Data-driven identification of vehicle dynamics using Koopman operator

Vít Cibulka*, Tomáš Haníš† and Martin Hromčík‡
Department of Control Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague
Email: *cibulka.vit@fel.cvut.cz †hanis.tomas@fel.cvut.cz ‡hromcik.martin@fel.cvut.cz

Abstract—This paper presents the results of identification of vehicle dynamics using the Koopman operator. The basic idea is to transform the state space of a nonlinear system (a car in our case) to a higher-dimensional space, using so-called basis functions, where the system dynamics is linear. The selection of basis functions is crucial and there is no general approach on how to select them, this paper gives some discussion on this topic. Two distinct approaches for selecting the basis functions are presented. The first approach, based on Extended Dynamic Mode Decomposition, relies heavily on expert basis selection and is completely data-driven. The second approach utilizes the knowledge of the nonlinear dynamics, which is used to construct eigenfunctions of the Koopman operator which are known by definition to evolve linearly along the nonlinear system trajectory. The eigenfunctions are then used as basis functions for prediction. Each approach is presented with a numerical example and discussion on the feasibility of the approach for a nonlinear vehicle system.

Index Terms—Koopman operator, eigenfunctions, basis functions, data-driven methods, identification

I. INTRODUCTION

Car is a highly nonlinear dynamical system which has been used for decades without any sort of sophisticated automatic control system. Our goal is to create full-time-full-authority control system, which means that the driver will rather set the reference for the car, and the ultimate control over the car inputs (steering angle, motor torques) will lie with the control system, protecting the driver from himself. However, control of nonlinear systems is not an easy task.

The Koopman operator is becoming an increasingly popular tool for nonlinear control. The Koopman operator is a linear operator which can in theory approximate any nonlinear system. The idea lies in lifting the state space of the nonlinear system to a higher-dimensional space by means of a nonlinear transformation. The lifted system should then be evolving linearly along with the original nonlinear system. The lifted state space is then transformed back with a linear transformation to the original state space and the nonlinear states (or system outputs) are recovered.

Such an approach is suitable for any linear control-design method such as LQG, MPC and $H_{\infty}$.

In this paper, we present the results of application of the Koopman operator framework to vehicle system in an attempt to create a linear representation of a vehicle and use it to control the vehicle optimally with linear MPC.

II. SINGLETEACK MODEL

The singletrack model is depicted in Fig[1] The model was derived from 16-state twin-track model described in [1]. The states and parameters of this model are a subset of states and parameters of the full twin-track model.

State vector of the singletrack model is $\begin{bmatrix} v_x (m/s) & v_y (m/s) & \dot{\psi} (rad/s) & \dot{\rho}_f (rad/s) & \dot{\rho}_r (rad/s) \end{bmatrix}^T$, where $v_x$ is longitudinal velocity, $v_y$ lateral velocity, $\dot{\psi}$ yawrate and $\dot{\rho}_f/\rho_r$ are front/rear wheel angular rates.
This model has 4 wheels, with two wheels always being in the same place. This allows for usage of asymmetric tire models, reduces the work needed to transition to twin-track model and is less error-prone than the standard approach (with only two tires, the user has to remember that each tire should generate twice as much force).

The vehicle body is modeled as a rigid body using Newton-Euler equations.

\[
m_v \left( \begin{array}{c} \dot{v}_x \\ \dot{v}_y 
end{array} \right) + \dot{\psi} \left[ \begin{array}{c} -v_y \\ v_x 
end{array} \right] = \sum_{i=1}^{4} \left[ \begin{array}{c} F_{i,x} \\ F_{i,y} 
end{array} \right] - \frac{1}{2} c_w \rho A_w \sqrt{v_x^2 + v_y^2} \left[ \begin{array}{c} v_x \\ v_y 
end{array} \right]
\]

(1)

\[
J_{zz} \ddot{\psi} = \sum_{i=1}^{4} r_i F_{i,y}
\]

(2)

\[
J_{Rx}\ddot{R}_x = M_{a,R_x} - M_{b,R_x} \text{sign}(\dot{R}_x) - rF_{R_i,x}, \forall i = 1, 3
\]

(3)

Where

\[
r = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix}^T = \begin{bmatrix} t_v \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ t_v \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} -l_i \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} -l_i \\ 0 \\ 0 \\ 0 \end{bmatrix}
\]

(4)

is the vector describing position of each wheel with respect to the center of gravity. The wheels are numbered in this order: front-left, front-right, rear-left, rear-right. \( m_v \) is the vehicle mass, \( F_{i,x/y} \) is a force acting on \( i \)-th wheel along x/y axis in body-fixed coordinates. \( F_{R_i,x} \) is a force acting along x axis in wheel coordinate system (direct output of the tire model).

