

Koopman-Based Data-Driven Techniques for Adaptive Cruise Control System Identification*

Yiming Meng¹, Hangyu Li², Melkior Ornik^{1,3}, and Xiaopeng Li²

Abstract—Accurately identifying the intrinsic model of Adaptive Cruise Control has the potential to enhance the prediction of automated car-following behavior, helping vehicles’ decision-making and contributing to safer and more efficient traffic flows. Moreover, white box models offer an analytical base for evaluating the impact of automated driving functions on macroscopic traffic dynamics, consequently aiding the management of the whole intelligent transportation system. Many existing system identification techniques have been applied to automated vehicles. However, most of these studies focus on identifying parameters for models of a fixed prototype. Their reliance on accurate estimation of state time derivatives prevents their real applications, challenged by low sampling rates, noisy measurements, and limited observation periods. In contrast, the Koopman operator learning framework presents a promising improvement that can identify the nonlinear evolutionary properties of continuous-time systems.

In this study, we apply Koopman-based methods to data-driven Adaptive Cruise Control model identification. Additionally, as the challenge remains in establishing a practical relationship between identification accuracy and sampling rate, we numerically compared the performance of three Koopman-based learning frameworks, finite-difference, Koopman-logarithm, and a newly devised resolvent-type method, with that of a commonly used offline simulation-based batch optimization approach. We introduce a novel modification to the resolvent-type method, and the experimental results demonstrate its state-of-the-art performance, particularly in identifying the potential existence of parametric noise at lower sampling rates.

I. INTRODUCTION

Automated Vehicles (AV) are expected to enhance traffic safety, increase roadway capacity, and reduce energy consumption [1], [2]. A prevalent example of AV functions is Adaptive Cruise Control (ACC), which enables the ego AV to adjust its velocity in response to the preceding vehicle’s behavior [3]. As an automated function for the most fundamental car-following behavior in traffic, it has been widely implemented by many production vehicles [4]. Despite its wide adoption and promising potential, conservative strategies [5], [6] and string instability [7], [8] that hinder traffic performance are observed due to uncertainties or design defects. The black box naturalistic of these non-

open-source commercial AV models further challenges accurate traffic predictions and utility analyses of these AVs in an analytical way [9], [10]. The intelligent transportation community is motivated to identify effective models that accurately represent how these vehicles behave in practice.

Efforts have been made to identify commercial ACC systems through offline optimization procedures to align with recorded data [11], [12]. The characteristics of low computational efficiency limit the scale of data it can process. The abovementioned methods assume all ACC models to be one specific type. Therefore, model identification degenerates to regressing the unknown parameters to fit the empirical data. As a result, these methods have poor accuracy when facing different models. They are also incapable of representing nonlinear interference or uncertainty, preventing them from restoring unstable and unsafe behavior in reality. In addition, these methods require state time derivatives to be accurately estimated, which may not be robustly satisfied due to potential challenges such as low sampling rates, noisy measurements, and short observation periods.

Comparatively, the learning structure of Koopman operators [13], based solely on observed state data snapshots, enhances the robustness of model identification through linear least squares optimization over a dictionary that includes nonlinear functions. Through certain transformations of the learned Koopman operator, such as [14], [15], [16], the transient state dynamics can be obtained as a linear combination of the dictionary functions. Furthermore, this indirect system identification framework does not require the estimation of time derivatives and can potentially circumvent the need for high sampling rates and long observation periods.

The Koopman-based identification approach has been successfully used in modeling discrete-time models for AVs, demonstrating practical applications of Model Predictive Control (MPC) [17], [18], [19]. However, they fail to identify the transient acceleration dynamics for ACC without the aforementioned *transformations* of the learned Koopman operators. Though challenges remain in establishing a practical relationship between identification accuracy and the sampling rate, various transformation methods that encode only discrete-time observation data, such as *finite-difference* method (FDM) [14], *Koopman-logarithm* method (KLM) [15], and newly proposed *resolvent-type* method (RTM) [16], have been investigated with theoretical guarantees of approximation soundness.

Considering all the above, in this work, we innovatively apply car-following physical laws and leverage Koopman-based data-driven techniques along with the three transfor-

*This research was supported by NASA under grant numbers 80NSSC21K1030 and 80NSSC22M0070, as well as by the Air Force Office of Scientific Research under grant number FA9550-23-1-0131.

¹ Yiming Meng and Melkior Ornik are with the Coordinated Science Laboratory, University of Illinois Urbana-Champaign, Urbana, IL, 61801, USA.

² Hangyu Li and Xiaopeng Li are with the Department of Civil and Environmental Engineering, University of Wisconsin-Madison, Madison, WI, 53706, USA.

³ Melkior Ornik is also with the Department of Aerospace Engineering, University of Illinois Urbana-Champaign, Urbana, IL, 61801, USA.

mation methods for ACC model identification. To better understand how data can be processed to learn the continuous-time ACC dynamical system with improved accuracy and data efficiency, we conduct simulation experiments and numerically compare the performance with a commonly used offline simulation-based batch optimization approach. We claim the overall advantages of Koopman-based learning techniques and the state-of-the-art capability of the RTM method. We also hope that the simulation results will shed light on processing real experimental field data in the future and call for more studies to explore the applications of this technique that achieves white box ACC models.

II. PREMISE

In this section, we introduce one important notation used, and briefly review common models assumed for ACC vehicle dynamics, then discuss a standard simulation-based optimization method used to benchmark the model parameters.

A. Notation

We denote by $\|\cdot\|$ the Euclidean norm. For a finite-dimensional matrix A , we denote by A^\dagger the pseudo inverse, and use the Frobenius norm $\|A\|_F$ as the metric. Let $C(\Omega)$ be the set of continuous functions with domain Ω . The uniform (sup) norm of $C(\Omega)$ is denoted as $\|\cdot\|_\infty$. We denote the set of continuously differentiable functions by $C^1(\Omega)$. For any positive integer P , we denote by $[P]$ the set $\{0, 1, \dots, P-1\}$. The identity operator is denoted by \mathbf{I} , whereas an $N \times N$ identity matrix is denoted by $\mathbf{I}_{N \times N}$.

