Preprints, 10th IFAC International Symposium on
Advances in Automotive Control
Columbus, Ohio, USA, August 28-30, 2022

Autonomous Driving using Linear Model Predictive Control with a Koopman Operator based Bilinear Vehicle Model

Siyuan Yu [*], **Congkai Shen** [*], **Tulga Ersal** [**]

[*] *These authors contributed equally to this work.*
[**] *Corresponding author: tersal@umich.edu*
Mechanical Engineering, University of Michigan, Ann Arbor, MI 48109

Abstract: This paper presents a real-time Model Predictive Control (MPC) formulation for autonomous driving based on a lifted bilinear vehicle model developed using the Koopman operator. Koopman operator based models can closely mimic the original nonlinear behaviors with a higher dimensional linear structure, which is attractive for computationally efficient linear MPC formulations for controlling nonlinear systems. However, current linear models based on linear Koopman realizations cannot capture the control-affine dynamics in nonlinear systems. This may result in large discrepancies between the original nonlinear system and the data-driven linear model, hindering its use in MPC. To address this gap, first, a novel Koopman bilinear vehicle model that takes control-affine dynamics into consideration is constructed and tested in open-loop simulations. This bilinear Koopman model is then linearized to serve as a prediction model in MPC, and is shown to have higher accuracy compared to the state-of-the-art linear models. The model is then used to develop a linear MPC formulation for simultaneous planning and control of an autonomous vehicle. The formulation is tested on lane change scenarios with obstacles against the nonlinear MPC and standard linear MPC benchmarks. The results show that the new formulation can achieve a lane change performance closer to the nonlinear MPC with a computational performance similar to the standard linear MPC. The new formulation is observed to be successful in handling high speeds where the standard linear MPC fails.

Keywords: Koopman operator, bilinear systems, real-time model predictive control, autonomous driving

1. INTRODUCTION

Generating optimal and safe trajectories in real time is a persisting challenge in autonomous driving. One of the methods used to address this challenge is model predictive control (MPC) due to its rigor and flexibility with incorporating constraints and performance metrics (Liu et al., 2017). MPC has been adopted in various autonomous driving scenarios including obstacle avoidance (Wurts et al., 2021) and lane keeping (Bujarbaruah et al., 2018). However, as nonlinearities become more important, such as when vehicles are driven to their limits in emergency maneuvers (Wurts et al., 2020, 2021), the formulations become more difficult to solve reliably in real time (Febbo, 2019). Even though recent advances in formulating and solving nonlinear optimal control problems address this challenge to a certain extent (Wurts et al., 2021, 2022), a reliable real-time performance is still an open research challenge.

Compared to nonlinear models, linear models offer greater advantage in terms of computational efficiency (Gros et al., 2020). Typical linear models rely on constructing the system matrix from the Jacobian. These kinds of models cannot maintain high fidelity as the states move away from the point about which linearization is performed and nonlinearities become significant. This makes them less suitable for MPC as prediction horizons need to become longer. Another track for building linear MPC for vehicle control is to have an initial guess for the changing states and use the average to replace the nonlinear terms (Turri et al., 2013). This method increases the range of validity of the model, but the accuracy relies on the initial guess.

The Koopman operator, which can generate globally linearized models, has been adopted to increase model fidelity, as the nonlinear model can be approximated better when the linear state space is lifted to a higher dimension (Koopman, 1931). Successfully applied to a variety of dynamic models, this method offers a strong ability to interpret nonlinear models as higher-dimensional linear ones (Mezić, 2013). To build this invariant linear operator, Extended Dynamic Mode Decomposition (EDMD) and Dynamic Mode Decomposition (DMD) have been noted as popular tools (Williams et al., 2015). In the automotive domain, researchers have adopted EDMD to approximate Koopman operators in MPC formulations for vehicle control. A linear vehicle model is developed by Cibulka et al. (2019) using linear Koopman realizations to approximate an ordinary bicycle model with slip ratio and steering angle as control inputs. To further improve the model performance, deep-learning based techniques are applied in the process to determine the desired basis functions so that the state space representation of the model can find a balance between model fidelity and complexity (Han et al., 2020).

The standard Koopman operator faces a trade-off between model fidelity and computational speed in MPC. On the one hand, linear Koopman models can have fast computation speed in MPC, but have low accuracy when the original states and controls are coupled. On the other hand, nonlinear Koopman models can be more accurate for modeling, but they result in nonlinear MPC, which is computationally less efficient. Bilinear models constructed by Bruder et al. (2021) through the bilinear Koopman realization aim to address this trade-off fundamentally. A bilinear model enjoys computational efficiency to some extent. However, few work has been done in the field of bilinear optimal control. Cebuhar and Costanza (1984) and Aganovic and Gajic (1994) deduced an iterative process of unconstrained bilinear optimal control based on Lyapunov's second method and the Hamilton-Jacobi equation, respectively. Halperin et al. (2020) extended the problem to Krotov's method with constraints in control inputs. However, no state constraints are involved in current bilinear optimal control frameworks, which makes this strategy infeasible for controlling a vehicle.

