Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Model Predictive Control of a Vehicle using Koopman Operator

Vít Cibulka * **Tomáš Haniš** * **Milan Korda** *,** **Martin Hromčík** *

* *Dept. of Control Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague, The Czech Republic (emails: cibulka.vit@fel.cvut.cz, hanis.tomas@fel.cvut.cz, korda.milan@fel.cvut.cz, hromcik.martin@fel.cvut.cz)*
** *CNRS, Laboratory for Analysis and Architecture of Systems, Toulouse, France (email: korda@laas.fr)*

Abstract: This paper continues in the work from Cibulka et al. (2019) where a nonlinear vehicle model was approximated in a purely data-driven manner by a linear predictor of higher order, namely the Koopman operator. The vehicle system typically features a lot of nonlinearities such as rigid-body dynamics, coordinate system transformations and most importantly the tire.
These nonlinearities are approximated in a predefined subset of the state-space by the *linear* Koopman operator and used for a *linear* Model Predictive Control (MPC) design in the high-dimension state space where the nonlinear system dynamics evolve *linearly*. The result is a nonlinear MPC designed by linear methodologies.
It is demonstrated that the Koopman-based controller is able to recover from a very unusual state of the vehicle where all the aforementioned nonlinearities are dominant. The controller is compared with a controller based on a classic local linearization and shortcomings of this approach are discussed.

Keywords: Koopman operator, Eigenfunction, Eigenvalues, Basis functions, Data-driven methods, Model Predictive Control

1. INTRODUCTION

A vehicle is a nonlinear system that is becoming more interesting from the control engineering point of view with the ever increasing number of electric vehicles. This gives an opportunity for sophisticated control systems to take the place of old-fashioned solutions which are currently present in the majority of vehicles today.

This paper examines the nonlinear control of the vehicle described by a linear predictor which is valid in a predefined subset state space, which allows for exploitation of linear control methods on the nonlinear system. The linear predictor used in this paper is the Koopman operator (Koopman (1931)).

The Koopman operator, an increasingly popular tool for global linearization and analysis of nonlinear dynamics (Mezić (2005), Korda and Mezić (2019) ,Korda and Mezić (2018), Mezić and Banaszuk (2004)), is used in this work to approximate the vehicle nonlinear dynamics in order to achieve a linear representation of the system in a predefined subspace of the state space.

This paper continues in the work from Cibulka et al. (2019), where different methods for global linearization of the single-track model were used. The most promising method (described in detail in Korda and Mezić (2019)) is used for approximation of autonomous and controlled behaviour of the nonlinear vehicle system by a high-dimensional linear system. The resulting linear system is then used for *linear* Model Predictive Control (MPC) design and verified against a MPC based on local linearization which was the prevalent approach of tackling nonlinear systems in the past.

2. SINGLE-TRACK MODEL

The vehicle model derived in Cibulka et al. (2019) will be reviewed here. The model is depicted in Fig. 1. State vector of the single-track model is

$$\left[v_x(\mathrm{m\,s^{-1}}),\ v_y(\mathrm{m\,s^{-1}}),\ \dot{\psi}_z(\mathrm{rad\,s^{-1}})\right]^\top, \tag{1}$$

where v_x is longitudinal velocity, v_y lateral velocity and $\dot{\psi}_\mathrm{z}$ is yawrate. Inputs to the model are rear longitudinal slip ratios κ_r and front steering angle δ_f.

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

$$m_v\left(\begin{bmatrix}\dot{v}_\mathrm{x}\\ \dot{v}_\mathrm{y}\end{bmatrix}+\dot{\psi}_\mathrm{z}\begin{bmatrix}-v_\mathrm{y}\\ v_\mathrm{x}\end{bmatrix}\right)=\sum_{i=1}^{4}\begin{bmatrix}F_{i,\mathrm{x}}\\ F_{i,\mathrm{y}}\end{bmatrix}-\frac{1}{2}c_\mathrm{w}\rho A_\mathrm{w}\sqrt{v_\mathrm{x}^2+v_\mathrm{y}^2}\begin{bmatrix}v_\mathrm{x}\\ v_\mathrm{y}\end{bmatrix} \tag{2}$$

and

$$J_\mathrm{zz}\ddot{\psi}_\mathrm{z}=\sum_{i=1}^{4}\mathbf{r}_i\mathbf{F}_i, \tag{3}$$

