Journal of Machine Learning Research 24 (2023) 1-35 Submitted 4/22; Revised 10/23; Published 10/23

Finite-time Koopman Identifier: A Unified Batch-online Learning Framework for Joint Learning of Koopman Structure and Parameters

Majid Mazouchi MAZOUCHI@MSU.EDU
Department of Mechanical Engineering
Michigan State University
East Lansing, MI, 48863, USA

Subramanya Nageshrao SNAGESHR@FORD.COM
Ford Research and Innovation Center
Ford Motor Company
Palo Alto, CA 94304, USA

Hamidreza Modares MODARESH@MSU.EDU
Department of Mechanical Engineering
Michigan State University
East Lansing, MI, 48863, USA

Editor: Maxim Raginsky

Abstract

In this paper, a unified batch-online learning approach is introduced to learn a linear representation of nonlinear system dynamics using the Koopman operator. The presented system modeling approach leverages a novel incremental Koopman-based update law that regains a mini-collection of samples stored in a memory to minimize not only the instantaneous Koopman operator's identification errors but also the identification errors for the collection of retrieved samples. Discontinuous modifications of gradient flows are presented for the online update law to assure finite-time convergence under easy-to-verify conditions defined on the batch of data. Therefore, this unified online-batch framework allows joint sample- and time-domain analysis to converge the Koopman operator's parameters. More specifically, it is shown that if the collected mini-batch of samples guarantees a rank condition, then the finite-time guarantee in the time domain can be certified, and the settling time depends on the quality of collected samples being reused in the update law. Moreover, the efficiency of the proposed Koopman-based update law is further analyzed by showing that the identification regret in continuous time grows sub-linearly with time. Furthermore, to avoid learning corrupted dynamics due to the selection of an inappropriate set of Koopman observables, a higher-layer meta-learner employs a discrete Bayesian optimization algorithm to obtain the best library of observable functions for the operator. Since finite-time convergence of the Koopman model for each set of observables is guaranteed under a rank condition on stored data, the fitness of each set of observables can be obtained based on the identification error on the stored samples in the proposed framework and even without implementing any controller based on the learned system. Finally, to confirm the effectiveness of the proposed scheme, two simulation examples are presented.

Keywords: Bayesian Optimization, Experience-Replay Technique, Finite-Time System Identifier, Koopman Theory

1. Introduction

Koopman operator theory provides an elegant framework to construct (infinite-dimensional) linear representations for nonlinear dynamical systems from input-output data and, as a result, a solid framework to use established model-based linear control methods. More specifically, Koopman operator theory relies on a nonlinear transformation to lift the original state-space to a space of functions, referred to as observable functions, under which their flow along trajectories of the system is described by a linear map with generally an infinite-dimensional state space. Finite-dimensional approximations of the Koopman operator from data has been widely considered (Budišić et al., 2012; Bakker et al., 2019), mainly using Dynamic Mode Decomposition (DMD) (Tu et al., 2013) and Extended Dynamic Decomposition (EDMD) (Williams et al., 2015). These approaches are mainly presented for discrete-time (DT) systems. Continuous-time (CT) formulations for system dynamics, however, can be methodologically preferred over DT formulations for the control of many systems, such as safety-critical systems. For example, control-theoretic analyses such as Nagumo's theorem (which is fundamental for characterizing the forward invariance of safe sets) are not applicable for general DT systems (Maghenem and Sanfelice, 2018). Despite its advantages, CT Koopman representation is scarce (Kaiser et al., 2021).

Koopman operator-based identification techniques mainly use a large batch of data samples as a full training data set to learn offline the Koopman model. On the one hand, since the size of the required batch of data set is generally huge for high-dimensional Koopman-based representations, training on the full data at once can incur a high cost of computing resources. Moreover, a majority of practical control systems are not able to wait until gathering a large and rich dataset and learning the accurate model to make decisions, and they need to make decisions by learning the model on the fly by using only available data samples which incrementally become available over time (Ayoobi et al., 2021). On the other hand, even though online or incremental learning allows fast and computationally cheap (at least sub-linear time and space complexity) learning algorithms, it can only provide generalization guarantees (i.e., learning the exact system parameters) if a persistency of excitation condition on the richness of data is satisfied, which is hard to guarantee when samples are collected incrementally over time.

Another factor that has a huge impact on the performance of the Koopman-based identification, besides its learning rule, is the selection of observable functions. EDMD can embody Koopman observables (i.e., eigenfunctions) that are nonlinear functions of the state (Generally sets of radial basis functions (Williams et al., 2016; Korda and Mezić, 2018) and polynomials (Williams et al., 2015; Proctor et al., 2018)). Even though a large size of observables might lead to better accuracy, it is desired to learn reduced-order Koopman-based models that are amenable to control-system approaches (Williams et al., 2015; Kutz et al., 2016). Moreover, some eigenfunctions can be distorted when projected into a finite-dimensional measurement subspace, causing undesirable performance degradation in the learning process. Instead of choosing a dictionary of observable functions a priori, another approach is to use neural networks to learn the observable functions from the data (Yeung et al., 2019; Li et al., 2017). These methods, however, typically separate learning the Koopman observables and its model parameters and are based on the availability of a huge batch of i.i.d samples.

In sharp contrast to the existing offline Koopman operator identifier, a novel data-driven bilevel learning algorithm is presented to jointly learn the efficient set of observables and the corresponding Koopman parameters. The proposed learning scheme is composed of two layers: a lower-layer parameter learning and a higher-layer structure learning. In the lower layer (so-called base learner), a unified batch-online learning framework is presented for learning the system Koopman operators' parameters through an experience-replay gradient descent (ERGD) (Modares et al., 2013; Chowdhary et al., 2013; Yang et al., 2014; Liu et al., 2014; Chowdhary and Johnson, 2010; Kamalapurkar et al., 2017; Vamvoudakis et al., 2015; Zhao et al., 2015; Zhang et al., 2016; Cao et al., 2017; Jha et al., 2019; Parikh et al., 2019; Tatari et al., 2017, 2018; Walters et al., 2018; Yang and He, 2019; Jiang et al., 2019; Yang et al., 2020; Vahidi-Moghaddam et al., 2020) update law that not only minimizes the current Koopman operator's identification errors online but also the identification errors for a batch of past samples collected in a history stack. To achieve finite-time convergence guarantee (Lu et al., 2016; He and Song, 2017; Romero and Benosman, 2021; Xia et al., 2019; Song and Wei, 2019; Romero and Benosman, 2020b; Zhao and Liu, 2020; Romero and Benosman, 2020a; Tatari et al., 2021) of the presented Koopman learning approach with a prescribed settling time, a discontinuous ERGD update law is presented and Filippov differential inclusions along with finite-time Lyapunov stability theory are leveraged to analyze these discontinuous flows and the performance of the Koopman learned model. It is shown that the settling time and generalization capability of the learned model depend on the maximum eigenvalue of the matrix formed by the mini-batch of collected data that are being continually reused during online learning. The efficiency of the proposed Koopman-based update law is further analyzed by showing that the identification regret in continuous time grows sub-linearly with time.

In the higher layer (so-called meta-learner), a discrete Bayesian optimization algorithm (Brochu et al., 2010) is used to learn the best library of observable functions. More specifically, adapting the lifted state (library of observable functions) of the base learner is performed in a higher layer to develop a model that is of high fidelity but also has as low dimensionality as possible. Since finite-time convergence of the Koopman model for each set of observables is guaranteed under a rank condition on stored data, the fitness of each set of observables can be obtained based on the identification error on the stored samples in the proposed framework and even without implementing any controller based on the learned system. Finally, two simulation examples are given to verify the effectiveness of the proposed scheme.

The rest of this paper is organized as follows: Some preliminaries are provided in Section 2. The problem formulation is provided in Section 3. The main theoretical results are provided and proved in Section 4. Section 5 presents some simulation results to verify the performance of the proposed control scheme. Finally, Section 6 provides the conclusion for the paper.

Notation. Throughout the paper, $\mathbb{R}$ and $\mathbb{R}^+$ represent the set of real numbers and positive real numbers, respectively. $\mathbb{R}^n$ and $\mathbb{R}^{n\times m}$ denote n-dimensional Euclidean space and $(n\times m)$-dimensional real matrices space. I denotes the identity matrix of the proper dimension. $\| \, . \, \|$ denotes the Euclidean norm of a vector or the induced 2-norm of a matrix. We use $^\top$ to denote the transpose operator. $A>0$ denotes that the matrix A is positive definite. We use $\lambda_{min}(A)$ (resp. $\lambda_{\max}(A)$) to denote the minimum (resp. maximum) eigenvalues of the square matrix A. The Kronecker product of two matrices A and B is denoted as $A\otimes B$. For a function f, $f\geq 0$ denotes that f is positive semi-definite. $\mathscr{K}: S_1 \rightrightarrows S_2$ denotes a set-valued map $\mathscr{K}$ for given sets S_1 and S_2 such that $s\in S_1 \Rightarrow \mathscr{K}(s)\subseteq S_2$.

2. Preliminaries

Some basic preliminaries are briefly introduced in this section. Consider a vector differential equation described in the form of

$$\begin{aligned} \dot{x}(t) &= \mathscr{I}(x(t)) \\ x(0) &= x_0 \end{aligned} \tag{1}$$

where $\mathscr{I} : \mathbb{R}^n \mapsto \mathbb{R}^n$ may be discontinuous, but Lebesgue measurable inside an open region $\mathscr{R} \subset \mathbb{R}^n$, $\mathscr{R} \neq \emptyset$, and essentially locally uniformly bounded (i.e., $\mathscr{I}$ is bounded a.e. (almost everywhere) inside a bounded neighborhood of the point $x \in \mathbb{R}^n$, $\forall x$). Note that the discontinuous system (1) can be seen as a Filippov differential inclusion, and in this sense, the following definition formally describes a solution of (1).

Definition 1 *(Filippov, 2013; Paden and Sastry, 1987) A Filippov solution of the vector differential equation* (1) *on an interval* $[0, \tau)$ *is defined to be an absolutely continuous function* $x : [0, \tau) \mapsto \mathbb{R}^n$ *with* $0 < \tau \leq \infty$ *such that* $x(0) = x_0$ *and*

$$\dot{x}(t) \in \mathbf{K}[\mathscr{I}](x(t)), \tag{2}$$

holds for almost all $t \in [0, \tau)$*, where the Filippov set-valued map* $\mathbf{K}[\mathscr{I}] : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ *is described by*

$$\mathbf{K}[\mathscr{I}](x) \equiv \bigcap_{\delta > 0} \bigcap_{S:\ \mu S = 0} \overline{\text{co}} \{\mathscr{I}(\mathscr{B}(x, \delta) \setminus S)\}, \tag{3}$$

where $\overline{\text{co}}$ *denotes the convex closure,* $\mathscr{B}(x, \delta)$ *denotes the open ball of radius* δ *with the center* x*,* μ *denotes the Lebesgue measure, , and* $\bigcap_{S:\ \mu S = 0}$ *denotes the intersection over all sets* S *where* S *runs over all zero-measure sets of* $\mathbb{R}^n$*. Moreover, the Filippov solution* x *is called maximal if it cannot be extended (Cortés, 2008).*

Definition 2 *(Thieme, 2019) A set-valued function* $\mathbf{K}[\mathscr{I}] : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ *is called upper semi-continuous at the point* x *if, for any* $\varepsilon > 0$*, there exists some* $\delta > 0$ *such that* $\mathbf{K}[\mathscr{I}](\mathscr{B}(x, \delta))$ *is a subset of an open* ε*-neighborhood of* $\mathbf{K}[\mathscr{I}](x)$*. If* $\mathbf{K}[\mathscr{I}]$ *is upper semi-continuous at each point* $x \in \mathbb{R}^n$*, then it is said to be upper semi-continuous.*

Condition 1 $\mathbf{K}[\mathscr{I}] : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ *is upper semi-continuous and has compact, nonempty, and convex values.*

Lemma 1 *(Paden and Sastry, 1987) Consider Filippov differential inclusion* (2)*. There exists a zero-level set* $\mathscr{N}_{\mathscr{I}}$*, i.e.,* $\mu \mathscr{N}_{\mathscr{I}} = 0$*, such that* (3) *can be computed as*

$$\mathbf{K}[\mathscr{I}](x) = \overline{\text{co}} \{\lim \mathscr{I}(x_i) \,|\, x_i \to x, x_i \notin S \cup \mathscr{N}_{\mathscr{I}}\}, \tag{4}$$

where S *is any Lebesgue-zero measure set, i.e.,* $\mu S = 0$*. In particular,* $\mathbf{K}[\mathscr{I}](x) = \{\mathscr{I}(x)\}$ *provided that* $\mathscr{I}$ *be continuous at the fixed point* x*.*

Now, let us introduce generalized derivatives and gradients to deal with non-smooth Lyapunov functions.

Definition 3 *Let $V(\cdot):\mathbb{R}^n \mapsto \mathbb{R}$ be a locally Lipschitz function. Then, the Clarke upper generalized derivative of $V(.)$ at x in the direction of v (also called generalized directional derivative) is defined as*

$$V^{\circ}(x;v) := \limsup_{x'\to x\, t\to 0^+} \frac{V(x'+tv)-V(x')}{t}. \tag{5}$$

Definition 4 *$V(\cdot):\mathbb{R}^n \mapsto \mathbb{R}$ is called regular if for all v:*
1) the usual one-sided directional derivative $V(\cdot)$

$$V'(x;v) := \lim_{t\to 0} \frac{V(x+tv)-V(x)}{t}, \tag{6}$$

exists for all v;
2) $V^{\circ}(x;v) = V'(x;v)$.

Definition 5 *The Clarke generalized gradient of $V(\cdot)$ at x is the set*

$$\partial V(x) = \mathrm{co}\left\{ \lim_{i\to+\infty} \nabla V(x_i) : x_i \to x,\ \ x_i \notin S \cup \mathcal{N}_V \right\}, \tag{7}$$

where co *denotes convex hull and $S \subset \mathbb{R}^n$ is any zero-measure set and $\mathcal{N}_V$ the zero-measure set over which V is not differentiable.*

Lemma 2 *(Bacciotti and Ceragioli, 1999) Let $V(\cdot)$ be a locally Lipschitz and regular function. Then,*
1)

$$\partial V(x) = \{\xi \in \mathbb{R}^n : V^{o}(x,v) \geq \xi \cdot v \ \ \forall v \in \mathbb{R}^n\}; \tag{8}$$

2)

$$V^{\circ}(x;v) = \max\{\langle \xi, v\rangle = \xi.v \mid \xi \in \partial V(x)\}; \tag{9}$$

3)

$$\mathbf{K}[\nabla V](x) = \partial V(x). \tag{10}$$

Condition 2 *$V(\cdot)$ is a locally Lipscthiz continuous and regular Lyapunov function* [1] *for the system* (1).

Definition 6 *(Bacciotti and Ceragioli, 1999) Given a locally Lipschitz function $V(.):\mathbb{R}^n \mapsto \mathbb{R}$ and a set-valued map $\mathbf{K}[\mathscr{I}](x)$, the set-valued time derivative of V with respect to the differential inclusion* (2) *is defined as*

$$\dot{V}(x) \triangleq \left\{ a \in \mathbb{R} : \exists v \in \mathbf{K}[\mathscr{I}](x)\ s.t.\ a = p^{\top}v, \forall p \in \partial V(x) \right\}, \tag{11}$$

for each $x \in \mathbb{R}^n$.

[1] In other words, it is a scalar function that is continuous, has continuous first derivatives, is strictly positive, except the origin, and has a non-positive time derivative $\dot{V}$.

The following proposition provides a sufficient condition for finite-time convergence of differential inclusions.

Proposition 1 *(Romero and Benosman, 2020b) Consider the system* (1) *and let Condition 1 be satisfied for a set-valued map* $\mathbf{K} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$*. Let* $\mathscr{R} \subseteq \mathbb{R}^n$ *be a positively invariant and open neighborhood of* x^* *and* $V : \mathscr{R} \mapsto \mathbb{R}$ *be positive definite* [2] *satisfying Condition 2. Consider the constants* $c_1 > 0$ *and* $c_2 < 1$ *exist such that*

$$\sup \dot{V}(x) \leq -c_1 V(x)^{c_2}, \tag{12}$$

a.e. in $x \in \mathscr{R}$*. Then,* (1) *converges to* x^* *within finite time and a settling-time which is upper bounded by*

$$t^\star \leq \frac{V(x(0))^{1-c_2}}{c_1(1-c_2)}. \tag{13}$$

3. Problem Formulation

The finite-time identification problem is formulated in this section for a continuous-time nonlinear system with dynamics given by

$$\dot{x} = F(x) + G(x)u(t),\ F(0) = 0, \tag{14}$$

where $x \in \mathbb{R}^n$ and $u(t) \in \mathbb{R}^m$ denote the state vector and the control input, respectively. Furthermore, $F(x) \in \mathbb{R}^n$ and $G(x) \in \mathbb{R}^{n \times m}$ denote the drift dynamics and the input dynamics of the system, respectively.

