2017 IEEE Intelligent Vehicles Symposium (IV)
June 11-14, 2017, Redondo Beach, CA, USA

Path Planning for Autonomous Vehicles using Model Predictive Control

Chang Liu[1], Seungho Lee[2], Scott Varnhagen[2], H. Eric Tseng[2]

***Abstract*— Path planning for autonomous vehicles in dynamic environments is an important but challenging problem, due to the constraints of vehicle dynamics and existence of surrounding vehicles. Typical trajectories of vehicles involve different modes of maneuvers, including lane keeping, lane change, ramp merging, and intersection crossing. There exist prior arts using the rule-based high-level decision making approaches to decide the mode switching. Instead of using explicit rules, we propose a unified path planning approach using Model Predictive Control (MPC), which automatically decides the mode of maneuvers. To ensure safety, we model surrounding vehicles as polygons and develop a type of constraints in MPC to enforce the collision avoidance between the ego vehicle and surrounding vehicles. To achieve comfortable and natural maneuvers, we include a lane-associated potential field in the objective function of the MPC. We have simulated the proposed method in different test scenarios and the results demonstrate the effectiveness of the proposed approach in automatically generating reasonable maneuvers while guaranteeing the safety of the autonomous vehicle.**

I. Introduction

The last decade has seen significant progress of autonomous vehicles, mainly fueled by advances in software and hardware on sensing, computing and control. The main benefit of autonomous driving is to improve the traffic safety, especially by reducing the drivers error, which accounts for 94% of the traffic related fatalities on U.S. roadways in 2014 [1]. Autonomous driving can also facilitate the productive use of transit time for daily commuters [2]. Among the various components that empower Autonomous Vehicles (AVs), the path planning capability composes one of the core techniques. Path planning for AVs is a difficult decision-making and control problem in that a vehicle is a nonlinear, non-holonomic system and such system need to be controlled to maintain the desired performance (e.g., running at a desired speed, keeping passengers comfortable) while avoiding collision with surrounding vehicles and infrastructure.

Research efforts have been continuously devoted to developing efficient path planning approaches for AVs. The early works mainly use graph search based planners to efficiently generate shortest paths on the graph constructed by the discretization of the environment. Well-known methods such as the Dijkstras algorithm, A* algorithm (and its variants) and the state lattices have all been utilized for real-time planning [3]–[5]. In spite of the efficiency and optimality guarantee of these methods, the quality of the planned trajectories highly depends on the resolution of the graph. Besides, the constraints of the vehicle dynamics cannot be easily taken into account during the planning process. Curve interpolation planners, including the Clothoid curves [6], polynomial curves [7], spline curves [8] and Bezier curves [9], have also been widely used for online path planning. Similar to the graph search based approaches, the advantage of curve interpolation planners is their low computational cost, since the curve behavior can be defined by only a few control points or parameters. However, the optimality of the generated trajectory cannot be guaranteed. In addition, since the vehicle dynamics is not considered in the planning process, a smoothing process is usually needed to refine the generated paths so that they are dynamically feasible for vehicles.

Sampling based methods, such as the Probabilistic Road Maps (PRM) and the Rapidly-exploring Random Trees (RRT) [10], have gained much interest in recent years. They offer the benefits of generating feasible trajectories for both holonomic and non-holonomic systems. The RRT* [11] can even guarantee the asymptotic optimality of the generated path, which amounts to almost-sure convergence to global optimal solutions with increasing number of samples. These sampling based methods and their variants have been widely applied to AVs in research communities [12]–[14]. However, the practical use of these approaches is hindered by the high computational complexity of the sampling procedure and thus is of limited practical use.

Trajectory optimization based approaches have become increasingly popular for AV path planning recently [15], [16]. The core idea is to formulate the path planning as an optimization problem, which takes into account the desired vehicle performance and relevant constraints. The main advantages of these approaches is the flexibility and easiness of encoding all kinds of requirements on the desired paths of AVs. In addition, recent progresses in online optimization solvers, such as the CVX [17] and Ipopt [18] have also enabled fast and reliable path generation in real time. Traditionally, trajectory optimization based approaches solve a complete trajectory that starts from the starting configuration to the goal configuration. However, since the driving environment is usually dynamic and stochastic, and cannot be fully predicted a-priori, the Model Predictive Control (MPC) approach is commonly used for online path planning of AVs in recent years [19]–[21]. MPC solves a sequence of finite-time trajectory optimization problem in a recursive manner and can take into account the updating of the environment states during its planning process. Therefore, we use the

*The first two authors equally contribute to this work.

