EasyChair Preprint

№ 5622

Deep Koopman Data-Driven Optimal Control Framework for Autonomous Racing

Rongyao Wang, Yiqiang Han and Umesh Vaidya

May 26, 2021

Deep Koopman Data-Driven Optimal Control Framework for Autonomous Racing

Rongyao Wang, Yiqiang Han*, Umesh Vaidya*
Department of Mechanical Engineering
Clemson University
Clemson, SC 29634 USA

***Abstract*—A model-based, data-driven control framework is introduced within the context of autonomous driving in this study. We propose a data-driven control algorithm that combines autonomous system identification using model-free learning and robust control using a model-based controller design. We present a full solution framework that is capable to automatically generate tire-friction limit path while performing system identification of a vehicle with unknown dynamics. We then design model-based control which is actively learned from a data-driven approach. Based on our new system identification algorithm, we can approximate an accurate, explainable, and linearized system representation in a high-dimensional latent space, without any prior knowledge of the system. To validate the algorithm, we conduct the model predictive control of an autonomous vehicle based on the augmented system identification on a scaled racing vehicle. The result indicates that we are able to design control in the lifted space to achieve tasks in path control and obstacle avoidance. The automatic path generation combined with the data driven control requires no a-priori knowledge of the vehicle and also proved to be effective that only requires less than 5 laps to design an low lap-time trajectory while identified a system that is able to achieve minimum lap time without extra learning episodes.**

***Index Terms*—Data-driven control, Linear operator approach, Deep Learning, Model Predictive Control, Dynamic Programming.**

I. INTRODUCTION

Recent developments of deep learning have greatly changed ways to understand data from a dynamical system, and subsequently, the controller design. We are able to extract features more efficiently in the latent space created by the neural network. The increased dimensions of the neural network, in turn, pose difficulties in explaining the decision-making process of a dynamical system with control, such as autonomous driving and autonomous racing. In the traditional control theory aspect, two types of well-documented controllers with excellent explainability can be utilized: Model-based Geometric controller and Model-based Optimization-based controller. One widely-used model-based geometric controller is a pure pursuit controller, which has been proven functional from Stanford's work in 2005 DARPA Grand-challenge [1]. However, the geometric controller performance heavily relies on road conditions and vehicle speed. Despite the fact that a number of studies and modifications have been developed to improve pure pursuit performance [2] [3], the decay in tracking performance still exist with higher navigation speed and aggressive controls.

With the massive improvement in GPU computing power for the recent decade, optimization-based control such as Model Predictive Control (MPC) has been widely used in path tracking controller design. The model predictive controller has been proven good performance in high speed and low-friction environments, exhibiting robustness against sampling noise [4] [5]. One disadvantage of conventional model predictive control is that the controller's accuracy heavily depends on vehicle dynamics accuracy. When the vehicle dynamics change over time, the controller's performance will inevitably be compromised.

This paper proposes a data-driven approach to design model predictive control—the resultant controller benefits from the model-free learning of the system identification and the model-based control robustness. Unlike conventional model predictive control, vehicle dynamics are learned rather than modeled. As a result, the vehicle dynamics can actively update itself with the recently collected data, which gives the vehicle's ability to change its dynamics in real-time. On the other hand, this method can also approximate the vehicle's nonlinear characteristic even without a fully observable system, which should have better performance on high speed or slippery conditions compared to the predefined vehicle dynamics.

Data-driven control includes model-based and model-free approaches [6] [7]. In this paper, we focus on incorporating model-based data-driven control using the Koopman operator theory. The first contribution we proposed in this paper is developing deep neural network as Koopman operator to achieve system identification of the vehicle dynamics, which is proved to be flexible and robust in different simulation scenarios. The second contribution we proposed in this paper is developing an iterative approach to generate a standard vehicle racing line aiming to achieve the minimum lap time.

II. PRELIMINARIES

This section discusses some preliminaries and introduces the notations, which will be used in deriving the main results on autonomous driving with the deep Koopman learning method.

Graduate Researcher, rongyaw@clemson.edu
Research Assistant Professor, yiqianh@g.clemson.edu
Professor, uvaidya@clemson.edu
The research work was partially supported from NSF CPS award 1932458.

