

Preview Path Tracking Control With Delay Compensation for Autonomous Vehicles

Shaobing Xu¹, Hui Peng¹, and Yifan Tang, *Senior Member, IEEE*

Abstract—Delay and lag deteriorate path tracking accuracy and system stability. If not properly compensated, they can cause instability or limit the driving speed of autonomous vehicles. This paper presents a preview steering control design considering both communication delay and steering lag to achieve accurate, smooth, and computationally efficient path tracking for highly automated vehicles. Two strategies are adopted for the delay: forward state predictor and delay augmentation. The steering lag is approximated by a first-order lag system. The path tracking problem with delay and lag is solved by the preview control theory. The resulted controller is in an analytical form and is computationally efficient for online implementation. We also analyze the system stability and closed-loop responses in both the time and frequency domain. The control is implemented on an automated vehicle platform and tested inside Mcity and on open roads. Experimental results showed lower tracking errors and significantly improved stability margin compared to the controls ignoring delay and lag.

Index Terms—Autonomous vehicles, path tracking, time-delay system, preview control.

I. INTRODUCTION

A. Motivation

SMOOTH, accurate, and computationally efficient vehicle motion control is crucial both for safety and ride comfort, and is a critical feature of connected and automated vehicles (CAVs). Vehicle motion control involves both longitudinal speed tracking and lateral path tracking. This paper focuses on the latter. Different path tracking algorithms were proposed in the literature [1], [2]. For instance, Paden *et al.* surveyed and discussed the typical path-tracking techniques for self-driving urban vehicles, including the pure pursuit control, rear/front wheel based feedback, feedback linearization, control Lyapunov design, and linear/nonlinear model predictive control (MPC) [1]. One critical factor that was often ignored in many of the existing path-tracking algorithms is the delay and lag of sensing, communication and steering system [1]–[6]. The delay mainly comes from the

communication between control modules and steering system, where lots of intermediate links exist, including when the CAN-bus is used. The steering lag, also known as steering dynamics, is just the fact it takes time for the tire to be steered, and for the tire contact-patch to develop the slip angle which then generates lateral tire force. It is dominated by hardware capability and the low-level control design [7]. Ignoring the delay and lag definitely creates model mismatch as well as deteriorated performance, e.g., steering oscillation and instability [8]. Nowadays, many academic research organizations use automated vehicle platforms that are modified from production passenger vehicles by installing by-wire control modules or new actuators (e.g., the widely-used MKZ/Fusion platform from Dataspeed, or the Kia platform from Polysync). The retrofit creates higher time delay, which may be reduced in future platforms but does pose challenges to the current motion control systems.

Apart from the delay and lag, computing efficiency is another focus of this paper. Many feedback controls using instantaneous information only are fast enough. Look-ahead controls using future path information and online optimization, e.g., the MPC designs, enable better tracking performance but suffer from higher computational load [7], [9]–[11]. In summary, the main motivation or the key contribution of this paper is to design computationally efficient path tracking control algorithms that consider system delay, steering lag, and future path information for better tracking accuracy and stability.

B. Literature Review of Path Tracking Control

Path tracking of automated vehicles has been widely studied [1]–[10]. Chaib *et al.* compared the H_∞ , adaptive, PID, and fuzzy control for lane tracking by simulations [2]. The pure-pursuit and PID design were introduced in [12] and [13]. The classic MPC strategy was also studied by many [3]–[5], [10]. An adaptive sliding-mode control was designed for non-holonomic wheeled mobile robots in [6]. A safety-oriented feedback controller was proposed to achieve bounded tracking errors [14]. A control to handle multiple plants was proposed for fault-tolerant lane-tracking, e.g., when one of the sensors failed [15]. Reference [16] presented a barrier-protected preview control to bound tracking errors. Most of these algorithms did not consider delay and lag explicitly. We should note the fact that time-delay control theories made significant progress over the past few decades, for example, stability analysis with fixed/varying delay, robust time-delay control, and optimal control of delayed

Manuscript received April 26, 2019; revised September 29, 2019 and December 24, 2019; accepted February 24, 2020. This work was supported by the UM-SFmotors automated Car Project and in part by the OpenCAV Project of Mcity. The Associate Editor for this article was B. Fidan. (*Corresponding author: Shaobing Xu.*)

Shaobing Xu and Hui Peng are with the Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 48109 USA, and also with the Mcity, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: xushao@umich.edu; hpeng@umich.edu).

Yifan Tang is with SF Motors Inc., Santa Clara, CA 95054 USA (e-mail: yifan.tang@sfmotors.com).

This article has supplementary downloadable material available at <http://ieeexplore.ieee.org>, provided by the authors.

Digital Object Identifier 10.1109/TITS.2020.2978417

1524-9050 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.

See <https://www.ieee.org/publications/rights/index.html> for more information.

system [8], [17]. Different strategies to handle time delay in linear systems were proposed, e.g., state augmentation to achieve delay-free systems [18], or using predicted future states. These strategies are used in many existing papers to solve various engineering problems [19], [20]. This paper also leverages them to solve the motion control problem of CAVs.

As a classical method, the MPC can deal with future road curvatures and time delay by online nonlinear optimizations. It enables better tracking performance but suffers from intensive computations [11], [21]. In this paper, we will use the preview control theory for path tracking design. The concept of preview control was proposed in the 1970s [22]. It is based on linear dynamic approximation and can directly respond to future information, not relying on online numerical optimization [23]. It has been applied to different applications. For instance, Salton *et al.* designed a preview controller to reduce the settling time of dual-stage actuators [24]; Peng proposed a frequency-shaping preview lane-keeping control for frequency domain specification and better ride comfort [25]. In our previous paper, a preview control was designed, analyzed, and validated, but the time delay and steering lag are not considered, which may result in poor system stability at a higher speed [26]. The stability deterioration did be found in our recent experiments on open roads. This paper focuses on this challenge and addressing how to compensate for the time delay and steering lag in the preview control design for better system stability.

C. Contributions

The contributions of this paper include 1) a preview steering control design considering pure input delay, steering lag, and future road curvatures for accurate, smooth, and computationally efficient path tracking. The input delay is handled by two methods, i.e., an augmentation strategy and a predictor strategy. The former reformulates the original problem as an augmented delay-free system, and the latter uses predicted ahead tracking errors to compensate for the delay. The steering lag is approximated by a first-order linear lag system. The path tracking control law is solved from an augmented optimization problem and has an analytical format. 2) Analysis of i) the closed-loop system response in the time domain, ii) system stability against time delay, and iii) system responses in the frequency domain. 3) The control algorithm is implemented on an automated vehicle platform and tested by both on-track and open road experiments. Testing results showed that the proposed control significantly extends the system stability margin and reduces tracking errors. Note that the theories of preview control and augmentation strategy are not invented but leveraged in this paper.

The remainder of this paper is organized as follows: Section II presents the vehicle model; Section III proposes the strategies to deal with the input delay and steering lag; Section IV shows the preview path-tracking controller and the benchmarks. The analyses of control stability and performance in time/frequency domain are presented in Section V; Section VI describes the experimental results and Section VII concludes this paper.



Fig. 1. The studied automated vehicle—a hybrid Lincoln MKZ.

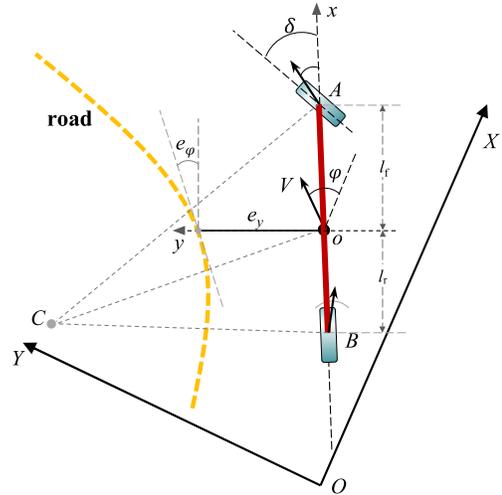


Fig. 2. Vehicle dynamics model with a reference trajectory. OXY is the inertial coordinate system and oxy is the local body-fixed coordinate system.

