

Autonomous Planning and Control for Intelligent Vehicles in Traffic

Changxi You Jianbo Lu Dimitar Filev Panagiotis Tsiotras

Abstract—This paper addresses the trajectory planning problem for autonomous vehicles in traffic. We build a stochastic Markov decision process (MDP) model to represent the behaviors of the vehicles. This MDP model takes into account the road geometry and is able to reproduce more diverse driving styles. We introduce a new concept, namely, the “dynamic cell,” to dynamically modify the state of the traffic according to different vehicle velocities, driver intents (signals) and the sizes of the surrounding vehicles (i.e., truck, sedan, etc.). We then use Bézier curves to plan smooth paths for lane-switching. The maximum curvature of the path is enforced via certain design parameters. By designing suitable reward functions, different desired driving styles of the intelligent vehicle can be achieved by solving a reinforcement learning problem. Desired driving behaviors (i.e., autonomous highway overtaking) are demonstrated with an in-house developed traffic simulator.

Keywords: Reinforcement learning, Bézier curve, curvature constraint, dynamic cell, path planning, autonomous vehicle.

I. INTRODUCTION

A self-driving vehicle is able to drive using different sensing modalities such as computer vision, localization, lidar, radar, ultrasound, etc to detect its environment. Autonomous vehicle technology is expected to significantly reduce collisions and resulting injuries, improve traffic congestion, enhance mobility for the disabled [1], and consequently, represents a major trend in the intelligent transportation systems in the future.

Several established automotive companies and startups are currently developing autonomous vehicles technology [2]–[6]. Technical issues related to the development of self-driving vehicles exist at three levels, including perception, planning and control. The perception level includes sensing and filtering of measured data. The filtering system removes the noise from the measurement data obtained using the sensing system and generates reasonable estimates of the states that cannot be directly measured in the localization process [7]–[9], and captures the necessary features from the environment (i.e., obstacles and road centerline) in the cognition process [10], [11]. In the planning level three tasks are completed, which include mission planning, where the vehicle solves a routing problem, behavioral planning, where a suitable action is selected from an available set, and path planning, where the vehicle's future trajectory is generated with respect to certain constraints and boundary conditions [12]–[15]. Finally, the control level stabilizes the vehicle and achieves the planned trajectory.

Numerous algorithms have been developed over the past decade for real-time path planning for autonomous vehicles, which can be categorized into three groups according to the methodology used to develop them, namely, sampling-based [14], [16], graph-search methods [17], [18], and geometry-based path planning [19]–[21]. In [14] the authors presented a path planning algorithm using Rapidly-exploring Random Trees (RRTs). They implemented the algorithm on a self-driving vehicle during the 2007 DARPA Urban Challenge. The path planning approaches using RRTs can efficiently explore the space and can handle obstacle avoidance problems. Nevertheless, since the search tree is built incrementally from the direction of the samples randomly chosen from the search space, an additional smoother may be required to smooth the path. Cimurs et al. [18] used Dijkstra's algorithm in order to find the shortest viable trajectory by connecting the Voronoi vertices, such that the trajectory keeps the largest clearance from all the obstacles in the environment. They then used Bézier curves as an additional smoother to locally smooth the path with respect to the maximum curvature constraint by selecting and aligning the control points. Some researchers plan a smooth trajectory without using any additional smoother. In [19], [20] Choi et al. proposed several geometry-based algorithms using Bézier curves for path planning. The curvature of such designed paths is continuous, and the paths are able to meet the requirement of the road boundary constraints. Shim et al. [21] represented a smooth path using a parameterized 6th-order polynomial. This path planning algorithm successfully avoided multiple static/moving obstacles in different tasks. Gómez et al. represented the state space of the vehicle using a number of grids and proposed to use the so-called control adjoining cell mapping and reinforcement learning (CACM-RL) algorithm to learn the vehicle dynamics and obtain the optimal motion planning that satisfies certain obstacle constraints [22]. A more extensive literature review on path planning for developing self-driving technologies is found in [23], [24].

In this paper we concentrate on the planning and control problems for self-driving vehicles in highway traffic. Based on the premise that true integration with existing traffic will not be possible until autonomous vehicles behave in a predictable manner, in this paper we wish to generate the behavioral planning of an expert driver such that the autonomous vehicle can mimic human-like driving patterns. That is, we wish to reproduce expert driving styles that involves typical driving actions such as lane-switching, lane-keeping, speed maintaining, braking and accelerating, by taking into account the stochastic driving actions of the traffic vehicles.

C. You is a Senior Researcher at Tencent Technology Company, Beijing 100084, China. Email: changxiyou@tencent.com

J. Lu is a Technical Expert, Research & Advanced Engineering, Ford Motor Company, Dearborn, MI 48121, USA. Email: jlu10@ford.com

D. Filev is a Technical Fellow, Research & Advanced Engineering, Ford Motor Company, Dearborn, MI 48121, USA. Email: dfilev@ford.com

P. Tsiotras is a Professor at the School of Aerospace Engineering and the Institute for Robotics & Intelligent Machines, Georgia Institute of Technology, Atlanta, GA 30332-0150, USA. Email: tsiotras@gatech.edu

Contributions

The main contributions of this work can be summarized as follows: 1) We propose to use an MDP to model the stochastic behaviors of the vehicles in highway traffic. Specially, this model takes into account the road geometry in order to show different driving policies during cornering. We solve the proposed MDP problem to obtain the optimal control strategy using reinforcement learning; 2) In order to deal with complicated traffic information such as different vehicle velocities, driver intent (signals) and the size of the other vehicles (i.e., truck, sedan, etc.), we introduce a new concept, namely, the “dynamic cell”, to dynamically modify the state of the model before making driving decisions. This approach is easy to implement and is easily scalable so as to incorporate other driving scenarios such as pedestrians, static/moving obstacles and road intersections; 3) We propose two path planning algorithms using both joint quadratic Bézier curves and fourth order Bézier curves. The optimal choices of the control points are provided in terms of the desired maximum curvature, hence a path satisfying the given curvature constraints can be generated in real-time. The fourth order Bézier curves are C^2 continuous and show better tracking performance when an output regulation tracking controller is implemented.

The rest of this paper is organized as follows: Section II builds a stochastic MDP traffic model and introduces the concept of the “dynamic cell”. Section III introduces the vehicle model used in this work. Section IV and V outline the algorithm to generate smooth paths using Bézier curves as well as the design of the low-level controllers. Section VI implements an RL algorithm to determine the optimal policy and implements the controllers on a driving simulator. Finally, Section VII summarizes the results of this work.

II. TRAFFIC MODELING

In this section we introduce the stochastic MDP traffic model used in this work. In the following sections we solve this MDP problem using RL techniques in order to learn the optimal policy and achieve the desired driving styles.

A. Markov Decision Process

An MDP can be represented with a 6-tuple $(S, A, R, \mathcal{T}, \gamma, D)$, where S denotes a set of states that characterizes a dynamically changing environment, A denotes a action set that provides all available actions the agent is able to select, R is the reward function that defines the reward the agent receives by taking certain action, $\mathcal{T}$ denotes a state transition matrix that specifies the transition probability between each pair of the states, $\gamma \in [0, 1)$ is the discount rate, and D represents the initial state distribution.

Given an MDP, we want to determine the optimal policy π^* for every state by maximizing the expected cumulative discounted reward as follows

$$\pi^* = \arg \max_{\pi} \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, \pi(s_t)) \right], \quad (1)$$

where $\pi : S \rightarrow A$ is a policy that determines an action the agent takes at the present state s_t , and $R(s_t, \pi(s_t))$ is the

corresponding reward to be obtained immediately by taking the action $a_t = \pi(s_t)$.

B. System Modeling

In order to characterize the behavior of the traffic, we build an MDP model as follows. Figure 1 shows a segment of multi-lane highway with a couple of vehicles. We assume that each vehicle wants to maintain a constant speed which represents the average traffic flow speed, and each vehicle wants to maximize the total rewards it receives from a set of possible actions. The idea is that if one is able to construct the reward function corresponding to an “experienced” driver, one will be possible to generate the driving style of such driver using certain RL techniques [25].

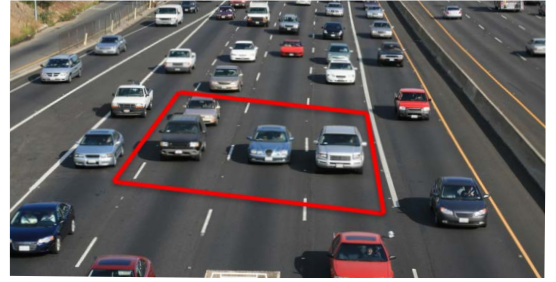


Fig. 1. A scene of the highway traffic.

