

# Incorporating ESO into Deep Koopman Operator Modeling for Control of Autonomous Vehicles

Hao Chen<sup>✉</sup> and Chen Lv<sup>✉</sup>, *Senior Member, IEEE*

**Abstract**—Koopman operator theory is a kind of data-driven modeling approach that accurately captures the nonlinearities of mechatronic systems such as vehicles against physics-based methods. However, the infinite-dimensional Koopman operator is impossible to implement in real-world applications. To approximate the infinite-dimensional Koopman operator through collection dataset rather than manual trial and error, we adopt deep neural networks (DNNs) to extract basis functions by offline training and map the nonlinearities of vehicle planar dynamics into a linear form in the lifted space. Besides, the effects of the dimensions of basis functions on the model accuracy are explored. Furthermore, the extended state observer (ESO) is introduced to online estimate the total disturbance in the lifted space and compensate for the modeling errors and residuals of the learned deep Koopman (DK) operator while also improving its generalization. Then, the proposed model is applied to predict vehicle states within prediction horizons and later formulates the constrained finite-time optimization problem of model predictive control (MPC), i.e., ESO-DKMPc. In terms of the trajectory tracking of autonomous vehicles, the ESO-DKMPc generates the wheel steering angle to govern lateral motions based on the decoupling control structure. The various conditions under the double-lane change scenarios are built on the CarSim/Simulink co-simulation platform, and extensive comparisons are conducted with the linear MPC (LMPC) and nonlinear MPC (NMPC) informed by the physics-based model. The results indicate that the proposed ESO-DKMPc has better tracking performance and moderate efficacy both within linear and nonlinear regions.

**Index Terms**—Autonomous vehicles, deep neural network (DNN), extended state observer (ESO), Koopman operator modeling, model predictive control (MPC), trajectory tracking control.

## I. INTRODUCTION

### A. Motivation and Related Work

SYSTEM modeling serves as the foundation for control and optimization, while model accuracy is the key to the design and performance of subsequent controllers [1]. In the case of nonlinear mechatronic systems, such as vehicles, it is difficult to identify nonlinear behaviors using a physics-based model. Some procedures can only be approximated or fit

Manuscript received 21 August 2023; revised 9 January 2024; accepted 7 March 2024. Date of publication 26 March 2024; date of current version 20 August 2024. This work was supported in part by the Agency for Science, Technology and Research (A\*STAR), Singapore, under Advanced Manufacturing and Engineering (AME) Young Individual Research under Grant A2084c0156, in part by the MTC Individual Research under Grant M22K2c0079, in part by the ANR-NRF Joint Grant NRF2021-NRF-ANR003 HM Science, and in part by the Ministry of Education (MOE), Singapore, under the Tier 2 under Grant MOE-T2EP5022-0002. Recommended by Associate Editor M. Ariola. (*Corresponding author: Chen Lv.*)

The authors are with the School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore 639798 (e-mail: chen.h@ntu.edu.sg; lyuchen@ntu.edu.sg).

Digital Object Identifier 10.1109/TCST.2024.3378456

using empirical models, but calibrating the hyperparameters within these empirical models is a challenging task [2], [3], [4], [5]. In turn, this poses difficulties for the formulation of control laws. The methodology of data-driven modeling offers a comprehensive guide to addressing these concerns. The data-driven approach does not depend on the internal physics of control plants. Instead, it utilizes input and output data to efficiently capture the characteristics of nonlinear components and enhance the accuracy of the system model [6], [7], [8], [9].

Unlike the local linearization [10], [11] or backstepping [12], [13] commonly used for nonlinear systems, Koopman operator lifts the original nonlinear state space into an infinite-dimensional linear state space through data [14], [15], [16], [17], [18]. The mapping from the original low-dimensional space to the infinite-dimensional lifted space enforces the nonlinearities to be linear behaviors, resulting in a linear representation of the nonlinear system. Therefore, the methodologies that have been established for the analysis and synthesis of linear systems have the potential for further extension. In addition, the original space can be embedded into the lifted space, i.e., subspace, and retrieved from the lifted space without any loss of accuracy. Petar Bevanda et al. have proven that Koopman representation has marginally better performance than other alternative data-driven modeling techniques, particularly in the context of nonlinear systems [19]. However, the infinite-dimensional Koopman operator is unattainable in applications. Thus, the suitable choice of finite-dimensional basis functions is considered to approximate the Koopman operator.

Deep learning method has been demonstrated in comparison to the two prevailing mainstream approaches, namely dynamic mode decomposition (DMD) [20], [21], [22] and extended DMD (EDMD) [21], [23], [24]. DMD and EDMD make a choice of basis functions based on a priori knowledge or by a non-trivial process of trial and error, thus restricting the generalization. Whereas the deep learning approach only builds the deep neural network (DNN), which is a kind of nonlinear mapping rather than DMD as linear mapping, and helps the Koopman operator to capture nonlinearities of the original space and evolve with linear form in the lifted space. Moreover, it can extract basis functions through an automated procedure following training [25]. Han et al. [26] employed the DNN for the data-driven identification of the appropriate basis functions to exploit the power of data. The OpenAI Gym environment is used for data generation and training of DNN to learn the basis functions of the Koopman operator. This particular controller is verified to be sufficient

for the reinforcement learning algorithm. Xiao et al. [27] introduced a DNN-based vehicle modeling approach with an interpretable Koopman operator. A model predictive controller with the learned Koopman model for velocity profile tracking of autonomous vehicles shows better performance and higher computational efficiency than other traditional and advanced modeling methods.

On the other hand, the finite-dimensional approximation for the Koopman operator introduces modeling errors, and residuals are inherent during offline network training. Moreover, vehicles exhibit time-varying dynamics when operating in complicated environments. If the real-world scenario extends beyond the feature space identified by the training dataset, the learned Koopman operator may result in inaccuracies, even faults. The extended state observer (ESO) augments the unknown components of the system, including modeling errors and uncertainties, as its extended state while replacing the linear feedback functions with a nonlinear form. This renders the observer not reliant on the precise physics model but on the necessary measurements, with promising convergency [28]. Guerrero et al. [29] designed adaptive ESO to observe unknown external disturbances and parametric uncertainties of underwater vehicles in finite time. With the adaptive laws to update the gains of ESO, the observer can reject bounded time-varying disturbances even if the upper bound of the perturbation is not known. Additionally, the stability of the observer scheme is proven using Lyapunov's arguments. Ren et al. [30] proposed a comprehensive Koopman operator-based robust data-driven control framework for wheeled mobile robots. The EDMD approach is applied to obtain the Koopman operator, and the modeling errors are online estimated by ESO and compensated in the control signal from the sliding mode controller. Experimental tests show robustness against disturbances.

### B. Contribution

Motivated by the aforementioned above, this article adopts DNNs to approximate the infinite-dimensional Koopman operator, i.e., the deep Koopman (DK) operator, to feature the vehicle dynamics based on the collection dataset. To reduce the modeling errors and residuals, ESO is specifically designed to online estimate the total disturbances to improve the accuracy of the DK operator model. Meanwhile, the decoupling control structure is utilized in order to simplify the trajectory tracking of autonomous vehicles. The ESO-DKMPc is developed to govern the vehicle's lateral motions. The main contributions of this article are listed as follows.

- 1) ESO is designed to online estimate the modeling errors and residuals, afterward as a complement to the learned DK operator to enhance its model fidelity and generalization. This is helpful for analyzing vehicle planar dynamics in the lifted space.

- 2) In terms of the vehicle's lateral motion regulation, by incorporating ESO, the DK operator model acts as the predictive model to formulate the constrained finite-time optimization problem of the model predictive control (MPC) scheme in the lifted space, referred to as ESO-DKMPc.

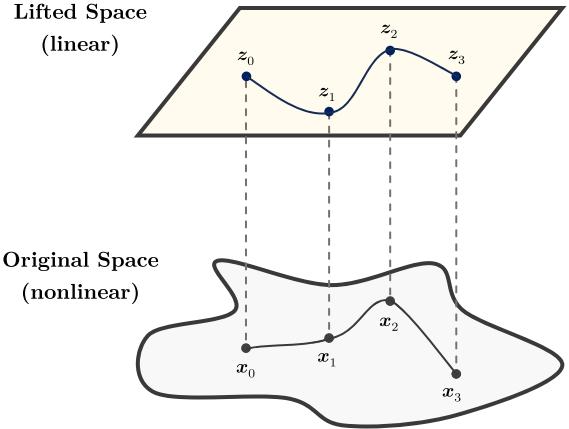


Fig. 1. Koopman operator  $\kappa$  lifting.

- 3) The effects of the dimensions of basis functions on the model fidelity are investigated, and various conditions are built on the CarSim/Simulink co-simulation platform to conduct rigorous comparisons with other methods.