To address this gap, this paper approximates the bilinear Koopman model further by keeping the coupling state value constant during the prediction to form a linear model. This approximation is then incorporated with constraints and maintains high computational efficiency along with high fidelity in linear MPC.

The original contributions are summarized as follows:

(1) A bilinear Koopman based vehicle dynamics model that takes the control-affine dynamics into account.
(2) A strategy to linearize the bilinear Koopman model by approximating the coupling state as constant in the prediction horizon.
(3) A comparison between the fidelity of the bilinear Koopman model, linearized bilinear Koopman model, linear Koopman model and locally linear model to show the importance of the control-affine dynamics.
(4) A linear MPC framework using the linearized bilinear Koopman model as the prediction model for simultaneous planning and control in autonomous driving, with comparisons to state-of-the-art nonlinear and linear approaches.

2. VEHICLE DYNAMICS

A 3-degrees-of-freedom (DOF) bicycle model with linear tire forces is used as the original nonlinear vehicle model to provide a large set of simulated data. This model is widely used in MPC formulations due to its balanced simplicity and fidelity (Liu et al., 2016), and is thus adopted here, as well.

The state and control vectors of the model are defined as:

$$\xi := \begin{bmatrix} x \\ y \\ r \\ v \\ \psi \\ u \end{bmatrix} = \begin{bmatrix} \text{global x position of CG} \\ \text{global y position of CG} \\ \text{yaw rate} \\ \text{lateral speed} \\ \text{yaw angle} \\ \text{longitudinal speed} \end{bmatrix} \tag{1}$$

$$\zeta := \begin{bmatrix} a_x \\ \delta_f \end{bmatrix} = \begin{bmatrix} \text{longitudinal acceleration} \\ \text{front tire steering angle} \end{bmatrix} \tag{2}$$

The vehicle model follows the dynamics expressed as:

$$\dot{\xi} = A(\xi) + B\zeta \tag{3}$$

where

$$A(\xi) = \begin{bmatrix} u\cos\psi - v\sin\psi \\ u\sin\psi + v\cos\psi \\ (F_{yf}L_f - F_{yr}L_r)/I_{zz} \\ (F_{yf} + F_{yr})/M - ur \\ r \\ 0 \end{bmatrix} \tag{4}$$

$$B^T = \begin{bmatrix} 0\ 0\ 0\ 0\ 0\ 1 \\ 0\ 0\ 0\ 0\ 0\ 0 \end{bmatrix} \tag{5}$$

Here, F_{yf} and F_{yr} are the lateral forces acting on the front and rear tires. M is the vehicle mass, I_{zz} is its yaw moment of inertia. L_f and L_r are distances from the center of gravity (CG) to front and rear axles, respectively.

In this model, the vehicle vertical load is needed to calculate the lateral tire forces. The front and rear vertical loads F_{zf} and F_{zr} are considered as steady state forces as:

$$F_{zf} = \frac{ML_f g}{L_f + L_r} \tag{6}$$

$$F_{zr} = \frac{ML_r g}{L_f + L_r} \tag{7}$$

where g is the acceleration of gravity. The front and rear cornering stiffnesses C_{α_f} and C_{α_r} are linear coefficients derived from the linear region of the Pacejka magic formula (Pacejka, 2005). The front and rear tire slip angles α_f, α_r and the lateral forces F_{yf}, F_{yr} are calculated as:

$$\alpha_f = \arctan\left(\frac{v + L_f r}{u}\right) - \delta_f \tag{8}$$

$$\alpha_r = \arctan\left(\frac{v - L_r r}{u}\right) \tag{9}$$

$$F_{yf} = C_{\alpha_f} F_{zf} \alpha_f \tag{10}$$

$$F_{yr} = C_{\alpha_r} F_{zr} \alpha_r \tag{11}$$

3. KOOPMAN OPERATOR

Let $\mathbf{F}$ denote the nonlinear function of the bicycle dynamics.

$$\dot{\xi}(t) = A(\xi(t)) + B\zeta(t) = \mathbf{F}(\xi(t), \zeta(t)) \tag{12}$$

Let $\mathbb{F}$ be an infinite-dimensional function space composed of all square-integrable real-valued functions. Each element $f \in \mathbb{F}$ is called an observable. In $\mathbb{F}$, the time evolution of each observable is characterized by the Koopman operator $\mathbb{K}$.

$$\mathbb{K}f(\xi(t), \zeta(t)) = \frac{\partial f}{\partial \xi}\frac{d\xi}{dt} = \frac{\partial f}{\partial \xi}\mathbf{F}(\xi(t), \zeta(t)) \tag{13}$$

The model realization is fulfilled by approximating the Koopman operator in a finite dimensional subspace. The approximation is based on the Extended Dynamic Mode Decomposition (EDMD) algorithm developed by Williams et al. (2015). Let $\Phi(\bullet)$ be the lifting function, which transforms the original state space and control space into the Koopman operator subspace.