A. Approximating the Data-driven Koopman Operator

Consider a controlled dynamical system as in equation (1).

$$x_{t+1} = f(x_t, u_t) \tag{1}$$

where $x_t \in \mathbb{R}^n$. Assume we have a recorded time-series of states $X = [x_0, x_1, x_2, ..., x_{k+1}]$ and corresponding control sequence $U = [u_0, u_1, u_2, ..., u_k]$ that drive the evolution of X. Then the Koopman operator $\mathcal{K}$ is an infinite-dimension operator that build linear system representation of dynamical system (1) based on the observable space $g(x_t)$ as shown in equation (2).

$$\mathcal{K}g(x_t) = g(f(x_t, u_t)) \tag{2}$$

where $g(x_t)$ lift the original space $\mathbb{R}^n$ to a higher dimension $\mathbb{R}^m$. The main objective of Koopman operator theory is to find $\mathcal{K}$ that best match the relationship as shown in Eqn. (2) based on the recorded state (X) and control sequence (U).

As infinite dimensional mapping $\mathcal{K}$ is impossible to obtained in real-applications, the Koopman operator has to be approximated with a predefined number of observation functions. We define a set of observable functions: $\psi_1(x_t)$, $\psi_2(x_t)$,..., $\psi_N(x_t)$. Then let $z_t = [\psi_1(x_t), \psi_2(x_t), ..., \psi_N(x_t)]^T$ with N $\gg$ n, the approximation of Koopman operator $\mathcal{K}$ is to find solution to the optimization problem (3).

$$\min_{\mathcal{K}} \sum_{t=0}^{t} \| \begin{bmatrix} z_{t+1} \\ u_t \end{bmatrix} - \mathcal{K} \begin{bmatrix} z_t \\ u_t \end{bmatrix} \|_2^2 \tag{3}$$

As depicted in the paper [8], the optimization solution $\mathcal{K}$ can be decomposed as $\mathcal{K} = [A, B]$, which is equivalent to solving the optimization problem (4).

$$\min_{A,B} \sum_{t=1}^{K} \|z_{t+1} - (A \cdot z_t + B \cdot u_t)\|_2^2 \tag{4}$$

Matrices A, B are the state-space representations of the dynamical system after being transformed by nonlinear observable function ψ. To create mapping between lifted state z_t and original state x_t, the matrix C is introduced as the solution to the optimization problem (5).

$$\min_{C} \sum_{t=1}^{K} \|X_t - C \cdot Z_t\|_2^2 \tag{5}$$

The analytical solution to (4) and (5) can be calculated as shown in Eqn. (6)

$$\begin{aligned} [A, B] &= z_{t+1}[z_t, U]^\dagger \\ C &= z_t x_t^\dagger \end{aligned} \tag{6}$$

where † denotes the Moore-Penrose pseudoinverse of matrix. In practice, when the number of collected data set is significantly larger than the dimensions of observable function (N $\ll$ k), it is more desirable to solve optimization problem (4) using Eqn. (7) than Eqn.(6). [8]

$$[A, B] = z_{t+1} \begin{bmatrix} z_t \\ U \end{bmatrix} (\begin{bmatrix} z_t \\ U \end{bmatrix}^T \begin{bmatrix} z_t \\ U \end{bmatrix})^\dagger \tag{7}$$

III. Koopman-based Approach for Model Predictive Control

A. Deep Koopman Representation for Autonomous driving

Koopman operator theory has been proven to be powerful for system identification and spectrum analysis for unsteady systems. We propose to use the lifting approach to facilitate the design of optimal control using Deep Koopman representation for Control (DKRC) in [9]. In this study, we use a high-dimensional neural network as Koopman operator to capture the nonlinear behavior of the system and accurately approximate the nonlinear dynamics into a linear system over lifted state-space.