### C. Article Organization

The remaining article is organized as follows: Section II provides a brief overview of the methodologies of deep learning Koopman operator, ESO, and the ESO-DKMPc. The framework of ESO-DKMPc for trajectory tracking of autonomous vehicles, including the decoupling control structure, the DK operator model with ESO regarding vehicle dynamics, and the corresponding constrained finite-time optimization problem, are formulated in Section III. Moreover, the dimensions of the basis functions are also discussed therein. Section IV validates the accuracy and robustness of the DK model, which incorporates an online estimated total disturbance by ESO. In addition, the comparison studies of four typical methods of tracking performance are conducted under various double-lane change scenarios on the CarSim/Simulink co-simulation platform. Section V concludes the article.

## II. METHODOLOGY

### A. Deep Learning Koopman Operator

A general discrete-time nonlinear system can be defined as

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \quad (1)$$

where  $\mathbf{x}_k \in \mathbf{R}^n$  is the state vector at time  $k$ ,  $\mathbf{u}_k \in \mathbf{R}^m$  is control input vector, and  $\mathbf{f} : \mathbf{R}^{n+m} \rightarrow \mathbf{R}^n$  is a smooth vector field of  $\mathbf{C}^\infty$ .

Koopman operator, denoted by  $\kappa$ , is an infinite linear operator defined on the space of embedding functions  $\mathbf{g}$  as

$$\kappa \mathbf{g}(\mathbf{x}_k, \mathbf{u}_k) = \mathbf{g}(\mathbf{x}_{k+1}, \mathbf{u}_{k+1}) = \mathbf{g}(\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k), \mathbf{u}_{k+1}) \quad (2)$$

where  $\mathbf{g} : \mathbf{R}^{m+n} \rightarrow \mathbf{R}^d$  is a smooth vector field of  $\mathbf{C}^\infty$ .

Therefore, the original state space has been lifted into the designated embedding space by  $\mathbf{g}$ . This mapping, characterized by the Koopman operator  $\kappa$  (Fig. 1), allows the nonlinear dynamics inherent in the original state space to be linear.

The embedding function  $\mathbf{g}$  can be further divided into two components

$$\mathbf{g}(\mathbf{x}_k, \mathbf{u}_k) = \begin{bmatrix} \mathbf{g}_x(\mathbf{x}_k) \\ \mathbf{u}_k \end{bmatrix} \quad (3)$$

where  $\mathbf{g}_x : \mathbf{R}^n \rightarrow \mathbf{R}^{d-m}$ .

However, the infinite-dimensional  $\kappa$  is impossible to store and compute in real applications but can be approximated by a finite-dimensional representation. Then, without loss of generality, (2) is simplified within finite dimension space as

$$\begin{bmatrix} \mathbf{g}_x(\mathbf{x}_{k+1}) \\ \mathbf{u}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{K}_{xx} & \mathbf{K}_{xu} \\ \mathbf{K}_{ux} & \mathbf{K}_{uu} \end{bmatrix} \begin{bmatrix} \mathbf{g}_x(\mathbf{x}_k) \\ \mathbf{u}_k \end{bmatrix} \quad (4)$$

where  $\mathbf{K} \in \mathbf{R}^{d \times d}$  is the finite-dimensional approximation of  $\kappa$  in the lifted space.

From (4), we denote  $\mathbf{K}_{xx} = \mathbf{A}$ ,  $\mathbf{K}_{xu} = \mathbf{B}$ , and the time-invariant state evolution in the lifted space can be written as

$$\mathbf{g}_x(\mathbf{x}_{k+1}) = \mathbf{A}\mathbf{g}_x(\mathbf{x}_k) + \mathbf{B}\mathbf{u}_k \quad (5)$$

where  $\mathbf{A} \in \mathbf{R}^{(d-m) \times (d-m)}$  and  $\mathbf{B} \in \mathbf{R}^{(d-m) \times m}$  are the finite-dimensional approximation of  $\kappa$  in terms of states. In terms of the input evolution, i.e.,  $\mathbf{u}_{k+1} = \mathbf{K}_{ux}\mathbf{g}_x(\mathbf{x}_k) + \mathbf{K}_{uu}\mathbf{u}_k$ ,  $\mathbf{u}_{k+1}$  is free to be chosen. Therefore,  $\mathbf{K}_{ux}$  and  $\mathbf{K}_{uu}$  are time-varying, since they are not concluded until  $\mathbf{u}_{k+1}$  is specified.

Further, to reconstruct the original state space from the lifted space at time  $k$ ,  $\mathbf{g}_x$  is defined as

$$\mathbf{g}_x(\mathbf{x}_k) := \mathbf{z}_k = \begin{bmatrix} \mathbf{x}_k \\ \Phi_k(\mathbf{x}_k) \end{bmatrix} \quad (6)$$

where

$$\begin{aligned} \Phi_k(\mathbf{x}_k) \\ = [\varphi_{1,k}(\mathbf{x}_k), \varphi_{2,k}(\mathbf{x}_k), \dots, \varphi_{d-m-n,k}(\mathbf{x}_k)]^T \in \mathbf{R}^{d-m-n} \end{aligned}$$

is the basis in the lifted space.

From (6), we have

$$\mathbf{x}_k = \mathbf{C}_P \mathbf{z}_k \quad (7)$$

where  $\mathbf{C}_P = [\mathbf{I}_n, \mathbf{0}] \in \mathbf{R}^{n \times (d-m)}$  is the projection matrix from the lifted space to the state space.

Consequently, the state space is encompassed within the lifted space, enabling its faithful reconstruction without any loss of accuracy through the  $\mathbf{g}_x$  projection rather than these approximations in [21].

The aforementioned analysis reveals that the choice of  $\Phi$  determines the approximation accuracy of  $\mathbf{K}$  in (4). Whereas DNN is regarded as a set of nonlinear basis functions to approximate complex relationships. Therefore, the basis  $\Phi$  can be directly learned by training DNN from data pairs, which enforces linear dynamics in the lifted space (Fig. 2). Its number of hidden layers and units are manually tuning parameters to balance accuracy and complexity. Besides, two single-hidden layers, without activation and bias, are employed as the linear transformation to feature  $\mathbf{A}$  and  $\mathbf{B}$  in the lifted space (Fig. 2), respectively. This ensures the continuous derivatives of the loss function in training, thus enabling efficient error backpropagation and a quick convergence of DNN.

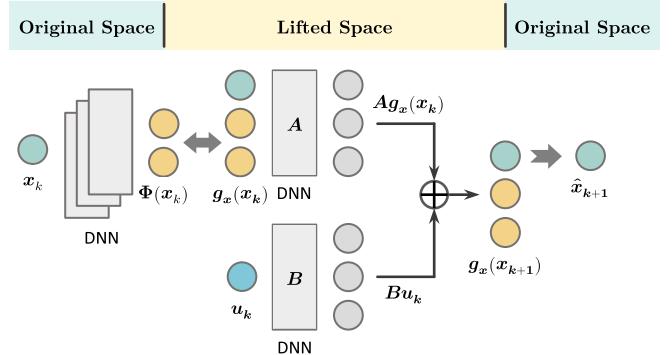


Fig. 2. Framework of DK operator.

The loss function  $L$  is defined to learn the weights and bias of DNN in the following equation:

$$L = L_1 + L_2 \quad (8)$$

where  $L_1$  is the total prediction loss of a data sequence

$$L_1(\theta_\Phi; \theta_A; \theta_B) = \sum_{i=1}^{p-1} \|\mathbf{x}_{k+i} - \hat{\mathbf{x}}_{k+i}\|_2^2 \quad (9)$$

where  $p$  is the sequence length,  $\theta_\Phi$ ,  $\theta_A$ , and  $\theta_B$  are the learning parameters of DNNs, respectively, and  $\hat{\mathbf{x}}_{k+i}$  ( $i = 1, 2, \dots, p-1$ ) is the  $i$ -step forward prediction from the lifted space by the DK operator, denoted as

$$\hat{\mathbf{x}}_{k+i} = \mathbf{C}_P [\mathbf{A}(\theta_A) \mathbf{z}_{k+i-1} + \mathbf{B}(\theta_B) \mathbf{u}_{k+i-1}] \quad (10)$$

and  $L_2$  is the reconstruction loss

$$L_2(\theta_\Phi; \theta_{\Phi^{-1}}) = \sum_{i=1}^p \|\mathbf{x}_k - \Phi^{-1}(\Phi(\mathbf{x}_k))\|_2^2 \quad (11)$$

where  $\Phi^{-1}$  is the decoder net of  $\Phi$ . This intricate interplay between the encoder and decoder networks enforces the encoded representation that encompasses the essential characteristics of the input data. As a result, the decoder can reconstruct the input based on the encoded features.