B. Model Review

Physics-based models for two vehicles, where a lead vehicle drives at a given velocity profile with the test ACC vehicle following behind, describe the relationship between speed deviations, the distance gap, and the following vehicle's acceleration. We express the dynamics in the following abstract form:

$$\frac{d}{dt} \begin{bmatrix} s(t) \\ v(t) \end{bmatrix} = \begin{bmatrix} u(t) - v(t) \\ f_\theta(s(t), v(t), u(t)) \end{bmatrix} \quad (1)$$

where s is the distance gap, v is the velocity of the following vehicle, and u denotes the velocity of the preceding vehicle. The mapping $f_\theta : \mathbb{R}^3 \rightarrow \mathbb{R}$ is parameterized by θ . In this form, we treat u as the external input variable assumed to be given. The equilibrium of (1) occurs when both vehicles travel at the desired speed \bar{v} , whence $u(t) = v(t) \equiv \bar{v}$ and $s(t) \equiv \bar{s}$ with a desired distance gap \bar{s} . Therefore, it follows that $f_\theta(\bar{s}, \bar{v}, \bar{v}) = 0$.

Suppose the input u is designed as $u = \kappa(s, v, u)$ for some Lipschitz continuous function κ . Then, (1) can be rewritten as an autonomous system

$$\dot{x}(t) = F(x(t)), \quad x(0) = x_0 \in \mathcal{X}, \quad t \in [0, \infty), \quad (2)$$

where $x = [s, v, u]^T$ is the state variables, $x_0 = [s_0, v_0, u_0]^T$ represents the initial condition, $F(x) = [u - v, f_\theta(s, v, u), \kappa(s, v, u)]^T$, and $\mathcal{X} \subseteq \mathbb{R}^3$ represents a bounded and invariant domain for (s, v, u) . The forward flow map

(solution map) is denoted as $\mathbf{x} : [0, \infty) \times \mathcal{X} \rightarrow \mathcal{X}$, which satisfies 1) $\partial_t(\mathbf{x}(t, x_0)) = F(\mathbf{x}(t, x_0))$, 2) $\mathbf{x}(0, x_0) = x_0$, and 3) $\mathbf{x}(s, \mathbf{x}(t, x_0)) = \mathbf{x}(t + s, x_0)$ for all t and s . Such an abstract autonomous system form favors system identification methods.

Below, we summarize commonly used realizations of f_θ .

1) Constant Time Headway Relative Velocity Model:

Among the numerous studies exploring the modeling of ACC vehicles using car-following models, a common variation is the Constant Time Headway Relative Velocity (CTH-RV) model. In this model, $f_\theta(s, v, u) = \alpha(s - \eta - \tau v) - \beta(u - v)$ with $\theta = [\eta, \alpha, \beta, \tau]^T$, where η is the desired space gap when the vehicles are stop; α and β represent the gain on the constant time headway term and the relative velocity term, respectively; $\tau = \frac{\bar{s} - \eta}{\bar{v}}$ represents the time gap at equilibrium. This linear model is a simplification of the proprietary control logic and complex vehicle dynamics of real ACC vehicles. The quality of fit can decrease for some specialized vehicles, such as hybrid vehicles [11], [20].

2) *Nonlinear Models*: Many models account for nonlinear effects [21], [22], [23]. Due to page limitations, we present only the GHR model [24] here.

The acceleration of the following vehicle calculated using the GHR model is formulated as: $f_\theta(s, v, u) = cv^\delta(u - v)s^{-l}$, where $\theta = [c, \delta, l]^T$ are model parameters. The model also recognizes the potential to incorporate a delay effect due to human reaction. However, to better illustrate the ideas in this paper, we do not take this effect into account.

C. Benchmark System Identification Algorithms

We present the following commonly used method for system identification that relies on known physical laws but have unknown parameters.

The Offline Batch Optimization (OBO) method estimates parameters by minimizing the root mean squared error (RMSE) between the actual time-series data $\{\mathbf{x}(t_k, x^{(m)})\}_{k=0}^\Gamma$ and the estimated data $\{\hat{\mathbf{x}}(t_k, x^{(m)})\}_{k=0}^\Gamma$ up to time T . Here, each $x^{(m)}$ represents an initial condition, and a total of M initial conditions are needed. The time $t_k = kT/\Gamma$ corresponds to the sampling instances, with Γ representing the sampling rate. The formula is given as:

$$\mathcal{E}_F = \sqrt{\frac{1}{M\Gamma} \sum_{k=1}^\Gamma \sum_{m=0}^{M-1} \|\mathbf{x}(t_k, x^{(m)}) - \hat{\mathbf{x}}(t_k, x^{(m)})\|^2}. \quad (3)$$

The identification can be directly addressed using simulation-based optimization with standard descent-based methods. It is important to note that this optimization problem is nonlinear with respect to the decision variables [20], including the state and model parameters, and may also be non-convex depending on the car-following model used. As a result, the computational time is usually long.

III. KOOPMAN-BASED SYSTEM IDENTIFICATION METHODS

A Koopman operator is a linear, infinite-dimensional operator that governs the evolution of continuous scalar test

functions $h : \mathcal{X} \rightarrow \mathbb{R}$. Let us now consider the normed continuous function space $(C(\mathcal{X}), \|\cdot\|_\infty)$ of all such h .

Definition 1: The Koopman operator family $\{\mathcal{K}_t\}_{t \geq 0}$ of system (see Equation (2)) is a collection of Koopman operators $\mathcal{K}_t : C(\mathcal{X}) \rightarrow C(\mathcal{X})$ defined by:

$$\mathcal{K}_t h = h \circ \mathbf{x}(t, \cdot), \quad h \in C(\mathcal{X}) \quad (4)$$

for each $t \geq 0$, where \circ is the composition operator. \diamond

Note that Koopman operators form a semigroup and satisfy 1) $\mathcal{K}_0 = \mathbf{I}$, and 2) $\mathcal{K}_t \circ \mathcal{K}_s = \mathcal{K}_{t+s}$ for every $t, s \geq 0$. In this view, for unknown systems, one can gather flow data of $\mathbf{x}(\tau, \cdot)$ and learn \mathcal{K}_τ for some fixed τ . Then, one can predict the data of $h(\mathbf{x}(k\tau, \cdot))$ for any $k \in \mathbb{N}$ and $h \in C(\mathcal{X})$, including the discrete flow of the ODE system itself. This knowledge can be integrated with discrete-time synthesis strategies to design controllers that guarantee vehicle behaviors at those discrete-time observation points.