The optimization objective to determine the A and B matrices in (4) is exactly the loss we wish to minimize when training the deep neural network lifting function. Therefore the loss function of the neural network is set to be (8) where $\psi(x_t, \theta)$ represent the neural network with original states x_t as input and θ as parameters. To further enhance the robustness of the training process, we incorporate pseudo-Huber loss function [10] as final cost value.

$$\begin{aligned} L_a &= \frac{1}{L-1} \sum_{n}^{L-1} |\psi(x_{t+1}, \theta) - (A \cdot \psi(x_t, \theta) + B \cdot u_t)| \\ L(\theta) &= \delta^2 (\sqrt{1 + (\frac{L_a}{\delta})^2} - 1) \end{aligned} \tag{8}$$

When training the neural network, the A and B matrix are updated as shown in (7) after every iterations based on the newly updated deep neural network observable function $\psi(x_t, \theta)$. Once the loss function value in (8) drops lower than termination criteria ϵ, we compute the C matrix as shown in Eqn. (6).

B. Model Predictive Control on Lifted Space(Augmented DKRC)

Once the performance of deep koopman identified model is satisfactory, we can use this new model to design optimization-based controllers such as model predictive control. As shown in Eqn. 9, the model predictive control design is based on the dynamical system approximated through the deep nerual network lifting function, hence the quadratic cost on the states has to be lifted as well.

$$\begin{aligned} \min_{u} \; & (z_N - z_{ref,N})^T C^T Q_f C (z_N - z_{ref,N}) + \\ & \sum_{t=1}^{N-1} (z_t - z_{ref,t})^T C^T Q C (z_t - z_{ref,t}) + u_t^T R u_t \\ s.t. \; & z_t = \psi(x_t, \theta) \\ & z_{t+1} = A \cdot z_t + B \cdot u_t \\ & z_{N,0} = z_0 \\ & u_{min} \leq u_t \leq u_{max} \\ & x_{min} \leq x_t \leq x_{max}, \;\; t = 1, \ldots, N \end{aligned} \tag{9}$$

IV. Experiment Result

In this section, we validate the performance of deep Koopman data-driven controller in both autonomous driving

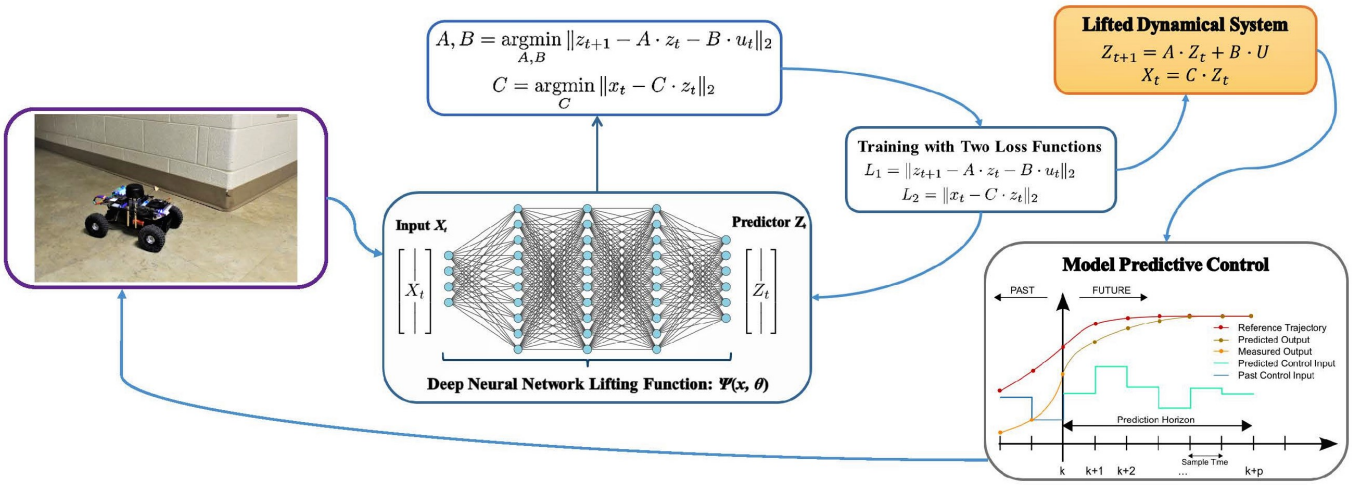


Fig. 1: Deep Koopman system training diagram

F1TENTH simulator [11] [12] as well as real F1TENTH robot. We compared the optimal control performance of the proposed deep Koopman data-driven MPC and the other two existing optimal control methods: nonlinear MPC [13] and the adaptive pure pursuit algorithm [2] [3]. Both the cost function values and the trajectories errors are considered as the optimality criteria.

