Data-Driven Feedback Linearization Using the Koopman Generator

Darshan Gadginmath †, Graduate Student Member, IEEE, Vishaal Krishnan †, and Fabio Pasqualetti †, Member, IEEE

Abstract—This article contributes a theoretical framework for data-driven feedback linearization of nonlinear control-affine systems. We unify the traditional geometric perspective on feedback linearization with an operator-theoretic perspective involving the Koopman operator. We first show that if the distribution of the control vector field and its repeated Lie brackets with the drift vector field is involutive, then there exists an output and a feedback control law for which the Koopman generator is finite-dimensional and locally nilpotent. We use this connection to propose a data-driven algorithm ‘Koopman generator-based feedback linearization (KGFL)’ for feedback linearization of single-input systems. Particularly, we use experimental data to identify the state transformation and control feedback from a dictionary of functions for which feedback linearization is achieved in a least-squares sense. We also propose a single-step data-driven formula which can be used to compute the linearizing transformations. When the system is feedback linearizable and the chosen dictionary is complete, our data-driven algorithm provides the same solution as model-based feedback linearization. Finally, we provide numerical examples for the data-driven algorithm and compare it with model-based feedback linearization. We also numerically study the effect of the richness of the dictionary and the size of the dataset on the effectiveness of feedback linearization.

Index Terms—Data-driven control, feedback linearization, geometric control, Koopman operator.

I. INTRODUCTION

Nonlinear control methods rooted in model-based approaches have received considerable attention [1]. Among these techniques, feedback linearization has emerged as a prominent strategy, offering the implementation of straightforward linear control methodologies to nonlinear systems. However, a notable limitation of this approach is its demand for a comprehensive knowledge of the system dynamics. Consequently, inadequate system identification in the context of complex, high-dimensional cyber–physical systems can lead to poor control performance. On the contrary, machine learning methodologies [2], [3], [4] offer a robust alternative, enabling the utilization of experimental data acquired from the system to facilitate feedback control, even in the absence of prior knowledge regarding the underlying system’s dynamics. Nevertheless, these machine learning methods frequently fall short of providing comprehensive insights into both their own performance and the intricate nature of the systems they operate on.

Furthermore, a detailed characterization of their limitations remains a subject of ongoing investigation, and the pursuit of a systematic framework for nonlinear data-driven control remains unresolved. Recently, significant attention has been directed towards the Koopman operator [5] due to its capacity to furnish a global (infinite-dimensional) linear representation of autonomous nonlinear systems. In [6], EDMD is proposed to approximate the linear action of the Koopman operator on a finite-dimensional space of observables using data. The commonality between feedback linearization and Koopman-based methods pertains to the concept of complete linearization, one for actuated systems and the other for autonomous systems. This connection has hitherto remained unexplored in the existing literature. In this work, we bridge the gap between the conventional technique of feedback linearization and the Koopman operator. Furthermore, leveraging this newfound connection, we develop a data-driven methodology capable of yielding valuable insights into the dynamics inherent to the system.

A. Problem Setup

We consider a continuous-time nonlinear control-affine system, with single input, of the form

$$\dot{x} = f(x) + g(x)u$$

(1)

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}$ is the control input, and $f, g : \mathbb{R}^n \to \mathbb{R}^n$ are the drift and control vector fields. In the data-driven setting, we do not have access to the drift and control vector fields $f, g$, but instead have access to $N$ data samples collected from a control experiment on System (1). The state and control trajectory during the experiment is $x_t, u_t$ where $t \in \mathbb{R}_{\geq 0}$. The data collected from an experiment are represented as matrices $X, U$ as follows:

$$X = [x_0, x_1, \ldots, x_N], \quad U = [u_1, u_2, \ldots, u_N]$$

where $x_i$ and $u_i$ are the sample at the $i$th instance of the experiment.

Our objective is to transform system (1) to a target linear system $\dot{z} = Az + Bu$, where $z$ and $u$ are transformed state and control, respectively. Motivated by model-based feedback linearization, we propose to transform the state as $z = H(x)$ and the control as $u = \alpha(x) + \beta(x)v$. We seek to identify the transformations $H, \alpha, \text{and } \beta$ using the data $X, U$.

B. Related Work

A comprehensive introduction to feedback linearization can be found in [1]. This technique provides a systematic method to identify the necessary state and control transformations in the model-based case. It is crucial to note that these transformations are dependent on the dynamics of the system and not all systems allow for feedback linearization. An approximate, but still model-based, approach for feedback linearization was proposed in [7]. These methods cannot be employed without a prior system identification step. Several works that combine learning methods for feedback linearization have been proposed [3], [4], [8], [9]. The works [3] and [4] primarily used neural networks to obtain state and control transformations, whereas the authors in [8] proposed a reinforcement learning approach. However, these methods do not...
provide a clear insight into the control and state transformations. In [10], a SISO full state-feedback linearizable system is considered and a data-driven solution is proposed by approximating the system using Taylor series. An extension of the Willems fundamental lemma for nonlinear systems is proposed in [11], which is used to present a predictive control methodology with data. However, a systematic approach to finding the state and control transformations in the data-driven setting for feedback linearization has not been addressed in the literature. In this work, we seek to establish a data-driven methodology for feedback linearization which not only provides a convenient solution but also insight into the dynamics of the system.

The main advantage of the Koopman operator is its ability to provide a global linear representation of a nonlinear system. However, its main drawback is its infinite-dimensional representation for only autonomous systems. Recent literature has focused on finite-dimensional approximations of the Koopman operator [6], [12]. Of particular interest is the gEDMD algorithm [13] which seeks a finite-dimensional approximation of the infinitesimal generator of the Koopman operator and is based on extended dynamic mode decomposition (EDMD) [6]. The gEDMD algorithm [13] used a dictionary of functions to lift full-state data from an autonomous system and seeks to find a linear relation in the evolution of the lifted system.