### B. Extended State Observer

ESO augments the total disturbance of system, including unmodelled dynamics and external disturbances, as a new state variable and later estimates all the states of the extended system. Notably, ESO estimates the external disturbances and unmodelled plant dynamics using the input-output data rather than the system and input matrices. This approach enables a more comprehensive estimation framework.

The DK operator has inevitable modeling errors. Therefore, the evolution dynamics in the lifted space are adherent to

$$\mathbf{z}_{k+1} = \mathbf{A}_\theta \mathbf{z}_k + \mathbf{B}_\theta \mathbf{u}_k + \mathbf{w}_k \quad (12)$$

where  $\mathbf{z}_k$  is defined in (6) with the learned basis  $\Phi$ ,  $\mathbf{A}_\theta$  and  $\mathbf{B}_\theta$  are the learned system matrix and input matrix, respectively, and  $\mathbf{w}_k \in \mathbf{R}^{d-m}$  is the total modeling error or disturbance in the lifted space.

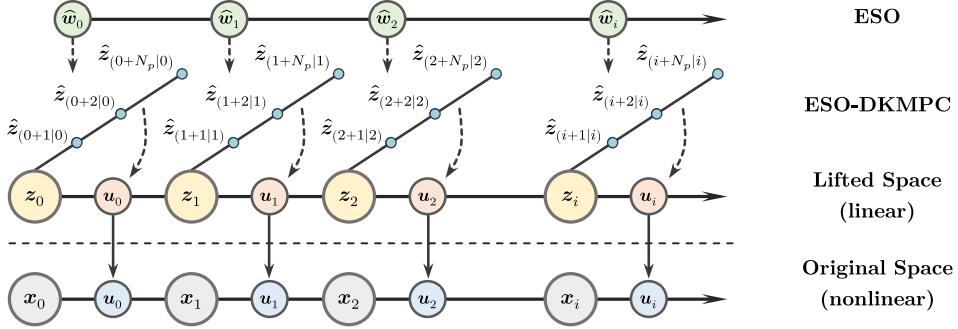


Fig. 3. Incorporating ESO into MPC in the lifted space.

Based on the dynamics in the lifted space of (12), the following ESO is defined as:

$$\begin{cases} \hat{z}_{k+1} = A_\theta \hat{z}_k + B_\theta u_k + [\hat{w}_k - \beta_1 g_1(\tilde{z}_k)] \\ \hat{w}_{k+1} = \hat{w}_k - \beta_2 g_2(\tilde{z}_k) \end{cases} \quad (13)$$

where  $\hat{z}_k$  and  $\hat{w}_k$  are the estimated results of state vector and total disturbance in the lifted space, respectively, and  $\tilde{z}_k$  is the estimation error between  $z_k$  and  $\hat{z}_k$  as

$$\tilde{z}_k = z_k - \hat{z}_k \quad (14)$$

where  $g_1$  and  $g_2$  are differentiable and continuous functions around the original with the scaling factors  $\beta_1$  and  $\beta_2$ .

*Remark 1:* By choosing appropriate differentiable and continuous functions, as well as scaling factors, (13) has been proven its convergency and consistency with (12) while providing a reliable estimation of  $w_k$  at each time step (see Appendix). To be brevity, we choose  $g_1(\tilde{z}_k) = g_2(\tilde{z}_k) = \tilde{z}_k$  in a linear form in this work.

### C. ESO-DK Operator-Informed MPC (ESO-DKMPc)

The framework of incorporating ESO into MPC in the lifted space is illustrated in Fig. 3. The total disturbance or modeling error is estimated online by ESO at each time step and then involved in the predictive model in the lifted space to improve the model fidelity and generate the control sequence. Moreover, the updated total disturbance remains unchanged during prediction horizons.

Therefore, based on (5)~(7) and (12), we formulate a new optimal problem of MPC in the corresponding lifted space that also integrates disturbance estimation from ESO, that is, ESO-DKMPc, as

$$\begin{aligned} \min_{U_k} \sum_{i=1}^{N_p} & \| \psi(C_P \hat{z}_{k+i|k}) - \eta_{r,k+i|k} \|_Q^2 + \sum_{i=0}^{N_c-1} \| u_{k+i|k} \|_R^2 \\ & + \sum_{i=0}^{N_c-1} \| \Delta u_{k+i|k} \|_P^2 \end{aligned}$$

$$\hat{z}_{k+i+1|k} = A_\theta \hat{z}_{k+i|k} + B_\theta u_{k+i|k} + \hat{w}_k \quad \forall i \in [0, N_c - 1]$$

$$\hat{z}_{k+i+1|k} = A_\theta \hat{z}_{k+i|k} + B_\theta u_{k+N_c-1|k} + \hat{w}_k \quad \forall i \in [N_c, N_p - 1]$$

$$\Delta u_{k+i|k} = u_{k+i|k} - u_{k+i-1|k} \quad \forall i \in [0, N_c - 1]$$

$$h_x(\hat{z}_{k+i|k}) \leq 0 \quad \forall i \in [1, N_p]$$

$$h_u(u_{k+i|k}) \leq 0 \quad \forall i \in [0, N_c - 1]$$

$$\hat{z}_{k|k} = z_k, u_{k-1|k} = u_{k-1} \quad (15)$$

where  $U_k = [u_{k|k}, u_{k+1|k}, \dots, u_{k+N_c-1|k}]^T$  is the optimal control sequence,  $\psi$  is the output function in the original space,  $\eta_r$  is the corresponding reference during prediction horizons,  $Q$  and  $P$  are semi-positive-definite weight matrices for states and control increments, respectively,  $R$  is positive definite weight matrix for control efforts,  $N_c$  and  $N_p$  are steps of control and prediction horizon, respectively,  $h_x$  and  $h_u$  are vector functions in terms of the constraints of  $x$  and  $u$ , respectively.

*Remark 2:* Because we decouple  $g_x(x)$  and  $u$  from  $g$  while maintaining  $u$  a linear form in (3), the optimal control sequence of (15) is equivalent to that of in the original space, and the first element of  $U_k$  can be applied straight to the control plant.

### III. ESO-DKMPc FOR TRAJECTORY TRACKING OF AUTONOMOUS VEHICLES

Fig. 4 depicts the scheme of ESO-DKMPc for trajectory tracking of autonomous vehicles. We employ the decoupling structure to control the longitudinal velocity and lateral motions because it is amenable to practical implementations. Concretely, the proportional controller (P control) is designed to track the desired longitudinal velocity profile  $V_{x,r}$  from the higher-level planner and obtain the total torque  $T$ . In terms of lateral motions, through using a learned DNN-informed Koopman operator, the longitudinal velocity  $V_x$ , lateral velocity  $V_y$ , and yaw rate  $\omega_r$  of the vehicle are initially mapped into the lifted space. Meanwhile, ESO online estimates the total disturbance  $w$  in the lifted space, later involved in the predictive model. The current pose signals, i.e., positions  $X$  and  $Y$ , yaw angle  $\theta$  in the global frame, are measured from sensors [31], [32]. Consequently, concerning the total torque  $T$  and the reference pose  $\eta_r$  also from the planner, the optimal control problem, ESO-DKMPc, as defined in (15), is formulated and solved to generate the control command.

#### A. Vehicle Planar Dynamics

The four-wheel vehicle planar dynamics model is demonstrated in Fig. 5. The Newton–Euler equations of planar

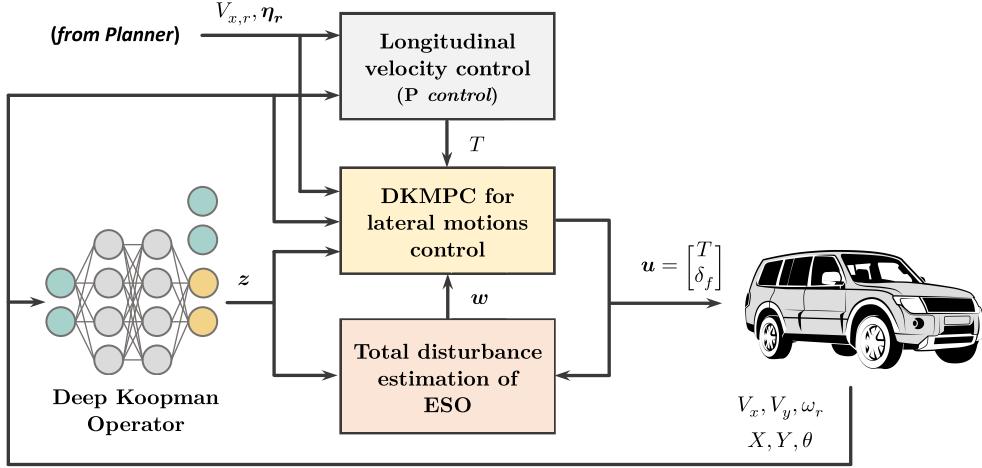


Fig. 4. Framework of ESO-DKMPc for trajectory tracking of autonomous vehicles.