A. *Tracking performance comparison analysis*

The scenario we applied with our data-driven MPC is the optimal tracking problem. The baseline models are nonlinear MPC and adaptive pure pursuit algorithms, which have been utilized in various autonomous driving community applications and the autonomous racing control for the F1TENTH environment. However, depending on the identification of the linearized model and the choice of ahead distance of the algorithm, those methods need a much more accurate model for the MPC control, and the control performance depends on the tuning of the model parameters. With the data-driven Koopman-based approach, the model can be identified in the high dimensional lifted space and hence provide more parameter robustness in the model identification.

More importantly, when the parameters of the vehicle dynamical system can not be explicitly obtained (center of gravity, cornering stiffness, racing track condition, etc.), data-driven system identification can capture the nonlinear feature of the system without direct knowledge of all parameters of the dynamical system. In our case, the nonlinear behavior of the vehicle dynamics can be properly captured through the identification process, which has proven to deliver much better control performances compared to pure pursuit or nonlinear kinematic MPC.

Since the assumption has been made that the vehicle dynamics are only partially known, parameters related to vehicle dynamics such as cornering stiffness, friction coefficient, vehicle mass are unavailable for designing the controller. The only information we can obtain from the vehicle is related to vehicle kinematics, such as wheelbase and track width. As shown in Fig. 3, due to the lack of information on vehicle dynamics parameters, the tracking performance of nonlinear kinematic MPC and adaptive pure pursuit is not as good as the data-driven MPC.

TABLE I: Tracking Performance Comparison Statistic

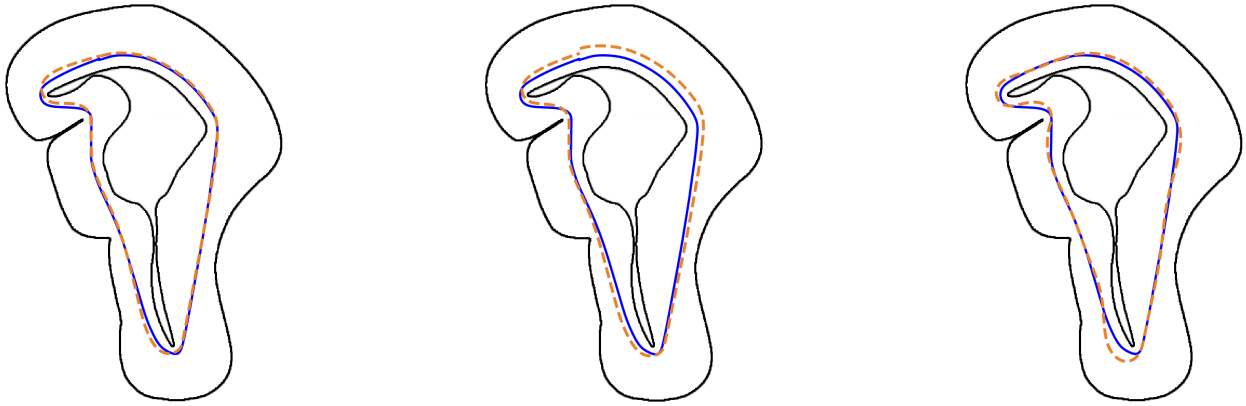
	Position Error [m]	Yaw Error [rad]	Speed Error [m/s]
Data-driven MPC	0.155	0.074	0.177
Nonlinear MPC	0.338	0.066	0.161
Pure Pursuit + PID	0.196	0.143	0.279

In Table (I) and Figure (2), the error represents the state difference between real-time vehicle pose and its closet reference pose. Data-driven MPC controller has the best performance compared to the other two controllers. In general, using geometric controller like pure pursuit achieve better position tracking performance, yet this method suffers from the unstable control output and large tracking error in sharp trajectory change. The Kinematic NMPC controller has better performance in tracking trajectory orientation and velocity since yaw angle and velocity has assigned the same cost as position states x and y. As mentioned before, data-driven MPC uses the approximated model based on recorded state and data. Consequently, the vehicle dynamics was capture in the approximation process, which gave the data-driven MPC higher position tracking accuracy than pure pursuit while maintaining the high yaw angle and velocity tracking performance from optimization-based control method.