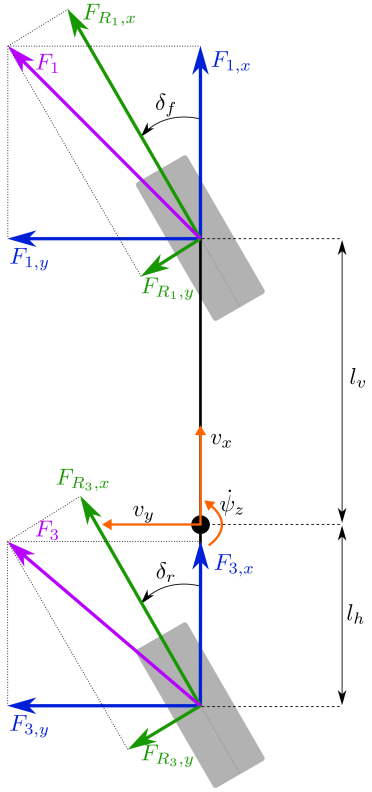

Fig. 1. The single-track model. Forces F_{R_2} and F_{R_4} are not depicted in the figure because in a general case with symmetric tires $F_{R_2} = F_{R_1}$ and $F_{R_4} = F_{R_3}$.

where

$$\mathbf{r} = [\mathbf{r_1}\ \mathbf{r_2}\ \mathbf{r_3}\ \mathbf{r_4}] = \left[\begin{bmatrix} l_\mathrm{v} \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} l_\mathrm{v} \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} -l_\mathrm{h} \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} -l_\mathrm{h} \\ 0 \\ 0 \end{bmatrix}\right] \tag{4}$$

is the vector describing position of each wheel with respect to the center of gravity and $\mathbf{F}_i = \begin{bmatrix} F_{i,x} \\ F_{i,y} \end{bmatrix}$ is a vector of forces acting on i^{th} wheel. The vector and its elements are depicted in Fig. 1. Note that although Fig. 1 might suggest that the model has 2 wheels, it is defined with 4 wheels, where the left and right wheels are in the same place. This allows for usage of asymmetrical tire models (such as the one used in this paper). The parameters l_v and l_h are distances of wheels from CG, as depicted in Fig. 1. The wheels are numbered in this order: front-left, front-right, rear-left, rear-right. m_v is the vehicle mass, $\mathbf{F}_{i,\mathrm{x/y}}$ is a force acting on i-th wheel along x/y axis in body-fixed coordinates. $F_{R_i,\mathrm{x}}$ is a force acting along x axis in wheel coordinate system. The term $-\frac{1}{2}c_\mathrm{w}\rho A_\mathrm{w}\sqrt{v_\mathrm{x}^2 + v_\mathrm{y}^2}\begin{bmatrix} v_\mathrm{x} \\ v_\mathrm{y} \end{bmatrix}$ is an approximation of air-resistance, c_w is a drag coefficient, ρ is air density and A_w is the total surface exposed to the air flow. J_{zz} is the vehicle inertia about z-axis and J_{R_i} is the wheel inertia about y-axis.

The forces $\begin{bmatrix} F_{R_i,\mathrm{x}} \\ F_{R_i,\mathrm{y}} \end{bmatrix}$ are calculated using the "Pacejka magic formula" Pacejka (2012)

$$F = D\cos(C\arctan(Bx - E(Bx - \arctan(Bx)))). \tag{5}$$

The same formula can be used for calculating $F_{R_i,\mathrm{x}}$ (tire longitudinal force) and $F_{R_i,\mathrm{y}}$ (tire lateral force) with a different set of parameters for each. The argument x can be either sideslip angle α or longitudinal slip ratio κ (usually denoted as λ which is used for eigenvalue in this paper) (see Pacejka (2012)) for calculating $F_{R_i,\mathrm{y}}$ or F_x respectively. The parameters B, C, D and E are generally time-dependent. This work uses the Pacejka tire model Pacejka (2012) 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

$$\begin{bmatrix} F_{i,\mathrm{x}} \\ F_{i,\mathrm{y}} \end{bmatrix} = \begin{bmatrix} \cos(\delta_i) & -\sin(\delta_i) \\ \sin(\delta_i) & \cos(\delta_i) \end{bmatrix} \begin{bmatrix} F_{R_i,\mathrm{x}} \\ F_{R_i,\mathrm{y}} \end{bmatrix}. \tag{6}$$

3. LINEAR PREDICTORS

Linear predictor is a linear model of a controlled system that is able to provide the prediction of the future behaviour of the controlled system with sufficient accuracy. The predictor used in this paper is the Koopman operator and it will be used as a control design model for MPC. The Koopman operator is infinite-dimensional linear system, which is able to describe the nonlinear behaviour of the controlled system. A finite-dimensional approximation of the Koopman operator will be used as a control design model for a linear MPC resulting in a control law that is linear in the state space of the Koopman operator, but nonlinear in the original state space of the nonlinear controlled system.