The works in [14], [15], [16], and [17] have focused on obtaining accurate finite-dimensional approximations of this linear operator for control. While [14], [15], and [16] have extended [6] for control, linear predictors for the control-affine nonlinear system are considered in [18]. In [17], it is shown that typical approximation of using LTI systems for Koopman-based methods as in [18] can be erroneous. They address this issue by converting the nonlinear system to a linear parameter-varying system with the control as the variable parameter. However, crucially, an understanding of when the LTI system approximation is less erroneous is yet to be fully understood. Further, the control transformations required for exact linearization and its connection to feedback linearization are absent. A Luenberger observer for the system’s nonlinearities is proposed using the Koopman operator in [19]. Here, the control is considered as a varying parameter, and the overall system is considered as a linear parameter varying system. Hence, existing literature that use the Koopman operator for control have crucially missed the connection to feedback linearization. Bilinearization using the Koopman operator has also been an area of interest [20], [21], [22]. In [20], the nonlinear system is approximated by interpolated bilinear systems. Then, a model predictive control scheme is applied to the identified interpolated bilinear model. Probabilistic error bounds on trajectories predicted by bilinearized models using the Koopman operator are given in [21]. Conditions for global bilinearizability using the Koopman operator are given in [22] by assuming that a finite number of Koopman eigenfunctions span a Koopman invariant subspace. The model-based feedback linearization approach and the modern data-driven Koopman operator approach are both linearization techniques, yet for controlled and autonomous systems, respectively. In this article, we focus on showing a connection between these two methods and developing a data-driven scheme for nonlinear control.

C. Contributions

The main contributions of this article are as follows. We first bridge the gap between the geometric framework of feedback linearization and the Koopman operator-theoretic framework. In particular, we show that, when the system is involutive to a certain degree, there exists an observable $h$ and a feedback control $\alpha$ such that the Koopman generator for the closed-loop system under the feedback $\alpha$ is nilpotent at the observable $h(x)$. Furthermore, there exists a finite-dimensional Koopman invariant subspace of the same dimension as the involutive distribution for the system. This connection to the Koopman operator allows us to develop a data-driven method for feedback linearization, by essentially casting the problem of data-driven feedback linearization as one of learning the closed-loop Koopman operator for the nonlinear control-affine system by a linearizing state/control transformation. To this end, we exploit the fact that involutivity permits a representation of the Koopman generator in the finite-dimensional Brunovsky canonical form under the linearizing state/control transformation. This allows us to fix the Brunovsky canonical form as the target linear representation and learn the linearizing transformation using a set of fixed dictionary functions by a least-squares method in our algorithm Koopman generator-based feedback linearization (KGFL). We also provide a numerical feedback linearization scheme with only input–output data. With input–output data, we show that the control transformations can be learned in a least-squares sense using a simple data-driven formula. The results in [23], which were developed independently from this work, deal with data-driven feedback linearization with complete dictionaries for fully feedback linearizable systems. In our work, we neither make the assumption of full feedback linearizability nor of complete dictionaries. The connection we establish with the Koopman generator allows us to develop a least-squares solution without the assumptions made in [23]. However, when the system is feedback linearizable and the dictionaries used in KGFL are complete, the solution is exact and is equal to the model-based solution. Finally, we demonstrate the performance of our algorithm with numerical simulations for multiple examples. We perform both full state feedback linearization and output feedback linearization on the Van der Pol oscillator and compare it against existing nonlinear data-driven control techniques. We consider a higher dimensional system with the control entering non-linearly and show that our algorithm can be used for complex systems. We also provide insight on the effect of richness of dictionary and data size on the accuracy of feedback linearization method.

II. PRELIMINARIES

Let $\mathbb{R}^d$ be the $d$-dimensional Euclidean space. Let $(\mathcal{X}, d_\mathcal{X})$ and $(\mathcal{Y}, d_\mathcal{Y})$ be metric spaces. A map $f : \mathcal{X} \rightarrow \mathcal{Y}$ is said to be Lipschitz with Lipschitz constant $\ell_f$ if $d_\mathcal{Y}(f(x_1), f(x_2)) \leq \ell_f d_\mathcal{X}(x_1, x_2)$ for any $x_1, x_2 \in \mathcal{X}$. The space of $k$-times continuously differentiable functions on $\mathcal{X}$ is denoted by $C^k(\mathcal{X})$. Let $V$ be a normed vector space and let $T : \mathcal{V} \rightarrow \mathcal{V}$ be a bounded linear operator on $V$. A subspace $W \subseteq V$ is said to be $T$-invariant if $T(W) \subseteq W$. The operator $T : \mathcal{V} \rightarrow \mathcal{V}$ is locally nilpotent with index $r$ at $v \in V$ if $T^r v \neq 0$ for all $k \in \{0, \ldots, r-1\}$ and $T^r v = 0$. Furthermore, if $\nu, T(\nu), \ldots, T^r(\nu)$ are linearly independent, then, span( $\nu, T(\nu), \ldots, T^r(\nu)$) is said to be a $T$-cyclic subspace of $T$. For a measure space $(\mathcal{X}, \Sigma, \mu)$ where $\Sigma$ is a sigma-algebra on $\mathcal{X}$ and $\mu$ is the measure on $(\mathcal{X}, \Sigma)$, a property $P$ is said to hold almost everywhere (a.e.) if the subset over which the property $P$ fails to hold is of $\mu$-measure zero. The Lie bracket between two vector fields $f$ and $g$ is denoted by $[f, g] = \text{ad}_f g = L_f g - L_g f$. The adjoint of order $k$ is defined recursively as $\text{ad}_f g = [f, \text{ad}_f (k-1) g]$ with $\text{ad}_f^0 g = g$. The Gateaux derivative $U(v; \eta)$ [24] of operator $T \in C^1(V, V)$ at $v \in V$ along $\eta \in V$ is given by

$$
\lim_{h \to 0^+} \frac{\|T(v + h\eta) - T(v) - U(v; \eta)h\|_V}{h} = 0.
$$

$W \subseteq V$ is called a stable subspace of $T$ with respect to perturbations along $\eta$ if $U(w; \eta) = 0$ for all $w \in W$. 