$$\begin{aligned} p^{(n)} &= \Phi\left(\xi^{(n)}(t), \zeta^{(n)}(t)\right) \\ q^{(n)} &= \frac{d}{dt}\left(\Phi(\xi^{(n)}(t), \zeta^{(n)}(t)\right) \\ &= \frac{\Phi\left(\xi^{(n)}(t+\Delta t), \zeta^{(n)}(t+\Delta t)\right) - \Phi\left(\xi^{(n)}(t), \zeta^{(n)}(t)\right)}{\Delta t} \end{aligned} \tag{14}$$

where $n \in \{1, 2, \ldots N\}$ denotes the index of sampling, $p^{(n)}$ denotes the lifted vector in the subspace from n^{th} sampling data, and $q^{(n)}$ denotes the corresponding time derivatives of $p^{(n)}$. Given the N samplings and their lifted values, the subspace of the Koopman operator is linearly regressed as

$$\min_{\mathbb{K}} \|\mathbb{K}\mathbf{P} - \mathbf{Q}\|_F^2 \tag{15}$$

$$\mathbf{P} = \left[p^{(1)}\ p^{(2)}\ \cdots\ p^{(N)}\right] \tag{16}$$

$$\mathbf{Q} = \left[q^{(1)}\ q^{(2)}\ \cdots\ q^{(N)}\right] \tag{17}$$

with the resulting optimizer

$$\mathbb{K}^* = \mathbf{Q}\mathbf{P}^\dagger \tag{18}$$

where $\mathbb{K}^*$ is the approximated continuous Koopman operator, $\mathbf{P}$ and $\mathbf{Q}$ are the matrices horizontally stacking all p and q, respectively, and $\dagger$ denotes the Moore pseudo inverse of a matrix.

3.1 Linear Koopman Model Realization

Common Koopman realization constructs linear models by lifting the model to a finite degree as an approximation of infinite dimensional Koopman operators. Linear Koopman realization creates a linear model that only lifts the state space. The lifting process of the linear Koopman realization for the vehicle states except for x and y is given as:

$$\beta_j(\xi,\zeta)_{j=1}^{N_\beta} \in \left\{ \prod_{i=3}^{6} \xi_i^{\rho_i} \middle| \sum_{i=3}^{6} \rho_i \le \rho, \forall \rho_i \ge 0 \right\} \tag{19}$$

where N_β is the dimension of subspace where only the original states are lifted. The global position x and y are not lifted due to their absence when calculating the derivatives of other states. The strategy is to lift the basis function ξ to higher order polynomials. $\beta(\xi,\zeta)$ are basis functions where the polynomial order is less than or equal to a pre-defined degree ρ. Then, the linear finite dimensional subspace Φ_L is derived as follows by augmenting the global position x, y at the beginning and the control inputs ζ at the end:

$$\Phi_L = \begin{bmatrix} x \; y \; \beta_1(\xi,\zeta) \; \cdots \; \beta_{N_\beta}(\xi,\zeta) \; \zeta^T \end{bmatrix}^T \tag{20}$$

3.2 Bilinear Koopman Model Realization

Bilinear systems have the form

$$\dot{\tilde{\xi}} = \tilde{A}\tilde{\xi} + \tilde{B}\tilde{\zeta} + \sum_{j=1}^{\tilde{n}} \tilde{\xi}_j \tilde{N}_j \tilde{\zeta} \tag{21}$$

where $\tilde{\xi} \in \mathbb{R}^{\tilde{n}\times 1}, \tilde{\zeta} \in \mathbb{R}^{\tilde{m}\times 1}, \tilde{A} \in \mathbb{R}^{\tilde{n}\times\tilde{n}}, \tilde{B} \in \mathbb{R}^{\tilde{n}\times\tilde{m}}$. $\tilde{N}_j \in \mathbb{R}^{\tilde{n}\times\tilde{m}}$ is a constant matrix that defines the coefficients in the control affine terms, j is the index of the states, and $\tilde{m}$ and $\tilde{n}$ are the number of control inputs and states in the lifted space, respectively. The difference between a linear model and a bilinear one is the control-affine dynamics defined as the last term. The bilinear Koopman realization is constructed in a similar way to Sec. 3.1; the difference is that instead of directly augmenting the control inputs, the formulation further considers the control-affine dynamics in Φ_B. This is expressed as:

$$\Phi_B = [\Phi_{B,\xi} \; \Phi_{B,\xi}\zeta_1 \; \Phi_{B,\xi}\zeta_2]^T \tag{22}$$

$$\Phi_{B,\xi} = [x \; y \; \beta_1(\xi,\zeta) \; \cdots \; \beta_{N_\beta}(\xi,\zeta)]^T \tag{23}$$

3.3 Sampling Strategy

For the training sets, 5000 trajectories are simulated from the original vehicle model described in Sec. 2. The sampling horizon is defined as 0.01s. Each trajectory is simulated with uniformly distributed randomized initial conditions and control inputs where the bounds are set as the same with Sec. 4.2.

$$\xi_{\text{min}} \le \xi(t) \le \xi_{\text{max}} \tag{24}$$

$$\zeta_{\text{min}} \le \zeta(t) \le \zeta_{\text{max}} \tag{25}$$

By stacking the values of state evolutions and control inputs, the values of $\mathbf{P}$ and $\mathbf{Q}$ are calculated through the steps given above.

