Data-Driven Feedback Linearization with Complete Dictionaries

C. De Persis, D. Gadginmath, F. Pasqualetti and P. Tesi

***Abstract*— We consider the feedback linearization problem, and contribute with a new method that can learn the linearizing controller from a library (a dictionary) of candidate functions. When the dynamics of the system are known, the method boils down to solving a set of linear equations. Remarkably, the same idea extends to the case in which the dynamics of the system are unknown and a linearizing controller must be found using experimental data. In particular, we derive a simple condition (checkable from data) to assess when the linearization property holds over the entire state space of interest and not just on the dataset used to determine the solution. We also discuss important research directions on this topic.**

I. Introduction

Data-driven control is becoming more and more central to control engineering. The motivation is to automate the control design process when the system of interest is poorly modeled (first-principles laws might be difficult to obtain) or when accurate models are too complex to be used for controller design. Data-driven control has been applied in almost every area of control theory, among which we can mention optimal [1], [2], [3], [4], robust [3], [5], [6], [7], and predictive control [8], [9], [10], [11]. Further applications include networked and distributed control [12], [13], [14], [15]. Also the techniques are quite varied, ranging from dynamical programming and behavioral theory to techniques typical of machine learning. The majority of the results consider linear systems. Unsurprisingly, deriving solutions for nonlinear systems is harder. The objective of this paper is to explore data-driven control for nonlinear systems, in particular data-driven *feedback linearization*.

Related work. Data-driven control for nonlinear systems is still largely unexplored. A way to deal with nonlinear systems is to exploit some structure, when it is a priori known the class to which the system belongs, for instance bilinear [16], [17], and polynomial (or rational) systems [7], [18], [19], [20], [21]. Methods for general nonlinear systems include linearly parametrized models with basis functions [22], [23], [24], Gaussian process models [25], [26], linear [3], [27], [28], [29] and polynomial approximations [30], [31], and the feedback linearization [24], [26], [32], [33], which is the focus of this work.

This material is based upon work supported in part by the project Digital Twin with project number P18-03 of the research programme TTW Perspective which is (partly) financed by the Dutch Research Council (NWO) and by awards ARO W911NF-20-2-0267, AFOSR-FA9550-20-1-0140, AFOSR-FA9550-19-1-0235, and NSF-2020246. C. De Persis is with ENTEG, University of Groningen, 9747AG Groningen, The Netherlands. Email: c.de.persis@rug.nl. D. Gadginmath and F. Pasqualetti are with the Department of Mechanical Engineering, University of California at Riverside, Riverside, CA, 92521, USA. Email: dgadg001,fabiopas@engr.ucr.edu. P. Tesi is with the Department of Information Engineering, University of Florence, 50139 Florence, Italy. Email: pietro.tesi@unifi.it.

As well known, the feedback linearization method aims at finding a coordinate transformation where the dynamics can be linearized via feedback (i.e., where all the nonlinearities can be canceled out with a feedback controller). Linearization is clearly appealing as it allows us to exploit all the concepts and tools that are available for linear systems, thus simplifying problems that would otherwise be difficult to address. In the context of data-driven control, recent examples are sampled-data stabilization [27] and output-matching control [34]. We remark that also the problem of lifting a nonlinear system to a linear system of a *higher* dimension has been studied, famous methods are *immersion* [35] and the *Koopman operator theory* [36], [37]. In this respect, feedback linearization can be viewed as the natural starting point for techniques of this kind because it searches for a mapping that *preserves* the state dimension. We refer the interested reader to [33] for a recent discussion on the connections between feedback linearization and the Koopman operator theory.

Contribution. In this paper, we address the problem of solving the feedback linearization problem from data. We approach this problem in the simplest possible setting that we can think of, namely full-state linearization and noise-free data. While this is a favorable setting, the problem remains far from straightforward. In fact, most of the existing works on this topic assume that the state transformation is given in the sense that they assume that we know the state coordinates that render the system linearizable (usually, that we know the function for which the system has relative degree equal to the state dimension) [24], [26], [27], [32], [34], and the problem essentially becomes that of finding a feedback that linearizes (and stabilizes) the closed loop.

