

# Design and Test of Speed Tracking Control for the Self-Driving Lincoln MKZ Platform

Shaobing Xu, Hui Peng, Ziyong Song, Kailiang Chen, Yifan Tang

**Abstract**— To accelerate self-driving technology developments, testing platforms modified from production passenger cars are widely used nowadays, but the powertrain models and control access to engine/motor are often not available, e.g., the popular hybrid Lincoln MKZ platform only opens brake/throttle pedal control protocol for longitudinal motion. These limitations motivate us to model the powertrain dynamics by experiments and to design new speed controls, differing from the conventional cruise control that manipulates engine/brake torque directly with accurate models. This paper explores the powertrain modeling and speed control algorithm design of these testing platforms, by taking the MKZ as an example. Two computationally-efficient algorithms are designed, i.e., a PID+C control when future information (i.e., reference speed and road slope profiles) is unavailable, and a preview control if the information is provided a priori. To guarantee speed tracking errors within a given safe range, a barrier control is also designed to supervise the two controllers, which adds additional brake/throttle actions when the error is approaching the boundary. The algorithms are implemented and tested on the Mcity MKZ platform, experimental results show the speed-tracking performance and the bounded errors achieved by the barrier function.

**Index Terms**—Autonomous vehicles, motion control, preview control, vehicle dynamics control

## I. INTRODUCTION

### A. Motivation

Connected and automated vehicle (CAV) is emerging to improve the traffic and mobility system. Lots of companies, startups, and research teams are dedicated to developing safe, intelligent, and robust CAV technologies [1][2]. During the developments, vehicle platforms with sensors, computing units, and by-wire control devices are usually required for pre-deployment and tests. Some companies like Waymo use platforms from their OEM partner, while many other teams select platforms that are modified from production passenger cars, e.g., the widely-used Lincoln MKZ and Fusion platform from Dataspeed [3], or the Kia platform from Polysync.

One limitation of these modified platforms is that the powertrain dynamics model (i.e., engine, motor, and brake) is often not available, and the by-wire control access is also

limited. This paper takes the hybrid Lincoln MKZ as an example [3]. It is one of the most widely used research platforms nowadays; companies including Baidu, Nvidia, and Intel, and hundreds of research teams are leveraging it to develop or test CAV technologies. However, this platform only opens the control protocol of brake/throttle pedals for longitudinal motion control, with no access to engine/motor torque or adaptive cruise control (ACC) system. Moreover, the powertrain is a black box for developers; no dynamics model is provided. These limitations prevent model-based motion control design for better performance; instead, model-free control such as PID design and pure pursuit control are often selected [4].

Under this background, we want to identify the system models and develop high-performance motion control algorithms for this self-driving platform, in order to provide potential references to other teams. For the lateral vehicle dynamics model and the path-tracking control algorithm, refer to our previous paper [5]. This paper concentrates on longitudinal dynamics models and developing speed tracking algorithms for the MKZ platform. With the fact that speed tracking control is a well-studied topic in the automotive industry, e.g., the cruise control (CC) and ACC systems have been available on cars for over 35 and 20 years [6]-[9], two main differences are highlighted below to distinguish this work from them:

1) Conventional CC and ACC systems usually control engine/brake torque directly using accurate powertrain models. However, for the MKZ platform with a hybrid powertrain, we do not have powertrain dynamics models and can control brake and throttle pedals only. Thus, the models between pedal opening and powertrain outputs will be identified and considered in speed tracking control design.

2) The CC and ACC usually use instantaneous vehicle states and target speed for highway driving; namely, no future information is used [7][8]. This is reasonable due to the future information, i.e., reference speed and road slope profiles, is often not available. In this paper, we also present a controller for this case. However, sometimes CAVs are able to obtain future slope from maps and future target speed profile from planning modules, e.g., driverless shuttles operating on fixed-route, or CAVs cruising in very sparse traffic, like the CC system [10][11]. In this case, we focus on developing speed tracking algorithms that can leverage future information for better performance.

### B. Related Work on Speed Tracking Control

Vehicle speed tracking is a classical dynamic control problem [12]-[19]. Here the existing speed tracking methods

This research was supported by the UofM-SFmotors automated driving project and the OpenCAV project of Mcity.

S. Xu and H. Peng are with the Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 48109 USA (Email: xushao@umich.edu; hpeng@umich.edu)

Z. Song is with the Department of Naval Architecture and Marine Engineering, University of Michigan, Ann Arbor, MI 48109 USA. (Email: ziyong.songthu@gmail.com)

K. Chen and Y. Tang are with the SF Motors Inc., Santa Clara, CA 95054 USA. (Email: kailiang.chen@sfmotors.com; yifan.tang@sfmotors.com)

are classified, very roughly, into three categories. The first category considers neither vehicle dynamics nor future information. The control commands are generated from tracking errors only, *e.g.*, the PID design [7][8][15]. The second type usually utilizes more accurate powertrain models, sometimes even including engine and transmission transient behaviors, for more accurate and/or smoother speed tracking. Most controllers fall into this category, *e.g.*, multiple-surface sliding mode control, Lyapunov design, and adaptive control [13], [16]-[18].

The third category utilizes not only powertrain models but also future information to further improve speed tracking. One classic framework is the model predictive control (MPC) [12][19][20], which minimizes a cost function over a finite receding horizon and optimizes the optimal command sequence by repeated online optimization [12]. It can fully utilize future information (*e.g.*, road slope or target speed) and achieve near-optimal solutions. But for nonlinear systems it often suffers from higher computing load. Another method is the preview control theory, proposed in the 1970s [21]. It targets linear systems only and cannot handle nonlinearities. Its main advantage is that it can directly respond to future information, not relying on online numerical optimization. It has been applied to various applications. For example, Shimmyo *et al.* proposed a preview controller for biped walking pattern generation of bipedal robots [23]. 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 this paper, we will also leverage the preview control theory for speed tracking design when future information is available.

### C. Contributions

This paper explores the longitudinal vehicle dynamics model and designs speed tracking control algorithms for the widely used CAV testing platform, the hybrid Lincoln MKZ, in order to facilitate motion control developments based on this or other similar platforms. In detail, 1) the longitudinal dynamic model is identified by experiments; 2) two computationally-efficient speed tracking algorithms are designed, *i.e.*, a discrete-time preview control if the future information is available, and a PID+C control using instantaneous information only. The former utilizes more information but is able to achieve smoother and accurate tracking; 3) To limit tracking errors, a barrier function is developed to supervise both the preview control and the PID+C for bounded errors; and 4) system responses in both the time and frequency domain are analyzed. The algorithms are finally implemented and tested on the Mcity MKZ platform to validate their control performances.

The remainder of this paper is organized as follows: Section II presents the vehicle dynamics model; Section III designs the preview control, the PID+C control, and the barrier function; the analysis of control performance is presented in Section IV; Section V describes the experiments and results. Finally, Section VI concludes this paper. Part of the work was presented in an earlier conference version [26].

## II. VEHICLE LONGITUDINAL DYNAMICS MODELING

The studied hybrid Lincoln MKZ (2017) platform is shown in Fig. 1. As mentioned earlier, its propulsion system (*i.e.*, engine and motor) is a black box for developers, and only brake and throttle pedals can be controlled, with no access to the engine, motor, or torque control module.



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

Without system models, a typical method is the PID design that ignores powertrain nonlinearity and uses tracking error only to generate brake/throttle commands. Here we turn to identify the system model and then leverage it for algorithm design. When the CAV tracks a desired speed trajectory  $v_d$  on varying slope  $\vartheta_r$ , the motion is described by

$$\dot{v} = a = \kappa\Psi(v, \phi_t, \phi_b) - a_r(v) - g\sin\vartheta_r \quad (1)$$

where  $a$  and  $v$  denote the vehicle acceleration and speed;  $\phi_t$  and  $\phi_b$  are the throttle and brake pedal openings, respectively;  $a_r$  is the load of aerodynamic drag and rolling resistance, approximated by a 2<sup>nd</sup>-order polynomial:

$$a_r(v) = \alpha_0 + \alpha_1v + \alpha_2v^2 \quad (2)$$

The term  $\Psi(\cdot)$  stands for the nonlinear powertrain dynamics, a function of  $v$ ,  $\phi_t$  and  $\phi_b$ . Its output is the torque transmitted to the tires, which is then converted to acceleration with a lumped coefficient  $\kappa$ . If  $\vartheta_r = 0$ ,  $\kappa\Psi - a_r$  equals vehicles acceleration, denoted by

$$a_\psi = \kappa\Psi(\cdot) - a_r(\cdot) \quad (3)$$

Definitely,  $\Psi$  is highly related to engine and motor dynamics, but the model is not available. As a tradeoff, we first identify the static mapping from pedal opening to the acceleration  $a_\psi$ , and then use a first-order linear system to approximate the transient dynamics.

