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

Data-Driven Design of Model Predictive Control for Powertrain-Aware Eco-Driving Considering Nonlinearities Using Koopman Analysis ⋆

Daliang Shen * **Jihun Han** * **Dominik Karbowski** *
Aymeric Rousseau *

* *Argonne National Laboratory, Vehicle and Mobility Systems Group, Lemont, IL 60439 USA (e-mail: {dshen; jihun.han; dkarbowski; arousseau}@anl.gov).*

Abstract: Eco-driving is a highly nonlinear control problem. The nonlinearities include the complex energy conversion/dissipation in the powertrain, environmental influences such as road grade and aerodynamic drag, constraints due to traffic signs, safety issues, and physical limits of the vehicle system. In recent years, researchers have increasingly revisited the Koopman operator to linearize nonlinear dynamics. This paper adopts such an approximation technique to construct the lifted state space in a data-driven procedure that allows us to incorporate nonlinearities and system perturbations in the cost function. In addition, the nonlinear constraints in states can also be handled linearly. The resultant formulation of a linearly constrained quadratic program can be readily applied to design a model predictive control that enjoys a low computation load as with a linear dynamic system. Simulation results demonstrate additional energy saving potential compared to a linear approach.

Keywords: Connected and automated vehicles, Nonlinear model reduction, Model predictive control, Koopman operator, Legendre polynomial approximation

1. INTRODUCTION

In recent years, automation control society revisits Koopman operator theory (Koopman, 1931), to obtain linearized formulations for system modeling and thereby design linear controls for their applications. The main idea is to represent a *nonlinear* dynamical system with an infinite-dimensional but *linear* operator on a Hilbert space of measurement functions of the original system states. In practice, an approximation of the operator acting on a high- but finite-dimensional subspace, is usually applied instead. In contrast to commonly adopted *local* linearization near a fixed point, such as using Taylor series, Koopman approximation offers a *global* linear representation. Recently, researchers have demonstrated that the approximation can also be extended to a *controlled* context, which makes it possible to design controls using linear control theories (Korda and Mezić, 2018; Kaiser et al., 2021).

Another factor contributing to the renewed interest in Koopman theory is that the procedure to obtain the linear model can be fully data-driven and only requires a linear least-squares fitting. Moreover, the computational complexity of the linear control designed through Koopman approximation depends on the number of control inputs and thus is comparable to conventional linear control.

This paper concerns the eco-driving problem of connected and automated vehicles, which is to minimize the energy consumption in traveling a given route. As the authors noted in Shen et al. (2021), the system dynamics, the cost function and path constraints of eco-driving contain various nonlinearities. Naturally, these nonlinearities can be well accommodated with solution methods such as dynamic programming (De Nunzio et al., 2013) and nonlinear programming (Xu et al., 2015; Shen et al., 2021). Unfortunately, the combinatorial complexity involved in evaluating the cost function (also known as the curse of dimensionality) of dynamic programming and the iterative corrections in solution convergence of nonlinear programming render them cumbersome to implement in real-time. By changing a coordinate system (e.g., integrating over speed [Wan et al., 2016] or distance [Kalabis and Müller, 2012] in lieu of time), nonlinearities may be eliminated in the selected state, but not in both states simultaneously.

In this paper, we introduce a data-driven procedure for system identification based on Koopman theory to construct the vehicle longitudinal motion in a high-dimensional *lifted* state-space. Thus, the nonlinear influence of powertrain-dependent energy inefficiency, friction losses, road grade, and speed limits on the cost function and inequality constraints can be embedded in that *lifted* linearly evolving invariant subspace. In Section 3.2, the **highlight** of the paper, the Koopman embedding

⋆ This report and the work described were sponsored by the U.S. Department of Energy (DOE) Vehicle Technologies Office (VTO) under the Systems and Modeling for Accelerated Research in Transportation (SMART) Mobility Laboratory Consortium, an initiative of the Energy Efficient Mobility Systems (EEMS) Program.

allows us to adapt the polynomial factors in the cost function and linear inequality conditions to the position-dependent changes in road grade and speed limit. The linear-quadratic formulation of the system and cost function can be further converted to a constrained quadratic program (QP) that can be efficiently solved. In the end of the paper, we evaluate the performance of the resultant linear model predictive control (MPC) in `RoadRunner`, [1] and compare it with an established linear controller (Han et al., 2020a).

2. DATA-DRIVEN KOOPMAN ANALYSIS

2.1 Koopman Operator

We start with the general form of a nonlinear controlled system in discrete time:

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k)$$

where k denotes an integer time step, $k = 0, 1, \dots$. $\boldsymbol{x} \in \mathbb{R}^n$ is the state of the system, $\boldsymbol{u} \in \mathbb{R}^m$ the control input, and $\boldsymbol{f}$ the nonlinear transition mapping.

Now we introduce the Koopman operator $\mathcal{K} : \mathcal{F} \mapsto \mathcal{F}$, which is a linear operator acting on *nonlinear* observable functions $\psi : \mathcal{F} \mapsto \mathbb{R}$ belonging to a space of invariant functions $\mathcal{F}$. Originally, the Koopman operator was introduced for uncontrolled dynamical systems and is infinite-dimensional. Papers such as Proctor et al. (2018) show that the Koopman operator can be generalized for *controlled* systems to evolve over a state space augmented by a control space,

$$(\mathcal{K}\psi)(\boldsymbol{x}_k, \boldsymbol{u}_k) = \psi(\boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k), \boldsymbol{u}_{k+1}).$$

2.2 Linear Lifted System and Quadratic Stage Cost

To design a time-domain control, we need to approximate the Koopman operator $\mathcal{K}$ in a finite-dimensional subspace $\mathbb{R}^N$, $N \gg n$. To achieve the approximation, we applied a data-driven method called extended dynamic mode decomposition (Korda and Mezić, 2018) that performs regression on a vector of observable functions $\boldsymbol{\psi}$, which produces the lifted state $\boldsymbol{z} \in \mathbb{R}^N$,

$$\boldsymbol{z} = \boldsymbol{\psi}(\boldsymbol{x}) = [\psi_1(\boldsymbol{x}), \dots, \psi_N(\boldsymbol{x})]^\mathsf{T}. \tag{1}$$

