Safe Planning for Self-Driving Via Adaptive Constrained ILQR

Yanjun Pan[1], Qin Lin[2], Het Shah[3], and John M. Dolan[2]

Abstract— Constrained Iterative Linear Quadratic Regulator (CILQR), a variant of ILQR, has been recently proposed for motion planning problems of autonomous vehicles to deal with constraints such as obstacle avoidance and reference tracking. However, the previous work considers either deterministic trajectories or persistent prediction for target dynamical obstacles. The other drawback is lack of generality - it requires manual weight tuning for different scenarios. In this paper, two significant improvements are achieved. Firstly, a two-stage uncertainty-aware prediction is proposed. The short-term prediction with safety guarantee based on reachability analysis is responsible for dealing with extreme maneuvers conducted by target vehicles. The long-term prediction leveraging an adaptive least square filter preserves the long-term optimality of the planned trajectory since using reachability only for long-term prediction is too pessimistic and makes the planner over-conservative. Secondly, to allow a wider coverage over different scenarios and to avoid tedious parameter tuning case by case, this paper designs a scenario-based analytical function taking the states from the ego vehicle and the target vehicle as input, and carrying weights of a cost function as output. It allows the ego vehicle to execute multiple behaviors (such as lane-keeping and overtaking) under a single planner. We demonstrate safety, effectiveness, and real-time performance of the proposed planner in simulations.

I. INTRODUCTION

On-road autonomous vehicles require obstacle motion prediction to efficiently plan for future trajectories. Though motion planning has been extensively studied in recent decades, safe planning with assurance in highly uncertain environments is still challenging, especially for safety-critical systems such as autonomous vehicles. One problem arises from the highly uncertain trajectories of surrounding vehicles due to sensing, localization, maneuver or intention uncertainties, etc. Many existing works are only capable of providing statistical guarantees in belief space. Constrained Iterative Linear Quadratic Regulator (CILQR) has been proposed for collision avoidance for autonomous vehicles [1], [2]. Unfortunately, previous work only demonstrates the validation of the optimization framework by considering deterministic trajectories or persistent prediction for target vehicles. *The first research question raised in this study is how to incorporate a sophisticated prediction for target vehicles into the Iterative Linear Quadratic Regulator (ILQR) framework and assure safe planning considering prediction uncertainty.*

The second challenge the existing CILQR framework faces is the tuning of the cost function weights. Cost function weights heavily influence planned trajectory. For instance, if the reference tracking term in the cost function dominates the velocity term, a lane keeping behavior behind a slow vehicle is encouraged. If the dominance reverses, overtaking behavior is encouraged instead. Unfortunately, there is no study about the weight tuning problem in CILQR. *The second research question we aim to address is how to design a scenario-aware mechanism to adaptively generate appropriate weights for diverse scenarios.*

For the first question, we use reachability analysis to enclose all kinematically possible trajectories of a target vehicle into an envelope represented by a set considering the uncertainty of its initial states (position, yaw, etc.). We allow the target vehicle to behave non-dermistically over the short-term prediction horizon. Using this envelope, a motion planner can safely avoid all possible extreme scenarios in the short term. However, as the size of the envelope grows with time, it might block the entire traversable path, which explains why we use it only for short-term prediction. To avoid being overly conservative and avoid a local optimum, an adaptive least square filter is used for long-term prediction. This part can be easily substituted by more complicated and more accurate predictions. Our optimization framework solves the collision avoidance problem in terms of occupancy from the short-term and the long-term prediction in a unified manner. To address the second question, we design a scenario-based analytical function taking the states from the ego vehicle and the target vehicle as input, and carrying weights of a cost function as output. It allows the ego vehicle to execute multiple behaviors (such as lane-keeping and overtaking) under a single planner.

Our work makes the following contributions:
1) We incorporate a hybrid prediction crossing short-term and long-term levels for a safety-assured and efficient planner under highly uncertain surrounding traffic environments.
2) We propose an analytical function based on ego and target vehicle state difference and target vehicle velocity. This function increases the generality of a single planner under the same set of cost function weights.
3) The whole framework shows real-time performance with average cost around 65ms over a 4s prediction horizon in simulations with a 0.1-second discretization step.

The rest of this paper is organized as follows. Section II

[1]Yanjun Pan is with Department of Mechanical Engineering, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213, USA `yanjunp@andrew.cmu.edu`
[2]Qin Lin and John Dolan are with Department of the Robotics Institute, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213, USA `qinlin,jdolan@andrew.cmu.edu`
[3]Het Shah is with Department of Mechanical Engineering, Indian Institute of Technology Kharagpur, Kharagpur, West Bengal 721302, India `hetshah036@gmail.com`

provides a review of some important related work. Section III is the detailed framework proposed in this paper and necessary related background materials. Section IV is about the problem formulation relating to the cost function used and the vehicle dynamics. Section V presents the experimental results. The conclusions and the future work are in Section VI.