Assumption 1 *The functions* $F(x)$ *and* $G(x)$ *are smooth. System states are available for measurement. However, the derivatives of the states of the system may not be available for measurement.*

The key step in obtaining a linear structured version of a nonlinear dynamical system (14) is a lifting of the state-space (Mauroy and Goncalves, 2019; Drmač et al., 2021) to a higher-dimensional space, where its evolution is approximately linear. Let us consider real-valued observables elements ξ_i, which are elements of an infinite-dimensional Hilbert space. Generally, the Hilbert space is provided by the Lebesgue-square integrable functions. In this study, it is our aim to identify the nonlinear vector fields $F(.)$ and $G(.)$ based on the stream of data generated by the control dynamical system and find a continuous-time dynamical system of the form (Huang et al., 2020; Brunton et al., 2016b; Korda and Mezić, 2018)

$$\frac{d}{dt}\xi(t) = A\xi(t) + B\Psi(u(t)), \tag{15}$$

where the term $B\Psi(.)$ captures how the observable $\xi(x)$ is modified by the control policy $u(.)$, e.g., $\Psi(u(t))$ is chosen as $\xi(x)u(t)$ in (Korda and Mezić, 2018).

[2] In other words, there exists some open neighborhood $\mathscr{R}$ of $x^\star$ such that V is defined in $\mathscr{R}$ and satisfies $V(x^\star) = 0$ and $V(x) > 0$ for every $x \in \mathscr{R} \backslash \{x^\star\}$.

Definition 7 *The finite set of all available observables, denoted by $\mathscr{C}$, is defined as $\mathscr{C} := \{\xi_1(x), \xi_2(x), \ldots, \xi_N(x)\}$. A library set $\mathscr{L}(\theta)$, where $\mathscr{L}(\theta) \subseteq \mathscr{C}$, is defined as $\mathscr{L}(\theta) := \{\xi_k : k \in \theta, \xi_k \in \mathscr{C}\}$ where $(\theta, \leq) \subseteq \{1, ..., N\}$ is a non-empty finite partially ordered set with $n_{\xi_\theta} := |\theta|$ distinct elements. The library vector corresponding to θ, i.e., $\mathscr{L}(\theta)$, is defined as $\xi_\theta(t) := [\xi_{\theta(1)}(x(t)), \cdots, \xi_{\theta(n_{\xi_\theta})}(x(t))]^\top$.*

A library vector $\xi_\theta(t)$ corresponding to the ordered set of θ, i.e., $\mathscr{L}(\theta)$, which is a subset of the finite set of all available observables $\mathscr{C}$, will be used in the sequel. It is worth noting that the library vector $\xi_\theta(t)$ will be selected by the meta-learner layer (as detailed in Subsection 4.3) to lift the system from a state-space to function space of observables with the benefit of reducing the dimension of the Koopman operator and avoiding selecting corrupted dynamics. Note that the dimension of the library vector $\xi_\theta(t)$ is the same as the size of set $\mathscr{L}(\theta)$, i.e., $\xi_\theta(t) \in \mathbb{R}^{n_{\xi_\theta}}$. In the rest of the paper, for simplicity of notation, we have used ordered set θ instead of the library of observables $\mathscr{L}(\theta)$.

Now, the infinite-dimensional Koopman operator is approximated with a finite-dimensional matrix A^*_θ and B^*_θ, and the linear representation continuous-time dynamics are given by

$$\frac{d}{dt}\xi_\theta(t) = A^*_\theta \xi_\theta(t) + B^*_\theta \Psi(u(t)) + \mathscr{P}_\theta(\xi_\theta(t)), \tag{16}$$

where $A^*_\theta \in \mathbb{R}^{n_{\xi_\theta} \times n_{\xi_\theta}}$, $B^*_\theta \in \mathbb{R}^{n_{\xi_\theta} \times m_\xi}$, $\Psi(.) \in \mathbb{R}^{m_\xi}$, and $\xi_\theta(t) \in \mathbb{R}^{n_{\xi_\theta}}$ denotes the nonlinear transformation of the state x through a vector-valued observable $\xi_\theta(t)$. The term $B^*_\theta \Psi(.)$ captures how the observable $\xi_\theta(t)$ is modified by the control policy $u(t)$. Here, we assumed that $\Psi(.)$ is known. Note that this assumption is standard and realistic and it is about adding available knowledge (e.g., in the form of basis functions) to the design procedure (See (Korda and Mezić, 2018; Brunton et al., 2016a)). Moreover, $\mathscr{P}_\theta(.) : \mathbb{R}^{n_{\xi_\theta}} \mapsto \mathbb{R}^{n_{\xi_\theta}}$ is a bounded continuous approximation error term that depends on the goodness of the chosen library of observable functions. In the rest of the paper, for simplicity of notation, we have used ξ_θ instead of $\xi_\theta(t)$, i.e., t is dropped, when it is clear from the context.

Remark 1 *Note that the dimension of the system* (16) *scales with the number of observables, and it is desired to use a few dominant observables associated with persistent dynamics to reduce the size of the transformed system, make it amenable to control design methods. If the observables span contains $F(.)$, $G(.)$, and x, then the model* (16) *may be well represented. Moreover, despite the fact that, based on Koopman's theory, infinite-dimensional lifting theoretically allows the representation of any nonlinear dynamics as linear operators, in practice, one needs to approximate this operator with finite-dimensional operators mainly due to computational feasibility, traceability, and data limitations. To this end, $\mathscr{P}_\theta(.)$ represents the bounded approximation error caused by using approximated finite-dimensional matrix A^*_θ and B^*_θ, which depends on the dimension of ξ_θ and the goodness of the chosen lifted states. Additionally, it should be noted that the assumption of the bounded $\mathscr{P}_\theta(.)$ is a standard in the literature based on the universal approximator characteristics (Tao, 2003).*

The identifier for the unknown Koopman operators is defined as

$$\frac{d}{dt}\hat{\xi}_\theta(t) = \hat{A}_\theta \xi_\theta(t) + \hat{B}_\theta \Psi(u(t)), \tag{17}$$

where $\hat{A}_\theta(t)$ and $\hat{B}_\theta(t)$ denote the identified matrices A_θ^* and B_θ^* at the time of t. Now, one can write the identification errors as

$$\begin{cases} e_{\hat{A}_\theta}(t) = \hat{A}_\theta(t) - A_\theta^* \\ e_{\hat{B}_\theta}(t) = \hat{B}_\theta(t) - B_\theta^* \\ e_{\hat{\xi}_\theta}(t) = \hat{\xi}_\theta(t) - \xi_\theta(t) \end{cases} \tag{18}$$

The objective is now to design a unified batch-online update law for the system identifier (17) to make the identification errors converge to zero, i.e., $\|e_{\hat{\xi}_\theta}\| \to 0$, $\|e_{\hat{A}_\theta}\| \to 0$, and $\|e_{\hat{B}_\theta}\| \to 0$, in finite time for the case where there is no approximation error and the identification errors remain uniformly ultimately bounded (UUB) for the case where there is bounded approximation error. These objectives are formally stated in Problems 1 and 2 in the sequel. We first consider the case with no approximation error term, i.e., $\mathscr{P}_\theta(.) = 0$. In the subsequent sections, then, the presence of the approximation error term will be analyzed. Moreover, the selection of an appropriate Koopman observables set that achieves the minimum approximation error and avoids corruption dynamics is performed through meta-learning later in the subsequent sections.

Problem 1 (Batch-online Koopman finite-time identifier when there is no approximation error). Let the vector-valued observable ξ_θ be fixed and result in no approximation error. Let exist a mini-batch of samples given by $\{\xi_\theta(t_1), \ldots, \xi_\theta(t_p)\}$. Consider the system (16) along with the identifier (17). Develop a unified batch-online update law to make $\|e_{\hat{\xi}_\theta}\| \to 0$, $\|e_{\hat{A}_\theta}\| \to 0$, and $\|e_{\hat{B}_\theta}\| \to 0$, in finite time, under an easy-to-verify condition on mini-batch of collected data samples.

Letting the approximation error be zero, rewrite the system (16) as

$$\frac{d}{dt}\xi_\theta(t) = \Sigma_\theta^{*\top} \mathscr{Z}_\theta(\xi_\theta, u(t)), \tag{19}$$

where $\Sigma_\theta^{*\top} = [A_\theta^*, B_\theta^*] \in \mathbb{R}^{n_{\xi_\theta} \times (n_{\xi_\theta} + m_{\xi_\theta})}$ is the unknown matrix, and $\mathscr{Z}_\theta(\xi_\theta, u(t)) = [\xi_\theta^\top(t), \Psi^\top(u(t))]^\top \in \mathbb{R}^{(n_{\xi_\theta} + m_\xi)}$.

To eliminate the need to measure state derivatives as is required by existing system identification methods, the system model is formulated as a filtered regressor as follows.

Proposition 2 *Consider the system* (16)*,* (19)*. This system can be expressed as the filtered form*

$$\begin{cases} \xi_\theta(t) = \Sigma_\theta^* h_\theta(t) + a l_\theta(t) + e^{-aI_{n_{\xi_\theta}} t} \xi_\theta(0) \\ \dot{h}_\theta(t) = -a h_\theta(t) + \mathscr{Z}_\theta(\xi_\theta, u(t)), \\ \dot{l}_\theta(t) = -a l_\theta(t) + \xi_\theta, \end{cases} \tag{20}$$

$\forall a > 0$ with $l_\theta(t) = 0$ and $h_\theta(t) = 0$, where $\xi_\theta(0)$ denotes the initial state of (19) *and $h_\theta(t) \in \mathbb{R}^{(n_{\xi_\theta} + m_\xi)}$ and $l_\theta(t) \in \mathbb{R}^{n_{\xi_\theta}}$ denote the filtered regressor form of $\mathscr{Z}_\theta(\xi_\theta, u)$ and ξ_θ, respectively.*

Proof. The proof is similar to that of Lemma 1 in (Modares et al., 2013) and it is provided here for the sake of completeness. By adding and subtracting $a\xi_\theta$ from the right-hand side of (19), we get

$$\frac{d}{dt}\xi_\theta(t) = -a\xi_\theta(t) + \Sigma_\theta^{*\top} \mathscr{Z}_\theta(\xi_\theta, u(t)) + a\xi_\theta(t). \tag{21}$$

One can see that (21) has the following solution

$$\xi_\theta(t) = \Sigma_\theta^{*\top} \int_0^t e^{-aI_{n_{\xi_\theta}}(t-\tau)} \mathscr{Z}_\theta(\xi_\theta(\tau), u(\tau))d\tau + a\int_0^t e^{-aI_{n_{\xi_\theta}}(t-\tau)} \xi_\theta(\tau)d\tau + e^{-aI_{n_{\xi_\theta}}t} \xi_\theta(0). \quad (22)$$

Define

$$h_\theta(t) := \int_0^t e^{-aI_{n_{\xi_\theta}}(t-\tau)} \mathscr{Z}_\theta(\xi_\theta(\tau), u(\tau))d\tau, \quad (23)$$

$$l_\theta(t) := \int_0^t e^{-aI_{n_{\xi_\theta}}(t-\tau)} \xi_\theta(\tau)d\tau. \quad (24)$$

Using (22)-(24), we have

$$\xi_\theta(t) = \Sigma_\theta^* h_\theta(t) + al_\theta(t) + e^{-aI_{n_{\xi_\theta}}t} \xi_\theta(0), \quad (25)$$

which is the first equation in (20). The second and third equations in (20) can be obtained by taking the derivatives of (23) and (24). This completes the proof. □

Taking (20) and dividing its both sides by the normalizing signal $n_{s_\theta} = 1 + h_\theta{}^\top(t)h_\theta(t) + l_\theta{}^\top(t)l_\theta(t)$, one has

$$\bar{\xi}_\theta(t) = \Sigma_\theta^{*\top} \bar{h}_\theta(t) + a\bar{l}_\theta(t) + e^{-aI_{n_{\xi_\theta}}t} \bar{\xi}_\theta(0), \quad (26)$$

where $\bar{\xi}_\theta = \xi_\theta / n_{s_\theta}$, $\bar{\xi}_\theta(0) = \xi_\theta(0)/n_{s_\theta}$, $\bar{h}_\theta(t) = h_\theta(t)/n_{s_\theta}$, and $\bar{l}_\theta(t) = l_\theta(t)/n_{s_\theta}$ are the normalized forms of ξ_θ, $h_\theta(t)$, and $l_\theta(t)$, respectively.

Using (26), and Proposition 2, the identifier's state (17) becomes

$$\hat{\bar{\xi}}_\theta(t) = \hat{\Sigma}_\theta^\top(t)\bar{h}_\theta(t) + a\bar{l}_\theta(t) + e^{-aI_{n_{\xi_\theta}}t} \bar{\xi}_\theta(0), \quad (27)$$

where $\hat{\Sigma}_\theta^\top(t) := [\hat{A}_\theta(t), \hat{B}_\theta(t)] \in \mathbb{R}^{n_{\xi_\theta} \times (n_{\xi_\theta} + m_\xi)}$. The normalized version of identification error, i.e., $\bar{e}_{\xi_\theta} = e_{\xi_\theta}/n_{s_\theta}$, is defined as

$$\bar{e}_{\xi_\theta}(t) = \hat{\bar{\xi}}_\theta(t) - \bar{\xi}_\theta(t) = \tilde{\Sigma}_\theta^\top \bar{h}_\theta(t), \quad (28)$$

where $\tilde{\Sigma}_\theta^\top(t) = \hat{\Sigma}_\theta^\top(t) - \Sigma_\theta^{*\top}$ describes the Koopman operators' identification errors. One can see that $\bar{e}_{\xi_\theta}$ is measurable at each time since it is a function of the observables and states of the system are assumed to be measurable. Furthermore, $\bar{e}_{\xi_\theta}$ linearly relates to the state of the filtered regressor and identification errors of Koopman operators. Using this formulation, as we see later, one can reuse the recorded data in the update law without having to compute the derivatives of the system states.

Now, (28) can be rewritten as

$$\bar{e}_{\xi_\theta}(t) = (\bar{h}_\theta^\top(t) \otimes I_{n_{\xi_\theta}})^\top \tilde{\Sigma}_\theta^{vec}(t), \quad (29)$$

where $(\bar{h}_\theta^\top(t) \otimes I_{n_{\xi_\theta}}) \in \mathbb{R}^{n_{\xi_\theta}(n_{\xi_\theta}+m_\xi) \times n_{\xi_\theta}}$ and $\tilde{\Sigma}_\theta^{vec} \in \mathbb{R}^{n_{\xi_\theta}(n_{\xi_\theta}+m_\xi)}$.

Remark 2 *Standard batch learning practice for parameter convergence of the Koopman identifier (i.e., for its generalization guarantees) (Han et al., 2020; Netto and Mili, 2018) leverages statistical learning theory to provide probably approximately correct (PAC) sample complexity bounds on the learned model. PAC analysis, however, requires a huge number of i.i.d samples to guarantee generalization, which depends on the VC-dimension (Vapnik, 2013; Blockeel et al., 2013) of the search space. On the other hand, online learning algorithms guarantee parameter convergence (usually asymptotic guarantees in time) under restrictive and hard-to-verify persistence of excitation conditions. It is critical to design learning algorithms that bring the best of both worlds together and provide finite-time guarantees under easy-to-verify conditions (rather than i.i.d conditions) on samples that depend only on the dimension of the search space and not its VC dimension.*

In this paper, we store a mini-batch of past samples in a history stack and retrieve them during online learning using the experience replay (also known as concurrent learning) technique (Chowdhary and Johnson, 2010; Modares et al., 2013). This technique needs to collect past data in the history stack as

$$\mathscr{S}_\theta = [\bar{h}_\theta(t_1), ..., \bar{h}_\theta(t_p)], \tag{30}$$

p is the number of collected data points that are stacked in the history stack, and $t_1, ..., t_p$ are their associated recorded times. The error of identification for the j-th collected data point is calculated as follows

$$\begin{aligned}\bar{e}_{\xi_\theta}(t,t_j) =& \hat{\bar{\xi}}_\theta(t,t_j) - \bar{\xi}_\theta(t_j)\\ =& \tilde{\Sigma}_\theta^\top(t)\,\bar{h}_\theta(t_j),\\ =& (\bar{h}_\theta^\top(t_j) \otimes I_{n_{\xi_\theta}})\,\tilde{\Sigma}_\theta^{vec}(t),\end{aligned} \tag{31}$$

for $j = 1, ..., p$, where $\bar{\xi}_\theta(t_j)$ denotes the normalized form of the state at t_j, $\hat{\bar{\xi}}_\theta(t,t_j)$ denotes identifier's state at t_j defined as

$$\hat{\bar{\xi}}_\theta(t,t_j) := \hat{\Sigma}_\theta^\top(t)\bar{h}_\theta(t_j) + a\bar{l}_\theta(t_j) + e^{-aI_{n_{\xi_\theta}} t_j}\,\bar{\xi}_\theta(0), \tag{32}$$

and $\bar{e}_{\xi_\theta}(t,t_j)$ is the error of identification at t_j. Furthermore, $\tilde{\Sigma}_\theta^\top(t)$ is the identification error of Koopman operators at the present time.