[1]Chang Liu is with Department of Mechanical Engineering, University of California, Berkeley, CA 94720, USA. He worked in Ford Motor Company as an intern. changliu@berkeley.edu

[2]Seungho Lee, Scott Varnhagen, and H. Eric Tseng are with Ford Research Laboratories, Dearborn, MI 48124, USA. {slee198,svarnha3,htseng}@ford.com

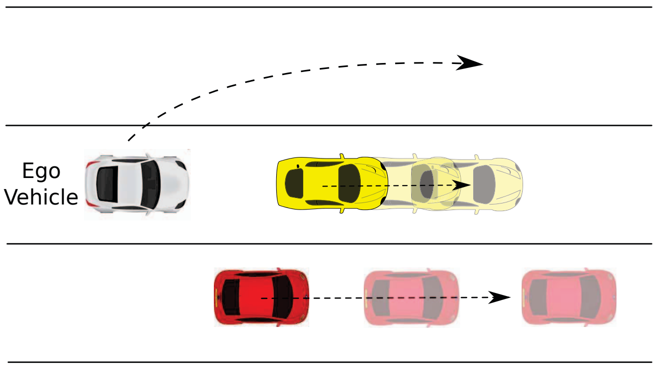

Fig. 1: A structured driving environment. The states of surrounding vehicles (yellow and red cars) are predicted in the time domain.

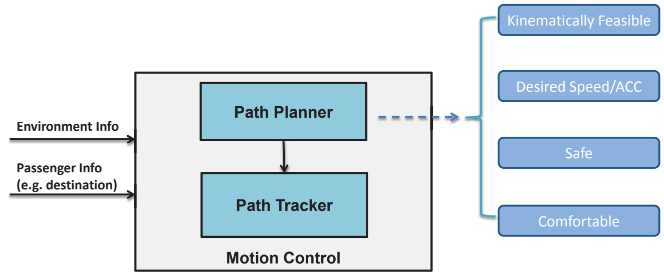

Fig. 2: A two-layer motion control strategy for the autonomous vehicle. The path generated from the Path Planner needs to be kinematically feasible, safe, comfortable, and maintain a desired speed or spacing from the surrounding vehicles.

MPC approach for online path planning of AVs.

A typical vehicle trajectory involves different types of maneuvers on the road. Previous works treat different maneuvers as separate driving modes, and only consider the control of AVs in one of these modes [16], [22]. Methods for the high-level decision making that decides the mode switching of AVs have also been proposed [23], [24]. However, such methods usually rely on predefined policies, which is laborious to design and prone to be unreliable under unexpected situations.

In this study, we propose a MPC based path planning approach, which deals with different modes under a unified framework. Instead of using predefined rules to decide certain maneuvers, we solve nonlinear MPC problems online. The maneuvers, including the lane change, lane keeping behavior, ramp merging, and intersection crossing, are all automatically decided by the generated path from the MPC. We utilize a relaxed convex combination method in the objective function of MPC to determine the lane change and lane keeping maneuvers. To ensure driving safety, we model vehicles as polygons and develop a type of constraints that enforces collision avoidance between the ego vehicle and surrounding vehicles. We also develop a lane-associated potential field that is applied to the surrounding vehicles in the same lane to achieve comfortable and natural maneuvers. The proposed path planner is evaluated in simulations and the results have demonstrated its effectiveness in automatically generating reasonable vehicle paths while guaranteeing the safety of AVs.

II. Problem Formulation

Fig. 1 illustrates the scenario of autonomous driving considered in this work. In a structured environment, including both the highway and the urban driving situations, the autonomous vehicle (ego vehicle) drives to a given destination at the speeds regulated by the roadway. The vehicle needs to avoid colliding with surrounding vehicles and maintain a comfortable driving experience.

We consider a dual-layer structure of motion control, as illustrated in Fig. 2. The upper-level controller, the *path planner*, takes in the environment and passenger information, and generates a reference trajectory for the autonomous vehicle. The lower-level controller, the *path tracker*, directly applies the control inputs to the vehicle to accurately track the reference trajectory. For the purpose of efficiency, the path planner uses a simplified kinematic model of the vehicle while the path tracker utilizes the high-fidelity dynamic model. This paper focuses on the design of the path planner.