In our method, we learn the state and control transformations from a *library* of basis functions. Approaches of this type have been widely used in the context of system identification, see for example [38], and recently also in the context of direct data-driven control [22], [24], [34]. We start by considering a model-based solution, that is, assuming that we know the dynamics of the systems (Section III-A). The model-based solution inspires the solution in case the dynamics are unknown and the feedback linearization must be solved using data alone. (Section III-B). Our approach shows some interesting features. From a theoretical point of view, we provide conditions under which the solution, which is determined from a finite number of datapoints, defines a coordinate transformation that is valid on the *whole* space of interest. Finally, from a practical point of view, the solution is obtained by determining the null space of a matrix, which is computationally fast even for big matrices. Further, numerical derivatives are not necessary for computing the data matrices [39] but it simplifies exposition.

Throughout the paper we also discuss practical issues, the choice of the library of functions, experiment design, and numerical aspects; see in particular Section III-B.

Notation and definitions. Given a function $h : \mathcal{D} \to \mathcal{E}$, its inverse function $h^{-1} : \mathcal{E} \to \mathcal{D}$, provided it exists, is the function such that $h^{-1}(h(x)) = x$ for all $x \in \mathcal{D}$. Given a function $h : \mathcal{D} \to \mathcal{E}$ taking on scalar (matrix) values, we denote by $h(x)^{-1}$ the reciprocal (inverse) of $h(x)$ at x. By *coordinate transformation* $\tau : \mathcal{D} \to \mathcal{E}$ it is meant a local diffeomorphism [40, p. 11], that is τ is a bijection and both τ and τ^{-1} are smooth mappings. We will also consider the notions of (vector) *relative degree* and *Lie derivative*. The reader is referred to [40, p. 220, 496] for their definitions and properties.

II. Problem setting

A. Feedback linearization problem

Consider a continuous-time nonlinear system

$$\dot{x} = f(x) + g(x)u \tag{1}$$

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ is the input, $f : \mathbb{R}^n \to \mathbb{R}^n$, and $g : \mathbb{R}^n \to \mathbb{R}^m$ are smooth vector fields. The objective is to stabilize the system at some $x = x^0$ with the so-called *feedback linearization* [40, Chapter 5], as we make precise in the sequel, in the case in which the vector fields $f(x), g(x)$ are not known but some priors about them are available.

We begin by recalling the *state space exact linearization problem* or *linearization problem*, for short [40, p. 228]. Given a state x^0, find a neighborhood $\mathcal{D}$ of x^0, functions $\alpha\colon \mathcal{D} \to \mathbb{R}^m$, $\beta\colon \mathcal{D} \to \mathbb{R}^{m\times m}$, a coordinate transformation $\tau\colon \mathcal{D} \to \mathbb{R}^n$ and a controllable pair (A, B) such that, for each $x \in \mathcal{D}$,

$$\frac{\partial \tau}{\partial x}(f(x) + g(x)\alpha(x)) = A\tau(x), \tag{2a}$$

$$\frac{\partial \tau}{\partial x} g(x)\beta(x) = B. \tag{2b}$$

We slightly reformulate the problem in a form that is more convenient for our purposes.

Lemma 1: The linearization problem is solvable if there exist a neighborhood $\mathcal{D}$ of x^0, functions $\gamma\colon \mathcal{D} \to \mathbb{R}^{m\times m}$, $\delta\colon \mathcal{D} \to \mathbb{R}^m$, with $\gamma(x)$ nonsingular for all $x \in \mathcal{D}$, and a coordinate transformation $\tau\colon \mathcal{D} \to \mathbb{R}^n$ such that, for each $x \in \mathcal{D}$ and $u \in \mathbb{R}^m$,

$$\frac{\partial \tau}{\partial x}(f(x) + g(x)u) = A_c\tau(x) + B_c(\delta(x) + \gamma(x)u) \tag{3}$$

where $A_c = \operatorname{diag}(A_1, \ldots, A_m)$, $B_c = \operatorname{diag}(B_1, \ldots, B_m)$,

$$A_i = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \\ 0 & 0 & 0 & \ldots & 0 \end{bmatrix} \in \mathbb{R}^{r_i\times r_i}, \quad B_i = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \in \mathbb{R}^{r_i\times 1}, \tag{4}$$

with $r_1, \ldots, r_m$ nonnegative integers such that $r_1 + \ldots + r_m = n$.