motion for the vehicle at time  $k$  are [33]

$$\begin{aligned} V_{x,k+1} &= V_{x,k} + T_s V_{y,k} \omega_{r,k} \\ &+ \frac{T_s}{m} (F_{x3,k} + F_{x4,k}) + \frac{T_s}{m} (F_{x1,k} + F_{x2,k}) \cos \delta_{f,k} \\ &- \frac{T_s}{m} (F_{y1,k} + F_{y2,k}) \sin \delta_{f,k} \\ V_{y,k+1} &= V_{y,k} - T_s V_{x,k} \omega_{r,k} \\ &+ \frac{T_s}{m} (F_{y3,k} + F_{y4,k}) + \frac{T_s}{m} (F_{x1,k} + F_{x2,k}) \sin \delta_{f,k} \\ &+ \frac{T_s}{m} (F_{y1,k} + F_{y2,k}) \cos \delta_{f,k} \\ \omega_{r,k+1} &= \omega_{r,k} - T_s \frac{l_r}{I_z} (F_{y3,k} + F_{y4,k}) \\ &- T_s \frac{w_B}{2I_z} [(F_{x1,k} \cos \delta_{f,k} - F_{y1,k} \sin \delta_{f,k}) + F_{x3,k}] \\ &+ T_s \frac{w_B}{2I_z} [(F_{x2,k} \cos \delta_{f,k} - F_{y2,k} \sin \delta_{f,k}) + F_{x4,k}] \\ &+ T_s \frac{l_f}{I_z} (F_{x1,k} \sin \delta_{f,k} + F_{y1,k} \cos \delta_{f,k}) \\ &+ T_s \frac{l_f}{I_z} (F_{x2,k} \sin \delta_{f,k} + F_{y2,k} \cos \delta_{f,k}) \end{aligned} \quad (16)$$

where  $T_s$  is the discrete time step,  $V_{x,k}$ ,  $V_{y,k}$ , and  $\omega_{r,k}$  are the longitudinal velocity, lateral velocity, and yaw rate of the vehicle, respectively,  $F_{xi,k}$  ( $i = 1, 2, 3, 4$ ) is the longitudinal tire force,  $F_{yi,k}$  ( $i = 1, 2, 3, 4$ ) is the lateral tire force,  $\delta_{f,k}$  is the steering angle of the front wheel,  $m$  is the vehicle mass,  $I_z$  is the mass moment,  $l_f$  is the distance from the front axle to c.g.,  $l_r$  is the distance from the rear axle to c.g., and  $w_B$  is the track width.

The longitudinal and lateral tire forces of the  $i$ th ( $i = 1, 2, 3, 4$ ) driven wheel are characterized using the empirical model [34], [35] as

$$\begin{aligned} F_{xi,k} &= F_{xi,k}(T_{i,k}, w_{i,k}, F_{zi,k}, \mu) \\ F_{yi,k} &= F_{yi,k}(\alpha_{i,k}, F_{zi,k}, \mu) \end{aligned} \quad (17)$$

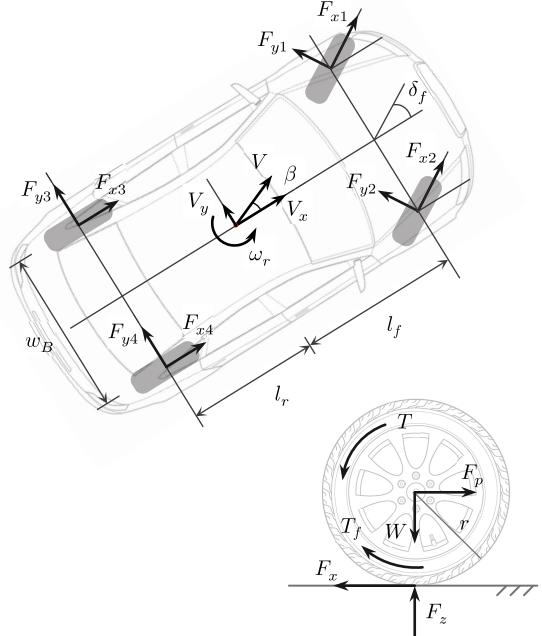


Fig. 5. Four-wheel vehicle planar dynamics model.

where  $w_{i,k}$  is the rotational velocity,  $T_{i,k}$  is the motor torque,  $\alpha_{i,k}$  is the tire sideslip angle related to velocities, yaw dynamics, steering angle, and vehicle specifications [26],  $F_{zi,k}$  is the normal force equal to the tire vertical load  $W_{i,k}$ , and  $\mu$  is the road friction coefficient.

The driving torque and the steering angle are the control inputs, and the longitudinal velocity, lateral velocity, and yaw rate are the states of the vehicle planar dynamics model

$$\begin{aligned} \mathbf{u}_k &= [T_k, \delta_{f,k}]^T \\ \mathbf{x}_k &= [V_{x,k}, V_{y,k}, \omega_{r,k}]^T \end{aligned} \quad (18)$$

where the driving torque  $T_k = T_{1,k} + T_{2,k} + T_{3,k} + T_{4,k}$  is the summation of motor torque of each wheel.

TABLE I  
STATISTIC RESULTS OF ONE-STEP PREDICTION ERRORS WITH RESPECT TO DIFFERENT DIMENSIONS

	$E_x$ (m/s)	$E_y$ (m/s)	$E_r$ (rad/s)
	Max / Avg / RMSE	Max / Avg / RMSE	Max / Avg / RMSE
<b>2</b>	0.815 / 0.211 / 0.270	0.229 / 0.057 / 0.076	0.223 / 0.042 / 0.059
<b>3</b>	0.812 / 0.361 / 0.403	0.301 / 0.070 / 0.093	0.164 / 0.036 / 0.049
<b>5</b>	1.077 / 0.254 / 0.322	0.151 / 0.037 / 0.047	0.067 / 0.016 / 0.022
<b>10</b>	0.268 / 0.059 / 0.077	0.154 / 0.034 / 0.045	0.097 / 0.025 / 0.032
<b>15</b>	0.367 / 0.120 / 0.146	0.035 / 0.010 / 0.012	0.052 / 0.011 / 0.014

Based on (14)~(16), (13) of discrete-time form is denoted as

$$\begin{bmatrix} V_{x,k+1} \\ V_{y,k+1} \\ \omega_{r,k+1} \end{bmatrix} = f \left( \begin{bmatrix} V_{x,k} \\ V_{y,k} \\ \omega_{r,k} \end{bmatrix}, \begin{bmatrix} T_k \\ \delta_{f,k} \end{bmatrix} \right) \quad (19)$$

where  $f$  maps the nonlinear evolution dynamics of the vehicle in a planar motion consistent with that of (1).

To follow the desired path,  $\psi$  is defined as  $\psi(\mathbf{C}_P \hat{\mathbf{z}}_{k+i|k}) = [X_{k+i|k}, Y_{k+i|k}, \theta_{k+i|k}]^T$ , and updated in the following equation:

$$\begin{cases} X_{k+i+1|k} = X_{k+i|k} \\ \quad + T_s (V_{x,k+i|k} \cos \theta_{k+i|k} - V_{y,k+i|k} \sin \theta_{k+i|k}) \\ Y_{k+i+1|k} = Y_{k+i|k} \\ \quad + T_s (V_{x,k+i|k} \sin \theta_{k+i|k} + V_{y,k+i|k} \cos \theta_{k+i|k}) \\ \theta_{k+i+1|k} = \theta_{k+i|k} + T_s \omega_{r,k+i|k}. \end{cases} \quad (20)$$

The vector function  $\mathbf{h}_x$  is defined as

$$\mathbf{h}_x(\hat{\mathbf{z}}_{k+i|k}) = \begin{bmatrix} \mathbf{C}_P \hat{\mathbf{z}}_{k+i|k} - \mathbf{x}_{\max} \\ -\mathbf{C}_P \hat{\mathbf{z}}_{k+i|k} + \mathbf{x}_{\min} \end{bmatrix} \quad (i \in [1, N_p]) \quad (21)$$

to respect the physical constraints of vehicle states, where  $\mathbf{x}_{\min}$  and  $\mathbf{x}_{\max}$  are the corresponding minimum and maximum values, respectively.