Condition 3 *The stacked data $\mathscr{S}_\theta$ at least consists of numbers of linearly independent elements equal to the dimension of the basis function $h_\theta(t)$. That is, $\sum_{j=1}^{p} \bar{h}_\theta(t_j)\,\bar{h}^\top(t_j) \geq d_\theta I_{n_{\xi_\theta}+m_\xi}$ for some $d_\theta \in \mathbb{R}^+$.*

4. Main result

The hierarchical learning architecture given in Fig. 1 is adopted to deal with the problem at hand, which consists of:

- A base learner (finite-time identifier), which is used to find the Koopman operator $\hat{A}_\theta(t)$ and $\hat{B}_\theta(t)$ corresponding to a library of the observable function θ.
- A meta-learner to learn the best library of observables θ that achieves a minimum approximation error based on the available collected data set.

4.1 Base Learner: Batch-online Finite-time Koopman Identifier with no Approximation Error

It is well known that solution trajectories of the locally Lipschitz continuous systems converge no faster than exponentially to equilibrium points, i.e., such systems at most can only have asymptotic convergence rates. Non-smooth or non-Lipschitz continuous systems, however, are able to have a finite-time convergence property. This emphasizes that enjoying the finite-time convergence property in continuous-time systems is feasible only by introducing update law with discontinuous or non-Lipschitz dynamics based on differential inclusions instead of ODEs. The following theorem provides a unified novel batch-online learning using a novel discontinuous flow of gradients to guarantee finite-time convergence for the Koopman identifier when there is no approximation error and the observables are fixed.

Theorem 1 *Consider the systems* (29) *and* (31). *Let Condition 3 and Assumption 1 hold. Then, the unknown parameter vector* $\hat{\Sigma}^{vec}_{\theta}(t)$ *converges to its true* Σ^{vec*}_{θ} *in finite time by using any maximal Filippov solution to the following discontinuous differential inclusion update law*

$$\dot{\hat{\Sigma}}^{vec}_{\theta}(t) = \dot{\tilde{\Sigma}}^{vec}_{\theta}(t) \in K_\theta = \mathbf{K}[\mathscr{I}_\theta] : \mathbb{R}^{n_{\xi_\theta}(n_{\xi_\theta}+m_\xi)} \rightrightarrows \mathbb{R}^{n_{\xi_\theta}(n_{\xi_\theta}+m_\xi)}, \tag{33}$$

with

$$\mathscr{I}_\theta = -\alpha_\theta \frac{\|\mathbf{H}_\theta(t)\|}{(\mathbf{H}_\theta(t))^\top} \frac{[\mathbf{A}_\theta]^r \mathbf{H}_\theta(t)}{[\mathbf{A}_\theta]^{r+1} \mathbf{H}_\theta(t)}, \tag{34}$$

where

$$\mathbf{H}_\theta(t) := \mathscr{H}_\theta(t) \bar{e}_{\xi_\theta}(t) + \sum_{j=1}^{p_\theta} \mathscr{H}_\theta(t_j) \bar{e}_{\xi_\theta}(t, t_j), \tag{35}$$

and

$$\begin{aligned} \mathscr{H}_\theta(t) &:= \bar{h}_\theta(t) \otimes I_{n_{\xi_\theta}}, \\ \mathscr{H}_\theta(t_j) &:= \bar{h}_\theta(t_j) \otimes I_{n_{\xi_\theta}}, \end{aligned} \tag{36}$$

$$\mathbf{A}_\theta := \mathscr{H}_\theta(t) \mathscr{H}_\theta{}^\top(t) + \sum_{j=1}^{p_\theta} \mathscr{H}_\theta(t_j) \mathscr{H}_\theta{}^\top(t_j), \tag{37}$$

where $r \in \mathbb{R}$, $\alpha_\theta > 0$ *and* $K_\theta = \mathbf{K}[\mathscr{I}_\theta] : \mathbb{R}^{n_{\xi_\theta}(n_{\xi_\theta}+m_\xi)} \rightrightarrows \mathbb{R}^{n_{\xi_\theta}(n_{\xi_\theta}+m_\xi)}$ *is a set-valued map. Furthermore, its convergence time is given by the exact settling time*

$$t_\theta{}^\star = \frac{1}{\alpha_\theta} \|\mathbf{H}_\theta(0)\|. \tag{38}$$

Proof. It follows from (29), (31), (35), and (36) that

$$\mathbf{H}_\theta(t) = \mathbf{A}_\theta \tilde{\Sigma}^{vec}_{\theta}(t). \tag{39}$$

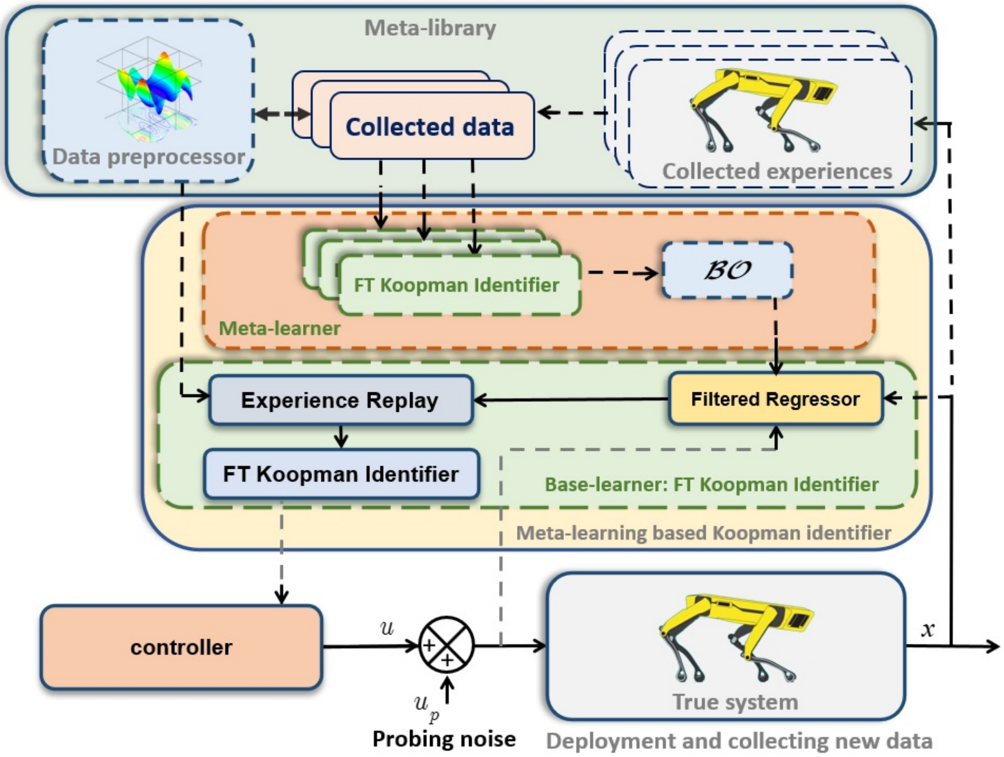

Figure 1: Proposed hierarchical learning architecture using Bayesian optimization (BO) for automatic lifted-state tuning. The lifted-state configuration θ_k is evaluated in the meta-learner in terms of cost functional $J^R(.)$. Based on all previous experiments, BO suggests the next lifted-state configuration θ_{k+1}, aiming at finding the global optimum with only a few iterations.

Using (39), (34) can be rewritten as

$$\mathscr{I}_\theta = -\alpha_\theta \frac{\|\mathbf{A}_\theta \tilde{\Sigma}_\theta^{vec}\|}{\tilde{\Sigma}_\theta^{vec\top} \mathbf{A}_\theta} \frac{[\mathbf{A}_\theta]^r \mathbf{A}_\theta \tilde{\Sigma}_\theta^{vec}}{[\mathbf{A}_\theta]^{r+1} \mathbf{A}_\theta \tilde{\Sigma}_\theta^{vec}}. \tag{40}$$

We begin now by proving that Condition 1 is satisfied. That is, $K_\theta = \mathbf{K}[\mathscr{I}_\theta] : \mathbb{R}^{n_{\xi_\theta}(n_{\xi_\theta}+m_\xi)} \rightrightarrows \mathbb{R}^{n_{\xi_\theta}(n_{\xi_\theta}+m_\xi)}$ is upper semi-continuous, and has compact, nonempty, and convex values. To this aim, using (40) and some manipulation, give

$$\begin{aligned} \|\mathscr{I}_\theta\| \ & \leq \alpha_\theta \|\frac{1}{2}\mathbf{A}_\theta \tilde{\Sigma}_\theta^{vec}\| \frac{\lambda_{\max}([\mathbf{A}_\theta]^r)}{\lambda_{\min}([\mathbf{A}_\theta]^{r+1})} \frac{\|\frac{1}{2}\mathbf{A}_\theta \tilde{\Sigma}_\theta^{vec}\|}{\|\frac{1}{2}\mathbf{A}_\theta \tilde{\Sigma}_\theta^{vec}\|^2} \\ & \leq \alpha_\theta \frac{\lambda_{\max}([\mathbf{A}_\theta]^r)}{\lambda_{\min}([\mathbf{A}_\theta]^{r+1})}. \end{aligned} \tag{41}$$

Based on Condition 3,

$$\sum_{j=1}^{p_\theta} \mathscr{H}_\theta(t_j)\mathscr{H}_\theta{}^\top(t_j) = \sum_{j=1}^{p_\theta} (\bar{h}_\theta(t_j)\bar{h}_\theta^\top(t_j)) \otimes I_{n_{\xi_\theta}} \geq d_\theta I_{n_{\xi_\theta}(n_{\xi_\theta}+m_\xi)}. \tag{42}$$

Hence, $\mathscr{H}_\theta(t)\mathscr{H}_\theta{}^\top(t) \geq 0_{n_{\xi_\theta}(n_{\xi_\theta}+m_\xi)}$ implies that $\lambda_{\min}((\mathscr{H}_\theta(t)\mathscr{H}_\theta^\top(t)+\sum_{j=1}^{p_\theta}\mathscr{H}_\theta(t_j)\mathscr{H}_\theta^\top(t_j))^{r+1}) \geq d_\theta$ everywhere near $\hat{\Sigma}_\theta^{vec} = \Sigma_\theta^{vec*}$ for some $d_\theta > 0$. As a result, (41) implies that $\mathscr{I}_\theta$ is defined a.e. and is Lebesgue measurable in a non-empty open region $\mathscr{R} \subset \mathbb{R}^{n_{\xi_\theta}(n_{\xi_\theta}+m_\xi)}$, and for every point $\tilde{\Sigma}_\theta^{vec} \in \mathbb{R}^{n_{\xi_\theta}(n_{\xi_\theta}+m_\xi)}$, $\mathscr{I}_\theta$ is bounded a.e. in some bounded neighborhood of $\tilde{\Sigma}_\theta^{vec}$. Using this observation and Theorem 5 of (Filippov and Arscott, 1988, Chapter 2), one can conclude that Condition 1 is satisfied, i.e., the Filippov set-valued map $K_\theta = \mathbf{K}[\mathscr{I}_\theta]$ is upper semi-continuous and has compact, nonempty, and convex values.

Now, take the following continuously differentiable Lyapunov function defined over $\mathbb{R}^{n_{\xi_\theta}(n_{\xi_\theta}+m_\xi)}$, which w.r.t. $\tilde{\Sigma}_\theta^{vec}$ is positive definite.

$$\begin{aligned} V(\tilde{\Sigma}_\theta^{vec}) &= \tilde{\Sigma}_\theta^{vec\top}(\mathbf{A}_\theta)^2\tilde{\Sigma}_\theta^{vec} \\ &= \left\|\mathbf{A}_\theta\tilde{\Sigma}_\theta^{vec}\right\|^2. \end{aligned} \tag{43}$$

Despite the update law (33)-(34) is continuous close to Σ_θ^{vec*}, it is not continuous at Σ_θ^{vec*}, and therefore, undefined at $\hat{\Sigma}_\theta^{vec} = \Sigma_\theta^{vec*}$ itself. To this end, given $\tilde{\Sigma}_\theta^{vec}(t) \in \mathbb{R}^{n_{\xi_\theta}(n_{\xi_\theta}+m_\xi)} \backslash \{\Sigma_\theta^{vec*}\}$, one has

$$\begin{aligned} \sup\dot{V}(\tilde{\Sigma}_\theta^{vec}) &= \sup\{a_\theta \in \mathbb{R} : \exists v \in \mathbf{K}[\mathscr{I}_\theta](\tilde{\Sigma}_\theta^{vec})\, s.t.\, a_\theta = p\cdot v, \forall p \in \partial V(\tilde{\Sigma}_\theta^{vec})\} \\ &= \sup\left\{2\,(\mathbf{A}_\theta)^2\tilde{\Sigma}_\theta^{vec}\cdot\left(\mathscr{I}_\theta(\tilde{\Sigma}_\theta^{vec}(t))\right)\right\} \\ &= -2\alpha_\theta\|\mathbf{A}_\theta\tilde{\Sigma}_\theta^{vec}\| \\ &= -(2\alpha_\theta)(V(\tilde{\Sigma}_\theta^{vec}))^{\frac{1}{2}}. \end{aligned} \tag{44}$$

Note that (43) satisfies Condition 2 since it is a continuously differentiable function. Furthermore, for $\hat{\Sigma}_\theta^{vec} = \Sigma_\theta^{vec*}$, i.e., $\tilde{\Sigma}_\theta^{vec}(t) = 0$, $\dot{V}\,(0) = 0$ since

$$\nabla_{\tilde{\Sigma}_\theta^{vec}}V\,(0) = 2((\mathscr{H}_\theta(t)\mathscr{H}_\theta{}^\top(t)+\sum_{j=1}^{p_\theta}\mathscr{H}_\theta(t_j)\mathscr{H}_\theta{}^\top(t_j)))^2\tilde{\Sigma}_\theta^{vec} = 0.$$

Moreover, using (33)-(36), one has

$$V(\tilde{\Sigma}_\theta^{vec}) = (\mathbf{H}_\theta(t))^\top\mathbf{H}_\theta(t), \tag{45}$$

which implies that

$$V(\tilde{\Sigma}_\theta^{vec}(0)) = (\mathbf{H}_\theta(0))^\top\mathbf{H}_\theta(0). \tag{46}$$

Now, invoking Proposition 1 and observing that $\sup\dot{V}(\tilde{\Sigma}_\theta^{vec}) = -(2\alpha)V(\tilde{\Sigma}_\theta^{vec})^{\frac{1}{2}}$ and $V\left(\tilde{\Sigma}_\theta^{vec}(0)\right)$ is computable, maximal Filippov solution to the discontinuous differential inclusion update law (33)-(34) converges to Σ_θ^{vec*} in finite time by the exact prescribed settling time (38). Using (29), the convergence of $\tilde{\Sigma}_\theta^{vec}$ within a finite time implies that e_{ξ_θ} also goes to the origin within a finite time. This completes the proof. □

Remark 3 *From* (41)*, one can see that a history stack's richness influences the convergence time of the identification error. To check whether finite-time convergence is possible, condition 3 provides an easy-to-verify rank condition. One can add a probing noise into the control signal to ensure that Condition 3 is satisfied.*

Remark 4 *It is also worth noting that even though the number of stored samples in the history stack is fixed after Condition 3 is satisfied, replacing old data with new rich data in the history stack can avoid numerical challenges in algorithm iterations in the case of poorly conditioned* $\mathbf{A}_\theta$*. To achieve this, new samples are periodically added, and old ones are removed if* $\lambda_{\min}([\mathbf{A}_\theta]^{r+1})$ *would be increased. This method, however, requires new data samples to stream. Therefore, in meanwhile, to keep the update law* (33)-(34) *practical in terms of being suitable for numerical solvers, one can also incorporate a small constant regularization term* $\eta > 0$ *(which is a small constant) in* (34)*, as*

$$\mathscr{I}_\theta = -\alpha_\theta \frac{\|\mathbf{H}_\theta(t)\|}{\eta + \mathbf{H}_\theta(t)} \frac{[\mathbf{A}_\theta]^r \mathbf{H}_\theta(t)}{[\mathbf{A}_\theta]^{r+1} \mathbf{H}_\theta(t)}, \tag{47}$$

and then remove it when $\mathbf{A}_\theta$ *is not poorly conditioned anymore. It is worth noting that the addition of this term transforms finite-time convergence into practical finite-time convergence. That is, one can select sufficiently small* η *to go into any given arbitrary neighborhood of the origin. Note that the concept of practical finite-time convergence has recently received significant attention (Liu et al., 2021; Chen et al., 2021; Chowdhary and Johnson, 2011) since it specifies a convergence bound and finite convergence time that can be effectively employed in control, identification, and monitoring to improve performance and reduce conservatism in control design.*

Remark 5 *To assure* $\|e_{A_\theta}\| \to 0$ *and* $\|e_{B_\theta}\| \to 0$*, the presented update law leverages modified gradient descent rule to select* $\hat{\Sigma}^\theta_{vec}$ *to minimize the following cost function for not only the current time, but also the past samples collected in the memory.*

$$J(t) = \bar{e}_{\xi_\theta}(t)^\top \bar{e}_{\xi_\theta}(t) + \sum_{j=1}^{p_\theta} \bar{e}_{\xi_\theta}(j)^\top \bar{e}_{\xi_\theta}(j), \tag{48}$$

where its gradient and the Hessian matrix can be calculated as

$$\nabla_{\tilde{\Sigma}^{vec}_\theta} J = \frac{1}{2}\left(\mathbf{A}_\theta + \mathbf{A}_\theta^\top\right) \tilde{\Sigma}^{vec}_\theta = \mathbf{A}_\theta \tilde{\Sigma}^{vec}_\theta, \tag{49}$$

$$\nabla^2_{\tilde{\Sigma}^{vec}_\theta} J = \frac{1}{2}\mathbf{A}_\theta + \frac{1}{2}\mathbf{A}_\theta^\top = \mathbf{A}_\theta. \tag{50}$$

Since $\mathbf{A}_\theta$ in (49) depends on both current (online) and past (batch) samples, the discontinuous gradient update law minimizes the identification error for both current and past samples. Before the rank condition is satisfied, only estimation error is guaranteed to converge to zero (no generalization guarantee), and as the samples are collected to satisfy Condition 3, the data samples provide a good representation of the dynamic system (16) across its entire operating regimes, if an appropriate set of observables is chosen (which will be performed in a meta-layer in this paper).

