Modeling Driver Lane Change Behavior Using Inverse Reinforcement Learning

Zhaodong Zhou and Jun Chen*, *Senior Member, IEEE*
Department of Electrical and Computer Engineering
Oakland University
Rochester, MI 48309, USA
Email: {zhaodongzhou,junchen}@oakland.edu

***Abstract*—This paper presents an approach for modeling driver lane change behavior using inverse reinforcement learning (IRL). The driver behaviors are assumed to be the solution of an optimal control problem with unknown weights that simultaneously optimizes several objectives and follows a Bezier curve. The IRL is then used to identify the weights that best explain the driver's preference. To test the efficacy of the proposed methodology, expert trajectories are generated using simulation, which serves as the data set for this model training. Test results demonstrate the capability of the proposed methodology to reproduce each driver's individual trajectory, which can then be fed into low level motion control to provide personalized autonomous driving.**

I. Introduction

THE Autonomous driving technologies have rapidly advanced, now capable of perceiving environments, navigating roads, and adhering to traffic rules [1]–[7]. However, many studies indicate that driving styles vary significantly among individuals [8]. Therefore, to enhance the user experience, personalized driving has also become an area that is attracting considerable attention. To create personalized trajectories based on driver demonstrations, extensive research has focused on using neural networks for imitation learning [9], [10]. However, the imitation learning methods based on neural networks have poor interpretability [11].

Beyond pure imitation learning, inverse reinforcement learning (IRL) has gained popularity for learning from expert demonstrations across a variety of domains, including car following [12], [13], behavior prediction [14], [15], and trajectory planning [16]. IRL is a machine learning method used for understanding complex behaviors from observed actions instead of directly copying observed behaviors [17]–[19]. Unlike traditional reinforcement learning, which aims to find the best strategy given a known reward function [20], IRL focuses on inferring the underlying reward function based on observed behavior. This inferred reward function can then be used to generate driving behaviors that align with the observed preferences, as opposed to being merely replicas of the observed actions. On the other hand, lane change is one of the most frequent driving behaviors on the road which requires personalization [21]. The way individual drivers execute lane changes is affected by efficiency, ride comfort, and safety. Most previous works that modeled lane-change behavior heavily focused on the operational level, aiming at generating a comfortable and efficient lane-change trajectory [22], [23]. However, these works often fall short in capturing the preference of personal driving styles during lane change maneuvers [24].

This paper focuses on utilizing IRL for modeling driver lane change maneuvers. Bezier curves are employed to parameterize the lane change trajectories, which are chosen for their theoretical ability to create smooth and continuous paths [25]. Bezier curves are shaped by the control points; by adjusting these points, the Bezier curve can fit various driving scenarios [26]. The driver behaviors are then assumed to be the solution of an optimal control problem (OCP) that determines the best control points of the Bezier curve to deliver the driver's preference. The IRL method is then used to learn the weights for the cost function of the OCP, which reflects the individual differences among different drivers. The proposed IRL-based driver behavior model is tested using simulation data, where it is demonstrated that with the learned weights, the model can generate trajectories aligned with driver behavior for varying starting conditions.

The remainder of this paper is organized as follows. Section II formulates the Bezier curve path representation and optimal control problem for trajectory generation. Section III details the *MaxEnt* IRL algorithm and feature design. Section IV presents the simulation of expert demonstrations and model training process. Testing results are analyzed to evaluate how well the model reproduces driver behavior.

II. Lane Change Path Modeling

A. Bezier Curve

A Bezier curve is a parametric curve that uses the Bernstein polynomials as a basis with the control points defining the shape of the curve. The nth-Bezier curve is represented as follows:

$$B(t) = \sum_{i=0}^{n} P_i b_{i,n}(t) = \sum_{i=0}^{n} P_i \binom{n}{i} (1-t)^{n-i} t^i, \quad (1)$$

This work is supported in part by SECS Faculty Startup Fund and URC Faculty Research Fellowship at Oakland University.