Note that here control $\boldsymbol{u}$ is not lifted. The extended dynamic mode decomposition algorithm seeks to identify the best-fit linear operators $A \in \mathbb{R}^{N\times N}$, $B \in \mathbb{R}^{N\times m}$ and $C \in \mathbb{R}^{n\times N}$ in the lifted space,

$$\boldsymbol{z}_{k+1} = A\boldsymbol{z}_k + B\boldsymbol{u}_k, \quad \hat{\boldsymbol{x}}_k = C\boldsymbol{z}_k, \tag{2}$$

where $\hat{\boldsymbol{x}}$ is the predicted state. In addition to the linear lifted system model (2), for the ultimate goal of designing an MPC, the stage cost needs to be defined in a quadratic form,

$$\begin{aligned} L_k &= \boldsymbol{z}_k^\mathsf{T} Q \boldsymbol{z}_k + \boldsymbol{u}_k^\mathsf{T} R \boldsymbol{u}_k + \boldsymbol{z}_k^\mathsf{T} P \boldsymbol{u}_k + q^\mathsf{T} \boldsymbol{z}_k + r^\mathsf{T} \boldsymbol{u}_k + c \\ &= \begin{bmatrix} 1 & \boldsymbol{z}_k^\mathsf{T} & \boldsymbol{u}_k^\mathsf{T} \end{bmatrix} \underbrace{\begin{bmatrix} c & \frac{1}{2}\boldsymbol{q}^\mathsf{T} & \frac{1}{2}\boldsymbol{r}^\mathsf{T} \\ \frac{1}{2}\boldsymbol{q} & Q & \frac{1}{2}P^\mathsf{T} \\ \frac{1}{2}\boldsymbol{r} & \frac{1}{2}P & R \end{bmatrix}}_{\triangleq \Omega} \begin{bmatrix} 1 \\ \boldsymbol{z}_k \\ \boldsymbol{u}_k \end{bmatrix}, \end{aligned} \tag{3}$$

where $Q \in \mathbb{R}^{N\times N}$, $R \in \mathbb{R}^{m\times m}$, $P \in \mathbb{R}^{N\times m}$, $\boldsymbol{q} \in \mathbb{R}^N$, $\boldsymbol{r} \in \mathbb{R}^m$, and $c \in \mathbb{R}$ are weighting matrices. Q and R are positive semi-definite. Ω is a symmetric matrix composed of Q, R, P, $\boldsymbol{q}$, $\boldsymbol{r}$, and c, as shown. To prepare for linear regression finding the quadratic-cost factors, we consider a vector $\boldsymbol{y}$ built by column-major vectorizing [denoted by $\mathrm{vec}(\cdot)$] the outer product of $\boldsymbol{\zeta}$ and itself,

$$\boldsymbol{y} = \mathrm{vec}\big(\boldsymbol{\zeta}\boldsymbol{\zeta}^\mathsf{T}\big), \quad \boldsymbol{\zeta} \triangleq \begin{bmatrix} 1 & \boldsymbol{z}^\mathsf{T} & \boldsymbol{u}^\mathsf{T} \end{bmatrix}^\mathsf{T}. \tag{4}$$

Thus, quadratic form (3) is equivalent to

$$L_k = \big(\mathrm{vec}(\Omega)\big)^\mathsf{T} \boldsymbol{y}_k. \tag{5}$$

2.3 Data-Driven Learning Procedure

The first step is to collect the direct measurement data (generated, in our case, by simulation) of the state, the control input, and stage cost in the snapshot matrices:

$$\begin{aligned} \mathbf{X} &= [\boldsymbol{x}_1, \dots, \boldsymbol{x}_K], & \mathbf{X}^+ &= [\boldsymbol{x}_2, \dots, \boldsymbol{x}_{K+1}], \\ \mathbf{U} &= [\boldsymbol{u}_1, \dots, \boldsymbol{u}_K], & \mathbf{L} &= [L_1, \dots, L_K], \end{aligned}$$

where $\mathbf{X} \in \mathbb{R}^{n\times K}$ and $\mathbf{X}^+ \in \mathbb{R}^{n\times K}$ refer to the current and the successor states, respectively, resulting from the control inputs $\mathbf{U} \in \mathbb{R}^{m\times K}$. $\mathbf{L} \in \mathbb{R}^{1\times K}$ collects the stage costs. K columns correspond to K time steps. Then we construct the *lifted* state matrices by evaluating (1),

$$\mathbf{Z} = [\boldsymbol{\psi}(\boldsymbol{x}_1), \dots, \boldsymbol{\psi}(\boldsymbol{x}_K)], \ \mathbf{Z}^+ = [\boldsymbol{\psi}(\boldsymbol{x}_2), \dots, \boldsymbol{\psi}(\boldsymbol{x}_{K+1})],$$

and a snapshot matrix collecting $\boldsymbol{y}$ applying (4) at K time steps,

$$\mathbf{Y} = [\boldsymbol{y}_1, \dots, \boldsymbol{y}_K].$$

The last step is to perform the least-squares minimization,

$$\min_{A,\,B} \big\|\mathbf{Z}^+ - A\mathbf{Z} - B\mathbf{U}\big\|_F, \quad \min_{C} \|\mathbf{X} - C\mathbf{Z}\|_F, \tag{6a}$$

$$\min_{\Omega} \big\|\mathbf{L} - \big(\mathrm{vec}(\Omega)\big)^\mathsf{T}\mathbf{Y}\big\|_F \tag{6b}$$

with $\|\cdot\|_F$ being the Frobenius norm of a matrix. In this way, the linear operators A, B, and C are obtained as the solutions to (6a),

$$[A,\, B] = \mathbf{Z}^+[\mathbf{Z}, \mathbf{U}]^\dagger, \quad C = \mathbf{X}\mathbf{Z}^\dagger, \tag{7a}$$

and the quadratic operator Ω as solution to (6b),

$$\Omega = \mathrm{vec}^{-1}\big(\mathbf{L}\mathbf{Y}^\dagger\big). \tag{7b}$$

Here, $\mathrm{vec}^{-1}(\cdot)$ inverses the vectorization and yields the square matrix Ω as in (3).