B. *Low Lap-Time Racing Line Generation*

In order to test the controller's performance in high-speed racing scenario, a single global reference trajectory is needed for the comparison. It is preferable that this global trajectory to include high-speed corner, harsh brake and acceleration so that extreme driving behavior can be tested.

To generate such racing line, we first define a initial global trajectory χ_0 as the reference trajectory, which can either be human driver's input or middle line of the track. Then we



(a) Data-driven MPC (b) Kinematic NMPC (c) Pure Pursuit + PID Control

Fig. 2: Trajectory tracking of three different vehicle controllers. (a): Data-driven MPC (b): Kinematic NMPC (c): Pure Pursuit + PID

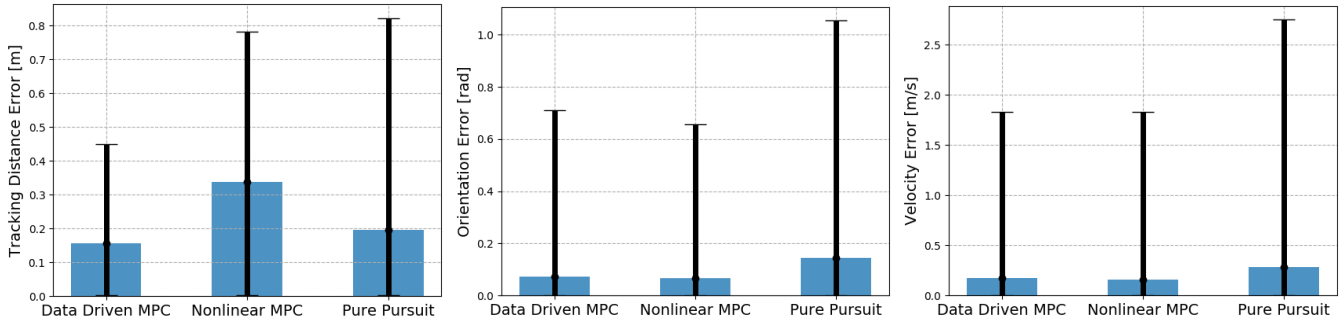


Fig. 3: Tracking Error Comparison, blue bar represents the mean error value while black bar shows the range of error

perform global racing-line iteration by recording the vehicle pose based on the solution a real-time local planning and control optimization problem. Enlightened by N.Kapania's

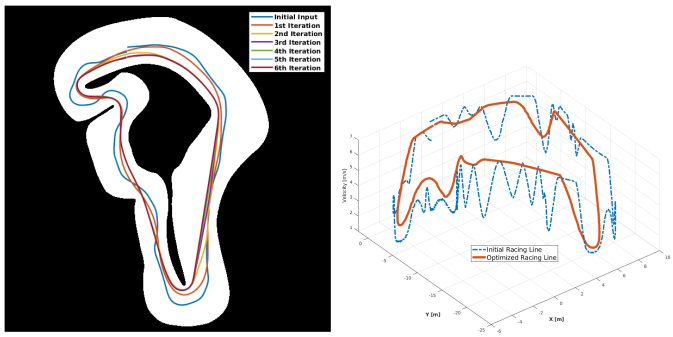


Fig. 4: Iterative Progress in Generating Global Trajectory over a closed track

work in his Ph.D. thesis of Stanford University, the local planning optimization problem is established as in Eqn. (11). This optimization is based on the 2D pose of robot $s_t = (x_t, y_t, \theta_t)$ where (x_t, y_t) defines the position and θ_t defines the orientation. In this case, the goal is to find the local path that satisfy the combination of two terms: 1. the shortest path between vehicle pose and local goal pose selected from the racing line 2. the overall curvature of the local path.