To identify the static mapping, on-track experiments are designed and conducted: control the car with fixed throttle pedal opening at different levels varying from 15% to 60% to accelerate the car from 0 m/s; the resulted speed and acceleration are recorded and used to generate the speed-pedal-acceleration mapping, as shown in Fig. 2 (a)<sup>1</sup>. The static mapping of brake, provided by the platform supplier, is shown in Fig. 2 (b). These maps are denoted by  $\mathbb{M}$  in the following.

Since fixed pedal openings are applied when creating the maps, transient dynamics is thus ignored, which is difficult to model without any information on the hybrid powertrain. Here we use a first-order linear system to approximate it,

<sup>1</sup> If readers are interested in the longitudinal and lateral vehicle dynamics models of the Lincoln MKZ platform, please email us for a copy.

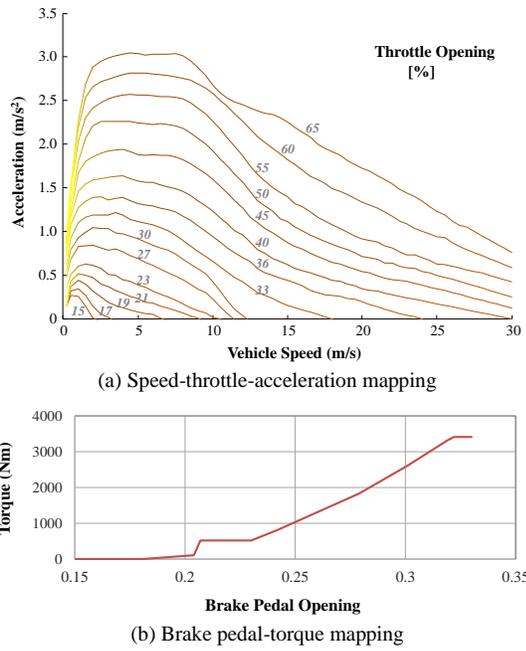


Fig. 2. Models of the vehicle traction and brake systems.

$$\tau \dot{u} = -u + u_c \quad (4)$$

where  $u_c$  is the acceleration command,  $u$  is called the effective acceleration command, and  $\tau$  is the time constant. Given  $a_\psi = u_c$ , the brake/throttle pedal opening commands are finally generated by querying the maps  $\mathbb{M}$ .

In summary, the longitudinal dynamics for speed tracking design is modeled as

$$\begin{aligned} \dot{x} &= \begin{bmatrix} 0 & 1 \\ 0 & -1/\tau \end{bmatrix} x + \begin{bmatrix} 0 \\ 1/\tau \end{bmatrix} u_c + \begin{bmatrix} -1 \\ 0 \end{bmatrix} \vartheta \\ &= \mathcal{A}_o x + \mathcal{B}_o u_c + \mathcal{D}_o \vartheta \\ v &= \mathcal{C}_o x(k) \\ (\phi_t, \phi_b) &= \mathbb{M}(v, u_c) \end{aligned} \quad (5)$$

where  $x = (v, u)^T$  is the state vector,  $u_c$  is the control input,  $\vartheta = \varrho \sin \vartheta_r$  is regarded as the system disturbance, and  $\mathcal{C}_o = [1 \ 0]$ . We emphasize that this model and the static maps are not absolutely accurate, but help to reduce model mismatch for control design.

To facilitate controller design, the continuous-time model (5) is converted into a discrete-time system with a fixed sampling period  $\Delta\tau$  and the zero-order holder (ZOH):

$$\begin{aligned} x(k+1) &= \bar{\mathcal{A}}x(k) + \bar{\mathcal{B}}u_c(k) + \bar{\mathcal{D}}\vartheta(k) \\ v(k) &= \bar{\mathcal{C}}x(k) \end{aligned} \quad (6)$$

where  $\bar{\mathcal{A}} \in \mathbb{R}^{2 \times 2}$ ,  $\bar{\mathcal{B}}$  and  $\bar{\mathcal{D}} \in \mathbb{R}^2$ ,  $\bar{\mathcal{C}} = \mathcal{C}_o \in \mathbb{R}^{1 \times 2}$ .

### III. SPEED-TRACKING CONTROL DESIGN

#### A. Formulation of Speed Tracking Problem

Leveraging the system models, in this section we design an optimal speed-tracking control considering future target speed  $v_d$  and road slope  $\vartheta$ . If the future information is unknown, the problem becomes a special case of the optimal control. When tracking a speed trajectory  $v_d$ , two features of  $v_d$  need special attention, *i.e.*, it may be noisy and subject to step-change, as

shown in Fig. 3. In general, both smooth brake/throttle operations when tracking error is small and fast responses when  $v_d$  changes suddenly are required in real applications.

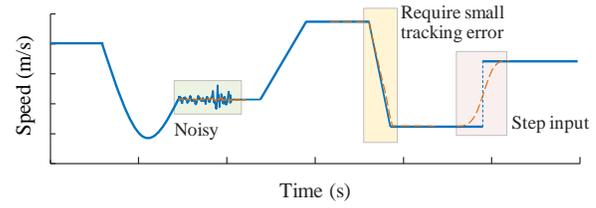


Fig. 3. Schematic diagram of input speed.

To balance the two requirements, the weighted sum of speed error  $e_v$  and increment of control input  $\Delta u_c$  is designed as cost function,

$$\mathcal{J} = \frac{1}{2} \sum_{k=0}^{\infty} q e_v^2(k) + r \Delta u_c^2(k) \quad (7)$$

where  $q, r \in \mathbb{R}$  are positive weights.  $e_v(k)$  and  $\Delta u_c(k)$  are defined by

$$\begin{aligned} e_v(k) &= v(k) - v_d(k) \\ \Delta u_c(k) &= u_c(k) - u_c(k-1) \end{aligned} \quad (8)$$

To avoid slow responses and unsafe control, a hard constraint is imposed on  $e_v$ ,

$$e_{\max} - |e_v| \geq 0 \quad (9)$$

where  $e_{\max} > 0$  is the acceptable maximum error.

As  $e_v$  becomes the key element in the problem definition, here we introduce it as a new state, with dynamics being

$$\begin{aligned} e_v(k+1) &= v(k+1) - v_d(k+1) \\ &= \bar{\mathcal{C}}\Delta x(k+1) + e_v(k) - \Delta v_d(k+1) \end{aligned} \quad (10)$$

$$\Delta x(k+1) = \bar{\mathcal{A}}\Delta x(k) + \bar{\mathcal{B}}\Delta u_c(k) + \bar{\mathcal{D}}\Delta \vartheta(k)$$

Then the speed tracking problem is formulated as

$$\mathcal{J} = \frac{1}{2} \sum_{k=0}^{\infty} \mathcal{X}^T(k) \mathcal{Q} \mathcal{X}(k) + \Delta u_c^T(k) \mathcal{R} \Delta u_c(k) \quad (11-a)$$

*s.t.*

$$\begin{aligned} \mathcal{X}(k+1) &= \mathcal{A}\mathcal{X}(k) + \mathcal{B}\Delta u_c(k) + \mathcal{D}\Delta \vartheta(k) \\ &\quad + \mathcal{E}\Delta v_d(k+1) \end{aligned} \quad (11-b)$$

$$e_{\max} - |e_v| \geq 0 \quad (11-c)$$

$$\mathcal{X}(k) = \begin{bmatrix} e_v(k) \\ \Delta x(k) \end{bmatrix}, \quad \mathcal{A} = \begin{bmatrix} 1 & \bar{\mathcal{C}}\bar{\mathcal{A}} \\ \mathbf{0}_{2 \times 1} & \bar{\mathcal{A}}_{2 \times 2} \end{bmatrix},$$

$$\mathcal{B} = \begin{bmatrix} \bar{\mathcal{C}}\bar{\mathcal{B}} \\ \bar{\mathcal{B}} \end{bmatrix}, \quad \mathcal{D} = \begin{bmatrix} \bar{\mathcal{C}}\bar{\mathcal{D}} \\ \bar{\mathcal{D}} \end{bmatrix}, \quad \mathcal{E} = \begin{bmatrix} -1 \\ \mathbf{0}_{2 \times 1} \end{bmatrix},$$

$$\mathcal{Q} = \begin{bmatrix} q & \mathbf{0}_{1 \times 2} \\ \mathbf{0}_{2 \times 1} & \mathbf{0}_{2 \times 2} \end{bmatrix}, \quad \mathcal{R} = r \quad (11)$$

where  $O/I$  stands for the zero/identity matrix,  $\mathcal{Q}$  is a positive semi-definite matrix.