However useful these applications may be, they lack guarantees for performance at moments other than those discrete-time points in continuous systems, especially when safety-critical operational requirements are considered. In addition, for the specific application of understanding the underlying automation logic of an ACC vehicle, the transient transitions need to be comprehensively analyzed. To capture dynamical behaviors related to the transient transitions, one needs to learn the infinitesimal generator of the system.

Definition 2: The (infinitesimal) generator \mathcal{L} of $\{\mathcal{K}_t\}_{t \geq 0}$ is defined by $\mathcal{L}h(x) = \lim_{t \rightarrow 0} \frac{\mathcal{K}_t h(x) - h(x)}{t}$, where the test functions should be within the domain of \mathcal{L} , defined as $\text{dom}(\mathcal{L}) = \left\{ h \in \mathcal{F} : \lim_{t \rightarrow 0} \frac{\mathcal{K}_t h(x) - h(x)}{t} \text{ exists} \right\}$. \diamond

Supposing that the test functions are continuously differentiable, the generator is explicitly given by $\mathcal{L}h(x) = \nabla h(x) \cdot F(x)$ for all $h \in C^1(\mathcal{X})$ [25]. For the purpose of system identification, we define $p_j : \mathcal{X} \rightarrow \mathbb{R}$, $x \mapsto x_j$, as the projection function to the j^{th} dimension. Then, $\mathcal{L}p_j(x) = \nabla p_j(x) \cdot F(x) = F_j(x)$ for all $j \in \{1, 2, 3\}$. Since F_1 and F_3 are known, we only need to utilize the generator information to obtain $\mathcal{L}p_2 = f_\theta$ for the ACC system.

Since Koopman-based methods can enhance approximation capabilities through the rich representability of the dictionary functions, which relaxes prior beliefs about the prototypical forms of the model, we will review two frequently used connections between \mathcal{L} and $\{\mathcal{K}_t\}$ and a recently developed approach in the following subsections. Such a connection is also intended for data-driven learning purposes. In contrast to the predictability of discrete-time flows achieved through learning Koopman operators, identifying $F(x)$ for continuous-time systems requires utilizing the discrete-time data of $\{\mathcal{K}_t\}$ to represent \mathcal{L} .

Note that the solutions to the above problem can be improved by increasing the observation frequency. But in practice, such a frequency is limited. Therefore, the primary focus will be on how these Koopman-based methods can potentially reduce the frequency of observation data collection.

A. Finite-Difference Method

Finite-difference method (FDM) is a numerical technique used to solve differential equations by approximating derivatives with finite differences. In the context of Koopman-based approximations, the finite-difference expression, given as

$$\mathcal{L} \approx \frac{\mathcal{K}_{\tau_s} - \mathbf{I}}{\tau_s}, \quad \tau_s > 0, \quad (5)$$

follows Definition 2 without taking the limit.

Through this approximation scheme of the time derivative, it can be anticipated that the precision heavily depends on the size of τ_s [14], [26], [27].

B. Koopman-Logarithm Method

The Koopman-Logarithm Method (KLM) leverages the expression $\mathcal{K}_{\tau_s} = e^{\mathcal{L}\tau_s}$ for bounded linear \mathcal{L} , and uses

$$\mathcal{L} \approx \frac{\log(\mathcal{K}_{\tau_s})}{\tau_s}, \quad \tau_s > 0 \quad (6)$$

as the converse representation of the generator [15], based on the knowledge of \mathcal{K}_{τ_s} . A potential advantage of the KLM is that it is more tolerant of variations in sampling rate compared to the FDM.

Two issues need to be paid attention to when applying this method: 1) Such an expression requires the existence of an invariant function subspace of $C(\mathcal{X})$ where the generator \mathcal{L} , when restricted to it, is bounded; 2) The sampling rate should be sufficient to ensure that the operator logarithm is not multi-valued.

Recent studies [28], [29] have investigated the sufficient and necessary conditions under which the KLM can guarantee learning accuracy. Although the results are sound, these conditions are less likely to be verifiable for unknown systems unless additional side information is provided.

C. Resolvent-Type Method

Recent research [16] has proposed the resolvent-type method (RTM) to address the proper converse representation of \mathcal{L} based on $\{\mathcal{K}_t\}$, which also has the potential to relax the sampling rate requirements. Unlike the FDM and KLM, which only depend on τ_s , the RTM approximates \mathcal{L} using two tunable parameters, as follows. For sufficiently large $\lambda > 0$ and $\tau_s > 0$,

$$\mathcal{L} \approx \lambda^2 \mathcal{R}_{\lambda, \tau_s} - \lambda \mathbf{I}, \quad (7)$$

where $\mathcal{R}_{\lambda, \tau_s} : C(\mathcal{X}) \rightarrow C(\mathcal{X})$ is defined as:

$$\mathcal{R}_{\lambda, \tau_s} h(x) = \int_0^{\tau_s} e^{-\lambda s} \mathcal{K}_s h(x) ds. \quad (8)$$

Note that as $\tau_s \rightarrow \infty$, $\mathcal{R}_{\lambda, \tau_s} \rightarrow \mathcal{R}_\lambda$ uniformly and exponentially, where $\mathcal{R}_\lambda = \int_0^\infty e^{-\lambda s} \mathcal{K}_s h(x) ds$ is the resolvent operator of \mathcal{L} . In this sense, (7) truncates the tail of the resolvent integral in favor of adapting to a finite amount of data. In contrast to KLM, the RTM does not require the boundedness of \mathcal{L} , and the approximation accuracy has been thoroughly investigated in [16].

