Vijayakumar, H., Zhao, D. , Lan, J. , Zhao, W., Tian, D., Li, D., Zhou, Q. and Song, K. (2023) A holistic safe planner for automated driving considering interaction with human drivers. *IEEE Transactions on Intelligent Vehicles*, 9(1), pp. 2061-2076. (doi: 10.1109/TIV.2023.3317338)

Reproduced under a Creative Commons License.
https://creativecommons.org/licenses/by/4.0/

This is the author version of the work. There may be differences between this version and the published version. You are advised to consult the published version if you want to cite from it:
http://doi.org/10.1109/TIV.2023.3317338

https://eprints.gla.ac.uk/307018/

Deposited on 22 September 2023

Enlighten – Research publications by members of the University of Glasgow
https://eprints.gla.ac.uk

A Holistic Safe Planner for Automated Driving Considering Interaction with Human Drivers

Harikirshnan Vijayakumar[1], Dezong Zhao[1], Jianglin Lan[1], Wenjing Zhao[2], Daxin Tian[3],
Dachuan Li[4], Quan Zhou[5] and Kang Song[6]

***Abstract*—This paper advances state-of-the-art automated driving systems with a comprehensive framework that encompasses decision making, maneuver planning, and trajectory tracking considering safety, computational efficiency, and passenger comfort. In face of the co-existence of automated vehicles (AVs) and human-driven vehicles (HDVs), a decision making framework of AVs is proposed for safe lane keeping or changing. The decision making is based on the HDVs' future motion predicted by a learning-based Long Short-Term Memory model. To quantify the uncertainties in prediction, an error ellipse is used to capture the model deviations from the ground truth to ensure driving safety. This paper develops a novel method that leverages lower-order parametric curves to efficiently generate feasible, safe, and comfortable lateral movements for AVs. The planner is complemented by maneuver replanning that can guide the AV back to the original lane when confronted with unexpected blockages from surrounding vehicles. Based on real-world datasets, simulation results show that the proposed method achieves curvature compatibility, shorter trajectory length in lateral maneuvers, accurate trajectory tracking, and effective collision avoidance in lane changing.**

***Index Terms*—Automated vehicles, decision making, maneuver planner and replanner, motion prediction, uncertainties.**

I. INTRODUCTION

STATISTICAL results showed that human errors are one of the major causes of road traffic accidents. A survey reported that around 57% of crashes in lane change scenarios were due to drivers' distractions while driving [1]. Automated vehicles (AVs) are promising solutions to reduce these accidents as their behaviors are rational. In addition, AVs can effectively reduce congestion, enhance passenger comfort and improve energy efficiency [2] [3]. One of the major challenges in automated driving is to generate a smooth, feasible, and safe trajectory. Meanwhile, this generated trajectory must consider road and vehicle constraints, and traffic rules. This work focuses on AVs' forced lane change (LC) in dynamic environments because LC happens when overtaking slow-moving traffic, avoiding obstacles, or the current lane ends.

The first step in AVs' forced LC is to generate a feasible trajectory, for which a few methods have been reported [4]. This work considers the Geometric trajectory planning method because it eases real-time implementation. Geometric forms of trajectory planning include clothoid [5], basic splines [6], polynomial curves [7] [8], and Bézier [9] [10]. Clothoid curves are generated using forward integration, so the computational burden is increased. The trajectory complexity of spline curves is higher than other geometrical methods [11]. The use of the quintic polynomial curve saw great success due to its smooth curvature and efficiency [12] [7]. The primary concern for generating higher-order polynomial curves is their requirements for many input parameters [8], while the lower-order polynomial compromises the curvature continuity.

This work uses the parametric Bézier method because of its advantages in complexity reduction, computational efficiency, and continuity [11] [13]. The efficiency of the lower-order Bézier curve is evident in [14] [10] [15]. Both piecewise Bézier [14] [10] and single Bézier curves [15] have been used in their works to generate trajectories for various driving environments. However, these works often fail to acknowledge the curvature continuity of the generated curves. Moreover, the method in [10] uses an optimization method to generate the curves, posing a challenge for real-time implementation. The fifth or higher-order Bézier curves can generate continuous curvature paths. For example, the fifth-order Bézier curve is used in [16] [17] for LC and obstacle avoidance. The main drawback of [16] [17] is the computational expense in iteratively generating multiple curves to find a collision-free path. An efficient trajectory generation method proposed in [18] uses a piecewise quartic Bézier curve for path smoothing, lane changing, and lane replanning. The curve maintains curvature continuity in the planned trajectory but relies on an offline database containing all potential turning angles and curve lengths, which is a drawback. This work uses a cubic piecewise Bézier curve for trajectory planning because a lower-order Bézier curve can reduce control points, thus lowering the computational burden [19]. Earlier applications of the cubic piecewise Bézier curve saw great success in path-

---

*This work was supported in part by the EPSRC Innovation Fellowship of the Engineering and Physical Sciences Research Council of U.K. under Grant EP/S001956/2, in part by the Royal Society-Newton Advanced Fellowship under Grant NAF\R1\201213, and in part by the State Key Laboratory of Engines at Tianjin University under Grant K2022-13. Jianglin Lan was supported by a Leverhulme Trust Early Career Fellowship under Award ECF-2021-517.

[1] H. Vijayakumar, D. Zhao and J. Lan are with James Watt school of Engineering, University of Glasgow, Glasgow, G12 8QQ, U. K. (e-mail: h.vijayakumar.1@research.gla.ac.uk, dezong.zhao@glasgow.ac.uk, jianglin.lan@glasgow.ac.uk).

[2] W. Zhao is with the Department of Civil Environmental Engineering, Hong Kong Polytechnic University, Hong Kong, China (e-mail: wenjing.zhao@polyu.edu.hk).

[3] D. Tian is with the School of Transportation Science and Engineering, Beihang University, Beijing 100191, China. (e-mail: dtian@buaa.edu.cn).

[4] D. Li is with Institute of Trustworthy Autonomous Systems, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: lidc3@mail.sustech.edu.cn).

[5] Q. Zhou is with the CASE Automotive Centre, University of Birmingham, Birmingham B15 2TT, U.K. (e-mail: q.zhou@bham.ac.uk).

[6] K. Song is with the State Key Laboratory of Engines, Tianjin University, Tianjin 300072, China (e-mail: songkangtju@tju.edu.cn).

smoothing applications, where two cubic curves are joined when there is an angle change in the traveling path [20] [21]. During LC maneuver the traveling path undergoes two angle changes, requiring four curves to generate the LC path. The LC curve used in this work can reduce the curve numbers to two, thereby reduces complexity while maintaining continuity.

Predicting the HDV future trajectory can be divided into physics-based, maneuver-based, and interaction-based models [22]. Physics-based models predict the future trajectory based on dynamic or kinematic models governed by the laws of physics. The kinematic models are simple and easy to implement, while the dynamic model captures the vehicle interaction more precisely with the expense of computational power. Assumptions like constant acceleration, constant velocity, or constant turn rate and velocity are commonly used in kinematic models [23]. Kalman filtering-based method [24] is a typical physics-based model that considers uncertainties or measurement noises in prediction. However, physics-based models are only suitable for short-term prediction of around 1 s [22]. The maneuver-based models predict the trajectory based on the assumption that the vehicle's future states are independent of the surrounding vehicles and can be categorized into trajectory matching and intention prediction methods [22]. The trajectory matching method often employs a cluster-based or probabilistic method trained on an offline dataset for trajectory prediction. Here, the future trajectory is learned by a model conditioned on either the identified cluster they belong to or the trajectory history. For example, a k-means clustering method is used in [25] to categorize the trajectories of the same nature, and a Gaussian model is trained on each cluster to predict the future trajectory. For the intention prediction method, the vehicle's intentions are initially identified using learning models [26], heuristics, or probabilistic models [27]. The future trajectory is then generated using the geometrical trajectory generation method based on the identified intention [27]. The limitation of maneuver-based prediction is the ignorance of vehicle dependencies on surrounding vehicles, leading to wrong predictions [22]. This work aims to predict HDV's future trajectory by capturing the HDV's interaction with its surrounding vehicles. Hence, an interaction-based model is used in this work as it accurately captures the vehicle's interaction with its surrounding vehicles. Markov model variants are among the popular choices for probabilistic interaction aware prediction. Results from Hidden Markov Models [28] [29] and Partially Observable Markov Models [30] [31] showed promising results in trajectory prediction considering HDV interactions. However, high computational demand is a major bottleneck for implementing these methods in real-time. One solution with low computational cost is using a sequential learning model to capture the complex influence of surrounding vehicles on the vehicle considered.

This work uses a sequential learning model called Long Short-Term Memory (LSTM) for trajectory prediction. The LSTM model developed in [32]–[34] shows prominent results in trajectory prediction but fails to integrate with AV decision making. Only a few works like [35] combined the LSTM prediction into the AV decision making framework. However, they did not consider the interaction with surrounding vehicles and possible uncertainties from the LSTM model. It is essential to recognise that prediction from learned models is not 100% accurate. Traditional methods often quantify uncertainties in training and then integrate uncertainties to make predictions [36] [37]. Nonetheless, these approaches often demand complex and laborious training procedures. In contrast, this work aims to simplify the training process while effectively capturing model uncertainties. This approach quantifies uncertainties during the model validation stage, incorporates them during real-time predictions, and provides efficient and reliable predictions.

After obtaining a feasible reference trajectory for the AV navigation, an efficient controller is needed to track the reference. The nonlinear model predictive controller (NMPC) is adopted as it can generate optimal control policy for constrained nonlinear systems. The prediction model, constraints, and cost function can be specified in nonlinear forms, offering better stability and reliability over the linear MPC [38]. NMPC is suitable for the proposed work as it adopts a nonlinear bicycle model traveling on a constrained road.

The rest of the paper is organized as follows. Section II introduces the overall architecture and the vehicle model. Section III details the target vehicle position prediction and uncertainty quantification, Section IV presents the Bézier curve based trajectory planning and velocity generation, Section V explains the decision making process for AV navigation, Section VI details the reference tracking using NMPC and Section VII provides the simulation results. Section VIII draws the conclusions. The main contributions of this paper are:

- A complete architecture is proposed for AV safe navigation, which consists of a predictor to predict human driver trajectory, a trajectory and velocity planner for the AV, a safety assessor to ensure the feasibility of generated trajectory, and a controller to track the final planned trajectory and velocity.
- A novel computationally efficient control point placement is proposed for cubic piecewise Bézier curve suitable for real-time application. A trajectroy replanning using Bézier curve is also proposed to deal with abnormal behaviors of other traffic participants.
- A path modification strategy is proposed with a low computational load, which is critical when the AV approaches the barrier during forced LC.
- An efficient way to contain the uncertainties from the learning model without complicating the model training process is proposed for safe AV navigation.

II. PROBLEM DESCRIPTION AND DESIGN OVERVIEW

This section outlines the driving environment and the safety distances that AV must maintain while performing the LC. Additionally, details of a realistic vehicle model that captures the real-world vehicle's dynamics are introduced. Also, the detailed architecture for safe AV navigation is described.

A. Problem Description

Figure 1 shows the forced LC scenario, where the ego vehicle is controlled to merge between the target and lead

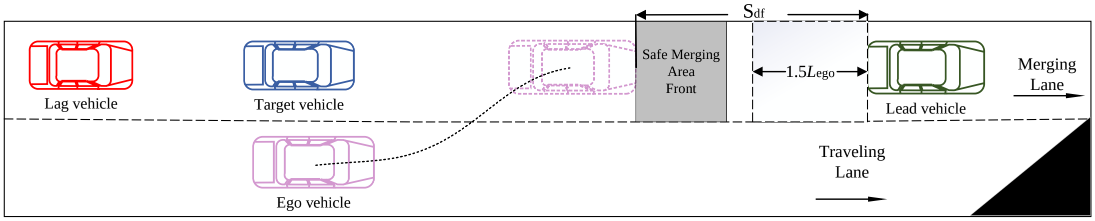


Fig. 1. Dynamic lane change scenario considered in this paper.

vehicles in the merging lane as a barrier in the traveling lane is encountered. The ego vehicle is an AV, while the other vehicles in the scenario are HDVs. The safe distance to be maintained between the AV and the lead vehicle is defined as:

$$S_{\text{df}} = T_h v_{\text{ego}} + 1.5 L_{\text{ego}} \tag{1}$$

where v_{ego} and L_{ego} are the longitudinal velocity and vehicle length, respectively. The term $1.5L_{\text{ego}}$ is the constant safe distance that the ego vehicle must maintain from the lead vehicle, even when the ego vehicle's longitudinal velocity is zero. For a safe normal LC, the value of T_h is chosen as 1.5 s [39]. However, achieving a normal safe space in real-time may not always be possible. Therefore, T_h can be reduced till 0.7 s to allow for a more aggressive LC [39]. Consequently, the width of the safe merging area is obtained by substituting different values of T_h from 1.5 to 0.7 s.