At each time step, the ego vehicle plans its path for a finite horizon using MPC: the vehicle computes an optimal control sequence and then implements the control input corresponding to the current time step; it then computes an optimal control sequence starting from the updated vehicle state, and implements the computed optimal control input. This procedure is implemented in the receding horizon way until the vehicle arrives at its destination. In the planning horizon, the road geometry is approximated online by the 4^{th} order polynomial based on offline mapped waypoints and lane markings.

III. Path Planning Algorithm

A. Vehicle and Environment Modeling

In this work, we use the road coordinate system (Fig. 3), where x and y denote the longitudinal and lateral positions of the vehicle. θ is the heading angle and v is the speed of the vehicle. A unicycle kinematic model is used for modeling the ego vehicle and also surrounding vehicles:

$$z_{k+1} = z_k + f(z_k, u_k)T,$$

where $z_k = [x_k, y_k, \theta_k, v_k]^T$ is the state of the vehicle at time k and $f(z_k, u_k) = [v_k \cos\theta_k, v_k \sin\theta_k, \omega_k, \alpha_k]^T$. The control input $u_k = [\alpha_k, \omega_k]$ consists of the linear acceleration α_k and angular velocity ω_k. We assume that the environment information, including the road waypoint, lane markings, and surrounding vehicles' current states, is fully known. The future states of surrounding vehicles are also considered as accurately predicted.

B. MPC Formulation

The model predictive control approach solves a finite-time constrained optimal control problem in a receding horizon

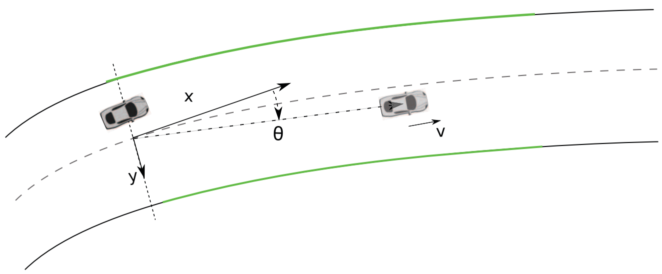

Fig. 3: The origin of the road coordinate lies on the center line of the road. The road geometry is approximated locally, as shown by the green curves.

manner. Let N be the prediction horizon. The path planning can be formulated as a nonlinear optimization problem:

$$\begin{aligned}\min_{\substack{u_{1:N}\\ \lambda_{M\times N}}} \quad & \sum_{k=1}^{N}\{w_{gx}D_k^2(x_k) + w_{gy}D_k^2(y_k) + w_v\|v_d - v_k\|^2 \\ & + w_a\|\alpha_k\|^2 + w_y\omega_k^2 + w_j\|\alpha_k - \alpha_{k-1}\|^2 \\ & + w_h(\theta_N - \theta_d)^2 + w_c\sum_{j=1}^{M}\lambda_{j,k}h(x_k, y_k, p_c^j)^2 \\ & + w_p\sum_{j\in V}P(d_{j,k}^o, v_{j,k}^o)\} \qquad (1a) \\ s.t. \quad & z_{k+1} = z_k + f(z_k, u_k)T, \qquad (1b) \\ & z_{k+1}\in\mathcal{Z},\, u_k\in\mathcal{U},\, \lambda\in\Lambda_{M\times N}, \qquad (1c) \\ & g(z_{k+1})\le 0, \qquad (1d) \\ & k = 0,\dots,N-1,\end{aligned}$$

where M is the number of lanes on the roadway; Eqn. (1b) is the constraint imposed by the vehicle kinematics; Eqn. (1c) constrains the feasible set of the state, which considers the actuator (linear acceleration and turning rate) limits of the vehicle; Eqn. (1d) enforces the collision avoidance between the ego vehicle and surrounding vehicles, the formulation of which will be described later.

The objective function (Eqn. (1a)) consists of several different terms to regulate the behavior of the ego vehicle. $D_k^2(x_k)$ and $D_k^2(y_k)$ are the longitudinal and lateral distance to the destination of the vehicle. Minimizing their sum drives the vehicle to its goal position. $\|v_d - v_k\|^2$ regulates the difference between the vehicle's actual speed and the desired speed v_d. $\|\alpha_k\|^2$ and ω_k^2 penalizes the large control input, and minimizing the jerk term $\|\alpha_k - \alpha_{k-1}\|^2$ improves the comfort of passengers in the vehicle. $(\theta_N - \theta_d)^2$ drives the vehicle to align its terminal heading with the curvature of the road. The coefficients w's are the weighting factors of these terms.

