This is the Author's Accepted Manuscript version of the paper:
E. Kochergin, C. Shen, S. Yu, B. Epureanu, and T. Ersal, "Obtaining computationally efficient solutions for high-speed emergency maneuvers using a Koopman operator-based framework," IEEE Intelligent Vehicles Symposium, Ann Arbor, MI, 2026.

Obtaining Computationally Efficient Solutions for High-Speed Emergency Maneuvers using a Koopman Operator-based Framework

Eugene Kochergin, Congkai Shen, Siyuan Yu, Bogdan I. Epureanu, and Tulga Ersal*

***Abstract*—This paper presents a Koopman Operator-based Model Predictive Control (KMPC) framework for emergency maneuvers that incorporates nonlinear tire-road interactions to obtain solutions to harder situations characterized by shorter time-to-collision and distance. Current autonomous driving strategies struggle to strike a balance between computational efficiency and model fidelity. Previous KMPC algorithms for autonomous driving have demonstrated success at low slip angles, but are not generalizable for large slip angles that are often encountered during emergency maneuvers at high speeds. In contrast, nonlinear MPC strategies have been employed in many safety-critical applications, but struggle to meet real-time constraints as tire nonlinearities increase. Our approach for the first time expands the capabilities of KMPC methods to large slip ranges, allowing for the execution of aggressive lane-switching maneuvers with an order-of-magnitude solve time reduction compared to nonlinear methods. To generalize KMPC performance to a broader range of slip angles, Latin Hypercube Sampling is used to generate the training data with an evenly spread slip angle distribution, resulting in capabilities that KMPC formulations regressed from other considered training generation methods could not achieve. We compare the performance and computational efficiency of the proposed KMPC solution with locally linearized and nonlinear MPC algorithms as state-of-the-art benchmarks. Leveraging the global linearization properties and corresponding offline initializations that Koopman formulations can offer, the proposed framework performs double-lane-change maneuvers at speeds and distances similar to those of nonlinear methods, while achieving the lowest solve time of all methods considered.**

***Index Terms*—bilinear systems, emergency maneuvers, Koopman operator, Latin hypercube sampling, model predictive control, Pacejka tire model**

I. Introduction

Autonomous Driving Systems (ADS) are a rapidly emerging technology that can reduce the risks of human error while driving. Recently, vehicles with level-three ADS started to appear, where the autonomy has full control of the vehicle during highway applications [1]. However, these applications require operating at high speeds, where emergency maneuvers may feature little time-to-collision resulting in an insufficient distance to brake. In this case, automatic emergency steering or combined emergency steering and braking systems can provide the vehicle with more opportunities to prevent an accident [2]. However, emergency maneuvers require quick and aggressive actions with optimal performance. The difficulty of meeting these demands is one of the reasons that higher level ADS functionality is not yet widely implemented.

To optimize performance during emergency maneuvers, Model Predictive Control (MPC) is a popular choice for handling complex decision making while satisfying safety constraints [3]. An emergency steering assist system was developed in [4], along with safety restrictions and stability constraints for performing emergency lane switches on roads that may possess curvature. These algorithms were developed to prevent accidents even if it required pushing the vehicle to its dynamical limits; they are able to perform aggressive steering maneuvers to avoid collisions up to the limits of handling. However, as the emergency maneuver becomes more difficult to perform due to shorter time-to-collision and distance, reliable real-time formulations with solve-time and convergence guarantees become difficult to obtain [5].

Table I summarizes the MPC frameworks developed to handle the general problem of navigating a vehicle, along with the computational and performance challenges in this domain. For example, efficient nonlinear architectures were found to accelerate Nonlinear MPC (NL-MPC) formulations significantly through analytic differentiation and GPU computing in [6]. Similarly, sampling-based approaches such as Model Predictive Path Integral (MPPI) [7] offer fixed solve times. However, these algorithms require permanent allocation of GPU resources, potentially requiring significant power and memory consumption. An approach to emergency maneuvers that is not GPU intensive is Hybrid MPC, which can use branch and bound methods to avoid local minima traps and converge to the global minimum with enough time [8], [9]. However, Hybrid MPC formulations utilize mixed integer linear programs, which may become computationally intractable.

Unlike previous NL-MPC formulations, Linear MPC (LMPC) results in a convex quadratic program that can be solved efficiently in real time [14]. In particular, the Locally Linearized MPC (LL-MPC) can re-linearize the system dynamics and constraints at the current prediction step [10]. However, traditional linearized models, such as those reconstructed from the Jacobian, cannot maintain high fidelity in the conditions necessary to drive a vehicle at its dynamical limits [11].

This material is based upon work supported by the National Science Foundation under Grant 1000377584.

Eugene Kochergin, Congkai Shen, Siyuan Yu, Bogdan I. Epureanu, and Tulga Ersal are with the Mechanical Engineering Department at the University of Michigan, Ann Arbor, MI, USA. Emails: {eugenk, cosh, johnysy, epureanu, tersal}@umich.edu

* Corresponding author

TABLE I
COMPARISON OF MPC FORMULATIONS FOR DRIVING A PLANT WITH NONLINEAR TIRE FORCES (NTF) IN REAL TIME

Source	**MPC Type**	Guaranteed Convergence	Guaranteed Solve Time	Takes NTF into Account*
[10]	Locally Linearized MPC	✓	✓	×
[4], [6]	Nonlinear MPC	×	×	✓
[8], [9]	Hybrid MPC	✓	×	✓
[7]	MPPI	×	✓	✓
[11], [12], [13]†	Prior KMPC Literature*	✓	✓	×
Proposed	NTF and LHS-based KMPC	✓	✓	✓

† Training data used the Pacejka tire model, but broad performance or comparisons to Nonlinear MPC were not included.