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
A. Lie Derivative as Koopman Generator

Consider the autonomous system \( \dot{x}(t) = f(x(t)) \) with bounded state space \( \mathbb{X} \subset \mathbb{R}^d \), where \( f : \mathbb{X} \to \mathbb{R}^d \) is Lipschitz. Let \( s : \mathbb{X} \times \mathbb{R}^d \to \mathbb{X} \) be the flow of the vector field \( f \), such that for any \( x \in \mathbb{X}, s(x, 0) = x \) and \( \frac{d}{dt}s(x, t) = f(s(x, t)) \). For a function \( \phi \in C^1(\mathbb{X}) \), let \( L_f \phi \) be the Lie derivative of \( \phi \) with respect to \( f \), such that for any \( x \in \mathbb{X}, L_f \phi \) uniquely satisfies

\[
\lim_{h \to 0^+} |\phi(s(x, h)) - \phi(x) - h(L_f \phi)(x)| = 0.
\]

Let \( \mathcal{K} : C^1(\mathbb{X}) \times \mathbb{R}^d \to C^1(\mathbb{X}) \) be the Koopman operator for the flow \( s \), such that for any \( x \in \mathbb{X} \) we have

\[
(\mathcal{K} \phi)(x) = \phi(s(x, t))
\]

where we have adopted the notation \( \mathcal{K}_t \phi = \mathcal{K}(\phi, t) \). The infinitesimal generator of the family of operators \( \{\mathcal{K}_t\}_{t \geq 0} \) is defined by the derivative operator \( \mathcal{L} \) if the following holds:

\[
\lim_{h \to 0^+} \|\mathcal{K}_h \phi - \mathcal{K}_0 \phi - h\mathcal{L} \phi\|_{L^2(\mathbb{X})} = 0.
\]

Since \( \mathcal{L} \) is called the infinitesimal generator of the family \( \{\mathcal{K}_t\}_{t \geq 0} \), and since \( \mathcal{K} \) is the Koopman operator, we thereby refer to \( \mathcal{L} \) as the Koopman generator. The following lemma establishes the relationship between the Koopman generator corresponding to the flow \( s \) and the Lie derivative \( L_f \) corresponding to the vector field \( f \).

**Lemma 2.1 (Lie derivative as Koopman generator):** Let \( f : \mathbb{X} \to \mathbb{R}^n \) be Lipschitz, with flow \( s : \mathbb{X} \times \mathbb{R}^d \to \mathbb{X} \), and let \( \phi \in C^1(\mathbb{X}) \). Then, almost everywhere in \( \mathbb{X} \), the Lie derivative \( L_f \phi \) is equal to the infinitesimal generator \( \mathcal{L} \) of the Koopman operator \( \mathcal{K} \) for the flow \( s \), i.e., \( L_f \phi = \mathcal{L} \phi \) a.e. in \( \mathbb{X} \) [25, Sec. 7.6].

We do not include the proof of Lemma 2.1 for the sake of brevity. We note that Lemma 2.1 establishes that the Lie derivative generator and the Koopman generator are equivalent (in an a.e. sense) over the space of continuously differentiable observable functions. We denote the closed-loop Koopman generator for System (1) with feedback \( u = \alpha(x) + L_f g(x) \). Note that the closed-loop Koopman generator is linear over the space of continuously differentiable functions \( C^1(\mathbb{X}) \), i.e., \( L_f g + g = L_f + \alpha \).

B. Feedback Linearization

Feedback linearization addresses the problem of designing a linearizing feedback controller in the model-based setting (given vector fields \( f \) and \( g \)).

An output function \( h(x) \in C^1(\mathbb{X}) \) of system (1) is said to have relative degree \( r \) [1] if \( L_g L_{f}^{k} h(x) = 0 \) for \( k = 0, \ldots, r - 2 \) and \( L_g L_{f}^{r-1} h(x) \neq 0 \). For an output function \( h \) of relative degree \( r \), the feedback linearizing transformation is given by

\[
z = H(x) = \begin{bmatrix} h(x) \\ L_f h(x) \\ \vdots \\ L_f^{(r-1)} h(x) \end{bmatrix}, \quad v = \alpha(x) + \beta(x) u \tag{2}
\]

yielding an \( r \)-dimensional linear system in the transformed state and control \((z, v)\) of the form

\[
z = Az + Bv, \quad s.a. A = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \tag{3}
\]

The above transformation results in an \((n-r)\)-dimensional residual dynamics, called zero dynamics, which is unobservable and uncontrollable, and the system is said to be input–output feedback linearizable. Therefore, this approach to linearization-based control relies crucially on the choice of the output function \( h \) which results in a stable zero dynamics. If the relative degree \( r = n \), the state space dimension, the system is said to be full-state feedback linearizable, which is the case iff it is both controllable and integrable [26, Theorem 6.2]. Furthermore, by the Frobenius theorem [26], a distribution of linearly independent vector fields \( f_1(x), f_2(x), \ldots, f_m(x) \) is completely integrable iff it is involutive.

Example 1 (Input–Output Feedback Linearizable System): Consider a system with the following vector fields:

\[
f(x) = \begin{bmatrix} 2x_2^3 \\ -1 \\ 0 \end{bmatrix}, \quad g(x) = \begin{bmatrix} -x_1 \\ -2x_2 \\ \frac{1}{2}x_3 \end{bmatrix}
\]

The distribution \( \Delta \) is

\[
\Delta = \{g, ad_g f, ad_g^2 f\} = \begin{bmatrix} -x_1 \\ -2x_2 \\ 2 \\ \frac{1}{2}x_3 \\ 0 \end{bmatrix}
\]

It can be noticed that \( \Delta \) has a rank equal to 2 for all \( x \). Further, \( ad_g^k g = 0 \) for all \( k \geq 2 \). Thus, \( \{g, ad_g f, ad_g^2 f\} \) has rank 2 for all \( k \) and \( x \in \mathbb{R}^3 \setminus 0 \), and \( \Delta \) is involutive. However, note that the system is not controllable. Therefore, the system is not full-state feedback linearizable and only input–output feedback linearizable.

III. DATA-DRIVEN FEEDBACK LINEARIZATION

In this section, we propose a data-driven technique, called KGFL, to perform data-driven feedback linearization to stabilize System (1). We first establish that there exists an observable and feedback control that render the closed-loop Koopman generator finite-dimensional. We then seek to find this transformation using experimental data.