Since the racing-line is designed for ackermann steering vehicle, we incorporate the non-holonomic constraints as suggested in the trajectory optimization paper [14]. Consequently, the consecutive pose update is written as a state-space function h that update the vehicle pose s_t based on two control input $\delta\theta$ and δS.

$$s_{t+1} = h(s_t, \delta\theta, \delta S) \tag{10}$$

where $\delta\theta$ is the change of orientation and δS is the length of curve between two curves. More detail of the pose update process can be found in the work [14] by C.Rosmann.

With the state update equation and initial vehicle pose information, the local planning problem is to minimize the cost function with two terms as suggested in (11).

$$\begin{aligned} \min_{\delta\theta_0, \delta\theta_1, \ldots, \delta\theta_{t-1}} & \sum_{i=0}^{i=t-1} (1 - w_d) \cdot \kappa_i + w_d \cdot \|x_i - x_g, y_i - y_g\|_2^2 \\ \textbf{s.t. } & \kappa_i = \frac{|\delta\theta_i|}{\delta S} \\ & s_{i+1} = h(s_i, \delta\theta_i, \delta S) \\ & s_g \in \chi_j \end{aligned} \tag{11}$$

where s_g is the local goal pose selected from the jth lap of global trajectory χ_j, κ_i represent the average curvature between two pose, δS represent the arc length two adjacent poses. In this case, δS is a pre-defined constant value. The minimum distance weight w_d control the balance between

minimum curvature and minimum distance for the optimization. The output of the optimization problem in Eqn.(11) is the a array of the vehicle pose that contains the information of vehicle's position and orientation.

Based on our experiment, it took around 4-5 laps for the global racing line to converge. With the given trajectory pose and orientation information, next step is to calculate the optimal velocity profile that can achieve the minimum lap-time. We can use dynamic programming approach to find the optimal velocity profile for the racing line based on the pose information (x_t, y_t, θ_t). As suggested in the work [15] by M. Althoff, we incorporate tire-friction cycle as the maximum tire-force constraint. The detail of dynamic programming optimization is shown in (12).

$$\begin{aligned} \min_{u_0,\ldots,u_t} \sum_{i=0}^{i=t-1} & \frac{2ds_i}{u_{i+1}+u_i} \\ ds_i &= \|x_{i+1}-x_i, y_{i+1}-y_i\|_2^2 \\ w_i &= \frac{|\theta_{i+1}-\theta_i|}{dt_i} \\ v_i &= \frac{u_{i+1}+u_i}{2} \\ a_i &= \frac{u_{i+1}-u_i}{dt_i} \\ \sqrt{a_i^2 + w_i \cdot v_i} &\leq \mu \end{aligned} \tag{12}$$

The friction coefficient between tire and road directly affect the amount of traction available for the vehicle. Since the friction cycle restrict the lateral and longitudinal acceleration, a small μ value will lead to a smoother operation while a high μ could deliver lower lap-time with more aggressive behavior.

Table (II) shows the analysis of how minimum lap-time changes with respect to two variables: minimum distance weight w_d as well as tire-road friction coefficient μ. As depicted in the table, it is not surprising that a higher tire-road friction of coefficient leads to a lower lap-time. However, the experiment's results suggested that the best choice of minimum distance weight is when w_d equals to around 0.2 and 0.65. One insight we can grasp from this pattern is neither low distance nor low curvature leads to the fastest lap-time, because low distance may compromise cornering speed for the sake of low distance while low curvature will yield longer distance despite the overall higher cornering speed. The optimal balance between minimum curvature and distance requires detailed analysis. On the other hand, this optimal balance may also vary with different track as different track has different character. Once the track is changed, the optimal combination of w_d and μ has to be recalculated.

C. End-to-End Model Deployment to a Scaled Racing Vehicle

To better test the performance of deep Koopman representation of control on the real vehicle under high speed racing scenario, we conduct the experiment on the F1TENTH robot in the basement of Clemson Fluor Daniel Engineering Innovation Building. In this case, an open area as which contains both straight line and corners are selected (shown in figure 5) to test both the speeding and steering performance of deep Koopman MPC controller.

During the initial part of the test, we first conduct the system identification of the vehicle dynamics. In this process, we ask a human to operate the vehicle within the selected area. When controlling the vehicle, it is preferable that the control input to be noisy so that more vehicle behaviors can be explored.