Further, if $g(x^0)$ has rank m then (3) is also necessary. ■ We refer the reader to [40, section 5.2] for the proof. By Lemma 1, condition (3) implies condition (2) and is equivalent to (2) whenever $g(x^0)$ has full column rank. The latter requirement is in fact quite mild, in particular for single-input systems it amounts to requiring that $g(x^0)$ is a nonzero vector. It is convenient to focus on (3) as it expresses the linearization condition through one single equation that involves the flow dynamics $f(x) + g(x)u$. This turns out to be useful when we want to solve the linearization problem with data, as we can use data (trajectories) as a proxy for $f(x) + g(x)u$. With this in mind, we will focus on (3).

B. Objective of the paper and assumptions

Our objective is to discuss how to solve the linearization problem, i.e. how we can find functions τ, δ and γ satisfying (3). We will first discuss in Section III-A the case where the dynamics of the system are known. Then, in Section III-B, we will discuss how we can approach the problem when the dynamics of the system are unknown. Clearly, if the dynamics of the system are known, we can deal with the linearization problem using classical results [40, Theorem 5.2.3]. The motivation to consider the case of known dynamics in Section III-A is that it introduces the line of thought that will be also used in Section III-B when the dynamics are unknown. The main assumptions in this work are:

Assumption 1: The linearization problem is solvable. ■

Assumption 2: We know functions $Z : \mathcal{D} \to \mathbb{R}^s$, $Y : \mathcal{D} \to \mathbb{R}^p$, and $W : \mathcal{D} \to \mathbb{R}^{r\times m}$, such that

$$\tau(x) = \overline{T}Z(x),\ \delta(x) = \overline{N}Y(x),\ \gamma(x) = \overline{M}W(x) \tag{5}$$

for some (unknown) matrices $\overline{T}$, $\overline{N}$, $\overline{M}$. ■

Assumption 2 can be expressed by saying that we know a library of functions that includes the real system in the τ-coordinates. We thus require that some prior knowledge about the system is available, as is the case with mechanical and electrical systems where some information about the dynamics can be derived from first principles. Assumptions of this type have been widely adopted in the context of system identification, see for example [38], and recently also in the context of direct data-driven control, see [22], [24]. We allow Z, W, Y to contain terms not present in τ, γ, α, which accounts for an imprecise knowledge of the dynamics of interest (in fact, we may well have $s \gg n$ and $p, r \gg m$). Thus, the goal is to discover the elements of the functions Z, W, Y to obtain the identity (3).

Before proceeding we make one final remark. When (3) holds, a linearizing control law is $u = \gamma(x)^{-1}(v - \delta(x))$, with v a signal. This returns $\dot{\eta} = A_c\eta + B_cv$, where $\eta = \tau(x)$. Since (A_c, B_c) is controllable, we can thus select $v = K\tau(x)$ with K a state-feedback matrix that renders $A_c + B_cK$ Hurwitz. As τ has a continuous inverse, under the extra property $\tau(x^0) = 0$ we will then have that $u = \gamma(x)^{-1}(K\tau(x) - \delta(x))$ stabilizes the nonlinear system at x^0. While the design of u is the obvious ultimate goal, it is important to stress that (3) does *not* involve the design of K; it rather defines an intermediate step after which K, and thus u, can be readily designed. This point is crucial for the approach considered in the paper in the sense that by looking at (3) –without K– we can obtain a numerically efficient method, as detailed next.

III. Main results

We begin by expressing condition (3) in a convenient form. Under Assumption 2, condition (3) can be written as

$$\overline{T}\frac{\partial Z}{\partial x}(f(x)+g(x)u) = A_c\overline{T}Z(x)+B_c(\overline{N}Y(x)+\overline{M}W(x)u). \tag{6}$$