3.1 Koopman operator

The Koopman operator is used as a linear predictor of the nonlinear dynamics of the system Sec. 2. The basic idea consists in transforming (*lifting*) the nonlinear state space to a new high-dimensional, linearly evolving state space. The control design is then performed in the linear state space using linear control methodology. Let us assume a discrete nonlinear uncontrolled system with state x_k at time step k, dynamics $f_\mathrm{u}(.)$, output y_k and output equation $h(x_k)$:

$$\begin{aligned} x_{k+1} &= f_\mathrm{u}(x_k) \\ y_k &= h(x_k). \end{aligned} \tag{7}$$

The Koopman operator $\mathcal{K} : \mathcal{C}(\mathbb{R}^n) \to \mathcal{C}(\mathbb{R}^n)$, with $\mathcal{C}(\mathbb{R}^n)$ denoting a space of continuous functions defined on $\mathbb{R}^n$, is defined as

$$(\mathcal{K}\phi)(x_k) = \phi(f_\mathrm{u}(x_k)) \tag{8}$$

for each basis function $\phi : \mathbb{R}^n \to \mathbb{R}$ where n is size of the state vector x_k. In our case, the function ϕ will also be an *eigenfunction* of the operator $\mathcal{K}$, meaning that the following holds:

$$\phi(x_{k+1}) = \lambda\phi(x_k), \tag{9}$$

for some eigenvalue $\lambda \in \mathcal{R}$. The functions ϕ will be constructed from trajectories of (7) according to

$$\phi(x_k^j) = \phi(x_k^j)_{\lambda,g} = \lambda^k g_\phi(x_0^j), \tag{10}$$

where j is a trajectory of (7) starting in x_0^j and x_k^j is the point to which the system will get after k time-steps. The state vector x_k^j is transformed with a function $\phi(x_k^j)$, defined according to (10) for an arbitrary eigenvalue λ and an arbitrary function $g_\phi : \mathbb{R}^n \to \mathbb{R}$.

Note that the definition from (10) fulfills the requirement of (9) because

$$\phi(x_{k+1}^j) = \lambda^{k+1} g_\phi(x_0^j) = \lambda \cdot \lambda^k g_\phi(x_0^j) = \lambda \phi(x_k^j). \quad (11)$$

In other words, $\phi(x_k^j)$ evolves linearly along trajectories of the system (7). The trajectories must fulfill certain assumptions in order for the definition (10) to be valid. The assumptions are beyond the scope of this paper and are discussed in Korda and Mezić (2019).

3.2 Uncontrolled case

The functions $g_\phi(.)$ can be replaced with scalars because they are evaluated only at the starting points x_0^j of the trajectories j. Let us denote the set of starting points x_0^j as Γ. The evaluation of $g_\phi(.)$ on a point from Γ will be denoted as

$$g_{p,i}^j = g_\phi(x_0^j), \text{ for } x_0^j \in \Gamma, \quad (12)$$

where p denotes the number of output ($p = 1, 2, ...N_y$) with N_y being the total number of outputs and i is the associated eigenvalue. The association of $g_{p,i}^j$ with a specific eigenvalue and a specific output allows for a trivial derivation of the A and C matrices, which will be discussed further below. The values $g_{p,i}^j$ can be optimized in a *convex* manner in order to approximate the output values by

$$y_{p,k}^j = \sum_{i=1}^{N_\Lambda} \lambda_i^k g_{p,i}^j, \quad (13)$$

where $y_{p,k}^j$ is the p^{th} output of j^{th} trajectory at time-step k and N_Λ is the number of eigenvalues. The solution of (13) for output p can be written in matrix form as

$$||Lg_p - F_p||_2^2 + \zeta ||g_p||_2^2, \quad (14)$$

where L is a matrix containing the eigenvalues λ, F_p is a matrix of outputs from all trajectories and ζ is a regularization term. The optimized value is the vector g_p which contains $g_{p,i}^j$ for all λ_i and all trajectories.

The concrete form of the matrices in (14) can be found in Korda and Mezić (2019), as well as the algorithm for finding the eigenvalues λ_i.