To expand the capabilities of LMPC frameworks, Koopman operator-based algorithms have emerged to identify globally linear representations of nonlinear dynamical systems [15], offering improved model fidelity over LL-MPC formulations. Finding an accurate finite-dimensional representation of the Koopman operator is akin to identifying coordinate transformations where the nonlinear system appears linear [16], which is usually achieved using Extended Dynamic Mode Decomposition (EDMD) [17]. Koopman Model Predictive Control (KMPC) formulations have seen increasing applications in vehicular control thanks to their computational efficiency. For example, KMPC was implemented for lane-keeping control of a high-fidelity vehicle model in CarSim [12] and it was demonstrated that NL-MPC performed worse than the KMPC at path-tracking when implemented on an RC car with hardware communication delays [13]. Recently, the capabilities of Koopman operator-based methods were expanded by considering bilinear basis functions to capture control-affine dynamics, referred to as bilinear Koopman (BK) operator-based methods. The BK representation can be linearized and utilized within LMPC to provide an accurate approximation of control-affine dynamics; therefore, it was applied to high-speed vehicle control in [11] where it demonstrated lower error than the linear Koopman representation. However, to the authors' best knowledge, no prior KMPC framework is capable of demonstrating broader application to nonlinear tire forces. Hence, KMPC solutions at present offer limited capability in high-speed maneuvers.

Thus, an important gap is identified: Emergency solutions obtained from current LMPC methods have limited fidelity, whereas solutions obtained from nonlinear methods generally require significant computational resources and do not guarantee convergence within the needed time limits. This work addresses this gap by extending the capabilities of KMPC formulations to consider broader nonlinear tire force regions. The original contribution is an emergency maneuver framework that can be generalized to maneuvers with limited time, computational resources, and distance by using KMPC. To this end, Latin Hypercube Sampling (LHS) is established as a training data generation methodology that allows EDMD to regress a Koopman operator representation that, when implemented in the KMPC from [11], offers a broader performance range in the aforementioned emergency situations. This framework's performance is evaluated by the speeds and distances at which it can perform successful emergency double lane change maneuvers in comparison to the state of the art.

The rest of the paper is organized as follows. Vehicle dynamics are discussed in Sec. II. Sec. III discusses Koopman operator-based methods and the proposed training data generation methodology. The MPC formulations are described in Sec. IV. Sec. V discusses the results for the KMPC framework, including modeling performance and control performance of different training data generation methodologies. This section also compares the proposed framework with locally linearized MPC and nonlinear MPC formulations.

II. VEHICLE DYNAMICS

The vehicle model used in this study is the single-track bicycle model of [5], which accounts for tire nonlinearities and longitudinal load transfer. This model demonstrated the best balance of simplicity and performance in MPC formulations for driving a vehicle to its dynamic limits, such as in [18].

The state and control vectors are:

$$\xi = \begin{bmatrix} x \\ y \\ \omega_z \\ V \\ \psi \\ U \end{bmatrix} = \begin{bmatrix} \text{global x position} \\ \text{global y position} \\ \text{yaw rate} \\ \text{lateral speed} \\ \text{yaw angle} \\ \text{longitudinal velocity} \end{bmatrix} \tag{1}$$

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

The vehicle dynamics are expressed as:

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

with the state and control matrices:

$$A(\xi) := \begin{bmatrix} U\cos\psi - V\sin\psi \\ U\sin\psi + V\cos\psi \\ (F_{Y,f} + F_{Y,r})/M - U\omega_z \\ (F_{Y,f}L_f - F_{Y,r}L_r)/I_{ZZ} \\ \omega_z \\ a_x \end{bmatrix} \tag{4}$$

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

where $F_{Y,f}$ and $F_{Y,r}$ are the lateral forces acting on the front and rear tires, M is the vehicle mass, I_{ZZ} is its yaw moment of inertia, and L_f and L_r are distances from the center of gravity to the front and rear axles.

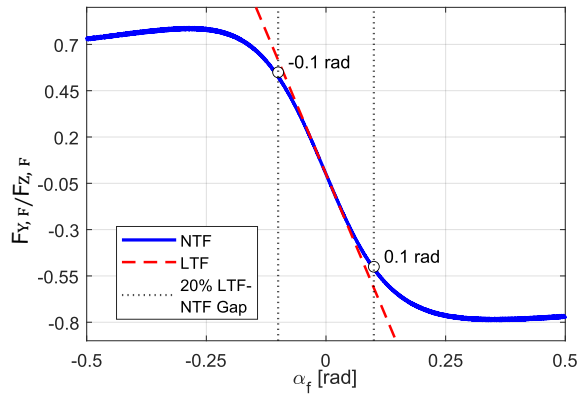

Fig. 1. Lateral force $F_{Y,f}$ normalized by the vertical force $F_{Z,f}$ versus slip angle α_f plotted for the nonlinear tire force (NTF) model in (8) (blue) and linear tire force (LTF) model from [11] (red). Dotted lines indicate the slip angle thresholds beyond which there is more than 20% discrepancy between the two models, which happens at approximately $|\alpha_f| = 0.1$ rad.

This study examines two tire force models: a nonlinear tire force model and a linear tire force approximation. The nonlinear model uses the Pacejka formula [19] to describe the mapping of the front slip angle α_f and rear slip angle α_r

$$\alpha_f = \arctan\left(\frac{V + L_f\omega_z}{U}\right) - \delta_f \tag{6}$$

$$\alpha_r = \arctan\left(\frac{V - L_r\omega_z}{U}\right) \tag{7}$$

to the lateral tire force $F_{Y,\bullet}$

$$F_{Y,\bullet} = D_\bullet \sin\Bigg(C_\bullet \arctan\left[B_\bullet X_\bullet - E_\bullet\Big(B_\bullet X_\bullet - \arctan\left(B_\bullet X_\bullet\right)\Big)\right]\Bigg) + S_{V,\bullet} \tag{8}$$

where $X_\bullet = \alpha_\bullet + S_{H,\bullet}$ and $\bullet = \{f, r\}$. Here, $B_\bullet$, $C_\bullet$, $D_\bullet$, $E_\bullet$, $S_{H,\bullet}$, and $S_{V,\bullet}$ are functions of the vertical load $F_{Z,\bullet}$. More information about these terms can be found in [18]. For this study, the front slip angle α_f is utilized to inform the performance metrics, as the front tires experience larger slip due to steering. Fig. 1 visualizes the saturation of the lateral forces against α_f and $F_{Z,F}$. The linear tire force formula serves as a model fidelity benchmark to demonstrate the improvement in performance that this framework offers over the work implemented in [11].