The term \( -\frac{1}{2} c_w \rho A_w \sqrt{v_x^2 + v_y^2} \left[ \begin{array}{c} v_x \\ v_y 
end{array} \right] \) is an approximation of air-resistance, \( c_w \) is a drag coefficient, \( \rho \) is air density and \( A_w \) is the total surface exposed to the air flow. \( J_{zz} \) is the vehicle inertia about z-axis. \( J_{Rx} \) the wheel inertia about y-axis. Inputs are wheel torques \( M_{a,R_i} \) (throttle), \( M_{b,R_i} \) (break) and steering angles \( \delta_{f/r} \).

Since this is a 4-wheel singletrack model, the following holds:

\[
\dot{R}_z = \dot{R}_z 
\]

(5)

\[
\dot{R}_x = \dot{R}_x 
\]

(6)

The forces \( \begin{bmatrix} F_{R,i,x} \\ F_{R,i,y} \end{bmatrix} \) are calculated using the “Pacejka magic formula” \([2]\)

\[
F = D \cos(C \arctan(Bx - E(Bx - \arctan(Bx))))
\]

(7)

The same formula can be used for calculating \( F_x \) (tire longitudinal force) and \( F_y \) (tire lateral force) with a different set of parameters for each. The argument \( x \) can be either sideslip angle \( \alpha \) or longitudinal slip \( \kappa \) (see \([2]\)) for calculating \( F_y \) or \( F_x \) respectively. The parameters B,C,D and E are usually time-dependent. This work uses the Pacejka tire model \([2]\) with coefficients from the Automotive challenge 2018 organized by Rimac Automobili.

The transformation of tire forces from wheel-coordinate system to car coordinate system is done as follows

\[
\begin{bmatrix} F_{i,x} \\ F_{i,y} \end{bmatrix} = \begin{bmatrix} \cos(\delta_i) & -\sin(\delta_i) \\ \sin(\delta_i) & \cos(\delta_i) \end{bmatrix} \begin{bmatrix} F_{R,i,x} \\ F_{R,i,y} \end{bmatrix}
\]

(8)

A. 3 state singletrack

To further simplify the model, equation (3) can be omitted, resulting in a model with only 3 states: \( [v_x \ v_y \ \dot{\psi}]^T \). The inputs are then longitudinal slips (which were previously derived from (3)) and steering angles. This model will be used for the Eigenfunction approach in III-B

B. 3 state singletrack without tire model

To simplify the model even more, one can omit the tire model and use the tire forces \( \begin{bmatrix} F_{i,x} \\ F_{i,y} \end{bmatrix} \) as input, assuming the existence of a higher level control system controlling the tire forces and thus securing the assumption that the car can be controlled directly by force reference.

This model will be used for the Extended Dynamic Mode Decomposition (EDMD) \([3]\) approach in III-A

III. THE KOOPMAN OPERATOR

A. Extended Dynamic Mode Decomposition approach

1) Framework description: The Koopman operator framework for controlled systems, as described in \([4]\) will be reviewed now. Let us assume uncontrolled discrete nonlinear dynamical system with state \( x \) and dynamics \( f(\cdot) \)

\[
x^{+} = f(x) \\
y = g(x)
\]

(9)

with \( x \) being the current state and \( x^+ \) the next state.

The Koopman operator \( K : \mathcal{F} \to \mathcal{F} \) is defined as

\[
(K\psi)(x) = \psi(f(x))
\]

(10)

for each basis function \( \psi : \mathbb{R}^N \to \mathbb{R} \) where \( \mathcal{F} \) is the space of basis functions.