3. APPLYING DATA-DRIVEN LEARNING TO ECO-DRIVING

The eco-driving problem discussed in this paper concerns a *single* vehicle and is to minimize energy consumption $\mathcal{J}$ for the travel duration $t \in [t_0, t_\mathrm{f}]$:

$$\min_a \ \mathcal{J} = \int_{t_0}^{t_\mathrm{f}} P_\mathrm{cons}(a, v, s)\,\mathrm{d}t \tag{8a}$$

$$\text{subject to} \qquad \dot{v} = a, \quad \dot{s} = v, \tag{8b}$$

$$\boldsymbol{h}(v, s, a, t) \le \boldsymbol{0}, \tag{8c}$$

$$v(t_0) = v_0, \ \ s(t_0) = s_0, \ \ s(t_\mathrm{f}) = s_\mathrm{f}. \tag{8d}$$

The minimization of $\mathcal{J}$ is subject to the longitudinal kinematics (8b), the inequality conditions (8c), [2] and the

[1] `RoadRunner` is a multi-vehicle energy consumption and performance simulation platform that emulates the interactions between vehicles and road infrastructure (Kim et al., 2018).

[2] Ref. Shen et al. (2021), eq. (6) for details of the nonlinearities in the constraints.

boundary value conditions of initial speed/position v_0, s_0 and terminal position s_f (8d). The state of dimension $n = 2$ is composed of the vehicle speed v and position s, and the single-element control input of dimension $m = 1$ is the vehicle acceleration a. To ease the following data analysis, it is necessary to normalize the state and control, and thus we define,

$$\boldsymbol{x} \triangleq [\tilde{v}, \tilde{s}]^\mathsf{T} = [v/v_\mathrm{max}, s/s_\mathrm{max}]^\mathsf{T}, \quad u \triangleq a/a_\mathrm{n}, \tag{9}$$

where the maximum speed $v_\mathrm{max} = 40\,\mathrm{m/s}$, the maximum preview distance $s_\mathrm{max} = 800\,\mathrm{m}$, and the acceleration normalization $a_\mathrm{n} = 2\,\mathrm{m/s^2}$.

To perform the lifting in (1), we choose an observable vector $\boldsymbol{\psi}$ that contains the monomials and their products of $\tilde{v}$ and $\tilde{s}$, up to the third order, $N_\mathrm{p} = 3$:[3]

$$\boldsymbol{\psi}\big([v, s]^\mathsf{T}\big) = \big[\tilde{v}, \tilde{s}, \tilde{v}^2, \tilde{v}\tilde{s}, \tilde{s}^2, \tilde{v}^3, \tilde{v}^2\tilde{s}, \tilde{v}\tilde{s}^2, \tilde{s}^3\big]^\mathsf{T}, \tag{10}$$

which renders $N = 9$.

The energetic stage cost considered in eco-driving comprises the energy consumed over the descretized time $\Delta t = 0.1\,\mathrm{s}$ for propelling the vehicle:

$$L_k = \int_{t_k}^{t_{k+1}} P_\mathrm{cons}(v, F_\mathrm{whl})\,\mathrm{d}t \approx \Delta t P_\mathrm{cons}(v_k, F_{\mathrm{whl},k}).$$

The power consumption $P_\mathrm{cons}\big(v, F_\mathrm{whl}(a, v, s)\big)$ is in general a nonlinear function of speed v and wheel force F_whl (Sciarretta et al., 2015):

$$F_\mathrm{whl}(a, v, s) = \big(a + g\sin\alpha(s)\big)\, m_\mathrm{veh} + F_\mathrm{res}(v), \tag{11}$$

where gravity acceleration is denoted by g, road grade by α, and vehicle mass by m_veh. The resistance force F_res sums up v-nonlinearities such as rolling friction and aerodynamic drag. Given that P_cons is close to a quadratic form and F_res is often considered quadratic in v, the basis $\boldsymbol{y}$ formed by applying (4) to the chosen observables (10) should be sufficient to accommodate high-order modes of the stage cost L_k.

3.1 Collecting Data

In total, we built 90 routes, each 10 km long, in `RoadRunner` and split them at an 8:2 ratio into a training and a validation set. Meanwhile, we prepared a set of common speed limit values $\{v_\mathrm{lim}\}$ and a uniformly spaced series of segmenting positions $\{s_\mathrm{seg}\}$. On each route, we sampled uniformly n_vlim points from $\{v_\mathrm{lim}\}$ and $\{s_\mathrm{seg}\}$ to define the speed limit profile, and n_stp points from $\{s_\mathrm{seg}\}$ to insert stop signs. In the training phase, **road grades are set to zero**.

An electric vehicle controlled by a human driver model (Han et al., 2020b) was simulated without surrounding traffic in `RoadRunner`. The measurement of $\boldsymbol{x}$, u, and L was sampled at $\Delta t = 0.1\,\mathrm{s}$, and collected in trajectories. These trajectories were then partitioned to the maximum preview distance s_max in line with the normalization (9). Inputting the collected $\mathbf{X}$, $\mathbf{X}^+$, $\mathbf{U}$, and $\mathbf{L}$ to the data-driven training (7) in Section 2.3 led to (A, B, C) for the system, and Ω for the energetic stage cost—the composite matrix of $(Q, R, P, \boldsymbol{q}, \boldsymbol{r}, c)$.

[3] As noted in Williams et al. (2015), a wide range for choice of observables $\boldsymbol{\psi}$ is available.