The problem formulation requires knowledge of  $\Delta \vartheta$  and  $\Delta v_d$  in the infinite horizon. However, a more sensible approach is to

use  $\Delta\vartheta(k)$  and  $\Delta v_d(k+1)$  only in a finite horizon, denoted by  $[k, k + N_\vartheta - 1]$  and  $[k + 1, k + N_v]$ , respectively, where  $N_\vartheta$  and  $N_v$  are the preview steps.  $\vartheta(k)$  and  $v_d(k+1)$  beyond the preview horizon are assumed to remain constant, *i.e.*,

$$\begin{aligned} \Delta\vartheta(k+i) &= 0, i \in [N_\vartheta, \infty) \\ \Delta v_d(k+i) &= 0, i \in [N_v + 1, \infty) \end{aligned} \quad (12)$$

This strategy works because  $\Delta\vartheta$  and  $\Delta v_d$  far in the future have little influence on the current control.

### B. Strategy of Solving the Speed Tracking Problem

The formulated problem (11) involving the state constraint and future  $\vartheta$  and  $v_d$  is a typical constrained nonlinear optimal control problem (OCP). If ignoring the computation load, the MPC will be the best method, refer to [12][19]. However, this paper pursues computationally efficient algorithm to simplify online implementation, thus we employ the preview control theory to design the speed tracking controller [21]-[25].

If the disturbances  $\Delta\vartheta(k)$  and  $\Delta v_d(k+1)$  are zero and the constraint on  $|e_v|$  is removed, the problem becomes a standard linear quadratic regulator (LQR) which can be solved analytically. However, the time-varying disturbances are too strong to ignore, and the constraint (9) on system state further strengthens the challenge. To solve the problem, our strategy is to split the original problem into two sub-problems: one ignores the state constraint first, *i.e.*, Eqs. (11-a) and (11-b); the other considers the constraint only, *i.e.*, Eqs. (11-c) and (11-b). The former pursues smooth speed tracking, and the latter guarantees safety performance. They are solved separately. Note that this strategy cannot guarantee global optimality of the solution, but compromises between optimality and computing efficiency.

For the first sub-problem, a preview control is designed to achieve optimal tracking considering future information, as presented in section III.C. If the future information is unavailable, a PID+C control is designed in section III.D. For the second sub-problem, a model-based barrier function is proposed to satisfy the constraint (9) in section III.E.

### C. Design of Preview Control Algorithm

The fundamental of preview speed-tracking control is to reformulate the first subproblem as an augmented LQR[21], solving which yields the optimal solution.

#### 1) Augmented Optimal Control System

To convert the problem, we transfer the system disturbances within the preview window, *i.e.*,  $\Delta\vartheta$  in  $[k, k + N_\vartheta - 1]$  and  $\Delta v_d$  in  $[k + 1, k + N_v]$ , to the system state vector  $\mathcal{X}(k)$ . Note that the two independent disturbances  $\Delta\vartheta$  and  $\Delta v_d$  are in a symmetrical form. To simplify the presentation, only  $\Delta v_d$  is considered in the following design but its result will be extended to  $\Delta\vartheta$  directly. The augmented state vector  $\mathbb{X}(k)$  is

$$\mathbb{X}(k) = \begin{bmatrix} \mathcal{X}(k) \\ \Delta\mathbb{V}(k) \end{bmatrix} \in \mathbb{R}^{3+N_v} \quad (13)$$

$$\Delta\mathbb{V}(k) = [\Delta v_d(k+1), \dots, \Delta v_d(k+N_v)]^T$$

Then the cost function and dynamics are augmented to [21]

$$J = \frac{1}{2} \sum_{k=0}^{\infty} \mathbb{X}^T(k) \mathbb{Q} \mathbb{X}(k) + \Delta u_c^T(k) \mathbb{R} \Delta u_c(k) \quad (14)$$

*s.t.*

$$\mathbb{X}(k+1) = \mathbb{A} \mathbb{X}(k) + \mathbb{B} \Delta u_c(k)$$

where  $\mathbb{Q}$ ,  $\mathbb{R}$ ,  $\mathbb{A}$ , and  $\mathbb{B}$  are the augmented matrices, defined as

$$\begin{aligned} \mathbb{Q} &= \begin{bmatrix} \mathcal{Q}_{3 \times 3} & \mathcal{O}_{3 \times N_v} \\ \mathcal{O}_{N_v \times 3} & \mathcal{O} \end{bmatrix}, \mathbb{R} \equiv \mathcal{R}, \\ \mathbb{A} &= \begin{bmatrix} \mathcal{A}_{3 \times 3} & \mathbb{E}_{3 \times N_v} \\ \mathcal{O}_{N_v \times 3} & \mathcal{L}_{N_v \times N_v} \end{bmatrix}, \mathbb{E} = [\mathcal{E}_3, \mathcal{O}_{3 \times (N_v-1)}], \\ \mathcal{L} &= \begin{bmatrix} \mathcal{O}_{(N_v-1)} & I_{(N_v-1) \times (N_v-1)} \\ 0 & \mathcal{O}_{1 \times (N_v-1)} \end{bmatrix}, \mathbb{B} = \begin{bmatrix} \mathcal{B}_{3 \times 1} \\ \mathcal{O}_{N_v \times 1} \end{bmatrix} \end{aligned} \quad (15)$$

where  $\mathcal{L}$  describes the mapping of the previewed  $\Delta v_d$ .

#### 2) Preview Speed Control Algorithm

The system (14) is essentially an augmented time-invariant LQR. Here we directly present its optimal control law [22]

$$\begin{aligned} \Delta u_c^*(k) &= -(\mathbb{R} + \mathbb{B}^T \mathbb{P} \mathbb{B})^{-1} \mathbb{B}^T \mathbb{P} \mathbb{A} \mathbb{X}(k) \\ &= -K \mathbb{X}(k) \end{aligned} \quad (16)$$

$$\mathbb{X}(k+1) = (I + \mathbb{B} \mathbb{R}^{-1} \mathbb{B}^T \mathbb{P})^{-1} \mathbb{A} \mathbb{X}(k) = \beta \mathbb{A} \mathbb{X}(k)$$

where  $K \in \mathbb{R}^{N_v+3}$  is the feedback gain vector,  $\beta$  is the lumped matrix, and  $\mathbb{P}$  is solved from the Riccati equation,

$$\mathbb{P} = \mathbb{Q} + \mathbb{A}^T \beta^T \mathbb{P} \mathbb{A} \quad (17)$$

Eqs. (16) and (17) deliver the optimal control of the proposed augmented system. Note that Eq. (17) is a high-dimensional (*i.e.*,  $N_v + 3$ ) Riccati equation. To streamline the control law, in the following we focus on decoupling the original state  $\mathcal{X}(k)$  and the augmented state  $\Delta\mathbb{V}(k)$ . First, the matrix  $\mathbb{P}$  is partitioned into four sub-matrices:

$$\mathbb{P} = \begin{bmatrix} \mathcal{P} & \mathcal{P}_v \\ \mathcal{P}_v^T & \mathcal{P}_{22} \end{bmatrix} \quad (18)$$