4. MODEL PREDICTIVE CONTROL

This section describes the formulation of a single-layer MPC; i.e., one that solves the planning and control problems simultaneously. The control inputs in each receding horizon are generated based on the optimal control problem (OCP):

$$\underset{\xi,\,\zeta,\,t_f}{\text{minimize}} \quad J = \int_0^{t_f} \mathcal{I}[\xi(t),\zeta(t)]dt \tag{26}$$

$$\text{subject to} \quad \dot{\xi}(t) = \mathcal{V}[\xi(t),\zeta(t)] \tag{27}$$

$$\xi_{\text{min}} \le \xi(t) \le \xi_{\text{max}} \tag{28}$$

$$\zeta_{\text{min}} \le \zeta(t) \le \zeta_{\text{max}} \tag{29}$$

where J is the cost function, $\mathcal{I}$ is the function that contains the cost terms on states and controls, and $\mathcal{V}$ is the vehicle dynamics. In the simulations, the locally linearized, the nonlinear and the linearized bilinear models are implemented for comparison. The prediction horizon N_p is set to 3s, the control horizon N_c is 0.5s, and the sampling horizon N_s is 0.1s.

4.1 Linearization of the Bilinear Model

The bilinear model itself is a nonlinear model due to the control-state coupling term. To transform the bilinear form in (21) into a linear form, the values of states in the control-affine term are kept constant within each receding horizon at its initial value, and a linear form is obtained as follows:

$$\dot{\tilde{\xi}} = \bar{A}\tilde{\xi} + \bar{B}\tilde{\zeta} = \tilde{A}\tilde{\xi} + [\tilde{B} + \sum_{j=1}^{\tilde{n}} \tilde{\xi}_j(0)\tilde{N}_j]\tilde{\zeta} \tag{30}$$

where $\bar{A}$, and $\bar{B}$ are the linearized forms of state matrices, and $\tilde{\xi}_j(0)$ is the initial value of the j^{th} element in $\tilde{\xi}$ in each receding horizon. Compared to the locally linearized model, the linearized bilinear Koopman model can offer higher accuracy due to the lifted space. Its advantage over the linearized Koopman model is its accounting for changing states in the bilinear term at the beginning of each receding horizon.

4.2 Control and State Constraints

The bounds of control inputs are determined to meet the limits of the actuators on the vehicle. The control constraints of (29) are expanded as:

$$\delta_{f_{\text{min}}} \le \delta_f \le \delta_{f_{\text{max}}} \tag{31}$$

$$a_{x_{\text{min}}} \le a_x \le a_{x_{\text{max}}} \tag{32}$$

The vehicle states are constrained through linear state constraints as follows:

$$y_{\text{min}} \le y \le y_{\text{max}} \tag{33}$$

$$\psi_{\text{min}} \le \psi \le \psi_{\text{max}} \tag{34}$$

$$u_{\text{min}} \le u \le u_{\text{max}} \tag{35}$$

To reach a collision-free trajectory, the drivable tube concept of Wurts et al. (2021) is used to prevent the vehicle from entering hazardous zones. The drivable tube is defined for the vehicle CG position; i.e., as long as the CG stays in the drivable tube, the trajectory is deemed collision free.

4.3 Plant Model

In the simulations, the vehicle model in Sec. 2 is utilized as the plant model. The rationale for this choice instead of using a higher fidelity plant model is twofold: (1) to create a controlled experiment by avoiding model discrepancies, so that the performance differences between the proposed and benchmark MPC formulations can be studied in isolation; and (2) to create a best-performance benchmark with nonlinear MPC using identical plant and prediction models.

4.4 Cost Function

The cost function is formulated as

$$J = \int_{t_0}^{t_f} \left(w_{\delta_f}\delta_f(t)^2 + w_{a_x}a_x(t)^2 + w_y\epsilon_y(t)^2 + w_\psi\epsilon_\psi(t)^2 \right) dt \tag{36}$$

where w_{δ_f}, w_{a_x}, w_y, w_ψ are weights corresponding to each term. The first and second terms minimize the control efforts. The third term minimizes the lane keeping error as captured by the y-position error and the last term minimizes the heading error of the vehicle.

