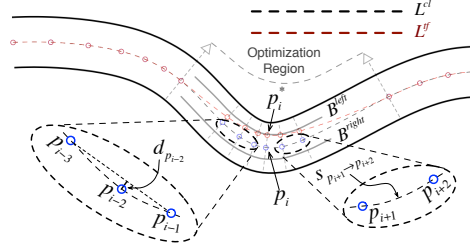


Fig. 3: Reference Curvature Reduction Formulation. The black dashed curve and the red dashed curve represent the lane reference centerline and curvature-reduced path, respectively. The blue and red circles are the sampled control points and the optimized control points, respectively.

control purposes. As illustrated by Fig. 3, a modified Douglas-Peucker algorithm is first used to adaptively sample the control points $\{\mathbf{p}_i\}$ on L^{cl} to be denser in curvy regions, such that the projected distance $d_{\mathbf{p}_i}$ from \mathbf{p}_i to the line connecting \mathbf{p}_{i-i} and \mathbf{p}_{i+i} is below a certain threshold d^{max} , and the longitudinal distance $s_{\mathbf{p}_i \rightarrow \mathbf{p}_{i+1}}$ between the two points \mathbf{p}_i and \mathbf{p}_{i+1} is always within the range $[s^{min}, s^{max}]$. The control points $\{\mathbf{p}_i\}$ are “nudged” laterally to obtain optimized control points $\{\mathbf{p}_i^*\}$, which are then interpolated to generate a curvature-reduced lane reference path L^{tf} :

$$\{\mathbf{p}_i^*\} = \underset{\{\mathbf{p}_i\}}{\operatorname{argmin}} \sum_{\text{all } i} C_{cr}(\mathbf{p}_i) \quad (1)$$

where

$$C_{cr}(\mathbf{p}_i) = \left\| \frac{\mathbf{p}_i - \mathbf{p}_{i-1}}{|\mathbf{p}_i - \mathbf{p}_{i-1}|} - \frac{\mathbf{p}_{i+1} - \mathbf{p}_i}{|\mathbf{p}_{i+1} - \mathbf{p}_i|} \right\|$$

The cost term C_{cr} measures the cumulative heading changes of the control points. The boundary for lateral nudges $[B^{right}, B^{left}]$ explicitly tunes the extent of optimization, which is applied to the control points whose curvature is above a certain threshold as represented by the grey region in Fig. 3.

2.2 Reference Speed Curve Negotiation

Based on the curvature-reduced reference path L^{tf} , a speed profile is generated as the speed plan. Imposing a few dynamic constraints with an iterative algorithm explained in [4] proves to be very effective in achieving smooth curve negotiations. The routine “chips away” the excessive dynamics starting from the speed limit. For each iteration, the preferred peak lateral acceleration a_{lat} is applied first to bound the maximum value along L^{tf} :

$$v \leq \inf \left\{ \sqrt{\frac{a_{lat}}{\kappa}}, v^{max} \right\} \quad (2)$$

where κ is the curvature of the traffic-free referene path.

Notice that $\sqrt{\frac{a_{lat}}{\kappa}}$ approaches/reaches singularity on low-curvature/straight segments as $\kappa \rightarrow 0$. Though capped by the speed limit v^{max} obtained from the

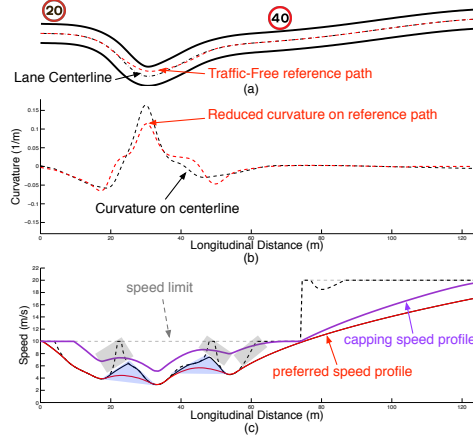


Fig. 4: Traffic-Free Reference Planning. (a) depicts a lane segment, its centerline and the curvature-reduced reference path. (b) compares the curvature plots of the centerline and the curvature-reduced reference path. (c) illustrates the preferred and capping speed profiles to compensate for lane curvature.