II. VEHICLE DYNAMICS MODEL

A. Vehicle Lateral Dynamics

The studied automated vehicle is a Hybrid Lincoln MKZ shown in Fig. 1. This platform is widely used by many research teams including Nvidia and Baidu. The by-wire control module enables automatic manipulation of the brake/throttle, steering, transmission, and turn signals. Fig. 2 shows the schematic diagram of the path tracking problem. The symbols and their values are shown in Table I. The dynamics of tracking errors are described by the following model without input delay and steering lag [26]:

$$\begin{aligned} \dot{x}_o &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\sigma_1 & \sigma_1 & \sigma_2 \\ 0 & mv_x & m & mv_x \\ 0 & 0 & 0 & 1 \\ 0 & \frac{\sigma_2}{I_z v_x} & -\frac{\sigma_2}{I_z} & \frac{\sigma_3}{I_z v_x} \end{bmatrix} x_o + \begin{bmatrix} 0 \\ \frac{2C_{af}}{m} \\ 0 \\ \frac{2l_f C_{af}}{I_z} \end{bmatrix} \delta_r + \begin{bmatrix} 0 \\ \frac{\sigma_2}{m} - v_x^2 \\ 0 \\ \frac{\sigma_3}{I_z} \end{bmatrix} c_R \\ &= \bar{A}_o x_o + \bar{B}_o \delta_r + \bar{D}_o c_R \\ x_o &= [e_y, \dot{e}_y, e_\phi, \dot{e}_\phi]^T \end{aligned} \quad (1)$$

where σ_i is the lumped coefficient, defined as

$$\begin{aligned} \sigma_1 &= 2(C_{af} + C_{ar}) \\ \sigma_2 &= -2(l_f C_{af} - l_r C_{ar}) \\ \sigma_3 &= -2(l_f^2 C_{af} + l_r^2 C_{ar}) \end{aligned} \quad (2)$$

TABLE I
SYMBOLS AND DEFINITIONS OF THE DYNAMICS MODEL

Definition	Symbol	Value & Unit
Vehicle mass	m	1800 kg
Yaw moment of inertia of the vehicle	I_z	3270 kg·m ²
Distance from c.g. to the front/rear axle	l_f/l_r	1.2/1.65 m
Cornering stiffness of the front wheel	C_{af}	70000 N/rad
Cornering stiffness of the rear wheel	C_{ar}	60000 N/rad
Longitudinal speed (in oxy)	v_x	m/s
Yaw angle of vehicle body (in OXY)	φ	rad
Front wheel steering angle	δ_r	rad
Road curvature	c_R	1/m
Orientation error with respect to road	e_φ	rad
Offset of c.g. from the trajectory	e_y	m

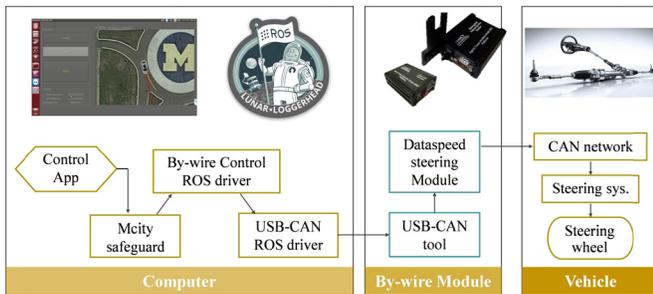


Fig. 3. The communication flow of steering command.

In this error model, $x_o \in \mathbb{R}^4$ is the state vector; e_y and e_φ are the lateral offset and heading angle error, respectively; the steering angle of front wheel $\delta_r \in \mathbb{R}$ is the control input, and the road curvature $c_R \in \mathbb{R}$ is regarded as a disturbance. Note that only non-evasive maneuvers, *i.e.*, lateral acceleration less than 0.35g, are considered in this paper.

To facilitate controller design, the continuous-time system (1) is converted into a discrete-time system with a fixed sampling period $\Delta\tau$ (*i.e.*, 0.04s) and the zero-order holder (ZOH). The **discrete-time single-track model** is

$$x_o(k+1) = A_o x_o(k) + B_o \delta_r(k) + D_o c_R(k) \quad (3)$$

B. Input Delay and Steering Lag

The input delay is mainly caused by communication and mechanical clearance. Fig. 3 shows the long communication pathway of the steering command, *i.e.*, generated by the control app, checked by the Mcity safeguard module, sent to the by-wire control ROS driver, USB-CAN ROS driver, USB-CAN tool, Dataspeed steering module, the onboard CAN network, steering system and finally the power steering motor actuates the front tires.

To identify the time delay, we input a step-wise steering command and the measured system response is shown in Fig. 4. Note that the dashed blue line denotes the output of the by-wire control module, which smoothed the steering command by limiting its changing rate. The steering wheel starts to rotate roughly 0.2s after the command δ is delivered.

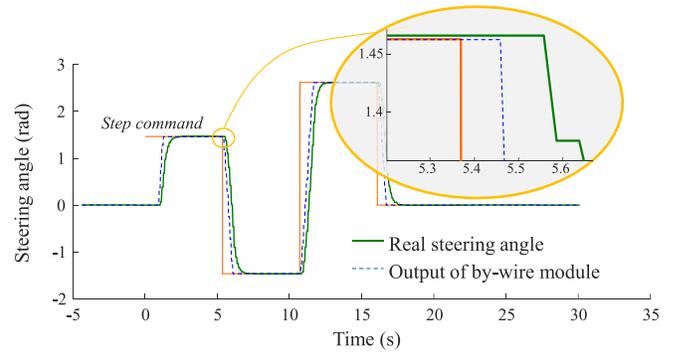


Fig. 4. Test results of input delay and steering lag.

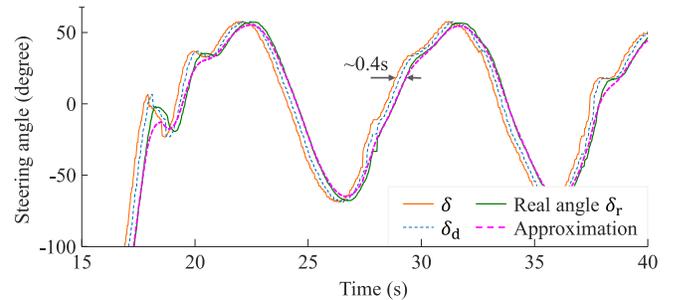


Fig. 5. Approximation of steering dynamics.

Therefore, the pure input delay τ_d is set to 0.2s for the studied platform,

$$\delta_d(t) = \delta(t - \tau_d) \quad (4)$$

where δ_d stands for the delayed steering angle command. Fig. 4 also shows that the actual steering angle falls behind the received command. This feature is called the steering lag, different from the input delay.

III. COMPENSATION STRATEGIES FOR DELAY AND LAG

Different strategies are presented in this section to deal with the steering lag and input delay.

A. Strategy for Steering Lag

The model of the steering column can be found in [27]. To simplify control design with steering lag, we use a first-order linear system to approximate:

$$\tau \dot{\delta}_r(t) = -\delta_r(t) + \delta(t - \tau_d) \quad (5)$$

where τ is the time constant, δ_r is the real steering angle, δ is the steering command. This strategy is inspired by the observation in Fig. 4, where the real steering angle tracks the commands smoothly without overshoot. In this paper, τ is set to 0.2s based on trial and error tests. To validate the model's performance, we conduct a new experiment, in which a sinusoidal steering command is used, and the vehicle speed is set to 17 m/s. The command, delayed command and real steering angle are shown in Fig. 5. With τ_d and $\tau = 0.2$ s, the steering angle profile estimated by the models (4)-(5) matches the real angle well.

Considering the first-order steering lag, the vehicle dynamics model becomes

$$\begin{bmatrix} \dot{x}_o \\ \dot{\delta}_r \end{bmatrix} = \begin{bmatrix} \bar{A}_o & \bar{B}_o \\ O & -1/\tau \end{bmatrix} \begin{bmatrix} x_o \\ \delta_r \end{bmatrix} + \begin{bmatrix} O \\ 1/\tau \end{bmatrix} \delta(t - \tau_d) + \begin{bmatrix} \bar{D}_o \\ 0 \end{bmatrix} c_R \quad (6)$$

with $\mathbf{x} = [x_o, \delta_r]^T$, it is denoted by Eq. (7) in both continuous/discrete-time format, called **full-state model**

$$\begin{aligned} \dot{\mathbf{x}} &= \bar{A}\mathbf{x} + \bar{B}\delta(t - \tau_d) + \bar{D}c_R \\ \mathbf{x}(k+1) &= A\mathbf{x}(k) + B\delta(k - N) + Dc_R(k) \\ \delta_d(k) &= \delta(k - N) \end{aligned} \quad (7)$$

where $A \in \mathbb{R}^{5 \times 5}$, $B \in \mathbb{R}^5$, $D \in \mathbb{R}^5$, and N is the number of delayed steps.