III. Koopman Operator-based Realizations for Optimal Control

A. Koopman Operator-based Modeling using EDMD

Let $\mathcal{F}(\xi, \zeta)$ denote the right-hand side of (3). Every element $f \in \mathcal{F}$ is a lifted state ("observable") that contains information about the original nonlinear system in a measurement lifting function. The evolution of the lifted states is characterized by the linear Koopman operator $\mathcal{K}$:

$$\mathcal{K}f(\xi(t), \zeta(t)) = \frac{\partial f}{\partial \xi}\frac{d\xi}{dt} = \frac{\partial f}{\partial \xi}\mathcal{F}(\xi(t), \zeta(t)) \tag{9}$$

The Koopman representation is obtained by approximating the Koopman operator in a finite-dimensional subspace with EDMD [17]. The lifted system and the corresponding time derivatives are formulated with the lifting function Φ as:

$$\begin{aligned} p^{(n)} &= \Phi\left(\xi^{(n)}(t), \zeta^{(n)}(t)\right) \\ q^{(n)} &= \frac{d}{dt}\left(\Phi(\xi^{(n)}(t), \zeta^{(n)}(t)\right) \\ &= \frac{\Phi\left(\xi^{(n)}(t+\Delta t), \zeta^{(n)}(t+\Delta t)\right) - \Phi\left(\xi^{(n)}(t), \zeta^{(n)}(t)\right)}{\Delta t} \end{aligned} \tag{10}$$

where state $\xi^{(n)}(t)$ and control $\zeta^{(n)}(t)$ pairs correspond to the n^{th} snapshot. $p^{(n)}$ and $q^{(n)}$ correspond to the values of all lifted state measurements and their time derivatives at that point. Given training data with N sampled snapshots, along with corresponding time derivatives and lifted state values corresponding to each state and control pair, the subspace of the Koopman operator is linearly regressed by forming matrices for lifted state measurements P and time derivatives measurements Q. The Koopman representation $\mathcal{K}$ best describes the linear evolution from P to Q as:

$$\min_{\mathcal{K}} \|\mathcal{K}P - Q\|_F^2 \tag{11}$$

$$P = \begin{bmatrix} p^{(1)} & p^{(2)} & \dots & p^{(N)} \end{bmatrix} \tag{12}$$

$$Q = \begin{bmatrix} q^{(1)} & q^{(2)} & \dots & q^{(N)} \end{bmatrix} \tag{13}$$

The EDMD solution $\mathcal{K}^*$ for $\mathcal{K}$ refers to the analytical solution to the least squares regression problem

$$\mathcal{K}^* = QP^\dagger \tag{14}$$

where $\dagger$ refers to the Moore-Penrose pseudo-inverse.

B. Bilinear Koopman Model Realization

The bilinear Koopman model uses a bilinear basis to capture control-affine dynamics in (3). Decomposing the Koopman representation, the bilinear lifted system has the form

$$\dot{\tilde{\xi}} = \tilde{A}\tilde{\xi} + \tilde{B}\zeta + \sum_{j=1}^{\tilde{n}} \tilde{\xi}_j \tilde{N}_j \zeta \tag{15}$$

where $\tilde{\xi}$ is a vector that contains all $\tilde{n}$ lifted states defined by the lifting transformation $\tilde{\xi} = \Phi(\xi, \zeta)$, $\tilde{A} \in \mathbb{R}^{\tilde{n}\times\tilde{n}}$, $\tilde{N}_j \in \mathbb{R}^{\tilde{n}\times m}$, and $\tilde{B} \in \mathbb{R}^{\tilde{n}\times m}$; $\tilde{N}_j$ is the term that contains the bilinear terms. The bilinear basis Φ_{BK} includes Φ_ξ, which is the lifted state basis consisting of polynomials of the states from (1) (except x and y), multiplied by a control input. Fourth-order polynomials are utilized as lifting functions in this study similar to [11]; i.e.,

$$\Phi_{\text{BK}} = \begin{bmatrix} \Phi_\xi & \Phi_\xi\zeta_1 & \Phi_\xi\zeta_2 & \zeta_1 & \zeta_2 \end{bmatrix} \tag{16}$$

The bilinear Koopman model leads to an optimal control formulation with bilinear constraints, which typically can only be solved using nonlinear programming tools. For this reason, the bilinear Koopman model is locally linearized at the current state $\tilde{\xi}_j(0)$ for local approximation of (15), known as Linearized

Bilinear Koopman (LBK) representation [11], to obtain a linear structure for LMPC implementation:

$$\dot{\tilde{\xi}} = \tilde{A}\tilde{\xi} + \bar{B}\zeta = \tilde{A}\tilde{\xi} + [\tilde{B} + \sum_{j=1}^{\tilde{n}} \tilde{\xi}_j(0)\tilde{N}_j]\zeta \tag{17}$$

C. Training Data Generation Methodology

Regressing an accurate Koopman representation of the vehicle dynamics with nonlinear tire forces is more challenging than fitting the linear tire force region. One concern about nonlinear tire forces is that significant α_f values are rare due to α_f's dependence on the difference between the wheel's direction and motion. If the training data is composed of sampled vehicle trajectories, the least squares solution to EDMD in (10) would be biased towards fitting the linear region better due to being overrepresented in the training set. To compensate for the difficulty of capturing more training snapshots in the nonlinear region, we propose two strategies: 1) varying the Control Range CR in the training data set, and 2) changing the training data generation methodology. The CR is summarized as:

$$\forall\, n = 1, \ldots, N, \quad CR\,\zeta_{\text{min}} \leq \zeta_n \leq CR\,\zeta_{\text{max}} \tag{18}$$

where n and N refer to the snapshot index and snapshot library size in the training set. CR, expressed as a percentage value, determines the amount of control effort used to generate the training set. Tuning the CR in the training set can adjust the training bias towards larger α_f. However, the data generation method has a greater significance on the resulting α_f distributions. A list of training data generation methods commonly used for EDMD in Koopman applications is found in Table II.