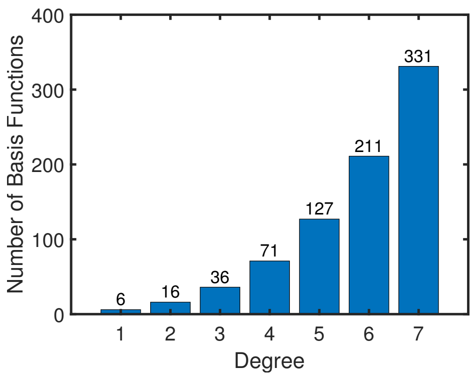

Fig. 1. Number of basis functions used in lifting for Koopman operator based bilinear systems

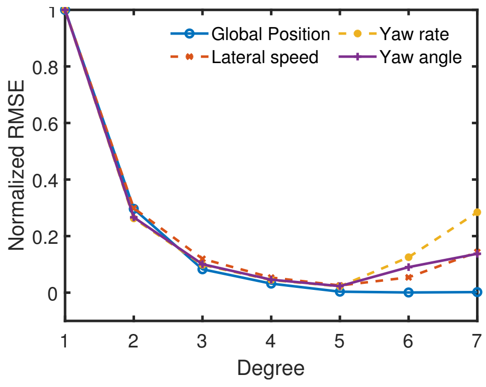

Fig. 2. Normalized root mean square error ($\tilde{\epsilon}_\rho^m$) in vehicle states. The RMSE is normalized based on the performance of the bilinear system of degree $\rho = 1$.

5. RESULTS

5.1 Model Fidelity

As described in Sec. 3.1, a pre-defined polynomial degree ρ needs to be set to determine the finite dimensional subspace Φ. As the degree ρ increases, the dimension of the Koopman operator explodes as shown in Fig. 1.

An increasing number of basis functions suggests higher computational burden both on training and control, whereas a small number of basis functions cannot capture the nonlinearities well. To find a balance, 100 testing scenarios are formulated in the same way as shown in Sec. 3.3, but with completely different initial states and control inputs. From (4) and (5), the derivative of longitudinal speed u is already a linear combination of control inputs. Thus, the discussion of errors in this term is of no use. Consequently, four kinds of errors are studied in the testing: global position (x,y), yaw angle ψ, yaw rate r and lateral speed v. Root mean square error (RMSE) is applied to the four errors denoted as ϵ_ρ^{xy}, ϵ_ρ^r, ϵ_ρ^v and ϵ_ρ^ψ. A normalization is applied as

$$\tilde{\epsilon}_\rho^m = \frac{\epsilon_\rho^m}{\epsilon_1^m} \quad \forall \rho \in \{1, 2, \cdots, 7\} \tag{37}$$

where m is a placeholder for the four error metrics.

In Fig. 2, four normalized RMSEs are plotted. The error in global position keeps decreasing as the degree increases, because the higher dimensional polynomials can better fit the $\sin$ and $\cos$ functions. The errors in yaw angle and yaw rate nearly reach the minimum at $\rho = 5$. The increase in their errors for $\rho > 5$ is attributed to over-fitting, which assigns improper weights on the high degrees. Judging from the overall performance and the computational efficiency, a degree of $\rho = 5$ is chosen to construct the bilinear and the linearized bilinear systems.

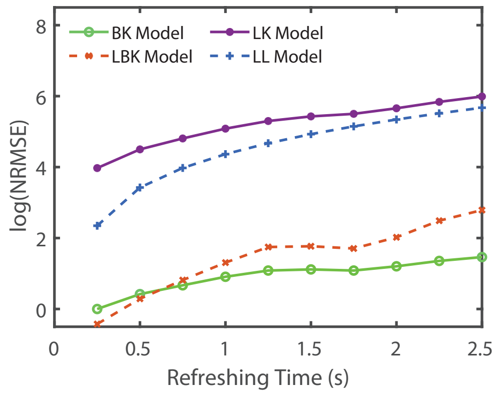

Fig. 3. Normalized root mean square error (NRMSE) on vehicle states. The RMSE of each scenario is normalized based on the performance of the Bilinear Koopman system under 0.25s refreshing time.

A comparison of model fidelity is shown in Fig. 3. The refreshing time is the time interval when the error of states are corrected to zero. In this comparison, four types of models are constructed:

(1) BK system: Bilinear Koopman system with degree $\rho = 5$.
(2) LL system: Locally linearized system described in (38). This serves as a non-lifted benchmark.
(3) LK system: Linear Koopman system described in Sec. 3.1.
(4) LBK system: Linearized bilinear Koopman system in (30).

$$\dot{\xi} = F(\xi(t_0), \zeta(t_0)) + \left.\frac{\partial F}{\partial \xi(t)}\right|_{t=t_0} (\xi - \xi(t_0)) + \left.\frac{\partial F}{\partial \zeta(t)}\right|_{t=t_0} (\zeta - \zeta(t_0)) \tag{38}$$

The four errors are then normalized based on the error of the BK system under 0.25s refreshing time:

$$\tilde{\epsilon}_{l,t}^m = \frac{\epsilon_{l,t}^m}{4\epsilon_{\mathrm{B},0.25}^m} \quad \forall t \in \{0.25, 0.5, \cdots, 2.5\} \tag{39}$$

where l is a placeholder for the four systems (BK, LL, LK and LBK). The testing sets are kept consistent with the ones used to determine the system degree.