This work employs the front-steered dynamic bicycle model [40] to represent the AV dynamics, as shown in Fig. 2. The model center of gravity is marked as CoG. The model uses a linear tire model based on the small slip angle assumption, where the lateral tire forces will be proportional to the slip angles. The dynamics of the vehicle are governed by

$$\begin{aligned}
\dot{v}_x &= a + \dot{\psi} v_y \\
\dot{v}_y &= \frac{F_{\text{yf}} \cos(\delta_f) + F_{\text{yr}}}{M} - \dot{\psi} v_x \\
\dot{\psi} &= \frac{v_x}{L_f + L_r} \tan(\delta_f) \\
\ddot{\psi} &= \frac{L_f F_{\text{yf}} \cos(\delta_f) - L_r F_{\text{yr}}}{I_z} \\
\dot{x} &= v_x \cos(\psi) - v_y \sin(\psi) \\
\dot{y} &= v_x \sin(\psi) + v_y \cos(\psi)
\end{aligned} \tag{2}$$

where v_x and v_y are the longitudinal and lateral velocities, respectively. ψ represents the yaw angle. M is the vehicle mass. I_z is the moment of inertia along the z axis. L_f and L_r represent the distance from the front and rear wheels to the CoG, respectively. x and y represent the global longitudinal and lateral positions of the vehicle in the inertial coordinates, respectively. a is the vehicle longitudinal acceleration. δ_f is the steering angle. F_{yf} and F_{yr} represent the lateral tire forces in the front and rear wheels that are given by

$$F_{\text{yf}} = -C_f \alpha_f, \ F_{\text{yr}} = -C_r \alpha_r \tag{3}$$

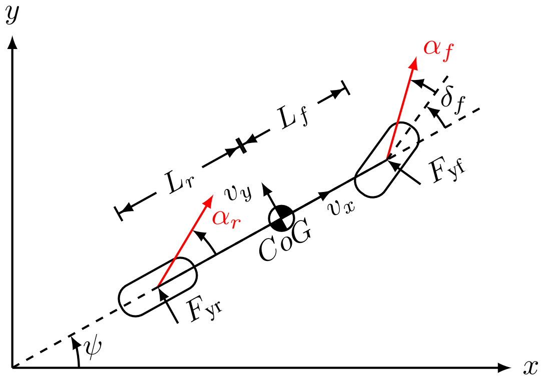


Fig. 2. Dynamic bicycle vehicle model.

where C_f and C_r are the front and rear cornering stiffness. α_f and α_r are the front and rear slip angles defined by

$$\alpha_f = \delta_f - \arctan \frac{v_y + L_f \dot{\psi}}{v_x}, \ \alpha_r = -\arctan \frac{v_y - L_r \dot{\psi}}{v_x} \tag{4}$$

The AV model used in NMPC assumes linear tire forces as mentioned in (3), while the ego vehicle in Fig. 1 takes into account nonlinear tire forces [41]. Thus, the lateral tire forces in (3) are modified to

$$F_{\text{yf}} = -\frac{2\alpha_{f,\text{op}} \alpha_f F_{\text{yf,max}}}{{\alpha_{f,op}}^2 + {\alpha_f}^2}, \ F_{\text{yr}} = -\frac{2\alpha_{\text{r,op}} \alpha_r F_{\text{yr,max}}}{{\alpha_{\text{r,op}}}^2 + {\alpha_r}^2} \tag{5}$$

where $\alpha_{f,\text{op}}$ and $\alpha_{r,\text{op}}$ are the optimal slip angles for the front and rear tires, respectively. The maximum lateral forces in the front ($F_{\text{yf,max}}$) and rear ($F_{\text{yr,max}}$) tires are

$$F_{\text{yf,max}} = \frac{C_f \alpha_{\text{f,op}}}{2}, \ F_{\text{yr,max}} = \frac{C_r \alpha_{\text{r,op}}}{2} \tag{6}$$

B. Overall Architecture for AV Safe Navigation

The overall architecture of the proposed AV safe navigation system design is outlined in Fig. 3. The system is initiated by collecting real-world data, then the data is pre-processed for training and validating a sequential learning model. The learned model will be utilized for predicting the future location of the target vehicle in real-time. To handle uncertainties in the learned model, a bivariate error ellipse defines a boundary that captures prediction errors. The ego vehicle planning module utilizes a Bézier curve to generate an LC trajectory and a velocity profile considering passenger comfort. The planned LC trajectory is then compared with the predicted future

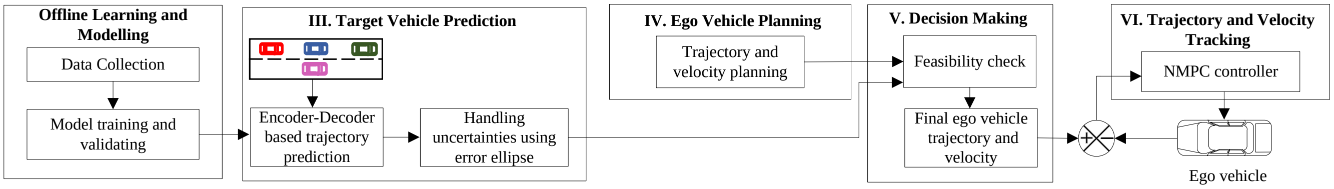

Offline Learning and Modelling
Data Collection
Model training and validating
III. Target Vehicle Prediction
Encoder-Decoder based trajectory prediction
Handling uncertainties using error ellipse
IV. Ego Vehicle Planning
Trajectory and velocity planning
V. Decision Making
Feasibility check
Final ego vehicle trajectory and velocity
VI. Trajectory and Velocity Tracking
NMPC controller
Ego vehicle

Fig. 3. AV decision making framework including the target vehicle position prediction and uncertainties.

trajectory of the target vehicle in a decision making module to estimate the feasibility of LC. If the LC cannot be performed, the AV reverts the decision to lane keeping. After the final path and velocity are obtained from the decision making module, they are tracked by the AV using the NMPC controller.

III. Target Vehicle Prediction

This section introduces the network structure, the training process of the HDV prediction model, and the mitigation to tackle the uncertainties in making reliable predictions.

A. LSTM for Target Vehicle Position Prediction

An Encoder-Decoder LSTM architecture for the target vehicle position prediction is shown in Fig. 4. An LSTM cell in the encode-decoder architecture captures the dependencies between the past and current information using different gating operations [33]. The LSTM mechanism is defined as

$$\begin{aligned}
F_t &= \sigma(W_{XF}X_t + W_{HF}H_{t-1} + b_F) \\
I_t &= \sigma(W_{XI}X_t + W_{HI}H_{t-1} + b_I) \\
O_t &= \sigma(W_{XO}X_t + W_{HO}H_{t-1} + b_O) \qquad (7) \\
C_t &= F_t \odot C_{t-1} + I_t \odot \tanh(W_{XC}X_t + W_{HC}H_{t-1} + b_C) \\
H_t &= O_t \odot \tanh(C_t)
\end{aligned}$$

where the subscripts t and $t-1$ represent the current and past time steps, respectively. σ and tanh are the nonlinear activation functions. F_t, I_t, and O_t represent the forget, input, and output gates. C_t is the cell state. H_t is the hidden state. W_{XF}, W_{HF}, W_{XI}, W_{HI}, W_{XO}, W_{HO}, W_{XC} , and W_{HC} are the weight matrices and b_F, b_I, b_O, and b_C are the bias vectors. $A \odot B$ represents the Hadamard product between quantities A and B. The encoder summarizes the input sequence from X_{h1} to X_{hi} to fixed size context vector represented as the weighted sum of the final hidden state H_{hi} of the encoder LSTM. The context vector is then provided as an input to the decoder at each time step to generate the output sequence. Also, the decoder's cell and hidden states are initiated with the encoder's last cell state (C_{hi}) and hidden state (H_{hi}).

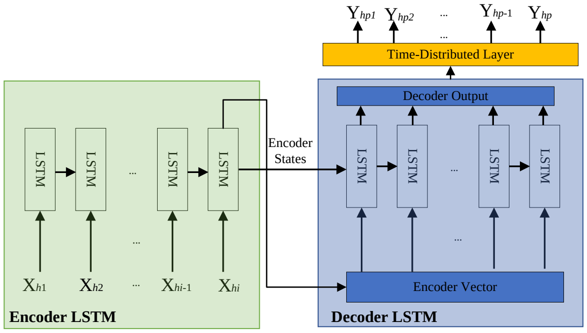


Fig. 4. Encoder-Decoder LSTM for position prediction.

To show the influence of surrounding vehicles on the target vehicle, the input for each time step will be the surrounding vehicles' states relative to the target vehicle and the states of the target vehicle. In Fig. 4, the interaction period varies from h_1 to h_i, and the prediction horizon varies from h_{p1} to h_p. The time step between the inputs of the interaction horizon and the outputs of the prediction horizon is 0.1 s. The encoder inputs are: $X_{\text{input}} = [X_{h1}, X_{h2}, \ldots, X_{hi}]$ where $X_n = [X^{\text{target,n}}, X^{\text{target,n}} - X^{\text{ego,n}}, X^{\text{target,n}} - X^{\text{lead,n}}, X^{\text{target,n}} - X^{\text{lag,n}}]$, with $n \in [h_1, h_i]$. Input states $X^{\text{vehicle,n}}$ where vehicle $\in$ [target, ego, lead, lag] are the vehicles' position and velocity in the $x-y$ direction and the yaw angle ψ. Therefore, there will be 20 inputs at each time step. The time step between each input is 0.1 s. The outputs from the decoder are: $Y_{\text{output}} = [Y_{hp1}, Y_{hp2}, \ldots, Y_{hp}]$ where $Y_m = (xt_m, yt_m)$ with $m \in [h_{p1}, h_p]$ is the position of target vehicle in the $x-y$ coordinates. All the input and output data are normalized before training the network due to considerable differences in the scale of each state using

$$Data_{\text{norm}} = \frac{Data - Data_{\text{min}}}{Data_{\text{max}} - Data_{\text{min}}} \qquad (8)$$

where $Data_{\text{max}}$ and $Data_{\text{min}}$ are the maximum and minimum values of $Data$, respectively. In scenarios where the lead vehicle is absent, +1 is assigned to the $x-y$ positions, while other input states are set to zero, indicating the minimal influence of the lead vehicle on the target vehicle. Similarly, -1 is assigned when the lag vehicle is absent. The interaction period h_i was substituted with 1 s, 2 s, and 3 s, respectively, to choose the best result.

Table I shows the range of network parameters used to optimize the network. The number of neurons in the network refers to the LSTM memory cells stacked at each time step to capture sequential dependencies. The batch size divides the training dataset into a fixed-length input fed to the network to update its weights and biases. Each batch will be fed to the network to make the prediction, and then these predictions will be compared to the ground truth to update the weights and biases. This work uses an Adam optimizer with a constant

TABLE I
RANGE OF NETWORK PARAMETERS CONSIDERED

Parameters	**Values**
Number of neurons	32, 64, 128
Batch size	5:10:145
Epoch	10:20:700

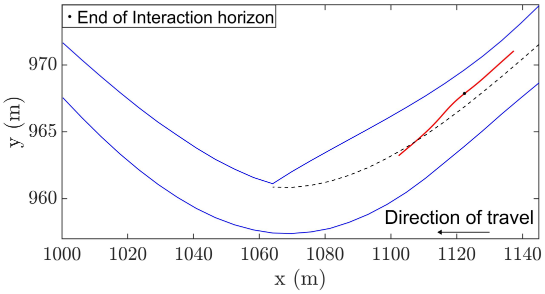


Fig. 5. Scenario from INTERACTION dataset used in the study.

learning rate of 0.001 in updating these parameters. Epoch refers to the number of times the neural network has processed the entire dataset to update the network parameters.