map, the speed profile is still unachievable since the singularity makes accelerations extremely high entering/exiting a turn (grey rectangular regions in Fig. 4c). Preferred longitudinal acceleration a_{lon} and deceleration d_{lon} are therefore applied:

$$d_{lon} \leq \dot{v} \leq a_{lon} \quad (3)$$

Still, acceleration jerks may be observed (blue triangular regions in Fig. 4c). A maximum longitudinal jerk j_{lon} constraint is therefore applied:

$$|\ddot{v}| \leq j_{lon} \quad (4)$$

The generated speed profile is denoted as ξ_{prefer}^* . The same algorithm can obtain a capping speed profile $\xi_{capping}^*$ (purple curve in Fig. 4c), that is on the verge of breaking vehicle dynamics constraints (e.g., tire grip) with a different set of parameters \bar{a}_{lat} , \bar{a}_{lon} , \bar{d}_{lon} and \bar{j}_{lon} . These parameters have strong physical meanings in characterizing speed variation, hence are intuitive to tune.

3 Traffic-Based Reference Planning

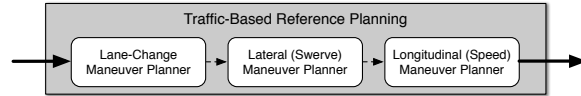


Fig. 5: Traffic-Based Reference Planning Sequence

Three planning modules are designed to sequentially vary the traffic-free reference to perform lane-change, lateral and longitudinal in-lane maneuvers to respond to commonly observed real-world objects, as shown in Fig. 5.

3.1 Lane Change Maneuver Planner

To avoid a lane blockage or overtake a slow-moving target, this maneuver planner generates a lane-change reference by connecting two traffic-free references of two parallel lanes (exit and entry lanes) with a concatenation reference whose start and end positions are specified by a high-level behavior decision module, such as the one described in [9]. If safety measurements are not met, an abort-lane-change maneuver will be generated by creating a reference from the current vehicle position to the departure lane for reentry.

3.2 Lateral (Swerve) Maneuver Planner

The objects whose predicted motion is static can usually be avoided with lateral swerve. The reference path L_{tf} is varied spatially to create a collision-free reference.

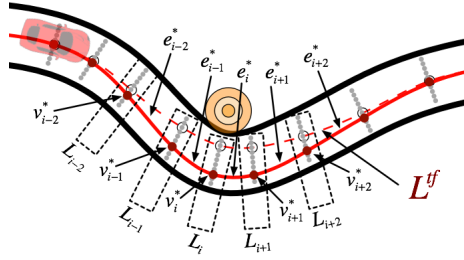


Fig. 6: Lateral Maneuver Planning. The thick black curves mark the lane boundary with an orange nearby obstacle. The dashed red curves represent the traffic-free reference path from the prior planning phase, around which the grey vertices are sampled and grouped with dashed rectangles. One red vertex in each group is obtained as the optimal vertex-to-visit. By interpolating all optimal vertices-to-visit, the solid red curve is generated as the swerve maneuver reference.

As shown in Fig. 6, an array of vertices consisting of multiple layers $\{L_i\}$ is generated based on L^{tf} . Instead of using a fixed sampling resolution as done in much prior work, the longitudinal sampling distance $\Delta S \in [\Delta S^{min}, \Delta S^{max}]$ between two neighboring layers is determined by the same modified Douglas-Peucker algorithm used in 2.1, and lateral uniformly sampled vertices on each layer are spaced by ΔL . Edge e_i connects a vertex v_i in layer L_i to a vertex v_{i+1} in layer L_{i+1} . The lateral shift of edge e_i and the lateral offset of vertex v_{i+1} from the traffic-free reference path are multiples of ΔL . The optimal sequence of vertices $\{v_i^*\}$ is defined to be optimal such that:

$$\{v_i^*\} = \underset{\{v_i \in L_i\}}{\operatorname{argmin}} \sum_{i=0}^{N-1} C_{action} + C_{offset} + C_{object} \quad (5)$$

where N is the index of the farthest layer, and

$$\begin{aligned} C_{action} &= \omega_{action} \cdot e^{\frac{|l(e_i)|}{\Delta L}} \\ C_{offset} &= \frac{|l(\mathbf{v}_{i+1}) - l^{tf}(\mathbf{v}_{i+1})|}{\Delta L} \\ C_{object} &= \begin{cases} 0 & \text{safe} \\ \infty & \text{unsafe} \end{cases} \end{aligned}$$