A. Koopman Generator-Based Feedback Linearization

We now establish the connection between feedback linearization and the closed-loop Koopman generator \( L_f g + g \), which will serve as the basis for the numerical algorithm to determine the linearizing state/control transformation. To this end, the following theorem establishes the relationship between the geometry of the system, as manifested in the involutivity of the distribution \( \Delta = \text{span}\{g, ad_g f, \ldots, ad_g^{r-1} f\} \), and \( \mathcal{K} \) an Koopman invariant subspace.

The system (1) is said to be controllable when the distribution \( \Delta(x) = \text{span}\{g(x), ad_g f(x), \ldots, ad_g^{r-1} f(x)\} \) is such that \( \dim(\Delta(x)) = n \), for all \( x \in \mathbb{R}^n \).

A set of linearly independent vector fields \( \{f_1(x), f_2(x), \ldots, f_m(x)\} \) on \( \mathbb{R}^n \) is said to be integrable [26, Definition 6.4] iff there exist \( n - m \) scalar functions \( \{h_1(x), h_2(x), \ldots, h_{n-m}(x)\} \), with \( \nabla h_i \) linearly independent, satisfying \( \forall h_i : f_j = 0 \) for any \( i \in \{1, \ldots, n - m\} \) and \( j \in \{1, \ldots, m\} \).

The distribution \( \Delta = \text{span}\{f_1, f_2, \ldots, f_m\} \) is said to be involutive [26, Definition 6.5] iff \( ad_{f_j}^i f_i \in \Delta \) for any \( i, j \in \{1, \ldots, m\} \).
Theorem 3.1 (Involutivity of distribution and nilpotency of Koopman generator): Let \( \Delta = \text{span}(g, \text{ad}_g g, \ldots, \text{ad}_g^{r-1} g) \) be an \((r-1)\)-dimensional distribution for \( f, g \) in System (1). The following are equivalent:

(i) \( \Delta \) is involutive; and

(ii) there exists \( h \in C^r(\mathbb{Z}) \) and \( \alpha \in C(\mathbb{Z}) \) such that \( L_{f+g} h = 0 \) is locally nilpotent (with index \( r \)) at \( h \) and the associated cyclic subspace is a stable subspace with respect to \( \alpha \). \( \square \)

We refer the reader to Appendix A for the proof of Theorem 3.1. Some comments on Theorem 3.1 are in order. We first note that \( h \) is an observable and \( L_{f+g} h \) is the closed-loop Koopman generator with feedback \( \alpha \). Theorem 3.1 shows that when \( \Delta \) is involutive, there exists a choice of an observable \( h \) and feedback \( \alpha \), that induces the linear system (3) of dimension \( r \). From an operator-theoretic perspective, Theorem 3.1 emphasizes the role of involutivity in ensuring the existence of a finite-dimensional invariant subspace for the closed-loop Koopman generator \( L_{f+g} \). The role of the feedback \( \alpha \) is to induce a finite-dimensional invariant subspace for the controlled Koopman generator \( L_{f+g} \), where \( u = \alpha(x) \).

From Theorem 3.1, we can see the existence of a subspace invariant to the closed-loop Koopman generator \( L_{f+g} \). This Koopman invariant subspace is indeed \( \text{span}\{h(x), L_f h(x), \ldots, L_f^{r-1} h(x)\} \). As the functions \( L_f h(x) \), for all \( i = \{0, 1, \ldots, r-1\} \), are linearly independent, the dimension of the Koopman invariant subspace is \( r \) and the closed-loop Koopman generator \( L_{f+g} \) has a finite-dimensional representation. We now consider \( H(x) = [h(x), L_f h(x) \ldots L_f^{r-1} h(x)]^\top \), whose dynamics under feedback \( \alpha \) is given by

\[
\frac{dH(x)}{dt} = L_{f+g} H = AH(x)
\]

where \( A \) is as defined in (3). It is important to note that for a system that is full-state feedback linearizable, the closed-loop Koopman generator has an \( n \)-dimensional invariant subspace. For the special case when the open-loop Koopman generator \( L_f \) already has a finite-dimensional invariant subspace, then we can simply let \( \alpha = 0 \). For instance, when the vector field \( f \) is a constant matrix in \( \mathbb{R}^{n \times n} \), \( L_f \) has an invariant subspace.

Remark 1 (Multiple inputs): We now give a straightforward extension of our analysis to feedback linearizable systems with multiple inputs. When there are multiple inputs, the observable \( h(x) \) has a relative \( r_i \) for each input \( i \). If there are \( m \) inputs, the feedback linearizing inputs will now be \( \alpha(x) \in \mathbb{R}^m \) and \( \beta(x) \in \mathbb{R}^{n \times m} \). Particularly, the transformed system will look as follows:

\[
\dot{z} = A_i z + B_i v
\]

where \( A_i = \text{blkdiag}\{A_1, A_2, \ldots, A_m\} \) and \( B_i = \text{blkdiag}\{B_1, B_2, \ldots, B_m\} \). Here, \( \text{blkdiag}(\cdot) \) returns the block matrix with its arguments on the diagonal and block zero matrices everywhere else.

\[
A_i = \begin{bmatrix}
0 & 1 & 0 & \cdots & 0 \\
0 & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & \ddots & \cdots & \vdots \\
0 & 0 & 0 & \cdots & 1 \\
0 & 0 & 0 & \cdots & 0
\end{bmatrix} \in \mathbb{R}^{r_i \times r_i}, \quad B_i = \\
\begin{bmatrix}
0 \\
0 \\
\vdots \\
0 \\
1
\end{bmatrix} \in \mathbb{R}^{r_i \times 1}
\]

Here, \( r_i \) is the relative degree of the observable \( h(x) \) with respect to the \( i \)th input such that \( r_1 + r_2 + \cdots + r_m \leq n \). This also means that there exists a Koopman-invariant subspace with respect to every input, and also all the inputs considered simultaneously.