To date, open-loop control trajectories are usually used to generate training data. Two open-loop approaches are common: 1) interpolating between a few randomly selected control points [20], referred to as OL_1 in Table II; or 2) randomizing the control effort at every step [11], [12], referred to as OL_2 in Table II. OL_1 experiences a select few control actions between which the control varies. Unlike OL_1, OL_2 features drastic control fluctuations in the training data which are likely to lead to larger tire nonlinearities being encountered.

For nonlinear dynamics that feature different regimes with different behaviors (e.g., saturation), the training data may feature an uneven and insufficient sampling of each region, even with aggressive open-loop methods such as OL_2. For such dynamics, we suggest using training data with forced ergodic sampling of each region to obtain better performance. For this reason, formulating the training data by sampling from the multi-variable ξ and ζ with Latin Hypercube Sampling (LHS) is considered to ensure maximum coverage of the state and control domains. The selection of the training data generation method is the main difference of this KMPC to the work in [11].

IV. Model Predictive Control Formulation

This formulation utilizes a single-layer MPC to solve the planning and control problems simultaneously. The MPC formulation used in this study is nearly identical to [11], but employs modified weights to address emerging tire nonlinearities. The optimal solution is found by solving the optimal control problem:

$$\underset{\xi,\,\zeta,\,t_f}{\text{minimize}} \qquad J = \int_0^{T_p} \mathcal{I}[\xi(t), \zeta(t)]dt \tag{19}$$

$$\text{subject to} \qquad \dot{\xi}(t) = \mathcal{V}[\xi(t), \zeta(t)] \tag{20}$$

$$\xi_{\text{min}} \leq \xi(t) \leq \xi_{\text{max}} \tag{21}$$

$$\zeta_{\text{min}} \leq \zeta(t) \leq \zeta_{\text{max}} \tag{22}$$

where J refers to the cost function, $\mathcal{I}$ is the incremental cost on states and controls, $\mathcal{V}$ refers to the prediction model, which is either (3) or (17), and T_p refers to the prediction horizon.

A. Cost Function

The cost function is a quadratic defined as

$$J = \int_0^{T_p} \Big[w_{\delta_f}\delta_f(t)^2 + w_{a_x}a_x(t)^2 + w_y y(t)^2 + w_\psi \psi(t)^2 + w_{s_y} s_y^2 \Big] dt \tag{23}$$

where w_{δ_f}, w_{a_x}, w_y, w_ψ, and w_{s_y} are the weights corresponding to steering, acceleration, lane keeping, heading, and s_y is a slack variable for obstacle constraints. The first and second terms prevent overly aggressive control actions ("aggressive" denoting frequent, large amplitude control inputs), the third and fourth terms encourage the vehicle to travel in a straight line or initiate the maneuver earlier, and the last term enforces soft constraints on the drivable tube discussed in Sec. IV-C.

TABLE II
Comparison of Training Data Generation Methodologies

Abbreviation	Training Data Method	Description	Example in Literature
OL_1	Open loop: Seeded trajectories	Generates a queue of trajectories using a few selected ("seeded") control inputs at distinct points with interpolation (cubic implemented in this study). Simulates control effort that varies gradually and continuously with respect to time.	[20] (linear interpolation)
OL_2	Open loop: Random control actions per each timestep trajectories	Applies random control inputs at each time step obtained from a random function.	[11], [12]
LHS	Latin hypercube sampling	Acquires state ξ and control ζ snapshot pairs from a multivariable distribution, with LHS used specifically to obtain maximum coverage of the domain given fixed training size.	Implemented here only