Defining

$$\ell_1(x,\dot{x}) := Z(x)^\top \otimes A_c - \left(\frac{\partial Z}{\partial x}\dot{x}\right)^\top \otimes I_n \tag{7a}$$

$$\ell_2(x) := Y(x)^\top \otimes B_c \tag{7b}$$

$$\ell_3(x,u) := (W(x)u)^\top \otimes B_c \tag{7c}$$

and recalling the properties of the vectorization operator, (6) can be thus given the equivalent form

$$F(x,u,\dot{x})\overline{v} = 0 \tag{8}$$

where,

$$F(x,u,\dot{x}) = \begin{bmatrix}\ell_1(x,\dot{x}) & \ell_2(x) & \ell_3(x,u)\end{bmatrix} \tag{9a}$$

$$\overline{v} = \begin{bmatrix}\mathrm{vec}(\overline{T})\\ \mathrm{vec}(\overline{N})\\ \mathrm{vec}(\overline{M})\end{bmatrix} \tag{9b}$$

By Assumptions 1 and 2, there exists $\overline{v} \neq 0$ such that (8) holds for all $x \in \mathcal{D}$ and all $u \in \mathbb{R}^m$. The property $\overline{v} \neq 0$ holds because otherwise $\tau(x)$ would not be a change of coordinates and $\gamma(x)$ would not be nonsingular. Having assumed that the linearization problem is feasible and bearing in mind that its solution takes the form (8), we thus focus on the problem of finding one solution of such a form, i.e., $v \neq 0$ such that

$$\begin{aligned}&F(x,u,\dot{x})v = 0,\\ &\text{for all } x \in \mathcal{D},\ u \in \mathbb{R}^m, \text{ and } \dot{x} \text{ such that } \dot{x} = f(x)+g(x)u.\end{aligned} \tag{10}$$

A. Model-based solution

We first derive a solution to the linearization problem assuming that the dynamics of the system is known. This we will serve as a basis for the case where the dynamics is unknown and we only have access to input-state data collected from the system.

We approach this problem in the following way. Since f, g are known, the matrix of functions $F(x,u,f(x)+g(x)u))$ is also known. Let F_{ij} denote the (i,j)-th entry of the matrix F. Knowing the functions $F_{ij}(x,u,f(x)+g(x)u))$ we can determine a set of linearly independent functions over $\mathbb{R}$ defined on $\mathcal{D}\times\mathbb{R}^m$, $\{\phi_k(x,u)\}_{k=1}^{n_b}$, such that each $F_{ij}(x,u,f(x)+g(x)u))$ can be expressed as a linear combination of this basis, that is

$$F_{ij}(x,u,f(x)+g(x)u)) = \phi(x,u)^\top c_{ij}, \tag{11}$$

$$\phi(x,u) := \begin{bmatrix}\phi_1(x,u) & \dots & \phi_{n_b}(x,u)\end{bmatrix}^\top \tag{12}$$

and $c_{ij} \in \mathbb{R}^{n_b}$ is a vector of coefficients. Notice that the subscript of c_{ij} is used to match the indexing of F_{ij}, and not to indicate some entry of the vector c. Here, each c_{ij} is unique because $\phi(x,u)$ consists of independent functions.

Theorem 1: ***(Model-based solution with complete dictionaries)*** Let $F(x,u,\dot{x})$ be as in (9a). Let $\{\phi_k(x,u)\}_{k=1}^{n_b}$ be a set of linearly independent functions over $\mathbb{R}$ defined on the set $\mathcal{D}\times\mathbb{R}^m$, such that each entry F_{ij} of F is expressed as in (11)-(12) on $\mathcal{D}\times\mathbb{R}^m$. Then, v satisfies (10) if and only if v is a solution of the system of linear equations

$$\begin{bmatrix}\sum_{j=1}^{\mu} c_{1j}e_j^\top\\ \vdots\\ \sum_{j=1}^{\mu} c_{nj}e_j^\top\end{bmatrix} v = 0 \tag{13}$$

where e_j denotes the j-th vector of the canonical basis of $\mathbb{R}^\mu$, and $\mu = ns+pm+rm$ denotes the size of v. □

Proof. Write the system of equations $F(x,u,f(x)+g(x)u))v = 0$ in (10) row by row as

$$\sum_{j=1}^{\mu} F_{ij}(x,u,f(x)+g(x)u))v_j = 0, \quad i = 1,2,\ldots,n \tag{14}$$