The model has been trained, tested, and validated by investigating a forced LC scenario in China from the INTERACTION dataset [42]. The target vehicle's interaction period with the surrounding vehicles and the output prediction period have been heuristically separated to train the prediction model, similar to [34]. The interaction horizon ends when the ego vehicle driven by a human shows a lateral deviation for LC or attempts the LC. Figure 5 shows an example test case, illustrating the heuristic selection of the interaction horizon endpoint. A total of 895 interaction is collected from the dataset to train and test the model. Out of which, 75% is used for training and the rest for testing. The dataset contains around 80% successful and 20% unsuccessful lane changes.

B. Handling Uncertainties using Error Ellipse

Uncertainties in the LSTM prediction must be critically considered. A bivariate variance-covariance error vector is used to construct an error ellipse that captures the model errors in both x and y directions. After that, a linear transformation is performed to obtain the eigenvalues and eigenvectors to determine the orientation of the error ellipse, as the $x - y$ positions are highly correlated. The transformed matrix is to find the new ellipse equation and the angle at which the ellipse is aligned. The square root of the largest and smallest eigenvalues are the length of the ellipse semi-major axis and semi-minor axis, respectively. The ellipse equation after linear transformation is

$$\text{EIG}\begin{pmatrix} {\sigma^2}_x & \sigma_{xy} \\ \sigma_{xy} & {\sigma^2}_y \end{pmatrix} \Rightarrow \frac{x^2}{\lambda_1} + \frac{y^2}{\lambda_2} = s \tag{9}$$

where EIG refers to eigenvalue transformation, ${\sigma_x}^2$ and ${\sigma_y}^2$ are the variance along x and y axis, respectively; σ_{xy} is the covariance between $x - y$ coordinates; λ_1 and λ_2 are the largest and smallest eigenvalues, respectively; s is the chi-square value. This work uses a chi-square value of 5.991, corresponding to the 95% confidence factor [43]. The ellipse orientation angle is

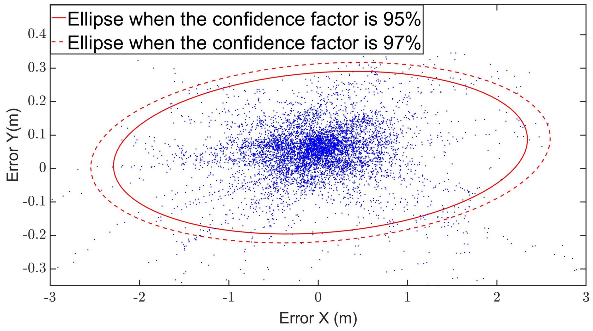


Fig. 6. Bivariate Error Ellipse with 95% and 97% confidence factor.

$$\epsilon_{\text{ellipse}} = \text{atan}\left(\frac{\nu_1(1)}{\nu_1(2)}\right) \tag{10}$$

where $\nu_1(1)$ and $\nu_1(2)$ are the first and second components of the largest eigenvector, respectively. An error ellipse after linear transformation on an example dataset with different confidence factors is shown in Fig. 6. The ellipse size increases and covers more error data points as the confidence factor value increases. As the expanded ellipse would occupy more navigation space, the confidence factor value is set to 95%.

IV. EGO VEHICLE PLANNING

This work focuses on local trajectory and velocity planning for an LC scenario with the following assumptions: precise waypoints representing the midpoints of both merging and traveling lanes are obtained before starting the journey. The scenario also contains a smooth trajectory connecting these waypoints called the "global path". The vehicle tracks the global path before and after a maneuver is performed. The LC trajectory always starts from the global path in the merging lane. Initial acceleration at the start of LC is zero.

A. Lane Change Trajectory Planning

To reduce the complexity of higher-order parametric curves without compromising the curvature continuity, a cubic Bézier curve is used for trajectory planning with the general form of

$$P(u) = \sum_{i=0}^{3} \binom{n}{i} u^i (1-u)^{3-i} P_i \quad (0 \leq u \leq 1) \tag{11}$$

where P_i represents the control points in the Cartesian coordinates. The curve starts at $u = 0$ and ends at $u = 1$. The curvature of the curve in Cartesian coordinates is

$$K(u) = \frac{\dot{x}(u)\ddot{y}(u) - \dot{y}(u)\ddot{x}(u)}{\left(\dot{x}(u)^2 - \dot{y}(u)^2\right)^{3/2}} \tag{12}$$

where $\dot{x}$ and $\ddot{x}$ are the first and second derivates of the curve in the x direction, respectively, and $\dot{y}$ and $\ddot{y}$ are the first and second derivates of the curve in the y direction, respectively. A piecewise curve is used, where P_0 to P_3 refers to the control points of the first Bézier curve B1, and P_4 to P_7 the control points of refers to the second Bézier curve B2.

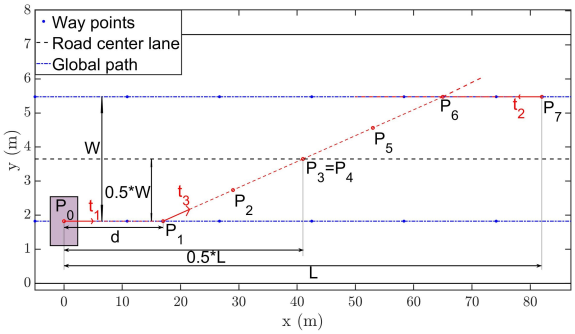


Fig. 7. Waypoint based trajectory planning using Bézier curve.

For a smooth curve transition from B1 to B2, the curves must obey the C0, C1, and C2 continuity criteria. Hence, the control points, the first and second derivatives at the end of B1 and the beginning of B2 should be equal, as described below.

$$\begin{aligned} \text{C0 continuity,} \quad & B1(1) = B2(0) \\ & P_3 = P_4 \\ \text{C1 continuity,} \quad & \dot{B}1(1) = \dot{B}2(0) \\ & P_3 - P_2 = P_5 - P_4 \\ \text{C2 continuity,} \quad & \ddot{B}1(1) = \ddot{B}2(0) \\ & P_3 - 2P_2 + P_1 = P_6 - 2P_5 + P_4 \end{aligned} \tag{13}$$

This work proposes a waypoint-based planning suitable for both straight and curved paths. The procedures and requirements to generate a LC curve are:

- The first control point P_0 will be at the center of gravity of the ego vehicle.
- The second control point P_1 will be on the tangent from the first control point P_0 and the immediate waypoint after P_0 in the direction travel, displaced at distance d from P_0.
- P_3 will be half the length of the total curve in the x direction and half the height of points P_1 and P_7 in the y direction. Therefore, P_3 will be at the center of the road when the lanes have equal dimensions and are parallel. Also, P_3 will be the starting point of B2, taken as P_4.
- P_6 lies on the intersection from extending the tangent from P_1 to P_4 and the tangent in the direction of P_7 to the waypoint before P_7 in the Merging Lane. Hence, P_6 will be the same distance d for symmetric B1 and B2, occuring when lanes are in parallel with the same width.
- The control point P_2 will be in the direction of tangent from P_1 to P_3 and at a half distance between P_1 and P_3. Similarly, P_5 will be at the center of P_4 and P_6.

Figure 7 shows the trajectory generation steps using waypoints, detailed in Algorithm 1. The parameter d is estimated by iteration that aims to minimizes the lateral acceleration as in Algorithm 1. The total linear length L, depends on the lead vehicle position after t seconds. W is the linear width between control points P_0 and P_7. If the lead vehicle is absent, L is the length that bounds the maximum lateral acceleration ($|a_{y,\max}|$) considered to optimize passenger comfort .

Algorithm 1: Bézier curve generation

```
Input: $P_{\text{lead,t}}$, $S_{\text{df}}$, $\Delta$, $L_0$
CASE 1 - Length known
$L$ = $P_0$ + ($P_{\text{lead},t}$ - $S_{\text{df}}$)
$d$ = $P_0$:increment:$L/2$
for $i$=1:length $(d)$ do
    $P_1$ = $P_0$ + $d(i)t_1$
    $P_3$ = $0.5[L, W]$
    $P_2$= $0.5D_1t_3$ from $P_1$
    $P_4$= $P_3$
    $P_6$ = intersection point after extending $t_2$ and $t_3$
    $P_5$ = $0.5D_2t_3$ from $P_4$
    for $u$=0:1 do
        Generate the 1$^{\text{st}}$ and 2$^{\text{nd}}$ Bézier curves using
          (11)
    end
    Compute the curvature $K$ using (12)
    $K_{\max}$ = max(abs($K$))
    $a_y$(i) = $v_x^2K_{\max}$
end
[$a_y$, idx] =min($a_y$)
$d$ = $d$(idx)
CASE 2 - Length to be estimated
Input: $L_0$
while $|a_y| < |a_{y,\max}|$ do
    CASE - 1
    $L$= $L$+ $\Delta$
end
$L$ = $L$− $\Delta$
$d$ = $d$(idx)
```

Algorithm 1 shows two cases of curve generation: L is known and needs to be determined. a_y is the lateral acceleration, Δ is a small increment value, and $P_{\text{lead},t}$ refers to the final lead vehicle position at the end of AV LC duration. t_1 refers to the unit tangent vector from the start point P_0 to the immediate next waypoint in the traveling lane, t_2 refers to the unit tangent vector from the endpoint P_7 to the previous waypoint in the merging lane, and t_3 refers to the unit tangent vector from P_1 to P_3. D_1 refers to the distance between P_1 and P_3, and D_2 refers to the distance between P_4 and P_6. For case 2, the initial guess for the Bézier curve length is taken as

$$L_0 = 3v_{\text{ego}} \tag{14}$$

where L_0 is the initial curve length. Note that control points P_1, P_2, P_3, P_5 and P_6 lie on the same tangent, thus C0 to C2 continuity are satisfied as mentioned in (13). Curves B1 and B2 will be symmetric if the lead vehicle is absent.

If an obstacle is detected in the traveling lane, P_3 of the B1 curve can be placed to leave a safe distance between the obstacle and the AV. According to the Bézier curve property, the control points will have precise control over the curve and can be adjusted to avoid this obstacle. Finally, B2 curve can be generated depending on the final position of the lead vehicle as described in Algorithm 1, or can be symmetric to the B1 curve if the merging lane is empty.

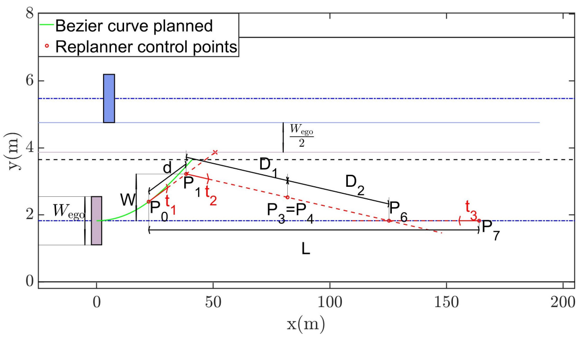


Fig. 8. Waypoint based trajectory replanning using Bézier curve.

B. Trajectory Replanning

A trajectroy replanning method is proposed to modify the LC decision if required. Trajectroy replanning will consider uncertainties from the target vehicle, including delays from the target vehicle prediction system or unexpected aggressive acceleration to block the ego vehicle planned LC. Algorithm 1 is modified with the following steps for trajectory replanning:

- The second control point will be tangent to the vehicle's current heading direction at a distance d. Distance d is obtained via iteration that gives a minimum of maximum lateral accelerations obtained with different d values.
- In the presence of a lead vehicle in the traveling lane, the Bézier curve length L depends on the final position of the lead vehicle after t seconds. If the lead vehicle is absent, the algorithm finds a length that contains $|a_{y,\text{max}}|$ to the considered maximum limit. The other control points remain similar to the LC trajectory.

Figure 8 shows the steps involved in replanning, where the green curve refers to the initial B1 planned for the LC, P_0 refers to the start point of trajectroy replanning where the AV decides to abandon LC, and t_1 refers to the unit tangent vector from P_0 to the current heading direction. Assuming the target vehicle travels along the center lane, the maximum limit of d, marked as a red cross, is obtained by subtracting the target and ego vehicles' half-width from merging lane center lane to ensure safety between the vehicles. It is not always valid to assume that the target vehicle follows the center of the lane. To this extent, the maximum limit of d can be modified to the upper border of the traveling lane, ensuring the trajectroy replanning is contained within the traveling lane.