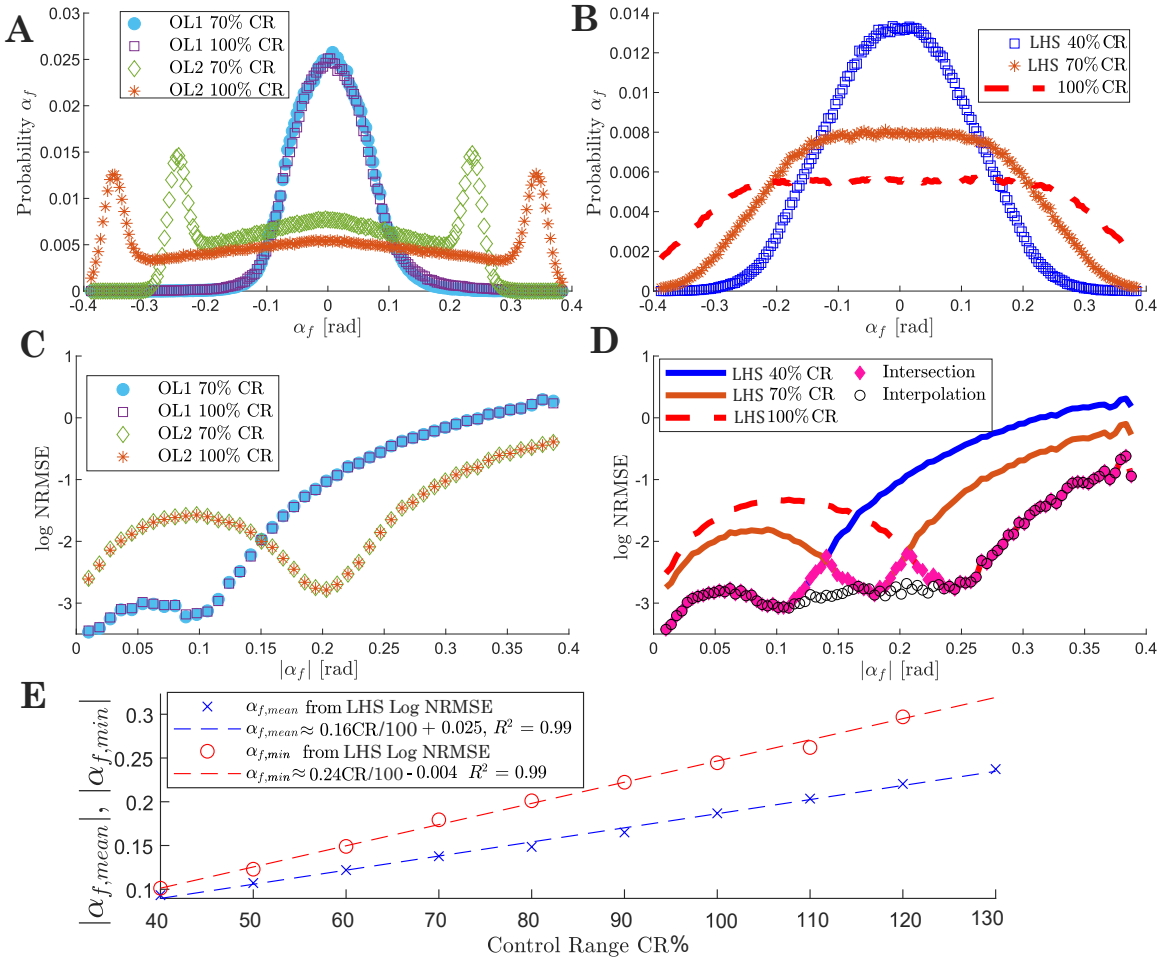

Fig. 2. Summary of modeling performance for each Koopman representation based on different training data methodologies from Table II. A) α_f distribution for open-loop methods OL_1 and OL_2, considering different training Control Range (CR) scalars corresponding to $\zeta \in [CR\zeta_{\min}, CR\zeta_{\max}]$. B) α_f distribution from Latin Hypercube Sampling (LHS) at varying CR. The log NRMSE is calculated in (28) as the summed difference of (3) and (15) for all states ξ normalized by the range of ξ, and binned into discrete α_f values. C) and D) contain the NRMSE plotted against the given α_f for the open-loop methods and LHS, respectively. The intersection curve in D) refers to using the $\mathcal{K}$ corresponding to the CR curve with the lowest NRMSE at the given α_f, while the interpolation uses a $\mathcal{K}$ that is the linear interpolation between two $\mathcal{K}$ with different CR values based on proximity to corresponding α_f minimum as calculated in (29). E) describes the location of the mean absolute α_f in rad, denoted $\alpha_{f,\text{mean}}$, and location of the corresponding residual error minima, $\alpha_{f,\text{min}}$, which describes the α_f value at which the $\mathcal{K}$ possesses the most accuracy as a function of CR for LHS generated training data using additional $\mathcal{K}$ values not included in D).

B. Control and State Constraints

The control constraints are obtained from fitting the actuator capabilities [18]:

$$\delta_{f_{\min}} \leq \delta_f \leq \delta_{f_{\max}} \tag{24}$$

$$a_{x_{\min}} \leq a_x \leq a_{x_{\max}} \tag{25}$$

and the state constraints are placed on the heading ψ to prevent the vehicle from turning around and leaving the lane boundaries:

$$\psi_{\min} \leq \psi \leq \psi_{\max} \tag{26}$$

C. Drivable Tube Constraint

The drivable tube constraint, adopted from [11], is implemented for obstacle avoidance. Deriving the constraints based on the kinematics and straight-road assumption, the vehicle is considered safe if the following condition is satisfied:

$$\begin{gathered} y_{\text{L, i}} - s_y \leq y(t) \leq y_{\text{U, i}} + s_y \\ t \in [t_{L,i}, t_{U,i}],\ t_{L,i} \approx \frac{x_{\text{L, i}} - x_c}{U_c},\ t_{U,i} \approx \frac{x_{\text{U, i}} - x_c}{U_c} \end{gathered} \tag{27}$$

with current position x_c and speed U_c. $(x_{\text{L, i}}, y_{L,i}) \times (x_{\text{U, i}}, y_{\text{U, i}})$ defines the rectangular safe set the vehicle must pass through to avoid collision and i refers to the obstacle index. $t_{L,i}$ and $t_{U,i}$ are approximated times.

V. Lane Switching Formulation Results

A. Training Data Generation Results

In this section, we analyze the modeling performance of the Koopman representations regressed from the training data generation methods discussed in Table II. The main analysis focuses on α_f distributions in the training data corresponding

TABLE III
PARAMETERS OF THE LANE-SWITCH SCENARIO FOR MPC VALIDATION

Parameter	Value/Expression
L_{OBS1}	20–45 m
$(x_{\text{L},1}, y_{\text{L},1}) \times (x_{\text{U},1}, y_{\text{U},1})$	$(L_{\text{OBS1}}, 0) \times (L_{\text{OBS1}} + d, 0.5)$ m
$(x_{\text{L},2}, y_{\text{L},2}) \times (x_{\text{U},2}, y_{\text{U},2})$	$(2L_{\text{OBS1}} + d, 2.6) \times (2L_{\text{OBS1}} + 2d, 3.6)$ m
U_0	15–30 m/s
d	10 m

The parameters define the lane-switch scenario, where the obstacle position L_{OBS1} and initial speed U_0 determine difficulty.

to each method and the distribution of error (expressed as normalized root mean squared error NRMSE) with respect to α_f to characterize high dimensional vehicle behavior by studying performance relative to the most dominant parameter. The corresponding α_f distributions are plotted in Fig. 2A-B, and the corresponding error is plotted against the absolute value of α_f in Fig. 2C-D. To plot the error, we use snapshots from different training methods (e.g., OL_1 with new control actions every 0.05s). Using the residual difference between (3) and (15), normalized by the state ranges, we obtain the NRMSE below:

$$\text{NRMSE}(\xi, \zeta) = \frac{1}{6} \sum_{k=1}^{6} \frac{\left| \dot{\xi}_k(\xi, \zeta) - \dot{\tilde{\xi}}_k(\xi, \zeta) \right|}{\text{range}(\xi_k)} \tag{28}$$

where k refers to the state index. The error curves of all training methods show similar behaviors, resembling a third-order polynomial function of $|\alpha_f|$ exhibiting one local maximum, one local minimum, and an error that diverges at large α_f.