Moreover, as the decoupling control structure,  $T_k$  is generated from the P controller, which remains unchanged during the control of lateral motions. As a result, (15) only finds the solution of wheel steering angle within its physical constraints, and thus  $\mathbf{h}_u$  is defined as

$$\mathbf{h}_u(\mathbf{u}_{k+i|k}) = \begin{bmatrix} \delta_{f,k+i|k} - \delta_{f,\max} \\ -\delta_{f,k+i|k} + \delta_{f,\min} \\ T_{k+i|k} - T_k \\ -T_{k+i|k} + T_k \end{bmatrix} \quad (i \in [0, N_c - 1]) \quad (22)$$

where  $\delta_{f,\min}$  and  $\delta_{f,\max}$  are the minimum and maximum values of admissible wheel steering angle, respectively. The admissible range is expected to cover the steering maneuvers that allow the vehicle to follow the desired path without exceeding the physical constraints of the steering system. In this study,  $\delta_{f,\max} = -\delta_{f,\min} = 0.3$  rad.

*Remark 3:* Using commercial software-CarSim and a hardware-in-the-loop test platform [5], the collected data were sampled every 0.025 s. In order to facilitate the network's coverage of the feature space, a wide variety of road segments and driving maneuvers are included.

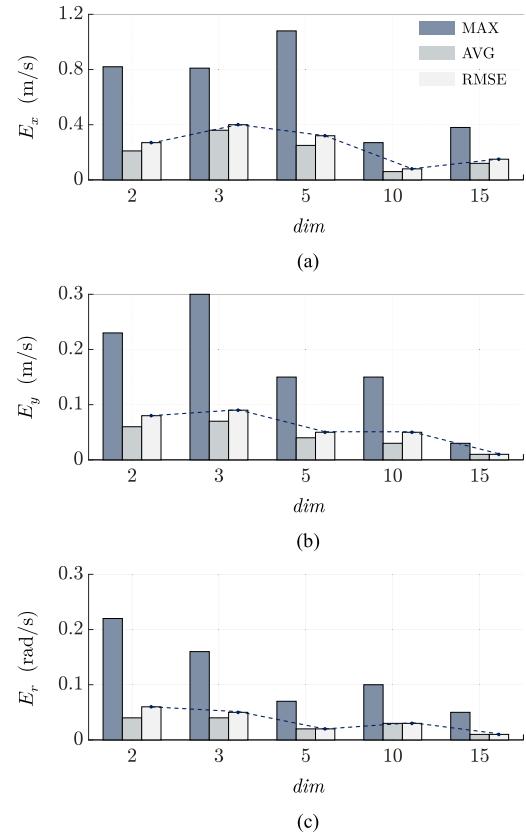


Fig. 6. One-step prediction errors with respect to different dimensions.  
(a) Longitudinal velocity. (b) Lateral velocity. (c) Yaw rate.

### B. Dimensions of $\Phi_k$

The dimensions of  $\Phi_k$  are crucial hyperparameters that need to be specified before building a DNN. The higher dimensions strengthen the model fidelity in the lifted space but at the expense of increased training time, storage resources, and the scale of optimization problem. The dimensions of  $\Phi_k$  should represent a “trade-off” between all these factors. Therefore, we define “one-step prediction error,” i.e., predicting longitudinal velocity, lateral velocity, and yaw rate one step further based on the learned Koopman operator with different dimensions of  $\Phi_k$  and comparing with the actual values to evaluate the reconstruction accuracy and select the appropriate dimensions in Fig. 6 and Table I.

As the number of dimensions increases, the model accuracy incrementally improves in Fig. 6. However, the improvement tends to be saturated beyond the dimensions of 15, followed

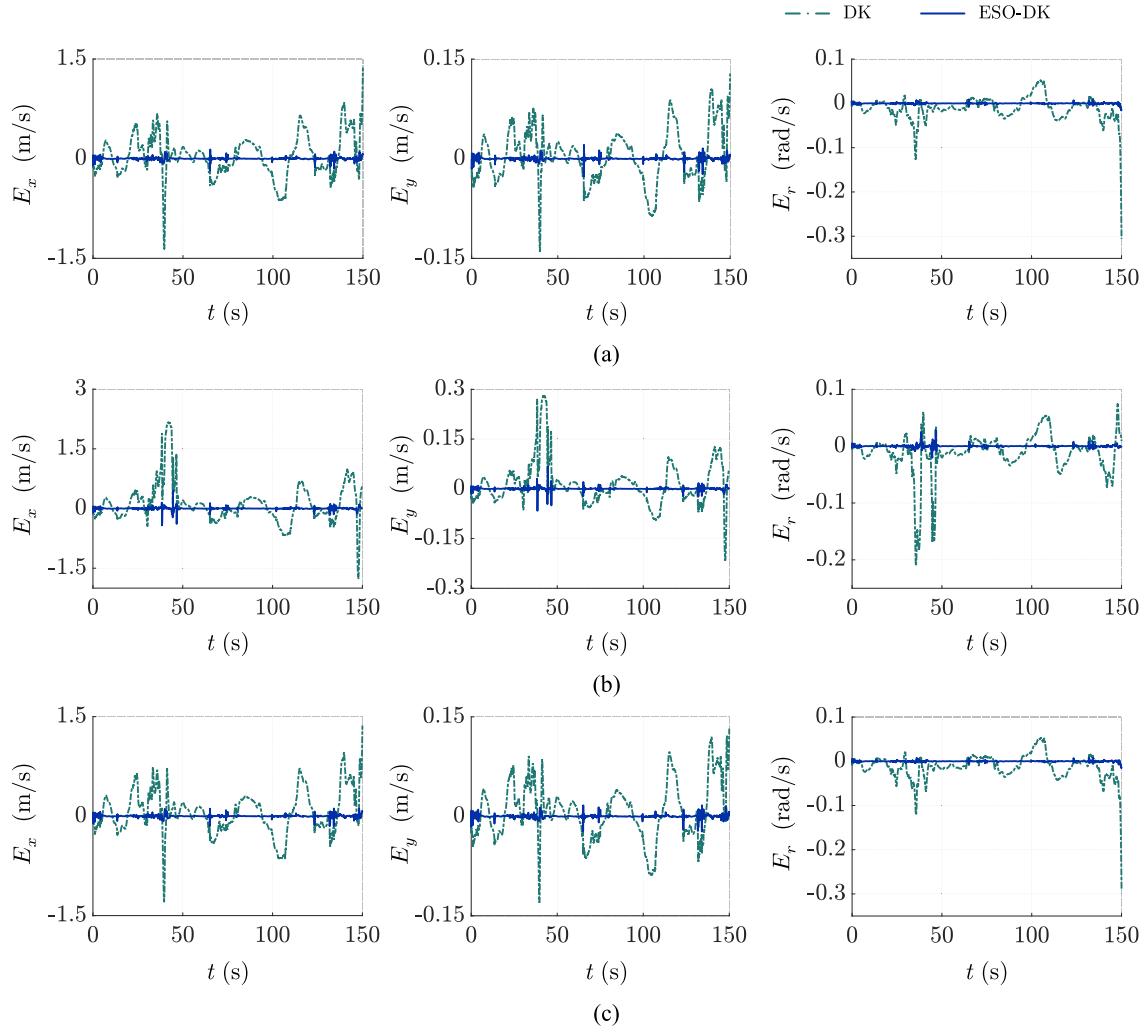


Fig. 7. Comparison results of deep Koopman operator model with ESO (ESO-DK) and without ESO (DK). (a) 1648 kg (sprung mass) and 0.85 (road adhesion coefficient). (b) 1648 kg (sprung mass) and 0.55 (road adhesion coefficient). (c) 1848 kg (sprung mass) and 0.85 (road adhesion coefficient).

by a greater network scale. The estimation error for the longitudinal velocity of the dimensions of 5 is higher than that of 10 [RMSE: 0.322 m/s (dims = 5) > 0.077 m/s (dims = 10)], but the estimation errors for lateral velocity are almost identical [RMSE: 0.047 m/s (dims = 5), 0.045 m/s (dims = 10)]. In addition, the 5-D one has less estimation error for the yaw rate [RMSE: 0.022 rad/s (dims = 5) < 0.032 rad/s (dims = 10)]. On the other hand, compared to longitudinal vehicle velocity, control strategies are more sensitive to numerical precision in lateral velocity and yaw rate. In light of the aforementioned considerations, this study suggests dimensions 5 for  $\Phi_k$ .

*Remark 4:* Once the dimensions of  $\Phi_k$  have been determined, an artificial neural network is built to map the lifting between  $x_k$  and  $\Phi_k$ . This network consists of three hidden layers, each containing 128 units. Additionally, two single-hidden layers, without activation and bias, are employed to represent  $\mathbf{A}$  and  $\mathbf{B}$  as introduced in Section II-A. The “Min-Max” method is used to normalize data in order to reduce standard deviations. The steepest descent is chosen for training based on the gradient of the loss function with respect to the network parameters.