B. Strategies to Handle Input Delay

1) *Delay Augmentation*: This strategy is leveraged from the existing optimal control theory of discrete time-delay system; it reforms the dynamics (7) to a delay-free system by augmenting the system state vector [19], *i.e.*, transform the delayed commands as new states, which yields the **delay-augmented model**

$$\begin{bmatrix} \mathbf{x}(k+1) \\ \delta(k-N+1) \\ \delta(k-N+2) \\ \vdots \\ \delta(k) \end{bmatrix} = \begin{bmatrix} A & B & & & \\ & 1 & \cdots & 0 & \\ & 0 & \ddots & 0 & \\ & & \vdots & \ddots & \ddots \\ & & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \delta(k-N) \\ \delta(k-N+1) \\ \vdots \\ \delta(k-1) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \delta(k) + \begin{bmatrix} D \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} c_R(k) \quad (8)$$

With $x(k) = [\mathbf{x}(k), \delta(k-N), \dots, \delta(k-1)]^T \in \mathbb{R}^{5+N}$, it is denoted by

$$x(k+1) = Ax(k) + B\delta(k) + Dc_R(k) \quad (9)$$

This method is called the augmentation strategy (AS) in this paper. Note that the augmented Eq. (9) is a linear delay-free system, but the system dimension changes from 5 to $5+N$.

2) *State Predictor*: This strategy uses predicted tracking errors at a future moment to replace the observed tracking errors for delay compensation. If the steering control law is a linear function of the current system states,

$$\delta(k) = \mu\mathbf{x}(k) + \hat{h}(c_R) \quad (10)$$

to cancel the delay between δ_d and δ , we then predict \mathbf{x} at the future $k+N$ step using the fixed $\delta_r(k)$ at the currently step k , denoted as \mathbf{x}_P ,

$$\begin{aligned} \mathbf{x}_P(k) &= \mathbf{x}(k+N) \\ &\cong A^N \mathbf{x}(k) + (I - A)^{-1} (I - A^N) B \delta_r(k) \\ &\quad + \sum_{i=0}^{N-1} A^{(N-1-i)} D c_R(k+i) \end{aligned} \quad (11)$$

The predicted $\mathbf{x}_P(k)$ is used to replace $\mathbf{x}(k)$ in Eq. (10) to generate steering command, *i.e.*,

$$\delta_d(k) = \mu\mathbf{x}_P(k - N) + \hat{h}(c_R) = \mu\mathbf{x}(k) + \hat{h}(c_R) \quad (12)$$

With this strategy, the command δ_d imposed on the steering system is a function of non-delayed states. We call this method the predictor strategy (PS). The main limitations of this strategy include: 1) the steering angle profile in horizon $[k, k+N]$ is unknown, but was set to a fixed value $\delta_r(k)$; and 2) prediction accuracy is sensitive to model mismatch, *e.g.*, error of driving on roads with bank angle may be significant.

IV. PREVIEW PATH-TRACKING CONTROL DESIGN

A. Formulation of Path Tracking Problem

Given the delay-augmented model (8) and (9), an accuracy and smoothness-oriented cost function \mathcal{J} over the infinite horizon is designed. Then an optimal control problem (OCP) is formulated:

$$\mathcal{J}(x, \delta) = \frac{1}{2} \sum_{k=0}^{\infty} x^T(k) Q x(k) + \mathcal{R} \delta^2(k)$$

s.t.

$$\begin{aligned} x(k+1) &= Ax(k) + B\delta(k) + Dc_R(k) \\ \delta_{\min} &\leq \delta \leq \delta_{\max} \\ Q &= \text{diag}[q_1, \dots, q_4, 0, \dots, 0] \end{aligned} \quad (13)$$

where $\mathcal{R} \in \mathbb{R}$ is a positive constant; $Q \in \mathbb{R}^{(N+5) \times (N+5)}$ is a semidefinite diagonal matrix, in which only the first 4 elements corresponding to tracking errors are nonzero. The saturation of control input δ is formulated as a hard constraint.

The problem in Eq.(13) requires knowledge of the road curvature c_R over the infinite horizon. However, the road curvature is usually available in a finite horizon only. Instead of using the infinite horizon, a more sensible approach is to use c_R only in a finite interval $[k, k+\mathbb{N}]$, where \mathbb{N} is the preview steps. c_R beyond $k+\mathbb{N}$ is simplified as zero (*i.e.*, straight road):

$$c_R(i) = 0, i \in [k+\mathbb{N}+1, \infty) \quad (14)$$

This strategy works because c_R in the far future has little effects on the current steering control, which will be shown in the next subsection.

B. Design of Preview Control With Delay and Lag

The formulated path tracking problem (13)-(14) is a typical constrained OCP with time-varying disturbance, *i.e.*, future road curvature c_R . This problem is similar to the one in our previous paper [26]; but consideration of the time delay and steering lag distinguishes them. Here we also design a preview control to deal with the future road curvature for this delay/lag-involved system.

Different from the MPC, the preview control pursues analytical solutions by reformulating the original problem as a standard LQR [22]. It removes the system disturbance within the preview window, *i.e.*, $c_R(i)$, $i \in [k, k+\mathbb{N}]$, and incorporates

them into the system state vector. The curvature-augmented state $\mathcal{X}(k)$ is denoted by

$$\mathcal{X}(k) = \begin{bmatrix} x(k) \\ \mathcal{C}_R(k) \end{bmatrix} \in \mathbb{R}^{N+N+6}$$

$$\mathcal{C}_R(k) = [c_R(k), c_R(k+1), \dots, c_R(k+N)]^T \quad (15)$$

where $\mathcal{C}_R \in \mathbb{R}^{N+1}$ is the new state vector. This paper will not present the detailed design process of preview controller, which can be found in our previous paper [26]. Instead, the final optimal control law is directly presented here,

$$\delta^*(k) = -K_b x(k) - K_f \mathcal{C}_R(k)$$

$$= -K_b x(k) - \sum_{i=1}^{N+1} K_{f,i} c_R(k+i-1) \quad (16)$$

$$K_b = (\mathcal{R} + \mathcal{B}^T \mathcal{P} \mathcal{B})^{-1} \mathcal{B}^T \mathcal{P} \mathcal{A}$$

$$K_{f,i} = (\mathcal{R} + \mathcal{B}^T \mathcal{P} \mathcal{B})^{-1} \mathcal{B}^T \zeta^{i-1} \mathcal{P} \mathcal{D}$$

where the matrix \mathcal{P} is introduced into the Riccati equation of the original system ignoring the road curvature, i.e.,

$$\mathcal{P} = \mathcal{Q} + \mathcal{A}^T (I + \mathcal{P} \mathcal{B} \mathcal{R}^{-1} \mathcal{B}^T)^{-1} \mathcal{P} \mathcal{A} \quad (17)$$

K_b is a $(5+N)$ -dimension feedback gain vector corresponding to the path-tracking errors and real steering angle, i.e., $\mathfrak{x}(k) = [e_y, \dot{e}_y, e_\phi, \dot{e}_\phi, \delta_r]^T$, as well as the delayed steering commands $[\delta(k-N), \dots, \delta(k-1)]$. K_f is an $(N+1)$ -dimension gain vector for future road curvatures. As a summary, the control law (16) consists of two parts:

- 1) Feedback control of the system states $x(k)$, and
- 2) Feedback control of the curvature c_R . Since this part deals with future information and generates preparative steering, it is also called feedforward action.

Further decoupling $x(k)$ into $x_o(k)$, $\delta_r(k)$, and $\delta(k-i)$, the control becomes

$$\delta^*(k) = - \underbrace{K_{b,1-4} x_o(k)}_{\text{errors}} - \underbrace{K_{b,5} \delta_r(k)}_{\text{steering angle}}$$

$$- \underbrace{\sum_{i=0}^{N-1} K_{b,6+i} \delta(k-N+i)}_{\text{delayed commands}} - \underbrace{\sum_{i=0}^N K_{f,i} c_R(k+i)}_{\text{feedforward}} \quad (18)$$

where the subscript of $K_{b,j}$ means the j -th element. The feedback part includes three items:

- 1) feedback control of tracking errors,
- 2) feedback control of actual steering angle, and
- 3) feedback control of the delayed steering commands.