OL_1 is significantly less aggressive than OL_2, and increasing the CR by 30% causes little difference for OL_1 in both the α_f distribution in Fig. 2A and the corresponding modeling error curve in Fig. 2C, as the control effort in these trajectories varies too slowly to explore the nonlinear tire force region. OL_2 has larger α_f variations in the training data with increasing CR in Fig. 2A; however, OL_2 modeling error curves are still independent of the CR in Fig. 2C despite this difference.

The LHS distribution, plotted in Fig. 2B, shows a well-distributed α_f magnitude in training with a flat probability region at zero slip. The length and height of the α_f distribution is determined by CR with a gradual drop-off between $\alpha_f \in [0.04, 0.25]$ rad for 40-100% CR. Unlike the performance of OL_1 and OL_2, Fig. 2D shows that LHS-based Koopman representations have different error curves as a function of CR. The local minima error $\alpha_{f,\text{min}}$ for LHS 40%, 70%, and 100% CR occurs at roughly 0.1, 0.18, and 0.24 rad, respectively. LHS-based Koopman representations have another advantage in that their capabilities can be extended to different α_f values by linear interpolation of Koopman representations with different CR:

$$\mathcal{K}_{\text{Interpolated}} = \mathcal{K}_L + \frac{\alpha_f - \alpha_{f,L,\text{ min}}}{\alpha_{f,U,\text{ min}} - \alpha_{f,L,\text{ min}}} \times (\mathcal{K}_U - \mathcal{K}_L) \tag{29}$$

where subscripts L and U denote the minimum and maximum CR Koopman representations available, and their corresponding $\alpha_{f,\text{min}}$. This gives it better accuracy over switching the Koopman representation to a higher CR whenever the α_f error curves of lower CR and higher CR representations intersect, and requires only two Koopman representations to be regressed.

B. Model Predictive Control Formulation Testing

In Sec. V-A, we introduce different α_f distributions, training data generation methods, and the corresponding modeling errors. In this section, we discuss how these factors impact the MPC performance. For brevity, the formulations are referred to as "[Training Method]-MPC." Each formulation is tasked with performing a double lane-change maneuver using the vehicle model from Sec. II.A. Performance is evaluated based on the range of speeds and distances the planner can complete a double lane change maneuver without violating drivable tube or lane boundary constraints from Sec. IV. The parameters of this simulation are given in Table III. The first and second lane switch scenario is performed with the same distance L_{OBS1}, and the maneuver is initiated with speed U_0. A prediction horizon of T_p = 3 s is used to allow the controller to prepare for the second lane change while performing the first. The execution horizon, T_{Update}, determines the amount of control actions applied to (3) before the MPC is resolved at the current position.

The results for each formulation in performing double lane change maneuvers are presented in Fig. 3. In row-wise sequence, OL_1-, OL_2-, and LHS-MPC formulations are compared at different L_{OBS1} and U_0, with CR set to 100% for $T_{\text{Update}} = 0.1$s. OL_2-MPC achieves a poor success rate of 17% at most, which could be a result of OL_2's α_f's distribution in training being a poor fit to the testing application. OL_1-MPC improves to a 27% success rate, performing better at higher U_0. LHS-MPC outperforms the other methods with a 63% success rate.

Fig. 3D compares method variants at different execution horizons from $T_{\text{Update}} \in [0.1, 0.5]$ s. Only OL_1 and LHS benefit from more aggressive training data, while OL_2 degrades. LHS-MPC outperforms the other KMPC formulations and maintains a steady 60% success rate for $T_{\text{Update}} = 0.1 - 0.3$s. OL_2-MPC 70% CR and OL_1-MPC 100% CR perform well at $T_{\text{Update}} = 0.4 - 0.5$s but inconsistently at $T_{\text{Update}} = 0.2 - 0.3$s, which is likely attributed to declining performance for slip angles outside the linear tire force region (i.e., $|\alpha_f| \gg 0.1$ rad). LHS-MPC 70% CR and LHS-MPC 100% CR can take advantage of frequent re-planning and control actions when $T_{\text{Update}} \in [0.1, 0.3]$s, with the less aggressive LHS-MPC 70% CR performing better at the other T_{Update} values. These results establish LHS as the better training data generation method.

Next, we compare LHS-MPC to locally linearized MPC (LL-MPC) and nonlinear MPC (NL-MPC) formulations as computational efficiency and performance benchmarks, respectively. We use offline initializations of the optimal control problem to enhance computational efficiency. For LL-MPC, this includes the cost function matrix calculations in (23) and (24). The LBK framework benefits from global $\tilde{A}$ and $\tilde{N}$ terms, which result in most MPC terms except those involving $\bar{B}$ being fixed, resulting in more offline initializations. The NL-MPC is warm-started using the previous solution. Finally, we also include the MPC formulation with a linear tire force (LTF-MPC) from [11] as a model fidelity benchmark. The comparisons are done with