We now analyze the efficiency of the update law (33)-(34) by using the notion of regret. To this aim, let the normed error $\mathscr{N}(.)$ and continuous regret be defined as

$$\mathscr{N}(\tilde{\Sigma}^{vec}_\theta(t)) = \|\mathbf{A}_\theta(t) \tilde{\Sigma}^{vec}_\theta(t)\| \tag{51}$$

where

$$\begin{aligned} Regret &:= \textstyle\int_0^{t_\theta^*} \mathscr{N}(\tilde{\Sigma}^{vec}_\theta(\tau)) d\tau - \min_{\tilde{\Sigma}^{vec}_\theta \in \Theta_\theta} \int_0^{t_\theta^*} \mathscr{N}(\tilde{\Sigma}^{vec}_\theta(\tau)) d\tau \\ &= \textstyle\int_0^{t_\theta^*} \mathscr{N}(\tilde{\Sigma}^{vec}_\theta(\tau)) d\tau \end{aligned} \tag{52}$$

Note that $\min_{\hat{\Sigma}_\theta^{vec}\in\Theta_\theta}\int_0^{t_\theta{}^*}\mathscr{N}(\tilde{\Sigma}_\theta^{vec}(\tau))d\tau=0$, since $\min_{\hat{\Sigma}_\theta^{vec}\in\Theta_\theta}\mathscr{N}(\tilde{\Sigma}_\theta^{vec}(t))=0$ and regret given in (52) is a non-decreasing function of the time span $t_\theta{}^*$ since it contains a sum of $\mathscr{N}(\tilde{\Sigma}_\theta^{vec}(t))$, which are non-negative costs.

Theorem 2 *Consider the regret given in* (52). *Regret grows sub-linearly with time and the convergence of identification errors to zero for the update rule* (33)-(34) *is upper bounded by a constant regret* $\frac{1}{2\alpha_\theta}V(0)$ *where* $V(.)$ *is given in* (43).

Proof. Regret is analyzed based on the Lyapunov stability condition provided in Theorem 1 by $\dot{V}(t)\leq-(2\alpha_\theta)\|\mathbf{A}_\theta(t)\tilde{\Sigma}_\theta^{vec}(t)\|\leq 0$. Based on (43), one has $V(t)\geq 0$. Now, integrating $\dot{V}(t)$ from 0 to $t_\theta{}^*$, one has

$$\int_0^{t_\theta{}^*}\|\mathbf{A}_\theta(\tau)\tilde{\Sigma}_\theta^{vec}(\tau)\|d\tau\leq-\frac{1}{2\alpha_\theta}\int_0^{t_\theta{}^*}\dot{V}(\tau)d\tau=\frac{1}{2\alpha_\theta}(V(0)-V(t_\theta^*))=\frac{1}{2\alpha_\theta}V(0). \tag{53}$$

Given that $\dot{V}(t)\leq 0$, it can be seen that $V(0)-V(t^*)\leq V(0)=\mathscr{O}(1)$. This implies that the regret bound does not grow as a function of time (i.e., regret grows sub-linearly with time) since the convergence of identification errors to zero for the update rule (33)-(34), which is upper bounded by a constant regret $\frac{1}{2\alpha_\theta}V(0)$. This completes the proof. □

4.2 Batch-online Koopman finite-time identifier with approximation error

Now, we investigate the sensitivity of the proposed update law (33)-(34) to the approximation error term $\mathscr{P}_\theta(.)$ by studying the behavior of solutions of the system identifier (17) in a neighborhood of the finite-time solution of the nominal Koopman identifier. Let's assume that there exists the approximation error term $\mathscr{P}_\theta(.)$.

Problem 2 (Batch-online Koopman finite-time identifier with approximation error). Let the vector-valued observable $\xi(\theta)$ be fixed and results in some approximation error. Let the exist a mini-batch of samples given by $\{\xi_\theta(t_1),\ldots,(\xi_\theta(t_p)\}$. Considering (16) and the system identifier (17), develop a unified batch-online update law to ensure that e_{ξ_θ}, e_{A_θ}, and e_{B_θ}, are uniformly ultimately bounded (UUB), under an easy-to-verify condition on a mini-batch of samples.

Assumption 2 *Continuous approximation error term* $\mathscr{P}_\theta(t,\xi_\theta(t))$ *is bounded, i.e.,* $\|\ \mathscr{P}_\theta(t,\xi_\theta(t))\ \|\leq\mathscr{P}_\theta^B$.

Rewrite the system (19) as

$$\frac{d}{dt}\xi_\theta(t)=\Sigma_\theta^*{}^\top\mathscr{Z}_\theta(\xi_\theta,u)+\mathscr{P}_\theta(t,\xi_\theta(t)). \tag{54}$$

Proposition 3 *Consider the system* (54). *This system can be expressed as the filtered form*

$$\begin{cases}\xi_\theta(t)=\Sigma_\theta^*{}^\top h_\theta(t)+al_\theta(t)+e^{-aI_{n_{\xi_\theta}}t}\xi_\theta(0)+\mathscr{P}_\theta(t)\\ \dot{h}_\theta(t)=-a_\theta h_\theta(t)+\mathscr{Z}_\theta(\xi_\theta,u),\ h_\theta(\xi_\theta(0))=0\\ \dot{l}_\theta(t)=-al_\theta(t)+\xi_\theta,\qquad l_\theta(\xi_\theta(0))=0\end{cases} \tag{55}$$

$\forall a>0$ *with* $l_\theta(0)=0$ *and* $h_\theta(0)=0$, *where* $\mathscr{P}_\theta=\int\limits_0^T e^{-aI_{n_{\xi_\theta}}(t-\tau)}\mathscr{P}_\theta(\tau,\xi_\theta(\tau))d\tau$ *and* $\xi_\theta(0)$ *denotes the initial state of* (54) *and* $h_\theta(t)\in\mathbb{R}^{(n_{\xi_\theta}+m_\xi)}$ *and* $l_\theta(t)\in\mathbb{R}^{n_{\xi_\theta}}$ *denote the filtered regressor form of* $\mathscr{Z}_\theta(\xi_\theta,u)$ *and* ξ_θ, *respectively.*

Proof. The proof follows from a similar development as Proposition 2.

Using Proposition 3, (28) is reformulated as

$$\begin{cases} \bar{e}_{\xi_\theta}(t) = \tilde{\Sigma}_\theta^\top \bar{h}_\theta(t) - \bar{\mathscr{P}}_\theta(t) \\ \xi_\theta(t) = \Sigma_\theta^{*\top} h_\theta(t) + a l_\theta(t) + e^{-aI_{n_{\xi_\theta}} t} \xi_\theta(0) + \mathscr{P}_\theta(t) \\ \hat{\bar{\xi}}_\theta(t) = \hat{\Sigma}_\theta^\top \bar{h}_\theta(t) + a\bar{l}_\theta(t) + e^{-aI_{n_{\xi_\theta}} t} \bar{\xi}_\theta(0) \\ \dot{h}_\theta(t) = -a_\theta h_\theta(t) + \mathscr{Z}_\theta(\xi_\theta, u), \;\; h_\theta(\xi_\theta(0)) = 0 \\ \dot{l}_\theta(t) = -a l_\theta(t) + \xi_\theta, \qquad l_\theta(\xi_\theta(0)) = 0 \end{cases} \tag{56}$$

where $\bar{\mathscr{P}}_\theta = \mathscr{P}_\theta \big/ n_{s_\theta}$.

Now, (29) and (31) are rewritten as

$$\bar{e}_{\xi_\theta}(t) = (\bar{h}_\theta^\top(t) \otimes I_{n_{\xi_\theta}}) \tilde{\Sigma}_\theta^{vec}(t) - \bar{\mathscr{P}}_\theta(t) \tag{57}$$

$$\bar{e}_{\xi_\theta}(t, t_j) = (\bar{h}_\theta^\top(t_j) \otimes I_{n_{\xi_\theta}}) \tilde{\Sigma}_\theta^{vec}(t) - \bar{\mathscr{P}}_\theta(t), \, j = 1, ..., p_\theta. \tag{58}$$

The following theorem concerns the behavior of identifier error (41) along with the update law (33)-(34) under finite-dimensional Koopman with bounded approximation error.

Theorem 3 *Consider* (16) *and* (55)*, and let Assumptions 1 and 2 and Condition 3 hold. In the case that a bounded Koopman approximation error exists, the update law* (33)-(34) *ensures that all the identification errors are UUB.*

Proof. Using (34), (56), (57), and some manipulations, one has

$$\dot{\hat{\Sigma}}_\theta^{vec}(t) = \dot{\tilde{\Sigma}}_\theta^{vec}(t) \in K_\theta = \mathbf{K}[\mathscr{I}_\theta] : \mathbb{R}^{n_{\xi_\theta}(n_{\xi_\theta} + m_\xi)} \rightrightarrows \mathbb{R}^{n_{\xi_\theta}(n_{\xi_\theta} + m_\xi)}, \tag{59}$$

where

$$\mathscr{I}_\theta(t, \tilde{\Sigma}_\theta^{vec}, \bar{\mathscr{P}}_\theta) = -\alpha_\theta \| \mathbf{A}_\theta \tilde{\Sigma}_\theta^{vec} - \mathrm{B}_\theta \bar{\mathscr{P}}_\theta \| \Gamma_\theta, \tag{60}$$

with

$$\Gamma_\theta = \frac{[\mathbf{A}_\theta]^r \left[\mathbf{A}_\theta \tilde{\Sigma}_\theta^{vec} - \mathrm{B}_\theta \bar{\mathscr{P}}_\theta \right]}{\left[\mathbf{A}_\theta \tilde{\Sigma}_\theta^{vec} - \mathrm{B}_\theta \bar{\mathscr{P}}_\theta \right]^\top [\mathbf{A}_\theta]^{r+1} \left[\mathbf{A}_\theta \tilde{\Sigma}_\theta^{vec} - \mathrm{B}_\theta \bar{\mathscr{P}}_\theta \right]}, \tag{61}$$

$$\mathrm{B}_\theta := \mathscr{H}_\theta(t) + \sum_{j=1}^{p_\theta} \mathscr{H}_\theta(t_j). \tag{62}$$

Note that the analysis to show the validity of Conditions 1 and 2 is similar to that in the proof of Theorem 1. Using (59)-(62) and Definition 6, the derivative of continuously differentiable Lyapunov function (43) becomes

$$\begin{aligned} \sup \dot{V}(\tilde{\Sigma}_\theta^{vec}) &= \sup\{a_\theta \in \mathbb{R} : \exists v \in \mathbf{K}[\mathscr{I}_\theta](\tilde{\Sigma}_\theta^{vec}) \, s.t. \; a_\theta = p \cdot v, \forall p \in \partial V(\tilde{\Sigma}_\theta^{vec})\} \\ &= \sup\{2(\mathbf{A}_\theta)^2 \tilde{\Sigma}_\theta^{vec} \cdot (\mathscr{I}_\theta(t, \tilde{\Sigma}_\theta^{vec}, \bar{\mathscr{P}}_\theta))\} \\ &= -2\alpha_\theta (\mathbf{A}_\theta)^2 \tilde{\Sigma}_\theta^{vec} \left\| \mathbf{A}_\theta \tilde{\Sigma}_\theta^{vec} - \mathrm{B}_\theta \bar{\mathscr{P}}_\theta \right\| \Gamma_\theta. \end{aligned} \tag{63}$$

Adding and subtracting $-2\alpha_\theta \mathbf{A}_\theta \mathrm{B}_\theta \bar{\mathscr{P}}_\theta \left\|\mathbf{A}_\theta \tilde{\Sigma}_\theta^{vec} - \mathrm{B}_\theta \bar{\mathscr{P}}_\theta\right\| \Gamma_\theta$ to the right-hand side and some manipulations yields

$$\begin{aligned}
\sup \dot{V}(\tilde{\Sigma}_\theta^{vec}) = & -2\alpha_\theta \mathbf{A}_\theta \left(\mathbf{A}_\theta \tilde{\Sigma}_\theta^{vec} - \mathrm{B}_\theta \bar{\mathscr{P}}_\theta\right) \left\|\mathbf{A}_\theta \tilde{\Sigma}_\theta^{vec} - \mathrm{B}_\theta \bar{\mathscr{P}}_\theta\right\| \Gamma_\theta \\
& -2\alpha_\theta \mathbf{A}_\theta \mathrm{B}_\theta \bar{\mathscr{P}}_\theta \left\|\mathbf{A}_\theta \tilde{\Sigma}_\theta^{vec} - \mathrm{B}_\theta \bar{\mathscr{P}}_\theta\right\| \Gamma_\theta \\
\leq & -2\alpha_\theta \left\|\mathbf{A}_\theta \tilde{\Sigma}_\theta^{vec} - \mathrm{B}_\theta \bar{\mathscr{P}}_\theta\right\| + 2\alpha_\theta \frac{\lambda_{\max}([\mathbf{A}_\theta]^r)}{\lambda_{\min}([\mathbf{A}_\theta]^{r+1})} \left\|\mathbf{A}_\theta \mathrm{B}_\theta \bar{\mathscr{P}}_\theta\right\| \\
\leq & -2\alpha_\theta \left\|\mathbf{A}_\theta \tilde{\Sigma}_\theta^{vec}\right\| + \varepsilon(\tilde{\Sigma}_\theta^{vec}, \bar{\mathscr{P}}_\theta) \\
\leq & -2\alpha_\theta \left(V(\tilde{\Sigma}_\theta^{vec})\right)^{\frac{1}{2}} + \varepsilon(\tilde{\Sigma}_\theta^{vec}, \bar{\mathscr{P}}_\theta), \qquad (64)
\end{aligned}$$

where

$$\varepsilon(\tilde{\Sigma}_\theta^{vec}, \bar{\mathscr{P}}_\theta) = 2\alpha_\theta \left[\left\|\mathrm{B}_\theta \bar{\mathscr{P}}_\theta\right\| + \frac{\lambda_{\max}([\mathbf{A}_\theta]^r)}{\lambda_{\min}([\mathbf{A}_\theta]^{r+1})} \left\|\mathbf{A}_\theta \mathrm{B}_\theta \bar{\mathscr{P}}_\theta\right\|\right]. \qquad (65)$$

Note that

$$-2\alpha_\theta \left\|\mathbf{A}_\theta \tilde{\Sigma}_\theta^{vec} - \mathrm{B}_\theta \bar{\mathscr{P}}_\theta\right\| \leq -2\alpha_\theta \left[\left\|\mathbf{A}_\theta \tilde{\Sigma}_\theta^{vec}\right\| - \left\|\mathrm{B}_\theta \bar{\mathscr{P}}_\theta\right\|\right], \qquad (66)$$

and

$$\begin{aligned}
& \left\|2\alpha_\theta \mathbf{A}_\theta \mathrm{B}_\theta \bar{\mathscr{P}}_\theta\right\| \left\|\mathbf{A}_\theta \tilde{\Sigma}_\theta^{vec} - \mathrm{B}_\theta \bar{\mathscr{P}}_\theta\right\| \left\|\Gamma_\theta\right\| \\
& = 2\alpha_\theta \left\|\mathbf{A}_\theta \tilde{\Sigma}_\theta^{vec} - \mathrm{B}_\theta \bar{\mathscr{P}}_\theta\right\| \frac{\left\|\mathbf{A}_\theta \mathrm{B}_\theta \bar{\mathscr{P}}_\theta\right\| \left\|[\mathbf{A}_\theta]^r \left[\mathbf{A}_\theta \tilde{\Sigma}_\theta^{vec} - \mathrm{B}_\theta \bar{\mathscr{P}}_\theta\right]\right\|}{\left[\mathbf{A}_\theta \tilde{\Sigma}_\theta^{vec} - \mathrm{B}_\theta \bar{\mathscr{P}}_\theta\right]^\top [\mathbf{A}_\theta]^{r+1} \left[\mathbf{A}_\theta \tilde{\Sigma}_\theta^{vec} - \mathrm{B}_\theta \bar{\mathscr{P}}_\theta\right]} \\
& \leq 2\alpha_\theta \left\|\mathbf{A}_\theta \tilde{\Sigma}_\theta^{vec} - \mathrm{B}_\theta \bar{\mathscr{P}}_\theta\right\| \frac{\lambda_{\max}([\mathbf{A}_\theta]^r) \left\|\mathbf{A}_\theta \mathrm{B}_\theta \bar{\mathscr{P}}_\theta\right\| \left\|\left[\mathbf{A}_\theta \tilde{\Sigma}_\theta^{vec} - \mathrm{B}_\theta \bar{\mathscr{P}}_\theta\right]\right\|}{\lambda_{\min}([\mathbf{A}_\theta]^{r+1}) \left\|\left[\mathbf{A}_\theta \tilde{\Sigma}_\theta^{vec} - \mathrm{B}_\theta \bar{\mathscr{P}}_\theta\right]\right\|^2} \\
& \leq 2\alpha_\theta \frac{\lambda_{\max}([\mathbf{A}_\theta]^r)}{\lambda_{\min}([\mathbf{A}_\theta]^{r+1})} \left\|\mathbf{A}_\theta \mathrm{B}_\theta \bar{\mathscr{P}}_\theta\right\|. \qquad (67)
\end{aligned}$$