In order to obtain the matrices A and C consider the eigenfunction definition from (9), for

$$\boldsymbol{\phi}(x_k^j) = \left[\phi_1(x_k^j)\ \phi_2(x_k^j)\ \dots \phi_{N_\phi}(x_k^j)\right]^\top \quad (15)$$

the dynamics

$$z_{k+1} = Az_k \quad (16)$$

can then be written as

$$\begin{bmatrix} \phi_1(x_{k+1}^j) \\ \phi_2(x_{k+1}^j) \\ \vdots \\ \phi_{N_\phi}(x_{k+1}^j) \end{bmatrix} = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_{N_\phi} \end{bmatrix} \begin{bmatrix} \phi_1(x_k^j) \\ \phi_2(x_k^j) \\ \vdots \\ \phi_{N_\phi}(x_k^j) \end{bmatrix}. \quad (17)$$

Choosing (9) as basis functions immediately yields the diagonal A matrix. The output matrix C is also trivial thanks to (13).

$$C = \begin{bmatrix} 1 \dots 1 & & \\ & 1 \dots 1 & \\ & & 1 \dots 1 \end{bmatrix}_{N_y \times (N_y \cdot N_\Lambda)}. \quad (18)$$

Note that in this case, the Koopman operator defined in (8) is implemented as the state matrix A.

3.3 Controlled case

In this work however, a controlled scenario will be considered. The discrete *nonlinear* controlled system with the input u_k

$$\begin{aligned} x_{k+1} &= f(x_k, u_k) \\ y_k &= g(x_k) \end{aligned} \quad (19)$$

will be approximated by a *linear* system

$$\begin{aligned} z_{k+1} &= Az_k + Bu_k \\ y_k &= Cz_k \end{aligned} \quad \text{for } z_0 = \phi(x_0), \quad (20)$$

where z_k is a *lifted* state vector at time-step k. The nonlinear state vector x_k will be considered as the output y_k, so $y_k := x_k$. The relationship between the two systems is shown in Fig. 2.

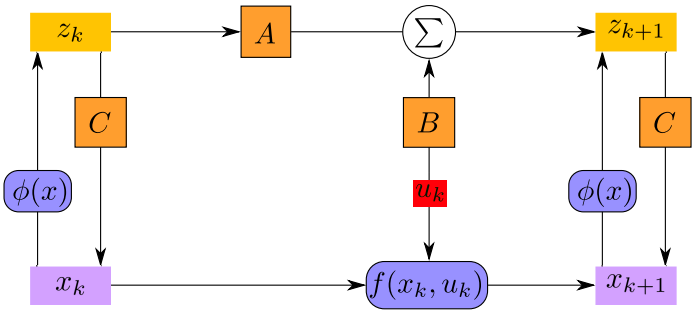

Fig. 2. Discrete-time scheme showing the relationship of a nonlinear system and its linear approximation.

Having the matrices A and C, the matrix B can be optimized over the whole trajectory, allowing for multiple-step prediction. The optimization problem can be formulated as

$$\min \sum_{j=1}^{N_T} \sum_{k=1}^{K} ||g(x_k^j) - \hat{y}_k(x_0^j)||_2^2, \quad (21)$$

where N_T is the number of trajectories, K is the number of samples in each trajectory and

$$\hat{y}_k(x_0^j) = CA^k z_0^j + \sum_{i=0}^{k-1} CA^{k-i-1} Bu_i^j, \quad \text{for } z_0^j = \phi(x_0^j) \quad (22)$$

is a prediction of the output vector by the matrices A, B and C.

For optimization over shorter window instead of the whole trajectory, see Cibulka (2019).

The problem (21) can be also solved as a least-squares problem, see Korda and Mezić (2019) for further details.

3.4 Algorithm summary

The uncontrolled dynamics is identified first according to Sec. 3.2 using an uncontrolled dataset. Then the control is added via the B matrix, using the approach described in Sec. 3.3 with a controlled dataset. This results in a system

$$\begin{aligned} z_{k+1} &= Az_k + Bu_k \\ y_k &= Cz_k \end{aligned} \tag{23}$$

which will be used for linear MPC design.