#### IV. VALIDATION AND RESULTS

In this section, we compare the accuracy and robustness of DK operator models with and without ESO throughout conditions different from training data. Next, we evaluate the tracking performance and computational efficacy of the ESO-DKMPc on various double-lane change scenarios that induce both linear and nonlinear responses.

##### A. Model Accuracy and Robustness

Xiao et al. [27] have discussed and verified the superiority of the Koopman operator model in system modeling over other popular data-driven modeling methods. The DNN-informed Koopman operator model with or without the ESO, denoted as ESO-DK and DK, respectively, in Fig. 7.

The initial longitudinal velocity is 40 km/h, the nominal sprung mass of the vehicle is 1648 kg, and the nominal road adhesion coefficient is 0.85 [Fig. 7(a)]. Next, we modify the adhesion coefficient to 0.55 [Fig. 7(b)] and the sprung mass to +200 kg over the nominal value [Fig. 7(c)], respectively, to further conduct the comparison.

Overall, the ESO-DK and DK offer promising prospects for modeling and predicting vehicle states under three conditions,

but the DK is inferior to the ESO-DK. For the longitudinal velocity, the DK model has notable deviations around 40, 100, and 150 s, while it exhibits contrary trends in Fig. 7(b). Similar observations can also be found in the results of the lateral velocity and yaw rate. On the other hand, the impact of road adhesion coefficient on the modeling accuracy surpasses that of increased mass, proving its sensitivity to friction changes in vehicle dynamics. Because of the online estimation of total disturbance in the lifted space by ESO, which is a complement to the learned DNN-informed Koopman operator model, the ESO-DK has enhanced model accuracy and robustness.

By incorporating ESO, the feature space is not limited to the collection data but is instead able to extend and generalize real scenarios. Therefore, the ESO-DK model is capable of capturing complex dynamics in real-time while maintaining a linear form, thus holding the potential for practical applications.

### B. Double-Lane Change Simulations

To further highlight the proposed ESO-DKMPc for trajectory tracking of autonomous vehicles, four different ways are outlined herein for comparison. Due to the implementation of a decoupling control structure for vehicle trajectory tracking, we emphasize the comparison of the performance of lateral motions. As a result, the proportional control (P control) for the longitudinal profile is utilized by all four methodologies. The primary differences reside in the accuracy of the vehicle dynamics model. Specifically, these distinctions manifest themselves in the following ways.

- 1) **LMPC:** the linear 2 DoF model (single-track model) is embedded to predict the vehicle states within the prediction horizon [36]. The linear model is widely carried out in real applications due to its simplicity and conciseness for MPC implementation by quadratic programming (QP) techniques, for example, *quadprog* [37].
- 2) **NMPC:** the nonlinear tire model—Magic Formula (MF), is employed to depict tire dynamics and later obtain the predictive vehicle states to follow the desired trajectory. The nonlinearities of tire forces have been proven effective for model accuracy enhancement but necessitate higher computational resources [38]. The constrained optimization problem is solved using CasADi/IPOPT toolbox [39].
- 3) **ESO-DKMPc:** the proposed method in this study. The vehicle states are lifted to the high-dimensional state space by the learned DK operator, and the total disturbances are also estimated using ESO as the complement to improve model fidelity in the lifted space. Based on (16)~(22), (15) is also solved by CasADi/IPOPT.
- 4) **DKMPc:** similar to 3). The learned DK operator model without ESO compensation is adopted to predict vehicle states in the lifted space and generate the control commands.

Moreover, we choose the typical double-lane change test with two different conditions: constant speed on the high- $\mu$  road and varying speed on the low- $\mu$  road, for validation and discussion. In the first scenario, the vehicle travels at a constant and moderate speed, resulting in a smoother steering response,

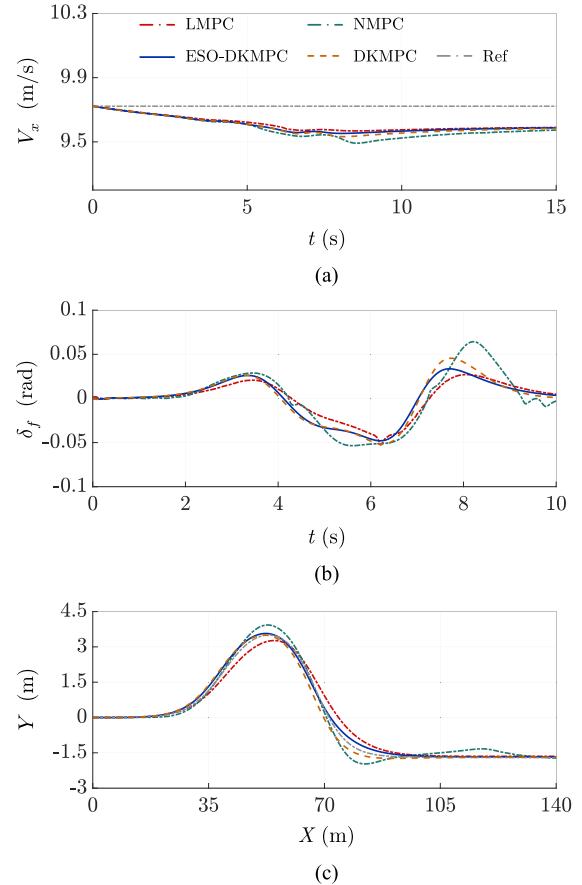


Fig. 8. Simulation results of constant speed on the high- $\mu$  road. (a) Longitudinal velocity. (b) Wheel steering angle. (c) Trajectory.

and operates in the linear region. On the other hand, the vehicle operates at a varying and high speed with aggressive steering input under the second scenario, which exhibits nonlinear behaviors. The comparisons enable us further to assess the performances and distinctions among the four methods.

1) *Constant Speed on the High- $\mu$  Road:* The adhesion coefficient of the road is 0.85, and the longitudinal velocity is maintained at 35 km/h. The comparison results are demonstrated in Fig. 8. In general, all four methods present good tracking performance on high- $\mu$  road in Fig. 8(b), exhibiting the ability to approximate vehicle dynamics in the linear regions. Specifically, the LMPC deviates more when the vehicle crosses the first curve. During the second lane change maneuver, the wheel steering angle of the NMPC [Fig. 8(c)] is greater than that of the LMPC, DKMPc, and ESO-DKMPc, and it fluctuates slightly after returning to the straight, resulting in an evident lateral deviation.

As a result of the coupling between the longitudinal and lateral dynamics of the vehicle, the lateral motions influence the longitudinal acceleration and, in turn, the longitudinal velocity. Consequently, the longitudinal velocity tracking corresponding to the four methods presents minor variations, but all are less than 0.3 m/s.

Regarding the lateral deviation error \$e\_Y\$ in Table II, the RMSE values of the ESO-DKMPc are 0.093 m. Whereas the RMSE values of \$e\_Y\$ for both the LMPC, NMPC, and

TABLE II

STATISTIC RESULTS OF CONSTANT SPEED ON THE HIGH- $\mu$  ROAD

	$e_Y$ (m)	$\Delta\varphi$ (rad)
	Max / Avg / RMSE	Max / Avg / RMSE
<b>LMPC</b>	0.606 / 0.136 / 0.225	0.104 / 0.025 / 0.039
<b>NMPC</b>	0.619 / 0.167 / 0.229	0.106 / 0.025 / 0.034
<b>DKMPC</b>	0.493 / 0.123 / 0.180	0.026 / 0.006 / 0.010
<b>ESO-DKMPC</b>	0.217 / 0.066 / 0.093	0.042 / 0.012 / 0.017

DKMPC are greater than 0.180 m. As for the yaw angle error  $\Delta\varphi$ , the ESO-DKMPC and DKMPC have comparable results with the RSME values less than 0.020 rad. In contrast, the RMSE values of the LMPC and NMPC exceed 0.030 rad. Incorporating with the total disturbance estimation of ESO, the DNN-informed DK operator model has improved the model fidelity, demonstrating its remarkable potential in representing vehicle dynamics.

2) *Varying Speed on the Low- $\mu$  Road:* The adhesion coefficient of the low- $\mu$  road is 0.5, and the desired longitudinal velocity varies from 45 to 55 km/h with an initial value of 50 km/h [Fig. 9(a)]. As the vehicle is expected to follow the trajectory with two curves, and the friction force is limited, both results in the vehicle and tires exhibiting nonlinear characteristics.