C_{action} induces an exponential penalty on excessive swerve. C_{offset} penalizes the deviation from the preferred reference path L^{tf} as a force of attraction. C_{object} takes the form of an infinite-or-zero cost for obstacle collision, which eliminates the extra complexity of crafting an arbitrary drop-off cost shape.

Dynamic programming is used to solve the search problem by calculating the optimal cost-to-go value in a backward recursive fashion. At times, no policy exists for all vertices at one layer, which is caused by the inability to swerve around obstacle(s). In this situation, the process is reset and replanning is initiated from this blocked layer, such that the farthest layer in consideration becomes the one just prior to the blockage. The selected optimal vertex sequence is then interpolated to reconstruct a smooth curve that passes through all the vertices (red curve in Fig. 6). Its shape may deviate from the traffic-free reference path, so the preferred/capping speed profiles need to be recalculated.

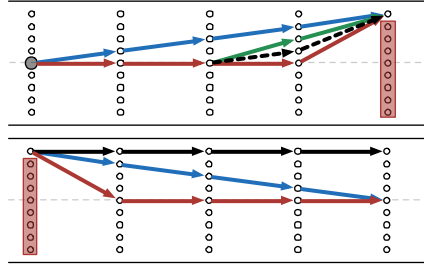


Fig. 7: Illustration of Cost Negotiation. The red route implies an abrupt swerve when the weight ω_{action} is too low, while the black route implies not getting back to the reference center after the swerve resulting from a high ω_{action} . Moreover, the exponential form of cost C_{action} favors the sequence with uniform actions, where the green route outshines the black dashed route. Proof is given by the fact that the minimum value of $e^x + e^{N-x}$ is obtained when $x = N/2$.

The goal is to generate a smooth sequence of actions (blue route in Fig. 7) that avoids obstacles and gets back to the traffic-free reference path. The weighting ω_{action} between C_{action} and C_{offset} should be tuned neither too low, nor too high (constant bias after avoidance, black route in Fig. 7). We refer to this conflict phenomenon as *cost negotiation*. Manual tuning can quickly become unmanageable if the number of weighting parameters increases. Sampling resolutions ΔS and ΔL are then adjusted to fine-tune the shape of the swerve.

3.3 Longitudinal (Speed) Maneuver Planner

In on-road navigation, longitudinal speed variation is the most common response in many situations. The proposed maneuver planner generates a traffic-based speed profile that converges to the preferred reference speed profile ξ_{prefer}^* , while reacting to other interfering moving objects.

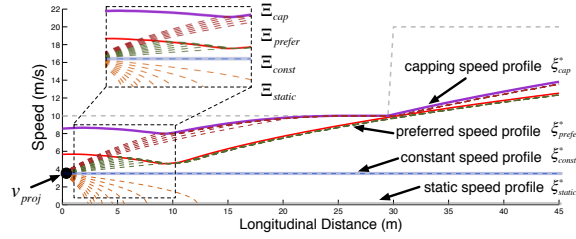


Fig. 8: Longitudinal Maneuver Planning. The purple, red, blue and grey solid curves represent capping, preferred, constant and static characteristic speed profiles. The black node marks the current longitudinally projected speed of the vehicle. Four clusters of candidate speed profiles in red, green, blue and orange dashed curves converge to the four characteristic speed profiles with varying accelerations.