4. IDENTIFICATION RESULTS

4.1 Uncontrolled

The model described in Sec. 2 was discretized with time-step $\mathrm{T_s} = 0.01$s and approximated by a Koopman operator using the following parameters: $N_\Lambda = 51$, $N_\mathrm{T} = 1078$ and $\zeta = 10^{-12}$. The values of the parameters were adopted from Cibulka (2019), they were chosen to provide a sufficient prediction accuracy while keeping computer resource usage at manageable levels. The starting points for the trajectories were selected from a set with constant kinetic energy $E_\mathrm{k} = 500\,\mathrm{kJ}$, an equivalent of a 1300 kg car driving straight at $100\,\mathrm{km\,h^{-1}}$. The Γ set can be seen in Fig. 3. Areas with large $|v_\mathrm{y}|$ and low $|v_\mathrm{x}|$ (car sliding sideways) contain more points because the vehicle leaves this area of state-space rather quickly, resulting in sparse data coverage. See Cibulka et al. (2019) for more details. Results of the uncontrolled dynamics identification can be seen in Fig. 4 and Fig. 5. The initial points used for evaluation were randomly generated inside the Γ surface depicted in Fig. 3 and the length of the trajectories used for uncontrolled identification was 0.5 s, which was the time after which the vehicle model managed to stabilize itself. Note that this time is rather short because the model defined in Sec. 2 uses longitudinal slip ratios as inputs and they were set to 0 during the uncontrolled identification. This allowed the tire to generate maximum force in the y direction which resulted in such short times. Please see Pacejka (2012) for more information on the tire model. The starting points with $||x_0||_2^2 < 8.3$ ($8.3\,\mathrm{m\,s^{-1}} \doteq 30\,\mathrm{km\,h^{-1}}$) were rejected from the testing dataset because the tire model Pacejka (2012) is ill-defined at low speeds.

4.2 Controlled case

Controlled trajectories were generated with randomly generated inputs drawn from a uniform distribution, where $\lambda_\mathrm{r} \in [-1, 1]$ and $\delta_\mathrm{f} \in [-30°, 30°]$. The control horizon for the MPC was chosen as 0.1 s (adopted from Cibulka (2019)) so the matrix B was optimized on 0.1 s long trajectories. The mean RMSE was 4% (the uncontrolled RMSE was 2.3%). The distribution of the error can be seen in Fig. 7.

5. MPC

The identified system described in Sec. 4 was used for MPC design. The MPC based on the Koopman operator will be called Koopman MPC (term first used in Korda and Mezić (2018)). The Koopman MPC framework is depicted in Fig. 6. The Koopman MPC will be compared with

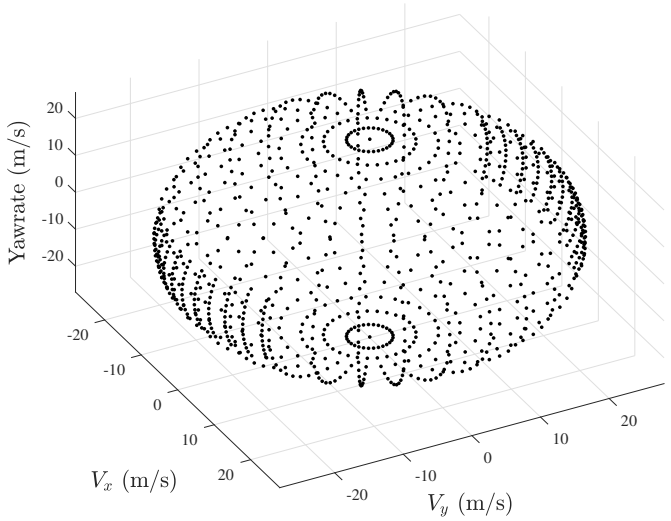

Fig. 3. The set of initial conditions for the trajectories used for identification. The points from this set have a constant kinetic energy $E_\mathrm{k} = 500\,\mathrm{kJ}$.

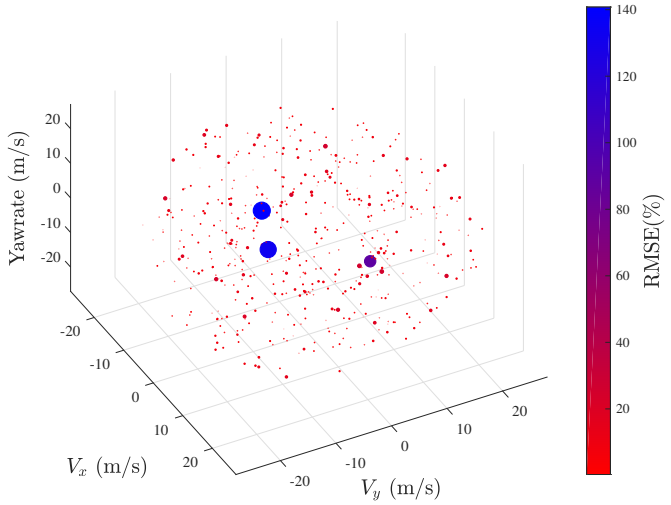

Fig. 4. Errors of the Koopman operator in the uncontrolled case. Each point in the figure corresponds to an initial condition of a 0.5 s long trajectory. The size and color of the points correspond to the prediction error of the associated trajectory. The mean RMSE is 6% with a standard deviation of 10.2%