3.2 Embedding Position-Dependent Variables

Readers may have noticed the original vehicle kinematics are already linear and do not require a "linearization." In fact, unlike choosing an effort quantity such as motor torque in Shen et al. (2021), choosing acceleration as control input eliminates all state-dependent nonlinearities from the system (e.g., terrain variation and aerodynamics); consequently, high-order nonlinear modes are passed on to the cost formulation, as in (11). While the nonlinear modes with reference to $\tilde{v}$ like $F_\mathrm{res}(v)$ are invariant, the $\tilde{s}$-dependent mode $\sin\alpha(s)$ varies with the terrain condition; thus, while the former have been embedded into the observable coordinates $\boldsymbol{\psi}$ during the data-driven learning, the latter is absent.

We introduce shifted Legendre polynomials up to N_p-th order:

$$\boldsymbol{F}(\tilde{s}) = [F_0(\tilde{s}), \ldots, F_{N_\mathrm{p}}(\tilde{s})]^\mathsf{T} = D_F \boldsymbol{\xi}_\mathrm{s} = D_F S_{\xi\zeta}\boldsymbol{\zeta}$$

with $\boldsymbol{\xi}_\mathrm{s} = [1, \tilde{s}, \ldots, \tilde{s}^{N_\mathrm{p}}]^\mathsf{T}$, the lower-triangular matrix

$$D_F = [d_{ij}]_{i,j=0}^{N_\mathrm{p}},\ d_{ij} = \begin{cases} 0, & i < j \\ (-1)^{i+j}\binom{i}{j}\binom{i+j}{j}, & i \geq j \end{cases}$$

and the indexing matrix

$$S_{\xi,\zeta} = [s_{\xi\zeta,ij}]_{i,j=0}^{N_\mathrm{p},(1+N+m)}, \quad s_{\xi\zeta,ij} = \begin{cases} 1, & j = \frac{(i+1)i}{2} \\ 0, & \text{otherwise} \end{cases};$$

an arbitrary function $\omega(\tilde{s})$ in the Lebesgue function space $\mathsf{L}^2([0,1])$ can be approximated in $\boldsymbol{F}(\tilde{s})$ (Quarteroni et al., 2007, Sect. 10.1.2), resulting in a polynomial $p_\mathrm{s}(\tilde{s})$,

$$p_\mathrm{s}(\tilde{s}) = \sum_{i=0}^{N_\mathrm{p}} \tilde{g}_i F_i(\tilde{s}),\ \tilde{g}_i = (2i+1)\int_0^1 \omega(\tilde{s}) F_i(\tilde{s})\,\mathrm{d}\tilde{s}. \tag{13a}$$

Further, $p_\mathrm{s}(\tilde{s})$ can be rewritten in linear algebra,

$$p_\mathrm{s}(\tilde{s}) = \tilde{\boldsymbol{g}}^\mathsf{T}\boldsymbol{F}(\tilde{s}) = \tilde{\boldsymbol{g}}^\mathsf{T} D_F \boldsymbol{\xi}_\mathrm{s} = \tilde{\boldsymbol{g}}^\mathsf{T} D_F S_{\xi\zeta}\boldsymbol{\zeta}. \tag{13b}$$

In the special case that ω is piecewise constant in $\tilde{s} \in [0,1]$, $\tilde{\boldsymbol{g}} \triangleq [\tilde{g}_0, \ldots, \tilde{g}_{N_\mathrm{p}}]^\mathsf{T}$ in (13) is trivial to calculate,

$$\begin{aligned} &\tilde{g}_i = (2i+1)\sum_{j=0}^{N_\mathrm{pwc}-1} \bar{\omega}_j \Big(F_i^{(-1)}(\tilde{s}_{j+1}) - F_i^{(-1)}(\tilde{s}_j)\Big), \\ &\omega(\tilde{s}) = \bar{\omega}_j, \text{ for } \tilde{s} \in [\tilde{s}_j, \tilde{s}_{j+1}) \in [0,1], \end{aligned} \tag{14}$$

where N_pwc is the number of piecewise steps of $\omega(\tilde{s})$, and $F_i^{(-1)}$ is the antiderivative of F_i.

Road Grade The nonlinear $g\sin\alpha(\tilde{s}_k)$ is embedded in the stage cost L_k online adaptively.[4] Notice it is an increment to a in (11), so we can replace u_k with $u_k + \frac{g}{a_\mathrm{n}}\sin\alpha(\tilde{s}_k)$ in (3) and the stage cost with the quadratic matrix Ω learned in Section 3.1 becomes

$$L_k = \big(\boldsymbol{\zeta}_k + \boldsymbol{\sigma}_\mathrm{u}\tfrac{g\sin\alpha(\tilde{s}_k)}{a_\mathrm{n}}\big)^\mathsf{T}\,\Omega\,\big(\boldsymbol{\zeta}_k + \boldsymbol{\sigma}_\mathrm{u}\tfrac{g\sin\alpha(\tilde{s}_k)}{a_\mathrm{n}}\big), \tag{15a}$$

$$\boldsymbol{\sigma}_\mathrm{u} = [\mathbf{0}_{1\times N}, 1]^\mathsf{T}. \tag{15b}$$

The Legendre approximation in (13) and (14) for the piecewise constant grade yields $p_{\mathrm{s,g}}(\tilde{s}) \approx \frac{g}{a_\mathrm{n}}\sin\alpha(\tilde{s})$, as Fig. 1 displays with an example grade profile. Once we apply the approximation $p_{\mathrm{s,g}}(\tilde{s})$ to (15), we obtain:

[4] Note that $\tilde{s}_k$ is linear to $\boldsymbol{u}$ but α is nonlinear in $\tilde{s}$.