*Jun Chen is the corresponding author.

where t is the parameter for curve construction, varying from 0 to 1, P_i represents the i^{th} control point, and $b_{i,n}$ denotes the Bernstein polynomial. In this paper, the lane change path modeling is based on a 5th-order Bezier curve defined by 6 control points. Substituting $n = 5$ into (1), a more succinct representation of the 5th-order Bezier curve is derived as follows:

$$B(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5,$$

where a_i are functions of the control points, as follows,

$$\begin{aligned}
a_0 &= P_0 \\
a_1 &= -5P_0 + 5P_1 \\
a_2 &= 10P_0 - 20P_1 + 10P_2 \\
a_3 &= -10P_0 + 30P_1 - 30P_2 + 10P_3 \\
a_4 &= 5P_0 - 20P_1 + 30P_2 - 20P_3 + 5P_4 \\
a_5 &= -P_0 + 5P_1 - 10P_2 + 10P_3 - 5P_4 + P_5
\end{aligned}$$

The curvature κ at any point on a Bezier curve can be calculated using the formula:

$$\kappa(t) = \frac{\dot{B}(t) \times \ddot{B}(t)}{|\dot{B}(t)|^3} \tag{2}$$

where $\dot{B}(t)$ and $\ddot{B}(t)$ are the first and second derivatives of the Bezier curve for t, respectively. For lane change, it is often desirable to start and end the maneuver with zero curvature to ensure a smooth transition. This implies that the curvature at the beginning $t = 0$ and end $t = 1$ of the Bezier curve should be zero. Substituting $t = 0$ into (2), it is evident that the curvature at $t = 0$ is zero only when $\dot{B}(0) \times \ddot{B}(0)$ equal to zero. According to the 5th-order Bezier curve defined earlier, the first and second derivatives of the Bezier curve can be expressed as:

$$\dot{B}(t) = a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4 \tag{3}$$

$$\ddot{B}(t) = 2a_2 + 6a_3 t + 12a_4 t^2 + 20a_5 t^3 \tag{4}$$

Substituting $t = 0$ into (3) and (4) and applying the zero curvature condition, the following is obtained:

$$5(P_1 - P_0) \times 20((P_2 - P_1) - (P_1 - P_0)) = 0 \tag{5}$$

Let v_{01} denote the vector from P_0 to P_1, and v_{12} the vector from P_1 to P_2. Then (5) can rewritten as follows:

$$5v_{01} \times 20(v_{12} - v_{01}) = 0 \tag{6}$$

To satisfy (6), $v_{01} \times v_{12}$ must be equal to zero. Since both v_{01} and v_{12} are non-zero vectors, v_{01} and v_{12} has to be in the same direction. Consequently, this indicates that points P_0, P_1, and P_2 are colinear. This condition ensures that the path begins with a straight line, a necessary requirement for zero curvature at the start of lane changes.

To ensure zero curvature at the end of the lane change, similar conditions must be satisfied at $t = 1$. By substituting $t = 1$ into (3) and (4) and applying the zero curvature condition, we have

$$5(P_4 - P_5) \times 20((P_4 - P_5) - (P_3 - P_4)) = 0. \tag{7}$$

This condition, similar to the one derived for the start of the maneuver, ensures that the lane change path ends with a straight line, which is essential to ensure that the control points P_3, P_4, and P_5 are colinear.

B. Complete Lane Change Model

Based on the definition of the Bezier curve, a Bezier curve can be generated as long as the coordinates of a set of control points are obtained. Therefore, in the lane change model, the state space is defined by the coordinates of these control points. We assume the driver solves an optimal control problem (OCP) to find a set of control points such that the corresponding Bezier curve optimizes a certain cost function. We further assume the starting point of the lane change, i.e., P_0, is known. Moreover, to ensure zero curvature at both the beginning and end of the lane change for the smooth transition, P_1, P_2 have the same y-coordinate as P_0, and P_3 and P_4 have the same y-coordinate as P_5, as discussed in II-A. Consequently, the optimization variables of the OCP can be defined as $X = [x_1, x_2, x_3, x_4, x_5, y_5]$ where the x_i are the x-coordinates of control points P_i, and y_5 is y-coordinate of control point P_5. Moreover, the following optimization problem is formed to determine a Bezier curve,