II. RELATED WORK

A. Trajectory optimization-related

Trajectory optimization can be formulated as a constrained quadratic program [3]. Ziegler et al. [4] used Sequential Quadratic Programming (SQP) to solve the nonlinear and non-convex problem, but the computation time is around 0.5s, which is not suitable for real-time implementation. Xu et al. [5] proposed an efficient real-time motion planner with trajectory optimization that discretizes the plan space and selects the best trajectory based on a cost function.

Control methods such as Model Predictive Control (MPC) have also been used for this purpose. Two-layer MPC [6], which reduces the computation cost as compared to one-layer MPC, has been proposed for motion planning with obstacle avoidance. Arab et al. [7] presented a Sparse-RRT* and nonlinear MPC-based motion planner for aggressive maneuvers. Ji et al. [8] present a path planning and tracking framework using 3D potential fields and multi-constrained MPC.

B. ILQR-related

Iterative linear quadratic regulator (ILQR) is an optimization-based method for nonlinear systems with lower computation time by utilizing dynamic programming. However, very few works have considered constraints in ILQR. Control-limited differential dynamic programming (DDP), proposed in [9], deals with upper- and lower-bound constraints on control inputs. Extended LQR [10] - [11] works with collision avoidance for circular obstacles, penalizing the distance from the center of the obstacle. Chen et al. proposed a constrained ILQR for autonomous vehicle motion planning [1]. It includes an elliptical over-fitting for obstacles and uses barrier functions to incorporate various constraints into the cost function. Chen et al. [2] proposed an improved barrier function to impose hard constraints on the optimization process. However, this does not solve the problem of weight-tuning. Another significant drawback in [1], [2] is that they only consider deterministic trajectories or persistent prediction for target vehicles.

C. Prediction-related

Readers are referred to a comprehensive literature review about trajectory prediction in Ref. [12]. Lin et al., propose a monitor to verify a black-box waypoint-following planner using online reachability analysis. The monitor warns unsafe situations when the predicted reachable state has a large deviation from the planned path [13]. Another similar online verification work for a learning-based cruise controller predicts and verifies if a bad state with insufficient relative

longitudinal distance is reachable [14]. for Althoff et al., perform an online verification of an existing planner using reachability analysis [15]. They propagate the reachability of an ego vehicle and its surrounding vehicles simultaneously over the whole planning horizon to verify the overlap. A re-planning step is needed if the path is verified to be unsafe. However, in our framework, the reachability has been plugged into the optimization, i.e., the planned trajectory has already been guaranteed to be safe to execute without re-planning. In addition, we argue for the use of a short-term horizon instead of a whole planning horizon for safety to avoid over-conservative planning. Our inspiration is the combination of short-term and long-term planning in [16]. They use two-layer planners, whereas we propose to use a single planner to combine short-term and long-term planning into a unified framework to improve the efficiency. A detailed analysis of the difference is given in the methodology section.

III. METHODOLOGY

A. Prediction of dynamic obstacles

The idea of hybrid prediction and planning is inspired by Liu et. al [16]. The authors proposed to combine a safety-oriented short-term planner and an efficiency-oriented long-term planner to deal with a safe human-robot interaction problem. The short-term planner considers the prediction uncertainty of the human trajectory, which is critical for safety. However, the disadvantage is that the uncertainty will propagate over time to make the planner excessively conservative. The prediction uncertainty is not considered in the long-term planner for the sake of efficiency. In every iteration, the overlap between long-term planned trajectory and short-term reachable region is checked. The long-term planning will be executed if no interaction is detected, i.e., the long-term planner is already safe. Otherwise they switch to the short-term planner. To further improve the efficiency, we propose to optimize the short-term and long-term collision avoidance at the same time in a unified framework without having two separate planners and extra intersection checking. A high-level illustration is shown in Figure 1. The short-term prediction considers the uncertainty of the target vehicle's initial state (e.g., perception error for the localization) and the uncertainty of its control actions over the short-term prediction horizon under a kinematically feasible but possibly non-deterministic assumption. The reachable positions are illustrated as rectangles. Note that they will be further inflated into ellipses considering the vehicle's shape. The long-term prediction does not consider the uncertainty. Each prediction mass point position will also be inflated into an ellipse. Another significant difference is that they use a tailored convex optimization framework by leveraging safe set and convex feasible set.