Fig. 1 shows several vehicles inside a red rectangle. The driver of the blue vehicle in the middle of the rectangle has several available actions. She can maintain the current speed, accelerate, or brake shortly to take up the space in front of or behind the vehicle, or switch to the left/right lane if they are available. For simplicity, we designate the vehicle we control as the host vehicle (HV), and all surrounding vehicles as traffic vehicles (TVs). Subsequently, there is only one controlled agent in the MDP system. The action set for the HV/TVs is represented by $A \triangleq \{\text{“accelerate”, “brake”, “maintain”, “left-turn”, “right-turn”}\}$.

1) *State Definition*: We define the state of the MDP using the position of the HV and the positions of the surrounding TVs. In Fig. 2, we partition the highway segment into small grids using the white dashed lines and denote the HV using the green vehicle. The states of the system consider the three scenarios in Fig. 2: In ① the HV is in the internal lane and the state is represented using nine cells, and in ② and ③ the HV is in the lane next to the road sides and the state is represented using six cells. The number of the internal-lane states is $2^8 = 256$, and the number of the left(right)-boundary states is $2^5 = 32$. In order to analyze how the road geometry effects on the driver’s policy, we consider three different road curvatures to demonstrate left-turn, right-turn, and straight roads. The total number of the states is therefore $320 \times 3 = 960$. This is not restrictive however. It is worth mentioning that the model is easily scalable to account for driving scenarios having as many lanes and vehicles as desired.

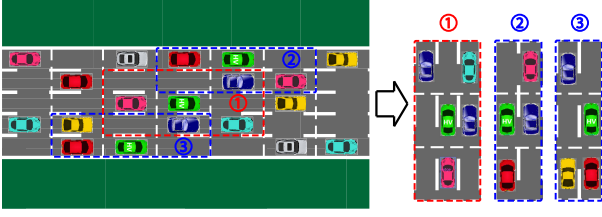


Fig. 2. State definition: ① 9-cell internal-lane state, ② 6-cell left-boundary state and ③ 6-cell right-boundary state

2) *State Transitions*: The state transition process is modeled to mimic real-world driving scenarios. Our assumptions are summarized as follows: 1) the number of lanes n is greater than equal to two ($n \geq 2$); 2) the number of TVs N is not larger than eight ($0 \leq N \leq 8$); 3) each TV has its own policy; 4) the TVs take a random action; 5) at each step each TV in the system takes one action; and 6) no accident can happen due to the actions of the TVs.

The state transition process can be divided into two steps: First, the HV takes an action according to the current state s_t and its policy $\pi(s_t)$. Second, the TVs take an action in a random sequence following their own policies. The next state s_{t+1} is formulated using the current positions of the vehicles (see [26]).

C. Reward Function

The driver's actual reward function is, of course, unknown and it is a difficult task to design the reward function for a specific driver. Moreover, the reward function may change with time and it may be different for each driver. A widely used approach to design the reward function is to express it as a function of certain features based on the state of the MDP and the action of the agent. In this paper we use a linear combination of features to represent the reward function [27]–[31], which is given as follows

$$R(s, a) = w^T \Phi(s, a), \quad (2)$$

where $\Phi(s, a)$ denotes the feature vector and w denotes the weight vector. In this paper the features in $\Phi(s, a)$ are defined using binary values that indicate whether a certain argument is true or not. The features are selected as follows:

- 1) Action feature. The driver may receive different rewards by taking different actions.
- 2) Lane of the HV. The driver may want to drive inside a certain lane.
- 3) Overtaking style. The driver may have different preferences for left/right lanes of the front car to complete overtaking in a corner.
- 4) Tailgating style. The feature value is “true” if the HV is behind a TV and “false” otherwise.
- 5) Accident incident. Accident happens if the HV enters a cell occupied by a TV.

In order to generate a certain driving behavior, we can change w and learn the optimal policy by maximizing the objective function in (1) using reinforcement learning [26], [32].

D. Dynamic Cell

The traffic model in Section II does not consider different vehicle sizes (i.e., truck, sedan, motorcycle, etc.), or vehicle

velocities. Instead of discretizing the speeds of the TVs [33], [34], which tend to increase the dimensionality of the MDP problem, here we introduce a new layer, namely, the “dynamic cell” layer, into the control architecture to account for these additional problem parameters.

We assume that each vehicle intends to keep a safe distance from the vehicle in front depending on its current longitudinal velocity. The length of the dynamic cell for the host vehicle is therefore defined by

$$L_{HV} = \Delta T \times V_{HV} + \ell_{HV} \quad (3)$$

where ΔT is the time constant that defines the minimum distance one wants to keep from the front vehicle, and ℓ_{HV} is the chassis length of the host vehicle. When the host vehicle is static, L_{HV} equals to the length of the vehicle itself. The 9-cell state of the traffic MDP model is therefore defined with a specified size as shown in the first picture in Fig. 3.

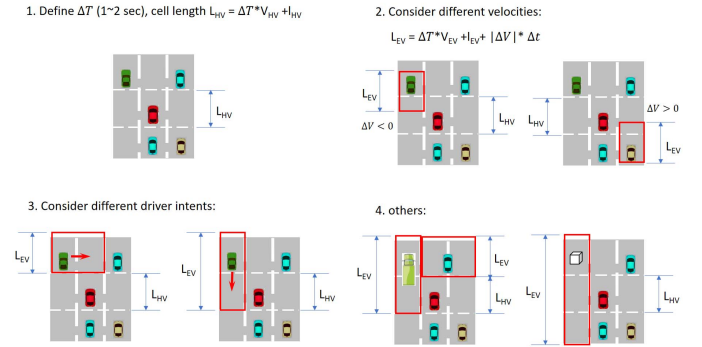


Fig. 3. Dynamic cells.

Similar to (3), the definition of the cell length for the TVs requires a modification term depending on the relative speed of the TV with respect to the HV

$$L_{TV} = \Delta T \times V_{TV} + \ell_{TV} + |\Delta V| \Delta t, \quad (4)$$

where V_{TV} and ℓ_{TV} are the longitudinal velocity and the chassis length of the TV, respectively and Δt is the time constant that determines how much the TV approaches the HV in the next step. This is shown at the second picture in Fig. 3. If the green TV is slower than the HV, the cell this TV occupies will move backward with a distance $|\Delta V| \Delta t$ due to the relative velocity ΔV , and hence it overlaps with the cell on the left of the HV. As a consequence, the left cell of the HV is not available for the lane-switching action. Other special cases, such as when there is a truck or some static obstacle in traffic, can also be handled by dynamically changing the cells. One can see Fig. 3 for a graphical explanation.

The cell width for either the HV or the TVs is naturally defined using the lane width. The signal light indicates the driver's intent and the HV is able to predict the motion of a TV by its signal light (i.e., the left/right turn signals and the braking light) and avoid taking dangerous actions. One can change the cell width of a TV according to its signal lights to indicate the area this TV occupies.

III. VEHICLE MODELING

We design low-level controllers to implement the desired actions for each vehicle. To this end, we first present the vehicle model used in this work.

A. Single Track Vehicle Model

The single track vehicle model takes into consideration the longitudinal and lateral translation, as well as the the vehicle's yaw motion, as shown in Fig. 4.

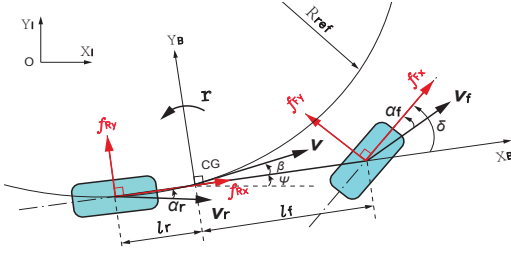


Fig. 4. Single-track vehicle model.

In this figure, $X_B - CG - Y_B$ and $X_I - O - Y_I$ denote the body frame fixed on the vehicle and the inertial frame fixed on the ground, respectively. V , V_f and V_r denote the velocities at the vehicle's center of mass (CG) and the front and rear wheels, respectively, and α_f , α_r and β denote the side slip angles of the front and the rear wheel and CG, respectively. The parameters f_{ij} ($i = F, R$ and $j = x, y$) denote the longitudinal tire forces at the front wheels and the lateral tire forces at the rear wheels, respectively, ℓ_f and ℓ_r denote the distances from CG to the front and rear axles, respectively, and ψ and r denote the yaw angle and the yaw rate, respectively. Finally, δ represents the front wheel steering angle. The equations of motion are given as follows [35]

$$\dot{V}_x = (f_{Fx} \cos \delta - f_{Fy} \sin \delta + f_{Rx}) / m + V_y \dot{\psi}, \quad (5a)$$

$$\dot{V}_y = (f_{Fx} \sin \delta + f_{Fy} \cos \delta + f_{Ry}) / m - V_x \dot{\psi}, \quad (5b)$$

$$\dot{r} = ((f_{Fy} \cos \delta + f_{Fx} \sin \delta) \ell_f - f_{Ry} \ell_r) / I_z, \quad (5c)$$

where m denotes the mass of the vehicle, I_z denotes the moment of inertia about the vertical axis, and V_x and V_y denote the two components of the velocity V along X_B and Y_B direction, respectively.