In Fig. 9(c), the tracking performance varies on the low- $\mu$  road. The LMPC method returns to the straight road with the most lateral deviation after the second curve. The corresponding wheel steering angle in Fig. 9(b) has an apparent oscillation and tends to diverge, indicating that the vehicle has almost lost stability. This is because the high speed on the low- $\mu$  curved road leads to the tire forces close to the adhesion limit, the linear vehicle dynamics model cannot adequately capture the nonlinear behaviors of the vehicle. Therefore, the LMPC is incapable of controlling the vehicle's lateral motions. As for the NMPC, the nonlinear model has the ability to identify the nonlinear characteristics in such conditions. The vehicle of the NMPC method can keep the stability but causes lateral deviations due to its limited model accuracy. Notably, although the wheel steering angle has a slight fluctuation in Fig. 9(b), the overall trajectory may still be controlled. The DKMPC enhances tracking performance, especially before  $X = 83$  m. Afterward, it exhibits less oscillations in steering angle and lateral deviation compared to the NMPC on the straight road. However, the proposed ESO-DKMPC is superior to the aforementioned three physics-based methods. Because of its high model fidelity to feature nonlinearities, the ESO-DKMPC is capable of stably following the desired path on the low- $\mu$  curved road. Combined with that in Section IV-B1, the results verify the feasibility and robustness of the proposed method. Besides, there are inevitable but acceptable tracking errors when applying the P controller for the longitudinal velocity profile in Fig. 9(a). The errors can be narrowed by calibrating the proportional coefficient, but this is not the focus of this study.

The maximum error of  $e_Y$  for the LMPC and NMPC are 1.752 and 1.174 m in Table III, respectively, while that of

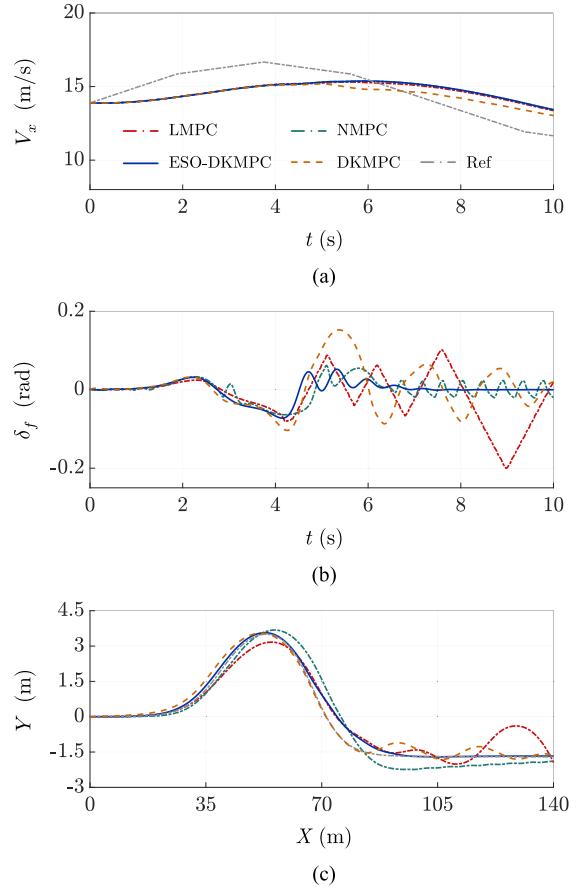


Fig. 9. Simulation results of varying speed on the low- $\mu$  road. (a) Longitudinal velocity. (b) Wheel steering angle. (c) Trajectory.

TABLE III  
STATISTIC RESULTS OF VARYING SPEED ON THE LOW- $\mu$  ROAD

	$e_Y$ (m)	$\Delta\varphi$ (rad)
	Max / Avg / RMSE	Max / Avg / RMSE
<b>LMPC</b>	1.752 / 0.345 / 0.498	0.416 / 0.064 / 0.104
<b>NMPC</b>	1.174 / 0.342 / 0.444	0.150 / 0.034 / 0.054
<b>DKMPC</b>	0.521 / 0.158 / 0.210	0.133 / 0.035 / 0.053
<b>ESO-DKMPC</b>	0.686 / 0.107 / 0.198	0.063 / 0.014 / 0.023

the ESO-DKMPC is 0.686 m. Besides, the RSME values of the ESO-DKMPC are also lower than those of the LMPC, NMPC, and DKMPC. In terms of  $\Delta\varphi$ , the maximum errors of the LMPC, NMPC, and DKMPC are 0.416, 0.150, and 0.133 rad, respectively, while the value for the ESO-DKMPC is 0.063 rad. Similarly, the RSME values of  $\Delta\varphi$  for the four methods have the same trends.

When the vehicle speed varies and is high on the low- $\mu$  curved road, the aggressive steering angle enforces the vehicle into the nonlinear region. The linear model, thereby, can hardly represent the nonlinear behaviors. Compared to the linear physics-based approach (LMPC), the nonlinear physics-based (NMPC) significantly elevates the model accuracy. However, the nonlinear physics-based model cannot accurately feature the nonlinearities and consists of a plurality of hyperparameters that are time-consuming to calibrate and obtain in

practical applications. In contrast, the deep learning-informed Koopman operator maps the nonlinearities in the original space to be a linear form in the lifted space by only collecting the input and output data, which proves significant superiority in system modeling. Moreover, the ESO achieves real-time identification of the total disturbance, including unmodelled dynamics and uncertainties in the lifted space, improving model accuracy beyond that of the DKMPC. This renders the proposed ESO-DKMPC feasible and effective under the vehicle working within linear and nonlinear regions.

### C. Computational Complexity

We compare the computation burden of four approaches in Sections IV-A and IV-B on Intel<sup>1</sup> Core<sup>2</sup> i7-CPU@1.10 GHz. Given that the network parameters of the DK operator have been finalized by offline training, the recording time does not include the training time. The average computational time: LMPC (2.36 ms) < DKMPC (261 ms) < ESO-DKMPC (370 ms) < NMPC (3076 ms). The computational complexity of NMPC highly increases as a result of its strong nonlinear optimization problem. Due to the linear evolution dynamics in the lifted space, the constrained finite-time optimization of ESO-DKMPC becomes a more computationally efficient problem. While the execution time of the proposed method is more than the LMPC, it is still an acceptable result that can be enhanced for the implementation of the controller. This confirms its applicability in real-world contexts.

## V. CONCLUSION

This article presents a control framework for autonomous vehicles using a DK operator model incorporating ESO. In terms of trajectory tracking, ESO is designed to online estimate the total disturbances, including modeling errors and residuals, as a complement to the learned DNN-informed Koopman operator model to feature vehicle planar dynamics. The model is then used to formulate the constrained finite-time optimization problem of the MPC scheme (ESO-DKMPC) to control the vehicle's lateral motions.

By incorporating ESO, the feature space is not limited to the collection data but is instead able to extend and generalize real scenarios. This demonstrates that the proposed model holds the potential for practical applications. Furthermore, two different conditions under the double-lane change scenario are built on the CarSim/Simulink co-simulation platform. Compared with the LMPC and NMPC informed by the physics-based model, the ESO-DKMPC exhibits superior tracking performance and robustness both within linear and nonlinear regions. The reason for this can be attributed to its higher model fidelity, which is a result of the learned DK operator and ESO in the lifted space.

Future research will the ESO-DKMPC in more challenging and critical scenarios involving higher speed and changing terrains on the hardware-in-the-loop or on-vehicle platform.

<sup>1</sup>Registered trademark.

<sup>2</sup>Trademarked.

## APPENDIX CONVERGENCE PROOF OF ESO

We define  $\mathbf{g}_1(\tilde{\mathbf{z}}_k) = \mathbf{g}_2(\tilde{\mathbf{z}}_k) = \tilde{\mathbf{z}}_k$ ,  $\tilde{\mathbf{w}}_k = \mathbf{w}_k - \hat{\mathbf{w}}_k$ , and assume  $\mathbf{w}_{k+1} = \mathbf{w}_k$ . Then, using (12)~(14), we have

$$\begin{cases} \tilde{\mathbf{z}}_{k+1} = (\mathbf{A}_\theta + \beta_1 \mathbf{I}) \tilde{\mathbf{z}}_k + \tilde{\mathbf{w}}_k \\ \tilde{\mathbf{w}}_{k+1} = \beta_2 \tilde{\mathbf{z}}_k + \tilde{\mathbf{w}}_k. \end{cases} \quad (23)$$

Further, the error space is denoted as

$$\begin{bmatrix} \tilde{\mathbf{z}}_{k+1} \\ \tilde{\mathbf{w}}_{k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{A}_\theta + \beta_1 \mathbf{I} & \mathbf{I} \\ \beta_2 \mathbf{I} & \mathbf{I} \end{bmatrix}}_{\Theta} \begin{bmatrix} \tilde{\mathbf{z}}_k \\ \tilde{\mathbf{w}}_k \end{bmatrix}. \quad (24)$$

If the spectral radius of  $\Theta$  is less than 1 by choosing appropriate  $\beta_1$  and  $\beta_2$ , i.e.,  $\rho(\Theta) < 1$ , then when  $k \rightarrow \infty$ , both the errors  $\tilde{\mathbf{z}}_k$  and  $\tilde{\mathbf{w}}_k$  converge to zero.