1) Short-term prediction with reachability analysis: In this paper we use a prototype continuous model checker based on the library of Flow* [17], which is a nonlinear reachability analysis tool. Given the uncertainty of states and a prediction horizon, our tool is able to compute the
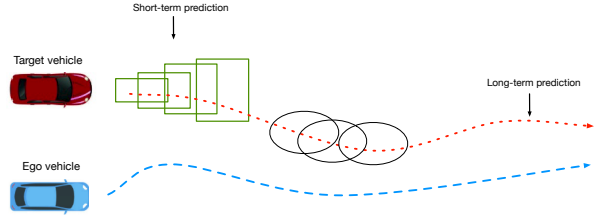
Fig. 1: High-level illustration for short-term and long-term prediction for planning. For a 4s predictive horizon, we use reachability in the first 0.5s, and an adaptive filter in the remaining 3.5s for prediction, respectively. The sampling time is 0.1s, thus we will have 40 ellipses as obstacles for collision avoidance.

reachable states. All possible future trajectories will fall within the reachable region with theoretical guarantee, since the reachability is an over-approximation of the dynamics.

The dynamics of a plant are defined as ordinary differential equations (ODEs), i.e., $\frac{dx}{dt} = F(x, t)$. The reachability analysis essentially tries to solve an *initial value problem*, i.e., given the interval of an initial state $x_0 = [\underline{x}_0, \bar{x}_0] \in \mathcal{X}_0 \in \mathcal{I}^n$ at time 0 and a time interval $[0, t]$, we compute the enclosure $\mathcal{X}_{[0,t]}$ as the explicit solution of F, which is only possible for simple ODEs. Iterative algorithms in Flow* derive a Taylor model consisting of polynomial and safe remainder to over-approximate the solution of F.

Definition 3.1: (Taylor Model): A Taylor Model (TM) of order k over domain D is denoted by a pair $(p, \mathcal{I})$, wherein p is a Taylor polynomial of degree at most k, and I is a *remainder interval*. Given a TM $(p, \mathcal{I})$ and a function f which are both over the same domain D, f is over-approximated by $(p, \mathcal{I})$, denoted by $f \in (p, \mathcal{I})$, i.e., $f(x) \in p(x) + \mathbb{I}, \forall x \in D$.

In this work, we consider the uncertainty of initial states of the target vehicle's x position, y position, and yaw. Their initial bounds are user-defined parameters due to perception errors. Note that the future control actions for the target vehicle over the prediction horizon are not observable. To ensure the safety guarantee, we augment ODEs for the control actions, i.e., $\dot{v} = \mathcal{I}_v$ and $\dot{\delta} = \mathcal{I}_\delta$. The upper bound and the lower bound set in our paper are $\mathcal{I}_v = [-4, 6]$ and $\mathcal{I}_\delta = [-0.1, 0.1]$ [18]. The interpretation for the velocity interval is that the maximum deceleration and acceleration are $-4 \ m/s^2$ and $6 \ m/s^2$ respectively. The δ interval is the maximal change in the steering angle within a second.

2) Long-term prediction with recursive least squares filter: In this paper, we use an adaptive linear autonomous dynamical model: $\dot{x} = A_t x$, where $x = [p_x, p_y, \theta]^T$. A_t is a time-variant parameter matrix. With the new observation of a true state, A_t is updated accordingly. The cost function and its gradient form are as follows:

$$\min_{\zeta} L_k = \frac{1}{2} \|x_k - f_\zeta(x_{k-1})\|^2 + \lambda L_{k-1} \quad (1)$$

$$\nabla_\zeta L_k = -\nabla_\zeta f_\zeta(x_{k-1})^T [x_k - f_\zeta(x_{k-1})] + \lambda \nabla_\zeta L_{k-1} = 0 \quad (2)$$

where ζ is a vector containing the parameters in A_t we want to optimize, f is the predictive model, and λ is the forgetting factor, which can be tuned as an adaptive rate. The explicit update form can be found in the literature [19], [20]. The advantage of such a linear filter is its efficiency and adaptation ability. The disadvantage is that kinematic constraints, road geometry, and drivers' intentions are not considered. Sophisticated machine learning-based prediction methods can be used to replace it if desired.

B. Iterative Linear Quadratic Regulator

Autonomous vehicle obstacle-free motion planning (i.e., lane-keeping) can be formulated as a standard ILQR problem with nonlinear system dynamics:

$$\min_{x,u} \quad J = x_N^T Q_N x_N + x_N^T p_N x_N + q_N$$
$$+ \sum_{t=0}^{N-1} (x_t^T Q_t x_t + x_k^T p + u_t^T R_t u_t) \quad (3)$$
$$\text{s.t.} \quad x_{t+1} = f(x_t, u_t) \quad (4)$$

where x_t and u_t are the state and the control input at time step t. System dynamics constraint (4) is a transition function mapping state and control at step t to state at step $t + 1$.

Since the standard LQR only solves optimization problems with quadratic cost and linear systematic constraints, this problem can be reformulated. By linearizing the systematic constraint at multiple points, we can relax the nonlinearity of the ILQR problem into the linear problem required by LQR.