B. Tire Forces Model

The tire slip is defined by the relative velocity of each tire with respect to the road, which is given by

$$s_{ijx} = \frac{V_{ijx} - \omega_{ijx} R_j}{\omega_{ijx} R_j}, \quad s_{ijy} = \frac{V_{ijy}}{\omega_{ijx} R_j}, \quad (6)$$

where $i = L, R$ and $j = F, R$. V_{ijk} ($k = x, y$) is the tire frame component of the vehicle velocity of each tire. The comprehensive slip of each tire can be therefore computed using $s_{ij} = \sqrt{s_{ijx}^2 + s_{ijy}^2}$. The friction coefficient for each tire is computed using Pacejka's "magic formula" (MF), which is given as follows [35]–[37]

$$\mu_{ij} = D \sin \left(\text{Catan} \left(B S_E - E (B S_E - \text{atan} S_E) \right) \right) + S_v, \quad (7)$$

where B, C, D, E are the stiffness, shape, peak and curvature factors, respectively; S_v is the vertical shift. We use S_h to denote the horizontal shift, then the term $S_E = s_{ij} - S_h$.

We compute the tire friction forces using the following equation,

$$f_{ijk} = -\frac{s_{ijk}}{s_{ij}} \mu_{ij} f_{ijz}, \quad i = L, R; \quad j = F, R; \quad k = x, y. \quad (8)$$

where f_{ijz} denotes the vertical load on each tire.

C. Model Linearization

During lane-keeping or certain similar path-tracking tasks, the steering angle of the front wheel is typically small and the tires may only work within their linear zone of friction. Under these conditions, one is able to simplify the equations in (5a)–(5c) and (7). By further assuming that the longitudinal velocity V_x of the vehicle can be treated as constant during these tasks, we can represent the vehicle's equation of motion as follows

$$\dot{\beta} = -\frac{C_f + C_r}{m V_x} \beta + \left(\frac{\ell_r C_r - \ell_f C_f}{m V_x^2} - 1 \right) r + \frac{C_f}{m V_x} \delta, \quad (9a)$$

$$\dot{r} = \frac{\ell_r C_r - \ell_f C_f}{I_z} \beta - \frac{\ell_r^2 C_r + \ell_f^2 C_f}{I_z V_x} r + \frac{\ell_f C_f}{I_z} \delta, \quad (9b)$$

where β denotes the slip angle, and C_f and C_r denote the cornering stiffness of the front wheels and the rear wheels, respectively.

IV. PATH PLANNING

A single lane change maneuver is required to be smooth and safe. A mathematical description of this maneuver naturally leads to two separated tasks, namely, path planning and path tracking.

TABLE I
AVAILABLE PATH GENERATING METHODS.

Method: circular trajectory [38].
Description: two constant radius arcs connected with a line segment.
Advantage: short computation time.
Disadvantage: discontinuities of the curvature.
Method: arcs combination [38].
Description: several arcs having different radii.
Advantage: smoother transition between arcs.
Disadvantage: discontinuities of the curvature.
Method: arcs and clothoids [39].
Description: constant radius arcs connected with clothoids.
Advantage: continuous curvature.
Disadvantage: More computing time.
Method: polynomial trajectory [38].
Description: the path is a 5th order polynomial trajectory.
Advantage: continuous curvature.
Disadvantage: hard to modify the shape of trajectory.
Method: Joint clothoids [40].
Description: four connected clothoids (use polynomial approximation).
Advantage: continuous curvature, lower costs than using clothoids.
Disadvantage: No obvious disadvantage.
Method: Joint Bézier curves [41], [42].
Description: two symmetrical cubic Bézier curves.
Advantage: curvature is continuous and minimized.
Disadvantage: curvature constraint is not guaranteed.

There are many path generating methods available in the literature for lane changing [38]–[42]. We summarize the

advantages and disadvantages of some typical methods in Table I. These approaches either plan a path without guaranteeing continuity or smoothness of the curvature [38], [41], or require additional time to compute clothoids [39], [40] or other tuning parameters [42].

Regarding path planning for lane-switching, one important design objective is to limit the maximum curvature of the path, which depends on the road friction conditions and the velocity of the vehicle. We also expect the curvature to be continuous in order to have better smoothness and riding comfort. In this work we use Bézier curves for path planning during lane-switching.

A. Joint Quadratic Bézier curves

Following on the results of [43], we use piecewise quadratic Bézier curves to plan the path for lane switching.

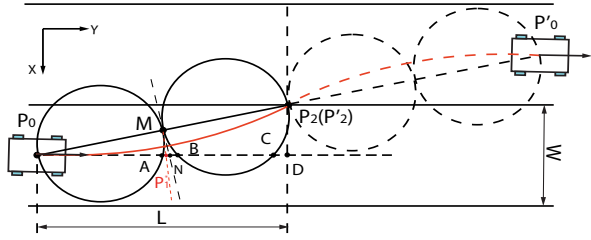


Fig. 5. Path planning for the single lane change.

Assume the path planning problem shown in Fig. 5. The lane width is denoted by W . Without loss of generality, we can assume the trajectory is symmetric with respect to the point P_2 , which is located at a distance L in front of the vehicle. The quadratic Bézier curve γ is given by

$$\gamma(P_1, t) = (1-t)^2 P_0 + 2t(1-t) P_1 + t^2 P_2, \quad t \in [0, 1]. \quad (10)$$

Mathematically, we want to solve the following problem,

$$P_1^* = \arg \min_{P_1 \in \overline{P_0 D}} \max_{t \in [0, 1]} \kappa(P_1, t) = \frac{|\gamma'(P_1, t) \times \gamma''(P_1, t)|}{\|\gamma'(P_1, t)\|^3}, \quad (11)$$

where κ denotes the curvature. Let M denote the midpoint of the segment $\overline{P_0 P_2}$. The next result refers to the geometry shown in Fig. 5 and can be used to determine the optimal point P_1^* .

Theorem 4.1: Let N be the intersection of $\overline{P_0 D}$ and the perpendicular bisector to the segment $\overline{P_0 P_2}$. The optimal choice of P_1^* is on the segment $\overline{AN}$, which satisfies $\angle NMP_1^* = \tan^{-1} \frac{\sqrt{8 \tan^2 \angle P_2 P_0 D + 9} - 3}{2 \tan \angle P_2 P_0 D}$. The minimal maximum curvature $\bar{\kappa}^*$ is given by

$$\bar{\kappa}^*(P_1^*) = \frac{3 \tan \angle NMP_1^*}{\|P_0 M\| \cos \angle NMP_1^*}. \quad (12)$$

The maximum curvature $\bar{\kappa}(P_1)$ decreases monotonically as P_1 moves from P_0 to P_1^* , and then increases monotonically as P_1 moves from P_1^* to D .

Proof: The proof is omitted owing to space limitations and can be found in [44]. $\square$

The path planning algorithm is summarized as follows:

Algorithm 1 Path Generation Using Joint Quadratic Bézier Curves

Input: $W, P_0, \bar{\kappa}_{\max}$

Output: $L, \gamma(t)$

- 1: $\angle P_2 P_0 D, \angle NMP_1^* \leftarrow$ by solving:

$$\begin{cases} \angle NMP_1^* = \tan^{-1} \frac{\sqrt{8 \tan^2 \angle P_2 P_0 D + 9} - 3}{2 \tan \angle P_2 P_0 D}, \\ 3 \tan \angle NMP_1^* \\ \|P_0 M\| \cos \angle NMP_1^* - \bar{\kappa}_{\max} = 0, \\ \|P_0 M\| = \frac{W}{2 \sin \angle P_2 P_0 D}. \end{cases}$$
- 2: $L \leftarrow W / \tan \angle P_2 P_0 D, \angle AMP_1^* \leftarrow \angle P_2 P_0 D - \angle NMP_1^*$
- 3: $P_2 \leftarrow (\pm W/2, L), P_1 \leftarrow P_0 + (0, L/2 + W \tan \angle AMP_1^*/4)$
- 4: $\zeta(t) = (1-t)^2 P_0 + 2t(1-t) P_1 + t^2 P_2, \quad t \in [0, 1]$
- 5: Curve: $\gamma(\tau) = \begin{cases} \zeta(2\tau) & \tau \in [0, 0.5], \\ (\pm W, 2L) - \zeta(2-2\tau) & \tau \in [0.5, 1]. \end{cases}$

B. Fourth Order Bézier Curves

The joint quadratic Bézier curves are easy to implement for path planning. However, this approach only guarantees C^1 continuity and the curvature is not continuous at both P_0 and P_2 . In this section, we use fourth order Bézier curves to generate paths with continuous curvature.