1) Lane Selection: The term $h(x_k, y_k, p_c^j)^2$ represents the lateral distance between the ego vehicle and the center line p_c^j of the j^{th} lane. The analytical expression of p_c^j is defined by a 4^{th}-order polynomial. To determine the lane that the vehicle needs to stay in using MPC, a mixed integer problem can be defined, with the integer variable reflecting the vehicle's choice of the lane. However, solving a mixed integer non-convex programming (the non-convexity comes from the constraints Eqn. (1b) and (1d)) is computationally heavy and thus not suitable for real-time implementation. Therefore we turn to a convex relaxation of the mixed integer problem and transform it into a nonlinear programming problem. In fact, we define an $M \times N$ matrix λ, with $\lambda_{j,k}$ being the entry on the j^{th} row and k^{th} column, as a design variable. The purpose of λ is to determine which lane the vehicle should be at in the prediction horizon. Each column of the matrix λ acts as multipliers for the convex combination of all M lanes; each row corresponds to the multipliers at different time steps. The λ belongs to $\Lambda_{M\times N}$, which is the set of unit simplex defined over $\mathbb{R}^{M\times N}$:

$$\begin{aligned}\Lambda_{M\times N} :=\{\lambda\in\mathbb{R}^{M\times N} \mid \sum_{j=1}^{M}\lambda_{j,k} = 1, \\ \lambda_{j,k}\ge 0, \forall j\in\mathbf{M}, \forall k\in\mathbf{N}\},\end{aligned}$$

where $\mathbf{M} = \{1, 2, \cdots, M\}$, $\mathbf{N} = \{1, 2, \cdots N\}$.

If we consider $\lambda_{j,k}, \forall j \in \mathbf{M}$ for some $k \in \mathbf{N}$, it is easy to notice that the optimal choice of $\lambda_{j,k}$ to minimize $\sum_{j=1}^{M}\lambda_{j,k}h(x_k, y_k, p_c^j)^2$ is by setting $\lambda_{\tilde{j},k} = 1$ and $\lambda_{j,k} = 0$, $\forall j\in\mathbf{M}\setminus\{\tilde{j}\}$, where $\tilde{j}$ is the lane that the vehicle is in. Therefore, by using the convex relaxation, we turn a mixed integer problem into a nonlinear programming formulation and the MPC can be solved online using nonlinear optimization solvers.

2) Lane-Associated Potential Field: In order to achieve comfortable and natural maneuvers, we introduce the lane-associated potential fields, $P(d_{j,k}^o, v_{j,k}^o)(\forall j \in V)$, to the surrounding vehicles in the longitudinal direction of the vehicle's current lane (the index set of these surrounding vehicles is represented by V). A potential field is defined as

$$P(d_{j,k}^o, v_{j,k}^o) = s \times e^{-\gamma(v_{j,k}^o){d_{j,k}^o}^2}, \qquad (2)$$

where $d_{j,k}^o$ is the longitudinal distance between the ego vehicle and the j^{th} surrounding vehicle in the same lane at time k. The parameters s and $\gamma(v_{j,k}^o)$ determine the maximum magnitude and the slope of potential field, respectively. To apply wider range of potential fields to the high-speed vehicles than the ones with low speeds, the parameter γ is set to be proportional to the j^{th} vehicle's speed $v_{j,k}^o$. An illustration of the potential field is shown in Fig. 4.

3) Collision Avoidance: Though the potential field can help vehicles to keep certain distances from each other in general, the additional mechanism to guarantee the collision avoidance is needed. We enforce the collision avoidance as a type of constraints in MPC to guarantee the safety of the ego vehicle in both longitudinal and lateral directions.