IV. DATA-DRIVEN TECHNIQUES

In this section, we introduce the data-driven adaptation of the previously introduced three Koopman-based system identification methods. We also propose a modified version of RTM, aiming to relax the sampling rate requirements.

A. Finite-Difference and Koopman-Logarithm Method

Observing that the expressions for \mathcal{L} in FDM and KLM rely on just one moment of the Koopman operator \mathcal{K}_{τ_s} , the data-driven versions of these methods are divided into two steps: 1) learning \mathcal{K}_{τ_s} ; 2) transforming the learned \mathcal{K}_{τ_s} based on (5) and (6), respectively.

The objective of the data-driven approximation of Koopman operators [15], [30], [31] is to obtain a fully discretized version, denoted \mathbf{K} , of \mathcal{K}_{τ_s} based on training data. This also relies on the selection of a discrete dictionary of continuously differentiable test functions, which is denoted by $\mathcal{Z}_N(x) = [\mathfrak{z}_0(x), \mathfrak{z}_1(x), \dots, \mathfrak{z}_{N-1}(x)]^T$ for $N \in \mathbb{N}$. Then, the approximation is achieved in a way that $\mathcal{L}h(\cdot) \approx \mathcal{Z}_N(\cdot)(\mathbf{L}_i \mathbf{w})$ for $h(x) = \mathcal{Z}_N(x)\mathbf{w}$ and for any $i \in \{\text{FDM, KLM}\}$.

The training data is obtained in the following way. By randomly sampling M initial conditions $\{x^{(m)}\}_{m=0}^{M-1} \subseteq \mathcal{X}$ and fixing a τ_s , we stack the *features* into \mathbf{X} , such that:

$$\mathbf{X} = [\mathcal{Z}_N(x^{(0)}), \mathcal{Z}_N(x^{(1)}), \dots, \mathcal{Z}_N(x^{(M-1)})]^T \quad (9)$$

and the *labels* into \mathbf{Y} :

$$\mathbf{Y} = [\mathcal{Z}_N(\mathbf{x}(\tau_s, x^{(0)}), \dots, \mathcal{Z}_N(\mathbf{x}(\tau_s, x^{(M-1)}))]^T. \quad (10)$$

After obtaining the training data (\mathbf{X}, \mathbf{Y}) , we can find \mathbf{K} by $\mathbf{K} = \operatorname{argmin}_{A \in \mathbb{C}^{N \times N}} \|\mathbf{Y} - \mathbf{X}A\|_F$. The \mathbf{K} is given in closed-form as $\mathbf{K} = (\mathbf{X}^T \mathbf{X})^\dagger \mathbf{X}^T \mathbf{Y}$ [30].

The data-driven approximation for \mathcal{L} based on FDM and KLM should be a quick modification using \mathbf{K} : 1) FDM: $\mathbf{L}_{\text{FDM}} = (\mathbf{K} - \mathbf{I}_{N \times N})/\tau_s$; 2) KLM: $\mathbf{L}_{\text{KLM}} = \log(\mathbf{K})/\tau_s$.

B. Resolvent-Type Method

The RTM relies on integrals for approximation, which require intermediate trajectory data for evaluation. In [16], the integrals are evaluated using an augmented initial value problem (IVP) related to Equation (2), where the evaluation points within $[0, \tau_s]$ are implicitly assigned and are not accessible for processing real data.

In this subsection, we first modify the evaluation algorithm for (8) as described in [16] and organize the data as follows. For a fixed τ_s and λ , the stack of features \mathbf{X} is the same as Equation (9). To numerically evaluate the integral in Equation (8), one needs to select the number of evaluation point Γ within $[0, \tau_s]$, and stack the data in the following intermediate matrix for $m \in \{0, 1, \dots, M-1\}$:

$$\begin{aligned} \mathbf{J}^{(m)} = & \lambda^2 [\mathcal{Z}_N(\mathbf{x}(0, x^{(m)})), \dots, e^{-\frac{\lambda k \tau_s}{\Gamma}} \mathcal{Z}_N(\mathbf{x}(\frac{k \tau_s}{\Gamma}, x^{(m)})), \\ & \dots e^{-\frac{\lambda \tau_s}{\Gamma}} \mathcal{Z}_N(\mathbf{x}(\tau, x^{(m)}))]^T. \end{aligned} \quad (11)$$

Denote $\mathcal{G}_{[0, \tau_s]}^\lambda(v)$, or simply $\mathcal{G}(v)$ for brevity, as the Gauss-Legendre quadrature based on the vector of points v within $[0, \tau_s]$, and denote $\mathbf{J}^{(m)}[:, j]$ by the j^{th} column of $\mathbf{J}^{(m)}$. The stack of labels is given by $\mathbf{Y}_\lambda = \mathcal{I}_\lambda - \lambda \mathbf{X}$, where:

$$\mathcal{I}_\lambda = \begin{bmatrix} \mathcal{G}(\mathbf{J}^{(0)}[:, 0]) & \cdots & \mathcal{G}(\mathbf{J}^{(0)}[:, N-1]) \\ \vdots & \ddots & \vdots \\ \mathcal{G}(\mathbf{J}^{(m)}[:, 0]) & \cdots & \mathcal{G}(\mathbf{J}^{(m)}[:, N-1]) \\ \vdots & \ddots & \vdots \\ \mathcal{G}(\mathbf{J}^{(M-1)}[:, 0]) & \cdots & \mathcal{G}(\mathbf{J}^{(M-1)}[:, N-1]) \end{bmatrix}. \quad (12)$$

Then, $\mathbf{L}_{\text{RTM}}^\lambda = (\mathbf{X}^T \mathbf{X})^\dagger \mathbf{X}^T \mathbf{Y}_\lambda$ is the solution to $\operatorname{argmin}_{A \in \mathbb{C}^{N \times N}} \|\mathbf{Y}_\lambda - \mathbf{X}A\|_F$. Furthermore, supposing that Γ and λ are selected such that each entry in Equation (12) can be precisely computed, we have $\mathcal{L}h(\cdot) \approx \mathcal{Z}_N(\cdot)(\mathbf{L}_{\text{RTM}}^\lambda \mathbf{w})$ for any $h(x) = \mathcal{Z}_N(x)\mathbf{w}$.