A typical fourth order Bézier curve is constructed using five control points, namely, $P_0, \dots, P_4$, and is represented by

$$\gamma(t) = \sum_{i=0}^4 B_i^4(t) P_i, \quad t \in [0, 1]. \quad (13)$$

In order to generate a symmetric path (see Fig. 6), we let $P_2 = (P_0 + P_4)/2$, and $\vec{\Delta} = P_1 - P_0 = P_4 - P_3$. Equation (13) is then simplified as

$$\gamma(t) = (1-t)^2(1+2t)P_0 + 4t(1-t)(1-2t)\vec{\Delta} + t^2(3-2t)P_4. \quad (14)$$

We then calculate the curvature as follows,

$$\kappa(t) = \frac{|\gamma'(t) \times \gamma''(t)|}{\|\gamma'(t)\|^3} = \frac{24(1-2t)|\vec{\Delta} \times (P_4 - P_0)|}{\|\gamma'(t)\|^3}. \quad (15)$$

From (15) one sees that $\kappa(0.5) = 0$ and $\kappa(0) = -\kappa(1) = 3|\vec{\Delta} \times (P_4 - P_0)|/8\|\vec{\Delta}\|^3$ (property of symmetry). Hence, we can analyze the first half of the curve by letting $t \in [0, 0.5]$ since the curve is symmetric. Taking the partial derivative of $\kappa(t)$ with respect to t yields

$$\kappa'(t) = 24|\vec{\Delta} \times (P_4 - P_0)| \frac{F(t)}{\|\gamma'(t)\|^5}, \quad t \in [0, 0.5], \quad (16)$$

where $F(t) = -2\|\gamma'(t)\|^2 - 3(1-2t)\gamma'(t) \cdot \gamma''(t)$. It is easy to see that the sign of $\kappa'(t)$ is the same as $F(t)$. We then simplify this equation, to obtain

$$F(x(t)) = 16 \left(90\|\vec{\Gamma}\|^2 x(t)^2 - (27\|\vec{\Gamma}\|^2 - 24\vec{\Delta} \cdot \vec{\Gamma})x(t) - 2\|\vec{\Delta}\|^2 - 9\vec{\Delta} \cdot \vec{\Gamma} \right). \quad (17)$$

where $x(t) = t(1-t) \in [0, 0.25]$ for $t \in [0, 0.5]$ and $\vec{\Gamma} \triangleq (P_4 - P_0)/2 - 2\vec{\Delta}$.

Theorem 4.2: The curvature of the fourth order Bézier curve $\kappa(t)$ is monotonically decreasing from $t = 0$ to $t = 1$ if and only if $\|\vec{\Delta}\| \leq 9L/32$. The maximum curvature is given by $\kappa(0) = 3|\vec{\Delta} \times (P_4 - P_0)|/8\|\vec{\Delta}\|^3$.

Proof: We just need to show the monotonicity on $t \in [0, 0.5]$ due to symmetry. To this end, we can equivalently show

that $F(x(t)) \leq 0$ for $x(t) \in [0, 0.25]$ and $\|\tilde{\Delta}\| \leq 9L/32$. Since $F(x(t))$ is parabolic and the coefficient of the second order term is positive, we need to ensure that $F(x(t)) \leq 0$ at the two endpoints of the interval of $x(t)$. Hence,

$$F(x=0.25) = -\|3\sqrt{2}\tilde{\Gamma} + 4\sqrt{2}\tilde{\Delta}\|^2 \leq 0, \quad (18a)$$

$$F(x=0) = -8\left(9L\|\tilde{\Delta}\| - 32\|\tilde{\Delta}\|^2\right) \leq 0, \quad \forall \|\tilde{\Delta}\| \leq 9L/32. \quad (18b)$$

and the result follows. $\square$

For the case that $\|\tilde{\Delta}\| > 9L/32$, the maximum curvature is obtained at $\gamma'(t^*) = F'(x^*(t^*)) = 0$, where

$$x^*(t^*) = t^*(1 - t^*) = \frac{9\|\tilde{\Gamma}\|^2 - 8\tilde{\Delta} \cdot \tilde{\Gamma} - \sqrt{81\|\tilde{\Gamma}\|^4 + 64(\tilde{\Delta} \cdot \tilde{\Gamma})^2 + 216\|\tilde{\Gamma}\|^2(\tilde{\Delta} \cdot \tilde{\Gamma}) + 80\|\tilde{\Delta}\|^2\|\tilde{\Gamma}\|^2}}{60\|\tilde{\Gamma}\|^2}. \quad (19)$$

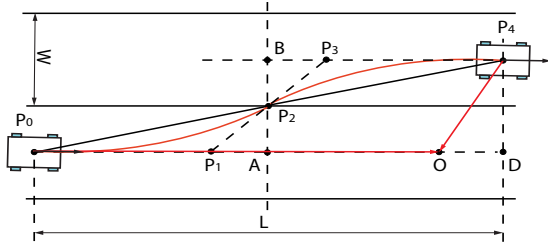


Fig. 6. A symmetric fourth order Bézier curve.

Theorem 4.3: The fourth order Bézier curve $\kappa(t)$ has the minimal jerk energy when $\|\tilde{\Delta}^*\| = L/4$, where $\tilde{\Delta}^*$ mathematically solves the following minimization problem

$$\tilde{\Delta}^* = \arg\min_{\tilde{\Delta}} E(\gamma) = \int_0^1 \|\gamma'''(\tilde{\Delta}, t)\|^2 dt. \quad (20)$$

Proof: It follows from the equation (3.2) in [45] that, the jerk energy of the fourth order Bézier curve generated using control points $P_0, \dots, P_4$ can be represented by

$$E(\gamma) = 3350\|Q_0\|^2 + 1440\|Q_1\|^2, \quad (21)$$

where $Q_0 = P_0 - 4P_1 + 6P_2 - 4P_3 + P_4$ and $Q_1 = -P_0 + 2P_1 - 2P_3 + P_4$. Recall that $P_2 = (P_1 + P_4)/2$, $P_1 = P_0 + \tilde{\Delta}$ and $P_3 = P_4 - \tilde{\Delta}$, then one can simplify the expressions for Q_0 and Q_1 , which are given by $Q_0 = (0, 0)$, $Q_1 = 4\tilde{\Delta} - (P_4 - P_0)$. Hence, in order to minimize $E(\gamma)$ we just need to minimize $\|Q_1\|$. The minimal $\|Q_1\|$ ($\|P_4O\|$) is obtained under the condition that $\overrightarrow{P_4O} \perp \tilde{\Delta}$ (see Fig. 6), which represents the minimal distance from P_4 to P_0D . This result indicates $\|\tilde{\Delta}^*\| = L/4$. $\square$

The curvature of the fourth order Bézier curve is not zero at the two endpoints P_0 and P_4 (see Fig. 6). Consequentially, the transition between the Bézier curve and the straight line is not smooth. One may consider to use clothoids to smooth the transition. However, clothoid computation takes time. In this paper we propose another method by extending the fourth order Bézier curve in order to obtain zero curvature at both the two endpoints.

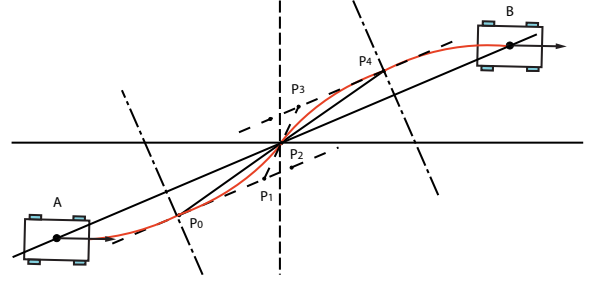


Fig. 7. Bézier curve reconstruction for smooth transition at endpoints.

Algorithm 2 Path Generation Using 4th Order Bézier Curves

Input: $W, A, \bar{\kappa}_{\max}$

Output: $B, L, \gamma(t)$

- 1: $K \leftarrow \sqrt{\left(\sqrt{1 + \frac{144}{W^2\bar{\kappa}_{\max}^2}} - 1\right)/2}$
- 2: $L \leftarrow KW, d \leftarrow \frac{\sqrt{1+K^2}}{4K}W, \ell \leftarrow 2Kd, B \leftarrow A + (\pm W, L)$
- 3: $\theta \leftarrow \arctan \frac{\ell}{2d}, \alpha \leftarrow 2\theta - \pi/2$
- 4: $\tilde{\Delta} \leftarrow (0, \ell/4)$ (Minimal jerk solution)
- 5: $P_0 \leftarrow (0, 0), P_4 \leftarrow (\pm d, \ell), P_2 \leftarrow (P_0 + P_4)/2, P_1 \leftarrow P_0 + \tilde{\Delta}, P_3 \leftarrow P_4 - \tilde{\Delta}$
- 6: $\hat{\gamma}(t) = \sum_{i=0}^4 B_i^4(t)P_i, t \in [0, 1]$
- 7: $\tilde{i} \leftarrow (1, 0)$
- 8: $\zeta(\tau) = \begin{cases} 2(\hat{\gamma}(0.5-2\tau) \cdot \tilde{i}, 0) - \hat{\gamma}(0.5-2\tau) & \tau \in [0, 0.25], \\ \hat{\gamma}(2\tau-0.5) & \tau \in [0.25, 0.75], \\ 2(\hat{\gamma}(2.5-2\tau) \cdot \tilde{i}, \ell) - \hat{\gamma}(2.5-2\tau) & \tau \in (0.75, 1]. \end{cases}$
- 9: Translation: $\zeta(\tau) \leftarrow \zeta(\tau) + A + (\mp d/2, \ell/2)$
- 10: Rotation: $\gamma(t) \leftarrow \zeta(\tau)$ rotated about A by $\mp(\pi/2 - \theta)$

To this end, notice that the curvature at the midpoint of a symmetric fourth order Bézier curve is always zero. The curvature of the curve between P_0 and P_2 changes gradually from $\kappa(0)$ to zero (similarly to a clothoid). Based on this fact, one can use half of the Bézier curve to smooth the path, as shown in Fig. 7. We reflect the curve segments $\overline{P_0P_2}$ and $\overline{P_2P_4}$ about the axes passing through P_0 and P_4 , respectively. Such designed path from A to B is everywhere C^2 continuous, with the curvature $\kappa(A) = \kappa(B) = 0$ at the endpoints. We summarize the path planning algorithm using fourth order Bézier curves subject to the given maximum curvature constraints $\bar{\kappa}_{\max}$ in Algorithm 2.