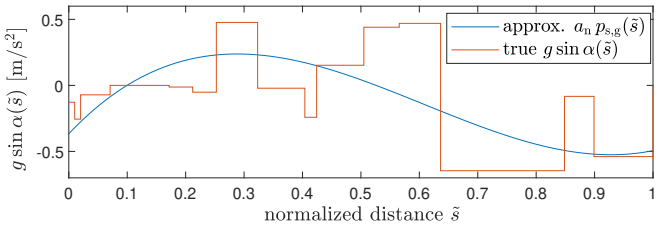

Fig. 1. Fitting road-grade-induced acceleration with Legendre polynomials.

$$L_k \approx \boldsymbol{\zeta}^\mathsf{T}\big(\Omega + \widehat{\Omega}\big)\boldsymbol{\zeta}, \tag{16a}$$

with $$\widehat{\Omega} = \Theta_\mathrm{g}^\mathsf{T}\Omega\Theta_\mathrm{g} + \Omega\Theta_\mathrm{g} + \Theta_\mathrm{g}^\mathsf{T}\Omega \tag{16b}$$

and $$\Theta_\mathrm{g} = \boldsymbol{\sigma}_\mathrm{u}\, \tilde{\boldsymbol{g}}_\mathrm{g}^\mathsf{T} D_F S_{\xi\zeta}, \tag{16c}$$

where $\tilde{\boldsymbol{g}}_\mathrm{g}$ instantiates (14) for the road grade approximation. We see from (16) that the variance in the road grade within a preview horizon is interpreted into a weighting increment $\widehat{\Omega}$ on the high-order coordinates $\boldsymbol{\zeta}$.

Speed Limit The speed limit also varies in the form of a shifted step function depending on position, like the road grade. By applying (13) again, we obtain the Legendre polynomial approximation of the varying speed limit:

$$\tilde{v} \le p_{\mathrm{s,vlim}}(\tilde{s}) = \tilde{\boldsymbol{g}}_\mathrm{vlim}^\mathsf{T} D_F S_{\xi\zeta}\boldsymbol{\zeta}, \tag{17}$$

with $\tilde{\boldsymbol{g}}_\mathrm{vlim}$ resulting from inputting the piecewise speed limit profile to (14). Because $\tilde{v}$ is within the span $\boldsymbol{\zeta}$, the approximated speed limit conforms to a linear inequality expression,

$$\big[\check{\boldsymbol{c}}\ \check{A}\ \check{B}\big]\,\boldsymbol{\zeta} = \big[\check{\boldsymbol{c}}\ \check{A}\ \check{B}\big]\,\big[1\ \boldsymbol{z}^\mathsf{T}\ \boldsymbol{u}^\mathsf{T}\big]^\mathsf{T} \le \mathbf{0},$$

with matrix $\check{A}$ and $\check{B}$ and vector $\check{\boldsymbol{c}}$ forming the linear factor.

Remark 1. The adaptive embedding of road grade and speed limit integrates well into the MPC framework; it requires essentially only re-evaluating $\tilde{\boldsymbol{g}}_\mathrm{g}$ and $\tilde{\boldsymbol{g}}_\mathrm{vlim}$ based on the updated profiles of road grade and speed limit, respectively, which are perceived on the receding horizon of MPC.

3.3 Validation

The learned linear system (2) and quadratic cost (16) was validated on the 18 routes we set aside in Section 3.1. Road grades with their lengths were injected to the routes according to the extracted joint distribution from a collection of real-world routes. [5] The validation was performed in a receding horizon of 20 s that will be the same length used in the MPC to be designed. Taking one route for example, Fig. 2 displays the trajectories of speed and distance, road grade acceleration, and stage cost predicted using (2), (13), and (16), respectively; the prediction results (solid red) are compared to the ground truth data (dashed blue), simulated using `RoadRunner`. Fig. 3 presents the overall statistics on the prediction error. Data points of prediction and truth are displayed in the scatter plots on the left, while the right plots show the error distributions of the observed variables.

[5] The statistics of road grade and length can be described by a bivaraite gamma distribution, whose parameters are found by maximum likelihood estimation.

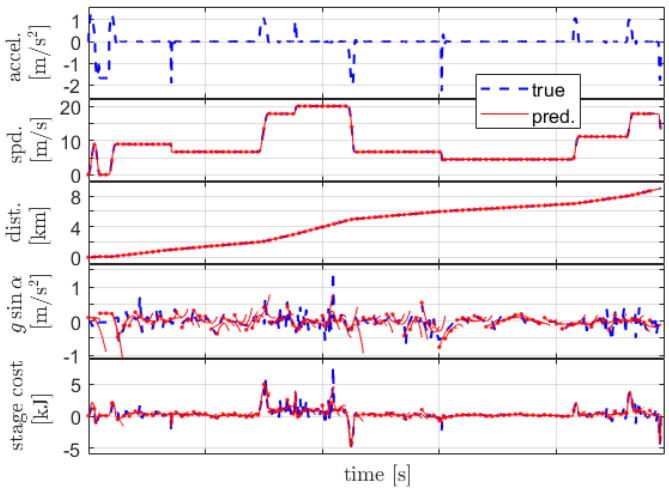

Fig. 2. Prediction results compared to truth on one route.

As discussed in Section 3.2, the original states transition is by nature linear, which explains the high correlation between the prediction and the truth (see the first two rows of Fig. 3). In contrast, in our formulation, the stage cost is intended to contain the most nonlinearities, not only in $\tilde{v}$ but also in $\tilde{s}$ over the adaptive approximation in Section 3.2.1. The last row in Fig. 3 reflects the distribution of road-grade approximation errors despite the higher-order monomial powers. Those errors then contributed to the deviation of stage costs prediction, as shown in the third row of Fig. 3. Still, we consider the prediction accuracy of the stage cost satisfactory given that the prediction's root mean square error (RMSE) is close to half of the data variance $\approx$ 1.35 kJ, and the prediction is well aligned with the truth in an average sense, as observed in the last row of Fig. 2.