B. Data-Driven Algorithm

We use a dictionary of functions to identify the Koopman generator to lift full-state data from the nonlinear control-affine system and seek to find a linear relation in the evolution of the lifted system. Furthermore, since we already know the form of the Koopman generator, particularly the matrices \( A \) and \( B \) as defined in (3) which define the action of the closed-loop Koopman generator, we seek to best approximate this structure. To this end, we first recall the state and control transformations (2), i.e., \( z = H(x) \) and \( u = \alpha(x) + \beta(x)v \). We note that the state transformation \( H \) is determined by the observable \( h \) for which we seek an estimate \( \hat{h} \), expressed using a dictionary \( \phi \) (consisting \( M \) real functions, i.e., \( \phi = [\phi_1, \phi_2, \ldots, \phi_M]^\top \), where \( \phi_i : \mathbb{R}^n \rightarrow \mathbb{R} \) as \( \hat{h}(x) = K^\top \phi(x) \), where \( K \in \mathbb{R}^M \). We then invert the control transformation to express the external control \( v \) in terms of \( u \) as

\[
v = \zeta(x) + \eta(x)u
\]

where \( \zeta \) and \( \eta \) are given by

\[
\zeta(x) = \frac{-\alpha(x)}{\beta(x)}, \quad \eta(x) = \frac{1}{\beta(x)}.
\]

Note that \( \alpha, \beta \) can in turn be obtained from \( \zeta, \eta \) as \( \alpha(x) = -\zeta(x)/\eta(x) \) and \( \beta(x) = 1/\eta(x) \). We now seek estimates \( \hat{\zeta} \) and \( \hat{\eta} \) for the above, which are expressed using dictionaries \( \theta, \gamma \) (consisting \( K \) real functions) as

\[
\hat{\zeta}(x) = G^\top \theta(x), \quad \hat{\eta}(x) = J^\top \gamma(x)
\]

where \( G, J \in \mathbb{R}^k \) are to be estimated from data. The dictionary of repeated time derivatives of the observable dictionary \( \phi \) is represented by \( D \) as

\[
D(x) = \begin{bmatrix}
\phi(x)^\top \\
\phi(x')^\top \\
\vdots \\
\phi(r-1)(x)^\top
\end{bmatrix}.
\]

Here, \( D(x) \) utilizes the structure of \( H(x) \) as the state transformation \( H(x) \) contains the observable \( h \) and its repeated derivatives. Utilization of this structure for the dictionary is also a novelty of our algorithm. The estimated state transformation \( \hat{z}(x) \) and control transformations can now be represented as follows:

\[
z = D(x)K, \quad v = G^\top \theta(x) + J^\top \gamma(x)u.
\]

We know from feedback linearization that the state and control transformation yield the linear system (3). Here, we emphasize the fact that we do not assume that the system is feedback linearizable or the dictionaries contain all the necessary functions for the required transformations. Therefore, we obtain the vectors \( K, G, \) and \( J \) that approximate the structure of the Koopman generator in a least-squares sense, as follows:

\[
\min_{K, G, J} \sum_{t=1}^{N} \| D(x_{t+1}) - D(x_t) \|_2 \| K - A D(x_t) K - B v_t \|^2, 
\]

s.t. \( v_t = G^\top \theta(x_t) + J^\top \gamma(x_t)u_t \), \( \forall t \in \{1, \ldots, N\} \),

\[
K \neq 0, \quad G \neq 0, \quad J \neq 0.
\]

(4)

Problem (4) seeks to obtain the vectors \( K, G, \) and \( J \), simultaneously.

The control inputs \( u_t \) is assumed to be such that the least-squares problem (4) is well-posed. For instance, \( u_t \) could be sampled from a Gaussian distribution. It is evident that \( K = 0, G = 0 \) and an arbitrary nonzero \( J \) minimizes the cost in (4). Hence, we enforce the constraint on nonzero solutions for \( K, G, \) and \( J \). Note that taking repeated numerical derivatives can be a source for error in the dictionary \( D \). In such a
case, choosing $D$ such that the span\{$D$\} contains functions and their
$r - 1$ repeated time derivatives is sufficient. Another approach is to use
$r - 1$ repeated integrals over a chosen time interval as given in [27, 
Appendix A]. Further, we use three different dictionaries $D$, $\theta$, and
$\gamma$ to emphasize three transformations, but a common dictionary can be 
employed for all the transformations. It is important to note that the 
derivative $\dot{z}$ is approximated using finite differences scheme in (4).
Existing works [13, 23] assumed that $\dot{z}$ is already available in the data.
Using finite-differences to approximate the derivative can introduce 
errors in the approximation of the generator depending on the dynamics 
of the system (1). A formal analysis of these errors is beyond the scope 
of this article. Another option is to learn the Koopman operator instead 
of the infinitesimal generator. However, the generator setting provides a 
significant advantage in terms of the straightforward parameterization
of the transformations, unlike in the operator setting.

In the preceding analysis, we had assumed that the system state is
directly observable. However, when we only have access to an output
$y = h(x)$, through an observable $h$, whereby the data consists of inputs 
and outputs in the following matrices:

$$ Y = [y_0 \ y_1 \ \ldots \ y_N], \quad U = [u_1 \ u_2 \ \ldots \ u_N] $$

the problem becomes one of input–output feedback linearization. The 
problem of data-driven input–output feedback linearization is to find 
the necessary transformations $H$, $\zeta$, and $\eta$ using only input–output 
data $Y, U$. However, the subproblem of finding the state transformation 
$H(x)$ in the input–output feedback linearization problem is simpler 
as $H(x)$ is computed directly from data, as $y_i$ and its repeated time 
derivatives, since the observable $h$ is a priori fixed

$$ z = H(x) = [y \ \dot{y} \ \ddot{y} \ \ldots \ \dot{y}^{r-1}]^T. $$

Therefore, the data-driven input–output feedback linearization problem
reduces to finding estimates $\zeta$ and $\eta$ using input–output data, formulated 
as follows:

$$ \min_{G,J} \sum_{t=1}^{N} \left\| \frac{dz_t}{dt} - Az_t - Bv_t \right\|^2, $$

where

$$ \begin{cases} v_t = G^T \theta(z_t) + J^T \gamma(z_t)u_t, \quad \forall t \in \{1, \ldots, N\} \\ z_t = [y_t \ \dot{y}_t \ \ddot{y}_t \ \ldots \ \dot{y}_t^{r-1}]^T \end{cases} \quad (5) $$

We now present two methods to obtain the solutions to problems (4) 
and (5).