Note that the feedforward control responds to the road curvatures by a set of optimized gains K_f , rather than online optimization. Our previous paper has shown the gain $K_{f,i}$ decreases as the preview step i increases, and finally converges to zero [26]. Namely, the impact of road curvature at future i th step attenuates gradually as i increases. Usually, 2 seconds is a long-enough preview horizon, i.e., $N = 50$.

For the control saturation, i.e., $\delta \in [\delta_{\min}, \delta_{\max}]$, we check the Hamiltonian function H of the augmented quadratic problem (13) and can find that the optimal control $\delta^*(k)$ should be

TABLE II
FEATURES OF THE PROPOSED CONTROLLER AND THE BENCHMARKS

	Controller Name	Consider time delay	Consider steering lag	Consider curvature
Main	Preview-DL control	Y (AS)	Y	Y
Benchmarks	Preview-Pure	N	N	Y
	Feedback -Pure	N	N	N
	Feedback-DL	Y(AS)	Y	N
	Preview-D	Y(AS)	N	Y
	Preview-L	N	Y	Y
	Preview-DL(PS)	Y (PS)	Y	Y

the boundaries (i.e., δ_{\min} or δ_{\max}) if $\delta^* \notin [\delta_{\min}, \delta_{\max}]$, which was analyzed in [26].

In the following, the designed Preview path control (18) that considered the time Delay and the steering Lag, is called the **Preview-DL** control for more concise presentations, where the strategy AS is used and set as the default strategy to compensate for the time delay.

C. Benchmarks

If ignoring the delay and lag in the vehicle dynamics model, then the control (18) is simplified to

$$\delta^*(k) = -K_b x_o(k) - K_f \mathcal{C}_R(k), \quad K_b \in \mathbb{R}^4 \quad (19)$$

This is the controller designed in our previous paper [26]. Here it is called the **Preview-Pure** control and is set as the main benchmark of the proposed Preview-DL control.

Further, if the road curvature is also ignored, i.e., no preview action, then the control (18) becomes

$$\delta^*(k) = -K_b x_o(k), \quad K_b \in \mathbb{R}^4 \quad (20)$$

called as the tracking-error **Feedback-Pure** control. If the time delay and lag is considered in the Feedback-Pure control, i.e.,

$$\delta^*(k) = -K_b x(k), \quad K_b \in \mathbb{R}^{5+N} \quad (21)$$

then it is called the **Feedback-DL** control.

If considering the steering lag but ignoring the input delay, then the feedback item of the delayed steering commands is removed from Eq. (18), and the controller retrogrades to

$$\delta^*(k) = K_b \mathfrak{x}(k) - K_f \mathcal{C}_R(k)$$

$$= K_{b,1-4} x_o(k) - K_{b,5} \delta_r(k) - K_f \mathcal{C}_R(k) \quad (22)$$

called as the **Preview-L** control. Similarly, if ignoring the steering lag, the control is called the **Preview-D** control.

If using the predictor strategy (PS) to replace the strategy AS, i.e., replace the states $x_o(k)$ by $\mathfrak{x}_P(k)$ in Eq. (18) and remove the delayed command, then the control (18) becomes

$$\delta^*(k) = -K_b \mathfrak{x}_P(k) - K_f \mathcal{C}_R(k), \quad K_b \in \mathbb{R}^5 \quad (23)$$

called as the **Preview-DL(PS)** control in the following. All the mentioned controls are set as benchmarks of the proposed Preview-DL control (18) to better understand its nature. Their features are summarized in Table II.

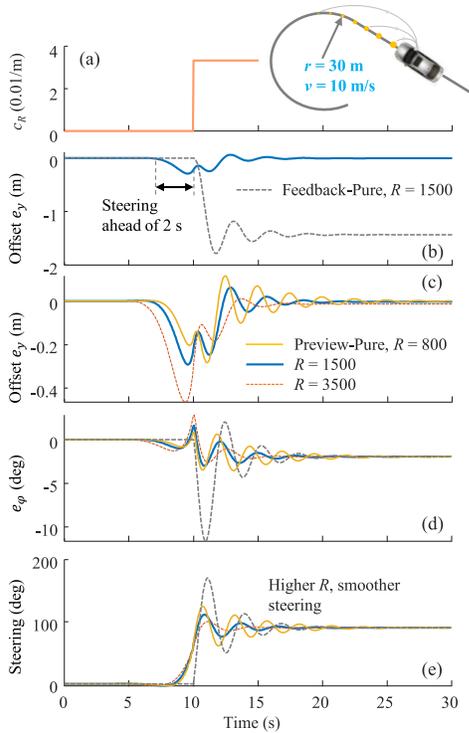


Fig. 6. Performance of the Feedback-Pure and the Preview-Pure control ignoring delay and lag. a) Road curvature; b) tracking error; c-e) Preview-Pure control under different R .

V. ANALYSIS OF CONTROL PERFORMANCE

Numerical simulations, stability analysis, and frequency response are presented in this section to gain insights into the proposed control algorithms.

A. Step Response in Time Domain

As shown in Fig. 6 (a), the vehicle runs along a straight road and then enters a circular section with a fixed radius of 30 meters. This scenario generates a step-shaped curvature profile. The vehicle parameters and their values are listed in Table I, $\Delta\tau$ is set to 0.04s, Q is set to $\text{diag}(3, 5, 7, 1, 0, \dots, 0)$, R is set to three levels: 800, 1500, and 3500. The vehicle speed is fixed at 10 m/s. The model (6) with both the input delay and the steering dynamics is used in this simulation to approximate real vehicle responses, which removes model mismatch and facilitates understanding of the controllers' behavior. The MATLAB and Simulink are used as the main software.

1) *Benefits of Using Future Road Curvature and Effects of Cost Function Design:* Compared to the Feedback-Pure control, the Preview-Pure control considering the future road curvatures achieved much better tracking accuracy. Note that neither of them considered delay and lag. As shown in Fig. 6 (b), under the same Q and $R = 1500$, the Preview-Pure controller reduces the maximal lateral offset e_y from 170 cm to 29cm and reduces the steady-state e_y from 143 cm to 3.6 cm. It takes actions ahead of entering the curve, while the Feedback-Pure control acts only after entering the curve. This preemptive operation results in smoother and more accurate path following.

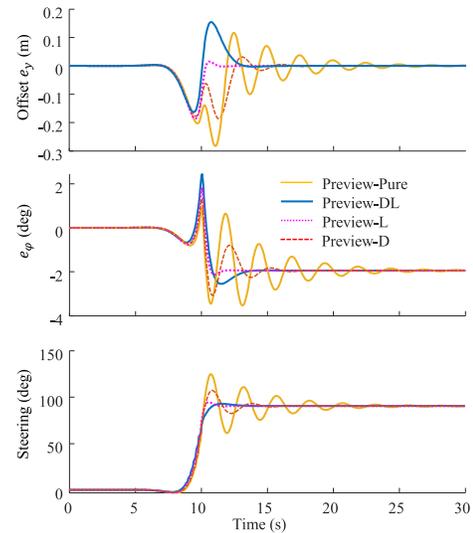


Fig. 7. Performance of the Preview-Pure, Preview-DL, Preview-D, and Preview-L control that considers or ignores delay or/and lag.

All controllers' performances are directly determined by the design of cost function, i.e., the matrices Q and R . Here we take the Preview-Pure control as an example. Given the same Q , decreasing R from 3500 to 1500, meaning we prefer tracking accuracy and reduce the need for ride comfort, leads to higher feedback gains and thus lower e_y , as shown in Fig. 6 (c-e). Further decreasing R for higher accuracy will no longer be feasible, due to the presence of delay and lag. For example, if R is reduced from 1500 to 800, the tracking accuracy is not improved, but the steering oscillation deteriorates significantly. Further smaller R will lead to unstable steering.

2) *Benefits from Considering Delay and Lag:* Holding $R = 800$, the proposed Preview-DL control (18) considering both the delay and lag achieves lower errors and much smoother steering compared to the Preview-Pure control, as shown in Fig. 7. If either steering lag or time delay is ignored, i.e., the Preview-D or Preview-L control, the performance gets worse but is still better than the Preview-Pure control that ignores both. Note that all feedback/feedforward gains of the involved controls are re-calculated under the same Q and R .