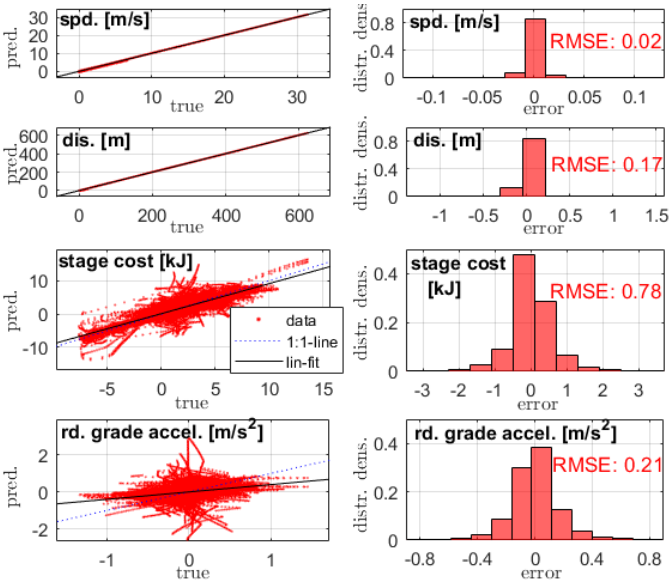

Fig. 3. Statistics of prediction results over the test set.

4. MODEL PREDICTIVE CONTROL DESIGN

Except for the *lifted* state coordinates, the Koopman MPC (KMPC) follows the same design procedure as with a classical MPC.

An MPC solves iteratively an optimal control problem (OCP) on a receding horizon and applies the first element in the optimized control sequence. In each iteration, the OCP is initialized with the current state measurement and needs to renew the constraint conditions. QP, as a simple case of OCP, aims to minimize the sum of *quadratic* stage costs $(L_k)_{k=0}^{N_\mathrm{h}-1}$ at the discretized steps within the horizon N_h, subject to the *linear* transition as in (2), *linear* inequality and initial boundary point conditions:

$$\min_{u_k, \boldsymbol{z}_k} \mathcal{J} = \sum_{k=0}^{N_\mathrm{h}-1} \left[1, \boldsymbol{z}_k^\mathsf{T}, \boldsymbol{u}_k^\mathsf{T}\right] \Omega \underbrace{\left[1, \boldsymbol{z}_k^\mathsf{T}, \boldsymbol{u}_k^\mathsf{T}\right]^\mathsf{T}}_{\zeta_k} \tag{18a}$$

$$\text{subject to} \quad \boldsymbol{z}_{k+1} = A\boldsymbol{z}_k + B u_k, \tag{18b}$$

$$\check{A}_k \boldsymbol{z}_k + \check{B}_k \boldsymbol{u}_k + \check{\boldsymbol{c}}_k \le \boldsymbol{0}, \tag{18c}$$

$$\boldsymbol{z}_0 = \boldsymbol{\psi}(\boldsymbol{x}_0), \tag{18d}$$

where matrix $\check{A}_k \in \mathbb{R}^{n_{\mathrm{ie},k} \times N}$, $\check{B}_k \in \mathbb{R}^{n_{\mathrm{ie},k} \times m}$ and vector $\check{\boldsymbol{c}}_k \in \mathbb{R}^{n_{\mathrm{ie},k}}$, with $n_{\mathrm{ie},k}$ being the rows of linear inequality condition at k-th time step. Thanks to the Koopman embedding (10), the inequality conditions (18c) can represent speed limits (17) and torque limits which are otherwise nonlinear in the original state/control space.

By evaluating the recursive stage transition condition (18b) as in Appendix A, the sparse-form QP (18) is reduced to the dense form to optimize the *input sequence* $U = [\boldsymbol{u}_0^\mathsf{T}, \boldsymbol{u}_1^\mathsf{T}, \ldots, \boldsymbol{u}_{N_\mathrm{h}-1}^\mathsf{T}]^\mathsf{T}$:

$$\begin{aligned} \min_{U} \ & \bar{\mathcal{J}} = U^\mathsf{T} \boldsymbol{\mathcal{H}} U + \boldsymbol{\mathcal{G}}^\mathsf{T} U \\ \text{subject to} \ & \check{\boldsymbol{\mathcal{A}}}_\mathbf{U} U \le \check{\boldsymbol{\mathcal{B}}}_\mathbf{U}, \end{aligned} \tag{19}$$

to which the parameters are $\boldsymbol{\mathcal{H}}$, $\boldsymbol{\mathcal{G}}$, $\check{\boldsymbol{\mathcal{A}}}_\mathbf{U}$, and $\check{\boldsymbol{\mathcal{B}}}_\mathbf{U}$ and are recomputed (see Appendix A) at each close-loop execution of the MPC due to updating $\boldsymbol{z}_0$, reassessing road grade cost weight $\widehat{\Omega}$, and adapting the speed limit constraint. [6]

The constrained QP (19) can be efficiently solved by an off-the-shelf toolbox such as `qpOASES` (Ferreau et al., 2014).

5. SIMULATION RESULTS IN ROADRUNNER

The KMPC designed in Section 4 was implemented as a closed-loop control for the vehicle selected for training in Section 3.1. The vehicle ran in `RoadRunner` on two real-world routes extracted from HERE maps: one composed mainly of highway sections (see Fig. 4), and the second from an urban area (see Fig. 5). For evaluation, KMPC is compared with two other controls: a human driver model (Han et al., 2020b) ("Human") and a successful optimal eco-driving control ("Spd-only") derived based on linearized kinetics that neglect nonlinear factors (e.g., road grade and aerodynamics) (Han et al., 2020a).

The travel time, energy consumption, and the relative benefits (against "Human") of the controls are shown in Table 1. Generally, KMPC achieves higher efficiency than Spd-only representing linear approaches at a low cost of potential increase of travel time. Apparently, including nonlinear features contributed to the higher saving potential while complying with all the constraints.

[6] Constraints such as speed-dependent torque limit, traffic light, and stop signs are also integrated in the KMPC. This paper does not discuss them in details because of the page limit.

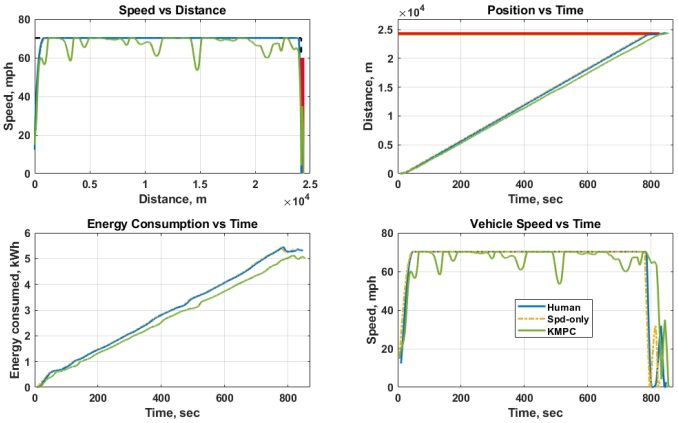

Fig. 4. `RoadRunner` simulation results on a highway route.