$$\begin{aligned}
\min_X \; J(X) = {} & W_1 \left(\int_0^1 (\kappa(t))^2 dt \right)^2 + W_2 (x_5 - x_0)^2 \\
& + W_3 y_5^2 + W_4 \left(\frac{(w - y_b)(x_a - x_b)}{y_a - y_b} + x_b - P_{0x} \right)^2
\end{aligned} \tag{8a}$$

$$\text{s.t. } B(t) = \sum_{i=0}^{5} P_i b_{i,5}(t) \tag{8b}$$

$$x_{i-1} < x_i, \quad 1 \le i \le 5 \tag{8c}$$

$$x_{\text{lb}} < x_5 - x_0 < x_{\text{ub}} \tag{8d}$$

$$y_{\text{lb}} < y_5 < y_{\text{ub}}, \tag{8e}$$

where the first term in (8a) represents the total curvature of lane change curve, the second and third terms penalize long and wide lane change curves, respectively, and the fourth term makes the trajectory cross the lane mark as soon as possible. The constraint (8b) is the expression for a 5th-order Bezier curve, (8c) ensures that the x-coordinates of the control points are in ascending order, which is essential for maintaining the Bezier curve in the horizontal direction. The constraint (8d) sets the bounds for the horizontal extent of the lane change, ensuring that it is neither too short nor excessively long. Finally, constraint (8e) limits the y-coordinate of the final control point, P_5, to ensure the end of the lane change is within the second lane, where y_{lb} and y_{ub} are the y-coordinates for the boundaries of the second lane.

Note that in (8), the weights $W = \begin{bmatrix} W_1 & W_2 & W_3 & W_4 \end{bmatrix}$ can be different for different drivers, and are usually unknown as drivers do not express their preference numerically. In the sequel, we will discuss how to use IRL to infer W based on the driver's historical data (termed as expert trajectories).

III. Inverse Reinforcement Learning

A. Maximum Entropy Inverse Reinforcement Learning

In this study, maximum entropy inverse reinforcement learning (*MaxEnt* IRL) [17] is proposed for learning the expert trajectory patterns and generating a model capable of producing similar trajectories in different situations. To enable the model to learn the characteristics of these expert trajectories, it is necessary to extract their features. Each feature is a function that maps the trajectory to a real value. Consequently, the vector of all features, denoted as f, is a function that maps the trajectory to a vector of features. For a single expert trajectory $\tilde{\zeta}_i$, the feature vector is denoted as $f(\tilde{\zeta}_i)$. The expert feature vector for the model to learn can be calculated by averaging all feature vectors from all expert trajectories, i.e., $\tilde{f} = \frac{1}{N}\sum_{i=1}^{N} f(\tilde{\zeta}_i)$ The goal of the *MaxEnt* IRL is to identify parameters that align the feature expectation values with the expert feature values observed,

$$\mathbb{E}_{p(\zeta|W)}[f] = \tilde{f}. \tag{9}$$

For *MaxEnt* IRL, the reward function is assumed to be a linear combination of features and their respective weights, $J = W^T f(\zeta_i)$, where the J is the reward function, W is the weight vector and f is feature vector. Moreover, following the principle of maximum entropy [17], which ensures that distribution is the least biased in describing the data, the probability distribution over all the possible trajectories is presented as follows:

$$P(\zeta_i|W) = \frac{1}{Z(W)} e^{-J} \tag{10}$$

where the partition function $Z(W) = \sum e^{-W^T f(\zeta_i)}$ to make sure this distribution is properly normalized, meaning that the sum of the probabilities of all possible paths equals 1. Given the distribution (10) and expert trajectories $\tilde{\zeta}_i$, the unknown weights W can be found by maximizing the log-likelihood:

$$W = \arg\max \; L(W) = \sum_{i=1}^{N} \log P(\tilde{\zeta}_i|W). \tag{11}$$

Although (11) is usually not solvable analytically, the gradient descent approach can be used, where the gradient of the log-likelihood is given as follows:

$$\nabla L(W) = \mathbb{E}_{p(\zeta|W)}[f] - \tilde{f}. \tag{12}$$

Note when the log-likelihood reaches the maximum, the gradient is zero, indicating that the feature expectation matches the expert feature, satisfying (9). According to the gradient descent approach, the feature weight vector is updated using the following equation:

$$W = W + \eta \frac{\nabla L(W)}{\|\nabla L(W)\|},$$

where the η is the learning rate. Note that the calculation of the gradient in (12) requires the computation of expectation either analytically or by sampling [27]. Instead, we utilize an estimated approach to determining the feature expectation values by calculating the feature values for the trajectory with the highest likelihood, as follows

$$\mathbb{E}_{p(\zeta|W)}[f] \approx f(\arg\max p(\zeta|W)).$$

B. Feature Designing

In this section, features f are real mappings of the characteristics of the lane change trajectory state which are shown capable of capturing the preference of lane change driving behaviors.

Comfort: Comfort is an important aspect of driving behavior. The integration of the square of the curvature over the curve is considered a feature to capture comfort, which reflects the rate of change in the vehicle's direction. A lower value indicates a smoother trajectory, indicative of a more comfortable drive. According to (2), the first feature, which is comfort, is defined as follows:

$$f_c = \int_0^1 (\kappa(t))^2 dt = \int_0^1 (\frac{\dot{B}(t) \times \ddot{B}(t)}{|\dot{B}(t)|^3})^2 dt \tag{13}$$

Traffic Efficiency: The efficient movement of a vehicle during a lane change is decomposed into two main components: longitudinal and lateral. The longitudinal efficiency considers the length of the lane change path and the specific point where the vehicle crosses the lane marking, denoted as f_{T1} and f_{T2}, respectively.

$$f_{T1} = P_{5x} - P_{0x} \tag{14}$$

$$f_{T2} = \frac{(w - y_b)(x_a - x_b)}{y_a - y_b} + x_b - P_{0x} \tag{15}$$

where P_{ix} represents the x-coordinate of the control point P_i, w refers to the width of the lane, and the coordinates (x_b, y_b) and (x_a, y_a) correspond to the nearest points on the curve located before and after the lane mark is crossed.

Lateral efficiency is gauged by the vehicle's lateral position at the end of the lane change maneuver, represented by the y-coordinate of the curve's endpoint. This is defined as the feature f_y, indicating the vehicle's final lateral position in the adjacent lane, as shown below:

$$f_y = y_5 \tag{16}$$

C. Lane Change Behavior Learning Algorithm

In this section, we presented an *MaxEnt* IRL-based algorithm for learning driver lane change behavior from expert trajectories, as summarized in Algorithm 1. The algorithm iteratively updates a weight vector W to align the expected feature values $\mathbb{E}_{p(\zeta|W)}[f]$ with the empirical feature values $\tilde{f}$. This process involves solving the OCP (8) based on current estimated W and adjusting the weights through gradient descent to minimize the difference between the expected and the expert feature values. The learning rate η is decreased during iterations to ensure convergence.

Algorithm 1 *MaxEnt* IRL Driver Behavior Learning Algorithm

Input: Expert trajectory $D = \{\tilde{\zeta}_1, \tilde{\zeta}_2, .., \tilde{\zeta}_N\}$, learning rate η
Output: W

$\tilde{f} = \frac{1}{N}\sum_{i=1}^{N} f(\tilde{\zeta}_i)$;
Initialize $c \leftarrow 0$, $W \leftarrow$ all-ones vector;
while W not converge **do**
  for all $\tilde{\zeta}_i$ in D **do**
    $\zeta_{opt,i} \leftarrow$ solving OCP (8) with aligned P_0 of $\tilde{\zeta}_i$;
    $f_{opt,i} \leftarrow f(\zeta_{opt,i})$;
  end for
  $\mathbb{E}_{p(\zeta|W)}[f] \leftarrow \frac{1}{N}\sum_{i=1}^{N} f_{opt,i}$;
  $\nabla L \leftarrow \mathbb{E}_{p(\zeta|W)}[f] - \tilde{f}$;
  $W \leftarrow W + \eta \frac{\nabla L}{\|\nabla L\|}$;
  $c \leftarrow c + 1$;
  if $c = 200$ **then**
    $\eta \leftarrow \eta/10$;
  end if
end while