V. CONTROLLER DESIGN

Beyond lane-switching, the other actions such as maintaining, accelerating and braking are taking place in a single lane. These actions can be completed by maintaining the longitudinal speed of the vehicle, associated with implementing certain lane tracking controllers (i.e., waypoint follower, two-point visual driver model, etc.). Here, we design the speed control and lane-switching control separately.

A. Speed Control

Speed control is designed to achieve maintaining, accelerating and braking actions. Based on this fact, one may assume that the vehicle behind will always yield to the vehicle in front. Consequentially, we only need to check the speed of the vehicle in front in order to decide the maximum allowed speed of the vehicle we want to control.

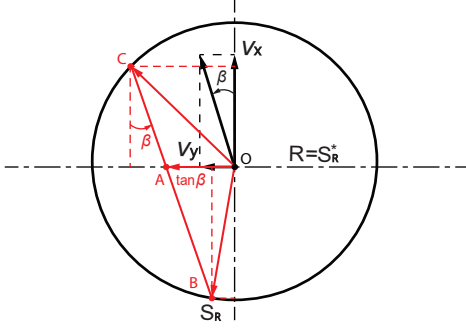


Fig. 8. Slip ratio circle.

Maintaining: The design objective for maintaining control is to keep a certain constant speed or certain constant distance from the vehicle in front.

Accelerating: The design objective for accelerating control is to speed up the vehicle and maintain a certain high speed afterwards if there is speed limit. The target speed of the vehicle in steady state should not exceed the speed of the vehicle in front.

Braking: The design objective for braking control is to slow down the vehicle and maintain a certain low speed. Under the assumption that every vehicle in traffic will yield to the vehicle in front, there is no need to consider the speed of the rear vehicle before taking braking action. Nevertheless, we still have to maintain the minimum safe distance from the front vehicle during braking.

Below we only show the controller design for maintaining constant distance from the vehicle in front. We assume that ΔL is the desired distance between vehicle A and B.

We let $e_1 = Y_A - Y_B - \Delta L$, $e_2 = V_A - V_B$. The error dynamics are given by

$$\dot{e}_1 = e_2, \quad \dot{e}_2 = \dot{V}_A - \dot{V}_B = \lambda_1 e_1 + \lambda_2 e_2, \quad (22)$$

By designing λ_1 and λ_2 , one can drive the errors $e_1 \rightarrow 0$ and $e_2 \rightarrow 0$ as time $t \rightarrow \infty$. Now, let us assume that the front wheel steer angle is small and that the vehicle's lateral motion can be neglected during the distance maintaining task. For vehicles having a rear wheel drive differential type, the longitudinal dynamics can be simplified to

$$m_B \dot{V}_B = f_{Rx}. \quad (23)$$

Let $\dot{V}_B = \dot{V}_A - \lambda_1 e_1 - \lambda_2 e_2$. Since the tire force must be bounded due to the friction condition, we define the following saturation function,

$$\text{sat}(\dot{V}_B) = \begin{cases} \bar{f}_{Rx}/m_B, & f_{Rx} \geq \bar{f}_{Rx}, \\ \dot{V}_A - \lambda_1 e_1 - \lambda_2 e_2, & \underline{f}_{Rx} < f_{Rx} < \bar{f}_{Rx}, \\ \underline{f}_{Rx}/m_B, & f_{Rx} \leq \underline{f}_{Rx}, \end{cases} \quad (24)$$

where $\bar{f}_{Rx}$ and $\underline{f}_{Rx}$ denote the upper and lower bounds for f_{Rx} (to be discussed later). The wheel dynamics is given as follows,

$$I_w \dot{\omega} = T_R - f_{Rx} R_w, \quad (25)$$

where T_R is the torque on the rear wheel, and I_w and R_w are the rotational inertia and the radius of the rear wheel,

respectively. Under the assumption of zero-slip rolling condition, the equation $V_B = \omega R_w$ holds. The control torque is determined as follows,

$$T_R = \text{sat}(\dot{V}_B) R_w (m_B + I_w / R_w^2). \quad (26)$$

Next, we discuss how to determine $\bar{f}_{Rx}$ and $\underline{f}_{Rx}$ if the lateral velocity V_y and the longitudinal velocity V_x (approximated by V_B) are given. The longitudinal load transfer arising from the longitudinal acceleration indicates the following equation,

$$f_{Rx} = \mu_{Rx} f_{Rz}, \quad f_{Rz} = \frac{f_{Rx} h + m g \ell_f}{\ell_f + \ell_r}, \quad (27)$$

where h is the height of the mass center and $\mu_{Rx} = -\mu_R s_{Rx} / s_R$. It follows from (27) that

$$f_{Rx} = \frac{m g \ell_f}{(\ell_f + \ell_r) / \mu_{Rx} - h}. \quad (28)$$

Based on the fact that the wheel base $\ell_f + \ell_r$ is larger than the height of the mass center h , the maximum longitudinal tire force $\bar{f}_{Rx}$ ($\underline{f}_{Rx}$) is obtained when μ_{Rx} reaches its maximum value $\bar{\mu}_{Rx}$ ($\underline{\mu}_{Rx}$).

Furthermore, we assume that the lateral sideslip is small, and for the sake of simplicity, we assume that the total slip of the rear tire does not exceed s_R^* , where s_R^* corresponds to the peak of μ_R . This assumption indicates that μ_R increases monotonically with $|s_R|$. Based on the definition of the slip ratio in (6), we can derive the following equation

$$s_{Ry} = (1 + s_{Rx}) \frac{V_{Ry}}{V_{Rx}} \approx (1 + s_{Rx}) \tan \beta. \quad (29)$$

In order to find $\bar{\mu}_{Rx}$ ($\underline{\mu}_{Rx}$), we plot the slip ratio circle in Fig. 8. The norm of the red arrow starting from O denotes $|s_R|$. The equation in (29) indicates that, when the value of s_{Rx} changes, the arrow representing s_R moves along the segment $\overline{BC}$. When $s_{Rx} = 0$, $|s_R| = |s_{Ry}| = \tan \beta$ (for $\beta > 0$).

Recall that $\mu_{Rx} = -\mu_R s_{Rx} / s_R$, since both $-s_{Rx} / s_R$ and μ_R reaches their maximum values at B, the upper bound $\bar{\mu}_{Rx}$ is therefore obtained at B (maximal acceleration force). Similarly, the lower bound $\underline{\mu}_{Rx}$ is obtained at C (maximal braking force). The results are summarized as follows

$$\bar{f}_{Rx} = \frac{m g \ell_f \mu_R^* \left(\sqrt{\sec^2 \beta s_R^{*2} - \tan^2 \beta} + \tan^2 \beta \right)}{(\ell_f + \ell_r) s_R^* \sec^2 \beta - h \mu_R^* \left(\sqrt{\sec^2 \beta s_R^{*2} - \tan^2 \beta} + \tan^2 \beta \right)}, \quad (30a)$$

$$\underline{f}_{Rx} = - \frac{m g \ell_f \mu_R^* \left(\sqrt{\sec^2 \beta s_R^{*2} - \tan^2 \beta} - \tan^2 \beta \right)}{(\ell_f + \ell_r) s_R^* \sec^2 \beta + h \mu_R^* \left(\sqrt{\sec^2 \beta s_R^{*2} - \tan^2 \beta} - \tan^2 \beta \right)}, \quad (30b)$$

where μ_R^* is the peak friction coefficient, namely, the magnitude D of the magic formula (7), and $s_R^* = (1/B) \tan(\pi/2C)$ (let $S_h = S_v = E = 0$).

B. Lane-Switching Control

We generate a smooth path for lane switching using Algorithm 1 or 2. We still need to design the tracking controller to follow the Bézier curve. To this end, we use

Fig. 9 to calculate the heading error $\Delta\psi$ and the lateral error Δy [46].