We model both the ego and surrounding vehicles as rectangles instead of the commonly used circles or ellipses since a rectangle can more accurately represent the shape of a vehicle. Besides, obstacles in common driving environments, such as road blocks and construction sites, can also be represented as polyhedra in general. Therefore, it is desirable

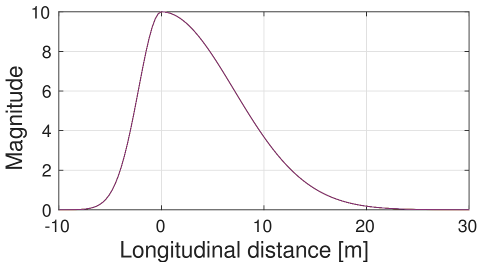

Fig. 4: The shape of the lane-associated potential field with $v^o = 10$. It is applied on the surrounding vehicles. The longitudinal distance is negative if the surrounding vehicle is in front of the go vehicle.

to develop a collision avoidance strategy for polyhedra that can be used in MPC.

A polyhedron can be represented by $Ax \leq b$, where A and b are of appropriate dimensions. The collision avoidance requires that two polyhedra not intersect, which is equivalent to the following condition:

$$\left\{z \in \mathbb{R}^n | \bar{A}z \leq \bar{b}\right\} = \emptyset, \tag{3}$$

where

$$\bar{A} = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}, \; \bar{b} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix},$$

and A_1, A_2, b_1, b_2 are the representations of two polyhedra. This condition is, however, not desirable for the MPC formulation, since it requires the nonexistence of variables. Instead, the MPC is well suited for finding variables that satisfies certain conditions. To deal with this problem, we follow the approach proposed in [25], which turns the collision avoidance condition into its dual form that is compatible for MPC formulation.

To be specific, according to Farka's lemma [26], an equivalent condition to Eqn. (3) is

$$\exists \beta \geq 0, \; s.t. \; \bar{A}^T\beta = 0 \text{ and } \bar{b}^T\beta < 0. \tag{4}$$

Therefore, instead of using Eqn. (3) as a constraint in MPC, we introduce a new variable β and include Eqn. (4) as the collision avoidance constraint (Eqn. (1d)).

We consider two rectangle-shaped vehicles that can be represented by their states as follows:

$$A_{i,k} = \begin{bmatrix} \sin(\theta_{i,k}) & -\cos(\theta_{i,k}) \\ -\sin(\theta_{i,k}) & \cos(\theta_{i,k}) \\ \cos(\theta_{i,k}) & \sin(\theta_{i,k}) \\ -\cos(\theta_{i,k}) & -\sin(\theta_{i,k}) \end{bmatrix},$$

$$b_{i,k} = \begin{bmatrix} W_i/2 \\ W_i/2 \\ H_i/2 \\ H_i/2 \end{bmatrix} + A_{i,k} \begin{bmatrix} x_{i,k} \\ y_{i,k} \end{bmatrix},$$

with $i = 1, 2$ representing the index associated with the corresponding vehicle. W_i, H_i represent the length and width of each vehicle. Therefore, $\bar{A} \in \mathbb{R}^{8\times 2}$, $\bar{b} \in \mathbb{R}^8$, $\beta \in \mathbb{R}^8$.

We assume q surrounding vehicles in the environment that the ego vehicle needs to avoid colliding with. The constraint Eqn. (1d) can be represented by

$$\bar{A}_{m,k}^T\beta_{m,k} = 0 \text{ and } \bar{b}_{m,k}^T\beta_{m,k} < 0, \; \beta_{m,k} \geq 0, \; m = 1 \ldots, q,$$

where $\bar{A}_{m,k}$, $\bar{b}_{m,k}$ are corresponding matrices between the ego and the m^{th} surrounding vehicle. We can actually reduce the dimension of β by utilizing the dependence of columns of $\bar{A}_{m,k}$. We refer interested readers to [25].

C. Fail-safe Strategy

Due to the vehicle kinematic model and the collision avoidance constraint, the MPC problem is a nonlinear, nonconvex problem. Therefore, it is possible that the solver fails to compute a feasible solution. A feasible initial solution will help eliminate this problem. We propose two relaxed MPC problems to compute a feasible initial solution, from which the solver can improve and calculate a local optimal solution. The first relaxed MPC problem is the same as the original MPC except that the objective function is set as a constant. If this MPC problem successfully computes a feasible solution, the solution is then used by the original MPC as the seed solution. However, if the first MPC problem fails, we turn to the second MPC problem, where we further remove the collision avoidance constraint. In our simulations, we find that these two relaxed MPC problems can always compute feasible solutions for the original MPC to improve upon and eliminate the infeasiblity issue.