\( K \) is infinite-dimensional linear operator, which can be approximated by EDMD. The approximation is done by solving optimization problem

\[
\min_A \sum_{j=1}^{K} ||\psi(x_j^+) - A\psi(x_j)||_2^2
\]

(11)

where \( \psi = [\psi_1(x) \ \psi_2(x) \ \ldots \ \psi_N(x)]^T \). The states \( x_j \) and \( x_j^+ \) are obtained by simulating the nonlinear system model, \( K \) is the cardinality of the simulated dataset \( \mathcal{D} = \{x_j \ x_j^+ \ y_j\}^{K}_{j=1} \). Note that the states \( x_j \) do not have come from some trajectory of the system. When the system model is available, it is sufficient to sample the state space with \( x_j \) and perform a one-step simulation to obtain the \( x_j^+ \).

The matrix \( A \) now defines a discrete linear system which approximates the nonlinear dynamics from (9)

\[
z^{+} = Az \\
\dot{y} = Cz
\]

(12)

where \( z = \psi(x) \) and \( \dot{y} \) is the output estimate. The matrix \( C \) is obtained in a similar fashion

\[
\min_C \sum_{j=1}^{K} ||y_j - C\psi(x_j)||_2^2
\]

(13)
For controlled system
\[
\begin{align*}
  x^+ &= f(x, u) \\
  y &= g(x)
\end{align*}
\] (14)

the approach is similar. The optimization problem (11) changes to
\[
\min_{A,B} \sum_{j=1}^K \| \psi(x_j^+) - (A\psi(x_j) + Bu_j) \|^2_2
\] (15)

while (13) stays the same. The Koopman operator \( \mathcal{K} \) is approximated both by \( A \) and \( B \) and acts as a one-step predictor. Note that the learning dataset now consists of \( \mathcal{D} = [x_j \ x_j^+ \ y_j \ u_j]^K_{j=1} \). See [4] for more information.

This approach will be showed on the model which does not include the tire nonlinearities, described in [II-B] because the selection of \( \psi \) and \( \mathcal{D} \) is much more difficult when the tire nonlinearities are present.

2) Data selection: The learning dataset was selected as follows. The velocities \( v_x \) and \( v_y \) were uniformly sampled in the interval \([-30, 30] \) m/s, yawrate was uniformly sampled in interval \([-10, 10] \) rad/s. The number of samples was 15 for each state, resulting in 3 1-by-15 vectors.

The input forces to the model were sampled uniformly in the interval \([-F_{m}, F_{m}] \) with \( F_{m} \in \{1, 5, 10, 100\} \), each interval with 15 samples, resulting in 4 4-by-15 vectors. The columns of sampled vectors were combined into all possible combinations, resulting in 4 \( 15^4 \) values in the dataset \( \mathcal{D} \).

Note that the tire force of a regular car is usually in order of thousands of Newtons. However, the results were much better with relatively small inputs forces. Large forces were too dominant and their influence on the system overshadowed the system dynamics and since there are no tire nonlinearities, the system can be sufficiently excited with small input forces.

3) Basis selection: There is unfortunately no general recommendation to determine which basis functions to choose for a given system. Some systems show satisfactory results with thin plate spline radial basis functions (see [5]). In our case, polynomial basis functions yielded much better results. A polynomial basis consists of monomials of the elements of the state vector.

\[
B_k = \{e_a^b \cdot \psi^c | a, b, c \in \{0, 1, 2, \ldots, k\} \}
\] (16)

where \( k \) is the order of the basis \( B_k \). Bases with various orders were compared, the comparison was made using averaged root mean square error (RMSE)

\[
RMSE = 100 \sqrt{\frac{\sum_k \| x_{k_{\text{koop}}}(kT_s) - x_{\text{real}}(kT_s) \|^2}{\sum_k \| x_{\text{real}}(kT_s) \|^2}}
\] (17)

calculated over 3375 trajectories with the same distribution of initial conditions as the learning dataset \( \mathcal{D} \). The length of each trajectory was 30 samples, with \( T_s = 0.01 \). The results can be seen in Fig. 2.

One might think the error would decrease with adding more basis functions but error grows exponentially as the order increases, which means that one cannot simply add more basis functions and expect the error to diminish. When considering linear Koopman system
\[
\dot{\psi}(x) = A\psi(x),
\] (18)

the more basis functions are in \( \psi \), the more derivatives \( \dot{\psi} \) have to be expressed by \( \dot{\psi} \). The derivatives of the polynomial basis grow in complexity with increasing order, which makes it harder to express them, so the result makes sense, although it might seem counter-intuitive at first glance.