Although dynamic programming provides an efficient solution for ILQR, ILQR itself has the drawback of its constraint-free nature, which is insufficient for collision avoidance of a motion planning problem.

C. Constrained Iterative Linear Quadratic Regulator

The constrained ILQR (CILQR) algorithm offers the inclusion of different constraints into the objective function through barrier functions. Constraints can be generalized into two categories by linearity. First, any nonlinear constraints can be converted to linear constraints via a first-order Taylor Expansion. Then, a barrier function is applied and quadratized. This process is demonstrated in (5) and (6). The quadratized linear barrier function can now be incorporated into the ILQR algorithm.

An exponential barrier function is chosen, as it has easy-to-derive analytic derivatives in contrast to use of a high-time-complexity finite difference method. The detailed algorithm can be found in [1].

$$b_k^x = q_1 exp(q_2 f_k^x) \quad (5)$$

$$b_k^x(x_k + \delta x_k) \approx \delta x_k^T \nabla^2 b_k^x(x_k) \delta x_k +$$
$$\delta x_k^T \nabla b_k^x(x_k) + b_k^x(x_k) \quad (6)$$

where f_k^x is the constraint function at time step k.

One drawback of the CILQR is its need to have specific sets of weights tuned for specific sets of scenarios. For

instance, using a set of weights tuned for an overtaking maneuver will always tend to encourage overtaking even if the target vehicle velocity is high. To improve the generality of the original CILQR and decrease the dependence on behavior-level planning, we propose analytical functions that take account of the state difference between the ego vehicle and the target vehicle along with the velocity of the target vehicle. These functions compute cost function weight adaptively and allow the planner to perform multiple tasks without switching weights.

D. Adaptive Weight Tuning Function

Intuitively, we want a lane keeping behavior when the forward target vehicle is far away from the ego vehicle irrespective of the speed of the target vehicle. When the distance gap closes, if the target vehicle is travelling close to the desired reference speed, a lane keeping behavior is desired. If the same target vehicle is travelling at a slow speed, a safe overtaking maneuver is desired. As a result, behaviors are closely related to target vehicle velocity and the relative distance between the ego vehicle and the target vehicle.

Based on this observation, weights that affect the velocity and reference tracking term are expressed as follows:

$$\omega_{vel} = \frac{a_1}{v_{tgt}} \exp\left(b_1 \Delta x\right) \tag{7}$$

$$\omega_{ref} = \frac{v_{tgt}}{a_2 \exp\left(b_2 \Delta x\right)} \tag{8}$$

where v_{tgt} is the target vehicle speed, Δx is the distance difference between target and ego vehicles, and a_1, a_2, b_1, and b_2 are parameters to tune. When Δx is large, a velocity-keeping behavior is encouraged due to the large velocity weight (see (7)). When Δx is small, depending on the target vehicle's velocity, overtaking for a slow vehicle and lane-keeping for a fast vehicle are encouraged, respectively (see (8)).

IV. PROBLEM FORMULATION

A. System Dynamics

The kinematic bicycle model is used in this paper. The state consists of $x = [p_x, p_y, v, \theta]^T$ and the control input is $u = [a, \delta]^T$, where p_x, p_y represent the position coordinates of the rear wheel, v is the vehicle velocity, θ is the orientation of the vehicle, and a and δ are the acceleration and steering angle of the vehicle respectively. The transition function in (9) represents with non-zero steering angle. The horizontal increment is calculated by $\int_0^l \cos(\theta_t + \kappa s)ds$ and the vertical increment is calculated by $\int_0^l \sin(\theta_t + \kappa s)ds$. $\kappa = \frac{\tan(\delta)}{L}$ is the curvature and $l_t = v_t dt + \frac{1}{2} a dt^2$ is the distance travelled at time t with discretization step dt.

$$
\begin{aligned}
p_{x,t+1} &= p_{x,t} + \frac{\sin(\theta_t + \kappa l) - \sin\theta_t}{\kappa} \\
p_{y,t+1} &= p_{y,t} + \frac{\cos\theta_t - \cos(\theta_t + \kappa l)}{\kappa} \\
v_{t+1} &= v_t + adt \\
\theta_{t+1} &= \theta_t + \kappa l
\end{aligned} \tag{9}
$$

However, when δ is zero, κ will be zero, which will a cause numerical issue in (9). Hence, a different update shown in (10) is used.

$$
\begin{aligned}
p_{x,t+1} &= p_{x,t} + \left(v_t dt + \frac{1}{2} adt^2\right) \cos(\theta_t) \\
p_{y,t+1} &= p_{y,t} + \left(v_t dt + \frac{1}{2} adt^2\right) \sin(\theta_t) \\
v_{t+1} &= v_t + adt \\
\theta_{t+1} &= \theta_t
\end{aligned} \tag{10}
$$