Now, if the following inequalities hold

$$\tilde{\Sigma}_\theta^{vec} > \frac{1}{2\alpha_\theta} \varepsilon(\tilde{\Sigma}_\theta^{vec}, \bar{\mathscr{P}}_\theta)(\mathscr{H}_\theta(t) \mathscr{H}_\theta{}^\top(t) + \sum_{j=1}^{p_\theta} \mathscr{H}_\theta(t_j) \mathscr{H}_\theta{}^\top(t_j)),$$

then, $\sup \dot{V}(\tilde{\Sigma}_\theta^{vec}) < 0$, which completes the proof. □

4.3 Meta Learner

To learn a Koopman operator model with minimum approximation error and dimension, a meta-learner is developed in this subsection. Toward this aim, a meta cost is defined as

$$J^R(\Xi) := \ell_\theta(\hat{A}_\theta, \hat{B}_\theta, \mathscr{D}^{\text{eval}}) + \lambda n_{\xi_\theta}, \qquad (68)$$

where $\Xi = [\theta(1), ..., \theta(n_{\xi_\theta})]^\top$ denotes the vector of the ordered set θ, and leads to the $(\hat{A}_\theta, \hat{B}_\theta)$ model by the base learner layer, λ is a constant, and $n_{\xi_\theta} = |\theta|$ promotes sparsity to find a set of observable with minimum cardinality. The meta loss function ℓ_θ is the average identification error for a set of observables with respect to the collected data set $\mathscr{D}^{eval}$ after the lower layer convergences to $(\hat{A}_\theta, \hat{B}_\theta)$ model, and, it is defined as

$$\ell_\theta\left(\hat{A}_\theta, \hat{B}_\theta, \mathscr{D}^{\text{eval}}\right) = \frac{1}{p_\theta} \sum_{k=1}^{p_\theta} \|\hat{\bar{\xi}}_\theta(\mathscr{D}^{\text{eval}}, \hat{A}_\theta, \hat{B}_\theta, t, t_k) - \bar{\xi}_\theta(\mathscr{D}^{\text{eval}}, t_k)\| \tag{69}$$

with

$$\begin{cases} \hat{\bar{\xi}}_\theta(t) = \hat{\Sigma}_{\theta i}^\top \bar{h}_\theta(t, \mathscr{D}^{\text{eval}}) + \mathscr{A}_\theta \bar{l}_\theta(t, \mathscr{D}^{\text{eval}}) \\ \qquad\quad + e^{-\mathscr{A}_\theta t} \bar{\xi}_\theta(t, \mathscr{D}^{\text{eval}}, 0) \\ \dot{h}_\theta(t) = -a_\theta h_{\theta i}(t, \mathscr{D}^{\text{eval}}) + \mathscr{Z}_{\theta i}(\xi_\theta(t, \mathscr{D}^{\text{eval}}), u) \\ \dot{l}_\theta(t) = -\mathscr{A}_\theta l_\theta(t, \mathscr{D}_i^{\text{eval}}) + \xi_\theta(t, \mathscr{D}^{\text{eval}}) \end{cases} \tag{70}$$

The collected data set $\mathscr{D}^{eval}$ is formed by samples that are stored in the memory and represent the state space well. Since the proposed novel batch-online learning in the lower layer allows learning in finite time, data set $\mathscr{D}^{eval}$ can provide a fair and control-agnostic comparison for sets of observables. It means that the Koopman parameters for selected observables by the meta layer can be computed in finite time without requiring additional rich incremental data. That is, different matrix pairs $(\hat{A}_\theta, \hat{B}_\theta)$ are obtained in finite time by the lower layer learner as the set of observables varies, which results in different identifier system (70). Therefore, $J^R(\Xi)$ is a function of sets θ which are parameterizing the lifted state vector of the base learner, i.e., $\xi_\theta(t)$.

As delineated in Algorithm 1, to optimize the meta cost, a Bayesian optimization (BO) strategy for discrete variables (Shahriari et al., 2015; Luong et al., 2019) is leveraged, which leverages the previously recorded data sets in the memory, i.e., $\mathscr{D}^{\text{eval}}$, to map the so-called meta-parameters θ to meta cost $J^R(\Xi)$. In Steps 1-2, the algorithm is started up by using the available collecting data samples $\mathscr{D}$ and $\mathscr{D}^{\text{eval}}$ for T_{outer} randomly chosen different vectors of Ξ_k (with $k = 1, \ldots, T_{outer}$). The training samples are used by the lower layer learner update law and can be a combination of the samples collected incrementally as well as the memory samples. For each vector θ_k, the meta cost, however, J_k^R is evaluated using (69)-(70) and leveraging only the samples in the memory (to make the identifier control agnostic). As a result, the initial set $\mathscr{D} = \left\{(\Xi_k, J_k^R) : k = 1, ..., T_{out}\right\}$ of libraries of observable functions and corresponding performance J^R is constructed. Since the function J^R is not known a priori, nonparametric Gaussian Process (GP) models should be used (Steps 3-4) to approximate it using the training dataset $\mathscr{D}$. Therefore, GP model characterizing "the best guess" of J^R corresponding to the library of observable functions θ, i.e., $\mathscr{L}(\theta)$, based on the available training dataset $\mathscr{D}$. To this aim, the function values J^R, which are related to different sets of θ, are assumed as random variables with a joint Gaussian distribution (i.e., J^R is a Gaussian variable) dependent on the vector of Ξ with the defined a prior mean function and variance

$$m_k(\Xi) = \mathbf{k}_k'(\mathbf{K}_k + \sigma_e^2 I)^{-1} J_k^R(\Xi), \tag{73}$$

$$\sigma_k^2(\Xi) = \kappa(\Xi, \Xi) - \mathbf{k}_k'(\mathbf{K}_k + \sigma_e^2 I)^{-1}\mathbf{k}_k + \sigma_e^2, \tag{74}$$

where σ_e^2 is the variance of Gaussian noise, and $\kappa(\Xi, \Xi_j)$ and $\kappa(\Xi_j, \Xi_m)$ denote the j-th and $[j, m]$-th entry of the vector $\mathbf{k}_k$ and Kernel matrix $\mathbf{K}_k$, respectively. The covariance function $\kappa(\Xi, \tilde{\Xi})$ determines the covariance between $J^R(\theta)$ and $J^R(\tilde{\theta})$ and is described as

$$\kappa(\Xi, \tilde{\Xi}) = \sigma_0^2 e^{-\frac{1}{2\lambda^2}\left[\Xi' - \tilde{\Xi}'\mu' - \tilde{\mu}'\right]\left[\Xi' - \tilde{\Xi}'\mu' - \tilde{\mu}'\right]'}, \tag{75}$$

```
Algorithm 1 Meta Learner
Result: Optimal observables vector $\Xi^*$.
 1: Evaluate meta-loss function: $J^R \leftarrow \left\{J^R(\Xi), \mathscr{D}^{eval}\right\}$ ;
 2: Assemble and construct a training dataset: $\mathscr{D} \leftarrow \left\{\Xi; J^R\right\}$ ;
 3: Place a GP prior on $J^R$;
 4: Initialize the GP using $\mathscr{D}$ ;
 for $k = 1$ to $T_{outer}$ do
     5.1: Train the GP that approximates $J^R$ on the basis of data $\mathscr{D}$;
     5.2: Determine the acquisition function $\alpha(\Xi \mid \mathscr{D})$ using the GP;
     5.3: Compute the subsequent vector $\Xi$ as
          $$\begin{aligned} \Xi_{k+1} &\leftarrow \arg\max_{\Xi \in \mathscr{M}} \alpha(\Xi \mid \mathscr{D}) \\ \Xi_{k+1} &= round(\Xi_{k+1}) \end{aligned} \qquad (71)$$
     where $\mathscr{M}$ is some design space of interest;
     5.4: Evaluate $J^R_{k+1}$ based on $\mathscr{D}^{\text{eval}}$, and append it into $\mathscr{D}$: $\mathscr{D} \leftarrow \mathscr{D} \cup \left\{\Xi_{k+1}; J^R_{k+1}\right\}$;
     5.5: If the stopping criterion is satisfied, exit the loop;
 end
 6: Return $\Xi^* = round(\Xi_{k^*})$ where
          $$k^* = \arg\min_k J^R_k \qquad (72)$$
```

where σ_0 and λ are the design parameters.

Afterward, the algorithm is repeated till a predefined termination criterion is satisfied. Following are the steps that are performed during each iteration. Based on the available training dataset $\mathscr{D}$, a GP is fitted in Step 5.1. To find the next library of observable functions θ_{k+1}, the function $\alpha(\Xi \mid \mathscr{D})$ (which is called the acquisition function) is getting optimized in Step 5.2. This function $\alpha(.)$ is determined using the estimated GP mean and covariance, and its main objective is to balance between exploration and exploitation. That is, exploring by evaluating the function J^R in domains of the search space with high variance and also exploiting the past recorded data and optimizing the expected improvement over domains with high mean. Now, let the acquisition function $\alpha(.)$ be defined as

$$\alpha(\Xi \mid \mathscr{D}) = \mathbb{E}\left[\max\left\{0, J^{R*} - J^R(\Xi)\right\}\right], \tag{76}$$

where the target value J^{R*} is the minimum of all explored data, represents the most optimal value of J^R, and defined as

$$J^{R*} = \arg\min_k J^R_k. \tag{77}$$

The acquisition function given in (76) can be computed analytically based on the mentioned GP framework as

$$\alpha(\Xi \mid \mathscr{D}) = \begin{cases} \left(J^{R*} - m_k(\Xi)\right)\Phi(Z) + \sigma_k(\Xi)\psi(Z) & \sigma_i(\Xi, v) > 0 \\ 0 & O.W. \end{cases} \tag{78}$$

where $Z = (J^{R*} - m_k(\Xi))/\sigma_k(\Xi)$, Φ denotes the cumulative density function, and ψ is the probability density function. Intuitively, the acquisition function given in (76) selects the succeeding parameter point at which the improvement over J^{R*} should be the most in expectation. Note that it is no need for real physical interaction with the system in order to optimize $\alpha(\Xi \mid \mathscr{D})$ in (71), and only need to evaluate the GP model. If any new data sets $\mathscr{D}_{k+1}$ and $\mathscr{D}_{k+1}^{\text{eval}}$ are available, they will be collected and augmented to the available data sets $\mathscr{D}$ and $\mathscr{D}^{\text{eval}}$ in Step 5.4. In Step 5.5, the new determined optimal vector Ξ^* is evaluated based on the updated evaluation data of real the system.

Remark 6 *For the sake of simplicity, the BO with naive rounding (Naive BO) approach is used to deal with the discrete nature of Ξ. That is, we treat discrete variables as continuous then apply a normal BO method, and finally rounds the suggested continuous point to the nearest discrete point before function evaluations.*

Remark 7 *Using Bayesian optimization, bounds can be set on the search space of θ. When maximization of the acquisition function is performed in Algorithm 1, then these bounds can be incorporated. In general, when the search space is narrowed down, the algorithm tends to converge faster, therefore, requiring fewer time-consuming evaluations of the functional $J^R(\Xi)$. Specifically, each evaluation of the functional $J^R(\Xi)$ takes*

$$t_\theta{}^\star = \frac{1}{\alpha_\theta}\|\mathbf{H}_\theta(0)\|. \tag{79}$$

Prior system knowledge and design choices may be exploited to define suitable bounds. Moreover, the proposed method is agnostic to the choice of the controller u since it relies on collected data in the memory to evaluate the meta cost. However, if we have some prior knowledge of the controller u, it would also be leveraged to narrow the search space and consequently accelerate the convergence of the algorithm.

Remark 8 *It is worth noting that a proposed batch learning method developed by (Lehrer et al., 2010), aims to identify uncertain discrete-time systems in finite-time, requiring the online invertibility check of regressor matrix and its inverse computation as well as interval excitation of regressor. In the case of large numbers of unknown parameters, however, the required inversion of the regressor matrix makes the method in (Lehrer et al., 2010) computationally inefficient for online learning.*

5. Simulation

In this section, two simulation results are presented to validate the theoretical results.

Example 1: The following nonlinear system is considered as the system dynamics

$$\begin{aligned} \dot{x}_1 &= \mu x_1 \\ \dot{x}_2 &= \lambda\left(x_2 - x_1^4 + 2x_1^2\right) + u \end{aligned} \tag{80}$$

where it can be rewritten as

$$\underbrace{\frac{d}{dt}\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}}_{\xi} = \underbrace{\begin{bmatrix} \mu & 0 & 0 & 0 \\ 0 & \lambda & 2\lambda & -\lambda \\ 0 & 0 & 2\mu & 0 \\ 0 & 0 & 0 & 4\mu \end{bmatrix}}_{A^*} \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}}_{\xi} + \underbrace{\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}}_{B^*} u, \tag{81}$$

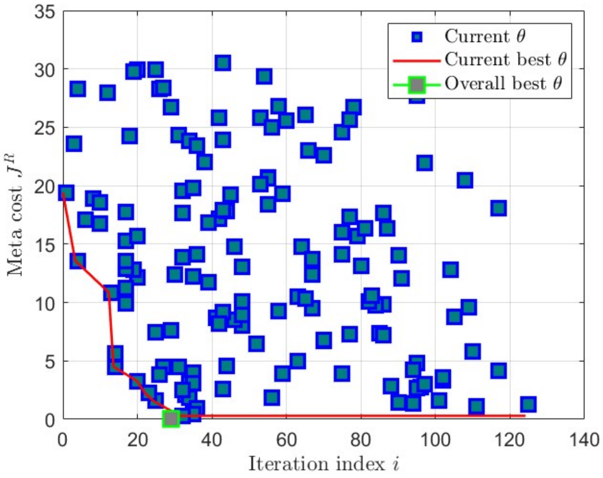

Figure 2: Meta cost J^R vs iteration i of Algorithm 1. At each iteration i, the cost of the current set of observable functions θ and of the current best one are depicted. The green square shows the overall best $\theta = \{1,2,4,9\}$ which is found at iteration 29.

with $\xi = [x_1, x_2, x_1^2, x_1^4]^\top$, $\mu = -1$, $\lambda = -1$.

To satisfy Condition 3, we inject the following probing noise into the control input (i.e., add it to the control input u) for $t \in (0\ -\ 0.5)\ sec$

$$\begin{aligned}&1.25e^{-t}(0.4(sin(0.1t))^6cos(1.5t)+0.3(sin(2.3t))^4cos(0.7t)+0.5cos(2.4t)(sin(7.4t))^2\\&+0.4(sin(2.6t))^5+0.7(sin(3t))^2cos(4t)+0.3sin(0.3t)(cos(1.2t))^2+0.4(sin(1.12t))^3\\&+0.3(sin(4t))^3+0.4(sin(3.5t))^5+0.4cos(2t)(sin(5t))^8+0.3sin(t)(cos(0.8t))^2\\&+0.5(cos(2.4t))^3(sin(7.4t))^2+0.1(sin(3.5t))^7+0.1(cos(2t))^4(sin(5t))^4+0.4(sin(2.1t))^2\\&+0.3(sin(2.1t))^3(cos(0.9t))^3+0.1(sin(1.7t))^5(cos(0.9t))^2+0.1(sin(0.4t))^2(cos(1.6t))^3)\end{aligned} \tag{82}$$

The proposed learning scheme is used to find the best library of observable functions θ, and matrices A^*, and B^*. In the history stack, the size of recorded data is set as 21.

We choose a finite set of polynomial observable functions based on the monomials of states x_1 and x_2 as follows

$$\mathscr{C}(k) = \Big\{x_1^a \cdot x_2^b \mid a,b \in \{0,1,2,\dots k\}\Big\} \cup \{x_1^4\}, \tag{83}$$

where $k = 2$ is the order of the basis functions, $\xi_1(x) = x_1$, $\xi_2(x) = x_2$, $\xi_3(x) = x_1 \cdot x_2$, $\xi_4(x) = x_1^2$, $\xi_5(x) = x_2^2$, $\xi_6(x) = x_1^2 \cdot x_2$, $\xi_7(x) = x_1 \cdot x_2^2$, $\xi_8(x) = x_1^2 \cdot x_2^2$, and $\xi_9(x) = x_1^4$. Now, for an ordered set θ where $\theta \subseteq \{1,...,9\}$ and $n_{\xi_\theta} = |\theta| \leq 9$, the vector $\xi_\theta(t) = [\xi_{\theta(1)}(x(t)), \cdots, \xi_{\theta(n_{\xi_\theta})}(x(t))]^\top$ can be defined, by using Definition 7, to lift the system from a state-space to function space of observables. For instance, the ordered set $\theta = \{1,2,4,9\}$ corresponds with the lifted state vector $\xi_{\theta=\{1,2,4,9\}}(t) = [\xi_1(x) = x_1,\ \xi_2(x) = x_2,\ \xi_4(x) = x_1^2,\ \xi_9(x) = x_1^4]^\top$.