If the delay increases, the benefits of considering delay become more significant. As shown in Fig. 8 (a) and (b), the Preview-Pure control ignoring delay remains stable at $N = 10$, i.e., $\tau_d = 0.4$ s, loses stability at $N = 11$, and becomes unstable at $N = 12$. However, the proposed Preview-DL controller enables stable control even at $N = 25$, i.e., $\tau_d = 1$ s. In addition, Fig. 8 (c-d) shows that the errors e_y under $N = 5, 15$, and 25 are very similar; the greater the delay, the earlier the steering. This result matches human behaviors well—act earlier to counteract delay, and the negative effects of delay are greatly reduced.

In summary, considering either input delay or steering lag in the preview control design can improve stability and tracking performance.

3) *Comparison Between AS and PS:* Applying both the AS and the PS to deal with the pure time delay,

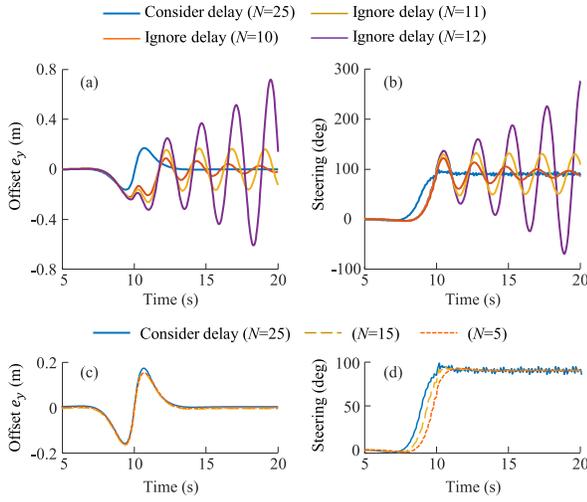


Fig. 8. Performance of the Preview-DL and the Preview-Pure control under different delay levels.

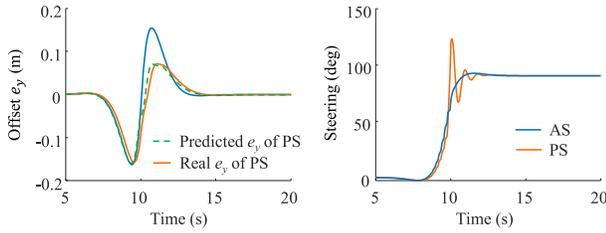


Fig. 9. Comparison between the augmentation strategy (AS) and the predictor strategy (PS) for input delay.

i.e., the Preview-DL and Preview-DL(PS) control, the control results under the same Q , $R = 800$, and $\tau_d = 0.2$ s are shown in Fig. 9. It shows that the steering of AS is much smoother than PS; the error e_y of AS is higher at 11 s, but their peaks of e_y are similar. Note that the AS considers the delay in the problem formulation and generates the theoretical optimal solution; while the PS adds an empirical compensation after the control design, which equivalently changes the feedback gains. As a result, the controller cannot guarantee global optimality, i.e., minimizing the given cost function.

B. System Stability and Response in the Frequency Domain

1) *Delay-Dependent Stability*: Assuming there is no input delay, the closed-loop system with the Preview-L control (22) is asymptotically stable due to the nature of the LQR design. However, the presence of delay changes this conclusion. Since the feedforward action does not determine system stability, let's start from the Feedback-DL control $\delta(k) = -K_b x$. Applying it to the system model (7), the characteristic function becomes

$$p(s, \tau_d) = \det(sI - \bar{A} + \bar{B}K_b e^{-\tau_d s}) \quad (24)$$

Referring to the frequency sweeping test [8], the system is stable independent of delay if and only if

$$\rho \left[(j\omega I - \bar{A}_o)^{-1} \bar{B}_o \bar{K}_b \right] < 1, \quad \forall \omega > 0 \quad (25)$$

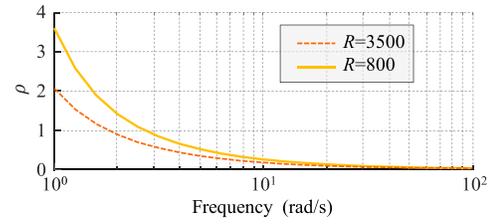


Fig. 10. Frequency sweeping test.

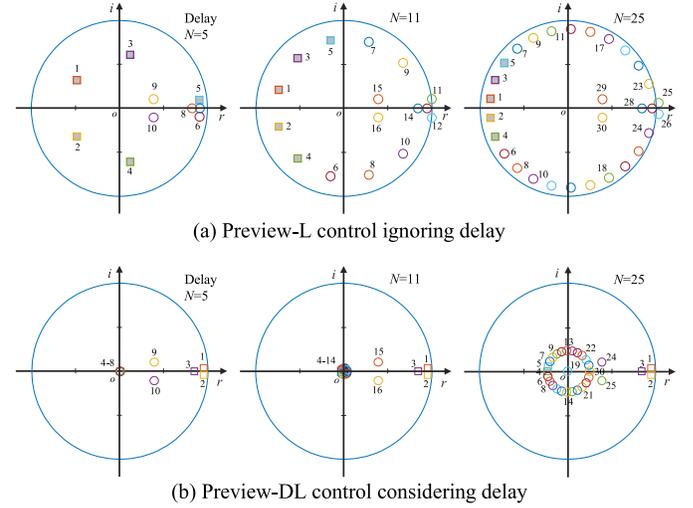


Fig. 11. Eigenvalues of $\mathcal{A} - \mathcal{B}K_b$ with the delay at 0.2 s, 0.44 s, and 1.0 s, i.e., $N = 5, 11$, and 25 , respectively. $v_x = 10$ m/s and $\mathcal{R} = 800$.

where $\rho(\cdot)$ is the spectral radius. When $v_x = 10$ m/s, $\tau_d = 0.2$ s, and $\mathcal{R} = 800$ or 3500 , ρ is shown in Fig. 10, where $\rho > 1$ if ω is lower than a certain threshold. Therefore, the system stability depends on the magnitude of the delay.

2) *Stability Analysis*: Applying the proposed Preview-DL control (18) to the delay-augmented model (9) yields the closed-loop system

$$x(k+1) = (\mathcal{A} - \mathcal{B}K_b)x(k) + (\bar{\mathcal{D}} - \mathcal{B}K_f)\mathcal{C}_R(k) \quad (26)$$

This discrete-time delay-free system is asymptotically stable if and only if [23]

$$|\lambda_i(\mathcal{A} - \mathcal{B}K_b)| < 1 \quad (27)$$

where λ_i is the i -th (or any) eigenvalue.

If the delay is ignored in the control design process, i.e., the Preview-L control, the eigenvalues of $\mathcal{A} - \mathcal{B}K_b$ are shown in Fig. 11 (a), where $v_x = 10$ m/s and $\mathcal{R} = 800$. When the delay is 0.2s or $N = 5$, all eigenvalues stay inside the unit circle, so the controller is asymptotically stable. The system stability deteriorates as the delay increases. At $N = 11$, the closed-loop system is marginally stable and loses stability when $N > 11$. These results match the simulation results of Fig. 8 (a) well.

Q and \mathcal{R} also affect system stability. As shown in Fig. 12 (left), if \mathcal{R} is decreased from 800 to 50, the system becomes unstable even at $N = 5$. This result shows that control design ignoring input delay can be stable, but only when R is high

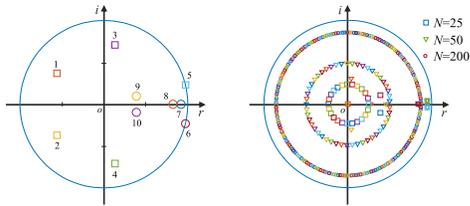


Fig. 12. Eigenvalues of $\mathcal{A} - \mathcal{B}K_b$ at $v_x = 10$ m/s. Left: Preview-L control with R reduced from 800 to 50, $N = 5$. Right: Preview-DL control with time delay at $\tau_d = 1.0$ s, 2.0 s, and 8 s, i.e., $N = 25, 50$, and 200.

enough (i.e., lower feedback gains) or the delay is small enough.