Then Eq. (17) can be simplified to

$$\begin{bmatrix} \mathcal{P} & \mathcal{P}_v \\ - & - \end{bmatrix} = \begin{bmatrix} \mathcal{Q} + \zeta^T \mathcal{P} \mathcal{A} & \zeta^T (\mathcal{P} \mathbb{E} + \mathcal{P}_v \mathcal{L}) \\ - & - \end{bmatrix} \quad (19)$$

where  $\zeta = \mathcal{A}^T (I + \mathcal{P} \mathcal{B} \mathcal{R}^{-1} \mathcal{B}^T)^{-1}$ . Based on Eq. (19), we can solve  $\mathcal{P}$  by

$$\mathcal{P} = \mathcal{Q} + \zeta^T \mathcal{P} \mathcal{A} \quad (20)$$

and  $\mathcal{P}_v$  by

$$\mathcal{P}_v = \zeta^T (\mathcal{P} \mathbb{E} + \mathcal{P}_v \mathcal{L}) \quad (21)$$

Eq. (20) holds the same form as Eq. (17); it is actually the Riccati equation of the original system (11) without disturbance  $\Delta\vartheta$  and  $\Delta v_d$ . Considering the special structure of  $\mathbb{E}$  and  $\mathcal{L}$  in Eq. (15), *e.g.*, only the first column of  $\mathbb{E}$  is non-zero, we partition the matrix  $\mathcal{P}_v$  into  $N_v$  sub-column-vectors, denoted by  $\mathcal{P}_i$ . Then we solve

$$\mathcal{P}_i = \zeta \mathcal{P}_{i-1} = \zeta^i \mathcal{P} \mathcal{E}, \quad i \in [1, N_v] \quad (22)$$

Substituting Eqs. (20) and (22) into Eq. (16) yields the full-state feedback control law:

$$\begin{aligned} \Delta u_c^*(k) &= -K_s \mathcal{X}(k) - K_v \Delta\mathbb{V}(k) \\ K_s &= (\mathcal{R} + \mathcal{B}^T \mathcal{P} \mathcal{B})^{-1} \mathcal{B}^T \mathcal{P} \mathcal{A} \end{aligned} \quad (23)$$

$$K_v = (\mathcal{R} + \mathcal{B}^T \mathcal{P} \mathcal{B})^{-1} \mathcal{B}^T (\mathcal{P} \mathcal{E} + \mathcal{P}_c \mathcal{L})$$

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

where  $K_s \in \mathbb{R}^3$ ,  $K_v \in \mathbb{R}^{N_v}$ . Since  $\Delta\vartheta$  and  $\Delta v_d$  have the symmetrical form, here we extend the feedback rule (23) to include  $\Delta\vartheta$  in  $[k, k + N_\vartheta]$ :

$$\begin{aligned} \Delta u_c^*(k) &= -K_s \mathcal{X}(k) - K_v \Delta \mathbb{V}(k) - K_\vartheta \Delta \Theta(k) \\ \Delta \Theta(k) &= [\Delta\vartheta(k), \dots, \Delta\vartheta(k + N_\vartheta)]^T \\ K_{\vartheta,j} &= (\mathcal{R} + \mathcal{B}^T \mathcal{P} \mathcal{B})^{-1} \mathcal{B}^T \zeta^{j-1} \mathcal{P} \mathcal{D} \end{aligned} \quad (24)$$

where  $j \in [1, N_\vartheta]$ .

Finally, integrating Eq. (24) yields the analytical preview control law:

$$\begin{aligned} u_c^*(k) &= \underbrace{-K_{s,1} \sum_{i=0}^k e_v(i)}_{\text{Feedback}} - \underbrace{K_{s,2} v(k)}_{\text{Pseudo P}} - \underbrace{K_{s,3} u(k)}_{\text{D}} \\ &\quad - \underbrace{\sum_{i=1}^{N_v} K_v(i) v_d(k+i)}_{\text{Prev. } v_d} - \underbrace{\sum_{j=1}^{N_\vartheta} K_\vartheta(j) \vartheta(k+j-1)}_{\text{Prev. } \vartheta} + \mathbb{I}_o \end{aligned} \quad (25)$$

in which we assume  $u_c(-1) = 0$ ,  $v(-1) = 0$ , and  $u(-1) = 0$ . In Eq. (25), the control consists of five parts:

- feedback on the integral of tracking error  $e_v$ ;
- feedback on speed  $v$  (not  $e_v$ , marked as pseudo P);
- feedback on effective command  $u$ ;
- feedforward control of future target speed  $v_d$ ;
- feedforward control of future road slope  $\vartheta$ .

The first three parts are feedback controls of system states; the last two parts are called preview or feedforward actions since they respond to future signals. The last item  $\mathbb{I}_o$  is generated when integrating Eq. (24):

$$\mathbb{I}_o = \sum_{i=1}^{N_v} K_v(i) v_d(-1+i) + \sum_{j=1}^{N_\vartheta} K_\vartheta(j) \vartheta(-2+j) \quad (26)$$

It is related to  $v_d$  and  $\vartheta$  in the preview window at  $k = -1$ . Once  $u_c^*$  is known, the brake/throttle pedal opening commands are obtained from the map  $\mathbb{M}(v, u_c^*)$  shown in Fig. 2.

To better understand the feedforward gains  $K_v$  and  $K_\vartheta$ , their profiles are plotted in Fig. 4. Their absolute values decrease as the preview step increases, meaning that the effect of future disturbances becomes weaker and converges to zero. When  $q=1$  and  $r=0.1$ , the gains beyond 5 seconds are close to zero, implying 5s is a long-enough preview horizon. If pursuing smoother control, e.g., increasing  $r$  from 0.1 to 15, then the gains become lower and the suggested horizon increases from 5s to 12s.

#### D. Design of PID+C Control

If the future information becomes unavailable, a more concise control is designed here by degrading the preview control (25). A wrong idea is to set the preview steps  $N_v$  and  $N_\vartheta$  to 1 directly. In theory, the correct horizon of the optimal control problem is  $[k, \infty)$ , but we only consider  $N_v$  and  $N_\vartheta$

steps, because the gains beyond that are close to 0, as shown in Fig. 4. Setting  $N_v = N_\vartheta = 1$  will ignore too many valid gains, or namely, will lead to a solution that is optimal to track the trajectory  $v_d(k+i) = 0$ ,  $i > 1$ , with  $\vartheta(k+j-1) = 0$ ,  $j > 1$ , i.e.,  $v_d$  and  $\vartheta$  beyond one step are 0. Different from this idea, here we assume that the future target speed and road slope keep constant at the current values, rather than being 0, i.e.,

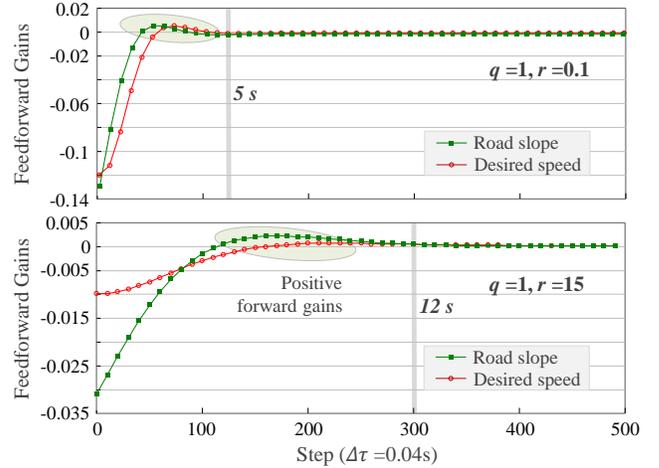


Fig. 4. Feedforward gains of the previewed road slope and desired speed.

$$\begin{aligned} v_d(k+1+i) &= v_d(k+1), i \in [0, \infty) \\ \vartheta(k+j) &= \vartheta(k), j \in [0, \infty) \end{aligned} \quad (27)$$

For this certain problem, the feedforward gains  $K_v$  and  $K_\vartheta$  coincidentally meet

$$\begin{aligned} \lim_{N_v \rightarrow \infty} \sum_{i=1}^{N_v} K_v(i) &= (\mathcal{R} + \mathcal{B}^T \mathcal{P} \mathcal{B})^{-1} \mathcal{B}^T (I - \zeta)^{-1} \mathcal{P} \mathcal{E} = -K_{s,2} \\ \lim_{N_\vartheta \rightarrow \infty} \sum_{i=1}^{N_\vartheta} K_\vartheta(i) &= -1 - K_{s,3} \end{aligned} \quad (28)$$