The system is initialized at $x_1 = 1$ and $x_2 = -1$. Let $\alpha_\theta = 3$. MATLAB Statistics and Machine Learning Toolbox is used to implement Algorithm 1, with the EI in (76) serving as an acquisition function. The meta cost J^R vs iteration i of Algorithm 1 is illustrated in Fig. 2. Fig. 3 shows that,

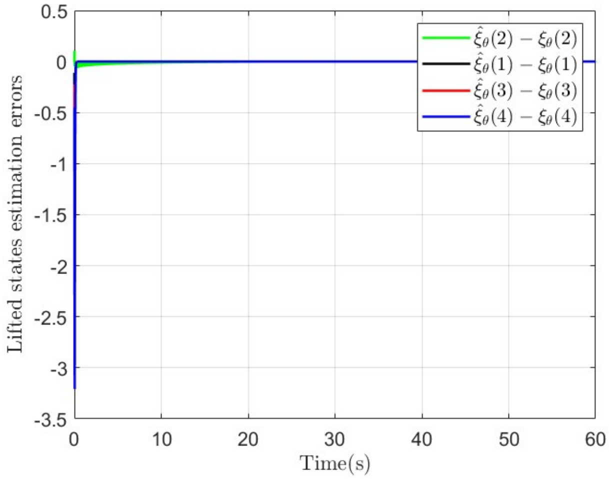

Figure 3: Lifted state estimation errors with the library of observables $\theta = \{1,2,4,9\}$.

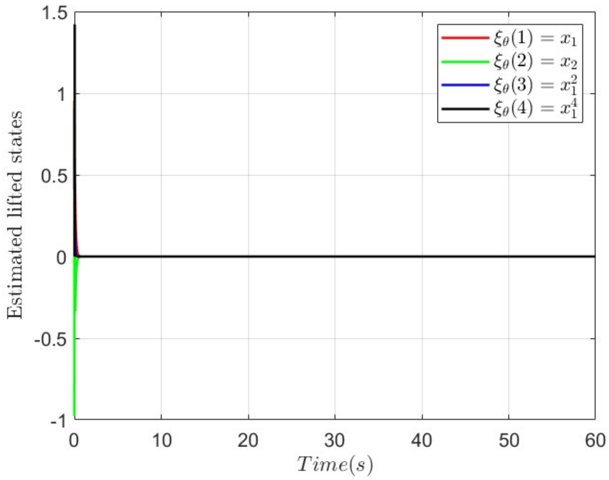

Figure 4: Lifted state trajectories with the library of observables $\theta = \{1,2,4,9\}$.

within finite time, the lifted state estimation error of the base learner for $\theta = \{1,2,4,9\}$ has been zeroed out. The lifted states' trajectories evolution is described in Fig. 4. The convergence of the base learner parameters

$$\hat{A}_{\theta=\{1,2,4,9\}} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & -2 & 1 \\ 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & -4 \end{bmatrix}, \hat{B}_{\theta=\{1,2,4,9\}} = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}^\top,$$

to the true values A^* and B^* within finite time and during online learning has been shown in Fig. 5 and Fig. 6 .

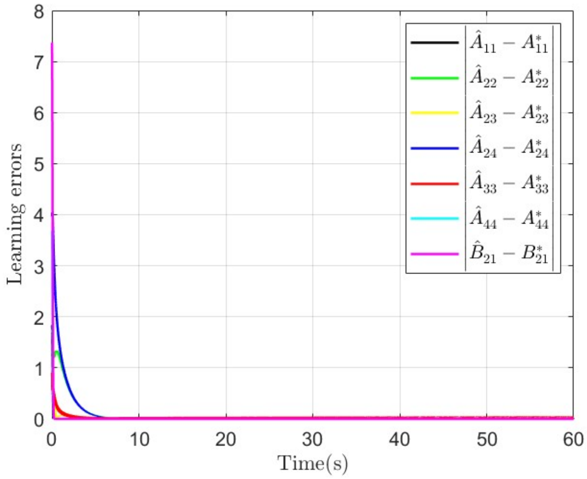

Figure 5: Online learning error with the library of observables $\theta = \{1,2,4,9\}$.

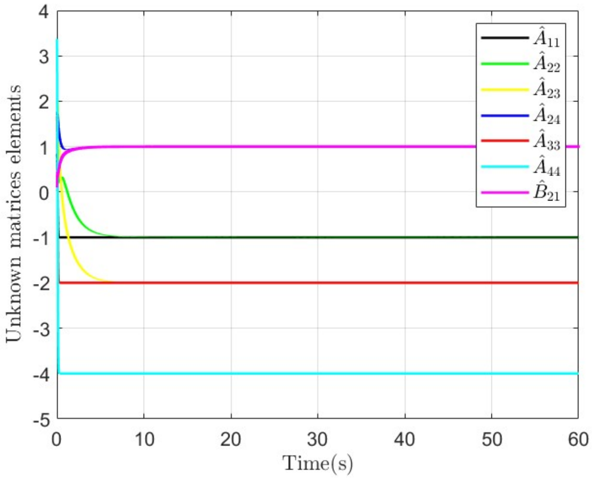

Figure 6: Estimated unknown elements of A^* and B^* with the library of observables $\theta = \{1,2,4,9\}$. Note that for simplicity of demonstration, we only show a non-zero elements of matrices.

For the case of $\theta = \{1,2,4,8\}$, the finite time convergence of the base learner parameters

$$\hat{A}_{\theta=\{1,2,4,8\}} = \begin{bmatrix} -0.8316 & 0.2151 & 1.1806 & -0.7588 \\ -0.1784 & -0.9781 & 0.1839 & -1.5642 \\ -0.3880 & -0.3554 & -2.0771 & -0.6459 \\ 0.6716 & 0.6625 & -4.5250 & -1.6133 \end{bmatrix}$$

$$\hat{B}_{\theta=\{1,2,4,8\}} = \begin{bmatrix} -0.0973 & 0.9653 & 0.1077 & -0.1402 \end{bmatrix}^\top$$

to neighborhood values of the true matrices A^* and B^* during online learning has been shown in Fig. 7

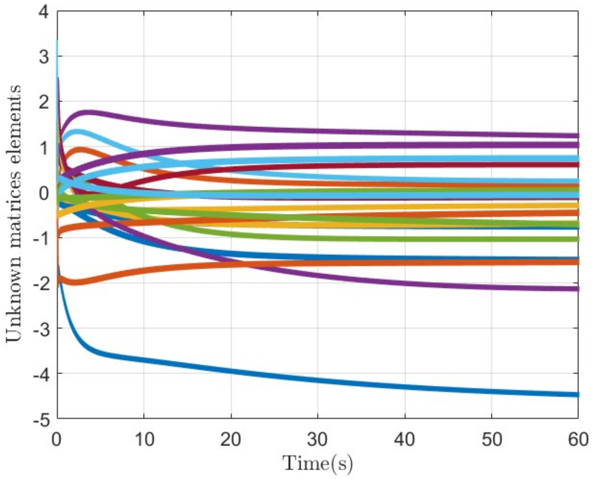

Figure 7: Estimated unknown elements of A^* and B^* with the library of observables $\theta = \{1,2,4,8\}$.

Example 2:

The vehicle is modeled as a nonlinear single-track bicycle model (See Fig. 8). The yaw dynamics of a simple vehicle can be qualitatively described and analyzed with this model in all operating conditions. Using the Lagrangian approach, the state equation of this vehicle model can be written as (Andrzejewski and Awrejcewicz, 2006; de Souza Mendes et al., 2016)

$$\begin{aligned}
&\dot{v}_T = \left(F_{x,\mathbf{F}}\cos\left(\alpha_T-\delta\right)+F_{x,\mathbf{R}}\cos\left(\alpha_T\right)\right)/m_T+\left(F_{y,\mathbf{F}}\sin\left(\alpha_T-\delta\right)+F_{y,\mathbf{R}}\sin\left(\alpha_T\right)\right)/m_T\\
&\dot{\alpha}_T = \left(-F_{x,\mathbf{F}}\sin\left(\alpha_T-\delta\right)-F_{x,\mathbf{R}}\sin\left(\alpha_T\right)\right)/\left(m_T v_T\right)+\left(F_{y,\mathbf{F}}\cos\left(\alpha_T-\delta\right)+F_{y,\mathbf{R}}\cos\left(\alpha_T\right)-m_T v_T\dot{\psi}\right)/\left(m_T v_T\right)\\
&\ddot{\psi} = \left(F_{x,\mathbf{F}}a\sin(\delta)+F_{y,\mathbf{F}}a\cos(\delta)-F_{y,\mathbf{R}}b\right)/I_T
\end{aligned} \tag{84}$$

where the longitudinal forces acting on the front and rear tires, respectively, are given by $F_{x,\mathbf{F}}$ and $F_{x,\mathbf{R}}$. Moreover, $F_{y,\mathbf{F}}$ and $F_{y,\mathbf{R}}$ (the lateral forces) are also determined by the selected tire model where the subscripts $_\mathbf{F}$ and $_\mathbf{R}$ denote the front and rear points to which theses forces are associated. Moreover, α_T denotes the slip angle, and v_T denotes the vehicle center of gravity velocity, m_T denotes the vehicle mass, δ denotes the front axle steering angle, I_T is the vehicle inertia, and the distances between the points **F**, CG, and **R** are determined by the constants a and b.

The following equation describes tire model

$$\begin{aligned}
F_{y,\mathbf{F}} &= K\alpha_{\mathbf{F}},\\
F_{y,\mathbf{R}} &= K\alpha_{\mathbf{R}},
\end{aligned} \tag{85}$$

where the constant K is stiffness, and

$$\begin{aligned}
\alpha_{\mathbf{F}} &= \alpha_{\mathrm{T}}+\tfrac{a}{v_{\mathrm{T},0}}\dot{\psi}-\delta,\\
\alpha_{\mathbf{R}} &= \alpha_{\mathrm{T}}-\tfrac{b}{v_{\mathrm{T},0}}\dot{\psi}.
\end{aligned} \tag{86}$$

We first reformulate the vehicle dynamics (84) into nonlinear affine in the control system and then develop a novel hierarchical learning structure that leverages a novel incremental finite-time

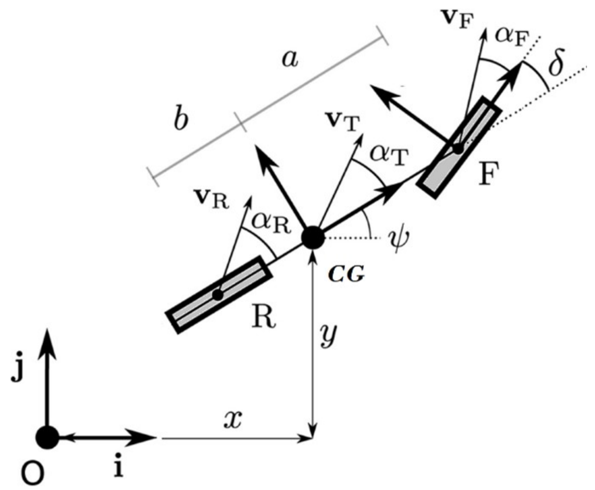

Figure 8: The single-track model. In this simulation, all tires on the same axle are assumed to have equal slip angles, and each axle is represented by a single tire with equivalent dynamic properties.

Koopman-based update law to approximately learn the linear representation of vehicle dynamics (84). To this aim, using the facts that

$$\begin{aligned}\cos(\alpha_\mathrm{T}-\delta) &= \cos\alpha_\mathrm{T}\cos\delta+\sin\alpha_\mathrm{T}\sin\delta,\\ \sin(\alpha_\mathrm{T}-\delta) &= \sin\alpha_\mathrm{T}\cos\delta-\cos\alpha_\mathrm{T}\sin\delta.\end{aligned}$$

One can rewrite (84) as

$$\begin{aligned}
\dot{v}_\mathrm{T} &= \tfrac{2}{m_T}K{\alpha_\mathrm{T}}^2+\tfrac{1}{m_T}K\tfrac{(a-b)}{v_{\mathrm{T},0}}\dot{\psi}\alpha_\mathrm{T}-\tfrac{1}{m_T}K\alpha_\mathrm{T}\delta-(\tfrac{1}{2m_T}K{\alpha_\mathrm{T}}^2+\tfrac{1}{2m_T}K\tfrac{a}{v_{\mathrm{T},0}}\dot{\psi}\alpha_\mathrm{T})\delta^2+\tfrac{1}{2m_T}K\alpha_\mathrm{T}\delta^3\\
&\quad+(\tfrac{1}{m_T}K\tfrac{a}{v_{\mathrm{T},0}}\dot{\psi}\tfrac{{\alpha_\mathrm{T}}^2}{2}-\tfrac{1}{m_T}K\tfrac{a}{v_{\mathrm{T},0}}\dot{\psi}-\tfrac{1}{m_T}K\alpha_\mathrm{T}+\tfrac{1}{m_T}K\alpha_\mathrm{T}\tfrac{{\alpha_\mathrm{T}}^2}{2})\sin\delta+(\tfrac{1}{m_T}K-\tfrac{1}{m_T}K\tfrac{{\alpha_\mathrm{T}}^2}{2})\delta\sin\delta\\
\dot{\alpha}_\mathrm{T} &= \tfrac{2}{m_Tv_\mathrm{T}}K\alpha_\mathrm{T}+\tfrac{1}{m_Tv_\mathrm{T}}K\tfrac{(a-b)}{v_{\mathrm{T},0}}\dot{\psi}-\tfrac{1}{m_Tv_\mathrm{T}}K\tfrac{(a-b)}{v_{\mathrm{T},0}}\dot{\psi}\tfrac{{\alpha_\mathrm{T}}^2}{2}-\tfrac{2}{m_Tv_\mathrm{T}}K\tfrac{{\alpha_\mathrm{T}}^3}{2}-\dot{\psi}+(\tfrac{1}{m_Tv_\mathrm{T}}K\tfrac{{\alpha_\mathrm{T}}^2}{2}-\tfrac{1}{m_Tv_\mathrm{T}}K)\delta\\
&\quad+(\tfrac{1}{2m_Tv_\mathrm{T}}K\alpha_\mathrm{T}\tfrac{{\alpha_\mathrm{T}}^2}{2}-\tfrac{1}{2m_Tv_\mathrm{T}}K\tfrac{a}{v_{\mathrm{T},0}}\dot{\psi}+\tfrac{1}{2m_Tv_\mathrm{T}}K\tfrac{a}{v_{\mathrm{T},0}}\dot{\psi}\tfrac{{\alpha_\mathrm{T}}^2}{2}-\tfrac{1}{2m_Tv_\mathrm{T}}K\alpha_\mathrm{T})\delta^2\\
&\quad+(\tfrac{1}{2m_Tv_\mathrm{T}}K-\tfrac{1}{m_Tv_\mathrm{T}}K\tfrac{{\alpha_\mathrm{T}}^2}{4})\delta^3+(\tfrac{1}{m_Tv_\mathrm{T}}K{\alpha_\mathrm{T}}^2+\tfrac{1}{m_Tv_\mathrm{T}}K\tfrac{a}{v_{\mathrm{T},0}}\dot{\psi}\alpha_\mathrm{T})\sin\delta-\tfrac{1}{m_Tv_\mathrm{T}}K\alpha_\mathrm{T}\delta\sin\delta\\
\ddot{\psi} &= \tfrac{1}{I_T}K\alpha_\mathrm{T}(a-b)+\tfrac{1}{I_T}K\tfrac{(a^2+b^2)}{v_{\mathrm{T},0}}\dot{\psi}-\tfrac{1}{I_T}K\delta a-(\tfrac{1}{2I_T}K\tfrac{a^2}{v_{\mathrm{T},0}}\dot{\psi}+\tfrac{1}{2I_T}K\alpha_\mathrm{T}a)\delta^2+\tfrac{1}{2I_T}Ka\delta^3
\end{aligned} \tag{87}$$

Observing the facts that $\sin\theta\approx\theta$ for $\theta\leq 13.99^o$ and $\cos\theta\approx 1-(\theta)^2\big/2$ for $\theta\leq 37.93^\circ$ and the facts that $0^\circ\leq\alpha_\mathrm{T}\leq 10^\circ$ for race and high-performance tires and the number is a little lower for street tires, and assuming $|\delta|\leq 35^\circ$, and some manipulations, (87) can be rewritten into the

Table 1: Vehicle parameters.