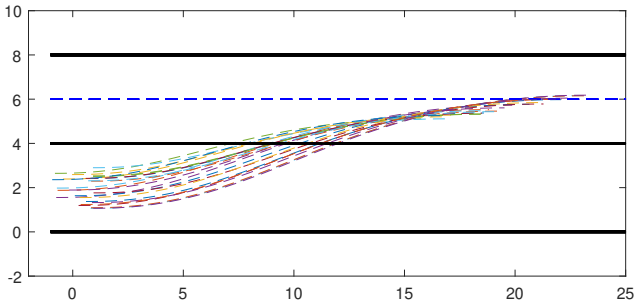

Fig. 1. Expert trajectories (training dataset).

IV. Results

To test the proposed method, the expert trajectories are generated using (8) with a selected W (for data generation only) and with random initial points. The road width is set at 4 meters, which places the first lane's lateral range between 0 to 4 meters and the adjacent lane's from 4 to 8 meters. The total length of the road is 25 meters, which means the lane change maneuvers are complete within this distance. The starting area for all initial points is confined to a specific region in the 2D plane, with x ranging from -1 to 1 and y from 0 to 4. This approach allows the model to understand different lane change behavior under varying initial conditions. To accurately represent the behavior of the same driver, the cost function used in the optimization problem remains consistent across all simulations. This consistency ensures that the variations in the generated expert trajectories are due to differences in initial conditions rather than changes in driving style or preferences. A total of 30 expert trajectories are generated, and the first 25 trajectories serve as the training data, denoted as D, as shown in Fig.1. The remaining 5 trajectories are used for testing.

To improve the numerical stability of the IRL algorithm, the Min-Max normalization is used to adjust the range of expert feature values. The feature ranges of the expert trajectories are shown in Table I, which are used for the aforementioned normalization except for f_y ($[6, 8]$ is used for f_y).

TABLE I
Expert features range.

	f_c	f_{T1}	f_{T2}	f_y
f_{min}	0.0013	15.932	7.033	5.108
f_{max}	0.0015	22.388	11.097	6.181

TABLE II
Learned feature weight

	f_c	f_{T1}	f_{T2}	f_y
W	1.434	1.3017	0.7947	4.4054

Table II listed the learned weights from IRL, and Fig. 2 compares the expert and predicted paths from various starting points. These results demonstrate that, with the learned cost function, the Bezier lane change model accurately generates lane change curves at different starting points. Furthermore, Table III presents the feature differences between the expert and predicted trajectories, with each row representing a distinct starting point. Notably, differences in curvature, total length, and the longitudinal distance a vehicle crosses a lane marking (x-cross) are minimal. This indicates that the predicted path closely mirrors the expert path in these aspects. However, the second case has a relatively large lateral efficiency error, probably because the method is only based on the most likely trajectory during learning when calculating feature expectations.

V. Conclusion

This paper presented an approach for modeling driver lane change behavior using Bezier curves and *MaxEnt* IRL. Bezier curves were utilized to generate smooth lane change paths that ensure zero curvature at the start and end. The *MaxEnt* IRL algorithm was used to learn driver preferences and cost function weights from expert demonstration trajectories. Features were designed to capture key aspects of driver behavior including comfort, longitudinal efficiency, lateral efficiency, and traffic efficiency. The learned cost function weights reflected the relative priorities of these factors. The model efficacy was demonstrated through testing on unused expert trajectories.

TABLE III
Feature difference between expert path and predicted path.