Therefore, the feedback of  $v$  and the feedforward of  $v_d$  merge into a proportional control  $-K_{s,2} e_v$ . The feedback of  $u$  and the feedforward of  $\vartheta$  are integrated into a derivative control with slope correction:

$$\begin{aligned} K_{s,3} u(k) + \lim_{N_\vartheta \rightarrow \infty} \sum_{j=1}^{N_\vartheta} K_\vartheta(j) \vartheta(k+j-1) \\ = K_{s,3} \dot{e}_v(k) - \vartheta(k) \end{aligned} \quad (29)$$

Then the preview control (25) degenerates into a PID control with road slope correction:

$$\begin{aligned} u_c^*(k) &= -K_{s,1} \sum_{i=0}^k e_v(i) - K_{s,2} e_v(k) \\ &\quad - K_{s,3} \dot{e}_v(k) + \vartheta(k) + a_d(k) \end{aligned} \quad (30)$$

$$(\phi_t, \phi_b) = \mathbb{M}(v, u_c^*)$$

Sometimes the desired acceleration  $a_d$  of target speed  $v_d$  is available, e.g., the Gipps and IDM car-following models output target acceleration directly. But in Eq. (27), the assumption  $v_d(k+i) = v_d(k)$  actually discards this information. Here we add a new correction  $a_d$  into Eq. (30) for more accurate tracking if  $a_d$  is available. The control (30) can be understood

as a PID control plus corrections to current road slope and desired acceleration, called PID<sup>+</sup>C in the following. It shares the same feedback gains as the preview control.

### E. Barrier Function for Tracking Error Constraint

For the second subproblem, we design a speed-error barrier control to satisfy the constraint (9). The concept of control barrier function (CBF) is proposed in [27], which can assure forward invariant system states. This concept inspires the development of speed-error barrier control in this paper.

To satisfy the constraint, we define a differentiable CBF,

$$h = e_{\max}^2 - e_v^2 \geq 0 \quad (31)$$

This inequality defines the safe error space  $\Omega = \{e_v | h > 0\}$  and its boundary  $\bar{\Omega} = \{e_v | h = 0\}$ .

The CBF acts similarly to the control Lyapunov function. It has a unique property, *i.e.*,  $h \rightarrow 0$  if  $x \rightarrow \bar{\Omega}$ , meaning zero energy on the boundary. If  $\Delta h$  is high enough when  $e_v \rightarrow \bar{\Omega}$ , then the system will stay inside  $\Omega$ . This idea is implemented by the following constraint,

$$\Delta h(x(k)) \geq -\gamma h(k) \quad (32)$$

where  $\gamma > 0$ . With Eq. (32),  $h$  can freely change when  $x(k)$  is far away from  $\bar{\Omega}$ ; while when  $e_v \rightarrow \bar{\Omega}$ ,  $\Delta h$  approaches zero and  $h$  stays to decrease. Considering the dynamics (6), we have

$$\begin{aligned} \Delta h &= -2e_v \Delta e_v = -2e_v (\Delta v - \Delta v_d) \\ &= -2e_v \bar{C}[(\bar{A} - I)x(k) + \bar{B}u_c(k) + \bar{D}\vartheta(k)] + 2e_v \Delta v_d \end{aligned} \quad (33)$$

Here  $\bar{C}\bar{B} \equiv 0$ , meaning that  $u_c(k)$  cannot affect  $\Delta h$  directly but determines  $u(k)$  and then affects  $\Delta h$  indirectly. Thus, we assume  $u_c = u$  but introduce a slack variable  $\varepsilon$  to the barrier function to compensate for the accuracy loss of the assumption. Then substituting Eq. (33) into Eq. (32) yields

$$2e_v u_c \leq \gamma(h - \varepsilon) + 2e_v (\Delta v_d + \vartheta) = \hat{\Phi} \quad (34)$$

Eq. (34) creates the boundary of command  $\bar{u}_c$ ,

$$\bar{u}_c(k) = 0.5 \hat{\Phi} / e_v \quad (35)$$

Having the output  $u_c^*$  of preview control or PID<sup>+</sup>C constrained by  $\bar{u}_c$  generates the final control,

$$u_c^*(k) = \begin{cases} \min(u_c^*, \bar{u}_c), & e_v > 0 \\ \max(u_c^*, \bar{u}_c), & e_v < 0 \\ u_c^*, & e_v = 0 \end{cases} \quad (36)$$

Eq. (36) shows the supervisor role of the barrier control. It was activated if and only if  $e_v$  is approaching  $e_{\max}$ . Fig. 5 shows the profiles of  $\bar{u}_c$  under different settings. Generally, a lower  $\gamma$  allows for earlier but weaker interventions and vice versa. Note that the barrier control uses instantaneous states only, its solution can satisfy the constraint but cannot guarantee global optimality, although the existing MPC solution can stay closer to the global optimality by horizon optimization. This strategy is proposed because we want to provide a different option that is computationally efficient for practical self-driving applications.

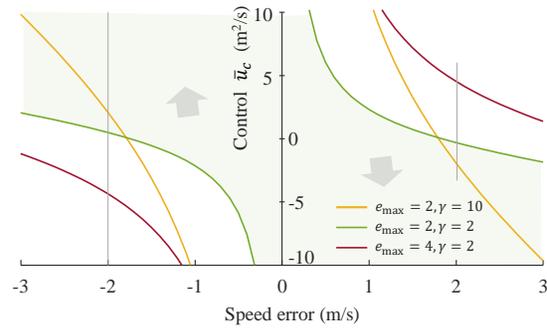


Fig. 5. The upper/lower bound of  $u_c$  calculated from the barrier function.

## IV. ANALYSIS OF CONTROL PERFORMANCE

### A. System Responses in the Frequency Domain

Substituting the preview control (23) into the system dynamics (14) yields

$$\mathcal{X}(k+1) = (\mathcal{A} - \mathcal{B}K_s)\mathcal{X}(k) + (\mathbb{E} - \mathcal{B}K_v)\Delta\mathcal{V}(k) \quad (37)$$

Applying the  $\mathcal{Z}$ -transformation generates the transfer function  $G_v(z)$  from  $\Delta\mathcal{V}(k)$  to the system states  $\mathcal{X}$ ,

$$G_v(z) = \frac{\mathcal{Z}(\mathcal{X})}{\mathcal{Z}(\Delta\mathcal{V})} = (zI - \mathcal{A} + \mathcal{B}K_s)^{-1}(\mathbb{E} - \mathcal{B}K_v)\mathbb{Z} \quad (38)$$

where  $\mathbb{Z} = [1, z, \dots, z^N]^T$ . If the future information is not available, then the transfer function becomes

$$G_v(z) = \frac{\mathcal{Z}(\mathcal{X})}{\mathcal{Z}(\Delta\mathcal{V})} = (zI - \mathcal{A} + \mathcal{B}K_s)^{-1}\mathcal{E} \quad (39)$$

The transfer function  $G_\vartheta(z)$  from the road slope change  $\Delta\Theta(k)$  to  $\mathcal{X}$  with and without future information is similar to Eq. (38) and (39). Their closed-loop system responses are presented in Fig. 6. We highlight the following observations:

- 1) The preview operations with future information can reduce speed tracking error  $e_v$ , but within a limited frequency range; beyond that, the controllers with and without future information respond similarly.
- 2) The future information helps to smooth tracking behaviors at high frequency, referring to  $\Delta v$  and  $\Delta u$ , which implies smoother brake/throttle operation and better ride comfort against fast-changing (e.g., noisy and step)  $\Delta v$  and  $\Delta u$ .
- 3) Increasing  $r$  from 0.1 to 15, tracking accuracy deteriorates, but tracking smoothness is improved, *i.e.*,  $\Delta v$ , and  $\Delta u$  are more inhibited.

### B. Tracking Performance in the Time Domain

Numerical simulations are used first to obtain more insights into the two controllers. In the simulation, the model (5) is used to approximate the real vehicle response. The results of tracking a given speed profile are shown in Fig. 7.