By (11), the equations above can be re-written as

$$\begin{aligned}&\sum_{j=1}^{\mu} F_{ij}(x,u,f(x)+g(x)u))v_j\\ &= \sum_{j=1}^{\mu} \phi(x,u)^\top c_{ij}v_j\\ &= \phi(x,u)^\top \sum_{j=1}^{\mu} c_{ij}e_j^\top v = 0, \quad i = 1,2,\ldots,n\end{aligned} \tag{15}$$

As the vector $\phi(x,u)^\top$ is made of linearly independent functions over $\mathbb{R}$, the equation

$$\phi(x,u)^\top \sum_{j=1}^{\mu} c_{ij}e_j^\top v = 0 \tag{16}$$

holds if and only if $\sum_{j=1}^{\mu} c_{ij}e_j^\top v = 0$. This equation must hold for any integer i. We conclude that the set of all the vectors v such that (10) holds is given by the solutions to the system of linear equations (13). ■

A solution of interest is obtained by excluding the solution $v = 0$ from the set of solutions to the system of linear equations. One can further seek among all these solutions at least one, say $\hat{v}$, for which $\hat{T}\frac{\partial Z}{\partial x}(x^0)$ and $\hat{M}W(x^0)$ are nonsingular.

Example 1: ***(Model-based feedback linearization)*** Consider the nonlinear system [36]

$$\begin{cases}\dot{x}_1 = \mu x_1 + u\\ \dot{x}_2 = \lambda(x_2 - x_1^2) + u\end{cases} \tag{17}$$

where $\mu \neq \lambda$ and where the equilibrium of interest is $x^0 = 0$. The system satisfies the necessary and sufficient conditions for the linearization problem to be solvable [40, Theorem 4.2.3]. To determine the coordinate transformation, one could solve the partial differential equation $\frac{\partial \lambda}{\partial x} g(x) = 0$ with the nontriviality condition $\frac{\partial \lambda}{\partial x}[f,g](0) \neq 0$, where $[\cdot,\cdot]$ denotes the Lie bracket. Once λ is obtained, the coordinate transformation τ would be obtained setting $\tau(x) = \begin{bmatrix} \lambda(x) & \frac{\partial \lambda}{\partial x} f(x) \end{bmatrix}^\top$. Here, we pursue the approach outlined in Theorem 1. Suppose that we choose the following library of basis functions (this choice is dictated by the willingness to keep the size of F and v small enough):

$$Z(x) = \begin{bmatrix} x \\ x_1^2 \\ x_2^2 \end{bmatrix}, \quad Y(x) = Z(x), \quad W(x) = \begin{bmatrix} 1 \\ Z(x) \end{bmatrix}. \tag{18}$$

We add the constant factor 1 in $W(x)$ to make sure that $\gamma(x) = \overline{M} W(x)$ is nonsingular about 0. With this choice, $F(x,u,\dot{x})$ is made of the following sub-matrices:

$$\ell_1(x,\dot{x}) = \begin{bmatrix} -\dot{x}_1 & x_1 & -\dot{x}_2 & x_2 & -2x_1\dot{x}_1 & x_1^2 & -2x_2\dot{x}_2 & x_2^2 \\ 0 & -\dot{x}_1 & 0 & -\dot{x}_2 & 0 & -2x_1\dot{x}_1 & 0 & -2x_2\dot{x}_2 \end{bmatrix}, \tag{19a}$$

$$\ell_2(x) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ x_1 & x_2 & x_1^2 & x_2^2 \end{bmatrix}, \tag{19b}$$

$$\ell_3(x,u) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ u & x_1 u & x_2 u & x_1^2 u & x_2^2 u \end{bmatrix}. \tag{19c}$$

Note that $\mu = \text{size}(v) = 17$. Bearing in mind the expression of $\dot{x}_1, \dot{x}_2$, the basis functions $\{\phi_k(x,u)\}_{k=1}^{n_b}$ in terms of which the entries of $F(x,u,\dot{x})$ can be expressed are

$$x_1,\ x_2,\ u,\ x_1^2,\ x_2^2,\ x_1 u,\ x_2 u,\ x_1^2 x_2,\ x_1^2 u,\ x_2^2 u. \tag{20}$$