IV. Simulation Results

The MPC path planner is evaluated in simulated scenarios that include normal highway driving, ramp merging, and intersection crossing. The MPC formulation (1a)-(1d) is solved with the Ipopt nonlinear programming solver via the YALMIP [27] toolbox in Matlab™. We present several simulations that we have conducted. The static figures below do not easily convey the evolution in time of the vehicle trajectories. We have therefore deposited animations in https://berkeley.box.com/v/IV17.

A. Normal Highway Driving

Fig. 5 shows the screen captures of a planned path in a normal highway driving situation. The lanes within the blue box are approximated as 4^{th}-order polynomials. Red rectangles are surrounding vehicles and the blue rectangle is the ego vehicle. The future trajectory of the ego vehicle is represented as a sequence of blue dots. Initially the ego vehicle is in the middle lane and there are five vehicles in front, and one on the left lane (Fig. 5(a)). It can be noticed that, though the desired speed of the ego vehicle is higher than obstacle vehicles in front, the ego vehicle maintains certain distances by decreasing the speed (Fig. 5(b)). As soon as the lane change becomes feasible, the ego vehicle increases the speed to make a smooth left lane change (Fig. 5(c)). and stays at the left lane (Fig. 5(d)). Note that the road has a certain curvature and the view point is rotated accordingly.

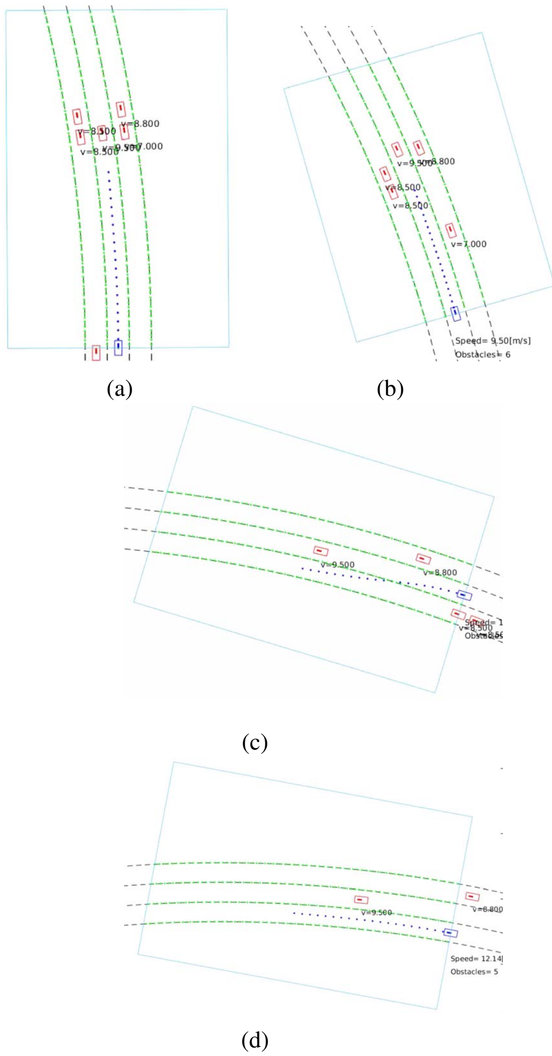

Fig. 5: Normal highway driving simulation. The ego vehicle (a) has five vehicles in the front; (b) maintains a certain distance from the preceding vehicles; (c) makes a lane change to the left lane; (d) stays in the current lane and maintains the desired speed. The texts in the subplots show the speed of each vehicle.