Both the preview control ( $r = 0.1$ ) and the PID<sup>+</sup>C can track the desired speed accurately. In Fig. 7, their maximal tracking error is about 0.7 m/s during a hard brake at 0.4g. The preview control acts before the desired speed changes, while the PID<sup>+</sup>C works only after the change. At  $t = 65s$ , the preview control accelerates the car slightly first before the hard braking; this action is caused by the non-minimum-phase feature, arising from the positive feedforward gains shown in Fig. 4.

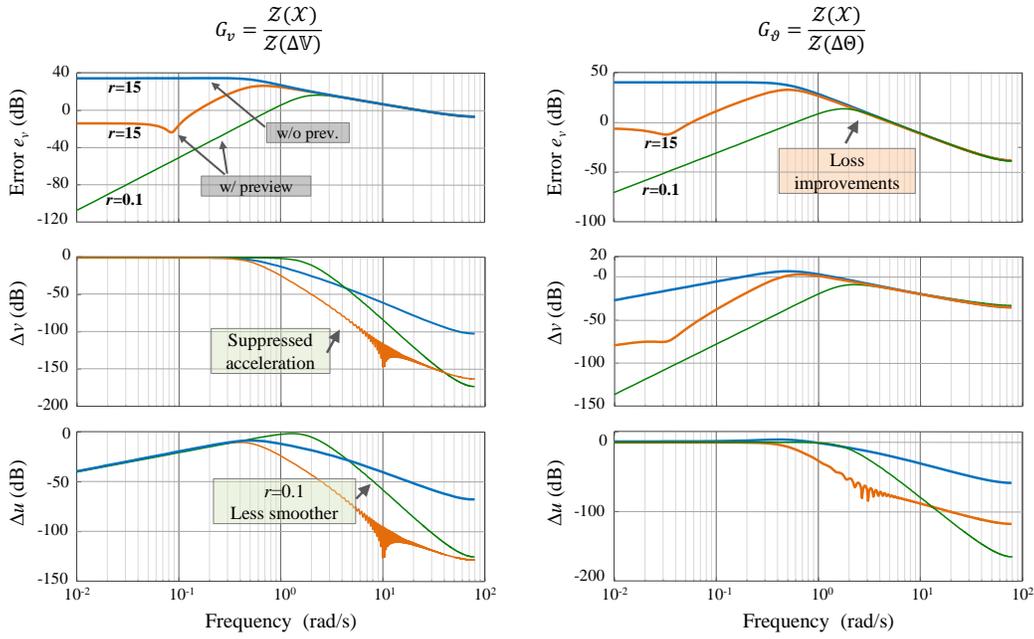


Fig. 6. Frequency response of the closed-loop system with and without previews. The left subfigures show the transfer function from speed change  $\Delta V(k)$  to states  $\mathcal{X}$ , and the right subfigures show the transfer function from road slope change  $\Delta\theta$  to states  $\mathcal{X}$ .

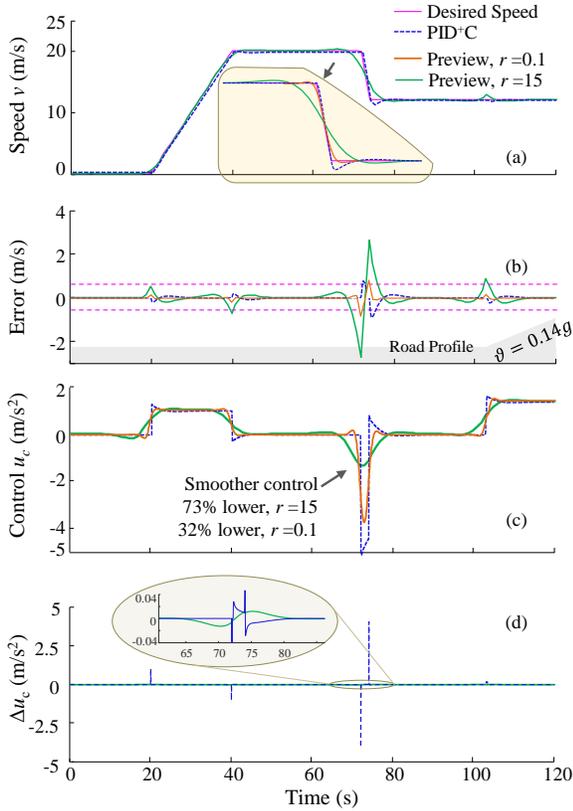


Fig. 7. Control results of the preview control and PID+C.

The preview control with future information is able to achieve smoother actions. Comparing the preview control ( $r = 0.1$ ) to the PID+C, their maximum errors are very close, but the maximal brake reduced from  $0.5g$  to  $0.34g$ , i.e., 32% lower, benefiting from the use of future information. A higher  $r$  can

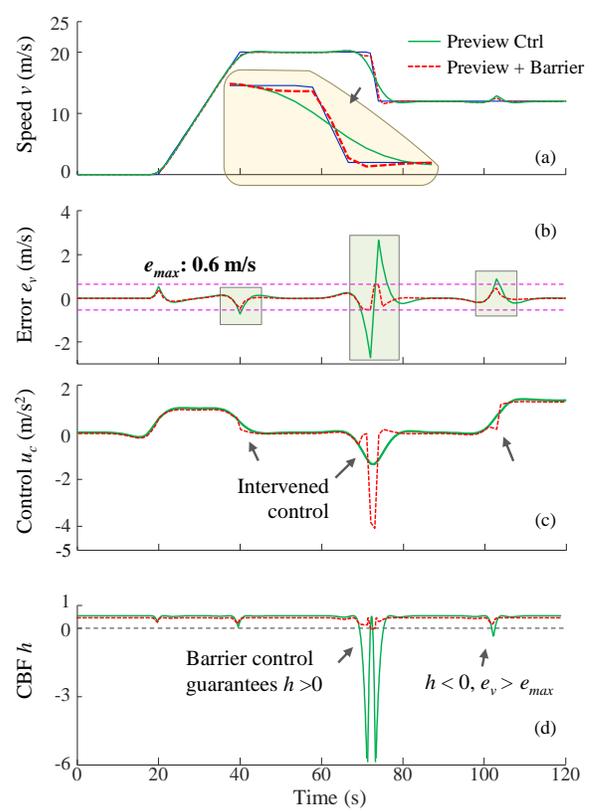


Fig. 8. Speed-tracking results of preview controller w/ and w/o barrier.

further improve tracking smoothness but worsen tracking accuracy, as designed in the cost function (7). For instance, if set  $r$  from  $0.1$  to  $15$ , the peak of error increases to from  $0.7$  to  $2.7$  m/s, the brake is reduced to  $0.135g$ , about 73% lower than the PID+C.

### C. Behavior of the Barrier Control

As mentioned earlier, both the preview control and PID+C cannot guarantee bounded errors. Let us take the preview control with  $r=15$  as an example (assuming a very smooth control is pursued), and set the acceptable  $e_{\max}$  to 0.6 m/s, the speed tracking fails during the hard brake where the maximal error 2.7 m/s is much higher than  $e_{\max}$ .

Applying the barrier control, the CBF  $h$  keeps  $h > 0$  over the whole trip, *i.e.*, the tracking errors are limited within the boundary  $e_{\max}$ , as shown in Fig. 8. The bounded errors are achieved at the expense of worse smoothness—the maximal brake command  $u_c$  increased from 0.135g to 0.4g. As a summary, the barrier control allows for smoothness-dominated operation (e.g., from the preview control) in safe scenarios, and can guarantee bounded errors in special cases, e.g., sudden hard brake in emergency scenarios. The strategy to split the original problem into two sub-problems is a tradeoff between the control optimality and computational efficiency. However, we did not tradeoff safety with other performance criteria, e.g., smoothness; instead, safety and smoothness are pursued by the barrier control and the preview controls separately; their cooperation delivers a smooth and safe speed tracking even in emergency scenarios.

## V. EXPERIMENTAL RESULTS

### A. Vehicle Platform and Testing Track

The Hybrid Lincoln MKZ testing platform of Mcity is used to implement the proposed controllers, as shown in Fig. 1. The key vehicle parameters are listed in Table 1. The platform is equipped with a high-precision Real Time Kinematic kit (RT3003 from Oxford Technical Solutions) and an Inertial Measurement Unit. These sensors measure the vehicle position, speed, and acceleration directly. Digital map provides slope information. The algorithms are implemented in Linux with C++; the software interface is shown in Fig. 9. The experiments are carried out in Mcity, an 18-acre test facility operated by the University of Michigan.