In Fig. 3, the four normalized errors are summed to obtain the total normalized RMSE, to which then a natural logarithm is applied. The BK system and the LBK system have compatible performance due to their accuracy in predicting yaw rate r and lateral speed v. The divergence shown between the two systems after 1s is due to the unchanged state values in control-affine dynamics in (30).

In Table 1, the normalized RMSEs are listed for each error using the same metric as Fig. 3 with 0.5s and 1.5s refreshing time. The BK and LBK models provide good global linearizations, as the errors accumulate only slowly as the refreshing time increases, whereas errors in LL increase dramatically. The LK model has the largest errors in all scenarios due to the lack of consideration of control-affine dynamics.

A notable difference between the LBK and BK models lies in the prediction of global position when refreshing time is 1.5s, where the LBK model is 3 times worse than the BK model. The fixed state values in control-affine terms miss the details in dynamics, but the formulation can still capture the overall

Table 1. Normalized root mean square error (NRMSE) of vehicle states under 0.5s and 1.5s refreshing time

	$\tilde{\epsilon}^{(x,y)}$	$\tilde{\epsilon}^{v}$	$\tilde{\epsilon}^{r}$	$\tilde{\epsilon}^{\psi}$
BK model (0.25s)	0.25	0.25	0.25	0.25
BK model (0.5s)	0.49	0.26	0.26	0.51
LBK model (0.5s)	0.53	0.15	0.29	0.36
LL model (0.5s)	3.78	13.32	5.04	8.56
LK model (0.5s)	21.93	30.54	17.48	20.03
BK model (1.5s)	1.36	0.26	0.26	1.17
LBK model (1.5s)	3.69	0.24	0.44	1.49
LL model (1.5s)	35.78	30.23	10.09	62.06
LK model (1.5s)	74.31	54.15	23.91	75.50

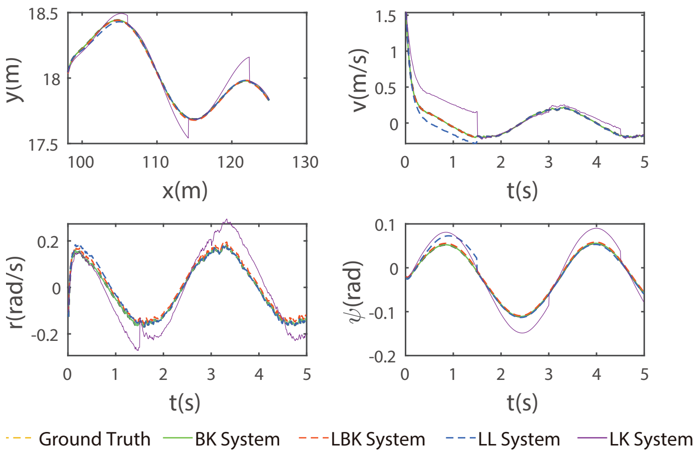

Fig. 4. Vehicle states under different controllers and refreshing time of 1.5s

trend of dynamics evolution. Fig. 4 demonstrates an open-loop simulation, where the steering angle commands are given as

$$\delta_f = \delta_{f_{\max}} \cos(k_1) e^{-k_2} \tag{40}$$

where $k_1 \sim U(0.02, 0.05)$ and $k_2 \sim U(10^{-2}, 10^{-3})$ are uniformly distributed random variables, and the acceleration commands are given as Gaussian distributed ($a_x \sim N(0, 3)$ m/s^2) random numbers within the control bound. In the testing, the refreshing horizon is given as 1.5s and the simulation runs for 5s. The ground truth trajectory is obtained by feeding the control signals to the model in Sec. 2. From the plots, the LK model deviates from the ground truth while the LL model can maintain high fidelity for approximately 0.5s before exhibiting errors. The LBK model does not create noticeable discrepancies from the BK model and aligns closely with the ground truth.

Based on these observations, the LBK system shows good potential as a prediction model in MPC, which is studied next.

5.2 MPC

Three controllers are tested to gain insight into the control performance that the LBK model can produce. The nonlinear MPC (NL MPC) and LL model based MPC (LL MPC) are utilized as benchmarks for the LBK model based MPC (LBK MPC). The LBK MPC and LL MPC are solved through quadratic programming. The NL MPC uses the midpoint collocation method, which is solved through CasADi (Andersson et al., 2019) using IPOPT (Wächter and Biegler, 2006). All simulations are run in Matlab 2020b with the same settings.

The scenario is designed such that the vehicle needs to perform evasive maneuvers before completing a lane change to the right ($y = 0$). Two experiments are conducted as follows.

In the first experiment, three hazard zones are placed at $x \in [21\,,\ 30] \cup [60\,,\ 70] \cup [100\,,\ 120]$ as seen in Fig. 5 and Fig. 6.