Empirically, for larger λ , the numerical integration needs a larger Γ , which represents higher sampling rate. To avoid this, one can rely on the (first resolvent) identity $[(\lambda - \mu)\mathcal{R}_\mu + \mathbf{I}]\mathcal{R}_\lambda = \mathcal{R}_\mu$ to express \mathcal{L} . Based on (7), a modification of the above identity yields $[\mathcal{R}_{\mu, \tau_s}(\lambda - \mu) + \mathbf{I}]\mathcal{L} \approx \lambda \mu \mathcal{R}_{\mu, \tau_s} - \lambda \mathbf{I}$. For the data-driven adaptation, supposing \mathcal{I}_μ has been evaluated for a small μ , we have that $\mathbf{L}_{\text{RTM}}^\lambda = A^\dagger B$, where:

$$A = \frac{\lambda - \mu}{\mu^2} \mathcal{I}_\mu + \mathbf{X}, \text{ and } B = \frac{\lambda}{\mu} \mathcal{I}_\mu - \lambda \mathbf{X}. \quad (13)$$

V. EXPERIMENTS

In this section, we test the performance of three Koopman-based methods and compare them to offline batch optimization using synthetically generated data. This numerical simulation aims to illustrate the potential of Koopman-based methods to accurately recover true system transitions, particularly in the presence of parametric noise.

Synthetic data is created by selecting physical models and a predefined preceding vehicle velocity profile. Letting $t_k = k\tau_s/\Gamma$, the time-series data $\{\mathbf{x}(t_k, x^{(m)})\}_{k=0}^\Gamma$ of velocity and space gap are then generated using a numerical IVP solver up to time τ_s under sampled initial conditions $\{x^{(m)}\}_{m=0}^{M-1}$.

To better illustrate the idea, the data is based on the following models:

- 1) CTH-RV with $\eta = 0[m]$, $\alpha = 0.08[m/s^2]$, $\beta = 0.12[m/s^2]$, and $\tau = 1.5[s]$;
- 2) CTH-RV with polynomial parametric noise $0.001(s - \tau v)^2$ and all other parameters remain unchanged;
- 3) Non-polynomial GHR with $c = 0.79$, $\delta = 0.08$, $l = 0$, and a polynomial parametric noise $0.001(s - 1.5v)^2$.

The preceding vehicle's velocities for operator training are set as constants (equivalently, $\kappa \equiv 0$ as in (2)) $\{2, 3.33, 4.67, 6, 7.33, 8.67, 10, 11.33, 12.67, 14\}[m/s]$. We sample 10 initial conditions s_0 uniformly from $[2, 20]$ ($[m]$), as well as 10 for v_0 uniformly from $[2, 14]$ ($[m/s]$). Along with the preceding vehicle's velocity profile, the total number of initial conditions is $M = 10^3$.

The parameters are chosen based on [8], [32], whereas the noise term is artificially provided. Note that the nominal models in Section II are arguably accurate enough to reflect the behaviors of the following car, and evidence from [32, Table III (IDM)] shows that the parameters fitting the model do not make logical sense. Therefore, the purpose of the simulation leans more towards using data to identify or

approximate the transition in the following car’s velocity profile rather than providing true physical meanings.

For Koopman-based methods, we choose monomial dictionary functions $\{s^p v^q u^l, p \in [P], q \in [Q], j \in [J]\}$. Under this setting, recalling Definition 2, we have that $f_\theta(s, v, u) = \mathcal{L}v$. For the OBO, the parameter values that yield the lowest RMSE (see Equation (3)) across 100 runs are selected as the optimal parameter set for the model.

Two measurements are used to demonstrate the error of identification. For any model and any method, one can use the RMSE of the flow data, \mathcal{E}_F , as defined in (3). For polynomial models, we can simplify by using the RMSE of the weights assigned to each monomial basis

$$\mathcal{E}_w := \sqrt{\frac{1}{PQJ} \sum_{p=0}^{P-1} \sum_{q=0}^{Q-1} \sum_{j=0}^{J-1} |\mathbf{w}_{p,q,k} - \hat{\mathbf{w}}_{p,q,k}|^2}. \quad (14)$$

A. CTH-RV

The significance of this model’s simulation is twofold: 1) the linear model simplifies the parameters and demonstrates its ability to fit by roughly simulating the real trajectories of ACC-equipped vehicles.; and 2) the precision of the numerical quadrature (8) in the RTM can be roughly estimated by the linear growth rate.

1) *Test of Parameters of the RTM:* We first test how the parameters μ , λ , and τ_s impact the precision of the RTM. In the experiments, we set $P = Q = J = 3$ and $\lambda = 1e8$. We first use μ to evaluate the integral under the sampling rate 100 Hz (equivalently, $\Gamma = 100$), and then use (13) to learn the generator. In Fig. 1, the relationship between τ_s , μ , and \mathcal{E}_w (\log_{10} -scale) is illustrated.

Under a sampling rate of 100 Hz, the highest accuracy ($\approx 1e-7$) is achieved at $\mu \approx 2$. As τ_s decreases, accuracy downgrades, and the value of μ needed for optimal accuracy evaluation increases. A similar pattern is observed for the cases of 10 Hz and 2 Hz (see Fig. 2). Furthermore, the overall accuracy decreases as the sampling rate decreases, while still maintaining a relatively high level of accuracy. These experiments also empirically provide insight into how to set the parameters (τ_s, μ, λ) for the RTM and can be utilized with real data.

The following parameters are used for the rest of this paper: for 2, 10, and 100 Hz sampling rate, we set $(\tau_s, \mu, \lambda) = (25, 0.35, 1e8)$, $(15, 1, 1e8)$, and $(10, 10, 1e8)$, respectively.

2) *Comparisons Among All Methods:* We compare three Koopman-based methods with the OBO. For Koopman-based methods, we choose monomials by setting $P = Q = J = 3$. The RMSE of weights (\mathcal{E}_w) are reported in Table I. In general, KLM and RTM perform better than the other two methods and can still achieve relatively high accuracy as the sampling rate decreases.