B. Objective Function

The general definition of the objective function in (3) can be made specific in (11). Each term in the summation represents the control effort cost, the reference trajectory tracking, the reference velocity tracking, and the constraints cost (i.e., control limits and obstacle avoidance). c_N is the end state cost, which is the sum of the latter three terms, as there is no control effort at the end state.

$$J = \sum_{t=0}^{N-1} \left(c_t^u + c_t^{ref} + c_t^{vel} + c_t^{con}\right) + c_N \tag{11}$$

1) control effort cost: The acceleration and the steering angle are weighted differently as shown in (12), but they are both judged against zero, as we desire a minimum amount of control effort.

$$c_t^u = u_t^T \begin{bmatrix} w_a & \\ & w_\delta \end{bmatrix} u_t \tag{12}$$

2) reference tracking and velocity cost: The reference tracking term assigns a cost to each state based on its distance to the closest point on the reference trajectory. The closest distance can either be calculated against the nearest waypoint, or it can be calculated based on its projection onto the reference trajectory. The velocity cost penalizes the ego vehicle for the difference between its velocity and the reference velocity. The combined cost is written in matrix form in (13). Δx_t is the difference between the ego vehicle state and the reference state at time t.

$$\Delta x_t = x_t - \begin{bmatrix} p_{x,t}^{ref} & p_{y,t}^{ref} & v_t^{ref} & 0 \end{bmatrix}^T$$

$$c_t^{ref} + c_t^{vel} = \Delta x^T \begin{bmatrix} w_{ref} & & & \\ & w_{ref} & & \\ & & w_{vel} & \\ & & & 0 \end{bmatrix} \Delta x \tag{13}$$

3) constraint cost: All inequality constraints can be expressed in a negative null form shown in (14) in which x_{lim} is the maximum or minimum boundary value and $f(x)$ is a function on the decision variable. For instance, this function can be the distance between the center of the rear axle to the non-drivable lane boundary.

$$g(x) = x_{lim} - f(x) \leq 0 \tag{14}$$

For linear constraints like acceleration and steering limits, we can write them as for instance:

$$g(u) = u - u_{max} \leq 0$$

Then, a barrier function can be used as stated in (5).

The ego vehicle is represented by two circles centered at the middle of the rear and front axle. Obstacles are formulated as ellipses with major and minor axes adjusted for ego vehicle geometry. The over-fitting is the minimum-area ellipse and the inequality constraint is shown in (16). θ_t^{obs} is the heading angle of the obstacle at time t, and a and b are the semi major and minor axes' lengths. Δx_t is the position difference between the ego vehicle and the obstacle.

$$R_t = \begin{bmatrix} \cos(\theta_t^{obs}) & -\sin(\theta_t^{obs}) \\ \sin(\theta_t^{obs}) & \cos(\theta_t^{obs}) \end{bmatrix}$$

$$A_t = R_t \begin{bmatrix} \frac{1}{a^2} & \\ & \frac{1}{b^2} \end{bmatrix} R_t^T \tag{15}$$

$$g(x_t) = 1 - \Delta x_t^T A_t \Delta x_t \leq 0 \tag{16}$$

For short-term prediction, since the output of the reachability analysis is a sequence of rectangular bounding boxes, each box essentially represents the upper bound and lower bound of the mass point's future (x, y) position. We further inflate the boxes into ellipses considering the vehicle's shape. These ellipses will be addressed as the obstacles over the short-term prediction horizon.

The same inequality constraint needs to be repeated for the front center of the ego vehicle as well. Reformulated inequality constraints can then be used after linearization and barrier function transformation.

V. EXPERIMENTAL RESULTS

In this section, the environment is set up to have two lanes. The top lane ranges from $y = 0\,m$ to $y = 6\,m$ with the center of the lane located at $y = 3\,m$. The bottom lane ranges from $y = 0\,m$ to $y = -6\,m$ with the center of the lane located at $y = -3\,m$. Solid black lines and red dashed lines represent lane separators and lane centers, respectively. The reference velocity for all test cases is set to 15 m/s (roughly 35 mph), which is a common speed limit. The acceleration is limited to $[-4, 6]m/s^2$ and the steering angle is limited to $\pm 30°$.

The ego vehicle is pictured in blue with its transparency decreased over time (i.e., white is the frame at t_0, and the less transparent the color, the later a frame is in time). Target vehicles are depicted in red and the transparency works the same way.

A. General Performance Evaluation

In this scenario, a slow vehicle is initially 20 meters ahead of the ego vehicle in the adjacent lane and it is travelling at 8 m/s. There is an incoming vehicle travelling in the opposing lane and the task is to safely overtake the front vehicle. As seen in the velocity profile of Figure 2, the ego vehicle first slows down behind the target vehicle to ensure safety and then pulls over for the overtake while avoiding the incoming vehicle. During the overtaking maneuver, the ego vehicle smoothly accelerates to the reference velocity of 15 m/s. This test case is used to show the basic ability of the planning framework we propose.