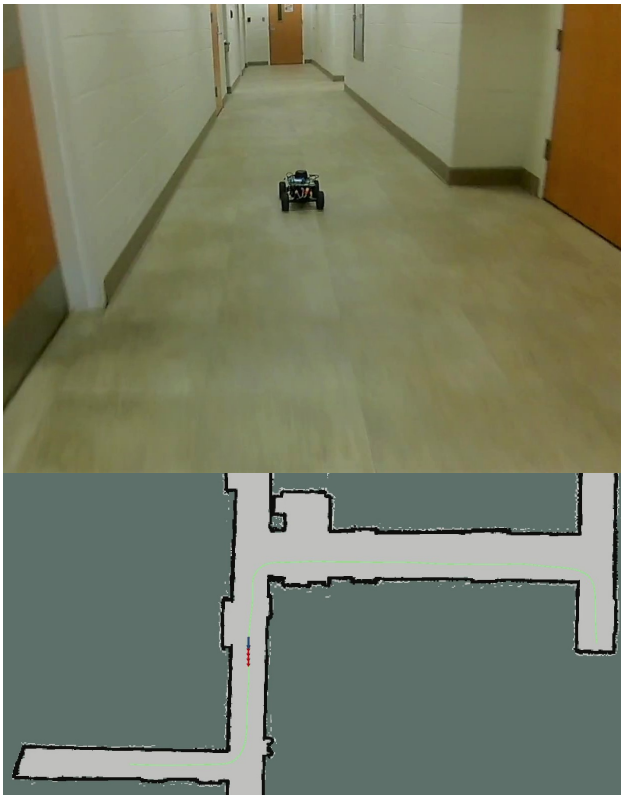


Fig. 5: End-to-end Model deployment to a 1/10th scale autonomous racing robot

Once the system identification is complete, we use the iterative global trajectory generation methods to provide a fast racing line to test the controller's performance in high speed. As suggested in the Ph.D thesis by N. Kapania from Stanford [16], the value of minimum distance weight w_d in optimization in Eqn. (11) can effect the time consumed for navigation. Based on our experiment result, we choose w_d around 0.65 as the final minimum distance weight as it delivered the shortest navigation time. As for the choice of tire-road friction of coefficient, we set μ to be 0.5 to avoid extreme control behavior, hence lower the risk of dangerous crash.

The statistics of trajectory evolution is shown in table (III), the navigation time converges to a minimum value of 9.228 seconds within only 4 laps of iteration.

The deep Koopman model predictive control is used for the vehicle control to track the optimized racing line. With the assistance of a deep neural network as the lifting function, we are able to control the number of lifted state-space so that we can use the lowest number of lifted dimension without compromising the accuracy of the vehicle dynamics too much.

TABLE II: Minimum Time Statistic over Different Road Friction and Minimum Distance Weight

	μ = 0.2	μ = 0.3	μ = 0.4	μ = 0.5	μ = 0.6	μ = 0.7	μ = 0.8	μ = 0.9
w_d = 0.05	19.18	15.6134	13.4987	12.0556	10.994	10.1829	9.5495	9.1014
w_d = 0.2	19.14	15.5798	13.4639	12.026	10.9656	10.1492	9.5318	9.0766
w_d = 0.35	19.19	15.605	13.49	12.0517	10.99	10.17	9.543	9.094
w_d = 0.50	19.1873	15.6342	13.4983	12.0573	10.9985	10.181	9.5533	9.1054
w_d = 0.65	19.147	15.5715	13.46	12.03	10.962	10.15	9.527	9.0956
w_d = 0.8	19.2392	15.6635	13.5218	12.0779	11.0146	10.2	9.5764	9.1387
w_d = 0.95	19.3256	15.7311	13.602	12.156	11.08	10.2555	9.6281	9.2004
w_d = 1.0	19.883	16.1719	13.9724	12.4789	11.3818	10.5328	9.8861	9.4607

Fig. 6: Iterative approach to generate trajectory over Clemson Fluor Daniel Basement Map

TABLE III: Navigation Time Comparison Statistic

	Initial Path	Lap 1	Lap 2	Lap 3
Time [sec]	11.946	9.737	9.652	9.228
Mean speed [m/s]	3.0534	3.457	3.441	3.595

As a result, the update frequency of the MPC solver is good enough for high speed navigation.