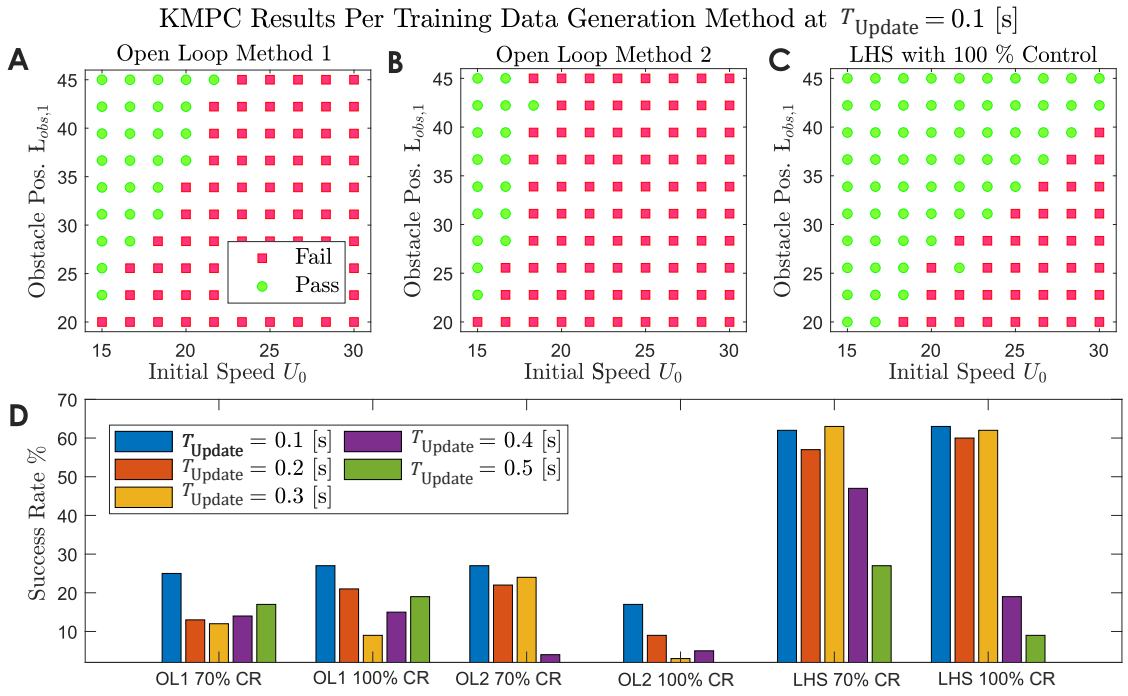

Fig. 3. Summary of the KMPC performance for each training method from Table II and the parameters from Table III showing pass (green) or fail (red) corresponding to whether or not the planner could complete a double lane change maneuver given initial speed U_0 and lane switch distance L_{OBS1} for execution horizon $T_{\text{Update}} = 0.1$s. (A) Open loop methods OL_1 "seeded-controls", (B) OL_2 "randomized control effort", (C) Latin Hypercube Sampling (LHS), (D) Comparison of all training methods across various execution horizons.

$T_{\text{Update}} = 0.1$ s to consider cases with frequent planning and control actions necessary for an emergency maneuver.

The performance of the KMPC formulations based on the representations in Fig. 2 in double lane change maneuvers is found in Fig. 4. The LTF-MPC formulation and the LHS-MPC are compared in Fig. 4A. This figure demonstrates the significant improvements the LHS-MPC formulation offers at higher speeds and shorter distances over the LTF-MPC, with a success rate of 63% compared to 36%. LTF-MPC demonstrates inferior performance to the LL-MPC with 56% success rate in Fig. 4B, likely because the LL-MPC could incorporate some of the nonlinear tire force region during frequent linearization. The NL-MPC features the highest success rate of 79% in Fig. 4C. The success rate gap of 16% between LHS MPC and NL-MPC is significant, as tighter performance is required for emergency maneuvers with increasing speed and decreasing distance. However, the NL-MPC considered in this study is not subject to solve-time constraints that could degrade performance by terminating the optimization before convergence (i.e., real-time constraints are not enforced in this study). The performance gap may be further reduced by selecting the Koopman representation with the optimal CR for the given T_{Update} in further studies.

The solve times for these algorithms are given in Table IV. The LHS-MPC formulation's mean solve time is 0.018 s compared to LL-MPC's 0.033 s, being more efficient thanks to better offline initialization opportunities and a higher fidelity prediction model due to global linearization. The NL-MPC without solve time restriction takes a minimum solve time of 0.107 s, rendering it unable to solve the problem in real time. Furthermore, the linear MPC formulations' computational burden decreases by 18-40% while NL-MPC maximum solve time increases by up to 130% when the time-to-collision is decreased due to higher starting speeds and less maneuver distance. As a result, LHS-MPC formulation offers the most significant reduction in solve time, especially when the vehicle is driven to its dynamic limits.

TABLE IV
LL-, LHS-, AND NL-MPC SOLVE TIMES PER 7-SECOND TRAJECTORY

Difficulty	Method	Mean [s]	Max [s]	Total [s]
Easy	LL-MPC	0.033	0.077	2.27
Easy	LHS-MPC	**0.018**	**0.066**	**1.27**
Easy	NL-MPC	0.107	0.165	7.35
Hard	LL-MPC	0.020	0.066	1.38
Hard	LHS-MPC	**0.015**	**0.034**	**1.03**
Hard	NL-MPC	0.124	0.394	8.59

VI. CONCLUSIONS

This paper introduces a Koopman Model Predictive Control (KMPC) framework regressed from training data generated using Latin Hypercube Sampling (LHS) for the purpose of performing emergency maneuvers at high speeds. Out of all training data generation methods considered in this study, the KMPC formulation using a LHS-based Koopman representation offers the best performance for emergency double lane change maneuvers at higher speeds and shorter distances. Furthermore, the LHS-based KMPC aggressiveness during the maneuver can

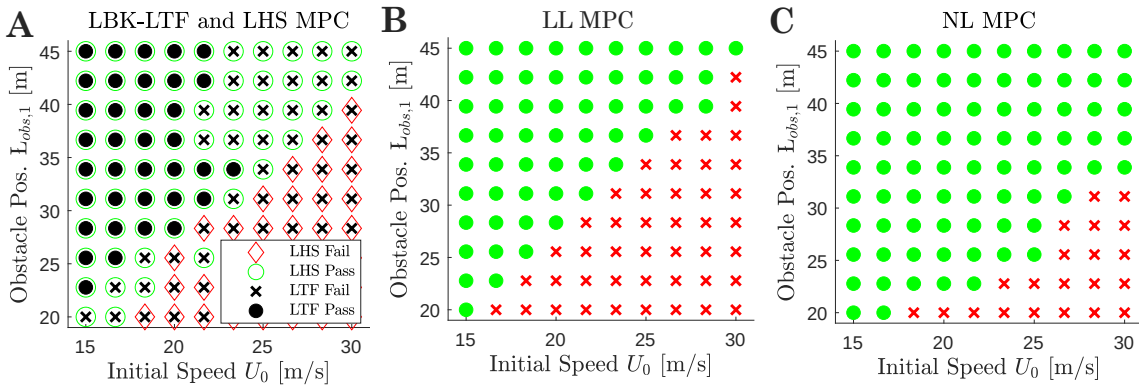

Fig. 4. Performance of various MPC formulations at controlling a vehicle model with nonlinear tire forces and longitudinal load transfer at $T_{\text{Update}} = 0.1$ [s]. (A) the LTF-MPC of [11] and the proposed LHS 100% CR MPC, (B) LL-MPC, and (C) NL-MPC.