As illustrated in Fig. 8, four characteristic speed profiles, the “preferred” ξ_{prefer}^* , the “constant” ξ_{const}^* , the “static” ξ_{static}^* and the “capping” ξ_{cap}^* are used to generate four clusters of sampling profiles Ξ_{prefer} , Ξ_{const} , Ξ_{static} , Ξ_{cap} . In each cluster, a few candidate profiles are generated to converge to the corresponding characteristic speed profile from the current longitudinal speed v_{proj} of the ego vehicle with a set of acceleration constants ranging from A^{min} to A^{max} at a sampling resolution ΔA . The aggressive cluster Ξ_{cap} will not be further evaluated if a feasible profile is found in the union $\Xi_{prefer} \cup \Xi_{const} \cup \Xi_{static}$. This order reflects the preference for a conservative response. The optimal profile is defined to satisfy:

$$\xi^* = \underset{\xi \in \Xi}{\operatorname{argmin}} C_{action} + C_{object} \quad (6)$$

where

$$C_{action} = e^{|a(\xi) - a^{sugg}|}$$

$$C_{object} = \begin{cases} 0 & \text{safe} \\ \infty & \text{unsafe} \end{cases}$$

$a(\xi)$ is the acceleration value of profile ξ ;

$$a^{sugg} = \begin{cases} a_{acc}^{sugg} & v_{proj} \text{ below } \xi_{prefer}^* \\ N/A & v_{proj} \text{ on } \xi_{prefer}^* \\ a_{dec}^{sugg} & v_{proj} \text{ above } \xi_{prefer}^* \end{cases} \quad (7)$$

C_{object} takes the form of an infinite-or-zero cost to prevent unsafe (possibly collision) navigation close to objects. We can project the object onto the lane

and calculate C_{object} based on its longitudinal gap to the ego vehicle:

$$C_{object}(\xi) = \sum_{i=0}^N c_{obj}(l(t_i)) \quad (8)$$

where $l(t)$ is the longitudinal distance gap between the ego vehicle and the predicted position of the moving objects at sampled time instant t_i , t_N represents the farthest lookahead horizon.

$$c_{obj}(l) = \begin{cases} \infty & 0 < l \leq L^{safe} \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

Since C_{object} is in a zero-or-infinite configuration, *cost negotiation* is avoided. The output of the planning can be clearly stated as “finding the viable speed profile that is closest to the suggested reference tracking acceleration”. Tuning can be done with a_{acc}^{sugg} and a_{dec}^{sugg} for adjusting the suggested effort of tracking, and L^{safe} for modifying the longitudinal safety distance.

4 Local Trajectory Planning

In this planning phase, we generate a dynamically feasible local trajectory to track the traffic-based reference from the current vehicle state (position, orientation, steering angle, speed, acceleration, etc.). A parametric trajectory χ makes use of the quintic polynomial path primitive [5][6] and the cubic polynomial speed primitive [4] that provide analytical continuity for the smoothness of tracking control.

Multiple trajectories $\{\chi_i\}$ are sampled in a focused pattern leading back to the traffic-based reference plan. As illustrated in Fig. 9, the trajectories start from the current vehicle state S^{init} and end at several lookahead states $\{S_i^{lh}\}$ on the reference, whose longitudinal distance ranges from S^{min} to S^{max} at the resolution of ΔS . With a model of subsequent controller and vehicle dynamics, the generated trajectories are forward-simulated to evaluate their validity. The predicted vehicle traces are used to guarantee the collision-free operation. The important dynamics metrics, including peak lateral acceleration a_{lat}^{peak} , peak longitudinal acceleration a_{lon}^{peak} , and peak longitudinal deceleration d_{lon}^{peak} , are used to eliminate the trajectories whose dynamics exceed the maximum endurable value. The chosen trajectory χ^* is the one with minimum converge time to the planned reference.

In emergency cases when no sampled trajectory is feasible, there are two possibilities: first, traffic-based reference planning may have failed to generate a valid plan⁵; second, it is impossible to get back to the traffic-based reference safely from the current vehicle state (e.g., the position of the ego vehicle deviates greatly from the reference path). In either case, an emergency planner is needed to plan an evasive trajectory, but this topic is out of the scope of this paper.

⁵ If traffic-based reference planning fails, it simply passes through the traffic-free reference, which results in collision.