Item	Value	Description
m_T	1300.0 kg	Total mass
I_T	$10^4 \text{kg} \cdot \text{m}^2$	Moment of inertia
K	4×10^4	Tires stiffness
$\alpha_{T,0}$	0 rad	Initial vehicle sideslip angle
$\dot{\Psi}_0$	0.01 rad/s	Initial yaw rate
v_0	20 m/s	Initial velocity
a	1.6154	Distance to CG
b	1.8846	Distance to CG

following affine in control form

$$\dot{X} = \begin{bmatrix} \dot{v}_{\mathrm{T}} \\ \dot{\alpha}_{\mathrm{T}} \\ \ddot{\psi} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{2}{m_T} K {\alpha_{\mathrm{T}}}^2 + \frac{1}{m_T} K \frac{(a-b)}{v_{\mathrm{T},0}} \dot{\psi} \alpha_{\mathrm{T}} \\ \frac{2}{m_T v_{\mathrm{T}}} K \alpha_{\mathrm{T}} + \frac{1}{m_T v_{\mathrm{T}}} K \frac{(a-b)}{v_{\mathrm{T},0}} \dot{\psi} - \frac{1}{m_T v_{\mathrm{T}}} K \frac{(a-b)}{v_{\mathrm{T},0}} \dot{\psi} \frac{{\alpha_{\mathrm{T}}}^2}{2} - \frac{2}{m_T v_{\mathrm{T}}} K \frac{{\alpha_{\mathrm{T}}}^3}{2} - \dot{\psi} \\ \frac{1}{I_T} K \alpha_{\mathrm{T}} (a-b) + \frac{1}{I_T} K \frac{(a^2+b^2)}{v_{\mathrm{T},0}} \dot{\psi} \end{bmatrix}}_{\mathscr{I}(.)}$$

$$+ \underbrace{\begin{bmatrix} \mathscr{G}_{11} & \mathscr{G}_{12} & \mathscr{G}_{13} & \mathscr{G}_{14} & \mathscr{G}_{15} \\ \mathscr{G}_{21} & \mathscr{G}_{22} & \mathscr{G}_{23} & \mathscr{G}_{24} & \mathscr{G}_{25} \\ \mathscr{G}_{31} & \mathscr{G}_{32} & \mathscr{G}_{33} & \mathscr{G}_{34} & \mathscr{G}_{35} \end{bmatrix}}_{\mathscr{G}(.)} \underbrace{\begin{bmatrix} \delta \\ \sin\delta \\ \delta \sin\delta \\ \delta^2 \\ \delta^3 \end{bmatrix}}_{\mathscr{U}(.)} \tag{88}$$

where $\mathscr{G}_{11} = \frac{-K\alpha_{\mathrm{T}}}{m_T}$, $\mathscr{G}_{12} = \frac{-Ka}{m_T v_{\mathrm{T},0}} \dot{\psi} + \frac{Ka{\alpha_{\mathrm{T}}}^2}{2m_T v_{\mathrm{T},0}} \dot{\psi} - \frac{K\alpha_{\mathrm{T}}}{m_T} + \frac{K{\alpha_{\mathrm{T}}}^3}{2m_T}$, $\mathscr{G}_{13} = \frac{K}{m_T} - \frac{K{\alpha_{\mathrm{T}}}^2}{2m_T}$, $\mathscr{G}_{14} = \frac{-K{\alpha_{\mathrm{T}}}^2}{2m_T} - \frac{Ka\alpha_{\mathrm{T}}}{2m_T v_{\mathrm{T},0}} \dot{\psi}$, $\mathscr{G}_{15} = \frac{K\alpha_{\mathrm{T}}}{2m_T}$, $\mathscr{G}_{21} = \frac{-K}{m_T v_{\mathrm{T}}} + \frac{K{\alpha_{\mathrm{T}}}^2}{2m_T v_{\mathrm{T}}}$, $\mathscr{G}_{22} = \frac{K{\alpha_{\mathrm{T}}}^2}{m_T v_{\mathrm{T}}} + \frac{Ka\alpha_{\mathrm{T}}}{m_T v_{\mathrm{T}} v_{\mathrm{T},0}} \dot{\psi}$, $\mathscr{G}_{23} = -\frac{K\alpha_{\mathrm{T}}}{m_T v_{\mathrm{T}}}$, $\mathscr{G}_{24} = \frac{K{\alpha_{\mathrm{T}}}^3}{4m_T v_{\mathrm{T}}} - \frac{Ka}{2m_T v_{\mathrm{T}} v_{\mathrm{T},0}} \dot{\psi} + \frac{Ka{\alpha_{\mathrm{T}}}^2}{4m_T v_{\mathrm{T}} v_{\mathrm{T},0}} \dot{\psi} - \frac{K\alpha_{\mathrm{T}}}{2m_T v_{\mathrm{T}}}$, $\mathscr{G}_{25} = \frac{K}{2m_T v_{\mathrm{T}}} - \frac{K{\alpha_{\mathrm{T}}}^2}{4m_T v_{\mathrm{T}}}$, $\mathscr{G}_{31} = \frac{-Ka}{I_T}$, $\mathscr{G}_{32} = 0$, $\mathscr{G}_{33} = 0$, $\mathscr{G}_{34} = \frac{-Ka^2}{2I_T v_{\mathrm{T},0}} \dot{\psi} - \frac{K\alpha_{\mathrm{T}} a}{2I_T}$, and $\mathscr{G}_{35} = \frac{Ka}{2I_T}$. Moreover, $\mathscr{I}(\mathscr{X})$, $\mathscr{G}(\mathscr{X})$, and $\mathscr{U}(\delta)$ are the drift dynamics of the system, the control input, and the input dynamics of the system, respectively.

The vehicle model for this study is a nonlinear single-track bicycle given in (88) and shown in Fig. 8. It is noteworthy that this model is capable of qualitatively describing a simple nonlinear vehicle yaw dynamics in all operating conditions corresponding to $|\alpha_{\mathrm{T}}| \leq 0.174533\, rad$, $|\psi| \leq 0.3\, rad$, $|\dot{\psi}| \leq 0.7\, rad/\mathrm{s}$, and $|\delta| \leq 37.93°$.

Assuming that $v_{\mathrm{T}} \geq 1 m/s$, and let the observable functions consist of the elements of the state vector $\mathscr{X}$ as:

$$\mathscr{C} = \{\psi, \alpha_T, \dot{\psi}, {\alpha_T}^2, {\alpha_{\mathrm{T}}}^3, {\alpha_{\mathrm{T}}}^4, {\alpha_{\mathrm{T}}}^5, \frac{\alpha_T}{v_{\mathrm{T}}}, \frac{\alpha_T}{v_{\mathrm{T}}}, \frac{{\alpha_T}^2}{v_{\mathrm{T}}}, \frac{{\alpha_T}^3}{v_{\mathrm{T}}}, \frac{{\alpha_T}^4}{v_{\mathrm{T}}}, \frac{{\alpha_T}^5}{v_{\mathrm{T}}}, \alpha_T \dot{\psi},$$
$$ {\alpha_{\mathrm{T}}}^2 \dot{\psi}, {\alpha_{\mathrm{T}}}^3 \dot{\psi}, {\alpha_{\mathrm{T}}}^4 \dot{\psi}, {\alpha_{\mathrm{T}}}^5 \dot{\psi}, \frac{1}{v_{\mathrm{T}}} \dot{\psi}, \frac{\alpha_T}{v_{\mathrm{T}}} \dot{\psi}, \frac{{\alpha_T}^2}{v_{\mathrm{T}}} \dot{\psi}, \frac{{\alpha_T}^3}{v_{\mathrm{T}}} \dot{\psi}, \frac{{\alpha_T}^4}{v_{\mathrm{T}}} \dot{\psi}, \frac{{\alpha_T}^5}{v_{\mathrm{T}}} \dot{\psi}\} \tag{89}$$

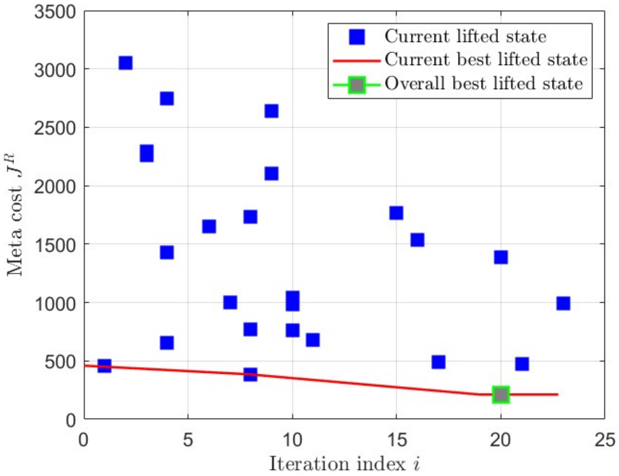

Figure 9: Meta cost J^R vs iteration i of Algorithm 1. At each iteration i, the cost of the current set of observable functions and of the current best one are depicted. The green square shows the overall best lifted state $\xi_\theta = [\psi, \alpha_T, \dot{\psi}, {\alpha_T}^2, {\alpha_\mathrm{T}}^3, \frac{\alpha_T}{v_\mathrm{T}}, \frac{{\alpha_T}^2}{v_\mathrm{T}}, \frac{{\alpha_T}^3}{v_\mathrm{T}}, \alpha_T\dot{\psi}, {\alpha_\mathrm{T}}^2\dot{\psi}, \frac{1}{v_\mathrm{T}}\dot{\psi}, \frac{\alpha_T}{v_\mathrm{T}}\dot{\psi}, \frac{{\alpha_T}^2}{v_\mathrm{T}}\dot{\psi}]^\top$ which is found at iteration 20.

The meta cost J^R vs iteration i of Algorithm 1 is illustrated in Fig. 9. Note that for the lifted state $\xi_\theta = [\psi, \alpha_T, \dot{\psi}, {\alpha_T}^2, {\alpha_\mathrm{T}}^3, \frac{\alpha_T}{v_\mathrm{T}}, \frac{{\alpha_T}^2}{v_\mathrm{T}}, \frac{{\alpha_T}^3}{v_\mathrm{T}}, \alpha_T\dot{\psi}, {\alpha_\mathrm{T}}^2\dot{\psi}, \frac{1}{v_\mathrm{T}}\dot{\psi}, \frac{\alpha_T}{v_\mathrm{T}}\dot{\psi}, \frac{{\alpha_T}^2}{v_\mathrm{T}}\dot{\psi}]^\top$, one has

$$\begin{aligned}\dot{\xi}_\theta &= \hat{A}^*\xi_\theta + \hat{B}_1^*\xi_\theta\delta + \hat{B}_2^*\xi_\theta sin(\delta) + \hat{B}_3^*\xi_\theta\delta sin(\delta) + \hat{B}_4^*\xi_\theta\delta^2 + \hat{B}_5^*\xi_\theta\delta^3 + \hat{\mathscr{P}}_\theta(t, \xi_\theta(t)) \\ &= \hat{A}^*\xi_\theta + \underbrace{\begin{bmatrix} I_{n_{\xi_\theta}} & I_{n_{\xi_\theta}} & I_{n_{\xi_\theta}} & I_{n_{\xi_\theta}} & I_{n_{\xi_\theta}} \end{bmatrix} diag(\hat{B}_1^*, \hat{B}_2^*, \hat{B}_3^*, \hat{B}_4^*, \hat{B}_5^*)}_{\hat{B}_\theta^*} \underbrace{(I_5 \otimes \xi_\theta)\begin{bmatrix} \delta \\ \sin\delta \\ \delta\sin\delta \\ \delta^2 \\ \delta^3 \end{bmatrix}}_{\Psi(\mathscr{U}(\delta(t)))} + \hat{\mathscr{P}}_\theta(t, \xi_\theta(t))\end{aligned} \tag{90}$$

$$\begin{aligned}\dot{v}_\mathrm{T} = &\left[0, 0, 0, \frac{2K}{m_T}, 0, 0, 0, 0, \frac{K(a-b)}{m_T v_{\mathrm{T},0}}, 0, 0, 0, 0\right]^\top \xi_\theta \\ &+ \begin{bmatrix} \bar{\mathscr{G}}_{11} & 0 & 0 & 0 & 0 \\ 0 & \bar{\mathscr{G}}_{12} & 0 & 0 & 0 \\ 0 & 0 & \bar{\mathscr{G}}_{13} & 0 & 0 \\ 0 & 0 & 0 & \bar{\mathscr{G}}_{14} & 0 \\ 0 & 0 & 0 & 0 & \bar{\mathscr{G}}_{15} \end{bmatrix} (I_5 \otimes \xi_\theta) \begin{bmatrix} \delta \\ \sin\delta \\ \delta\sin\delta \\ \delta^2 \\ \delta^3 \end{bmatrix}\end{aligned} \tag{91}$$

where $\bar{\mathscr{G}}_{11} = \left[0, \frac{-K}{m_T}, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\right]$, $\bar{\mathscr{G}}_{12} = \left[0, \frac{-K}{m_T}, \frac{-Ka}{m_T v_{\mathrm{T},0}}, 0, \frac{K}{2m_T}, 0, 0, 0, 0, \frac{Ka}{2m_T v_{\mathrm{T},0}}, 0, 0, 0\right]$, $\bar{\mathscr{G}}_{13} = \left[0, \frac{K}{m_T}, 0, \frac{-K}{2m_T}, 0, 0, 0, 0, 0, 0, 0, 0, 0\right]$, $\bar{\mathscr{G}}_{14} = \left[0, 0, 0, \frac{-K}{2m_T}, 0, 0, 0, 0, \frac{-Ka}{2m_T v_{\mathrm{T},0}}, 0, 0, 0, 0\right]$, and $\bar{\mathscr{G}}_{15} = \left[0, \frac{K}{2m_T}, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\right]$.

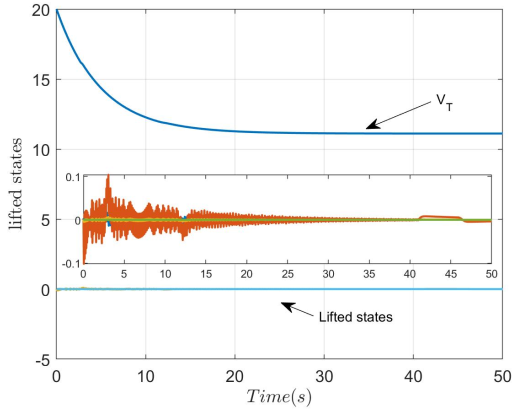

Figure 10: Evolution of the lifted state ξ_θ and v_T trajectories.

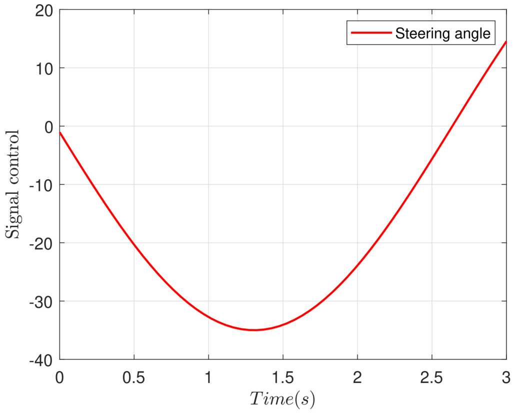

Figure 11: Control input $\delta = -35Sin(1.2t)$ Degree.

To satisfy Condition 1, a rich enough input signal δ is injected into the nonlinear vehicle system (88) for excitation of the lifted states ξ_θ. The evolution of the lifted states trajectories under the rich enough input signal δ is shown in Fig. 10. The proposed update law (34)-(36) is used to find the approximated matrices $\hat{A}^*_\theta$, and $\hat{B}^*_\theta$. After the convergence of $\hat{A}^*_\theta$ and $\hat{B}^*_\theta$, the control input δ, given in Fig. 11, is employed for the identified linear representation system (90) and the nonlinear vehicle system (88) to compare the evolution of the states trajectories of the identified model and the nonlinear vehicle system (88). Using the equations of motion (Andrzejewski and Awrejcewicz, 2006; de Souza Mendes et al., 2016)

$$\begin{aligned} \dot{\mathrm{X}} &= \mathrm{v}_T \cos(\alpha_T + \psi), \\ \dot{\mathrm{Y}} &= \mathrm{v}_T \sin(\alpha_T + \psi). \end{aligned} \tag{92}$$

Fig. 13, illustrates the comparison of $X - Y$ positions of the nonlinear vehicle system (88) and the identified model (90).

Now, to test the learned model, we assumed that a disturbance as a single steering pulse with the magnitude of $\delta = 30°$ is simulated at $t = 0 - 0.2$ sec. The control input δ as depicted in Fig. 14

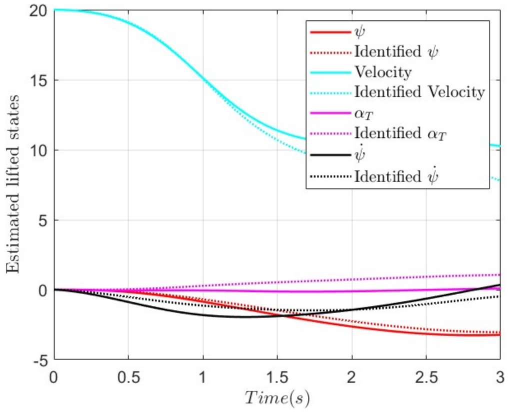

Figure 12: States trajectories of the nonlinear vehicle system and the identified model under control input of $\delta = -35Sin(1.2t)$ Degree.