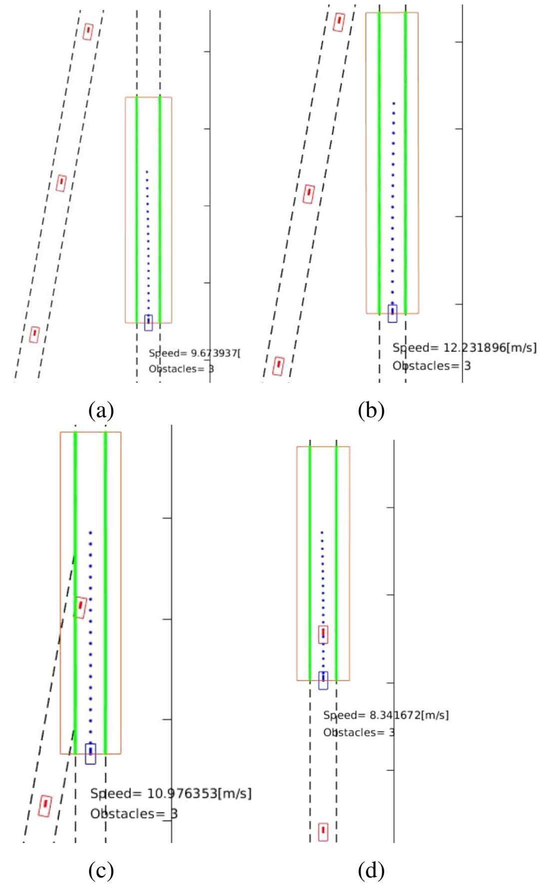

Fig. 6: Merging simulation. The ego vehicle (a) has surrounding vehicles merging into its lane; (b) increases the speed; (c) decreases the speed; (d) maintains a certain distance.

B. Merging in Highway Driving

Fig. 6 shows screen captures of the path planning in a ramp merging scenario. Three surrounding vehicles are in the merging lane (Fig. 6(a)). The ego vehicle generates a safe longitudinal path such that it can fit into the gap between two adjacent surrounding vehicles. The ego vehicle first accelerates to catch up with the gap (Fig. 6(b)) and then decelerates to keep a comfortable distance from the car in front (Fig. 6(c)). The ego vehicle then successfully merges into the lane (Fig. 6(d)).

C. Intersection Crossing

Fig. 7 shows screen captures of a four-way intersection crossing scenario. In this case, the target is chosen as the "stop" sign by setting the $D(y_k)$ term in Eqn. (1a). The ego vehicle first approaches the "stop" sign, stop there, and remain in the "stop" state until it becomes safe to cross the intersection (Fig. 7(b)-(d)).

V. Conclusion

In this work, we propose a Model Predictive Control based path planner for autonomous vehicles in structured driving environments. The proposed planner automatically decides the mode of maneuvers under a unified optimization framework. A convex relaxation approach is used to determine the lane change and lane keeping maneuvers. We have developed a collision avoidance condition for polygon-shaped vehicles to ensure the safety of the ego vehicle. We have also introduced the lane-associated potential field to achieve comfortable and natural maneuver. Simulation has been conducted for various different scenarios, including the lane change, lane keeping, ramp merging, and intersection crossing. Results have shown that the proposed path planner is effective in generating safe and comfortable path for autonomous vehicles. Future work will focus on the implementation of the path planner in real experiments. The prediction of surrounding vehicles' motion is also the focus of the next-step work.

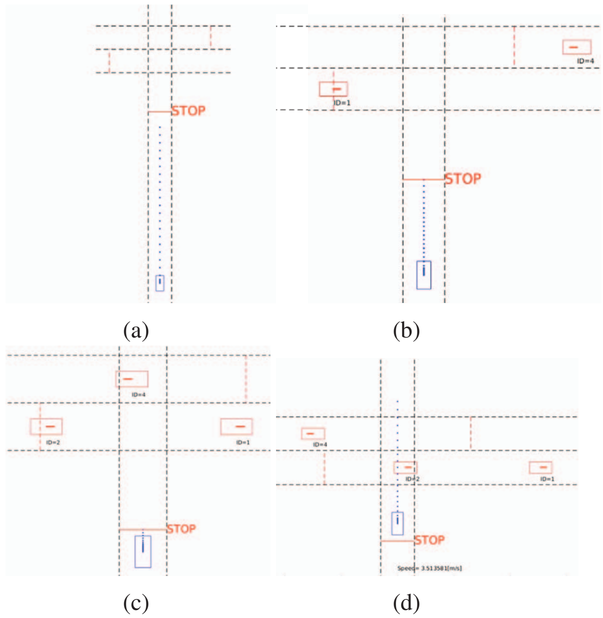

Fig. 7: Four-way intersection simulation. The ego vehicle (a) approaches the four-way intersection; (b) plans to stop at the "stop" sign; (c) stops and waits until it can cross the intersection; (d) crosses the intersection.

REFERENCES

[1] S. Singh, "Critical reasons for crashes investigated in the national motor vehicle crash causation survey," Tech. Rep., 2015.

[2] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[3] R. Kala and K. Warwick, "Multi-level planning for semi-autonomous vehicles in traffic scenarios based on separation maximization," *Journal of Intelligent & Robotic Systems*, vol. 72, no. 3-4, pp. 559–590, 2013.