Hence, $n_b = 10$. This choice gives a unique set of coefficients c_{ij}, with $i = 1,2$ and $j = 1,2,\dots,\mu$ which, once replaced in (13), gives rise to the following system of equations:

$$\begin{aligned} -\mu v_1 + v_2 &= 0, & -\lambda v_3 + v_4 &= 0, \\ \lambda v_3 - 2\mu v_5 + v_6 &= 0, & -2\lambda v_7 + v_8 &= 0, \\ -\mu v_2 + v_9 &= 0, & -\lambda v_4 + v_{10} &= 0, \\ -v_2 - v_4 + v_{13} &= 0, & \lambda v_4 - 2\mu v_6 + v_{11} &= 0, \\ -2v_6 + v_{14} &= 0, & 2v_8 = v_{15} &= 0, \\ v_1 + v_3 &= 0, & v_5 = v_7 &= 0, \\ -2\lambda v_8 + v_{12} &= 0, & v_{16} = v_{17} &= 0, \end{aligned} \tag{21}$$

whose solutions are given by

$$\begin{aligned} v^\top = [&v_1 \ \ \mu v_1 \ \ -v_1 \ \ -\lambda v_1 \ \ 0 \ \ \lambda v_1 \ \ 0 \ \ 0 \ \ \mu^2 v_1 \ \ -\lambda^2 v_1 \\ &\lambda(2\mu+\lambda) v_1 \ \ 0 \ \ (\mu-\lambda) v_1 \ \ 2\lambda v_1 \ \ 0 \ \ 0 \ \ 0] \end{aligned} \tag{22}$$

with v_1 a free parameter. The first 8 entries of v define $\overline{T}$:

$$\overline{T} = v_1 \begin{bmatrix} 1 & -1 & 0 & 0 \\ \mu & -\lambda & \lambda & 0 \end{bmatrix} \tag{23}$$

which returns the change of coordinates

$$\tau : x \mapsto v_1 \begin{bmatrix} x_1 - x_2 \\ \mu x_1 - \lambda(x_2 - x_1^2) \end{bmatrix} \tag{24}$$

The 9th–12th entries of v define $\overline{N}$ and return $\delta(x) = \overline{N} Y(x) = v_1(\mu^2 x_1 - \lambda^2 x_2 + \lambda(2\mu+\lambda) x_1^2)$. The 13th–17th entries define $\overline{M}$ and return $\gamma(x) = \overline{M} W(x) = (\mu - \lambda) v_1 + 2\lambda v_1 x_1$. For any $v_1 \neq 0$, we obtain feasible $\tau(x), \gamma(x), \delta(x)$. The solution coincides with the one guaranteed by [40, Theorem 4.2.3]. ■

B. Data-based solution and generalization from the sample space

The approach described in the previous section requires knowledge of the vector fields f, g. We can draw inspiration from the model-based approach to find a solution achievable using data alone. We start again from the equation $F(x,u,\dot{x})v = 0$, but instead of expressing $F(x,u,\dot{x})$ via the basis functions $\{\phi_k(x,u)\}_{k=1}^{n_b}$, which is not possible since the vector fields defining $\dot{x}$ are unknown, we evaluate $F(x,u,\dot{x})$ on a dataset and use this information to build a solution v to (10) when f, g are unknown.

Namely, assume that is is possible to make an experiment on the system and collect a dataset $\mathbb{D} := \{(x_i, u_i, \dot{x}_i)\}_{i=0}^{L-1}$ with $x_i \in \mathcal{D}$, $u \in \mathbb{R}^m$, and $\dot{x}_i = f(x_i) + g(x_i) u_i$. The idea is then to solve (10) at the collected data points, namely to find a vector $v \neq 0$ belonging to the kernel of $\mathcal{F}(\mathbb{D})$ where

$$\mathcal{F}(\mathbb{D}) := \begin{bmatrix} F(x_0, u_0, \dot{x}_0) \\ \vdots \\ F(x_{L-1}, u_{L-1}, \dot{x}_{L-1}) \end{bmatrix}. \tag{25}$$

Note that each entry of $\mathcal{F}(\mathbb{D})$ is of the form $F(x,u,\dot{x})$ with $\dot{x}_i = f(x_i) + g(x_i) u_i$ for each $i = 0, \dots, L-1$.