For the proposed Preview-DL control, the eigenvalues are shown in Fig. 11 (b). All eigenvalues are inside the unit circle even at $\tau_d = 1.0$ s or $N = 25$. Most of them are much closer to the origin point, meaning faster convergence and better stability. We notice that the first three eigenvalues are almost unchanged, and they dominate convergence. This result matches the step responses in Fig. 8 (c). Other eigenvalues become higher as the delay increases. At $\tau_d = 8.0$ s, the proposed control can still remain stable, as shown in Fig. 12 (right), much higher than the threshold of 0.44s if the control ignores delay.

3) *System Response in the Frequency Domain:* Applying Z-transformation to Eq. (26), the transfer function $G(z)$ from the road curvature to the states is

$$G(z) = (zI - \mathcal{A} + \mathcal{B}K_b)^{-1} (\bar{\mathcal{D}} - \mathcal{B}K_f) Z \quad (28)$$

where $Z = [1, z, \dots, z^N]^T$. The term $(zI - \mathcal{A} + \mathcal{B}K_b)^{-1}$ is independent of K_f , meaning that the feedforward control does not change system stability. If ignoring it, i.e., the Feedback-DL control, the transfer function becomes

$$G(z) = (zI - \mathcal{A} + \mathcal{B}K_b)^{-1} \mathcal{D} \quad (29)$$

Its Bode plot is given in Fig. 13, where the frequency responses of e_y and e_ϕ are detailed. The Bode plot of the Feedback-Pure control is also presented in Fig. 13.

Their results are similar at a low delay level, e.g., $N = 5$. If the delay increases to $N = 50$, the phase plot of the control considering delay increases to a positive higher level, meaning earlier action, as shown in Fig. 8 (d). However, the phase lag of the control ignoring delay is close to -180 degrees, indicating a very low phase margin for stability.

If the feedforward control is added, i.e., the proposed Preview-DL control, the system response can be further improved, as shown in Fig. 14. In the low frequency region, the amplitude is reduced, meaning lower tracking errors.

VI. EXPERIMENTAL VALIDATION

The vehicle platform shown in Fig. 1 is used to implement the proposed controllers. It is equipped with a Mobileye and an RT3003 module from OxTS. The former can measure the lane markers. The latter contains a GPS module and an Inertial Measurement Unit (IMU); it can provide real-time kinematic (RTK) positioning and directly measure the vehicle

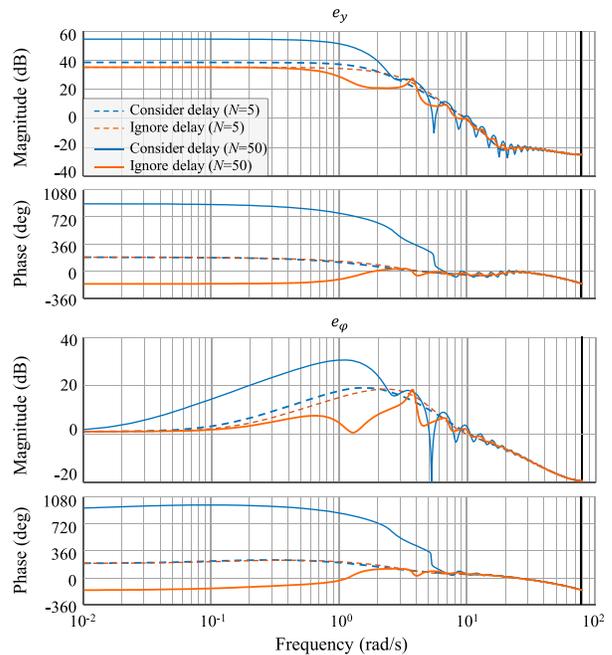


Fig. 13. Frequency response of the Feedback-DL and the Feedback-Pure control that consider and ignore the delay and lag, respectively. The time delay is set to $N = 5$ or 50.

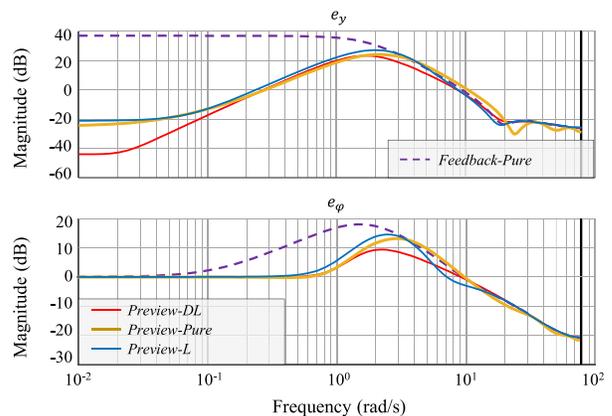


Fig. 14. Frequency response of the proposed Preview-DL control that considering all future curvatures, time delay, and steering lag.

position, lateral speed, yaw angle ϕ , and its rate $\dot{\phi}$. By-wire control allows for automation of steering, throttle, brake, and transmission shift. The designed controllers are implemented in C++ under Ubuntu. The vehicle speed is maintained by a PID controller. In the following, the proposed Preview-DL control is mainly compared with the Preview-Pure control.

A. Path Tracking Control Using RT3003

In this experiment, the car uses the RT3003 for positioning and tracks predefined trajectories. The developed software HMI is shown in Fig. 15. The tests are conducted inside Mcity, a test facility operated by the University of Michigan, as shown in Fig. 1. Two scenarios are designed, as shown in Fig. 16, 1) *shuttle loop*, which has two straight sections connected by turn-around sections with varying but continuous curvatures,



Fig. 15. Developed path-tracking control software.



Fig. 16. Two testing scenarios.

and 2) *urban loop* with repeated left turns. It consists of four curved sections connected by straight lines, ideal for testing the step response. In the tests, the road surface is clean and dry, but small road bank angle and slope do exist.

1) *Shuttle Loop*: In this scenario, the road curvature is smooth and continuous. The minimum curve radius is about 10 meters. The vehicle speed fluctuates between 0 and 60 km/h adapting to the curvature. The maximum speed is limited by the length of the testing track.

Here the Preview-DL(PS) and the Preview-Pure control are tested; the former considers the delay and lag, but the latter ignores. Note that the strategy PS is used to compensate for delay in the Preview-DL(PS) control. Experimental results are shown in Fig. 17. Their tracking error e_y in the straight sections is around 6 cm and e_ϕ is less than 1 degree (the road bank angle exists). The errors increase to about 10 cm and 8 degrees at the sharpest curve. In this case, the performances of the two controllers are comparable. One reason is that this scenario with continuous and smooth road curvatures is less sensitive to input delay and steering lag.

2) *Urban Loop*: The route consists of four quarter circles with constant radii 18, 20, 22, and 24m; the curvature profile is step-shaped, as shown in Fig. 18 (a). Here the proposed Preview-DL control using the AS strategy and the Preview-Pure control are tested. The vehicle speed is 25 km/h, corresponding to a maximal lateral acceleration of 3.9 m/s^2 , which is higher than normal human driving. The speed fluctuation is caused by the tire corner forces, while a low-gain speed control tries to keep the speed at a constant level.

Fig. 18 (d) shows that the Preview-Pure control suffers from unsmooth steering—high overshoot and oscillation. Once the delay and lag are considered (under the same Q and R),

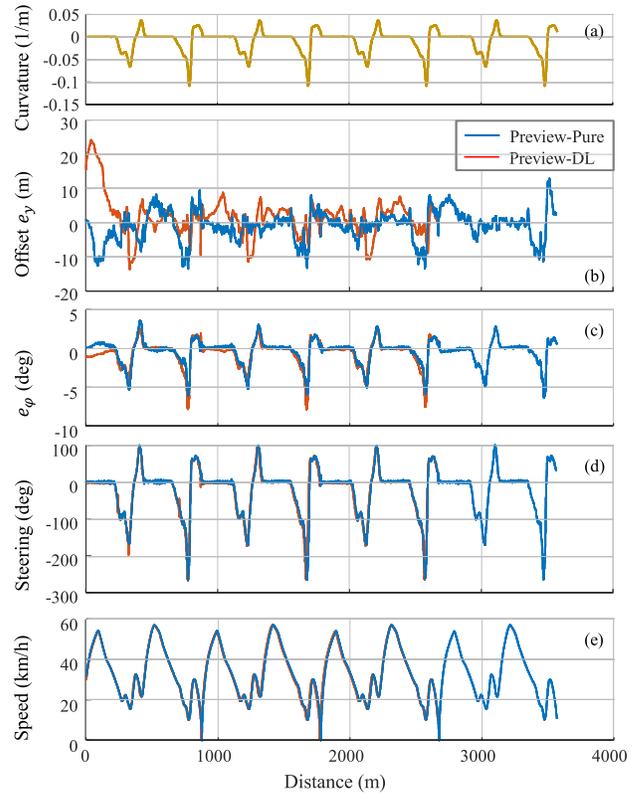


Fig. 17. Experimental results of the shuttle loop.