For comparisons with the original CILQR method, we use the transient behavior percentage as a performance
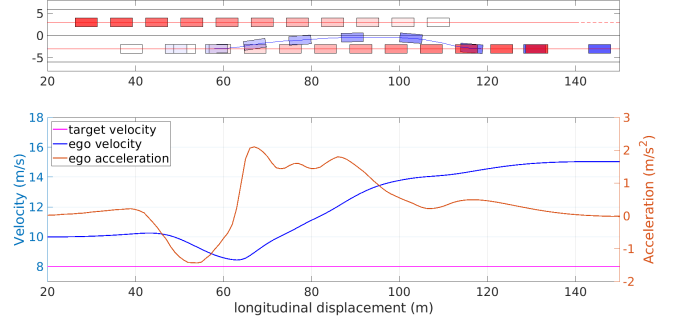

Fig. 2: Critical Section of Overtaking a Slow Front Vehicle Travelling 8m/s

benchmark. We define the transient behavior as behaviors in which the ego vehicle performs a hesitating maneuver with a sinusoidal like trajectory. The transient behavior is potentially dangerous because these sinusoidal trajectories have significant deviations from the lane center and could pose collision hazards to surrounding vehicles especially when the surrounding vehicles need to perform a lane change into the ego vehicle's lane. A set of weights is tuned for overtaking in the CILQR method and a set of adaptive weighting function parameters is tuned to encourage over-taking against target vehicles with velocity less than 13 m/s. The results are listed in Table I. For CILQR, when the target vehicle velocity exceeds 12 m/s, all planned trajectories are transient ones. For the method we propose, there are no transient behaviors observed and planned trajectories are either car-following ones or overtaking ones. To achieve the two different behaviors with CILQR, two sets of weights need to be tuned manually, which is avoided in the proposed method. This indicates that the framework we propose is a planner with increased generality.

TABLE I: CILQR vs. Adaptive Weight Tuning Performance Comparison for Target Vehicle Speed Range from 8 m/s to 14.9 m/s

Method	CILQR			Adaptive Weight Tuning		
Behaviors	Over-take	Tran-sient	Lane-keep	Over-take	Tran-sient	Lane-keep
occurrences	48	22	0	56	0	14
percentage (%)	68.57	31.43	0	80.00	0	20.00
speed range (m/s)	8-11.9	12-14.9	0	8-13.5	0	13.6-14.9

B. Adjacent Vehicle Violently Cuts in

Due to blind spot awareness, drivers from adjacent lanes might perform a lane change maneuver without environmental awareness. This is a dangerous situation, as the target vehicle is unlikely to yield and distance between two vehicles is very close, which limits the reaction time.

In this scenario, a target vehicle that is initially 5 meters ahead of the ego vehicle in the adjacent lane violently cuts into the ego vehicle's lane, as shown in Figure 3. Trajectories of the ego vehicle and the target vehicle are depicted in

solid blue and red lines in the top figure correspondingly. The bottom plot shows the velocity and the acceleration profile. The blue and the purple lines are velocities of the ego vehicle and the target vehicle, respectively. The red line is the acceleration of the ego vehicle. The target trajectory is generated by a fifth-order polynomial fitting. It completes the lane changing maneuver in 3 seconds.

Around $x = 30m$, the planner reacts to the cut-in vehicle. It slows down and performs a yielding turn to avoid the incoming target vehicle. As the target vehicle passes by, it then follows the target vehicle using a car-following behavior. We further test for the safety margin of the planner with decreased cut-in vehicle velocity and shorter reaction time.

We then compare the planned trajectory against a planner using deterministic knowledge of the obstacle. In this planner, we used the obstacle future trajectory directly as the prediction information to generate an optimal solution as the baseline. The baseline optimal solution, shown in Figure 4, is to speed up and go around the incoming target vehicle. Compared to the optimal solution, we can see that the planner we proposed is more conservative in terms of whether to overtake the target vehicle when it cuts in.
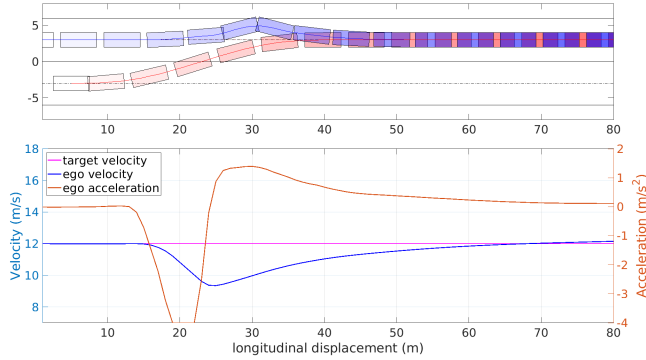