### B. Experimental Results

Fig. 10 shows the test scenario: the vehicle runs in the downtown area of Mcity, starting from point S and ending at point E. The path contains 5 left turns, 3 right turns, and a roundabout. The path, road slope, and target speed profile are shown in Fig. 10. At point M and B, the slopes are about -3.5 degrees. The car accelerates on the downhill from point A to B, and then aggressively brakes from B to C, where the vehicle turns right. The average acceleration and deceleration are 2.2 and -3.9 m/s<sup>2</sup>. The maximum speed is about 35 km/h at B, and the distance between A and C is 35 meters only. In the validation, both the PID+C (ignoring future information) and the preview control are tested; we assume that tracking smoothness is preferred and set  $r=15$ . Test results of the preview control and PID+C are shown in Fig. 11 (a)-(c).

Similar to the simulation results in Fig. 7, the preview control with future information achieves much smoother control in this case because of its look-ahead optimization feature. Its peak of

deceleration is 82% lower than the PID+C, but the maximal error is also much higher. Without using future information, the PID+C can track the target speed accurately but less smoothly, with more oscillations and switches between accelerating and braking, as shown in Fig. 12, where the two controls' brake/throttle pedal openings are compared.

Table 1 Vehicle Parameters of the Mcity test vehicle

Definition	Symbol	Value
Time constant	$\tau$	0.3 s
Sampling period	$\Delta\tau$	0.04 s
Weight of speed error	$q$	1
Weight of control input increment	$r$	15/ $\Delta\tau^2$
Preview steps of road slope	$N_\theta$	300
Preview steps of target speed	$N_v$	300
Minimal control input	$u_{\min}$	-6 m/s <sup>2</sup>
Maximal throttle pedal opening	$\phi_{t,\max}$	60%



Fig. 9. Software HMI.

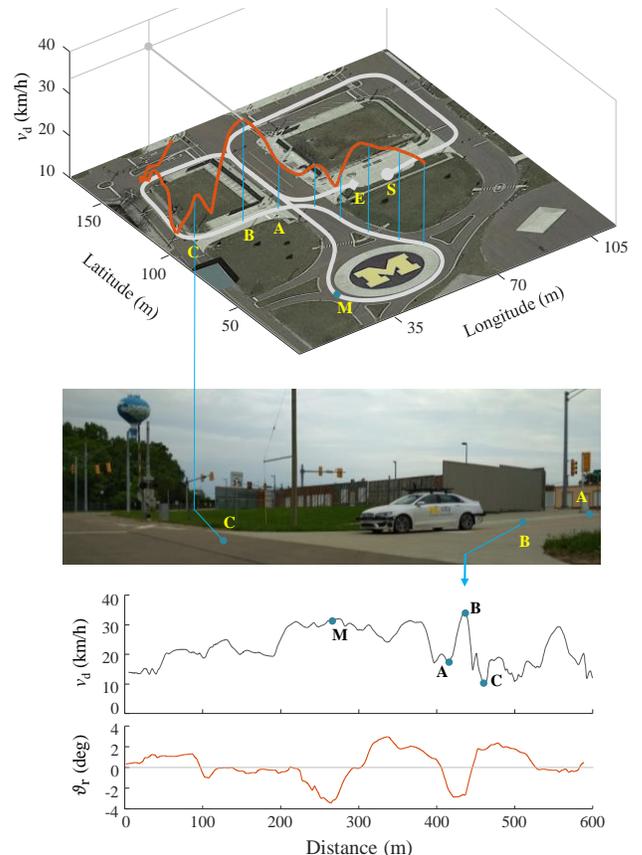


Fig. 10. Test scenario inside Mcity.

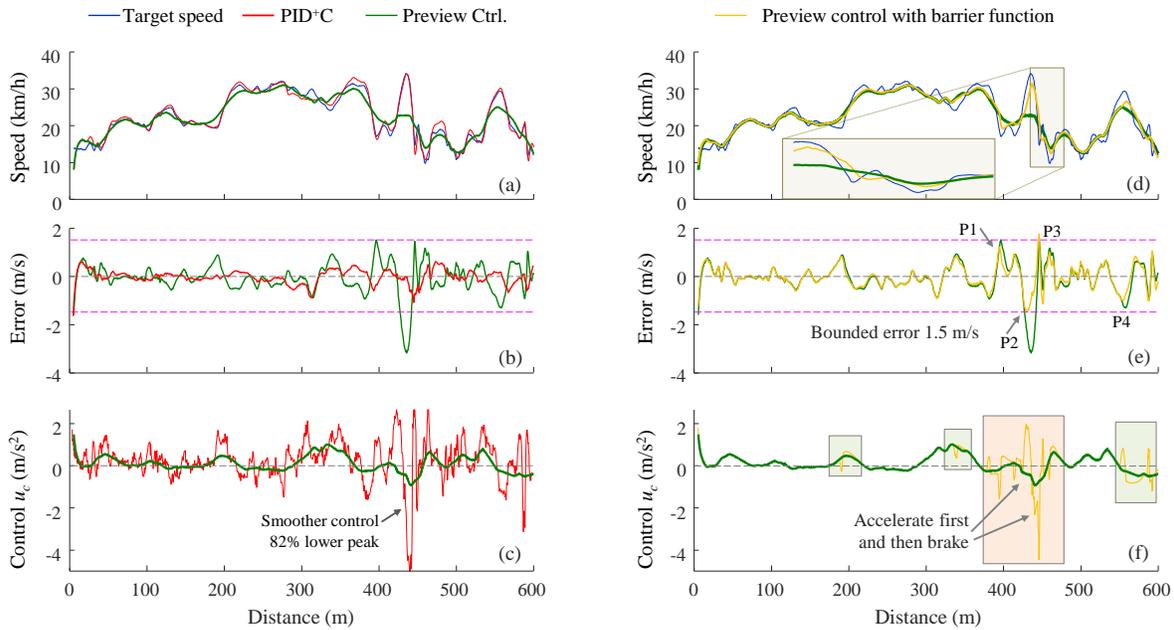


Fig. 11. Control results of preview control, PID+C, and preview control with the barrier function.

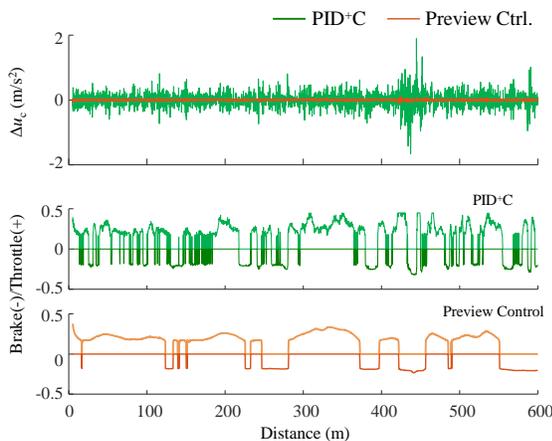


Fig. 12. Control actions of the preview and PID+C controls.

### C. Performance of Barrier Control

As mentioned in Fig. 8, a high  $r$  excites better smoothness but deteriorates tracking accuracy. In this test, the maximal error of the preview control ( $r = 15$ ) is as high as 3.2 m/s, as shown in Fig. 11 (b). Here we apply the barrier control to supervise the preview control and set  $e_{\max}$  to 1.5 m/s. As shown in Fig. 11 (d)-(f), the speed tracking errors are successfully limited. At P1, P3, and P4, the original errors are close to the bound, the barrier control thus gently intervenes, while at P2 it increases brake significantly. This result implies that the barrier control enables hard brake in emergency scenarios. The maximal deceleration is at a similar level with the PID+C during the hard brake, but the control commands are much smoother from the holistic perspective.