In Fig. 9, the red solid curve denotes the reference path we want to track, ℓ_s denotes the preview distance along the vehicle's heading direction, ψ_t denotes the angle between the tangent direction of the reference curve at the current location (M) and the X_I axis. The lateral error Δy denotes the distance between the preview point A and the reference point B on the target path. The heading error is given by $\Delta\psi = \psi_t - \psi$, where ψ denotes the vehicle's yaw angle. The

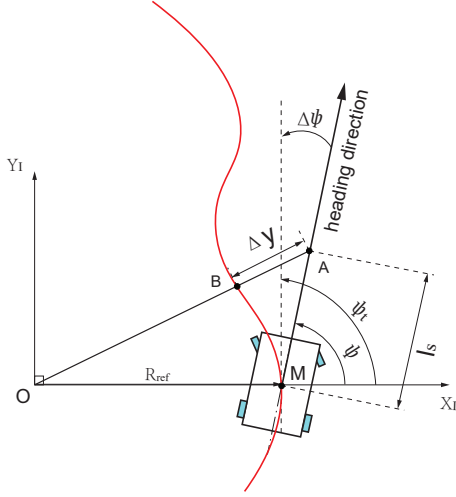


Fig. 9. Path tracking error.

dynamics equations for Δy and $\Delta\psi$ can be approximately given by

$$\Delta\dot{y} = -V_x(\beta - \Delta\psi) - \ell_s r + V_x \ell_s \rho_{ref}, \quad (31a)$$

$$\Delta\dot{\psi} = V_x \rho_{ref} - r. \quad (31b)$$

where $\rho_{ref} = 1/R_{ref}$ denotes the reference road curvature. We then combine the vehicle model in (9a)-(9b) and the perception model in (31a)-(31b). We treat the road curvature ρ_{ref} as a noise term and write the dynamics equation in the form of $\dot{x} = Ax + Bu + Ew$, where the state is $x = [\beta, r, \Delta y, \Delta\psi]^T$, the control is $u = \delta$ and the noise is $w = \rho_{ref}$.

We design the path tracking control following the approach of [47], with the aim of eliminating the tracking error at the near preview point, namely, $\lim_{t \rightarrow \infty} \Delta y(t) = 0$. Such a controller is referred to as the Output Regulation Theory (ORT) controller in that paper. The control input δ of the ORT controller is a linear combination of a feedback term and a feedforward term as follows,

$$\delta = Gx + H\rho_{ref}, \quad (32)$$

where the matrix G is chosen such that the matrix $A + BG$ is Hurwitz. Then H is determined by solving the following equations, for some matrices Γ and Π

$$H = \Gamma - G\Pi, \quad (33a)$$

$$A\Pi + B\Gamma + E = 0, \quad (33b)$$

$$C\Pi = 0. \quad (33c)$$

The unknown G and H can also be determined by solving a series of linear matrix inequalities (see [46]).

VI. RESULTS AND ANALYSIS

In this section we implement the previous RL algorithm for the traffic model of Section II to determine the optimal policy. We demonstrate the policy using a traffic simulator along with implementing the path planning algorithms in Section IV and the low level controllers designed in Section V.

A. Path Planning

We implemented both Algorithms 1 and 2 to plan paths for lane switching. The maximum curvature of each curve is assigned different values, namely, $\bar{\kappa}_{max} = 0.05, 0.1, 0.15, 0.2$ and 0.25 . The width of the lane is $W = 4$ [m]. We show only the fourth order Bézier curve paths in Fig. 10.

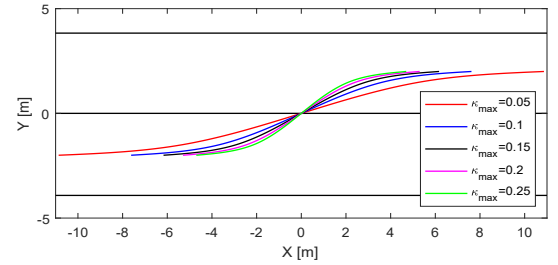


Fig. 10. Fourth order Bézier curves for lane switching.

We also plot the curvature profile for each path in Fig. 11. In contrast, the curvature profiles for the quadratic Bézier curve paths are plotted in Fig. 12.

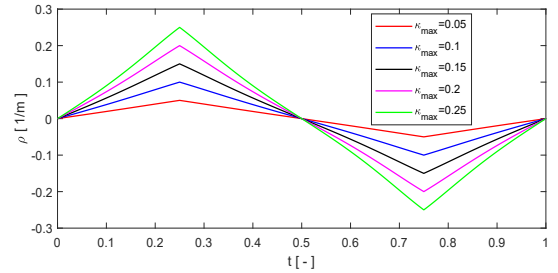


Fig. 11. The curvature of the fourth order Bézier curves.

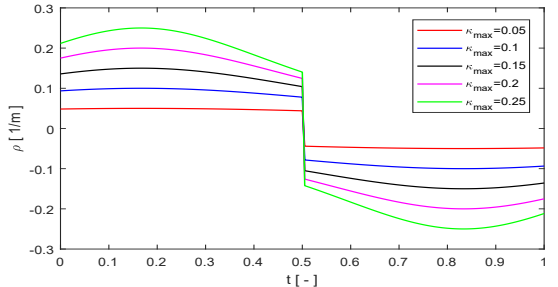


Fig. 12. The curvature of the quadratic Bézier curves.

The results in Fig. 11 and Fig. 12 show that the maximum curvature for each path satisfies the design requirement. This result validates the effectiveness of both Algorithm 1 and 2. Nevertheless, we notice that the quadratic Bézier curves are only C^1 continuous, since the curvature changes sign at the joint point of the two Bézier curves. The fourth order Bézier curves are C^2 continuous and the curvature is continuous everywhere. Since the paths have zero curvature at both the two endpoints, Algorithm 2 provides much better performance than Algorithm 1.

B. Path Tracking Control

We then implemented the tracking controller in Section V-B to follow the Bézier curves. The vehicle model parameters are summarized in Table II.

TABLE II
VEHICLE MODEL PARAMETERS.

$m[\text{kg}]$	850	total mass	$I_z[\text{kgm}^2]$	1401	rotational inertia
$I_{\text{wr}}[\text{kgm}^2]$	0.6	wheel rotational inertia	$\ell_f[\text{m}]$	1.5	distance to front axle
$h[\text{m}]$	0.5	height of mass center	$\ell_s[\text{m}]$	1	preview distance
$L[\text{m}]$	2.4	wheel base	$R[\text{m}]$	0.311	wheel radius
Tire force model parameters $B = 3.9$ $C = 5.4$ $D = 0.7$ $E = S_h = S_v \approx 0$					

We first generated several reference paths using both the joint quadratic Bézier curves and the fourth order Bézier curves. Next, the tracking controller in V-B was implemented to track the reference paths for lane switching. Since the fourth order Bézier curves have better smoothness, we show only the result for tracking the fourth order Bézier curves (see Figure 13).

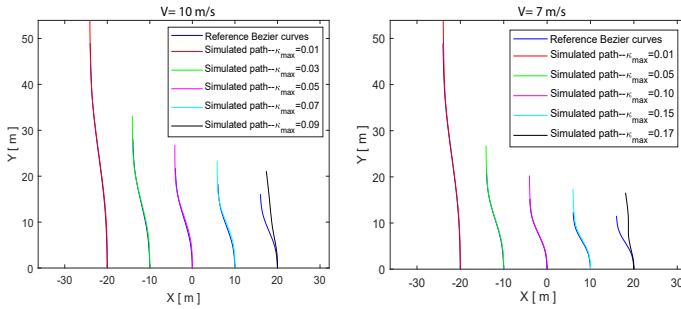


Fig. 13. Tracking control for fourth order Bézier curves.

Fig. 13 (left) indicates that, given the vehicle velocity (approximately fixed at 10 [m/s]), the controller is not able to track the reference path when the maximum curvature is larger than 0.09 [1/m]. Tracking errors for small curvature paths are satisfactory. Next, we reduce the velocity of the vehicle to 7 [m/s] and implement the controller again. The results are plotted in Fig. 13 (right). One sees from Fig. 13 that, a larger maximum curvature may be allowed for tracking control if the velocity of the vehicle is lower.

We recall that the maximum friction coefficient between the tire and the ground was chosen as $\mu_{\text{max}} = D = 0.7$ in Table II. One can evaluate the maximum curvature for path tracking using $\bar{\kappa}_{\text{max}} = \mu_{\text{max}}g/V^2$, which further indicates that $\bar{\kappa}_{\text{max}} = 0.07$ [1/m] and 0.14 [1/m] for $V = 10$ [m/s] and 7 [m/s], respectively. This result agrees with Fig. 13. The fourth order Bézier curves have better smoothness, and the

controller designed in Section V-B is able to track the Bézier curves even if κ_{max} is close to $\bar{\kappa}_{\text{max}}$.