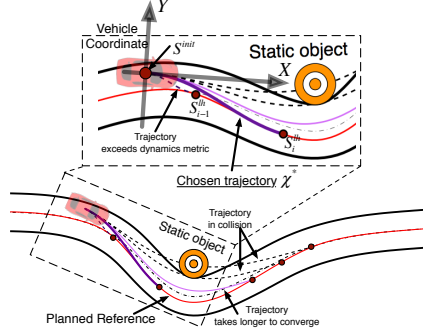


Fig. 9: Local Trajectory Planning. The red solid curve represents the traffic-free reference path, the black nodes on which are the sampled lookahead states. The local trajectories generated from current vehicle state to these sampled states are represented by dashed black curves if they are invalid and by purple curves if they are valid.

5 Results

5.1 Computation

In traffic-free reference planning, nonlinear optimization and an iterative numerical algorithm are used. The nonlinear optimization uses the Nelder-Mead algorithm. The best-so-far result which is at least as good as the original center-line reference, is used if convergence is not achieved within the bounded runtime T_{tf} . The iterative method used for speed profile generation typically converges within a few iterations, given its pure constraining nature. A fixed number N_{tf} of iterations are performed.

In traffic-based reference planning, the different maneuver planners generate different search spaces. For the lateral maneuver planner, longitudinal and lateral horizons S^H and L^H are determined by the speed limit and lane width, respectively. Dimension resolutions, ΔS^{min} , ΔS^{max} and ΔL , are chosen with a tradeoff between expressiveness and computation. In our implementation, $(S^H, \Delta S^{min}, \Delta S^{max}, L^H, \Delta L) = (200.0, 5.0, 10.0, 4.0, 0.2)$, so the upper-bound number of path edges generated and evaluated is $\frac{S^H}{\Delta S^{min}} \cdot (\frac{L^H}{\Delta L})^2 = 16,000$. For the longitudinal maneuver planner, there is a range limit of the acceleration constant for all four clusters from maximum allowed longitudinal acceleration A^{max} to the deceleration A^{min} with a constant acceleration sample resolution ΔA . In our implementation, $(A^{max}, A^{min}, \Delta A, N^{cluster}) = (2.0, -4.0, 0.1, 4)$. A pessimistic estimate of the maximum number of evaluated profiles is $N^{cluster} \cdot \frac{A^{max} - A^{min}}{\Delta A} = 240$, since not all profiles will converge to the characteristic profile of their cluster. Comparing to [10] [7], whose number of evaluated parametric trajectories is respectively 170,000 and 400,000 in similar use cases, the computation requirements of both maneuver planners presented here are minuscule.

In local trajectory planning, we sampled trajectories that connect the current vehicle state to a range of lookahead states on the reference whose longitudinal

distance ranges from S^{min} to S^{max} with a station increment ΔS . In our implementation, $(S^{min}, S^{max}, \Delta S) = (5.0, 60.0, 1.0)$, so the number of trajectories to be evaluated is $\frac{S^{max} - S^{min}}{\Delta S} = 55$, which is computationally very cheap to generate and evaluate.

5.2 Experiment

The proposed planning framework has been evaluated both in urban and highway environments. In the urban scenario, the planner demonstrates the execution of a sequence of maneuvers in both simulation and reality, including overtaking by lane change, static obstacle avoidance and pedestrian yielding, as shown in Fig. 10. The on-vehicle experiment is performed in a closed parking lot near General Motors Technical Center at Warren, MI.