1) **Iterative Algorithm - KGFL**: The KGFL algorithm is an it-
erative algorithm based on gradient descent. The gradients of the cost 
in Problem (4) with respect to the parameters $K$, $G$, and $J$ can be explicitly 
computed. These gradients are utilized in an iterative manner to obtain 
the estimates $\dot{z}$, $\zeta$, and $\theta$. Algorithm 1 outlines the KGFL algorithm. 
The parameters $K$, $G$, and $J$ at iteration $i$ are denoted by $K(i)$, $G(i)$, and $J(i)$, respectively.
In the algorithm, the state transformation parameter $K$ is computed while 
keeping the control transformation parameters $G$ and $J$ fixed from the previous iterations. Subsequently, the control transformation parameters $G$ and $J$ are computed while keeping the state transformation parameters fixed. 
In the algorithm, we denote the cost function in Problem (4) as $C$ and its gradient with a parameter $Q$ as

$$ \nabla_Q C. $$

We emphasize that the dictionaries $D$, $\theta$, and $\gamma$ may not contain 
all the nonlinearities of the system’s dynamics. Hence, we solve the data-driven feedback linearization problem (4) in a least-squares sense.

For input–output feedback linearization, step 4 in KGFL need not 
be performed as the state transformation is fixed a priori.

2) **Single-Step Method**: The solutions to Problems (4) and (5)
can also be computed in a single step. We characterize the solutions to 
the two problems in the following theorem.

| Algorithm 1: Koopman Generator-Based Feedback Linearization (KGFL) |
| --- |
| 1 Data: $X$ and $U$ from System (1) |
| 2 Initialize: Dictionaries $\phi$, $\theta$, $\gamma$; Number of iterations $E$; Initial guess for $K$, $G$, and $J$; Learning rate $\epsilon$ |
| 3 For $i = 1$ to $E$: |
| 4 $K(i) = (K(i-1) - \epsilon \nabla_K C(K(i-1), G(i-1), J(i-1)))$ |
| 5 $G(i) = G(i-1) - \epsilon \nabla_G C(K(i), G(i-1), J(i-1))$ |
| 6 $J(i) = J(i-1) - \epsilon \nabla_J C(K(i), G(i), J(i-1))$ |

**Theorem 3.2 (Full-state and input–output linearizing transformations):** (a) Full-state feedback linearization: The solution to Problem (4) is given by

$$ [G^T \ J^T] = B^T \left( \frac{dD(X)}{dt} - AD(X) \right) K \left( \frac{\Theta(Y)}{\Gamma(Y)} \nabla U \right)^T. $$

(b) Input–Output linearization: The solution to problem (5) is given by

$$ [G^T \ J^T] = B^T \left( \frac{\dot{Z} - A \dot{Z}}{\nabla U} \right) \left( \frac{\Theta(Y)}{\Gamma(Y)} \nabla U \right)^T $$

where

$$ \begin{align*}
Z &= [z_1 \ z_2 \ \ldots \ z_N] \\
\dot{Z} &= [\dot{z}_1 \ \dot{z}_2 \ \ldots \ \dot{z}_N] \\
\Theta(Y) &= [\theta(z_1) \ \theta(z_2) \ \ldots \ \theta(z_N)] \\
\Gamma(Y) &= [\gamma(z_1) \ \gamma(z_2) \ \ldots \ \gamma(z_N)].
\end{align*} $$

We refer the reader to Appendix B for the proof of Theorem 3.2.

Some comments on Theorem 3.2 are now in order. For full-state 
feedback linearization, [23] presents a similar solution when the dic-
tionaries are complete. The solution to $K$, $G$, and $J$ are represented 
as vectors belonging to the null space of a matrix only dependent on 
the data $X, U$. Further, the input–output feedback linearization case 
is unaddressed in [23]. The closed-form least-squares solution provides 
a numerically simple method to compute the control transformations. It 
is important to note that the relative degree affects the number of 
repeated derivatives that are required to compute the solution in (7). If 
the relative degree of the output is not known, the control designer can 
solve with a different number of repeated derivatives and choose one 
that results in the least error.

**IV. NUMERICAL RESULTS**

In this section, we demonstrate the effectiveness of the proposed 
data-driven feedback linearization technique KGFL. We run the num-
berical experiments on an i9-9900K CPU with 128 GB of RAM. We 
sample $u_t$ from $N(0, 5)$ to collect data, and the state is sampled every 
0.01 s. We choose the control task of stabilization. The exogenous 
input to the linearized system $v$ is chosen to be $v_t = [-2 - 2] \ z_t$ for 
full state feedback linearization. This places the poles of system (3) at 
$-1 \pm 1i$. For input–output feedback linearization with relative degree 1, 
we choose $v_t = -2z_t$. Since the system (3) is controllable, we can 
arbitrarily place the poles of the system using state feedback.

The dictionaries $\phi$, $\theta$, and $\gamma$ are chosen as proposed in [28]. Particularly, these dictionaries contain Kronecker products of the Hermite 
polynomials of individual states. The probabilist’s Hermite polynomial 
$H_n(x)$ [29] of order $n$ is defined as $H_n(x) = (-1)^n e^{\frac{x^2}{2}} \frac{d^n}{dx^n} e^{-\frac{x^2}{2}}$. For example, the Hermite polynomials for $n = \{0, \ldots, 4\}$ are $1, x, x^2$ --
1, \(x^3 - 3x, x^4 - 6x^2 + 3\), respectively. Hermite polynomials serve as a useful choice for dictionaries as they form an orthogonal basis of the Hilbert space of functions [29].

A. Numerical Testbeds

We consider two testbed systems for our numerical experiments, to demonstrate the proposed data-driven feedback linearization algorithm. The first testbed system we consider is the classical Van der Pol oscillator but with the input entering the system non-linearly as

\[
\dot{x}_1 = x_2, \\
\dot{x}_2 = -x_1 + 0.5(1 - x_1^2)x_2 + (1 - x_2^2)u.
\]

We demonstrate both full-state feedback linearization and input–output feedback linearization for the Van der Pol oscillator (8). The second testbed system we consider is an arbitrary feedback linearizable system of 6-D. The system is represented in state space as follows:

\[
\begin{align*}
\dot{x}_1 &= x_2, \\
\dot{x}_2 &= x_3 + x_2, \\
\dot{x}_3 &= x_4 + \sin(x_1), \\
\dot{x}_4 &= x_5 + \cos(x_2), \\
\dot{x}_5 &= x_6 + x_1, \\
\dot{x}_6 &= -x_1^2 + \sin(x_1) + u.
\end{align*}
\]

We demonstrate full-state feedback linearization with stabilization at the origin. For System (9), we choose a dictionary similar to that chosen for the Van der Pol oscillator in Section IV-A. Further, we augment the dictionary with sine and cosine functions of all state variables.