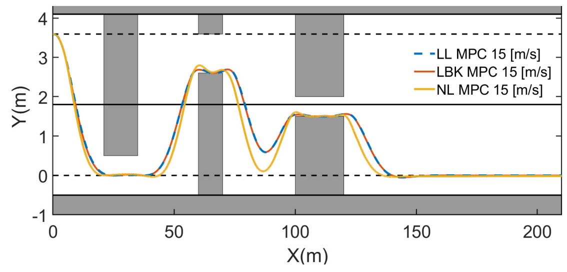

Fig. 5. Vehicle paths for low speed case

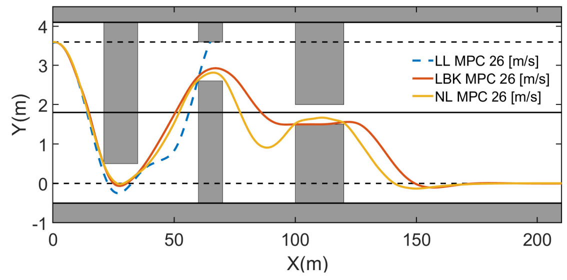

Fig. 6. Vehicle paths for high speed case

Table 2. Randomized scenario setting

	High Speed	Low speed
u_0	[21 , 23] m/s	[13 , 17] m/s
$x_{h,1}$	[22 , 25] m	[27 , 30] m
d_h	[35 , 38] m	[40 , 45] m

The initial states are set as $\xi_0 = [0\ 3.6\ 0\ 0\ 0\ u_0]^T$, where two initial speeds of $u_0 = 15$m/s and $u_0 = 26$m/s are considered as low and high speed examples.

Results are shown in Fig. 5 and Fig. 6 for the low and high speed cases, respectively. When the speed is low, LL MPC and LBK MPC produce nearly identical responses. However, in the high speed case, LL MPC fails to turn the vehicle in time and hits the second hazard, while the LBK MPC successfully produces a safe maneuver. This contrast shows the necessity of consideration of nonlinearities in the prediction model in extreme maneuvers and the LBK model's ability to do so. In both cases, the NL MPC has the lowest cost as expected.

Next, the maximum initial speeds are increased for each controller until the plant model hits an hazard. For both NL MPC and LBK MPC, the maximum initial speed is recorded as 26.2m/s, while it is noted as 22.6m/s for the LL MPC. Despite the cost difference, LBK MPC is still capable of pushing the vehicle to the limit. To gain further confidence in the insights from this experiment, a statistical analysis is done next to test the generalizability.

The second experiment offers a statistical perspective. Two hazard zones are used, namely, the first two hazards of the first experiment. The length of both hazard zones are modified to 10m. The widths and the y positions of the openings in hazard zones are constant and the same as the first experiment. The x position of the first hazard zone $x_{h,1}$, and the distance between the first and second hazard zone d_h complete the description of the configuration of two hazard zones. Two speed ranges are considered to capture high and low speeds. For each case, the range of $x_{h,1}$, d_h, and the initial speed u_0 are specified in Table 2. Each case is randomized as a Latin hypercube of 50 samples. The result is plotted in Fig. 7.

Because comparing the costs across scenarios is not meaningful, the cost of NL MPC is treated as benchmark, based on

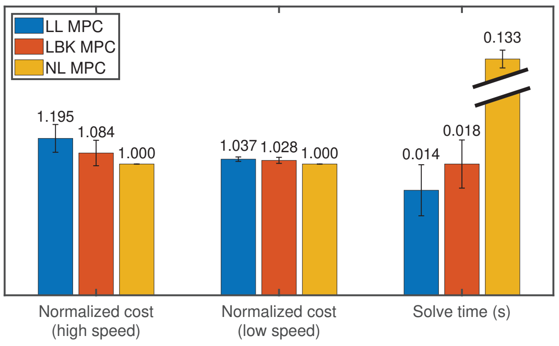

Fig. 7. Statistical performance analysis for the three controllers. The whiskers represent standard deviation.

which the other two controllers are normalized. In the first test, the LL MPC has about 10.3% higher cost than the LBK MPC, whereas the NL MPC is 7.7% lower. This shows that the LBK MPC better controls the vehicle than LL MPC when the vehicle is near the limits. In the second test, the three controllers all perform similarly as indicated by the negligible cost difference. As for the computational cost, the solve time difference between LL MPC and LBK MPC is only 4ms on average and they are both one order of magnitude smaller than the NL MPC with 133ms average solve time.