C. Decision Making from Reinforcement Learning

We implemented the previous RL algorithm for the MDP problem of Section II to determine the optimal policy. We only demonstrate two driving styles using RL, namely, highway overtaking and tailgating. We design the weights w_1 and w_2 for the features selected in Section II-C to learn the two driving strategies, respectively. Table III provides the features and weights used in this study.

TABLE III
THE DESIGN OF THE REWARD FUNCTION.

$\Phi(s, a)$	w_1	Interpretation	w_2	Interpretation
accelerate	0.075	Encourage accelerating	0.05	Encourage accelerating
brake	-0.625	Less braking	-0.5	Less braking
maintain	0	NA	0	NA
left-turn	-0.05	Less lane-switching	-0.025	Less lane-switching
right-turn	-0.05	Less lane-switching	-0.025	Less lane-switching
HV position	0	NA	0	NA
overtake	0.05	Prefer shorter path	0.025	Prefer shorter path
tailgate	0	NA	0.225	Encourage tailgating
accident	-0.15	Penalize accident	-0.15	Penalize accident

The desired overtaking behavior is summarized as follows: 1) The HV takes up the cell in front of it if this cell is vacant; 2) The HV keeps its current speed if a TV in front is detected; 3) If only one adjacent lane is available for overtaking, the HV switches lane first and then accelerates to overtake the TV vehicle in front; 4) If both the two adjacent lanes are available for overtaking in a corner, the HV should use the lane that is closer to the inner curb of the road; 5) No lane-switching can occur unless the HV is overtaking the TV in front; 6) The HV avoids braking to take up the cell behind it; 7) The action of HV should not cause any accident.

The desired tailgating behavior is summarized as follows: 1) The HV keeps its current speed if a TV vehicle in front is detected; 2) If there is no available TVs to tailgate, the HV takes up the cell in front of it if this cell is available; 3) If the HV does not detect a TV in front of it, the HV tries to switch lane to tailgate a TV in the adjacent lanes; 4) If both the two adjacent lanes are available for tailgating in a corner, the HV should use the lane that is closer to the inner curb of the road; 5) No lane-switching can occur unless the HV is tailgating a TV in the adjacent lanes; 6) The HV avoids braking to take up the cell behind it; 7) The action of HV should not cause any accident.

We used Q-learning to determine the optimal policies π_1^* and π_2^* corresponding to w_1 and w_2 , respectively, with a discount rate of $\gamma = 0.5$, learning rate $\alpha = 0.75$, and parameter $\epsilon = 8e^{-2}$ for the ϵ -greedy principle (see [26]).

Next, we implemented the policies π_1^* and π_2^* in simulation. In order to save the space, this paper only shows the result for overtaking by implementing π_1^* . Fig. 14 shows two driving scenarios with two single rows of images.

The images in the first row of Fig. 14 show a driving scenario where there is not a TV in front of the HV (green). The HV overtakes the two trucks in the front on the neighboring lanes and then maintains a certain high speed. The images in the second row show a driving scenario

where one TV is driving in front of the HV. Since the HV is not in a corner, it is free to overtake the front TV using either of the two adjacent lanes. One observes from Fig. 14 that the HV switches to the left lane first, and then accelerates to overtake the front yellow TV. The HV has to brake to maintain a minimal distance from the newly detected dark blue TV in the front. These driving behaviors using π_1^* validate the design of the reward function and also validate the use of “dynamic cells” to deal with complicated traffic information¹ (i.e., different vehicle velocities, sizes and signals).

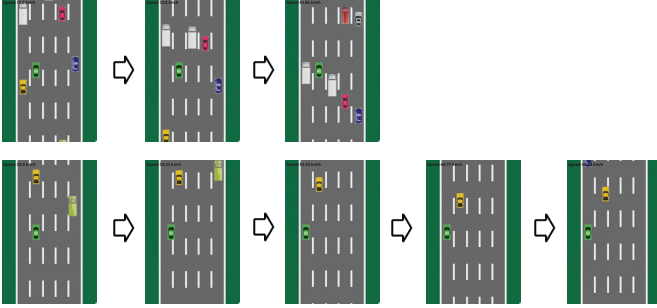


Fig. 14. Overtaking scenarios in simulation by implementing π_1^* .

VII. CONCLUSION

We use a stochastic Markov decision process to characterize the driving behaviors of autonomous vehicles in traffic. The desired driving styles are achieved using reinforcement learning. The “dynamic cell” approach is proposed to address different vehicle velocities, vehicle sizes and driver intents in traffic. We also take into consideration the different road geometry such that we are able to show more diverse driving styles when the road curvature changes.

By designing the reward function of the driver, we successfully show some typical driving behaviors such as overtaking and tailgating. We have demonstrated these driving behaviors using a five-lane road with each TV implementing a random policy on a driving simulator based on Pygame. In order to complete lane-switching, we separate the task into a path planning task and a tracking control task. We then formulate two different algorithms to generate smooth paths using both joint quadratic Bézier curves and fourth-order Bézier curves subject to a certain maximum curvature constraint. The joint quadratic Bézier curves use a smaller space to generate a path. Nevertheless, this path is only C^1 continuous and it is more difficult to track it than a path generated using fourth-order Bézier curves, which is C^2 continuous and therefore has better smoothness properties. We also design a path tracking control based on the output regulation theory. Simulation results validate the effectiveness of both the path planning algorithms and the design of the controller.

Future work will focus on improving the work to incorporate pedestrians, traffic signals and more road intersections.

¹The videos are available on the DCSL youtube channel: <https://www.youtube.com/watch?v=maUt8Cac2WU> and <https://www.youtube.com/watch?v=393MJA6Kp3I>.

ACKNOWLEDGMENT

This work is supported by National Science Foundation award CPS-1544814 and the Ford Motor Company.

REFERENCES

- [1] S. D. Pendleton, H. Andersen, X. Du, X. Shen, M. Meghjani, Y. H. Eng, D. Rus, and M. H. Ang, “Perception, planning, control, and coordination for autonomous vehicles,” *Machines*, vol. 5, no. 1, p. 6, 2017.
- [2] A. J. Hawkins. (2017) Google’s new self-driving minivans will be hitting the road at the end of January 2017. [Online]. Available: <https://www.theverge.com/2017/1/8/14206084/google-waymo-self-driving-chrysler-pacifica-minivan-detroit-2017>
- [3] J. Berr. (2016) Uber’s audacious plan to replace human drivers. [Online]. Available: <https://www.cbsnews.com/news/ubers-audacious-plan-to-replace-human-drivers>
- [4] C. Thompson. (2016) Tesla just revealed new cars and Model 3 will have fully self-driving hardware. [Online]. Available: <http://www.businessinsider.com/tesla-announces-new-autopilot-self-driving-2016-10>
- [5] D. Lee. (2016) Ford’s self-driving car ‘coming in 2021’. [Online]. Available: <http://www.bbc.com/news/technology-37103159>
- [6] Auto Tech. (2017) 44 corporations working on autonomous vehicles. [Online]. Available: <https://www.cbinsights.com/research/autonomous-driverless-vehicles-corporations-list>
- [7] E. A. Wan and R. Van Der Merwe, “The unscented Kalman filter for nonlinear estimation,” in *Adaptive Systems for Signal Processing, Communications, and Control Symposium*, Alberta, Canada, October 1–4, 2000, pp. 153–158.
- [8] G. Chowdhary and R. Jategaonkar, “Aerodynamic parameter estimation from flight data applying extended and unscented Kalman filter,” *Aerospace Science and Technology*, vol. 14, no. 2, pp. 106–117, 2010.
- [9] C. You, J. Lu, and P. Tsiotras, “Nonlinear driver parameter estimation and driver steering behavior analysis for ADAS using field test data,” *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 5, pp. 686–699, 2017.
- [10] X. Hu, Y. Li, J. Shan, J. Zhang, and Y. Zhang, “Road centerline extraction in complex urban scenes from LiDAR data based on multiple features,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 11, pp. 7448–7456, 2014.
- [11] F. Castaño, G. Beruvides, R. E. Haber, and A. Artuñedo, “Obstacle recognition based on machine learning for on-chip LiDAR sensors in a cyber-physical system,” *Sensors*, vol. 17, no. 9, p. 2109, 2017.
- [12] S. Shalev-Shwartz, N. Ben-Zrihem, A. Cohen, and A. Shashua, “Long-term planning by short-term prediction,” *arXiv preprint arXiv:1602.01580*, 2016.
- [13] S. Brechtel, T. Gindele, and R. Dillmann, “Probabilistic MDP-behavior planning for cars,” in *14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Washington, DC, October 5–7 2011, pp. 1537–1542.
- [14] 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.
- [15] S. Karaman and E. Frazzoli, “Sampling-based algorithms for optimal motion planning,” *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.
- [16] L. Jaillet, J. Cortés, and T. Siméon, “Sampling-based path planning on configuration-space costmaps,” *IEEE Transactions on Robotics*, vol. 26, no. 4, pp. 635–646, 2010.
- [17] M. García, A. Viguria, and A. Ollero, “Dynamic graph-search algorithm for global path planning in presence of hazardous weather,” *Journal of Intelligent & Robotic Systems*, vol. 69, no. 1–4, pp. 285–295, 2013.
- [18] R. Cimurs, J. Hwang, and I. H. Suh, “Bézier curve-based smoothing for path planner with curvature constraint,” in *IEEE International Conference on Robotic Computing*, Taichung, Taiwan, April 10–12 2017, pp. 241–248.
- [19] J.-w. Choi, R. Curry, and G. Elkaim, “Path planning based on Bézier curve for autonomous ground vehicles,” in *World Congress on Engineering and Computer Science*, San Francisco, CA, October 22–24 2008, pp. 158–166.
- [20] J.-w. Choi, R. E. Curry, and G. H. Elkaim, “Continuous curvature path generation based on Bézier curves for autonomous vehicles,” *International Journal of Applied Mathematics*, vol. 40, no. 2, 2010.