MPC based on a locally linearized model, which will be called Linear MPC. Both MPC regulators are defined as a quadratic optimization problem

$$\min_{u_m} \sum_{m=0}^{N} [(y_m - r_m)^\top Q_\mathrm{y} (y_m - r_m) + u_m^\top R u_m + s_m^\top S s_m]$$

s.t.

$$\begin{aligned} z_{m+1} &= Az_m + Bu_m && m = 0..N-1 \\ y_m &= Cz_m && m = 0..N-1 \\ y_\mathrm{min} - s_m &\le y_m \le y_\mathrm{max} + s_m && m = 0..N-1 \\ u_\mathrm{minrate} &\le u_{m+1} - u_m \le u_\mathrm{maxrate} && m = 0..N-1 \\ u_\mathrm{min} &\le u_m \le u_\mathrm{max} && m = 0..N-1, \end{aligned} \tag{24}$$

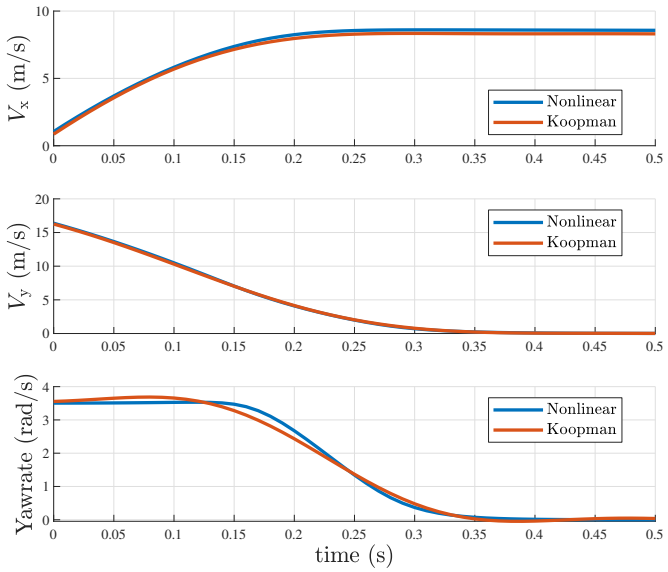

Fig. 5. A comparison of the nonlinear and linear system on a trajectory with RMSE = 6% which is equal to the mean RMSE of the whole dataset.

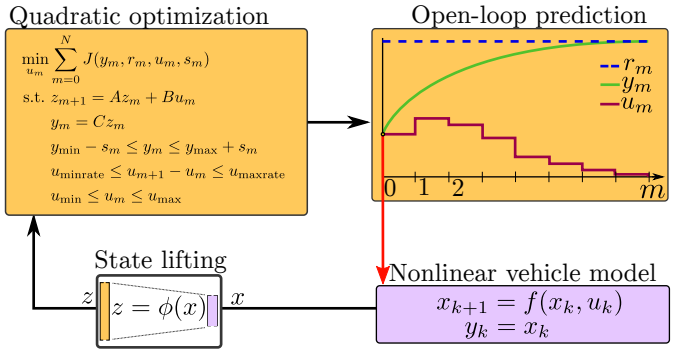

Fig. 6. Scheme describing the Koopman MPC algorithm. Areas operating in the lifted state-space are depicted in orange color, The non-linear space is depicted in violet.

where Q_y,S and R are positive semidefinite cost matrices, N is the prediction horizon, $y_{\min/\max}$ are soft constraints on the output vector y_k with slack variables s and $u_{\text{minrate/maxrate}}$ are constraints on the system input rates. The only difference between Koopman MPC and Linear MPC are the state matrices A, B and C. Both MPC regulators were parametrized as follows:

$$Q_y = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix}, R = \begin{bmatrix} 0 & & \\ & 100 & \\ & & 30 \end{bmatrix}, S = 10^5 \cdot \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix} \tag{25}$$

$$y_{\min} = -\begin{bmatrix} 25 \\ 2 \\ 2 \end{bmatrix}, y_{\max} = \begin{bmatrix} 25 \\ 2 \\ 2 \end{bmatrix}, \tag{26}$$

$$u_{\min} = -\begin{bmatrix} 0 \\ 1 \\ 0.45 \\ 0 \end{bmatrix}, u_{\max} = \begin{bmatrix} 0 \\ 1 \\ 0.45 \\ 0 \end{bmatrix}, \tag{27}$$

$$u_{\text{minrate}} = -\begin{bmatrix} 0 \\ 0.1 \\ 0.8 \\ 0 \end{bmatrix}, u_{\text{maxrate}} = \begin{bmatrix} 0 \\ 0.1 \\ 0.8 \\ 0 \end{bmatrix}. \tag{28}$$

The scheme of the Koopman MPC is depicted in Fig. 6. The implementation of (24) was done in YALMIP Löfberg (2019).