The matrix $\mathcal{F}(\mathbb{D})$ can be computed from data, after which we easily determine its null space. This procedure, however, only guarantees that the dynamics are linearized at the collected data points, i.e., in the *sample* space. Clearly, we would like that to ascertain whether a solution defines a coordinate transformation that is valid on a whole space of interest, and this involves the problem of generalizing from a *finite* set of data points to *infinitely* many data points. It is actually possible to give a simple condition that resolves this issue, and this condition can be checked using data alone. This is formalized in the next result.

Theorem 2: ***(Data-driven solution with complete dictionaries)*** Let nullity$(\mathcal{F}(\mathbb{D})) = 1$ where nullity(M) denotes the dimension of the null space of a matrix M. Then, any vector $v \neq 0$ belonging to the null space of $\mathcal{F}(\mathbb{D})$ satisfies (10), hence it solves the linearization problem. □

Proof. By Assumption 2, the identity (8) holds for all $x \in \mathcal{D}$ and all $u \in \mathbb{R}^m$. In particular, it holds at the collected data points, which implies that $\mathcal{F}(\mathbb{D})\overline{v} = 0$. Since nullity$(\mathcal{F}(\mathbb{D})) = 1$, then any nonzero vector $v \in \ker(\mathcal{F}(\mathbb{D}))$ is such that $v = \lambda \overline{v}$, with $\lambda \in \mathbb{R} \setminus \{0\}$. Multiplying both sides of (8) by λ, we obtain $\lambda F(x,u,\dot{x})\overline{v} = 0$, that is, $F(x,u,\dot{x})v = 0$. This concludes the proof. ■

Example 2: ***(Data-driven feedback linearization)*** Consider again the nonlinear system

$$\begin{cases} \dot{x}_1 = \mu x_1 + u \\ \dot{x}_2 = \lambda(x_2 - x_1^2) + u \end{cases} \tag{26}$$

where the equilibrium of interest is $x^0 = 0$. We assume $\mu = -0.5$ and $\lambda = 0.2$. According to Example 1,

$$\tau : x \mapsto \begin{bmatrix} x_1 - x_2 \\ -0.5x_1 - 0.2(x_2 - x_1^2) \end{bmatrix} \tag{27}$$

defines a change of variables linearizing the system about the origin. Specifically, in the coordinates $\eta = \tau(x)$ we have

$$\begin{cases} \dot\eta_1 = \eta_2 \\ \dot\eta_2 = \underbrace{0.25x_1 - 0.04x_2 - 0.16x_1^2}_{\delta(x)} + \underbrace{(0.4x_1 - 0.7)}_{\gamma(x)} u \end{cases} \tag{28}$$

Suppose that the model is unknown and we want to discover this coordinate transformation using data collected from the system. We make an experiment on the system of duration 10s in which we apply a piecewise constant input uniformly distributed within $[-0.1, 0.1]$ and with initial conditions in the same interval. We collect $L = 100$ samples $\{x_i, u_i, \dot x_i\}$ with period $0.1s$. Suppose that we choose the following library of basis functions:

$$Z(x) = \begin{bmatrix} x \\ x^2 \\ x^3 \\ \sin(x) \\ \cos(x) \end{bmatrix}, \quad Y(x) = Z(x), \quad W(x) = \begin{bmatrix} 1 \\ Z(x) \end{bmatrix}. \tag{29}$$

where by x^2 we mean the vector with components x_1^2 and x_2^2, and the same meaning holds for the sine, the cosine and the cubic function. Here the library is richer than the one of Example 1 to illustrate the situation where we add several candidate functions to compensate the lack of knowledge of the dynamics of the system. Using this library and the dataset, we compute $\mathcal{F}(\mathbb{D})$ and determine its null space. The null space of $\mathcal{F}(\mathbb{D})$ is 1, and the solution is indeed as in next (30) (modulo a constant factor):

$$\overline{T} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.5 & -0.2 & 0.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{30a}$$

$$\overline{N} = \begin{bmatrix} 0.25 & -0.04 & -0.16 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{30b}$$

$$\overline{M} = \begin{bmatrix} -0.7 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{30c}$$

One sees that the algorithm automatically discards sine, cosine and cubic functions, recovering the model-based solution. In this example, we systematically obtained the same solution with different datasets generated considering different input signals. ■

We conclude this section with some remarks discussing theoretical and practical aspects of the data-based solution.