## VI. CONCLUSIONS

This paper presented the vehicle longitudinal dynamics models and the speed tracking control of the hybrid Lincoln

MKZ. This platform provides neither powertrain models nor control access to the powertrain (including engine and motor), and only brake/throttle pedals can be controlled. Therefore, an affine map was identified to describe the static characteristic between pedal opening and powertrain output; and a first-order system is used to approximate the transient dynamics. Leveraging these models, two speed-tracking algorithms are designed, *i.e.*, the preview control when future target speed and road slope information is available, and the PID+C control when the future information becomes unavailable. Both controllers have analytical laws and are computationally efficient. Both controls achieve accurate tracking, but the preview control using future information further improves speed-tracking smoothness, showed by the closed-loop system analysis in both the time and frequency domain. The barrier control is also designed to limit tracking errors. It supervises smoothness-oriented controls such as the preview control but guarantees bounded errors in emergency cases. On-track tests validated the two controls' performances and the effectiveness of the speed-error barrier function. The consideration of system communication delay and position constraint will be explored in future work.

## REFERENCES

- [1] L. D. Burns, "Sustainable mobility: a vision of our transport future," *Nature*, vol. 497, no. 7448, pp. 181-182, 2013.
- [2] S. E. Li, S. Xu, X. Huang, B. Cheng, and H. Peng, "Eco-departure of connected vehicles with V2X communication at signalized intersections," *IEEE Trans. on Vehicular Technology*, vol. 64, no. 12, pp. 5439-5449, 2015.
- [3] <https://autonomoustuff.com/product/astuff-automotive/>
- [4] B. Paden, M. Cap, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Trans. on Intelligent Vehicles*, vol. 1, no.1, pp. 33-55, 2016.
- [5] S. Xu and H. Peng, "Design, Analysis, and Experiments of Preview Path Tracking Control for Autonomous Vehicles," *IEEE Trans. on*

*Intelligent Transportation Systems*, pp. 1-11, 2019.

- [6] D. Yanakiev and I. Kanellakopoulos, "Speed tracking and vehicle follower control design for heavy-duty vehicles," *Vehicle System Dynamics*, vol. 25, no. 4, pp.251-276, 1996.
- [7] P. Ioannou, Z. Xu, S. Eckert, D. Clemons, and T. Sieja, "Intelligent cruise control: theory and experiment," *In Decision and Control, Proceedings of the 32nd IEEE Conf. on*, pp. 1885-1890, 1993.
- [8] P. Shakouri, A. Ordys, D.S. Laila, and M. Askari, "Adaptive cruise control system: comparing gain-scheduling PI and LQ controllers," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp.12964-12969, 2011.
- [9] P. Nilsson, O. Hussien, A. Balkan, *et al.*, "Correct-by-construction adaptive cruise control: Two approaches," *IEEE Trans. on Control Systems Technology*, vol. 24, no. 4, pp.1294-1307, 2016.
- [10] 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.
- [11] X. Li, Z. Sun, D. Cao, D. Liu, and H. He, "Development of a new integrated local trajectory planning and tracking control framework for autonomous ground vehicles," *Mechanical Systems and Signal Processing*, vol. 87, pp.118-137, 2017.
- [12] S. Li, K. Li, R. Rajamani, and J. Wang, "Model predictive multi-objective vehicular adaptive cruise control," *IEEE Trans. on Control Systems Technology*, vol. 19, no. 3, pp.556-566, 2011.
- [13] J. C. Gerdes and J. K. Hedrick, "Vehicle speed and spacing control via coordinated throttle and brake actuation," *Control Engineering Practice*, vol. 5, pp.11, pp.1607-1614, 1997.
- [14] S. Xu, S. E. Li, B. Cheng, and K. Li, "Instantaneous Feedback Control for a Fuel-Prioritized Vehicle Cruising System on Highways With a Varying Slope," *IEEE Trans. on Intelligent Transportation Systems*, vol. 18, no. 5, pp.1210-1220, 2017.
- [15] R. M. De Santis, "A novel PID configuration for speed and position control," *Journal of dynamic systems, measurement, and control*, vol. 116, no. 3, pp. 542-549, 1994.
- [16] S. E. Shladover, C. A. Desoer, J. K. Hedrick, *et al.*, "Automated vehicle control developments in the PATH program," *IEEE Transactions on vehicular technology*, vol. 40, no.1, pp.114-130, 1991.
- [17] H. Kim, D. Kim, I. Shu, and K. Yi, "Time-varying parameter adaptive vehicle speed control," *IEEE Trans. on Vehicular Technology*, vol. 65, no. 2, pp.581-588, 2016.
- [18] Z. Xu, and P. Ioannou, "Adaptive throttle control for speed tracking," *Vehicle System Dynamics*, vol. 23, no. 1, pp.293-306, 1994.
- [19] M. Zhu, H. Chen, and G. Xiong, "A model predictive speed tracking control approach for autonomous ground vehicles," *Mechanical Systems and Signal Processing*, vol. 87, pp.138-152, 2017.
- [20] 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. on Mechatronics*, vol. 20, no. 4, pp. 1499-1510, 2015.
- [21] M. Tomizuka, "The Optimal Finite Preview Problem and its Application to Mari-Machine Systems," Ph.D. dissertation, Mass. Inst. of Tech., Cambridge, MA, 1973.
- [22] T. Katayama, T. Ohki, T. Inoue, and T. Kato, "Design of an optimal controller for a discrete-time system subject to previewable demand," *International Journal of Control*, vol. 41, no. 3, pp.677-699, 1985.
- [23] S. Shimmyo, T. Sato, and K. Ohnishi, "Biped walking pattern generation by using preview control based on three-mass model," *IEEE Trans. on Industrial Electronics*, vol. 60, no. 11, pp. 5137-5147, 2013.
- [24] T. Salton, Z. Chen, J. Zheng, and M. Fu, "Preview control of dual-stage actuator for superfast transition time," *IEEE/ASME Trans. on Mechatronics*, vol. 16, no. 4, pp. 758-763, 2011.
- [25] H. Peng and M. Tomizuka, "Preview control for vehicle lateral guidance in highway automation," *Journal of Dynamic Systems, Measurement and Control*, vol. 115, pp. 679-679, 1993.
- [26] S. Xu, H. Peng, Z. Song, K. Chen, and Y. Tang, "Accurate and Smooth Speed Control for an Autonomous Vehicle," *In 2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1976-1982, June 2018.
- [27] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in Proc. 53rd IEEE Conf. Decision Control, pp. 6271-6278, 2014.



**SHAOBING XU** received his 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 at the University of Michigan, Ann Arbor. 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.



**HUEI PENG** received the Ph.D. degree in mechanical engineering from the University of California, Berkeley, CA, USA, in 1992.

He is currently a Professor with the Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI, USA. He currently serves as the Director of the University of Michigan Mobility Transformation Center, which is a center that oversees the Mcity Test Facility and studies connected and

autonomous vehicle technologies and promotes their deployment. His research interests include adaptive control and optimal control, with emphasis on their applications to vehicular and transportation systems. His current research focuses include the design and control of electrified vehicles and connected/automated vehicles. Dr. Peng is an Active Member of the Society of Automotive Engineers (SAE) and the American Society of Mechanical Engineers (ASME). He is both an SAE and an ASME Fellow.



**ZIYOU SONG** (M'18) received the B.E. degree and the Ph.D. degree (with the highest honor) in Automotive Engineering from Tsinghua University, Beijing, China, in 2011 and 2016, respectively.

He is currently a postdoctoral researcher with the University of Michigan, Ann Arbor. His research interests include optimal control theory, battery parameter estimation, hybrid energy storage system, and electric and hybrid electric

vehicles.



**KAILIANG CHEN** received dual bachelor degrees from Sun-yat-sen University in Guangzhou China, and Purdue School of Engineering and Technology in Indianapolis, USA in the year of 2016. He then received a Master degree in University of California, Berkeley in 2017, all in Mechanical Engineering.

From 2017 till now, he works as a researcher and engineer in the Intelligent Driving team in SF Motors Inc., Santa Clara, CA, USA, focus on the area of path planning and control system.



**YIFAN TANG** (M'94-SM'02) received the Ph.D. degree in electrical engineering from The Ohio State University, Columbus, Ohio, 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, Drivetrain Engineering with Lucid Motors. From 2015 to 2017, he was a Technical Lead, Power Electronics Systems with Facebook. Since 2017, he is the Chief Technology Officer with SF Motors Inc., Santa Clara, CA, USA. His current research interest includes electric vehicles and autonomous vehicles.

Dr. Tang's awards include an IEEE Industry Application Society Annual Meeting 2nd Prize Paper Award in 1998, and an IBM Outstanding Technical Achievement Award in 2002.