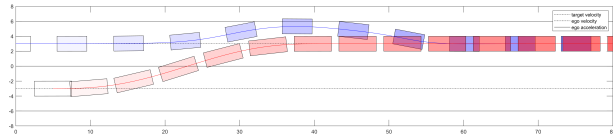Fig. 3: Target Vehicle Travelling 12 m/s Violently Cuts into the Ego Lane

Fig. 4: Cut-in Scenario Time Optimal Solution

C. Target Vehicle Suddenly Accelerates during Overtaking Maneuver

In this scenario, the target vehicle that is being overtaken suddenly starts to accelerate to avoid being overtaken. It imitates an aggressive style of driving. Green boxes are reachability analysis at a certain time step. It predicts 0.5 seconds ahead of time with a step of 0.1 seconds each. Reachability boxes are also drawn in Figure 5 for visualiza-

tion purposes against the planned trajectory for 0.5 seconds (shown in blue asteroid lines).

In Figure 5, only the long-term prediction is used for the whole 4s planning. Note again that the short-term prediction is disabled in this case and the visible green reachable set is only used for investigating its capability of safety guarantee. The target vehicle pictured in red starts to accelerate at around the 105-meter mark in Figure 5a. At time t_0, the planned trajectory that is 0.5 seconds ahead intersects with the reachable set, which is labeled as p_0 in Figure 5b. It demonstrates that the planned trajectory is not guaranteed to be collision-free. Not surprisingly, at the succeeding time step t_1, the ego vehicle collides with the target vehicle because the long-term planner does not react timely to the acceleration.

(a) Unsafe Trajectory and Velocity Profile
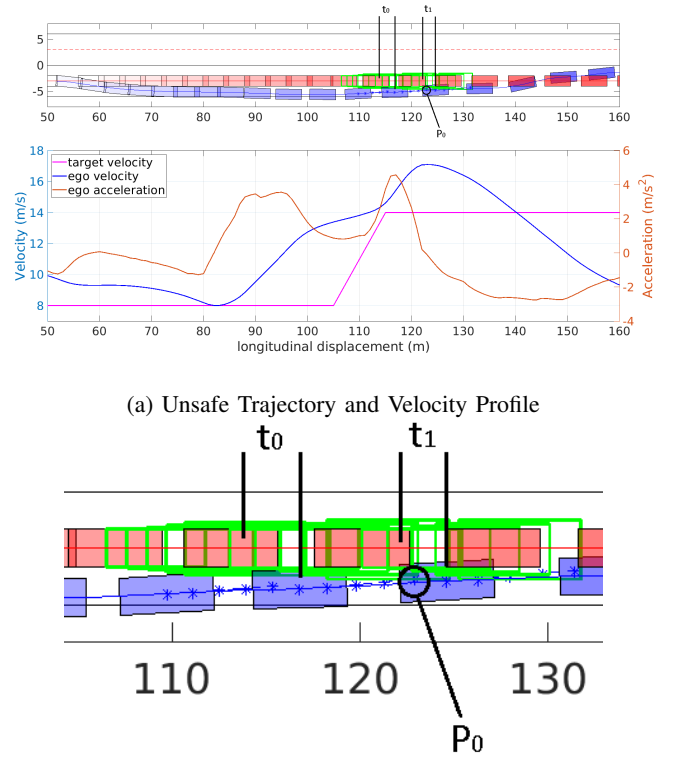
(b) Enlarged Critical Section. Same time frames of two cars are aligned with pairwise vertical black lines.

Fig. 5: Unsafe Corner Case Using Only Long-Term Prediction

In Figure 6, all planned trajectories at each time step take reachability analysis into account and the planner avoids the collision case in Figure 5a. As a result, for extreme cases, the short-term prediction guarantees the planned trajectory robust and safe enough against sudden target vehicle movement changes. The reason is that planning to avoid the reachable region allows the target vehicle to perform any non-deterministic but kinematically feasible behaviors over the short-term horizon.

The average loop time for these experiments is 60 ms with a standard deviation of 25 ms. The simulation is written in
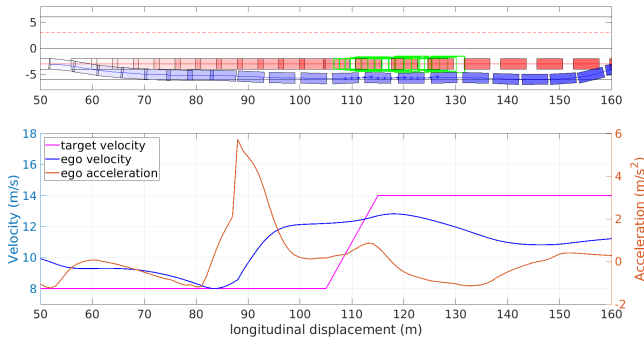
Fig. 6: Safe Planning Using Reachability for the Short-Term Prediction

Matlab and runs on a laptop with a 2.80GHz Intel Core i7-7700HQ CPU.