be tuned by varying the amount of control effort in training; in contrast, KMPC formulations obtained using other training methods show either relative insensitivity or significant degradation under this change. Additionally, it is demonstrated that the proposed algorithm leverages global linearization and offline initializations to achieve solve times that are faster than the LL-MPC framework, achieving maximum solve times of 0.018 s in the worst case per 3-second prediction horizon—compared to LL-MPC's 0.033 s. LL-MPC and KMPC formulations are observed to be less computationally intensive than NL-MPC in scenarios with lower time-to-collision, due to being solved with warm-started active set quadratic programming. The computational burden decreases by 18-40% between the easiest and hardest scenarios. In contrast, the NL-MPC requires at minimum a 0.107 s mean solve time and a maximum solve time increased by 130% between the easiest and hardest scenarios. The resulting KMPC has a success rate of 63% compared to NL-MPC's 79% at completing double lane change maneuvers at various speeds and distances, a reasonable trade-off to achieve real-time feasibility. These conclusions demonstrate the potential KMPC offers in computational burden relief when the vehicle is driven to the limits during emergency maneuvers.

While the proposed methodology is promising, it has limitations to be addressed in future research. The main limitation is that if this algorithm is implemented on a high-fidelity vehicle model capable of predicting rollover (e.g., see [18]), most maneuvers studied here will no longer be feasible without additional stability constraints for high-center of gravity vehicles. Hence, extending the KMPC's capabilities to enforce these constraints is necessary for on-board vehicle implementation. This can be done either within the higher dimensional lifted space or by augmenting the KMPC with another controller which tracks the Koopman's solution as closely as possible while enforcing these constraints.

References

[1] M. Reda, A. Onsy, A. Y. Haikal, and A. Ghanbari, "Path planning algorithms in the autonomous driving system: A comprehensive review," *Robotics and Autonomous Systems*, vol. 174, p. 104630, 2024.

[2] A. Eckert, B. Hartmann, M. Sevenich, and P. E. Rieth, "Emergency steer & brake assist – a systematic approach for system integration of two complementary driver assistance systems," in *International Technical Conference on the Enhanced Safety of Vehicles*, 2011, paper 11-0111.

[3] M. Hamidaoui, M. Z. Talhaoui, M. Li, M. A. Midoun, S. Haouassi, D. E. Mekkaoui, A. Smaili, A. Cherraf, and F. Z. Benyoub, "Survey of autonomous vehicles' collision avoidance algorithms," *Sensors*, vol. 25, no. 2, 2025.

[4] J. Wurts, J. L. Stein, and T. Ersal, "Collision imminent steering on curved roads using one-level nonlinear model predictive control," *IEEE Access*, vol. 9, pp. 39 292–39 302, 2021.

[5] H. Febbo, J. Liu, P. Jayakumar, J. L. Stein, and T. Ersal, "Moving obstacle avoidance for large, high-speed autonomous ground vehicles," in *American Control Conference*, 2017, pp. 5568–5573.

[6] J. Wurts, J. L. Stein, and T. Ersal, "Design for real-time nonlinear model predictive control with application to collision imminent steering," *IEEE Transactions on Control Systems Technology*, 2022.

[7] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Aggressive driving with model predictive path integral control," *International Conference on Robotics and Automation*, pp. 1433–1440, 2016.

[8] L. Gharavi, B. D. Schutter, and S. Baldi, "Efficient MPC for emergency evasive maneuvers, Part I: Hybridization of the nonlinear problem," *arXiv:2310.00716*, 2023.

[9] ——, "Efficient MPC for emergency evasive maneuvers, Part II: Comparative assessment for hybrid control," *arXiv:2310.00717*, 2023.

[10] S. Gros, M. Zanon, R. Quirynen, A. Bemporad, and M. Diehl, "From linear to nonlinear MPC: bridging the gap via the real-time iteration," *International Journal of Control*, vol. 93, pp. 1–19, 2016.

[11] S. Yu, C. Shen, and T. Ersal, "Autonomous driving using linear model predictive control with a Koopman operator based bilinear vehicle model," *IFAC-PapersOnLine*, vol. 55, no. 24, pp. 254–259, 2022.

[12] J. S. Kim, Y. S. Quan, and C. C. Chung, "Koopman operator-based model identification and control for automated driving vehicle," *International Journal of Control, Automation and Systems*, vol. 21, no. 8, pp. 2431–2443, 2023.

[13] M. Švec, Šandor Ileš, and J. Matuško, "Optimizing vehicle handling through Koopman-based model predictive torque vectoring: An experimental investigation," *Control Engineering Practice*, vol. 158, p. 106272, 2025.

[14] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[15] I. Mezíc, "Spectral properties of dynamical systems, model reduction and decompositions," *Nonlinear Dynamics*, vol. 41, pp. 309–325, 08 2005.

[16] S. L. Brunton, M. Budišić, E. Kaiser, and J. N. Kutz, "Modern Koopman theory for dynamical systems," *SIAM Review*, vol. 64, no. 2, pp. 229–340, 2022.

[17] M. O. Williams, M. S. Hemati, S. T. Dawson, I. G. Kevrekidis, and C. W. Rowley, "Extending data-driven Koopman analysis to actuated systems," *IFAC-PapersOnLine*, vol. 49, no. 18, pp. 704–709, 2016.

[18] J. Liu, J. Paramsothy, J. L. Stein, and T. Ersal, "Combined speed and steering control in high speed autonomous ground vehicles for obstacle avoidance using model predictive control," *IEEE Transactions on Vehicular Technology*, vol. 99, no. 11, pp. 1629–1650, 2016.

[19] H. Pacejka, *Tire and Vehicle Dynamics*. Elsevier, 2005.

[20] D. Bruder, X. Fu, R. B. Gillespie, C. D. Remy, and R. Vasudevan, "Data-driven control of soft robots using Koopman operator theory," *IEEE Transactions on Robotics*, vol. 37, no. 3, pp. 948–961, 2021.