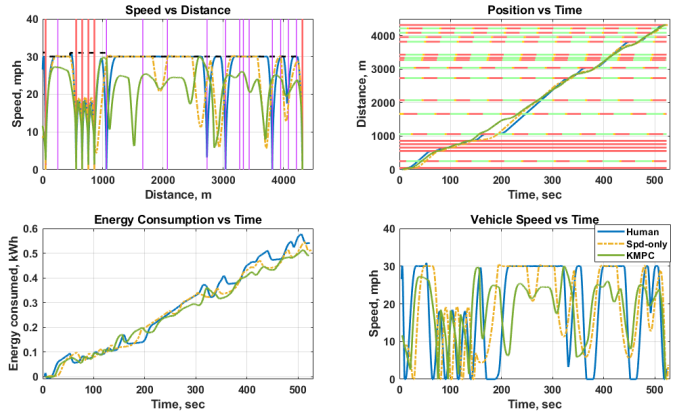

Fig. 5. `RoadRunner` simulation results on a urban route.

Table 1. Result comparison between controls.

Route	Control	Travel Time [s]		Energy [kWh]	
highway	Human	847.2	(–)	5.327	(–)
	Spd-only	826.9	(−2.40 %)	5.301	(−0.49 %)
	KMPC	855.3	(+0.96 %)	5.034	(−5.50 %)
urban	Human	522.7	(–)	0.5418	(–)
	Spd-only	525.8	(+0.59 %)	0.5125	(−5.41 %)
	KMPC	521.3	(−0.27 %)	0.4913	(−9.32 %)

6. CONCLUSION

This paper presents a general data-driven process based on Koopman analysis to build a linear-quadratic system-cost formulation in a *lifted* higher-dimension to design a KMPC for eco-driving control. Specifically, higher-order monomial powers of the original states are chosen as the Koopman embedding coordinates so that they can be utilized to approximate the varying road grade and speed limit using shifted Legendre polynomials. Hence, the road grade influence is converted to a weighting increment of the quadratic cost; the speed limit becomes linear inequality in the Koopman embedding. The approximation of these two position-dependent nonlinearities is renewed in adaptively along the preceding horizon in the KMPC. The KMPC was then validated in `RoadRunner` as a feedback control and demonstrated the efficacy considering nonlinearities compared to a linear approach.

ACKNOWLEDGEMENTS

DOE Office of Energy Efficiency and Renewable Energy (EERE) manager David Anderson played an important role in establishing the project concept, advancing implementation, and providing ongoing guidance.

REFERENCES

De Nunzio, G., Canudas de Wit, C., Moulin, P., and Di Domenico, D. (2013). Eco-driving in urban traffic networks using traffic signal information. In *Proc. IEEE 52nd Annu. Conf. Decis. and Control (CDC 2013)*, 892–898. Florence, Italy. doi:10.1109/CDC.2013.6759995.

Ferreau, H.J., Kirches, C., Potschka, A., Bock, H.G., and Diehl, M. (2014). qpOASES: A parametric active-set algorithm for quadratic programming. *Math. Program. Comput.*, 6(4), 327–363.

Han, J., Karbowski, D., and Kim, N. (2020a). Closed-form solutions for a real-time energy-optimal and collision-free speed planner with limited information. In *Proc. 2020 Annu. Amer. Control Conf. (ACC 2020)*, 268–275. Denver, CO, USA.

Han, J., Karbowski, D., Kim, N., and Rousseau, A. (2020b). Human driver modeling based on analytical optimal solutions: Stopping behaviors at the intersections. *ASME Lett. in Dyn. Syst. Control*, 1(1). Art. 011010.

Kaiser, E., Kutz, J.N., and Brunton, S.L. (2021). Data-driven discovery of Koopman eigenfunctions for control. *Mach. Learn.: Sci. Technol.*, 2(3), 035023.

Kalabis, M. and Müller, S. (2012). A model predictive approach for a fuel efficient cruise control system. In H. Proff, J. Schönharting, D. Schramm, and J. Ziegler (eds.), *Zukünftige Entwicklungen in der Mobilität*, 201–211. Gabler Verlag, Wiesbaden.

Kim, N., Karbowski, D., and Rousseau, A. (2018). A modeling framework for connectivity and automation co-simulation. In *WCX World Congr. Experience*. SAE Int., Detroit, MI, USA.

Koopman, B.O. (1931). Hamiltonian systems and transformation in Hilbert space. *Proc. Natl. Acad. Sci. U.S.A.*, 17(5), 315–318.

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

Proctor, J.L., Brunton, S.L., and Kutz, J.N. (2018). Generalizing Koopman theory to allow for inputs and control. *SIAM J. Appl. Dyn. Syst.*, 17(1), 909–930.

Quarteroni, A., Sacco, R., and Saleri, F. (2007). *Numerical Mathematics*. Springer Berlin Heidelberg, 2nd edition.

Sciarretta, A., de Nunzio, G., and Luis, L.O. (2015). Optimal ecodriving control: Energy-efficient driving of road vehicles as an optimal control problem. *IEEE Control Syst. Mag.*, 35(5), 71–90.

Shen, D., Karbowski, D., and Rousseau, A. (2021). Solving eco-driving problems using indirect collocation method and smooth representation. *IEEE Control Syst. Lett.*, 5(5), 1501–1506.

Wan, N., Vahidi, A., and Luckow, A. (2016). Optimal speed advisory for connected vehicles in arterial roads and the impact on mixed traffic. *Transp. Res. Part C Emerg. Technol.*, 69(1), 548–563.