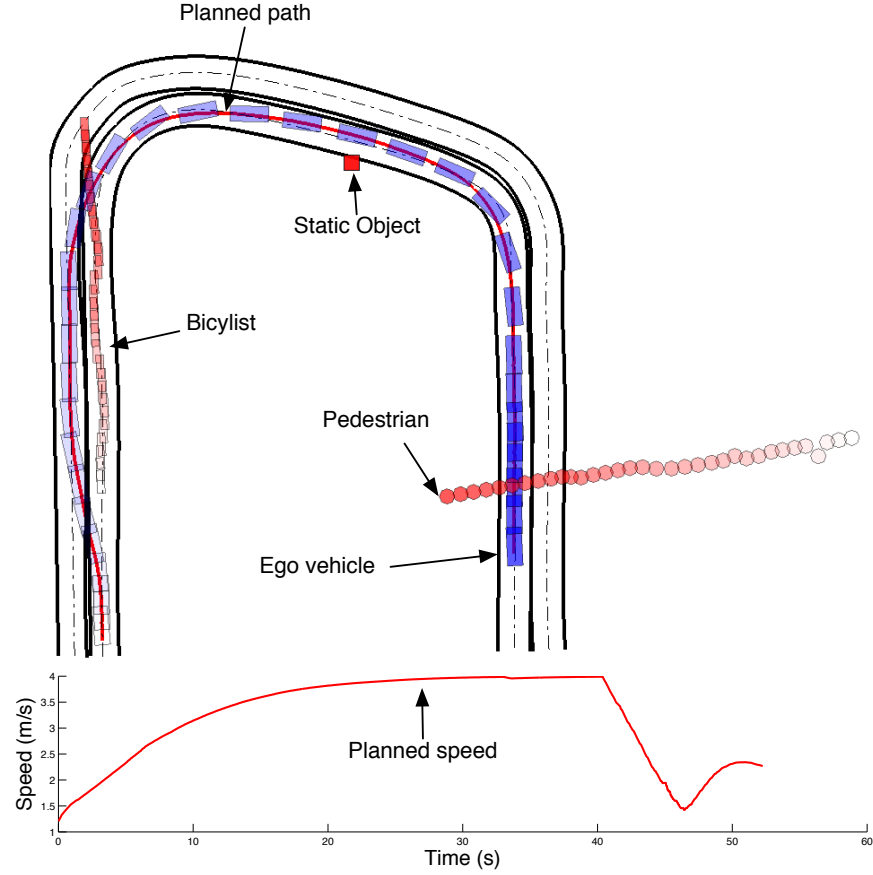


Fig. 10: On-vehicle planning result in an urban scenario. The red curve is the planned path to execute, while the red plot in the lower figure is the planned speed to execute. The planning involves overtaking an bicyclist, lateral swerve obstacle avoidance and yielding to pedestrian in sequence.

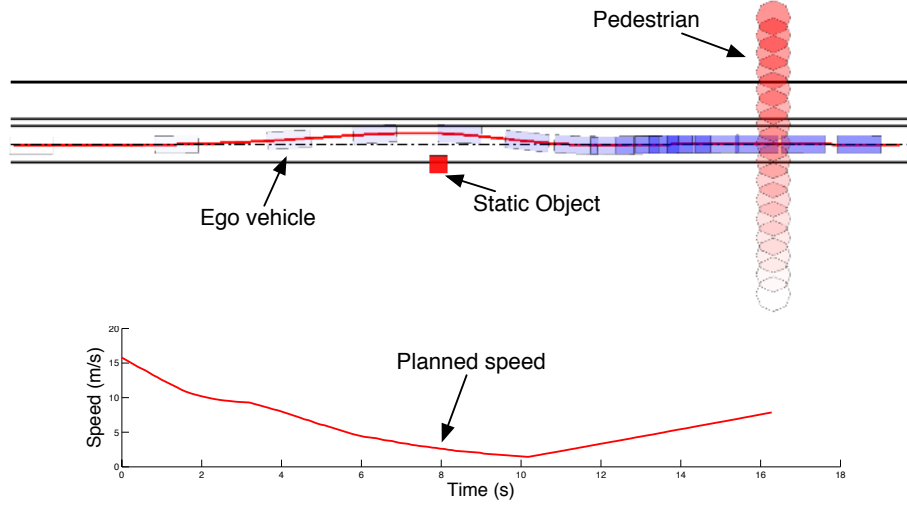


Fig. 11: Simulation planning result in highway scenario 1. The red curve is the planned path to execute, while the red plot in the lower figure is the planned speed to execute. The planning involves swerving around a static obstacle and yielding to a pedestrian.