C. Global Path Modification

The AV must consider the scenarios when the LC maneuver cannot be performed before the barrier. In this case, the AV must consider the approaching barrier, stop at a safe distance from the barrier and be ready to merge at the next LC opportunity. This is done differently from the works of [44]–[46], where the controller solves an optimization problem to ensure a safe distance from the barrier in addition to the trajectory tracking. The modified global path aims to ease the computational load of the controller while obtaining a trajectory that avoids collision into a barrier. Meanwhile, the workload on the control system is reduced to track the reference trajectory and velocity for lane keeping. This work uses a cubic Bézier curve at the end of the traveling lane to leave a safe distance from the barrier. Figure 9 shows the modified global path for two different barrier shapes in the traveling lane. The dashed-dotted line passing through the center of the ego vehicle shows the initial global path, and the red trajectory highlighted shows the modification added.

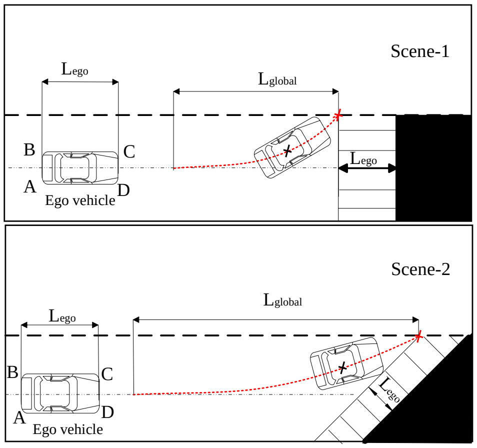


Fig. 9. Global path modified while encountering a barrier.

$P_{3,\text{global}}$, marked as a red cross in Fig. 9, is the last control point of the modified global path. Initial estimation for $P_{3,\text{global}}$ to avoid collision is chosen as the difference between barrier edge and L_{ego}. However, if the initial estimation results in collision, $P_{3,\text{global}}$ is moved backward. The process is repeated until the AV obtains a collision-free trajectory that leaves a safe distance of S_t from the barrier. Algorithm 2 shows the pseudo-code to generate a modified global path B_{global}. Collision check is carried out by ensuring the AV edge D shown in Fig. 9 leaves a safe distance (L_{ego}) from the barrier. Point D is selected for the collision check as this will be the first point to contact the barrier. The collision check process uses a translation and rotation matrix described in line 9 of Algorithm 2 to ease the computation.

In Algorithm 2, points D^r, D, O_{orgin}, and $\text{Seg}_{x,y} \in \mathbb{R}^2$ are in Cartesian coordinates. D^r refers to the final point after translation and rotation of the AV edge D. O_{orgin} is the origin considered for the rotation. The points A, B, and C of the AV marked in Fig. 9 will be transformed similarly to point D for projecting the AV shape to B_{global}. The length of the curve (L_{global}) is 2.5 times the AV length (L_{ego}), as the AV will be cruising to a stop. Once the AV enters the modified global path, the ego vehicle follows the trajectory till the end of the curve. Therefore, only the B2 curve must be generated, and the length of the B2 depends on the lead vehicle position. During LC, if P_3 of B1 obtained via Algorithm 1 moves beyond $P_{3,\text{global}}$, P_3 will be taken as $P_{3,\text{global}}$ for safety.

This work also finds a point in the modified global path marked as a black cross in Fig. 9, where the AV comes to rest and is ready to change the lane. Before LC, it is necessary to ensure that the AV stays in the traveling lane. This requirement

Algorithm 2: Modified global path

```
1  Input: Collision = 1, $L_{\text{ego}}$, $L_{\text{global}}$, $O_{\text{origin}}$, $P_{3,\text{global}}$, $\Delta$,
       $S_t$ = 0.5 $L_{\text{global}}$, and AV corners ($A, B, C$, and $D$)
2  while Collision= 1 do
3      $P_{0,\text{global}}$ = $P_{3,\text{global}}$ - $L_{\text{global}}$
4      $P_{1,\text{global}}$, $P_{2,\text{global}}$ = generated similar to $P_1$ and $P_2$
5      $B_{\text{global}}$ = generate the curve similar to Algorithm 1
6      Seg$_{x,y}$=linspace($B_{\text{global}}$,15)
7      $\theta_s$ = atan(Seg$_x$/Seg$_y$)
8      for $i$=1:length (Seg$_{x,y}$) do
9          $D^r = \begin{bmatrix} \cos(\theta_s(i)) & -\sin(\theta_s(i)) \\ \sin(\theta_s(i)) & \cos(\theta_s(i)) \end{bmatrix} * (O_{\text{origin}}\text{-}D)$
             + $O_{\text{origin}}$ + (Seg$_{x,y}(i) - O_{\text{origin}}$)
10         if $D^r > S_t$ then
11             Collision = 1
12             $P_{3,\text{global}} = P_{3,\text{global}} - \Delta$
13         else
14             Collision = 0
15         end
16     end
17 end
```

can be satisfied when the AV edge C does not cross the boundary of the traveling lane. Restricting the AV within the traveling lane can be done by modifying lines 10 to 15 in Algorithm 2. The AV will be projected and cross-checked to each point in B_{global}, to determine if C crosses the traveling lane boundary. The black cross in Fig. 9 is taken as the point before the edge C intersects with the traveling lane boundary.

D. Velocity Planning

1) Lane changing: Velocity planning for the planned trajectory depends on velocity limits at each point, which depends on maximum allowable lateral acceleration and curvature. The relation between these quantities can be expressed as

$$v_{\text{limit}} = \sqrt{\frac{|a_{y,\max}|}{K}} \tag{15}$$

where v_{limit} is the velocity limit along the planned trajectory and K is the curvature at the point. A trapezoidal acceleration profile [47] is used in this work, considering the maximum acceleration possible and the first derivative of acceleration (jerk). A general case scenario as in Fig. 10 is considered, where the profile includes 7 phases. Phases 1 to 3 are the acceleration zone, phase 4 is the constant acceleration zone, and phases 4 to 7 are the deceleration zone. The acceleration, velocity, and traveled distance at time step t are

$$\begin{aligned} a_t &= a_{t-1} + J_t \Delta t \\ v_t &= v_{t-1} + a_{t-1}\Delta t + \tfrac{1}{2} J_t \Delta t^2 \\ s_t &= s_{t-1} + v_{t-1}\Delta t + \tfrac{1}{2} a_{t-1}\Delta t^2 + \tfrac{1}{6} J_t \Delta t^3 \end{aligned} \tag{16}$$

where $\Delta t = t_i - t_{i-1}$ and the time intervals for phases 1 to 3 are calculated using

$$\Delta t_1 = \Delta t_3 = \tfrac{a_{\max}}{J_{\max}},\ \Delta t_2 = \tfrac{v_f - v_0}{a_{\max}} - \tfrac{a_{\max}}{J_{\max}} \tag{17}$$

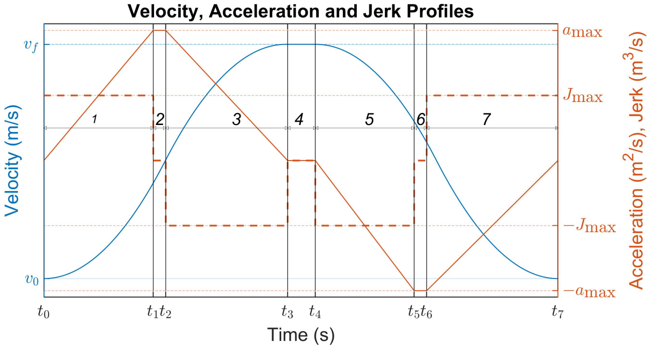


Fig. 10. Velocity profile for lane change.

the velocities at each interval are

$$\begin{aligned} v_1 &= v_0 + \tfrac{1}{2} J_{\max} \left(\tfrac{a_{\max}}{J_{\max}}\right)^2 \\ v_2 &= v_f \\ v_3 &= v_f - \tfrac{a_{\max}}{2} \tfrac{a_{\max}}{J_{\max}} \end{aligned} \tag{18}$$

and the total traveling distance for phases 1 to 3 is given by

$$s_{1,2,3} = (v_f + v_0) \frac{a_{\max}}{J_{\max}} + \frac{\left(v_f - \frac{a^2{}_{\max}}{2J_{\max}}\right)^2 - \left(v_0 + \frac{a^2{}_{\max}}{2J_{\max}}\right)^2}{2a_{\max}} \tag{19}$$

where v_0 and v_f refer to the initial and final velocities, respectively, $a_{\max}$ is the maximum longitudinal acceleration, and $j_{\max}$ is the maximum allowable jerk.

The jerk value for phase 1 is $+J_{\max}$, phase 2 is 0, and phase 3 is $-J_{\max}$. Phase 2 may be neglected if the traveling distance is short. If the distance required to reach v_f from v_0 is greater than the planned LC length, the AV accelerates after completing the LC and performs lane keeping in the merging lane until the desired velocity is achieved. Detailed explanations and special cases of velocity profiles are given in [47]. Phases 5 to 7 are the reverse case of phases 1 to 3, where the velocity profile is calculated by swapping the jerk values in phases 1 and 3. The initial velocity will be $v_{\max}$, and the final velocity be v_0. Phase 4 depends on the total curvilinear length of the Bézier (s_{total}), i.e., if $s_{\text{total}} > s_{1,2,3}$ + $s_{5,6,7}$, then Δt_4 is defined by (20).

$$\Delta t_4 = \frac{s_{\text{total}} - s_{1,2,3} + s_{5,6,7}}{v_{\max}} \tag{20}$$

2) Lane keeping: While performing lane keeping in a forced LC scenario, the velocity planner must ensure that AV decelerates from the traveling velocity to zero as it reaches the resting point mentioned in Section IV-C. To this end, the ego vehicle calculates the distance required to decelerate from the current velocity to zero using (19) considering the maximum deceleration rate and comfort. The velocity profile for lane keeping will be the deceleration phases from 5 to 7 in Fig. 10.

V. DECISION MAKING

The data collected for the study contained LC scenarios with final target velocity reaching up to 11 $\mathrm{m/s}$. Approximately

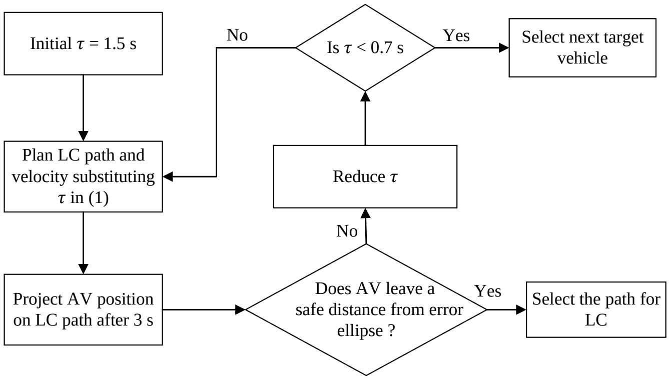

Initial τ = 1.5 s
No
Is τ < 0.7 s
Yes
Select next target vehicle
Plan LC path and velocity substituting τ in (1)
Reduce τ
No
Project AV position on LC path after 3 s
Does AV leave a safe distance from error ellipse ?
Yes
Select the path for LC

Fig. 11. Flow diagram for feasibility check.

79% of LC velocity occurs between 2 to 6 $\mathrm{m/s}$. Hence, the average LC duration is 4.5 s. The target vehicle position prediction is limited to 3 s as a longer horizon leads to a larger position error. Moreover, a larger position error increases the error ellipse size, so more navigation space is excluded. While limiting the target vehicle prediction horizon to 3 s, AV will cover around 66% of its total LC trajectory, leading to a reliable decision. It is preferred to leave a safe gap between the target and ego vehicles, similar to (1), throughout the LC process. However, while considering this safe gap, AV LC's success rate is drastically reduced compared to human drivers. Hence for this work, during the prediction horizon of 3 s, the minimum safety threshold to be kept between the target vehicle and the AV is taken as the length of the target vehicle. Thus, a safety threshold equivalent to the length of the target vehicle is added with the outputs from the encoder-decoder LSTM to guarantee safe navigation. This padded output will be added to the ground truth and used to construct the error ellipse to avoid further calculations.