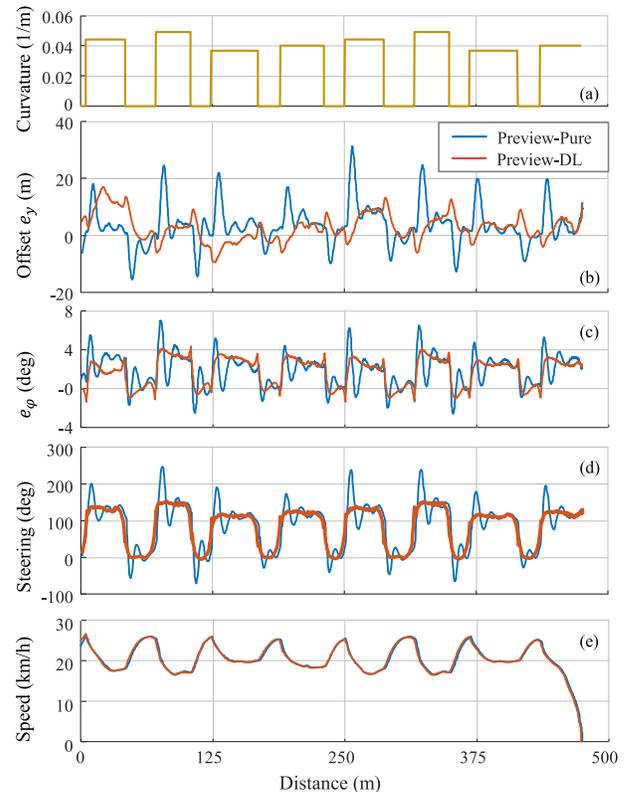


Fig. 18. Results of the urban loop, repeated left turns with step curvature.

the tracking performance is significantly improved. The overshoot and oscillation disappeared while the maximal tracking error is reduced from 35 cm to 18 cm. The maximal lateral

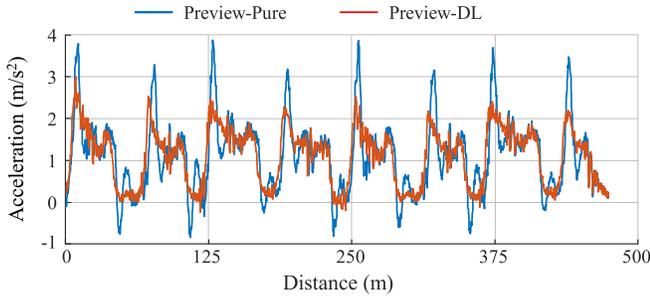


Fig. 19. Comparison of lateral acceleration.



Fig. 20. The software HMI for lane keeping control.

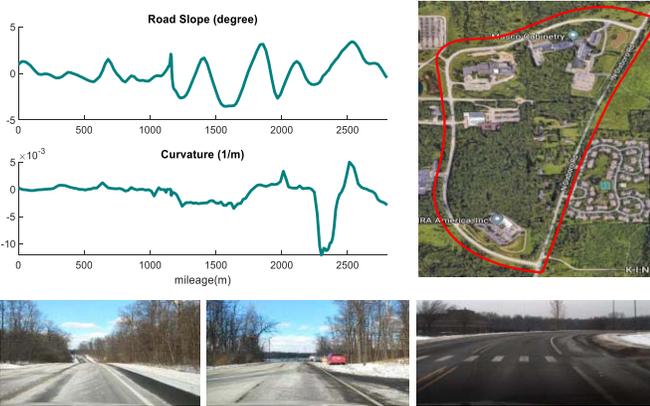


Fig. 21. The testing scenario of lane keeping control.

acceleration is reduced from 3.9 m/s^2 to about 3.0 m/s^2 in Fig. 19. These results imply better ride comfort and tracking accuracy.

B. Lane Keeping Control Using Mobileye

In this experiment, the car does not use the RTK to track a given path, but uses the Mobileye to track lane markers of a public road outside Mcity. The main difference is that the vision-based Mobileye is not as accurate and robust as the RTK, and its output is less smooth with more disturbances. This feature poses more challenges to motion control, *i.e.*, the system is more sensitive to delay and lag. The developed software is shown in Fig. 20. The testing scenario consists of a near-straight road and a highly curved road, as shown in Fig. 21. The total mileage is about 3 kilometers. The minimal radius is about 75 meters; high slope and bank angle also exist. The speed limits of the straight and curved parts are 45 and 35 mph, or 72.2 and 56.3 km/h.

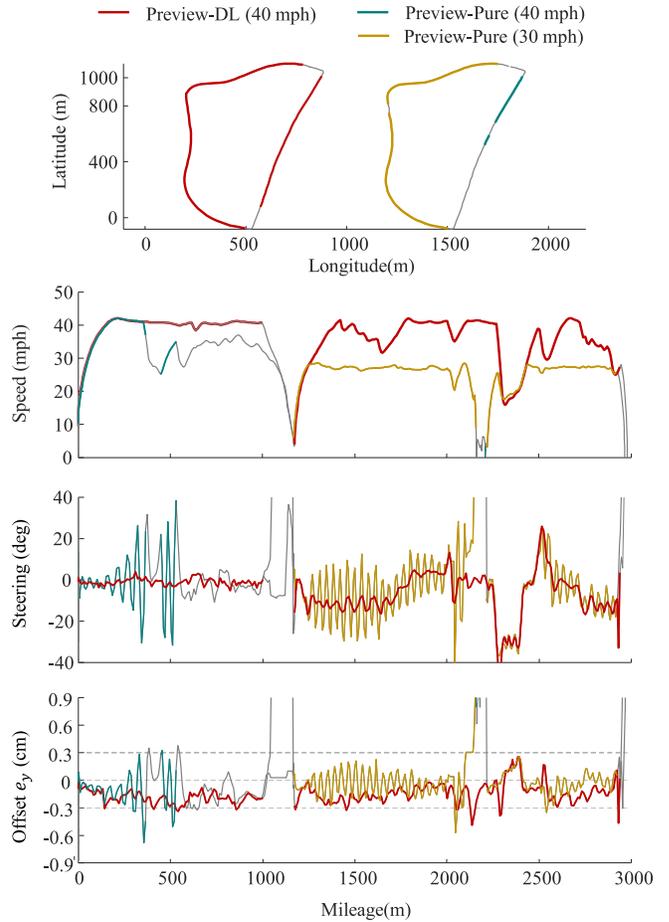


Fig. 22. The testing results of lane-keeping control on an open road using the Preview-DL and Preview-Pure controls.

Fig. 22 shows the testing results. The grey lines mean there is no lane marker, thus the automatic control disengaged. Under the same cost function design, setting the target speed to 40 mph (or 64.37 km/h), the Preview-Pure control oscillated at the beginning stage, with increasingly oscillatory steering angles and offset, and finally failed to track the lane. Then we decreased the target speed to 30 mph. The control did not fail but is still less stable, suffering from the obvious oscillations. As a comparison, the proposed Preview-DL control considering delay and lag achieved very stable control at 40 mph over the whole trip with smooth steering operation, refer to the supplementary videos:

- 1) Video1_Lane keeping control considering delay and lag.avi;
- 2) Video2_Lane keeping control ignoring delay and lag.avi.

VII. CONCLUSION

In this paper, a preview steering control considering input delay, steering system lag, and future road curvatures is designed for accurate, smooth, and computationally efficient path tracking. The main findings include:

- 1) The input delay can be handled by the augmentation strategy and the predictor strategy. The former includes the delayed commands in the enhanced state vector and forms an augmented delay-free problem. The outcome directly utilized the original optimal control theory. The latter is an empirical

rule to compensate for the delay but loses the optimality of control.