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

Xu, S., Li, S.E., Deng, K., Li, S., and Cheng, B. (2015). A unified pseudospectral computational framework for optimal control of road vehicles. *IEEE/ASME Trans. Mechatronics*, 20(4), 1499–1510.

Appendix A. DENSE-FORM QP

Evaluating the recursive discrete system (18b) gives us the state sequence on the horizon $Z = [\boldsymbol{z}_0^\mathsf{T}, \boldsymbol{z}_1^\mathsf{T}, \dots, \boldsymbol{z}_{N_\mathrm{h}-1}^\mathsf{T}]^\mathsf{T}$ as a linear combination of the initial state $\boldsymbol{z}_0$ and the control sequence $U = [\boldsymbol{u}_0^\mathsf{T}, \boldsymbol{u}_1^\mathsf{T}, \dots, \boldsymbol{u}_{N_\mathrm{h}-1}^\mathsf{T}]^\mathsf{T}$ involving batch matrices $\boldsymbol{\mathcal{S}}_\mathrm{z}$ and $\boldsymbol{\mathcal{S}}_\mathrm{u}$:

$$Z = \underbrace{\begin{bmatrix} I \\ A \\ A^2 \\ \vdots \\ A^{N_\mathrm{h}-1} \end{bmatrix}}_{\boldsymbol{\mathcal{S}}_\mathrm{z}} \boldsymbol{z}_0 + \underbrace{\begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ B & 0 & \cdots & 0 & 0 \\ AB & B & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & 0 \\ A^{N_\mathrm{h}-2}B & \cdots & AB & B & 0 \end{bmatrix}}_{\boldsymbol{\mathcal{S}}_\mathrm{u}} U. \quad \text{(A.1)}$$

Using (A.1), the cost function $\mathcal{J}$ in (18) is rewritten as

$$\mathcal{J} = Z^\mathsf{T}\bar{\mathbf{Q}}Z + U^\mathsf{T}\bar{\mathbf{R}}U + Z^\mathsf{T}\bar{\mathbf{P}}U + \bar{\mathbf{q}}^\mathsf{T}Z + \bar{\mathbf{r}}^\mathsf{T}U + N_\mathrm{h}c,$$

where block-diagonal matrices $\{\bar{\mathbf{Q}}, \bar{\mathbf{R}}, \bar{\mathbf{P}}\} = \mathrm{diag}(\{Q, R, P\}, \dots, \{Q, R, P\})$, and stacked vectors $\{\bar{\mathbf{q}}, \bar{\mathbf{r}}\} = \big[\{\boldsymbol{q}^\mathsf{T}, \boldsymbol{r}^\mathsf{T}\}, \dots, \{\boldsymbol{q}^\mathsf{T}, \boldsymbol{r}^\mathsf{T}\}\big]^\mathsf{T}$. Dropping the term independent of U using (A.1) leads to $\bar{\mathcal{J}}$,

$$\bar{\mathcal{J}} = \mathcal{J} - \boldsymbol{z}_0^\mathsf{T}\boldsymbol{\mathcal{S}}_\mathrm{z}^\mathsf{T}\bar{\mathbf{Q}}\boldsymbol{\mathcal{S}}_\mathrm{z}\boldsymbol{z}_0 - N_\mathrm{h}c = U^\mathsf{T}\boldsymbol{\mathcal{H}}U + \boldsymbol{\mathcal{G}}U$$

$$\text{with} \quad \begin{aligned} \boldsymbol{\mathcal{H}} &\triangleq \boldsymbol{\mathcal{S}}_\mathrm{u}^\mathsf{T}\bar{\mathbf{Q}}\boldsymbol{\mathcal{S}}_\mathrm{u} + \bar{\mathbf{R}} + \boldsymbol{\mathcal{S}}_\mathrm{u}^\mathsf{T}\bar{\mathbf{P}}, \\ \boldsymbol{\mathcal{G}} &\triangleq \boldsymbol{z}_0^\mathsf{T}\big(2\boldsymbol{\mathcal{S}}_\mathrm{z}^\mathsf{T}\bar{\mathbf{Q}}\boldsymbol{\mathcal{S}}_\mathrm{u} + \boldsymbol{\mathcal{S}}_\mathrm{z}^\mathsf{T}\bar{\mathbf{P}}\big) + \bar{\mathbf{q}}^\mathsf{T} + \bar{\mathbf{r}}^\mathsf{T}\boldsymbol{\mathcal{S}}_\mathrm{u}. \end{aligned} \quad \text{(A.2)}$$

The sparse expression of constraints (18c) for steps $k = 0, \dots, N_\mathrm{h}$ can be written in a single block-matrix form,

$$\check{\mathbf{A}}Z + \check{\mathbf{B}}U + \check{\mathbf{c}} \leq \mathbf{0},$$

with $\{\check{\mathbf{A}}, \check{\mathbf{B}}\} \triangleq \mathrm{diag}\big(\{\check{A}_0, \check{B}_0\}, \dots, \{\check{A}_{N_\mathrm{h}-1}, \check{B}_{N_\mathrm{h}-1}\}\big)$ and $\check{\mathbf{c}} \triangleq \big[\check{\boldsymbol{c}}_0^\mathsf{T}, \check{\boldsymbol{c}}_1^\mathsf{T}, \dots, \check{\boldsymbol{c}}_{N_\mathrm{h}-1}^\mathsf{T}\big]^\mathsf{T}$. Further, by substituting Z with (A.1), inequality condition becomes dependent on $\boldsymbol{z}_0$ and U:

$$\underbrace{\big(\check{\mathbf{A}}\boldsymbol{\mathcal{S}}_\mathrm{u} + \check{\mathbf{B}}\big)}_{\triangleq \check{\boldsymbol{\mathcal{A}}}_\mathbf{U}} U \leq \underbrace{-\check{\mathbf{A}}\boldsymbol{\mathcal{S}}_\mathrm{z}\boldsymbol{z}_0 - \check{\mathbf{c}}}_{\triangleq \check{\boldsymbol{\mathcal{B}}}_\mathbf{U}}.$$