Figure 11 shows the flow chart for the feasibility check. The initial value of τ with 1.5 s will be substituted in (1) to find the last control point of LC trajectory P_7. Once the LC trajectory is generated using Algorithm 1, the corresponding velocity is obtained, as mentioned in Section IV-D . Once the trajectory and velocity are planned, the AV position after 3 s is projected using rotation and translation matrix as described in Algorithm 2, into the LC trajectory to check for collision with the error ellipse. Since the error ellipse does not consider the width of the target vehicle while checking for collision, the target vehicle width is added with the ego vehicle width for a more comprehensive collision assessment. If there is no collision with the error ellipse, the trajectory is selected as the final LC trajectory and velocity. If there is a collision, the value of τ is reduced, and the process collision check is repeated. The process continues till τ reaches 0.7 s. If collisions happen to all the LC trajectories the LC decision is abandoned, and the AV continues to drive in the traveling lane and selects the lag vehicle as the next target vehicle.

VI. TRAJECTORY AND VELOCITY TRACKING

An NMPC is used for trajectory and velocity tracking. The following discretized prediction model derived from (2), using the Euler method with sample time T_s, is used to obtain the control sequence for the NMPC.

$$\begin{aligned}
v_x(k+1) &= v_x(k) + \Big(a + \dot{\psi}(k) v_y(k)\Big) T_s \\
v_y(k+1) &= v_y(k) + \bigg(\frac{F_{\mathrm{yf}} \cos(\delta_f) + F_{\mathrm{yr}}}{M} \\
&\qquad - \dot{\psi}(k) v_x(k)\bigg) T_s \\
\psi(k+1) &= \psi(k) + \left(\frac{v_x(k)}{L_f + L_r} \tan(\delta_f)\right) T_s \\
\dot{\psi}(k+1) &= \dot{\psi}(k) + \left(\frac{L_f F_{\mathrm{yf}} \cos(\delta_f) - L_r F_{\mathrm{yr}}}{Iz}\right) T_s \\
x(k+1) &= x(k) + \left(v_x(k) \cos(\psi(k)) - v_y(k) \sin(\psi(k))\right) T_s \\
y(k+1) &= y(k) + \left(v_x(k) \sin(\psi(k)) + v_y(k) \cos(\psi(k))\right) T_s
\end{aligned} \tag{21}$$

The outputs of the controller are δ_f and a. The planning module gives the reference lateral position, longitudinal position and velocity. The controller has to bring the error between state and reference to zero, i.e.

$$X_e = X_s - X_r = 0 \tag{22}$$

where X_s represents the states of the prediction model (lateral position, longitudinal position, and velocity) and X_r refers to the corresponding reference state from the planning module.

The cost function to be minimized for the NMPC problem can be formulated as

$$\begin{aligned}
J(k) = \sum_{i=1}^{np} & X_e(k+i)^{\mathrm{T}} Q X_e(k+i) + \\
& \sum_{i=0}^{nc} \Delta U(k+i)^{\mathrm{T}} R \Delta U(k+i)
\end{aligned} \tag{23}$$

The NMPC constraints are

$$\begin{aligned}
X_s(K+i+1) &= F_{\mathrm{dis}}(X_s(K+i), U(K+i)) \\
X_s(K) &= X_{s0} \\
-a_{\max} &\leq a \leq a_{\max} \\
-\Delta a_{\max} &\leq \Delta a \leq \Delta a_{\max} \\
\delta_{f\,\min} &\leq \delta_f \leq \delta_{f\,\max} \\
\Delta\delta_{f\,\min} &\leq \Delta\delta_f \leq \Delta\delta_{f\,\max}
\end{aligned} \tag{24}$$

where the prediction and control horizons are denoted by nc and np, respectively. $X_s(K+i+1)$ defines the current states as a discretized function (F_{dis}) of previous states ($X_s(K+i)$) and inputs ($U(K+i)$) as in (21). X_{s0} represents the system's initial states. X_e is the tracking error as in (22). U $\in \mathbb{R}^{m\times 1}$, where m represents the number of control inputs. $Q \in \mathbb{R}^{n\times n}$ is the weight matrix for state error, and $R \in \mathbb{R}^{m\times m}$ is the weight matrix to penalize the rate of change of control inputs. $-a_{\max}$ and $a_{\max}$ represent the maximum deceleration and maximum acceleration, respectively. $\delta_{f\,\min}$ and $\delta_{f\,\max}$ represent the minimum and maximum steering angle, respectively.

VII. SIMULATION RESULTS

Simulations have been carried out using MATLAB on a laptop with Intel Core i7-10850H CPU @ 2.70 GHz and 16 GB

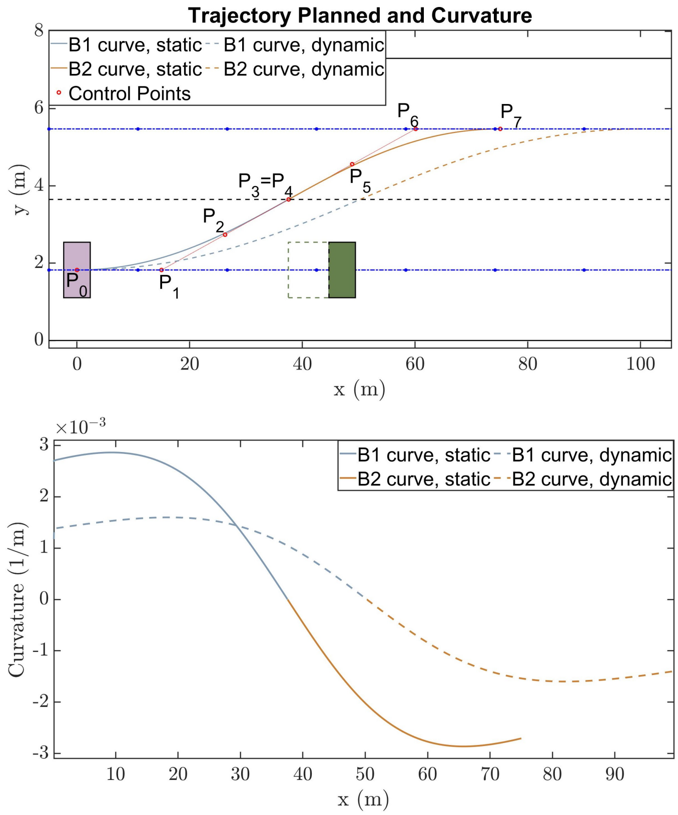


Fig. 12. Bézier curve and curvature considering a static and dynamic obstacle.

RAM. The vehicle parameters are as follows: $M = 1573$ kg, $L_{\text{ego}} = 4.419$ m, $H = 1.44$ m, $L_f = 1.1$ m, $L_r = 1.58$ m, C_f = C_r = 50000 N/rad, I_z = 2873 kg $\cdot$ m^2, $\alpha_{\text{f,op}}$ = 20°, and $\alpha_{\text{r,op}}$ = 11°. The comfort parameters are jerk, lateral acceleration, and rate of change in the steering angle. Their values are: $|J_{\max}| = 1$ m/s^3, $|a_{y,\max}| = 1$ m/s^2, and $|\Delta\delta_{f\,\min}| = 2$ °/s, respectively. The parameters in the NMPC are: T_s = 0.1 s, $Q = \text{diag}(1, 1, 10)$, $R = \text{diag}(10, 10)$, np = 5, nc = 3, $\delta_{f\,\max}$ = +30°, $\delta_{f\,\min}$ = −30°, $\Delta\delta_{f\,\min}$ = −2°/s, $\Delta\delta_{f\,\max}$ = +2°/s, $a_{\max}$ = +2.5 m/s^2, $\Delta a_{\max}$ = +1 m/s^3, and $J_{\max}$ = $\Delta a_{\max}$. The NMPC optimization problem is solved using the MATLAB nonlinear optimization function $fmincon$.

The simulation results reported in Section VII-A show the proposed trajectory planner's details and advantages compared to similar methods. Section VII-B covers the details of the velocity planner efficiency and altering the velocity profile according to the scenarios considered. Sections VII-C, VII-D, and VII-E cover the results obtained from the NMPC controller, the trajectroy replanning, and the modified global path, respectively. Section VII-F details the results obtained from the target vehicle prediction and AV planner integration using the INTERACTION dataset.

A. *Performance of Trajectory Planning*

Figure 12 shows two planned LC trajectory for the AV and its corresponding curvatures at a constant velocity of 20 m/s. The first trajectory represented with the solid lane, the vehicle must quickly steer to avoid the static vehicle. Here, the efficiency of the proposed method on precise curve control can be observed. The control point P_3 is placed at a distance of $1.5L_{\text{ego}}$ from the rear of the static lead vehicle and the curve is contained within the control points. The dotted lines in Fig. 12 show the second case where the lead vehicle in the traveling lane is moving. The curve here is generated to limit $|a_{y,\max}|$ to 1 m/s^2 according to Algorithm 1.

TABLE II
BÉZIER CURVE PROPERTIES COMPARISON

Bézier curve	Curvature continuity	Trajectory Length (m)	Time (ms)
Cubic piecewise	C1 in the ends and C2 elsewhere	**79**	16.7
Quadratic piecewise [9]	C1 in ends and joints	140	11.3
Quintic [17]	C2 continuous	94	16.5

TABLE III
OBSTACLE AVOIDANCE BÉZIER CURVE COMPARISON

Bézier curve	Time (ms)	Obstacle avoidance
Cubic piecewise	**12.8**	Achieved by control point placement
Quadratic piecewise [9]	N/A	Not defined
Quintic [17]	greater than 16.5	Achieved by multiple curve generation via iteration

It should be noted that the curves are C1 continuous at the beginning and end since they start from a non-zero value. However, these values are close to zero. Consider the solid curve in Fig. 12, which gives a curvature value of ≈ 0.0027 m^{-1} at the beginning and end. For this, if front steering angle approximation by Ackerman is considered:

$$\delta_{\text{Ackerman}} = \tan^{-1}(L_f K) = 0.17° \tag{25}$$

the steering angle value is within the comfort steering angle change ($\Delta\delta$) that the controller can handle. Thus, the steering angle change from zero to the specific curvature value at the beginning and end can be carried out without affecting performance or passenger comfort. The same claim is mentioned and supported in [8] for trajectory planning using a cubic spline. While generating an LC Bézier curve using Algorithm 1 or a modified global path using Algorithm 2, any curve that requires the initial steering angle computed using (25) to be greater than the desired $\Delta\delta$ is neglected. In such cases, the curve length is increased, which reduces the curvature thus, keeping the $\Delta\delta$ within the comfort limit.

The results are compared with a quintic Bézier curve [17] and a quadratic [9] piecewise Bézier curve. Table II summarizes the results of a LC scenario that ignores the obstacles. The traveling velocity is taken as 20 m/s. Quintic and piecewise cubic curves are generated to limit $|a_{y,\max}|$ to 1 m/s^2. The quadratic piecewise Bézier curve is generated based on the traveling velocity [9] rather than limiting $|a_{y,\max}|$. The method used by [9] gives the least lateral acceleration with

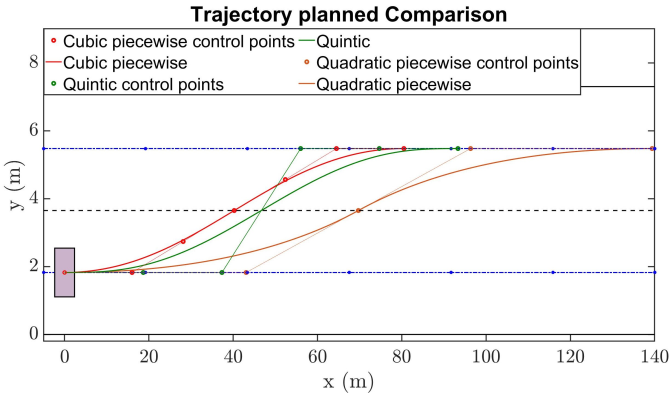


Fig. 13. Comparison of cubic piecewise, quadratic piecewise, and quintic Bézier curves.

$|a_{y,\max}| = 0.82\ \mathrm{m/s^2}$. However, considering the same velocity, the curve length is significantly longer than other curves.