V. Conclusion and Discussion

In this paper, we propose a fully data-driven approach for system identification and control using deep neural network as Koopman operator. Not only did the nonlinear dynamics are properly approximated, the learning-based Koopman operator is capable of controlling the number of lifted dimension so that the computational efficiency and the accuracy of system can be balanced properly.

The experiment result suggests that data-driven Koopman MPC is able to achieve better tracking performance than vehicle controllers such as pure pursuit and kinematic nonlinear MPC due to the highly accurate approximated model. More importantly, the entire learning process of the system dynamics is entirely data-driven without a-priori knowledge of the dynamical system, which allows the real-time modification in the system to achieve the best control performance.

On the other hand, the deep Koopman model predictive control can safely control the vehicle to track a high-speed optimized racing line without knowing the detailed vehicle information like cornering stiffness or center of gravity, which indicate the potential of deep Koopman MPC in autonomous racing scenario with limited vehicle information.

References

[1] G. M. Hoffmann, C. J. Tomlin, M. Montemerlo, and S. Thrun, "Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing," in *2007 American Control Conference*, 2007, pp. 2296–2301.

[2] W. Wang, T. Hsu, and T. Wu, "The improved pure pursuit algorithm for autonomous driving advanced system," in *2017 IEEE 10th International Workshop on Computational Intelligence and Applications (IWCIA)*, 2017, pp. 33–38.

[3] Y. Chen, Y. Shan, L. Chen, K. Huang, and D. Cao, "Optimization of pure pursuit controller based on pid controller and low-pass filter," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 3294–3299.

[4] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "Predictive active steering control for autonomous vehicle systems," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 3, pp. 566–580, 2007.

[5] Y. Xiao, X. Zhang, X. Xu, X. Liu, and J. Liu, "A deep learning framework based on koopman operator for data-driven modeling of vehicle dynamics," 2020.

[6] B. Huang, X. Ma, and U. Vaidya, "Feedback stabilization using koopman operator," in *2018 IEEE Conference on Decision and Control*. IEEE, 2018, pp. 6434–6439.

[7] W. Hao and Y. Han, "Data driven control with learned dynamics: Model-based versus model-free approach," 2020.

[8] M. Korda and I. Mezić, "Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control," *Automatica*, vol. 93, p. 149–160, Jul 2018. [Online]. Available: http://dx.doi.org/10.1016/j.automatica.2018.03.046

[9] Y. Han, W. Hao, and U. Vaidya, "Deep learning of koopman representation for control," in *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 1890–1895.

[10] J. T. Barron, "A general and adaptive robust loss function," 2019.

[11] M. O'Kelly, H. Zheng, D. Karthik, and R. Mangharam, "F1tenth: An open-source evaluation environment for continuous control and reinforcement learning," in *Post Proceedings of the NeurIPS 2019 Demonstration and Competition Track*, ser. Proceedings of Machine Learning Research, H. J. Escalante and R. Hadsell, Eds. PMLR, 2020.

[12] M. O'Kelly, V. Sukhil, H. Abbas, J. Harkins, C. Kao, Y. V. Pant, R. Mangharam, D. Agarwal, M. Behl, P. Burgio *et al.*, "F1/10: An open-source autonomous cyber-physical platform," *arXiv preprint arXiv:1901.08567*, 2019.

[13] X. Du and K. K. Tan, "Autonomous vehicle velocity and steering control through nonlinear model predictive control scheme," in *2016 IEEE Transportation Electrification Conference and Expo, Asia-Pacific (ITEC Asia-Pacific)*, 2016, pp. 001–006.

[14] C. Rösmann, F. Hoffmann, and T. Bertram, "Integrated online trajectory planning and optimization in distinctive topologies," *Robotics and Autonomous Systems*, vol. 88, pp. 142–153, 2017.

[15] M. Althoff, M. Koschi, and S. Manzinger, "Commonroad: Composable benchmarks for motion planning on roads," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 719–726.

[16] N. R. Kapania, "Trajectory planning and control for an autonomous race vehicle," Ph.D. dissertation, Stanford University, 2016.