6. CONCLUSION

This work presents an MPC formulation with a bilinear Koopman operator based vehicle model for real-time trajectory planning for autonomous driving. A bilinear Koopman realization for the vehicle dynamics is presented first by transferring the original nonlinear model to a bilinear one by lifting the states to a higher dimension. By keeping the initial states for the control-affine term constant at the beginning of the prediction horizon, the model is further reduced to a linear one. The bilinear model and the linearized bilinear model are shown to have higher accuracy compared to the locally linearized model and the linear Koopman model that neglects control-affine dynamics. The linearized bilinear model is then utilized in an emergency steering simulation scenario. The new model, albeit still a linear one, represents the nonlinear plant well and outperforms the locally linearized model when the vehicle is operated at a higher speed. Additionally, benefiting from its linear form, the computation time is decreased by one order of magnitude compared to the original nonlinear vehicle dynamics in a nonlinear MPC formulation. This acceleration in optimization with minimal compromise in performance is an important step for real-time applications.

Directions for extension of this work are noted as follows. Generating a more accurate bilinear model from a higher fidelity model to increase the operating region could be of interest. Model reduction and basis selection techniques can be investigated in constructing the bilinear Koopman model. Stability analysis and experimental validation of the proposed formulation are also important next steps.

REFERENCES

Aganovic, Z. and Gajic, Z. (1994). The successive approximation procedure for finite-time optimal control of bilinear systems. *IEEE Trans. on Automatic Control*, 39(9), 1932–1935.

Andersson, J.A.E., Gillis, J., Horn, G., Rawlings, J.B., and Diehl, M. (2019). CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1), 1–36.

Bruder, D., Fu, X., and Vasudevan, R. (2021). Advantages of bilinear Koopman realizations for the modeling and control of systems with unknown dynamics. *IEEE Robotics and Automation Letters*, 6(3), 4369–4376.

Bujarbaruah, M., Zhang, X., Rosolia, U., and Borrelli, F. (2018). Adaptive MPC for iterative tasks. In *IEEE Conf. on Decision and Control*, 6322–6327.

Cebuhar, W. and Costanza, V. (1984). Approximation procedures for the optimal control of bilinear and nonlinear systems. *J. of Optimization Theory and Applications*, 43(4), 615–627.

Cibulka, V., Haniš, T., and Hromčík, M. (2019). Data-driven identification of vehicle dynamics using Koopman operator. In *Int. Conf. on Process Control*, 167–172.

Febbo, H. (2019). *Real-time trajectory planning to enable safe and performant automated vehicles operating in unknown dynamic environments*. Ph.D. thesis, University of Michigan.

Gros, S., Zanon, M., Quirynen, R., Bemporad, A., and Diehl, M. (2020). From linear to nonlinear MPC: bridging the gap via the real-time iteration. *Int. J. of Control*, 93(1), 62–80.

Halperin, I., Agranovich, G., and Ribakov, Y. (2020). Extension of the constrained bilinear quadratic regulator to the excited multi-input case. *J. of Optimization Theory and Applications*, 184(1), 277–292.

Han, Y., Hao, W., and Vaidya, U. (2020). Deep learning of Koopman representation for control. In *IEEE Conf. on Decision and Control*, 1890–1895.

Koopman, B.O. (1931). Hamiltonian systems and transformation in Hilbert space. *Proc. of the National Academy of Sciences of the United States of America*, 17(5), 315.

Liu, J., Jayakumar, P., Stein, J.L., and Ersal, T. (2016). A study on model fidelity for model predictive control-based obstacle avoidance in high-speed autonomous ground vehicles. *Vehicle System Dynamics*, 54(11), 1629–1650.

Liu, J., Jayakumar, P., Stein, J.L., and Ersal, T. (2017). Combined speed and steering control in high speed autonomous ground vehicles for obstacle avoidance using model predictive control. *IEEE Trans. on Vehicular Technology*, 66(10), 8746–8763.

Mezić, I. (2013). Analysis of fluid flows via spectral properties of the Koopman operator. *Annual Review of Fluid Mechanics*, 45, 357–378.

Pacejka, H. (2005). *Tire and vehicle dynamics*. Elsevier.

Turri, V., Carvalho, A., Tseng, H.E., Johansson, K.H., and Borrelli, F. (2013). Linear model predictive control for lane keeping and obstacle avoidance on low curvature roads. In *IEEE Conf. on Intelligent Transportation Systems*, 378–383.

Wächter, A. and Biegler, L.T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1), 25–57.

Williams, M.O., Kevrekidis, I.G., and Rowley, C.W. (2015). A data–driven approximation of the Koopman operator: Extending dynamic mode decomposition. *J. of Nonlinear Science*, 25(6), 1307–1346.

Wurts, J., Stein, J.L., and Ersal, T. (2020). Collision imminent steering at high speed using nonlinear model predictive control. *IEEE Trans. on Vehicular Technology*, 69(8), 8278–8289.

Wurts, J., Stein, J.L., and Ersal, T. (2021). Collision imminent steering on curved roads using one-level nonlinear model predictive control. *IEEE Access*, 9, 39292–39302.

Wurts, J., Stein, J.L., and Ersal, T. (2022). Design for real-time nonlinear model predictive control with application to collision imminent steering. *IEEE Trans. on Control Systems Technology*. doi:10.1109/TCST.2022.3154370.