Table II lists the computational time when the trajectory length L is unknown. The proposed method consumes slightly longer computation time because both L and the parameter d must be determined. Nonetheless, the computational time for generating three curves is substantially low and can be easily implemented for real-time application. It should be noted that if L is known, the computation time for the cubic piecewise Bézier curve can be further reduced. Figure 13 shows the cubic, quadratic, and quintic Bézier curves and their corresponding control points. The quintic Bézier curve uses a longer length to achieve the same lateral acceleration compared to the proposed method.

Table III compares the three methods during an obstacle avoidance scenario. For the quintic curve, as there are no control points in the middle, the curve cannot be precisely directed. Hence, obstacle avoidance is achieved by iteration, where multiple curves are generated until a collision-free curve is obtained. The computation time for the quintic Bézier curve increases during obstacle avoidance, as it may require generating multiple curves to find the optimal one. In contrast, the computational time reduces during obstacle avoidance for cubic piecewise curves as it can be treated as the case where L is known. Since the precise location of the obstacle and safe distance to be kept from the obstacle are known, L is determined before the curve generation. In any case, if L has to be determined, unlike the quintic Bézier curve, the computational time remains constant at 16.7 ms. For quadratic Bézier curve [9], as the curve generation depends on the velocity, obstacle avoidance is difficult and not covered in their study.

B. *Performance of Velocity Planning*

Assume that the lead vehicle in the traveling lane is moving for the static obstacle avoidance LC with a fixed curve length. Considering a constant velocity of $20\ \mathrm{m/s}$, $|a_{y,\max}|$ comes around $1.7\ \mathrm{m/s^2}$. In this case, limiting $|a_{y,\max}|$ to the specified range can be achieved by the velocity planner. Consider the maximum limits for the vehicles as: $|a_{y,\max}| = 1\ \mathrm{m/s^2}$, $|a_{\max}| = 1.25\ \mathrm{m/s^2}$ and $|J_{\max}| = 1\ \mathrm{m/s^3}$. The final velocity considered is also $20\ \mathrm{m/s}$. Figure 14 shows the planned trajectory, velocity profile, acceleration profile, and corresponding velocity limits. The AV should decelerate from the current velocity to $V_{1,\min}$, while phases 1 to 3 of the velocity planner are used. Similarly, the AV should accelerate from $V_{2,\min}$ to $20\ \mathrm{m/s}$. Phases 5 to 7 are used for acceleration. For region $V_{1,\min}$ to $V_{2,\min}$, general velocity planning is used to maximize the travel time. For this region, the AV cannot achieve the $a_{\max}$ since the distance required to travel from $V_{1,\min}$ to $V_{2,\min}$ is short. Hence, velocity is planned with a lower value of $a_{\max}$ for this region. The traveling distance from $V_{1,\min}$ to $20\ \mathrm{m/s}$ is calculated using (19), which is around 48 m. However, with the initial trajectory marked in red, the $V_{1,\min}$ point comes around 9 m. Hence, the AV should perform lane keeping around 39 m to achieve $V_{1,\min}$. The same calculation applies for the acceleration case $V_{2,\min}$ to $20\ \mathrm{m/s}$. Lane keeping regions are marked as green line in Fig. 14. $V_{1,\min}$ and $V_{2,\min}$ appear at an equal distance since the road shape is symmetric.

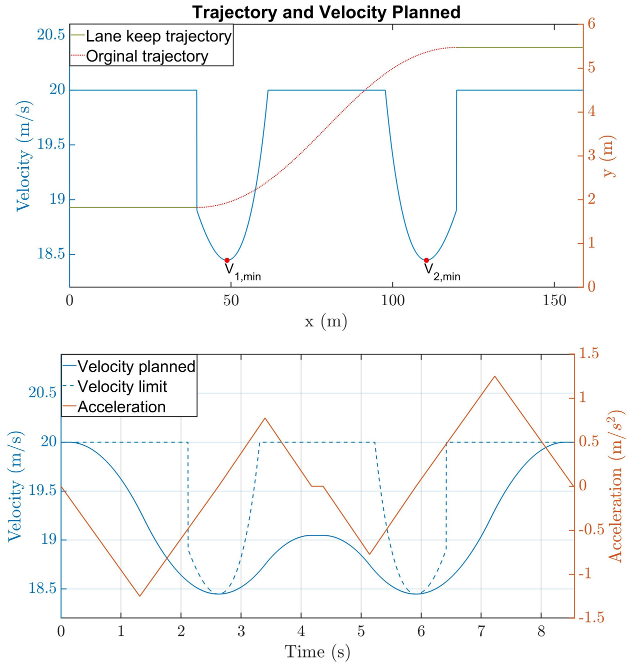


Fig. 14. : AV Lane change trajectory, velocity and acceleration.

C. *Performance of Trajectory and Velocity Tracking*

Figure 15 shows the results of trajectory and velocity tracking in the dynamic environment mentioned in Fig. 14. The Root mean square error came out to be 0.0059 m for trajectory reference tracking and around 0.016 m/s for velocity tracking. The time required to complete the tracking from start to end is around 2.21 s. From the trajectory and velocity obtained from the planner, the time required to complete the LC scenario mentioned in Fig. 14 is around 8.3 s. The NMPC can provide the control inputs to complete the LC trajectory and velocity tracking within a quarter of the total LC duration. Thus, the NMPC used for trajectory tracking can be efficiently implemented in real-time applications.

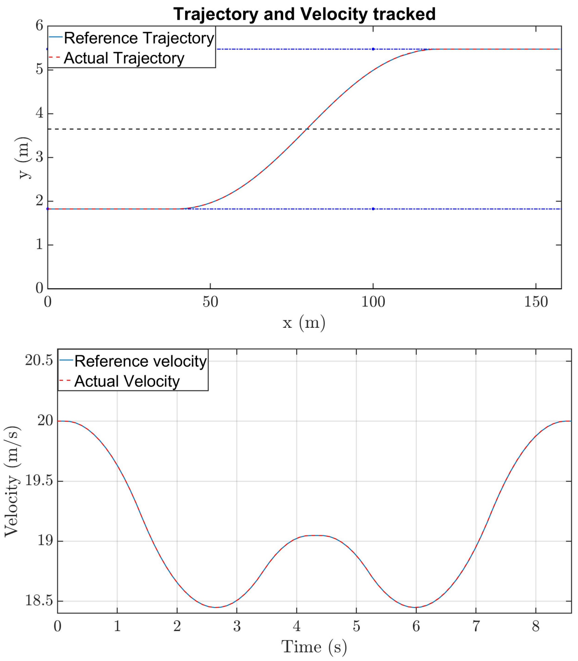


Fig. 15. Trajectory and velocity tracked using NMPC.

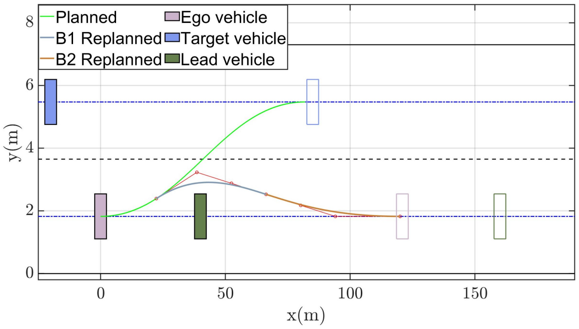


Fig. 16. Bézier curve based replanned trajectory during unexpected delay in target vehicle position prediction.

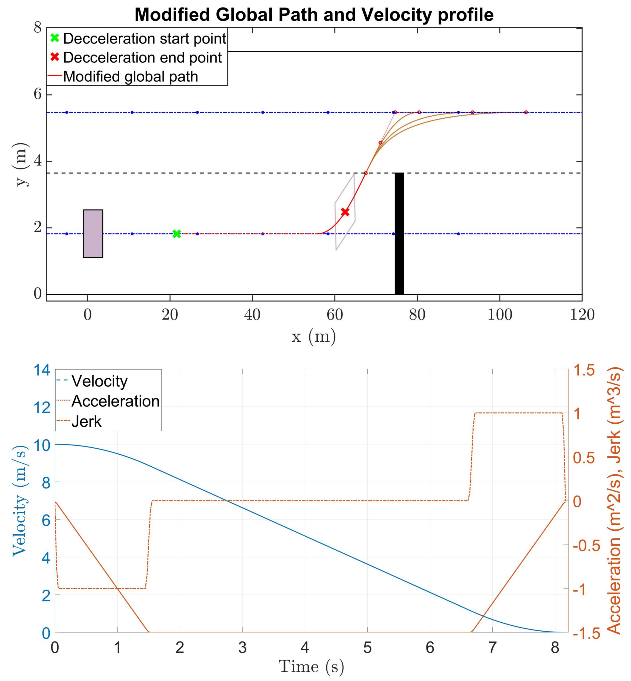


Fig. 17. Modified global path, velocity profile for lane keeping, and LC trajectories from modified global path with different lead vehicle position.

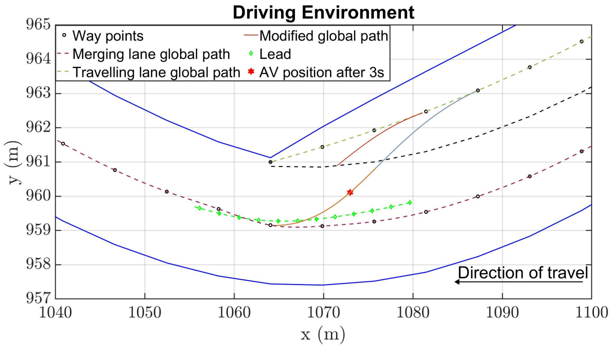


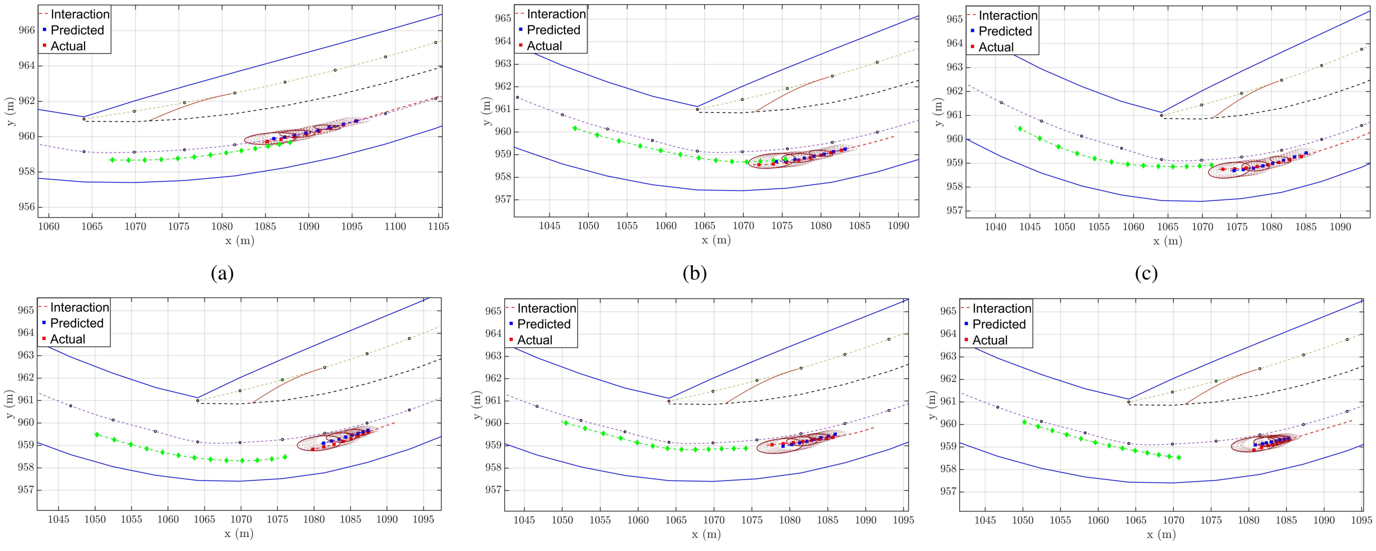
Fig. 18. Forced LC Scenario from INTERACTION dataset.