B. CTH-RV with Polynomial Parametric Noise

In this scenario, we first compare the performance among all Koopman-based methods. The results are reported in Table II. Note that, in this case, the exact number of monomial functions required is $(P, Q, J) = (3, 3, 2)$. When (P, Q, J) less than this number, the dictionary is insufficient

TABLE I
RMSE OF WEIGHTS (\mathcal{E}_w) FOR CTH-RV

| Approaches | 100 Hz | 10 Hz | 2 Hz |
|------------|-----------------------------|-----------------------------|-----------------------------|
| FDM | $2.07e-3$ | $2.07e-2$ | $8.53e-2$ |
| KLM | $5.76e-6$ | $5.78e-6$ | $5.84e-5$ |
| RTM | $4.96e-7$ | $1.30e-6$ | $4.36e-5$ |
| OBO | $3.46e-3$ | $2.46e-3$ | $5.96e-3$ |

to represent the true function f_θ exactly, resulting in distorted assigned weights and less accuracy. The results demonstrate that the RTM has a much better overall ability to identify parametric noise, regardless of the sampling rate.

The RMSE of weights for the OBO are as follows: $\mathcal{E}_w = 0.041, 0.040$, and 0.041 under 100, 10, and 2 Hz sampling rate. When the true model is perturbed, parameter fitting based on a priori believed model is not advisable.

C. GHR Model with Parametric Noise

We use the last example to examine the ability of Koopman-based methods in identifying non-polynomial f_θ . Additionally, to streamline the examination, we intentionally add polynomial parametric noise to perturb the system, intending to demonstrate that data fitting using the nominal model with OBO does not possess robustness.

In this experiment, we use a 100 Hz sampling rate for all the identification methods. In addition, for Koopman-based methods, in order to better approximate the non-polynomial function f_θ , we opt to use a relatively large number of dictionary functions with $(P, Q, J) = (4, 4, 4)$. For the OBO method, we set the initial guess of parameters to be $(c_0, \delta_0) = (0.8, 0.08)$, which are close enough to the nominal model without noise. The final optimized parameters are $(c, \delta) = (0.6, 0.06)$.

After learning the model, a simulation of v and s is presented in Fig. 3. The preceding vehicle executes a sinusoidal velocity profile with $u(t) = 0.8 \sin(3t)$. It is shown that the FDM and RTM inferred models can reproduce the actual velocity more accurately, while the OBO generates a larger deviation. The KLM fails to generate the same shape of the velocity pattern. It is shown that the FDM drifts away from the actual data for a long-term run, while the RTM reproduces it more accurately. The RMSE of flow for each method is reported in Table III.

VI. CONCLUSION

In this paper, we have developed a modified resolvent-type method (RTM), a Koopman-based data-driven technique, for system identification of ACC. We compared the identification performance with two other Koopman-based methods and one commonly used offline batch optimization for parameter estimation with a priori-believed model. Through simulation experiments, it has been demonstrated that the newly proposed RTM has the best overall performance, considering prediction accuracy under parametric noise.

Though not presented in this paper, utilizing less biased neural network dictionary functions would be a straightforward extension of this study, potentially leading to better

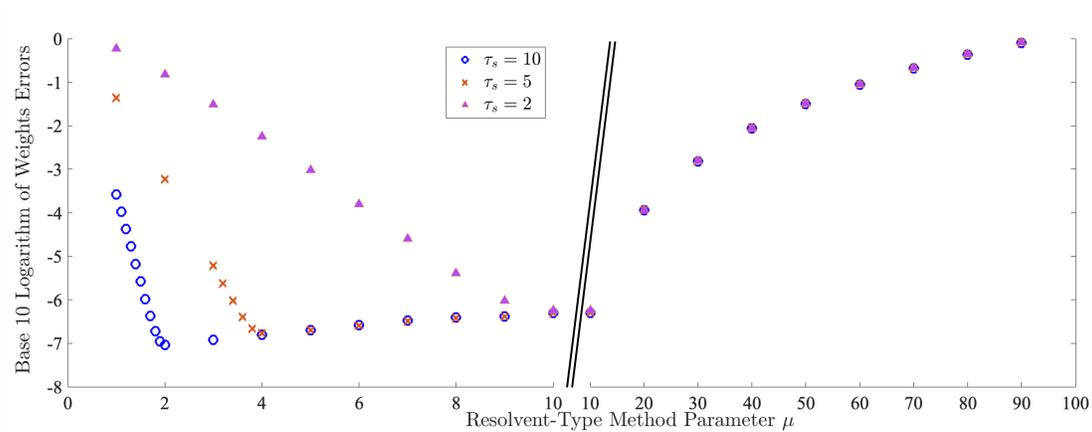


Fig. 1. Empirical Testing of Relations between Parameters for the RTM under 100 Hz Sampling Rate.

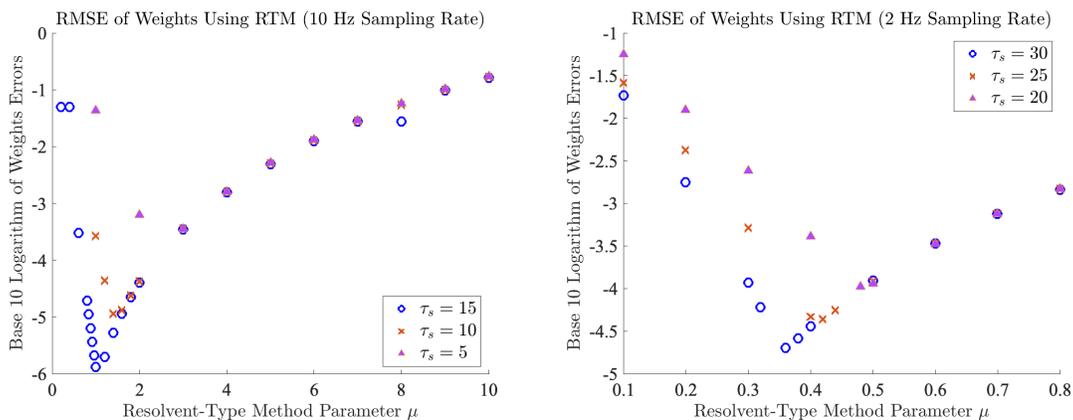


Fig. 2. Empirical Testing of Relations between Parameters for the RTM under 10 and 2 Hz Sampling Rates.