2) The steering lag is also a crucial factor that affects tracking performance. It is approximated by a first-order lag system, which reduces model mismatch and improves tracking performance.

3) When the delay and lag are approximately known, the proposed control can expand the system stability margin to a much higher level. Namely, for a given delay system, it also allows for higher feedback gains, lower tracking error, and higher vehicle speed.

4) The experiments validated the benefits of the proposed controller, *i.e.*, smoother steering and better accuracy in the RTK-based path tracking test; better stability at higher speed and smoother steering in the Mobileye-based lane keeping test.

REFERENCES

- [1] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Trans. Intell. Vehicles*, vol. 1, no. 1, pp. 33–55, Mar. 2016.
- [2] S. Chaib, M. S. Netto, and S. Mammar, "H_∞ adaptive, PID and fuzzy control: A comparison of controllers for vehicle lane keeping," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2004, pp. 139–144.
- [3] E. Kayacan, H. Ramon, and W. Saeys, "Robust trajectory tracking error model-based predictive control for unmanned ground vehicles," *IEEE/ASME Trans. Mechatronics*, vol. 21, no. 2, pp. 806–814, Apr. 2016.
- [4] H. Guo, J. Liu, D. Cao, H. Chen, R. Yu, and C. Lv, "Dual-envelop-oriented moving horizon path tracking control for fully automated vehicles," *Mechatronics*, vol. 50, pp. 422–433, Apr. 2018.
- [5] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "Predictive active steering control for autonomous vehicle systems," *IEEE Trans. Control Syst. Technol.*, vol. 15, no. 3, pp. 566–580, May 2007.
- [6] Y. Koubaa, M. Boukattaya, and T. Dammak, "Adaptive sliding-mode dynamic control for path tracking of nonholonomic wheeled mobile robot," *J. Autom. Syst. Eng.*, vol. 9, no. 2, pp. 119–131, 2015.
- [7] A. Badawy, J. Zuraski, F. Bolourchi, and A. Chandy, "Modeling and analysis of an electric power steering system," SAE Tech. Paper 1999-01-0399, 1999.
- [8] K. Gu, J. Chen, and V. L. Kharitonov, *Stability of Time-Delay Systems*. Boston, MA, USA: Springer, 2003.
- [9] S. Xu, H. Peng, Z. Song, K. Chen, and Y. Tang, "Design and test of speed tracking control for the self-driving Lincoln MKZ platform," *IEEE Trans. Intell. Veh.*, to be published, doi: [10.1109/TIV.2019.2955908](https://doi.org/10.1109/TIV.2019.2955908).
- [10] S. Yu, X. Li, H. Chen, and F. Allgöwer, "Nonlinear model predictive control for path following problems," *Int. J. Robust Nonlinear Control*, vol. 25, no. 8, pp. 1168–1182, May 2015.
- [11] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, Dec. 2014.
- [12] R. Marino, S. Scalzi, and M. Netto, "Nested PID steering control for lane keeping in autonomous vehicles," *Control Eng. Pract.*, vol. 19, no. 12, pp. 1459–1467, Dec. 2011.
- [13] M. Samuel, M. Hussein, and M. Binti, "A review of some pure-pursuit based path tracking techniques for control of autonomous vehicle," *Int. J. Comput. Appl.*, vol. 135, no. 1, pp. 35–38, 2016.
- [14] E. J. Rossetter and J. C. Gerdes, "Lyapunov based performance guarantees for the potential field lane-keeping assistance system," *J. Dyn. Syst., Meas., Control*, vol. 128, no. 3, pp. 510–522, Sep. 2006.
- [15] S. Suryanarayanan, M. Tomizuka, and T. Suzuki, "Design of simultaneously stabilizing controllers and its application to fault-tolerant lane-keeping controller design for automated vehicles," *IEEE Trans. Control Syst. Technol.*, vol. 12, no. 3, pp. 329–339, May 2004.
- [16] S. Xu, H. Peng, P. Lu, M. Zhu, and Y. Tang, "Design and experiments of safeguard protected preview lane keeping control for autonomous vehicles," *IEEE Access*, vol. 8, pp. 29944–29953, 2020.
- [17] M. S. Mahmoud, *Robust Control and Filtering for Time-Delay Systems*. Boca Raton, FL, USA: CRC Press, 2000.
- [18] T. Fujinaka and M. Araki, "Discrete-time optimal control of systems with unilateral time-delays," *Automatica*, vol. 23, no. 6, pp. 763–765, Nov. 1987.
- [19] J. Chu, "Application of a discrete optimal tracking controller to an industrial electric heater with pure delays," *J. Process Control*, vol. 5, no. 1, pp. 3–8, Jan. 1995.
- [20] A. Nahidi, A. Khajepour, A. Kasaeizadeh, S.-K. Chen, and B. Litkouhi, "A study on actuator delay compensation using predictive control technique with experimental verification," *Mechatronics*, vol. 57, pp. 140–149, Feb. 2019.
- [21] S. Xu and H. Peng, "Design and comparison of fuel-saving speed planning algorithms for automated vehicles," *IEEE Access*, vol. 6, pp. 9070–9080, 2018.
- [22] M. Tomizuka, "The optimal finite preview problem and its application to Mari-machine systems," Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, MA, USA, 1973.
- [23] N. Birla and A. Swarup, "Optimal preview control: A review," *Optim. Control Appl. Methods*, vol. 36, no. 2, pp. 241–268, Mar. 2015.
- [24] A. T. Salton, Z. Chen, J. Zheng, and M. Fu, "Preview control of dual-stage actuator systems for superfast transition time," *IEEE/ASME Trans. Mechatronics*, vol. 16, no. 4, pp. 758–763, Aug. 2011.
- [25] H. Peng and M. Tomizuka, "Preview control for vehicle lateral guidance in highway automation," *J. Dyn. Syst., Meas., Control*, vol. 115, no. 4, pp. 679–686, Dec. 1993.
- [26] S. Xu and H. Peng, "Design, analysis, and experiments of preview path tracking control for autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 1, pp. 48–58, Jan. 2020.
- [27] T. Raharijaona, G. Duc, and S. Mammar, "H_∞ controller synthesis and analysis with application to lateral driving assistance," in *Proc. IFAC Symp. Adv. Automot. Control*, Salerno, Italy, 2004, pp. 1–6.
- [28] S. Xu, S. E. Li, K. Deng, S. Li, and B. Cheng, "A unified pseudospectral computational framework for optimal control of road vehicles," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 4, pp. 1499–1510, Aug. 2015.



Shaobing Xu received the Ph.D. degree in mechanical engineering from Tsinghua University, Beijing, China, in 2016.

He is currently an Assistant Research Scientist with the Department of Mechanical Engineering and Mcity, University of Michigan, Ann Arbor, MI, USA. His research interests include vehicle motion control, decision making, and path planning for autonomous vehicles. He was a recipient of the outstanding Ph.D. Dissertation Award of Tsinghua University, the Best Paper Award of AVEC'2018, the First Prize of the Chinese 4th Mechanical Design Contest, and the First Prize of the 19th Advanced Mathematical Contest.



Hui Peng received the Ph.D. degree in mechanical engineering from the University of California, Berkeley, in 1992.

He is currently a Professor with the Department of Mechanical Engineering, University of Michigan, and the Director of Mcity. His research interests include adaptive control and optimal control, with emphasis on their applications to vehicular and transportation systems. His current research focuses include design and control of electrified vehicles, and connected/automated vehicles. He is a fellow

of an SAE and an ASME. He is also a Changjiang Scholar with Tsinghua University, China.



Yifan Tang (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from The Ohio State University, Columbus, OH, USA, in 1994.

From 2005 to 2007, he was a Senior Electronics Engineer with Trimble Navigation Ltd. From 2007 to 2012, he was a Principal Motor Technologist with Tesla Motors. From 2012 to 2015, he was a Vice President of drivetrain engineering with Lucid Motors. From 2015 to 2017, he was a Technical Lead of power electronics systems with Facebook. Since 2017, he has been the Chief Technology Officer with SF Motors Inc., Santa Clara, CA, USA. His current research interests include electric vehicles and autonomous vehicles.

Dr. Tang's awards include an IEEE Industry Application Society Annual Meeting Second Prize Paper Award in 1998 and an IBM Outstanding Technical Achievement Award in 2002.