D. Performance of Trajectroy Replanning

Consider an unlikely scenario with the proposed decision making framework where a considerable delay was observed in the target vehicle trajectory prediction. The AV initially planned an LC trajectory and started the LC process in this case. However, the prediction obtained from the encoder-decoder LSTM shows that the target vehicle would block the LC, and there is no feasible trajectory for LC. Therefore the AV changes the LC decision initiated to lane keeping. Figure 16 shows resulting trajectory obtained after changing the LC decision to lane keeping. The green line shows the initially planned LC trajectory, assuming that the target vehicle yields for the ego vehicle. Instead, the target vehicle blocks the LC. Consider the traveling velocity of 20 $\mathrm{m/s}$ and a lead vehicle in the traveling lane with constant velocity. Since the LC occurs at a higher velocity, the LC duration is taken as 6 s [9]. In Fig. 16, the filled plot shows the vehicles in the initial position, and the other box shows the position after 6 s. The replanned trajectory length is obtained by subtracting the final position of the lead vehicle with S_{df} in (1). A constant velocity replanning is considered here. For this scenario $|a_{y,\max}|$ came around 0.9 $\mathrm{m/s^2}$. If the obtained $|a_{y,\max}|$ is higher than the considered $|a_{y,\max}|$, either the replanning duration is increased, or a variable velocity planning is considered.

E. Analysis on Forced LC

Figure 17 shows the modified global path in red and LC trajectories from the traveling lane to the merging lane. Different LC length indicates merging behind the lead vehicle at different positions. Consider the ego vehicle in the traveling lane travels at 10 $\mathrm{m/s}$. Using (19), the distance required to decelerate with 1.5 $\mathrm{m/s^2}$ from 10 to 0 $\mathrm{m/s}$ comes around 40.83 m. The deceleration distance lies between the green and red cross in Fig. 17. For this work, the deceleration limit for lane keeping is considered lower than $-a_{\max}$ to consider the possibility of an LC. Hence, AV does not have to start from $-a_{\max}$ to begin the LC maneuver effectively.



(a) (b) (c)

(d) (e) (f)

Fig. 19. Trajectory prediction from encode-decoder LSTM with error ellipse.

F. Analysis on the INTERACTION Dataset

Figure 18 shows the INTERACTION scenario plotted in MATLAB with a random LC trajectory. The plot indicates waypoints, global paths on traveling and merging lanes, respectively, and the modified global path. The green plot shows the rear of the lead vehicle position during the LC. The red spot indicates the position of ego vehicle after 3 s, which is used to check for collision with the target vehicle.

Figures 19(a) to 19(e) show the prediction results when the encoder-decoder LSTM is implemented in the test set against the ground truth. The red dotted line without the solid block represents the initial interaction horizon for 3 s. The plot demonstrates the effectiveness of the error ellipse, encompassing the target vehicle deviation within the ellipse boundary. Upon analyzing the different combinations of network structures, the best results arises from the encoder-decoder LSTM architecture featuring 128 neurons, 570 epochs, and 25 batch sizes. However, the LSTM architecture with 64 neurons, 550 epochs, and 15 batch sizes also presents a competitive result. Since the prediction result shows only a small variation in the average RMSE ($\approx$0.1 m in the x and $\approx$ 0.005 m in the y direction), the model with 64 neurons, 550 epoch, and 15 batch size is chosen as the final combination to avoid a deep structure and overfitting. The error ellipse is constructed in each second to show the error propagation in the prediction horizon. It is clear that as the prediction horizon increases, the ellipse size also increases, taking up useful navigation space.

Table IV shows the effect of different interaction horizons (h_i) on the prediction horizon (h_p), where x_e and y_e show the RMSE error in the x and y positions. For shorter predictions, the worst performance emerges when h_i = 1 s, and the performance of h_i = 2 s outperforms that of h_i = 3 s. However, the average x_e is much larger when h_i = 2 s and h_p = 3 s, even though the average y_e is slightly better. Thus, when h_p = 3 s, the interaction horizon is set to 3 s. It is important to note that the data should be normally distributed to construct the ellipse. With a large test sample size containing 223 LC scenarios, with each LC scenario recorded at 0.1 s time step satisfies this condition and makes the error data large enough at each second to construct the ellipse. If the test samples were fewer, one ellipse could be plotted with all the error data accumulated over the prediction horizon.

TABLE IV
EFFECT OF h_i ON h_p

	$h_i = 1$ s		$h_i = 2$ s		$h_i = 3$ s	
h_p(s)	x_e	y_e	x_e	y_e	x_e	y_e
0.0 to 1.0	1.47	0.18	0.77	0.07	0.82	0.08
1.1 to 2.0	2.1	0.23	1.22	0.09	1.24	0.10
2.1 to 3.0	2.61	0.28	1.95	0.12	1.56	0.13
Average	2.06	0.22	1.31	0.09	1.2	0.10

Figures 20 to 23 show some of the selected decision making scenarios in the test set. Figure 21 and 20 show the results two successful LC cases when τ = 1.5 s, including their corresponding velocity and acceleration profiles from the test samples. In both cases, acceleration phases 1 to 3 are used for velocity planning, as the curve satisfies the considered $|a_{y,\max}|$ due to the low-velocity. Figure 21 shows the results of the case where LC happens before the modified global path. In this case, the AV follows the global path till the end and plans the LC trajectory considering the lead vehicle position. Figure 20 shows the results of the LC case where the initial P_3 for the LC trajectory goes beyond $P_{3,\text{global}}$. In this instance, P_3 is modified to $P_{3,\text{global}}$ to ensure a collision-free trajectory, as mentioned in Section IV-C.

Figure 22 shows the results of the case where the LC trajectory with τ = 1.5 s collides with the error ellipse. The collision is not directly observable in the plot. Nevertheless, when the target vehicle width gets added to the AV, LC trajectory intercepts with the error ellipse. However, a safe LC trajectory is achieved when τ = 0.9 s, enabling the AV to

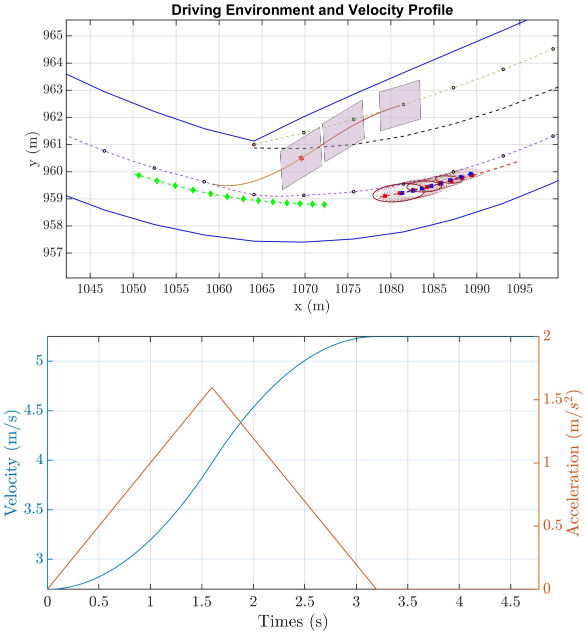


Fig. 20. AV Lane change scenario from modified global path and corresponding velocity profile.

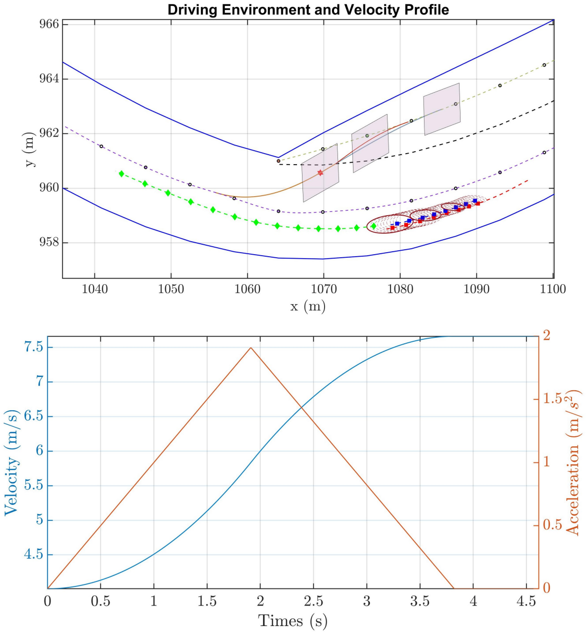


Fig. 21. AV Lane change scenario and corresponding velocity profile.

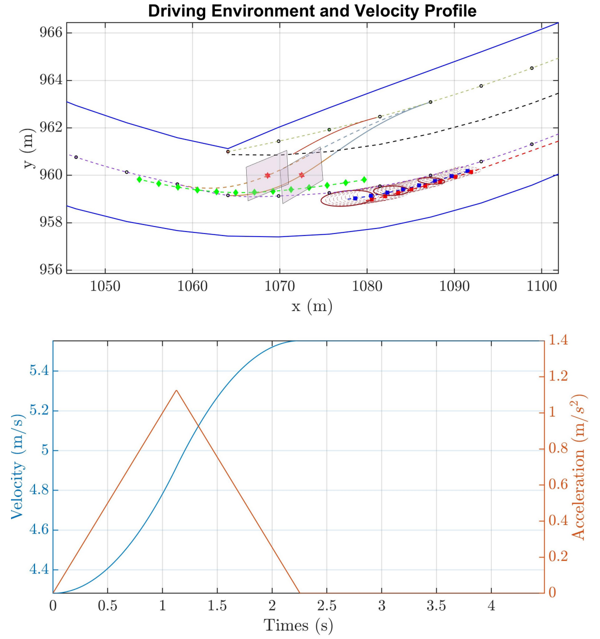


Fig. 22. AV LC scenario with different τ values and velocity profile.

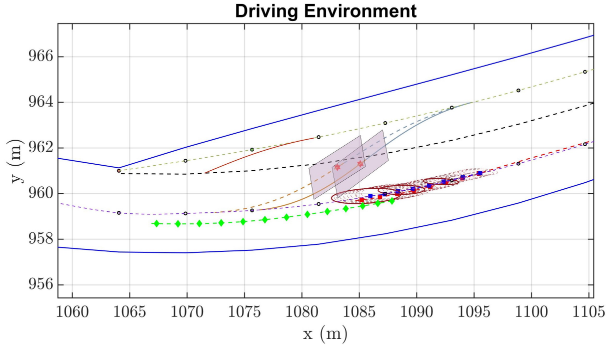


Fig. 23. Unsuccessful LC scenario.

execute the LC successfully. The velocity profile for the LC trajectory when τ = 0.9 s is also displayed in Fig. 22.

Figure 23 shows the case where the target vehicle blocks the LC of the AV. Here, two LC paths are depicted, i.e., when τ = 1.5 s and when τ = 0.7 s. AV collides with the ellipse in both cases and performing an LC is unsafe.

Upon expanding the proposed decision making framework on the test set, the AV was able to detect all the blocking behaviors of the target vehicle during the LC. On the other hand, about 2 % of the successful LC were misidentified as a blocking scenario. This may be due to the assumptions used in the study, like starting the LC from the global path, or because human drivers might have performed an aggressive LC and the AV considered these LC unsafe.

VIII. CONCLUSIONS

The paper proposes a holistic approach for automated driving, combining an efficient lateral maneuver planner using piecewise cubic Bézier curves with a safe decision making framework. Compared to the quintic Bézier curve, the piecewise curve shows better controllability, shorter trajectory length, and compatible curvature characteristics. The coupled velocity planner generates reference velocities considering passenger comfort, and an NMPC ensures accurate tracking of the planned trajectory and velocity with negligible errors. The decision making framework incorporates an LSTM-based encoder-decoder architecture for predicting the future position of the target vehicle, ensuring collision-free courses by bounding uncertainties with error ellipses. Although initial lateral position deviation by the AV affects the accuracy of successful lane changes, the framework successfully detects blocking behaviors of all vehicles. Future work involves determining the lead vehicle position in addition to the target vehicle position for determining the Bézier curve length.

REFERENCES

[1] M. Shawky, "Factors affecting lane change crashes," *IATSS Research*, vol. 44, no. 2, pp. 155–161, 2020.

[2] E.-K. Lee, M. Gerla, G. Pau, U. Lee, and J.-H. Lim, "Internet of vehicles: From intelligent grid to autonomous cars and vehicular fogs," *International Journal of Distributed Sensor Networks*, vol. 12, no. 9, p. 1550147716665500, 2016.

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

[4] F. Gul, I. Mir, L. Abualigah, P. Sumari, and A. Forestiero, "A consolidated review of path planning and optimization techniques: Technical perspectives and future directions," *Electronics*, vol. 10, no. 18, p. 2250, 2021.