TABLE II
RMSE OF WEIGHTS (\mathcal{E}_w) FOR CTH-RV WITH POLYNOMIAL PARAMETRIC NOISE

| Approaches | Sampling Rate | (P, Q, J) $= (2, 2, 2)$ | (P, Q, J) $= (3, 2, 2)$ | (P, Q, J) $= (3, 3, 2)$ | (P, Q, J) $= (3, 3, 3)$ | (P, Q, J) $= (4, 3, 3)$ | (P, Q, J) $= (4, 4, 3)$ | (P, Q, J) $= (4, 4, 4)$ |
|------------|---------------|------------------------------|------------------------------|------------------------------|------------------------------|------------------------------|------------------------------|------------------------------|
| FDM | 100 Hz | 0.620 | 0.271 | $2.07e-3$ | $3.70e-2$ | 2.19 | 2.54 | 2.93 |
| KLM | 100 Hz | 0.406 | 0.190 | $8.20e-3$ | $2.17e3$ | $6.79e3$ | $1.04e4$ | $1.30e4$ |
| RTM | 100 Hz | 0.615 | 0.275 | $5.69e-7$ | $4.18e-7$ | $4.30e-7$ | $6.72e-7$ | $1.16e-6$ |
| FDM | 10 Hz | 0.624 | 0.278 | $2.07e-2$ | $4.47e-2$ | 2.19 | 2.54 | 2.94 |
| KLM | 10 Hz | 0.614 | 0.267 | $2.70e-4$ | $1.72e2$ | $7.48e2$ | $9.99e2$ | $1.24e3$ |
| RTM | 10 Hz | 0.551 | 0.283 | $1.05e-6$ | $1.32e-6$ | $1.10e-6$ | $1.40e-6$ | $1.76e-6$ |
| FDM | 2 Hz | 0.650 | 0.320 | 0.105 | 0.133 | 2.19 | 2.54 | 2.94 |
| KLM | 2 Hz | 0.590 | 0.258 | $1.18e-4$ | 35.13 | $1.49e2$ | $1.19e2$ | $2.50e2$ |
| RTM | 2 Hz | 0.433 | 0.202 | $1.61e-4$ | $1.63e-4$ | $1.41e-4$ | $1.60e-4$ | $1.82e-4$ |

TABLE III
RMSE OF FLOW USING GHR MODEL

| Approaches | RMSE of Distance Gap | RMSE of Velocity |
|------------|-----------------------------|-----------------------------|
| FDM | $4.24e-2$ | 0.448 |
| KLM | 0.564 | 2.79 |
| RTM | $5.76e-3$ | $1.10e-2$ |
| OBO | 0.11 | 11.69 |

expressibility of more general models. In general, RTM has great potential for identifying slight parametric noise. The comprehensive empirical testing of the tunable parameters

for RTM operator learning can also be applied to real ACC vehicle data, provided the underlying physical laws governing car-following behavior. Utilizing real data to identify acceleration logic for commercial automated vehicles will be pursued as future work. Additional work using this system identification scheme includes addressing filtering problems based on observed data, which provides the potential to identify additive stochastic noise within the model.

REFERENCES

- [1] F. Jiménez, J. E. Naranjo, J. J. Anaya, F. García, A. Ponz, and J. M. Armingol, "Advanced driver assistance system for road environments

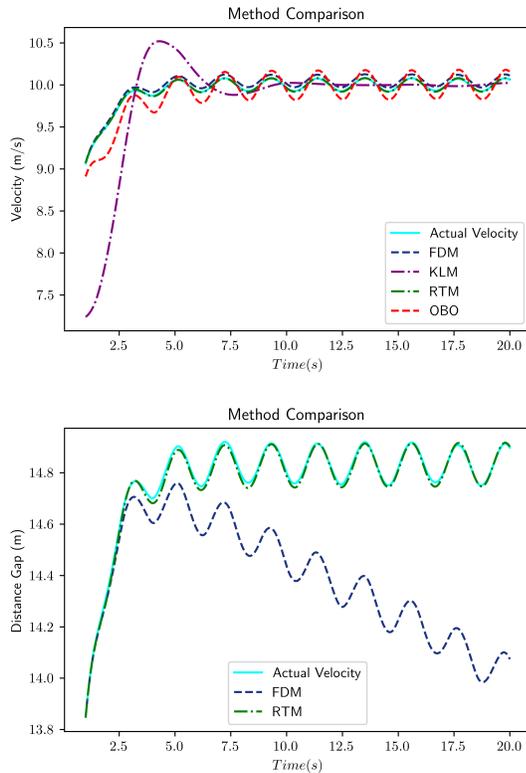


Fig. 3. Comparison among Synthetic Data and Identified ACC Models.

to improve safety and efficiency,” *Transportation Research Procedia*, vol. 14, pp. 2245–2254, 2016.