1) Full State Feedback Linearization: The Van der Pol oscillator is full-state feedback linearizable as the system is both controllable and integrable. Particularly, choosing \(h(x) = x_1, \alpha(x) = x_1 - 0.5(1 - x_1^2)x_2\), and \(\beta(x) = (1 - x_2^2)^{-1}\) fully linearizes the system in its normal form. In Fig. 1, the dashed lines represent model-based feedback linearization whereas the solid lines represent the learned linearization transformation using the proposed algorithm. We choose a learning rate \(\epsilon = 0.01\) in KGFL. In the figure, \(x^m\) represents the model-based states. In Fig. 1(a), we choose a dictionary of Hermite polynomials of order 2 and their mutual Kronecker products. We use KGFL to perform full-state feedback linearization. It is clear from the figure that the data-driven algorithm is able to learn a transformation and stabilize the system almost as well as the model-based linearization. We also compare our proposed algorithm against the algorithms in [18], [20], and [10] in Fig. 1(b). In [18], a linear predictor of the nonlinear system is constructed without any control transformations. Hence, it cannot achieve exact linearization of the nonlinear system, especially when the control affects the system non-linearly. The data-driven algorithm proposed in [10] is motivated by an intelligent PID controller that makes use of sampled measurements in an online fashion. The algorithm proposed in [20] bilinearizes the system using offline data and performs model predictive control. To maintain fairness in comparison, we used the same initial conditions and the same offline data for the proposed algorithm and [20]. It is evident from simulations that the proposed algorithm stabilizes faster than the compared algorithms. The online algorithm in [10] uses past measurements to compute piecewise constant inputs, whereas the algorithm in [20] performs MPC by bilinearizing the system by performing state transformations. Our proposed algorithm linearizes the system using both state and control transformations and applies a state-feedback approach for pole placement which makes our algorithm effective. Furthermore, Fig. 1(c) shows that the data-driven feedback linearization algorithm performs well even for the high dimensional system (9). We use a static feedback controller \(v = -Lz\), where the matrix \(L\) places the poles of the transformed system at \([-2.08, -2.13, -2.63, -3.02, -0.92, -1.50]\). The desired poles of the system were sampled from a uniform distribution \(U(-3,0)\).

2) Output Feedback Linearization: We now demonstrate data-driven input–output feedback linearization on the Van der Pol oscillator using the single-step method described in Theorem 3.2. We choose the output \(y = h(x) = 0.5x_2^2\). We make the observation that the system has a relative degree \(r = 2\) for the selected output. For the data-driven setting, we choose a dictionary that contains the Hermite polynomials of the output and its derivatives up to the third degree. We learn the control transformation in problem (5). In Fig. 3(b), it is observed that the proposed data-driven algorithm stabilizes the output at 0. Further, the feedback linearization of the selected output induces no zero dynamics, hence the overall system is also stable, as it can be observed in Fig. 3(a).

B. Effect of Richness of Dictionary

Here, we investigate the effect of the richness of the dictionary on input–output feedback linearization for the single-step method. We choose the modified Van der Pol oscillator introduced in (8) and the output function \(h(x) = x_1\). Hence, the state transformation becomes \(z = [x_1\ x_2]^T\), and we only learn the control transformations.

Fig. 1. (a) Full state feedback linearization of the Van der Pol oscillator (8). It can be seen that the data-driven algorithm(solid lines) performs almost as good as model-based feedback linearization (dashed lines) and much better than a simple linearization with no control transformation (dotted lines) [18]. The data-driven algorithm has slightly higher overshoot and settling time than the model-based method due to imperfections in the learned transformations. (b) Comparison of the proposed algorithm [10], [18], [20]. (c) Full state feedback linearization for the complex 6 dimensional system (9). The dictionary with Hermite polynomials of order 3 is augmented with \(\sin\) and \(\cos\) of all the state variables. It is evident that the proposed algorithm stabilizes even a system of the third order.
Fig. 2. (a) Richness of the dictionary versus loss. The deviation of the data-driven algorithm from model-based feedback linearization is considered as loss which is plotted on the y-axis. The orders of Hermite polynomials used in the dictionaries are plotted on the x-axis. As the dictionaries get richer, the loss initially decreases until polynomials of degree 3. However, as we consider richer dictionaries beyond order 4 Hermite polynomials, the model overfits the data and we incur high losses. (b) Data size versus loss. The loss is defined similarly to the previous comparison. As the size of the dataset increases, the loss uniformly decreases for a dictionary with Hermite polynomials up to the second degree. (c) The comparison of our algorithm KGFL for different sampling intervals. It can be seen that the sampling rate does not have a significant effect on the performance of the algorithm for the Van der Pol oscillator (8).

Fig. 3. (a) State trajectory for input–output feedback linearization of the modified Van der Pol oscillator (8). The states are plotted on the y-axis whereas the time is plotted on the x-axis. (b) Trajectory of the output $y = 0.5x_1^2$. The output is plotted on the y-axis. It is evident that both the output and the states are stabilized as feedback linearization of the output does not create any zero dynamics.

\[ y = h(x) = x_1 \]  

The dictionaries for control transformations contain Hermite polynomials up to the second degree. From Fig. 2(b) we can see that as the data size increases, the average loss uniformly decreases until 800 data points.

D. Effect of Sampling Interval

We analyze the effect of sampling interval of the data on the performance of KGFL. For the Van der Pol oscillator considered in (8), we sample the system at rates 0.1, 0.01, and 0.001, respectively. The results of the performance of these numerical experiments are presented in Fig. 2(c). It can be seen that the sampling rate does not have a significant effect on the performance of the algorithm. A fast sampling rate of 0.001 seems to perform only marginally better than sampling rates of 0.01 and 0.1. A complete analysis of the sampling rate is beyond the scope of the article as it is dependent on the nature of the vector fields that describe the system. Typically, systems that evolve fast in the state space require faster sampling rates.