[4] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.

[5] D. Ferguson, T. M. Howard, and M. Likhachev, "Motion planning in urban environments," *Journal of Field Robotics*, vol. 25, no. 11-12, pp. 939–960, 2008.

[6] M. Brezak and I. Petrović, "Real-time approximation of clothoids with bounded error for path planning applications," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 507–515, 2014.

[7] P. Petrov and F. Nashashibi, "Modeling and nonlinear adaptive control for autonomous vehicle overtaking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 4, pp. 1643–1656, 2014.

[8] A. Piazzi, C. L. Bianco, M. Bertozzi, A. Fascioli, and A. Broggi, "Quintic g 2-splines for the iterative steering of vision-based autonomous vehicles," *IEEE transactions on Intelligent Transportation Systems*, vol. 3, no. 1, pp. 27–36, 2002.

[9] J. P. Rastelli, R. Lattarulo, and F. Nashashibi, "Dynamic trajectory generation using continuous-curvature algorithms for door to door assistance vehicles," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*. IEEE, 2014, pp. 510–515.

[10] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[11] S. Karaman, "Incremental sampling-based algorithms for optimal motion planning," *Robotics Science and Systems VI*, vol. 104.

[12] A. Pongpunwattana and R. Rysdyk, "Real-time planning for multiple autonomous vehicles in dynamic uncertain environments," *Journal of Aerospace Computing, Information, and Communication*, vol. 1, no. 12, pp. 580–604, 2004.

[13] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J. P. How, "Real-time motion planning with applications to autonomous urban driving," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 5, pp. 1105–1118, 2009.

[14] D. Braid, A. Broggi, and G. Schmiedel, "The terramax autonomous vehicle," *Journal of Field Robotics*, vol. 23, no. 9, pp. 693–708, 2006.

[15] 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.

[16] P. Liu and Ü. Özgüner, "Predictive control of a vehicle convoy considering lane change behavior of the preceding vehicle," in *2015 American Control Conference (ACC)*. IEEE, 2015, pp. 4374–4379.

[17] M. Grant and S. Boyd, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control*, ser. Lecture Notes in Control and Information Sciences, V. Blondel, S. Boyd, and H. Kimura, Eds. Springer-Verlag Limited, 2008, pp. 95–110.

[18] 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, 2006.

[19] Y. Gao, A. Gray, J. V. Frasch, T. Lin, E. Tseng, J. K. Hedrick, and F. Borrelli, "Spatial predictive control for agile semi-autonomous ground vehicles," in *11th International Symposium on Advanced Vehicle Control*, 2012.

[20] C. E. Beal and J. C. Gerdes, "Model predictive control for vehicle stabilization at the limits of handling," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 4, pp. 1258–1269, 2013.

[21] A. Carvalho, Y. Gao, A. Gray, H. E. Tseng, and F. Borrelli, "Predictive control of an autonomous ground vehicle using an iterative linearization approach," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. IEEE, 2013, pp. 2335–2340.

[22] J. Nilsson, M. Brännström, E. Coelingh, and J. Fredriksson, "Longitudinal and lateral control for automated lane change maneuvers," in *2015 American Control Conference (ACC)*. IEEE, 2015, pp. 1399–1404.

[23] A. Furda and L. Vlacic, "Enabling safe autonomous driving in real-world city traffic using multiple criteria decision making," *IEEE Intelligent Transportation Systems Magazine*, vol. 3, no. 1, pp. 4–17, Spring 2011.

[24] M. Bahram, A. Wolf, M. Aeberhard, and D. Wollherr, "A prediction-based reactive driving strategy for highly automated driving function on freeways," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, June 2014, pp. 400–406.

[25] M. Gerdts, R. Henrion, D. Hömberg, and C. Landry, "Path planning and collision avoidance for robots," *Numerical Algebra, Control and Optimization*, vol. 2, no. 3, pp. 437–463, 2012.

[26] D. G. Luenberger and Y. Ye, *Linear and nonlinear programming*. Springer, 2015, vol. 228.

[27] J. Lofberg, "Yalmip : a toolbox for modeling and optimization in matlab," in *2004 IEEE International Conference on Robotics and Automation (IEEE Cat. No.04CH37508)*, Sept 2004, pp. 284–289.