## REFERENCES

- [1] O. Nelles, *Nonlinear System Identification: From Classical Approaches to Neural Networks, Fuzzy Systems... and Gaussian Processes*. Berlin, Germany: Springer, 2020.
- [2] B. A. H. Vicente, S. S. James, and S. R. Anderson, "Linear system identification versus physical modeling of lateral-longitudinal vehicle dynamics," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 3, pp. 1380–1387, May 2021.
- [3] A.-T. Nguyen, T.-M. Guerra, C. Sentouh, and H. Zhang, "Unknown input observers for simultaneous estimation of vehicle dynamics and driver torque: Theoretical design and hardware experiments," *IEEE/ASME Trans. Mechatronics*, vol. 24, no. 6, pp. 2508–2518, Dec. 2019.
- [4] A. Alan, A. J. Taylor, C. R. He, A. D. Ames, and G. Orosz, "Control barrier functions and input-to-state safety with application to automated vehicles," *IEEE Trans. Control Syst. Technol.*, vol. 31, no. 6, pp. 2744–2759, Nov. 2023.
- [5] H. Chen, S. Lou, and C. Lv, "Hybrid physics-data-driven online modelling: Framework, methodology and application to electric vehicles," *Mech. Syst. Signal Process.*, vol. 185, Feb. 2023, Art. no. 109791.
- [6] M. Da Lio, D. Bortoluzzi, and G. P. Rosati Papini, "Modelling longitudinal vehicle dynamics with neural networks," *Vehicle Syst. Dyn.*, vol. 58, no. 11, pp. 1675–1693, Nov. 2020.
- [7] U. Rosolia and F. Borrelli, "Learning how to autonomously race a car: A predictive control approach," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 6, pp. 2713–2719, Nov. 2020.
- [8] H. Chen and C. Lv, "RHONN-modeling-based predictive safety assessment and torque vectoring for holistic stabilization of electrified vehicles," *IEEE/ASME Trans. Mechatronics*, vol. 28, no. 1, pp. 450–460, Feb. 2023.
- [9] N. A. Spielberg, M. Brown, N. R. Kapania, J. C. Kegelman, and J. C. Gerdes, "Neural network vehicle models for high-performance automated driving," *Sci. Robot.*, vol. 4, no. 28, Mar. 2019, Art. no. eaaw1975.
- [10] L. Martins, C. Cardeira, and P. Oliveira, "Feedback linearization with zero dynamics stabilization for quadrotor control," *J. Intell. Robot. Syst.*, vol. 101, no. 1, pp. 1–17, 2021.
- [11] J. Hu, P. Bhowmick, F. Arvin, A. Lanzon, and B. Lennox, "Cooperative control of heterogeneous connected vehicle platoons: An adaptive leader-following approach," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 977–984, Apr. 2020.
- [12] W. Xie, D. Cabecinhas, R. Cunha, and C. Silvestre, "Adaptive backstepping control of a quadcopter with uncertain vehicle mass, moment of inertia, and disturbances," *IEEE Trans. Ind. Electron.*, vol. 69, no. 1, pp. 549–559, Jan. 2022.
- [13] L. Zhang et al., "An adaptive backstepping sliding mode controller to improve vehicle maneuverability and stability via torque vectoring control," *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 2598–2612, Mar. 2020.
- [14] B. O. Koopman, "Hamiltonian systems and transformation in Hilbert space," *Proc. Nat. Acad. Sci. USA*, vol. 17, pp. 315–318, Jan. 1931.
- [15] J. L. Proctor, S. L. Brunton, and J. N. Kutz, "Generalizing Koopman theory to allow for inputs and control," *SIAM J. Appl. Dynamical Syst.*, vol. 17, no. 1, pp. 909–930, 2018.

- [16] D. Bruder, X. Fu, R. B. Gillespie, C. D. Remy, and R. Vasudevan, "Data-driven control of soft robots using Koopman operator theory," *IEEE Trans. Robot.*, vol. 37, no. 3, pp. 948–961, Jun. 2021.
- [17] M. Korda and I. Mezić, "Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control," *Automatica*, vol. 93, pp. 149–160, Jul. 2018.
- [18] S. Peitz and S. Klus, "Koopman operator-based model reduction for switched-system control of PDEs," *Automatica*, vol. 106, pp. 184–191, Aug. 2019.
- [19] P. Bevanda, S. Sosnowski, and S. Hirche, "Koopman operator dynamical models: Learning, analysis and control," *Annu. Rev. Control*, vol. 52, pp. 197–212, 2021.
- [20] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *J. Fluid Mech.*, vol. 656, pp. 5–28, Jul. 2010.
- [21] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*. Philadelphia, PA, USA: SIAM, 2016.
- [22] J. H. Tu, "Dynamic mode decomposition: Theory and applications," Ph.D. dissertation, Dept. Mech. Aerosp. Eng., Princeton Univ., Princeton, NJ, USA, 2013.
- [23] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, "A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition," *J. Nonlinear Sci.*, vol. 25, no. 6, pp. 1307–1346, Dec. 2015.
- [24] S. Klus, F. Nüske, S. Peitz, J.-H. Niemann, C. Clementi, and C. Schütte, "Data-driven approximation of the Koopman generator: Model reduction, system identification, and control," *Phys. D, Nonlinear Phenomena*, vol. 406, May 2020, Art. no. 132416.
- [25] B. Lusch, J. N. Kutz, and S. L. Brunton, "Deep learning for universal linear embeddings of nonlinear dynamics," *Nature Commun.*, vol. 9, no. 1, p. 4950, Nov. 2018.
- [26] Y. Han, W. Hao, and U. Vaidya, "Deep learning of Koopman representation for control," in *Proc. IEEE Conf. Decis. Control*, Dec. 2020, pp. 1890–1895.
- [27] Y. Xiao, X. Zhang, X. Xu, X. Liu, and J. Liu, "Deep neural networks with Koopman operators for modeling and control of autonomous vehicles," *IEEE Trans. Intell. Vehicles*, vol. 8, no. 1, pp. 135–146, Jan. 2023.
- [28] H. Yang, H. Cheng, Z. Zuo, and H. Li, "ESO-based lateral control for electrical vehicles with unmodeled tire dynamics on uneven road," *Mech. Syst. Signal Process.*, vol. 177, Sep. 2022, Art. no. 109132.
- [29] J. Guerrero, J. Torres, V. Creuze, and A. Chemori, "Adaptive disturbance observer for trajectory tracking control of underwater vehicles," *Ocean Eng.*, vol. 200, Mar. 2020, Art. no. 107080.
- [30] C. Ren, H. Jiang, C. Li, W. Sun, and S. Ma, "Koopman-operator-based robust data-driven control for wheeled mobile robots," *IEEE/ASME Trans. Mechatronics*, vol. 28, no. 1, pp. 461–472, Feb. 2023.
- [31] X. Zhang, A. Liniger, and F. Borrelli, "Optimization-based collision avoidance," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 3, pp. 972–983, May 2021.
- [32] J. Wang, Y. Zheng, K. Li, and Q. Xu, "DeeP-LCC: Data-enabled predictive leading cruise control in mixed traffic flow," *IEEE Trans. Control Syst. Technol.*, vol. 31, no. 6, pp. 2760–2776, Nov. 2023.
- [33] R. N. Jazar, *Advanced Vehicle Dynamics*. Berlin, Germany: Springer, 2019.
- [34] M. Bartolozzi, G. Savino, and M. Pierini, "Novel high-fidelity tyre model for motorcycles to be characterised by quasi-static manoeuvres—Rationale and numerical validation," *Vehicle Syst. Dyn.*, vol. 60, no. 12, pp. 4290–4316, Dec. 2022.
- [35] H. Pacejka, *Tire and Vehicle Dynamics*. Amsterdam, The Netherlands: Elsevier, 2005.
- [36] 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.
- [37] J. Nocedal and S. J. Wright, "Quadratic programming," in *Numerical Optimization*. New York, NY, USA: Springer, 2006, pp. 448–492.
- [38] P. Stano et al., "Model predictive path tracking control for automated road vehicles: A review," *Annu. Rev. Control*, vol. 55, pp. 194–236, Jan. 2023.
- [39] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi: A software framework for nonlinear optimization and optimal control," *Math. Program. Comput.*, vol. 11, no. 1, pp. 1–36, Mar. 2019.