VI. CONCLUSION

In this paper, the existing CILQR framework is improved in two respects. We first added uncertainty awareness to the framework and made it more robust to sudden target vehicle maneuvers. The safety can be guaranteed because reachability is essentially an enclosure of all possible trajectories of the target vehicle over the short-term horizon. The long-term prediction is used to avoid over-conservativeness. Secondly, an adaptive weight calculation function is added to the algorithm. It allows the planner to perform multiple maneuvers without tuning for different cost function weights.

There still exist several interesting problems. The first is the tradeoff between the hard obstacle avoidance constraint and the numerical stability when incorporating the barrier function. The second problem is the inclusion of jerk within the cost function for the purpose of comfort.

VII. ACKNOWLEDGMENT

REFERENCES

[1] J. Chen, W. Zhan, and M. Tomizuka, "Constrained iterative lqr for on-road autonomous driving motion planning," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017, pp. 1–7.

[2] ——, "Autonomous driving motion planning with constrained iterative lqr," in *IEEE Transactions on Intelligent Vehicles (IV)*. IEEE, 2019, pp. 244–254.

[3] J. K. Subosits and J. C. Gerdes, "From the racetrack to the road: Real-time trajectory replanning for autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 2, pp. 309–320, 2019.

[4] J. Ziegler, P. Bender, T. Dang, and C. Stiller, "Trajectory planning for berthaa local, continuous method," in *2014 IEEE intelligent vehicles symposium proceedings*. IEEE, 2014, pp. 450–457.

[5] W. Xu, J. Wei, J. M. Dolan, H. Zhao, and H. Zha, "A real-time motion planner with trajectory optimization for autonomous vehicles," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 2061–2067.

[6] Y. Gao, T. Lin, F. Borrelli, E. Tseng, and D. Hrovat, "Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads," in *ASME 2010 dynamic systems and control conference*. American Society of Mechanical Engineers, 2010, pp. 265–272.

[7] A. Arab, K. Yu, J. Yi, and D. Song, "Motion planning for aggressive autonomous vehicle maneuvers," in *2016 IEEE International Conference on Automation Science and Engineering (CASE)*. IEEE, 2016, pp. 221–226.

[8] J. Ji, A. Khajepour, W. W. Melek, and Y. Huang, "Path planning and tracking for vehicle collision avoidance based on model predictive control with multiconstraints," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 2, pp. 952–964, 2016.

[9] Y. Tassa, N. Mansard, and E. Todorov, "Control-limited differential dynamic programming," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 1168–1175.

[10] J. van den Berg, "Extended lqr: Locally-optimal feedback control for systems with non-linear dynamics and non-quadratic cost," in *Robotics Research*. Springer, 2016, pp. 39–56.

[11] ——, "Iterated lqr smoothing for locally-optimal feedback control of systems with non-linear dynamics and non-quadratic cost," in *2014 American Control Conference*, 2014, pp. 1912–1918.

[12] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH journal*, vol. 1, no. 1, p. 1, 2014.

[13] Q. Lin, X. Chen, A. Khurana, and J. M. Dolan, "Reachflow: An online safety assurance framework for waypoint-following of self-driving cars," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. –.

[14] Q. Lin, S. Verwer, and J. M. Dolan, "Safety verification of a data-driven adaptive cruise controller," in *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2020, pp. –.

[15] M. Althoff and J. M. Dolan, "Online verification of automated road vehicles using reachability analysis," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 903–918, 2014.

[16] C. Liu and M. Tomizuka, "Robot safe interaction system for intelligent industrial co-robots," *arXiv preprint arXiv:1808.03983*, 2018.

[17] X. Chen, E. Ábrahám, and S. Sankaranarayanan, "Flow*: An analyzer for non-linear hybrid systems," in *International Conference on Computer Aided Verification*. Springer, 2013, pp. 258–263.

[18] Y. Zhang, Q. Lin, J. Wang, S. Verwer, and J. M. Dolan, "Lane-change intention estimation for car-following control in autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 3, pp. 276–286, 2018.

[19] G. C. Goodwin and K. S. Sin, *Adaptive filtering prediction and control*. Courier Corporation, 2014.

[20] C. Liu and M. Tomizuka, "Safe exploration: Addressing various uncertainty levels in human robot interactions," in *2015 American Control Conference (ACC)*. IEEE, 2015, pp. 465–470.