V. CONCLUSION

We establish a connection between the traditional model-based feedback linearization technique and the Koopman generator. Particularly, we show that here exists an observable and a state feedback control that renders the Koopman-generator finite-dimensional and nilpotent when the system is feedback linearizable. Using this connection, we develop an algorithm called KGFL to feedback linearize a control-affine system using experimental data. We demonstrate the algorithm numerically on complex dynamical systems and discuss tradeoffs related to the size of the dictionaries and the size of the dataset. We also show that it performs better than existing algorithms in the literature as KGFL exploits the feedback linearizable structure of the system. Directions of future research include the problem of choosing the right observables to obtain stable zero dynamics.

APPENDIX A

PROOF OF THEOREM 3.1

Proof: (i) Forward proof (Feedback linearization to Koopman generator): Since $\{ad^k_{P}g\}_{k=0}^{r-2}$ is involutive, Frobenius theorem guarantees that there exists $n - r + 1$ functions $\{h_i\}_{i=1}^{n-r+1}$ that satisfy the
following properties:
\[ L_{g}L_{\ell}^{-1}h_{i} = 0, \forall k \in \{1, \ldots, r - 1\}, \forall i \in \{1, \ldots, n - r\}, \]
\[ L_{g}L_{\ell}^{-1}h_{i} \neq 0. \]

It, then, follows that for any \( h \in \{h_{i}\}_{i=1}^{n-r+1}, k \in \{1, \ldots, r - 1\}\) and any feedback \( \alpha \), we have \( L_{\ell}g_{\alpha}L_{\ell}^{-1}h = L_{\ell}^{1}g_{\alpha} + \alpha L_{\ell}L_{\ell}^{-1}h = L_{\ell}^{1}h \). Now, let \( \alpha(x) = -(L_{\ell}L_{\ell}^{-1}h(x))^{-1}L_{\ell}^{1}h(x) \). It is important to note that \( \alpha \) is well-defined for any \( x \) since \( L_{\ell}L_{\ell}^{-1}h(x) \neq 0 \). We then, get \( L_{\ell}g_{\alpha}L_{\ell}^{-1}h = L_{\ell}^{1}h + \alpha L_{\ell}L_{\ell}^{-1}h = 0 \). Therefore, \( L_{\ell}g_{\alpha} \) is locally nilpotent at \( h \), with index \( r \).

(ii) Converse proof (Koopman generator to feedback linearization):
First, we show that the subspace span(\( h, L_{\ell}h, \ldots, L_{\ell}^{r-1}h \)) is \( L_{\ell}g_{\alpha} \)-invariant. Let \( V := \text{span}(h, L_{\ell}h, \ldots, L_{\ell}^{r-1}h) \). For any \( p \in V \), we have \( p = \sum_{k=1}^{r} c_{k}L_{\ell}^{k-1}h \), and it follows that:
\[ L_{\ell}g_{\alpha}p = L_{\ell}g_{\alpha} \sum_{k=1}^{r} c_{k}L_{\ell}^{k-1}h = \sum_{k=1}^{r} c_{k}L_{\ell}g_{\alpha}L_{\ell}^{k-1}h = \sum_{k=1}^{r-1} c_{k}L_{\ell}^{k}h \in V. \]

Therefore, we get that \( V = L_{\ell}g_{\alpha} \)-invariant. Further, from Lemma 6.5 in [26] we have that functions \( L_{\ell}^{i}h \) for \( i = \{0, 1, \ldots, r - 1\} \) are linearly independent. Therefore \( V \) is an \( L_{\ell}g_{\alpha} \)-cyclic basis.

We prove the converse of the theorem by induction. Let \( \ell \) be an observable such that \( L_{\ell}g_{\alpha} \) is nilpotent at \( \ell \) with index \( r \). Further, let \( \ell_{1} = L_{\ell}^{(-1)} \ell \) for \( k \in \{1, \ldots, r\} \). Consider the functions \( \ell_{1} \) and \( \ell_{2} = L_{\ell}^{(-1)} \ell_{1} \) have that \( L_{\ell}g_{\alpha}L_{\ell}^{(-1)} \ell_{2} = \ell_{2} \). However, we have that \( \frac{\partial g_{\alpha}}{\partial \ell} \) is due to the stability of \( L_{\ell}g_{\alpha} \)-cyclic subspace, where \( \frac{\partial g_{\alpha}}{\partial \ell} \) denotes the Gateaux derivative with respect to \( \alpha \). This is a consequence of the definition of subspace stability. This implies that \( L_{\ell}g_{\alpha}L_{\ell}^{(-1)} \ell = 0 \). Similarly, considering \( \frac{\partial g_{\alpha}}{\partial \ell} \neq 0 \), we get that \( L_{\ell}g_{\alpha}L_{\ell}^{(-1)} \ell = 0 \). By doing this every basis function \( \ell_{i} \), we obtain that \( \text{Ker}(L_{\ell}) = \text{span}(\ell_{1}, \ell_{2}, \ldots, \ell_{r-1}) \).

APPENDIX B

PROOF OF THEOREM 3.2

Proof: If the dictionaries \( D, \theta \) and \( \gamma \) are complete, we can express the feedback linearized system with data as follows:
\[ \dot{X}(X)K = A(D)X + B(\Theta(Y) + R^{T}(\Gamma(U) \otimes U)). \]

The least-squares solution in terms of \( K, Q, \) and \( R \) can be obtained by simply solving for the first-order condition of optimality. For input-output feedback linearization, \( \dot{X}(X)K \) and \( D(X) \) are replaced by \( \dot{Z} \) and \( Z \), respectively.

REFERENCES

[1] A. Isidori, Nonlinear Control Systems: An Introduction. New York, NY, USA: Springer, 1985.
[2] K. J. Hunt, D. Sbarbaro, R. Zbikowski, and P. J. Gawthrop, “Neural networks for control systems-a survey,” Automatica, vol. 28, no. 6, pp. 1083–1112, 1992.
[3] A.Yeşildirek and F. L. Lewis, “Feedback linearization using neural networks,” Automatica, vol. 31, no. 11, pp. 1659–1664, 1995.