Initial point	Curvature	Length (m)	x-cross (m)	x-end (m)
[0.515,1.997]	-3.02E-06	0.006633	0.022394	-0.06938
[0.486,2.919]	9.43E-05	0.264324	0.124813	-0.40642
[-0.216,1.681]	-5.30E-07	0.38732	-0.06683	0.065326
[0.311,2.171]	1.22E-05	-0.05343	-0.0008	-0.07845
[-0.658,1.448]	7.51E-07	0.534073	-0.10584	0.116749

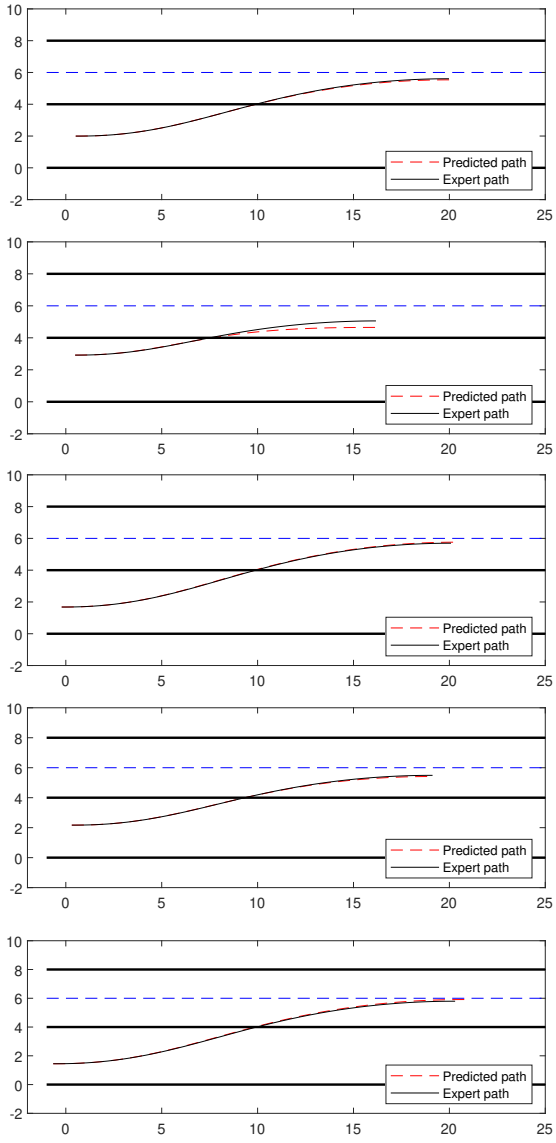

Fig. 2. Predicted trajectory in different initial points for the testing.

The predicted paths closely matched the expert trajectories in terms of curvature, length, and lane change timing. For future research, the incorporation of real driver data and considering more complicated initial conditions will be a significant step forward.

REFERENCES

[1] A. Irshayyid and J. Chen, "Comparative study of cooperative platoon merging control based on reinforcement learning," *Sensors*, vol. 23, no. 2, pp. 1–23, 2023.

[2] J. Chen, M. Liang, and X. Ma, "Probabilistic analysis of electric vehicle energy consumption using MPC speed control and nonlinear battery model," in *2021 IEEE Green Tech. Conf.*, Denver, CO, April 7–9, 2021.

[3] Z. Zhou, C. Rother, and J. Chen, "Event-triggered model predictive control for autonomous vehicle path tracking: Validation using CARLA simulator," *IEEE Trans. Intel. Veh.*, vol. 8, no. 6, pp. 3547–3555, June 2023.

[4] J. Chen and Z. Yi, "Comparison of event-triggered model predictive control for autonomous vehicle path tracking," in *2021 IEEE Conf. Control Tech. and App.*, Austin, TX, USA, 2021, pp. 808–813.

[5] J. Chen, X. Meng, and Z. Li, "Reinforcement learning-based event-triggered model predictive control for autonomous vehicle path following," in *American Control Conference*, Atlanta, GA, June 8–10, 2022.

[6] F. Dang, D. Chen, J. Chen, and Z. Li, "Event-triggered model predictive control with deep reinforcement learning," *IEEE Transactions on Intelligent Vehicles*, accepted for Publication, October 2023.

[7] C. Rother, Z. Zhou, and J. Chen, "Development of a four-wheel steering scale vehicle for research and education on autonomous vehicle motion control," *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 5015–5022, August 2023.

[8] "The multidimensional driving style inventory—scale construct and validation," *Accident Analysis & Prevention*, vol. 36, no. 3, pp. 323–332, 2004.