- [2] Y. Han and S. Ahn, “Stochastic modeling of breakdown at freeway merge bottleneck and traffic control method using connected automated vehicle,” *Transportation Research Part B: Methodological*, vol. 107, pp. 146–166, 2018.
- [3] P. Ioannou, Z. Xu, S. Eckert, D. Clemons, and T. Sieja, “Intelligent cruise control: theory and experiment,” in *Proceedings of 32nd IEEE Conference on Decision and Control*, pp. 1885–1890, IEEE, 1993.
- [4] M. Makridis, K. Mattas, A. Anesiadou, and B. Ciuffo, “Openacc. an open database of car-following experiments to study the properties of commercial acc systems,” *Transportation Research Part C: Emerging Technologies*, vol. 125, p. 103047, 2021.
- [5] V. L. Knoop, M. Wang, I. Wilmink, D. M. Hoedemaeker, M. Maaskant, and E.-J. Van der Meer, “Platoon of sae level-2 automated vehicles on public roads: Setup, traffic interactions, and stability,” *Transportation Research Record*, vol. 2673, no. 9, pp. 311–322, 2019.
- [6] W. J. Schakel, C. M. Gorter, J. C. De Winter, and B. Van Arem, “Driving characteristics and adaptive cruise control? a naturalistic driving study,” *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 2, pp. 17–24, 2017.
- [7] M. Makridis, K. Mattas, B. Ciuffo, F. Re, A. Kriston, F. Minarini, and G. Rognelund, “Empirical study on the properties of adaptive cruise control systems and their impact on traffic flow and string stability,” *Transportation Research Record*, vol. 2674, no. 4, pp. 471–484, 2020.
- [8] G. Gunter, D. Gloudemans, R. E. Stern, S. McQuade, R. Bhadani, M. Bunting, M. L. Delle Monache, R. Lysecky, B. Seibold, J. Sprinkle, et al., “Are commercially implemented adaptive cruise control systems string stable?,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 11, pp. 6992–7003, 2020.
- [9] X. Li, “Trade-off between safety, mobility and stability in automated vehicle following control: An analytical method,” *Transportation research part B: methodological*, vol. 166, pp. 1–18, 2022.
- [10] H. Li and X. Sun, “On the robotic uncertainty of fully autonomous traffic,” *arXiv preprint arXiv:2309.12611*, 2023.
- [11] G. Gunter, C. Janssen, W. Barbour, R. E. Stern, and D. B. Work, “Model-based string stability of adaptive cruise control systems using field data,” *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 1, pp. 90–99, 2019.
- [12] V. Milanés and S. E. Shladover, “Modeling cooperative and autonomous adaptive cruise control dynamic responses using experimental data,” *Transportation Research Part C: Emerging Technologies*, vol. 48, pp. 285–300, 2014.
- [13] S. E. Otto and C. W. Rowley, “Koopman operators for estimation and control of dynamical systems,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, pp. 59–87, 2021.
- [14] J. J. Bramburger and G. Fantuzzi, “Auxiliary functions as Koopman observables: Data-driven analysis of dynamical systems via polynomial optimization,” *Journal of Nonlinear Science*, vol. 34, no. 1, p. 8, 2024.
- [15] A. Mauroy and J. Goncalves, “Koopman-based lifting techniques for nonlinear systems identification,” *IEEE Transactions on Automatic Control*, vol. 65, no. 6, pp. 2550–2565, 2019.
- [16] Y. Meng, R. Zhou, M. Ornik, and J. Liu, “Koopman-based learning of infinitesimal generators without operator logarithm,” *arXiv preprint arXiv:2403.15688*, 2024.
- [17] Y. Xiao, X. Zhang, X. Xu, X. Liu, and J. Liu, “Deep neural networks with koopman operators for modeling and control of autonomous vehicles,” *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 135–146, 2022.
- [18] J. Zhan, Z. Ma, and L. Zhang, “Data-driven modeling and distributed predictive control of mixed vehicle platoons,” *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 572–582, 2022.
- [19] J. S. Kim, Y. S. Quan, and C. C. Chung, “Koopman operator-based model identification and control for automated driving vehicle,” *International Journal of Control, Automation and Systems*, vol. 21, no. 8, pp. 2431–2443, 2023.
- [20] Y. Wang, G. Gunter, M. Nice, M. L. Delle Monache, and D. B. Work, “Online parameter estimation methods for adaptive cruise control systems,” *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 2, pp. 288–298, 2020.
- [21] H. Yao, Q. Li, and X. Li, “A study of relationships in traffic oscillation features based on field experiments,” *Transportation Research Part A: Policy and Practice*, vol. 141, pp. 339–355, 2020.
- [22] G. Orosz and S. P. Shah, “A nonlinear modeling framework for autonomous cruise control,” in *Proceedings of Dynamic Systems and Control Conference*, vol. 45301, pp. 467–471, American Society of Mechanical Engineers, 2012.
- [23] M. Treiber, A. Hennecke, and D. Helbing, “Congested traffic states in empirical observations and microscopic simulations,” *Physical Review E*, vol. 62, no. 2, p. 1805, 2000.
- [24] R. E. Chandler, R. Herman, and E. W. Montroll, “Traffic dynamics: studies in car following,” *Operations Research*, vol. 6, no. 2, pp. 165–184, 1958.
- [25] B. Oksendal, *Stochastic Differential Equations: an Introduction with Applications*. Springer Science & Business Media, 2013.
- [26] A. Nejati, A. Lavaei, S. Soudjani, and M. Zamani, “Data-driven estimation of infinitesimal generators of stochastic systems,” *IFAC-PapersOnLine*, vol. 54, no. 5, pp. 277–282, 2021.
- [27] C. Wang, Y. Meng, S. L. Smith, and J. Liu, “Data-driven learning of safety-critical control with stochastic control barrier functions,” in *Proceedings of 61st IEEE Conference on Decision and Control (CDC)*, pp. 5309–5315, 2022.
- [28] Z. Zeng, Z. Yue, A. Mauroy, J. Goncalves, and Y. Yuan, “A sampling theorem for exact identification of continuous-time nonlinear dynamical systems,” in *Proceedings of 61st IEEE Conference on Decision and Control (CDC)*, pp. 6686–6692, 2022.
- [29] Z. Zeng and Y. Yuan, “A generalized nyquist-shannon sampling theorem using the koopman operator,” *arXiv:2303.01927*, 2023.
- [30] 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, pp. 1307–1346, 2015.
- [31] Y. Meng, R. Zhou, and J. Liu, “Learning regions of attraction in unknown dynamical systems via Zubov-Koopman lifting: Regularities and convergence,” *arXiv preprint arXiv:2311.15119*, 2023.
- [32] G. Gunter, R. Stern, and D. B. Work, “Modeling adaptive cruise control vehicles from experimental data: model comparison,” in *Proceedings of 22nd IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 3049–3054, 2019.