6. RESULTS

It can be seen in Fig. 8 that the Koopman-controlled vehicle was able to recover from a state where the vehicle drifts sideways in one continuous motion, unlike the MPC based on local linearization. Notice how each algorithm steered the vehicle in a different direction. The Koopman MPC steered left in order to shift the momentum from $\mathrm{y-axis}$ to $\mathrm{x-axis}$ while the locally linearized MPC steered to the right because it was trimmed in state $\mathbf{x}_0 = [16.7\ 0\ 0]^\top$. Steering to the right decreases v_y (or increases it in negative direction) in this state.

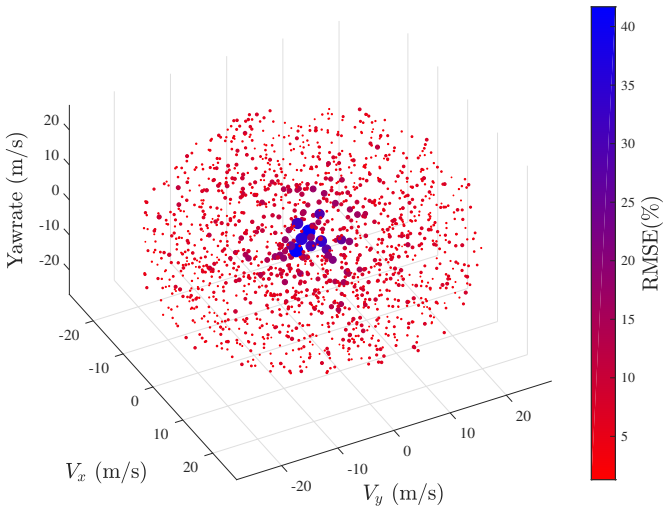

Fig. 7. Errors of the Koopman operator in the controlled case. Each point in the figure corresponds to an initial condition of a 0.1 s long trajectory with random control inputs. The size and color of the points correspond to the prediction error of the associated trajectory. The mean RMSE is 4% with a standard deviation of 2.7%

Unfortunately, the Koopman MPC does not always outperform the local linearization. See Fig. 9 for example. In this case, the goal was to steadily increase yawrate while keeping v_x stable. in other words, the vehicle should be driving in an increasingly tighter spiral while keeping its forward velocity v_x the same.

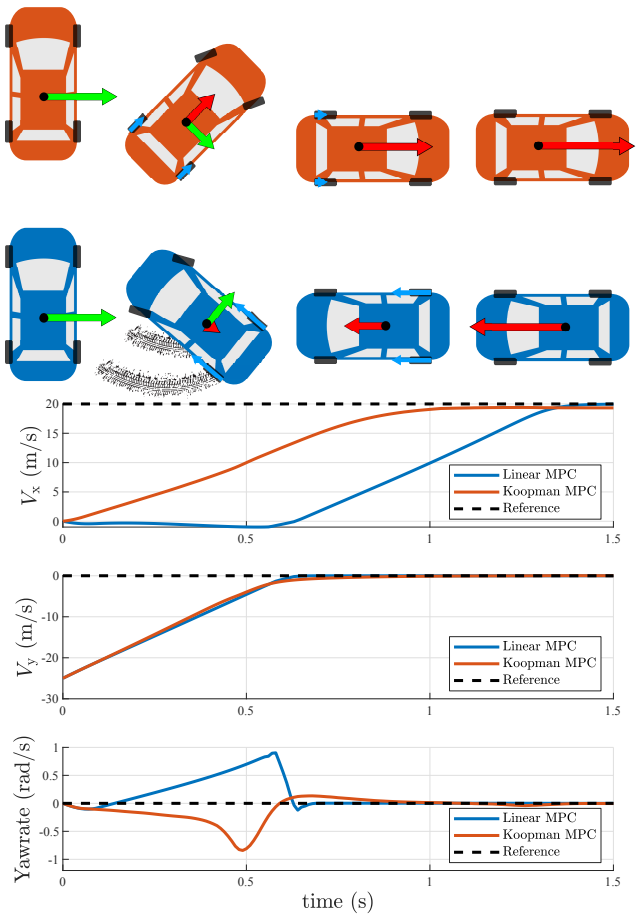

Fig. 8. Comparison of maneuvers for recovery from unusual vehicle motion. The Koopman MPC stabilized the vehicle faster in one continuous motion. The Linear MPC brought the vehicle to full stop and then accelerated to reach the desired velocity. Notice how both algorithms steered the vehicle in different directions.