[9] L. Sun, C. Peng, W. Zhan, and M. Tomizuka, "A fast integrated planning and control framework for autonomous driving via imitation learning," in *Dynamic Systems and Control Conference*, vol. 51913. American Society of Mechanical Engineers, 2018, p. V003T37A012.

[10] Y. S. Quan, J. S. Kim, S.-H. Lee, and C. C. Chung, "Multi-model recurrent neural network control for lane change systems under speed variation," in *Int. Conf. Intelligent Transp. Syst.* IEEE, 2020, pp. 1–6.

[11] R. Chen, I. C. Paschalidis *et al.*, "Distributionally robust learning," *Foundations and Trends® in Optimization*, vol. 4, no. 1-2, pp. 1–243, 2020.

[12] Y. Ma and J. Wang, "Personalized driving behaviors and fuel economy over realistic commute traffic: Modeling, correlation, and prediction," *IEEE Trans. Veh. Tech.*, vol. 71, no. 7, pp. 7084–7094, 2022.

[13] M. F. Ozkan and Y. Ma, "Personalized adaptive cruise control and impacts on mixed traffic," in *2021 American Control Conference (ACC)*. IEEE, 2021, pp. 412–417.

[14] Z. Huang, J. Wu, and C. Lv, "Driving behavior modeling using naturalistic human driving data with inverse reinforcement learning," *IEEE Trans. Intelligent Transp.n Syst.*, vol. 23, no. 8, pp. 10 239–10 251, 2021.

[15] M. Shimosaka, K. Nishi, J. Sato, and H. Kataoka, "Predicting driving behavior using inverse reinforcement learning with multiple reward functions towards environmental diversity," in *2015 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2015, pp. 567–572.

[16] M. Kuderer, S. Gulati, and W. Burgard, "Learning driving styles for autonomous vehicles from demonstration," in *IEEE Int. Conf. Robotics and Auto.* IEEE, 2015, pp. 2641–2646.

[17] B. D. Ziebart, A. L. Maas, J. A. Bagnell, A. K. Dey *et al.*, "Maximum entropy inverse reinforcement learning." in *Aaai*, vol. 8. Chicago, IL, USA, 2008, pp. 1433–1438.

[18] S. Arora and P. Doshi, "A survey of inverse reinforcement learning: Challenges, methods and progress," *Artificial Intelligence*, vol. 297, p. 103500, 2021.

[19] A. Y. Ng, S. Russell *et al.*, "Algorithms for inverse reinforcement learning." in *Icml*, vol. 1, 2000, p. 2.

[20] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[21] Z. Liu, Z. Wang, B. Yang, and K. Nakano, "Learning personalized discretionary lane-change initiation for fully autonomous driving based on reinforcement learning," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2020, pp. 457–463.

[22] J.-w. Choi, R. Curry, and G. Elkaim, "Path planning based on bézier curve for autonomous ground vehicles," in *Advances in Electrical and Electronics Engineering-IAENG Special Edition of the World Congress on Engineering and Computer Science 2008*. IEEE, 2008, pp. 158–166.

[23] Y.-G. Choi, K.-I. Lim, and J.-H. Kim, "Lane change and path planning of autonomous vehicles using gis," in *12th International Conference on Ubiquitous Robots and Ambient Intelligence*. IEEE, 2015, pp. 163–166.

[24] Y. Ding, W. Zhuang, L. Wang, J. Liu, L. Guvenc, and Z. Li, "Safe and optimal lane-change path planning for automated driving," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 235, no. 4, pp. 1070–1083, 2021.

[25] H. Li, Y. Luo, and J. Wu, "Collision-free path planning for intelligent vehicles based on bézier curve," *IEEE Access*, vol. 7, pp. 123 334–123 340, 2019.

[26] R. Lattarulo, L. González, E. Martí, J. Matute, M. Marcano, and J. Pérez, "Urban motion planning framework based on n-bézier curves considering comfort and safety," *Journal of Advanced Transportation*, vol. 2018, 2018.

[27] M. Kuderer, S. Gulati, and W. Burgard, "Learning driving styles for autonomous vehicles from demonstration," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2015, pp. 2641–2646.