Results: In our experiments, a polynomial basis with order 7 was used. The Koopman operator was approximated using the dataset described above. The results can be seen in Fig. 3.

The yawrate is tracked without error because it is simply an integral of the input force as seen in [2].

B. Basis functions from data

To deal with the problems of basis function selection, the approach from [5] can be used. The approach consists of creating the eigenfunctions of the Koopman operator from data and using them as basis functions \( \psi \). The matrices \( A \) and \( B \) are calculated separately, contrary to the previous approach, which allows for optimizing the resulting system for a multiple-step prediction as opposed to one-step prediction of the previous approach. More in [5].

The basic idea of an eigenfunction will now be established. Let us consider uncontrolled nonlinear continuous system
\[
\frac{d}{dt} x = f(x)
\] (19)

Starting in \( x_0 \), the system will get to state \( x_t \) in time \( t \). The state vector \( x_t \) will be transformed with a function \( \psi(x_t) \), defined as
\[
\psi(x_t) = \psi(x_t)_{\lambda, g} = e^{\lambda t} g(x_0)
\] (20)
Fig. 3. This figure shows the comparison between the real nonlinear system and linear system approximated by the Koopman operator with polynomial basis of order 7. The input is random with uniform distribution in the interval $[-10^4, 10^4]$. The initial conditions were chosen randomly as $(v_x, v_y, \psi) = (20.8533, -0.7222, -4.3479)$. The prediction is restarted after every 50 samples, the restart is visualized by segmenting the whole trajectory into pieces, each 50 samples long.

for arbitrary eigenvalue $\lambda$ and function $g: \mathbb{R}^n \rightarrow \mathbb{R}$. The time derivative of $\psi(x_t)$ is

$$
\frac{d}{dt}\psi(x_t) = \frac{d}{dt} e^{\lambda T} g(x_0) = \lambda e^{\lambda T} g(x_0) = \lambda \psi(x_t)
$$

(21)

We see that $\psi(x_t)$ evolves linearly with the system (19). For $\psi(x_t) = [\psi_1(x_t) \ \psi_2(x_t) \ \ldots \ \psi_{N_\psi}(x_t)]^T$ we would get

$$
\frac{d}{dt} \begin{bmatrix}
\psi_1(x_t) \\
\psi_2(x_t) \\
\vdots \\
\psi_{N_\psi}(x_t)
\end{bmatrix} =
\begin{bmatrix}
\lambda_1 & 0 & \cdots & 0 \\
0 & \lambda_2 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \lambda_{N_\psi}
\end{bmatrix}
\begin{bmatrix}
\psi_1(x_t) \\
\psi_2(x_t) \\
\vdots \\
\psi_{N_\psi}(x_t)
\end{bmatrix}
$$

(22)

$$
\frac{d}{dt} z = Az
$$

(23)

for $z = \psi(x)$. Choosing (20) as basis functions immediately yields the diagonal $A$ matrix. Notice that in this case, the Koopman system is derived as continuous system, contrary to the discrete-time derivation in III-A. This is done simply because of the continuous-time definition of (20), the system can be discretized after the whole procedure.

The idea is to select a set of initial conditions $\Gamma$ for the system (19) and simulate the system for a time $T$ with a sampling period $T_s$. Then for each sampled data point $x_{kT_s}$, evaluate a set of eigenfunctions

$$
\psi(x_{kT_s})_{\lambda,g} = e^{\lambda g T_s} g(x_0)
$$

(24)

for some $\lambda \in \Lambda$, $g \in G$, where $x_0 \in \Gamma$ is the initial condition of the state $x_{kT_s}$.

The values of $\psi(x)_{\lambda,g}$ can be interpolated in order to approximate their values for states $x$ which are not in the learning dataset. The approximation will be denoted as $\hat{\psi}(x)_{\lambda,g}$.

The set $\Gamma$ should be a non-recurrent set, meaning that the sampled trajectories of the system (19) should not return to the states from set $\Gamma$. This condition is not difficult to fulfill with a dissipative system such a car. If, for example, the set consisted of states with the same kinetic energy, none of the trajectories would return to the set $\Gamma$, because the system naturally loses its kinetic energy as time progresses. For more information and proofs of the above stated facts, see [5].