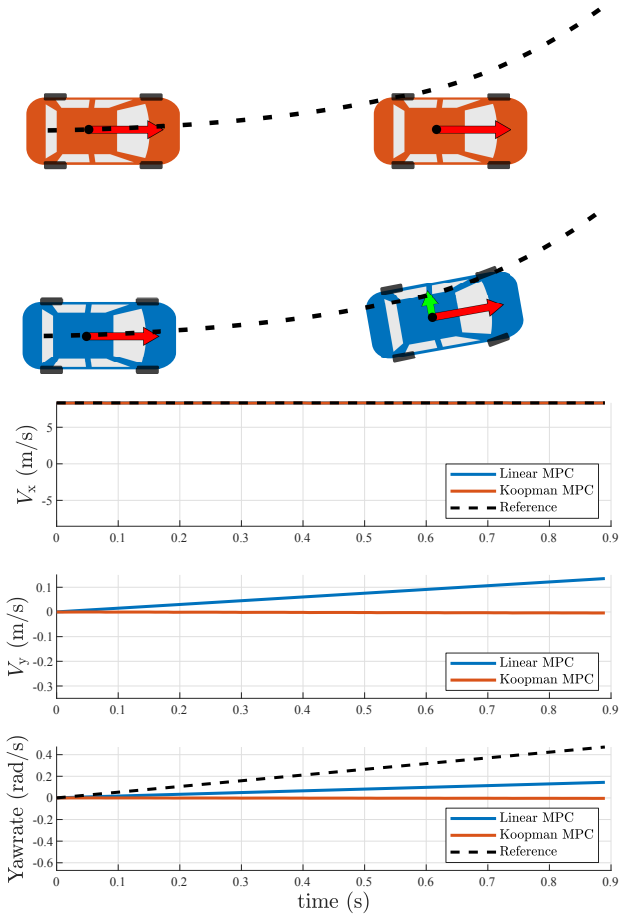

Fig. 9. The reference here is increasing yawrate and constant forward velocity v_x. The state v_y is without any reference. The Koopman MPC is unable to track the reference and the Linear MPC is performing much better in this case, although it still isn't able to track the reference signal.

7. CONCLUSION

The Koopman MPC showed very promising results by stabilizing a vehicle from a 90-degree drift while also preserving energy by shifting the vehicle's already present sideways momentum into a forward momentum. This result is in stark contrast with the fact the same controller was unable to perform rather simple steering maneuver.The reason behind this behaviour will be examined in our future work.

ACKNOWLEDGEMENTS

This research was supported by the Czech Science Foundation (GACR) under contracts No. 19-16772S, 19-18424S, 20-11626Y, and by the Grant Agency of the Czech Technical University in Prague,
grant No. SGS19/174/OHK3/3T/13.

REFERENCES

Cibulka, V., Hanis, T., and Hromcik, M. (2019). Data-driven identification of vehicle dynamics using Koopman operator. URL `http://arxiv.org/abs/1903.06103v1`.

Cibulka, V. (2019). *MPC Based Control Algorithms for Vehicle Control*. Master's thesis, CTU in Prague.

Koopman, B.O. (1931). Hamiltonian Systems and Transformation in Hilbert Space. *Proceedings of the National Academy of Sciences*, 17(5), 315–318. doi:10.1073/pnas.17.5.315.

Korda, M. and Mezić, I. (2018). Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. *Automatica*, 93, 149–160. doi:10.1016/j.automatica.2018.03.046.

Korda, M. and Mezić, I. (2019). Optimal construction of Koopman eigenfunctions for prediction and control. URL `http://arxiv.org/abs/1810.08733v2`.

Löfberg, J. (2019). YALMIP. https://yalmip.github.io/.

Mezić, I. (2005). Spectral Properties of Dynamical Systems, Model Reduction and Decompositions. *Nonlinear Dynamics*, 41(1-3), 309–325. doi:10.1007/s11071-005-2824-x.

Mezić, I. and Banaszuk, A. (2004). Comparison of systems with complex behavior. *Physica D: Nonlinear Phenomena*, 197(1-2), 101–133. doi:10.1016/j.physd.2004.06.015.

Pacejka, H. (2012). *Tire and Vehicle Dynamics*. Elsevier LTD, Oxford. URL `https://www.ebook.de/de/product/18341528/hans_pacejka_tire_and_vehicle_dynamics.html`.