- [21] T. Shim, G. Adireddy, and H. Yuan, "Autonomous vehicle collision avoidance system using path planning and model-predictive-control-based active front steering and wheel torque control," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of automobile engineering*, vol. 226, no. 6, pp. 767–778, 2012.
- [22] M. Gómez, R. V. González, T. Martínez-Marín, D. Meziat, and S. Sánchez, "Optimal motion planning by reinforcement learning in autonomous mobile vehicles," *Robotica*, vol. 30, no. 2, pp. 159–170, 2012.
- [23] C. Katrakazas, M. Quddus, W.-H. Chen, and L. Deka, "Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions," *Transportation Research Part C: Emerging Technologies*, vol. 60, pp. 416–442, 2015.
- [24] 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.
- [25] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press Cambridge, 1998, vol. 1, no. 1.
- [26] C. You, J. Lu, D. Filev, and P. Tsiotras, "Highway traffic modeling and decision making for autonomous vehicle using reinforcement learning," in *IEEE Intelligent Vehicles Symposium*, Changshu, China, June 26–30 2018.
- [27] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the 21st International Conference on Machine Learning*, Banff, Canada, July 4–8 2004, p. 1.
- [28] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning," in *AAAI*, vol. 8. Chicago, IL, 2008, pp. 1433–1438.
- [29] —, "Human behavior modeling with maximum entropy inverse optimal control," in *AAAI Spring Symposium: Human Behavior Modeling*, 2009, p. 92.
- [30] S. Levine, Z. Popovic, and V. Koltun, "Nonlinear inverse reinforcement learning with Gaussian processes," in *Advances in Neural Information Processing Systems*, 2011, pp. 19–27.
- [31] S. Levine and V. Koltun, "Continuous inverse optimal control with locally optimal examples," *arXiv preprint arXiv:1206.4617*, 2012.
- [32] C. J. C. H. Watkins, "Learning from delayed rewards," Ph.D. dissertation, King's College, Cambridge, 1989.
- [33] N. Li, D. Oyler, M. Zhang, Y. Yildiz, A. Girard, and I. Kolmanovsky, "Hierarchical reasoning game theory based approach for evaluation and testing of autonomous vehicle control systems," in *IEEE 55th Conference on Decision and Control*, Las Vegas, NV, December 12–14 2016, pp. 727–733.
- [34] D. W. Oyler, Y. Yildiz, A. R. Girard, N. I. Li, and I. V. Kolmanovsky, "A game theoretical model of traffic with multiple interacting drivers for use in autonomous vehicle development," in *American Control Conference (ACC)*, 2016, Boston, MA, July 6–8 2016, pp. 1705–1710.
- [35] E. Velenis, E. Frazzoli, and P. Tsiotras, "Steady-state cornering equilibria and stabilisation for a vehicle during extreme operating conditions," *International Journal of Vehicle Autonomous Systems*, vol. 8, no. 2-4, pp. 217–241, 2010.
- [36] E. Bakker, L. Nyborg, and H. B. Pacejka, "Tyre modelling for use in vehicle dynamics studies," SAE Technical Paper, Tech. Rep., 1987.
- [37] M. S. Burhaumudin, P. M. Samin, H. Jamaluddin, R. A. Rahman, S. Sulaiman, et al., "Integration of magic formula tire model with vehicle handling model," *International Journal of Research in Engineering and Technology*, vol. 1, no. 3, pp. 139–145, 2012.
- [38] W. Chee and M. Tomizuka, "Vehicle lane change maneuver in automated highway systems," Institute Of Transportation Studies, University of California, Berkeley, Tech. Rep. UCB-ITS-PRR-94-22, 1994.
- [39] T. Fraichard and A. Scheuer, "From Reeds and Shepp's to continuous-curvature paths," *IEEE Transactions on Robotics*, vol. 20, no. 6, pp. 1025–1035, 2004.
- [40] N. Montés and J. Tomero, "Lane changing using s-series clothoidal approximation and dual-rate based on Bézier points to controlling vehicle," *WSEAS Transactions on Circuits and Systems*, vol. 3, no. 10, pp. 2285–2290, 2004.
- [41] J. Chen, P. Zhao, T. Mei, and H. Liang, "Lane change path planning based on piecewise Bézier curve for autonomous vehicle," in *Vehicle Electronics and Safety (ICVES), 2013 IEEE International Conference on*, 2013, pp. 17–22.
- [42] D. Korzeniowski and G. Ślaski, "Method of planning a reference trajectory of a single lane change manoeuvre with Bézier curve," in

IOP Conference Series: Materials Science and Engineering, vol. 148, no. 1. IOP Publishing, 2016, p. 012012.

- [43] H. Deddi, H. Everett, and S. Lazard, "Interpolation with curvature constraints," Ph.D. dissertation, INRIA, 2000.
- [44] C. You, "Autonomous aggressive driving: Theory and experiments," Ph.D. dissertation, Georgia Institute of Technology, 2019.
- [45] H. Erişkin and A. Yücesan, "Bézier curve with a minimal jerk energy," *Mathematical Sciences and Applications E-Notes*, vol. 4, no. 2, pp. 139–148, 2016.
- [46] C. You and P. Tsiotras, "Optimal two-point visual driver model and controller development for driver-assist systems for semi-autonomous vehicles," in *American Control Conference*, Boston, MA, July 6–8 2016, pp. 5976–5981.
- [47] B. A. Francis, "The linear multivariable regulator problem," *SIAM Journal on Control and Optimization*, vol. 15, no. 3, pp. 486–505, 1977.



Changxi You received his PhD degree from the School of Aerospace Engineering, Georgia Institute of Technology, his B.S. and M.S. degrees from the Department of Automotive Engineering, Tsinghua University of China, and an M.S. degree from the Department of Automotive Engineering, RWTH-Aachen University of Germany. His current research interests are in system identification, aggressive driving, path planning and control of (semi)autonomous vehicle.

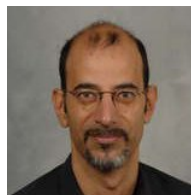


Jianbo Lu is a technical expert in Advanced Vehicle Controls at Ford Motor Company. He holds more than 100 US patents and numerous pending patent applications, and has published more than 70 journal and conference articles. He is a two-time recipient of Henry Ford Technology Reward. His research interests include automotive controls and sensing, adaptive vehicle systems, driver assistance systems, smart mobility, semi-autonomous and autonomous systems.



Dimitar Filev is Henry Ford Technical Fellow at the Ford Research & Innovation Center, Dearborn, Michigan. He is conducting research in computational intelligence, AI and intelligent control, and their applications to autonomous driving, vehicle systems, and automotive engineering. Dr. Filev has published over 250 journal articles and conference papers, and holds 106 US patents and numerous foreign patents. He is the recipient of the 2008 Norbert Wiener Award of the IEEE SMC Society and the 2015 Pioneer's Award of the IEEE

CIS Society. He received his PhD. degree in Electrical Engineering from the Czech Technical University in Prague in 1979. Dr. Filev is a Fellow of the IEEE and a member of the NAE. He was president of the IEEE Systems, Man, & Cybernetics Society (2016-2017).



Panagiotis Tsiotras is the David and Andrew Lewis Chair Professor at the School of Aerospace Engineering at the Georgia Institute of Technology (Georgia Tech). He has held visiting research appointments at MIT, JPL, INRIA Rocquencourt, and Mines ParisTech. His research interests include optimal control of nonlinear systems and ground, aerial and space vehicle autonomy. He has served in the Editorial Boards of the Transactions on Automatic Control, the IEEE Control Systems Magazine, the AIAA Journal of Guidance, Control and Dynamics and the journal Dynamics and Control. He is the recipient of the NSF CAREER award, the Outstanding Aerospace Engineer award from Purdue, and the IEEE Award for Technical Excellence in Aerospace Control. He is a Fellow of AIAA, IEEE, and AAS.