The matrix $C$ can be obtained in a similar fashion as in the previous case. To approximate the output $y = g(x)$, solve

$$
\min_C \sum_{i=1}^{M} ||g(x_i) - C \hat{\psi}(x_i)||_2^2
$$

(25)

where $M$ is the total number of sampled states.
Fig. 4. Set of states with constant kinetic energy equivalent to 1300 kg car riding at 100 km/h. Each vertex of the mesh is one of the 441 initial states from which the learning dataset was constructed.

The controlled case is not considered here, for more information on the derivation of the $B$ matrix, see [5].

Results: The simulation experiments were performed with the model described in II-A. The non-recurrent set $\Gamma$ was chosen as set of states with the same kinetic energy, the reasoning for this selection was mentioned above.

The level of the kinetic energy was set to $500 kJ$, which is an energy equivalent of a car weighting 1300 kg, riding at 100 km/h. The set is depicted in Fig. 4.

Next, the nonlinear system was simulated from 441 initial states from the set $\Gamma$ (the set $\Gamma$ is visualized in Fig. 4) for sufficiently long time $T$ in order to cover the state space with enough data, in our case $T = 1s$. The trajectories were sampled at sampling rate $T_s = 0.01s$.

Then the eigenfunctions $\psi_{\lambda,g}$ were calculated. The eigenvalues were chosen in a similar fashion as in [5]. By applying Dynamic Mode Decomposition (DMD) algorithm [6] to the dataset, three eigenvalues $\Lambda_{DMD}$ were obtained. Additional eigenvalues were constructed as linear combinations of elements from $\Lambda_{DMD}$, more in [5].

For the functions $g$, thin plate spline radial basis functions were used:

$$ g(x) = ||x - x_c||^2 \log(||x - x_c||) $$

The centers $x_c$ were chosen randomly with normal distribution $N(\mu, \sigma^2)$, where $\mu$ and $\sigma$ are the mean and standard deviation of the whole dataset. Note that instead of interpolating the functions $\psi_{\lambda,g}$, nearest neighbour was chosen. This was done for simplicity and it yielded good results.

The approximated system was evaluated using 2000 randomly selected initial conditions, all within the ellipsoid in Fig. 4. Both the Koopman system and the real one were then simulated for 0.5 s (which is unnecessarily long for MPC control, but it demonstrates the capabilities of the approach).

The error of each trajectory is again calculated using RMSE, defined in (17).

Results can be seen in Fig. 5. There are two clusters of initial conditions with large prediction error. These are the areas with very little data. The trajectories converge to planes, forming X-shape around origin. The shape is caused by the vehicle not having enough energy to spin and going into a drift while decreasing $v_y$ and $\dot{\psi}$.

IV. CONCLUSION

This paper presented two algorithms based on the Koopman operator for identifying vehicle dynamics.
The EDMD approach in III-A showed good result, but only with the model II-B which does not include the tire nonlinearities which are the greatest challenge in vehicle dynamics.

The eigenfunction approach in III-B yielded very promising results on a model with the tire nonlinearities II-A. This approach will be the focus of our future work including the controlled case, which wasn’t discussed in this work, and its implementation with linear MPC.

ACKNOWLEDGMENT

This research was supported by the Czech Science Foundation (GACR) under contract No. 19-16772S. We would like to thank Milan Korda for valuable discussions on the topic of eigenfunctions and Koopman theory.

REFERENCES

[1] D. Schramm, M. Hiller, and R. Bardini, Vehicle Dynamics. Springer Berlin Heidelberg, 2014.
[2] H. Pacejka, Tire and Vehicle Dynamics. Elsevier LTD, Oxford, 2012. [Online]. Available: https://www.ebook.de/de/product/18341528/hans_pacejka_tire_and_vehicle_dynamics.html
[3] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, “A data-driven approximation of the koopman operator: Extending dynamic mode decomposition,” Journal of Nonlinear Science, vol. 25, no. 6, pp. 1307–1346, jun 2015.
[4] M. Korda and I. Mezić, “Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control,” Automatica, vol. 93, pp. 149–160, jul 2018.
[5] M. Korda and I. Mezić, “Learning koopman eigenfunctions for prediction and control: the transient case.”
[6] P. J. Schmid, “Dynamic mode decomposition of numerical and experimental data,” Journal of Fluid Mechanics, vol. 656, pp. 5–28, jul 2010.