[5] H. Mouhagir, R. Talj, V. Cherfaoui, F. Aioun, and F. Guillemard, "Evidential-based approach for trajectory planning with tentacles, for autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 8, pp. 3485–3496, 2019.

[6] T. Mercy, R. Van Parys, and G. Pipeleers, "Spline-based motion planning for autonomous guided vehicles in a dynamic environment," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 6, pp. 2182–2189, 2017.

[7] A. Mehmood, M. Liaquat, A. I. Bhatti, and E. Rasool, "Trajectory planning and control for lane-change of autonomous vehicle," in *Proceedings of the 5th International Conference on Control, Automation and Robotics*, 2019, pp. 331–335.

[8] D. Yang, S. Zheng, C. Wen, P. J. Jin, and B. Ran, "A dynamic lane-changing trajectory planning model for automated vehicles," *Transportation Research Part C: Emerging Technologies*, vol. 95, pp. 228–247, 2018.

[9] C. Chen, Y. He, C. Bu, J. Han, and X. Zhang, "Quartic bézier curve based trajectory generation for autonomous vehicles with curvature and velocity constraints," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2014, pp. 6108–6113.

[10] L. Chen, D. Qin, X. Xu, Y. Cai, and J. Xie, "A path and velocity planning method for lane changing collision avoidance of intelligent vehicle based on cubic 3-d bezier curve," *Advances in Engineering Software*, vol. 132, pp. 65–73, 2019.

[11] L. Han, H. Yashiro, H. T. N. Nejad, Q. H. Do, and S. Mita, "Bezier curve based path planning for autonomous vehicle in urban environment," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2010, pp. 1036–1042.

[12] 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.

[13] J. Moreau, P. Melchior, S. Victor, M. Moze, F. Aioun, and F. Guillemard, "Reactive path planning for autonomous vehicle using bézier curve optimization," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2019, pp. 1048–1053.

[14] J. Chen, P. Zhao, T. Mei, and H. Liang, "Lane change path planning based on piecewise bezier curve for autonomous vehicle," in *Proceedings of the IEEE International Conference on Vehicular Electronics and Safety*, 2013, pp. 17–22.

[15] H. Naseri, A. Nahvi, and F. S. N. Karan, "A real-time lane changing and line changing algorithm for driving simulators based on virtual driver behavior," *Journal of Simulation*, vol. 11, pp. 357–368, 2017.

[16] A. Artunedo, J. Villagra, and J. Godoy, "Real-time motion planning approach for automated driving in urban environments," *IEEE Access*, vol. 7, pp. 180 039–180 053, 2019.

[17] I. Bae, J. H. Kim, J. Moon, and S. Kim, "Lane change maneuver based on bezier curve providing comfort experience for autonomous vehicle users," in *Proceedings of the IEEE Intelligent Transportation Systems Conference*, 2019, pp. 2272–2277.

[18] F. Garrido, L. González, V. Milanés, J. P. Rastelli, and F. Nashashibi, "A two-stage real-time path planning: Application to the overtaking manuever," *IEEE Access*, vol. 8, pp. 128 730–128 740, 2020.

[19] F. Zhou, B. Song, and G. Tian, "Bézier curve based smooth path planning for mobile robot," *Journal of Information &Computational Science*, vol. 8, no. 12, pp. 2441–2450, 2011.

[20] L. Xu, B. Song, and M. Cao, "A new approach to optimal smooth path planning of mobile robots with continuous-curvature constraint," *Systems Science & Control Engineering*, vol. 9, no. 1, pp. 138–149, 2021.

[21] K. Yang and S. Sukkarieh, "An analytical continuous-curvature path-smoothing algorithm," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 561–568, 2010.

[22] Q. Zou, Y. Hou, and K. Xiong, "An overview of the motion prediction of traffic participants for host vehicle," in *Proceedings of the Chinese Control Conference*, 2019, pp. 7872–7877.

[23] M. Gulzar, Y. Muhammad, and N. Muhammad, "A survey on motion prediction of pedestrians and vehicles for autonomous driving," *IEEE Access*, vol. 9, pp. 137 957–137 969, 2021.

[24] B. Jin, B. Jiu, T. Su, H. Liu, and G. Liu, "Switched kalman filter-interacting multiple model algorithm based on optimal autoregressive model for manoeuvring target tracking," *IET Radar, Sonar & Navigation*, vol. 9, no. 2, pp. 199–209, 2015.

[25] S. A. Goli, B. H. Far, and A. O. Fapojuwo, "Vehicle trajectory prediction with gaussian process regression in connected vehicle environment," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2018, pp. 550–555.

[26] S. Ramyar, A. Homaifar, A. Karimoddini, and E. Tunstel, "Identification of anomalies in lane change behavior using one-class svm," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, 2016, pp. 004 405–004 410.

[27] G. He, X. Li, Y. Lv, B. Gao, and H. Chen, "Probabilistic intention prediction and trajectory generation based on dynamic bayesian networks," in *Proceedings of the Chinese Automation Congress*, 2019, pp. 2646–2651.

[28] J. Li, H. Ma, W. Zhan, and M. Tomizuka, "Generic probabilistic interactive situation recognition and prediction: From virtual to real," in *Proceedings of the 21st International Conference on Intelligent Transportation Systems*, 2018, pp. 3218–3224.

[29] W. Song, B. Su, G. Xiong, and S. Li, "Intention-aware decision making in urban lane change scenario for autonomous driving," in *Proceedings of the IEEE International Conference on Vehicular Electronics and Safety*, 2018, pp. 1–8.

[30] C. Hubmann, J. Schulz, G. Xu, D. Althoff, and C. Stiller, "A belief state planner for interactive merge maneuvers in congested traffic," in *Proceedings of the 21st International Conference on Intelligent Transportation Systems*, 2018, pp. 1617–1624.

[31] J. Li, L. Sun, W. Zhan, and M. Tomizuka, "Interaction-aware behavior planning for autonomous vehicles validated with real traffic data," in *Proceedings of the Dynamic Systems and Control Conference*, vol. 84287, 2020, p. V002T31A005.

[32] L. Lin, W. Li, H. Bi, and L. Qin, "Vehicle trajectory prediction using lstms with spatial–temporal attention mechanisms," *IEEE Intelligent Transportation Systems Magazine*, vol. 14, no. 2, pp. 197–208, 2021.

[33] S. H. Park, B. Kim, C. M. Kang, C. C. Chung, and J. W. Choi, "Sequence-to-sequence prediction of vehicle trajectory via lstm encoder-decoder architecture," in *Proceedings of the IEEE intelligent vehicles symposium*, 2018, pp. 1672–1678.

[34] L. Tang, H. Wang, W. Zhang, Z. Mei, and L. Li, "Driver lane change intention recognition of intelligent vehicle based on long short-term memory network," *IEEE Access*, vol. 8, pp. 136 898–136 905, 2020.

[35] Y. Jeong, S. Kim, and K. Yi, "Surround vehicle motion prediction using lstm-rnn for motion planning of autonomous vehicles at multi-lane turn intersections," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 1, pp. 2–14, 2020.

[36] N. Djuric, V. Radosavljevic, H. Cui, T. Nguyen, F.-C. Chou, T.-H. Lin, N. Singh, and J. Schneider, "Uncertainty-aware short-term motion prediction of traffic actors for autonomous driving," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 2095–2104.

[37] X. Tang, K. Yang, H. Wang, J. Wu, Y. Qin, W. Yu, and D. Cao, "Prediction-uncertainty-aware decision-making for autonomous vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 4, pp. 849–862, 2022.

[38] H. Van Essen and H. Nijmeijer, "Non-linear model predictive control for constrained mobile robots," in *Proceedings of the European Control Conference*, 2001, pp. 1157–1162.

[39] E. R. Khansari, M. Tabibi, and F. M. Nejad, "A study on following behavior based on the time headway," *Jurnal Kejuruteraan*, vol. 32, no. 2, pp. 187–195, 2020.

[40] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "Predictive active steering control for autonomous vehicle systems," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 3, pp. 566–580, 2007.

[41] Z. Li, J. Jiang, W.-H. Chen, and L. Sun, "Autonomous lateral maneuvers for self-driving vehicles in complex traffic environment," *IEEE Transactions on Intelligent Vehicles*, 2022.

[42] W. Zhan, L. Sun, D. Wang, H. Shi, A. Clausse, M. Naumann, J. Kummerle, H. Konigshof, C. Stiller, A. de La Fortelle *et al.*, "Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps," *arXiv preprint arXiv:1910.03088*, 2019.

[43] H. O. Lancaster and E. Seneta, "Chi-square distribution," *Encyclopedia of Biostatistics*, vol. 2, 2005.

[44] Q. Zhang, R. Langari, H. E. Tseng, D. Filev, S. Szwabowski, and S. Coskun, "A game theoretic model predictive controller with aggressiveness estimation for mandatory lane change," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 1, pp. 75–89, 2019.

[45] Q. Zhang, D. Filev, H. E. Tseng, S. Szwabowski, and R. Langari, "Addressing mandatory lane change problem with game theoretic model predictive control and fuzzy markov chain," in *Proceedings of the American Control Conference*, 2018, pp. 4764–4771.

[46] J. Nilsson and J. Sjöberg, "Strategic decision making for automated driving on two-lane, one way roads using model predictive control," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2013, pp. 1253–1258.

[47] S. Liu, "An on-line reference-trajectory generator for smooth motion of impulse-controlled industrial manipulators," in *Proceedings of the 7th International Workshop on Advanced Motion Control*, 2002, pp. 365–370.

Harikrishnan Vijayakumar received the B.Tech degree in mechanical engineering from Lourdes Matha College of Science and Technology, India, in 2017, and the M.S. degree in automotive engineering from Loughborough University, U.K., in 2019. He is currently pursuing the Ph.D. degree at University of Glasgow, U.K. His current research interest include decision making, path planning and control of automated vehicles.

Dezong Zhao (Senior Member, IEEE) received the B.Eng. and M.S. degrees from Shandong University in 2003 and 2006, respectively, and the Ph.D. degree from Tsinghua University in 2010, all in Control Engineering. Since 2023 he is a Reader in Autonomous Systems with the University of Glasgow. His research interests include connected and automated vehicles, robotics, machine learning and control engineering. He has been an EPSRC Innovation Fellow since 2018 and a Royal Society-Newton Advanced Fellow since 2020.

Jianglin Lan received the Ph.D. degree from University of Hull in 2017. He is a Leverhulme Early Career Fellow and Lecturer at University of Glasgow since May 2022. Prior to this, he held postdoc positions at Imperial College London, University of Glasgow, Loughborough University, and University of Sheffield, respectively. His research interests include machine learning, optimization and control for autonomous systems.

Wenjing Zhao received the Ph.D. degree in traffic engineering with Central South University, Changsha, China, in 2022. She is currently a Postdoctoral Fellow with the Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Hong Kong. Her research interests include traffic safety, driving behaviour analysis, and connected vehicles.

Daxin Tian (Senior Member, IEEE) is currently a professor in the School of Transportation Science and En- gineering, Beihang University, Beijing, China. He is IEEE Senior Member, IEEE Intelligent Trans- portation Systems Society Member, and IEEE Vehicular Technology Society Member, etc. His current research interests include mobile computing, intelligent transportation systems, vehicular ad hoc networks, and swarm intelligent.

Dachuan Li received his Ph.D. in Control Science and Engineering from Tsinghua University (2015), China. He is currently a Research Assistant Professor in Research Institute for Trustworthy Autonomous Systems, Southern University of Science and Technology, China. His research interests in clude autonomous driving vehicles, trustworthy autonomous systems, intelligent decision-making and motion planning.

Quan Zhou (Member, IEEE) received the Ph.D. degree from the University of Birmingham, Birmingham, U.K., in 2019. He is currently an Assistant Professor with the University of Birmingham. His research interests include fuzzy inferences, evolutionary computation, deep reinforcement, and adversarial learning and their applications in automotive engineering. Hisworkwas the recipient of an award from Innovate U.K., for the commercialization of university research.

Kang Song received the B.S. and Ph.D. degree from Tianjin University, Tianjin, China, in 2011 and 2015, respectively. He is currently an Associate Professor with the State Key Laboratory of Engines, Tianjin University. His research interests include modeling, and control of Engines and Vehicles.