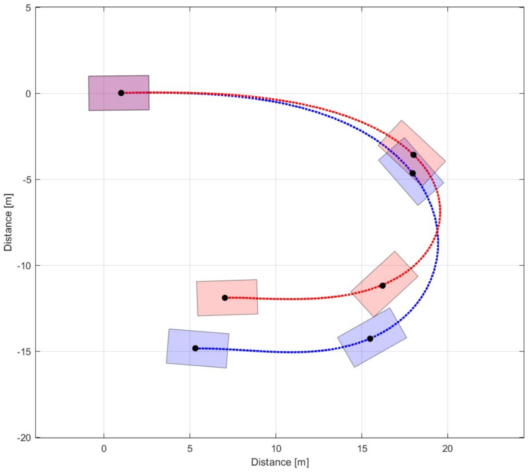

Figure 13: X and Y position trajectories of the nonlinear vehicle system (Light-Blue square) and the identified model (Light-Coral square) under control input of $\delta = -35Sin(1.2t)$ Degree.

is employed for the nonlinear vehicle system (88) to stabilize the states $\{\psi, \alpha_T, \dot{\psi}\}$. For the comparison, the feedback gain K is also applied to the identified model. Fig. 15 depicts the evolution of state trajectories of identified (dashed line) and real (solid line) nonlinear system, which shows that the both state trajectories. Fig. 16, illustrates the comparison of $X - Y$ positions of the nonlinear vehicle system (9) and the identified model (72) under the control input given in Fig. 16.

6. Conclusion

A novel data-driven learning algorithm is presented to learn a linear representation of nonlinear system dynamics using Koopman operator theory. To jointly learn the set of observables (structure) and Koopman parameters, a bilevel learning mechanism with two layers of learning is developed. The lower layer learner leverages a unified batch-online learning-based finite-time Koopman identifier,

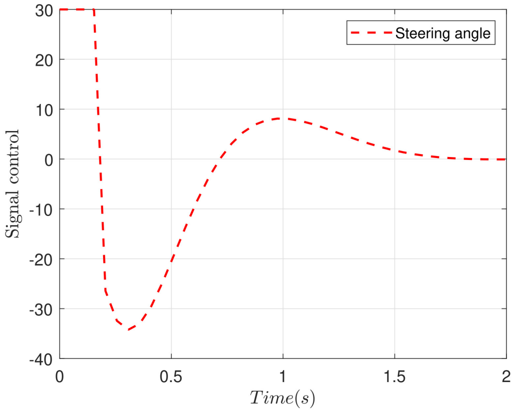

Figure 14: Control input δ.

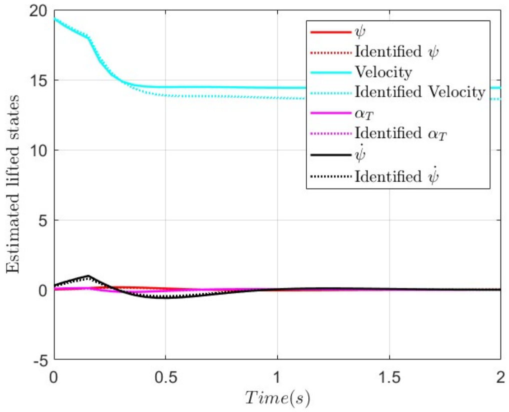

Figure 15: State trajectories of identified (dashed line) and real (solid line) nonlinear system with δ.

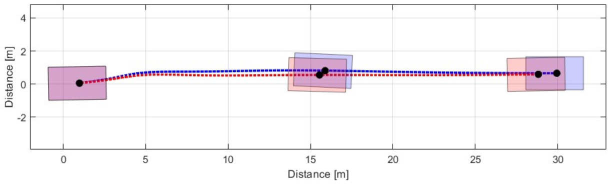

Figure 16: X and Y position trajectories of the nonlinear vehicle system (Light-Blue square) and the identified model (Light-Coral square) with control input of δ.

which uses discontinuous gradient update laws to minimize the instantaneous Koopman operator's identification errors as well as the identification errors for a batch of past samples collected in a history stack. It is guaranteed that the lower layer identifier will converge in a finite time under easy-to-verify conditions on a batch of samples. A higher layer employs a discrete Bayesian optimization algorithm to find a set of observables with minimum approximation errors and minimum cardinality. Finally, the effectiveness of the proposed framework was verified on a simulation example.

Acknowledgments

This work was supported by Ford Motor Company-Michigan State University Alliance.

References

Ryszard Andrzejewski and Jan Awrejcewicz. *Nonlinear dynamics of a wheeled vehicle*, volume 10. Springer Science & Business Media, 2006.

Hamed Ayoobi, Ming Cao, Rineke Verbrugge, and Bart Verheij. Argumentation-based online incremental learning. *IEEE Transactions on Automation Science and Engineering*, pages 1–15, 2021.

Andrea Bacciotti and Francesca Ceragioli. Stability and stabilization of discontinuous systems and nonsmooth lyapunov functions. *ESAIM: Control, Optimisation and Calculus of Variations*, 4: 361–376, 1999.

Craig Bakker, Steven Rosenthal, and Kathleen E Nowak. Koopman representations of dynamic systems with control. *arXiv preprint arXiv:1908.02233*, 2019.

Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, and Filip Železnỳ. *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part I*, volume 8188. Springer, 2013.

Eric Brochu, Vlad M Cora, and Nando De Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

Steven L. Brunton, B. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control. *PLoS ONE*, 11, 2016a.

Steven L Brunton, Bingni W Brunton, Joshua L Proctor, and J Nathan Kutz. Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control. *PloS one*, 11(2):e0150171, 2016b.

Marko Budišić, Ryan Mohr, and Igor Mezić. Applied koopmanism. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(4):047510, 2012.

Zhengcai Cao, Qing Xiao, Ran Huang, and Mengchu Zhou. Robust neuro-optimal control of underactuated snake robots with experience replay. *IEEE transactions on neural networks and learning systems*, 29(1):208–217, 2017.

Guopei Chen, Feiqi Deng, and Ying Yang. Practical finite-time stability of switched nonlinear time-varying systems based on initial state-dependent dwell time methods. *Nonlinear Analysis: Hybrid systems*, 41:101031, 2021.

Girish Chowdhary and Eric Johnson. Concurrent learning for convergence in adaptive control without persistency of excitation. In *49th IEEE Conference on Decision and Control (CDC)*, pages 3674–3679. IEEE, 2010.

Girish Chowdhary and Eric Johnson. A singular value maximizing data recording algorithm for concurrent learning. In *Proceedings of the 2011 American Control Conference*, pages 3547–3552. IEEE, 2011.

Girish Chowdhary, Tansel Yucelen, Maximillian Mühlegg, and Eric N Johnson. Concurrent learning adaptive control of linear systems with exponentially convergent bounds. *International Journal of Adaptive Control and Signal Processing*, 27(4):280–301, 2013.

Jorge Cortés. Discontinuous dynamical systems. *IEEE Control Systems*, 28, 2008.

André de Souza Mendes, Douglas De Rizzo Meneghetti, Marko Ackermann, and Agenor de Toledo Fleury. Vehicle dynamics-lateral: Open source simulation package for matlab. Technical report, SAE Technical Paper, 2016.

Zlatko Drmač, Igor Mezić, and Ryan Mohr. Identification of nonlinear systems using the infinitesimal generator of the koopman semigroup—a numerical implementation of the mauroy–goncalves method. *Mathematics*, 9(17):2075, 2021.

A. F. Filippov and F. M. Arscott. *Differential Equations with Discontinuous Righthand Sides: Control Systems (Mathematics and its Applications, 18)*. Dordrecht, Netherlands: Kluwer Academic Publishers Group., 1988.

Aleksei Fedorovich Filippov. *Differential equations with discontinuous righthand sides: control systems*, volume 18. Springer Science & Business Media, 2013.

Yiqiang Han, Wenjian Hao, and Umesh Vaidya. Deep learning of koopman representation for control. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 1890–1895. IEEE, 2020.

Shuping He and Jun Song. Finite-time sliding mode control design for a class of uncertain conic nonlinear systems. *IEEE/CAA Journal of Automatica Sinica*, 4(4):809–816, 2017.

Bowen Huang, Xu Ma, and Umesh Vaidya. Data-driven nonlinear stabilization using koopman operator. In *The Koopman Operator in Systems and Control*, pages 313–334. Springer, 2020.

Sumit Kumar Jha, Sayan Basu Roy, and Shubhendu Bhasin. Initial excitation-based iterative algorithm for approximate optimal control of completely unknown lti systems. *IEEE Transactions on Automatic Control*, 64(12):5230–5237, 2019.

Lan Jiang, Hongyun Huang, and Zuohua Ding. Path planning for intelligent robots based on deep q-learning with experience replay and heuristic knowledge. *IEEE/CAA Journal of Automatica Sinica*, 7(4):1179–1189, 2019.

Eurika Kaiser, J Nathan Kutz, and Steven Brunton. Data-driven discovery of koopman eigenfunctions for control. *Machine Learning: Science and Technology*, 2021.

Rushikesh Kamalapurkar, Benjamin Reish, Girish Chowdhary, and Warren E Dixon. Concurrent learning for parameter estimation using dynamic state-derivative estimators. *IEEE Transactions on Automatic Control*, 62(7):3594–3601, 2017.

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

J Nathan Kutz, Joshua L Proctor, and Steven L Brunton. Koopman theory for partial differential equations. *arXiv preprint arXiv:1607.07076*, 2016.

Devon Lehrer, Veronica Adetola, and Martin Guay. Parameter identification methods for non-linear discrete-time systems. In *Proceedings of the 2010 American Control Conference*, pages 2170–2175. IEEE, 2010.

Qianxiao Li, Felix Dietrich, Erik M Bollt, and Ioannis G Kevrekidis. Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the koopman operator. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27(10):103111, 2017.

Quan Liu, Xin Zhou, Fei Zhu, Qiming Fu, and Yuchen Fu. Experience replay for least-squares policy iteration. *IEEE/CAA Journal of Automatica Sinica*, 1(3):274–281, 2014.

Yang Liu, Xiaoping Liu, Yuanwei Jing, and Ziye Zhang. Semi-globally practical finite-time stability for uncertain nonlinear systems based on dynamic surface control. *International Journal of Control*, 94(2):476–485, 2021.

Wenlian Lu, Xiwei Liu, and Tianping Chen. A note on finite-time and fixed-time stability. *Neural Networks*, 81:11–15, 2016.

Phuc Luong, Sunil Gupta, Dang Nguyen, Santu Rana, and Svetha Venkatesh. Bayesian optimization with discrete variables. In *Australasian Joint Conference on Artificial Intelligence*, pages 473–484. Springer, 2019.

Mohamed Maghenem and Ricardo G Sanfelice. Barrier function certificates for forward invariance in hybrid inclusions. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 759–764. IEEE, 2018.

Alexandre Mauroy and Jorge Goncalves. Koopman-based lifting techniques for nonlinear systems identification. *IEEE Transactions on Automatic Control*, 65(6):2550–2565, 2019.

Hamidreza Modares, Frank L Lewis, and Mohammad-Bagher Naghibi-Sistani. Adaptive optimal control of unknown constrained-input systems using policy iteration and neural networks. *IEEE transactions on neural networks and learning systems*, 24(10):1513–1525, 2013.

Marcos Netto and Lamine Mili. A robust data-driven koopman kalman filter for power systems dynamic state estimation. *IEEE Transactions on Power Systems*, 33(6):7228–7237, 2018.

Brad Paden and Shankar Sastry. A calculus for computing filippov's differential inclusion with application to the variable structure control of robot manipulators. *IEEE transactions on circuits and systems*, 34(1):73–82, 1987.

Anup Parikh, Rushikesh Kamalapurkar, and Warren E Dixon. Integral concurrent learning: Adaptive control with parameter convergence using finite excitation. *International Journal of Adaptive Control and Signal Processing*, 33(12):1775–1787, 2019.

Joshua L Proctor, Steven L Brunton, and J Nathan Kutz. Generalizing koopman theory to allow for inputs and control. *SIAM Journal on Applied Dynamical Systems*, 17(1):909–930, 2018.

Orlando Romero and Mouhacine Benosman. Robust time-varying continuous-time optimization with pre-defined finite-time stability. *IFAC-PapersOnLine*, 53:6743–6748, 2020a.

Orlando Romero and Mouhacine Benosman. Finite-time convergence in continuous-time optimization. In *International Conference on Machine Learning*, pages 8200–8209. PMLR, 2020b.

Orlando Romero and Mouhacine Benosman. Time-varying continuous-time optimisation with pre-defined finite-time stability. *International Journal of Control*, 94:3237 – 3254, 2021.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1): 148–175, 2015.

Ruizhuo Song and Qinglai Wei. Neural-network-based approach for finite-time optimal control. In *Adaptive Dynamic Programming: Single and Multiple Controllers*, pages 7–23. Springer, 2019.

Gang Tao. *Adaptive control design and analysis*, volume 37. John Wiley & Sons, 2003.

Farzaneh Tatari, Mohammad-Bagher Naghibi-Sistani, and Kyriakos G Vamvoudakis. Distributed optimal synchronization control of linear networked systems under unknown dynamics. In *2017 American Control Conference (ACC)*, pages 668–673. IEEE, 2017.

Farzaneh Tatari, Kyriakos G Vamvoudakis, and Majid Mazouchi. Optimal distributed learning for disturbance rejection in networked non-linear games under unknown dynamics. *IET Control Theory & Applications*, 13(17):2838–2848, 2018.

Farzaneh Tatari, Christoforos Panayiotou, and Marios M. Polycarpou. Finite-time identification of unknown discrete-time nonlinear systems using concurrent learning. *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 2306–2311, 2021.

Cameron Thieme. Multiflows: A new technique for filippov systems and differential inclusions. *arXiv preprint arXiv:1905.07051*, 2019.

Jonathan H Tu, Clarence W Rowley, Dirk M Luchtenburg, Steven L Brunton, and J Nathan Kutz. On dynamic mode decomposition: Theory and applications. *arXiv preprint arXiv:1312.0041*, 2013.

Amin Vahidi-Moghaddam, Majid Mazouchi, and Hamidreza Modares. Memory-augmented system identification with finite-time convergence. *IEEE Control Systems Letters*, 5(2):571–576, 2020.

Kyriakos G Vamvoudakis, Marcio Fantini Miranda, and João P Hespanha. Asymptotically stable adaptive–optimal control algorithm with saturating actuators and relaxed persistence of excitation. *IEEE transactions on neural networks and learning systems*, 27(11):2386–2398, 2015.

Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

Patrick Walters, Rushikesh Kamalapurkar, Forrest Voight, Eric M Schwartz, and Warren E Dixon. Online approximate optimal station keeping of a marine craft in the presence of an irrotational current. *IEEE Transactions on Robotics*, 34(2):486–496, 2018.

Matthew O Williams, Ioannis G Kevrekidis, and Clarence W Rowley. A data–driven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015.

Matthew O Williams, Maziar S Hemati, Scott TM Dawson, Ioannis G Kevrekidis, and Clarence W Rowley. Extending data-driven koopman analysis to actuated systems. *IFAC-PapersOnLine*, 49 (18):704–709, 2016.

Yuanqing Xia, Jinhui Zhang, Kunfeng Lu, and Ning Zhou. Finite-time tracking control of rigid spacecraft under actuator saturations and faults. In *Finite Time and Cooperative Control of Flight Vehicles*, pages 141–169. Springer, 2019.

Xiong Yang and Haibo He. Adaptive critic learning and experience replay for decentralized event-triggered control of nonlinear interconnected systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 50(11):4043–4055, 2019.

Xiong Yang, Derong Liu, and Qinglai Wei. Near-optimal online control of uncertain nonlinear continuous-time systems based on concurrent learning. In *2014 International Joint Conference on Neural Networks (IJCNN)*, pages 231–238. IEEE, 2014.

Yongliang Yang, Kyriakos G Vamvoudakis, Hamidreza Modares, Yixin Yin, and Donald C Wunsch. Safe intermittent reinforcement learning with static and dynamic event generators. *IEEE transactions on neural networks and learning systems*, 31(12):5441–5455, 2020.

Enoch Yeung, Soumya Kundu, and Nathan Hodas. Learning deep neural network representations for koopman operators of nonlinear dynamical systems. In *2019 American Control Conference (ACC)*, pages 4832–4839. IEEE, 2019.

Qichao Zhang, Dongbin Zhao, and Ding Wang. Event-based robust control for uncertain nonlinear systems using adaptive dynamic programming. *IEEE transactions on neural networks and learning systems*, 29(1):37–50, 2016.

Dongbin Zhao, Qichao Zhang, Ding Wang, and Yuanheng Zhu. Experience replay for optimal control of nonzero-sum game systems with unknown dynamics. *IEEE transactions on cybernetics*, 46(3):854–865, 2015.

Zhijia Zhao and Zhijie Liu. Finite-time convergence disturbance rejection control for a flexible timoshenko manipulator. *IEEE/CAA Journal of Automatica Sinica*, 8(1):157–168, 2020.