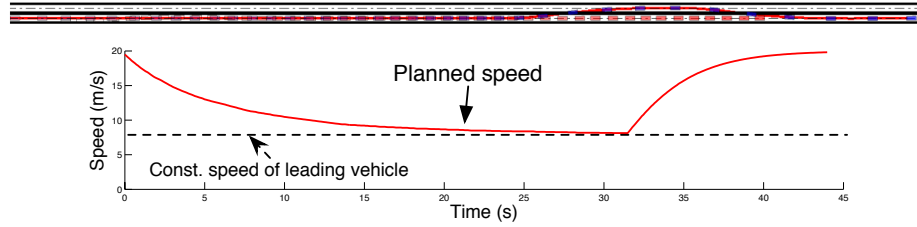


Fig. 12: Simulation planning result in highway scenario 2. The red curve is the planned path to execute, while the red plot in the lower figure is the planned speed to execute. The planning involves leading vehicle distance keeping and overtaking by consecutive lane changes.

In the highway scenario 1 shown in Fig. 11, the planner slows down and swerves around the static obstacle and yields to a slow-moving pedestrian. In the highway scenario 2, the planner demonstrates the capability to perform distance keeping to a leading vehicle, and overtaking by executing two lane changes consecutively.

6 Discussion

In this paper, a three-phase cascaded motion planning system is developed for general urban/highway on-road navigation. Both deliberative and reactive planning quality are maintained in this framework. Meanwhile, we focus on the enhancement of planning tunability by reducing the high-dimensional planning problem into multiple lower-dimensional ones with clear planning goals. We make use of various constraining parameters with clear physical meanings. Meanwhile, in subproblems where weighting of incommensurable costs is “negotiable”, our approach tunes for clear patterns in several planning steps in succinct formulations. Further can be achieved by adjusting the sampling pattern of search space.

Future work includes testing on a broader range of scenarios in reality. The learning of multiple tuning parameters from human demonstrations is also worth investigation to plan human-like results based on individual preferences. Another important question is whether depending on higher-level lane-change directives is sufficient to deal with real-world traffic, especially when a lane-change has to be performed in heavy traffic, where precise vehicle response of the vehicle’s motion should be considered when initiating (or aborting) the lane change.

References

1. Martin Buehler, Karl Iagnemma, and Sanjiv Singh, *The darpa urban challenge: Autonomous vehicles in city traffic*, vol. 56, springer, 2009.
2. Dave Ferguson, Thomas Howard, and Maxim Likhachev, *Motion Planning in Urban Environments: Part I*, International Conference on Intelligent Robots and Systems (2008), 1063–1069.
3. Tianyu Gu and John Dolan, *On-Road Motion Planning for Autonomous Vehicle*, Intelligent Robotics and Applications (2012), 588–597.
4. Tianyu Gu, Jarrod Snider, Jin-Woo Lee, and John Dolan, *Focused Trajectory Planning for Autonomous On-Road Driving*, 2013 IEEE Intelligent Vehicles Symposium.
5. Jin-Woo Lee and Bakhtiar Litkouhi, *Control and Validation of Automated Lane Centering and Changing Maneuver*, ASME Dynamic Systems and Control Conference (2009).
6. Jin-Woo Lee and Bakhtiar Litkouhi, *A unified framework of the automated lane centering/changing control for motion smoothness adaptation*, The 15th IEEE Intelligent Transportation Systems Conference (2012).
7. Matthew McNaughton, *Parallel algorithms for real-time motion planning*, Ph.D. thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, July 2011.
8. Sean Quinlan et al., *Elastic bands: connecting path planning and control*, IEEE International Conference on Robotics and Automation (1993), 802–807.
9. Junqing Wei, John M Dolan, and Bakhtiar Litkouhi, *A prediction-and cost function-based algorithm for robust autonomous freeway driving*, Intelligent Vehicles Symposium (IV), 2010 IEEE, IEEE, 2010, pp. 512–517.
10. Julius Ziegler and Christoph Stiller, *Spatiotemporal state lattices for fast trajectory planning in dynamic on-road driving scenarios*, The International Conference on Intelligent Robots and Systems (2009).