Remark 1: ***(Domain of linearization and region of attraction (RoA))*** Assumption 2 requires that we know the domain of linearization $\mathcal{D}$. In practice, this can be inferred from data by determining the domain over which $TZ(x)$ is a diffeomorphism. It is interesting to note that the knowledge of $\mathcal{D}$ permits us to immediately obtain an estimate of the RoA. In fact, once we have the linear dynamics $\dot\eta = A_c\eta + B_c v$ in the coordinates $\eta = \tau(x)$ we can determine a stabilizing control law $v = K\tau(x)$ and a RoA, say $\mathcal{S}$, for the system $\dot\eta = (A_c + B_c K)\eta$ that is also an invariant set (for instance, a Lyapunov sublevel set). Thus, a RoA for the nonlinear system can be obtained as any invariant set $\mathcal{R} \subseteq \mathcal{S}$ such that $\tau^{-1}(\eta) \in \mathcal{D}$ for all $\eta \in \mathcal{R}$. ■

Remark 2: ***(Choice of the library and noisy data)*** Our approach crucially depends on Assumption 2. As shown in Example 2, we can be generous with the number of candidate functions as the solution is eventually obtained by determining the null space of a matrix ($\mathcal{F}(\mathbb{D})$), which is computationally fast even for big matrices. The situation can be different with noisy data. The main issue with noisy data is that all the functions might be selected (the overfitting problem). In this case, some prior knowledge can help to keep the size of the library moderate. In parallel, it can be useful to explicitly search for *sparse* solutions, for example: minimize$_{v:\|v\|=1}\|\mathcal{F}(\mathbb{D})v\|_2^2 + \alpha\|v\|_0$, where $\|\mathcal{F}(\mathbb{D})v\|_2^2$ replaces the constraint $\mathcal{F}(\mathbb{D})v = 0$, while $\alpha\|v\|_0$, $\alpha > 0$, penalizes the complexity of the solution; finally, the constraint $\|v\| = 1$ ensures that the solution is different from zero. Formulations of this type have been widely adopted in the context of data-driven control *cf.* [24], [34], [38], [41]. Treatment of this problem is left for future research. ■

Remark 3: ***(Experiment design and the data acquisition scheme)*** Besides noisy data, there are other practical aspects that are worth mentioning. The matrix $\mathcal{F}(\mathbb{D})$ has dimension $Ln\times(ns+pm+rm)$. Thus, a necessary condition to have the condition nullity$(\mathcal{F}(\mathbb{D})) = 1$ fulfilled is that rank$(\mathcal{F}(\mathbb{D})) = ns + pm + rm - 1$ and therefore we need that $Ln \geq (ns + pm + rm) - 1$. This means that we need to collect a sufficient amount of data, and this amount increases with the number of basis functions. This is not surprising as the solution is obtained through an interpolation problem. Another aspect related to experiment design is that to have rank$(\mathcal{F}(\mathbb{D})) = ns + pm + rm - 1$ we need data of good quality. The design of (persistently) exciting input signals for nonlinear systems is still not well understood, we refer the interested reader to [42] for a recent discussion. The choice of the sampling time is instead less critical except when we sample too fast. In this case, the trajectories become almost constant, thus we can lose information regarding the dynamics of the system and the problem can become numerically ill-conditioned. ■

IV. Concluding remarks

In this paper, we addressed the problem of feedback linearization from data. Unlike existing results where the state transformation is assumed to be known, we propose a method to learn both the change of coordinates and the state feedback control. Our method has some interesting features. Most notably, it allows us to determine a solution from a finite set of datapoints. From a practical point of view, the solution is obtained by determining the null space of a matrix, which is computationally fast even for big matrices. There are many important venues for future research, we list three. An important research line is how to extend the result presented here to the case of *input-output* feedback linearization, which occurs when only a portion of the state is linearizable. We are confident that the ideas discussed here can be successfully extended to this case as well. A second research line pertains how to deal with neglected dynamics (incomplete basis functions) and noisy data, all cases where the solution might return an approximate linearization only. Here the challenge is to understand how errors made on the estimate of $\tau(x)$ propagate to the control law, and how we can mitigate this through the choice of the basis functions. Finally, it is interesting to consider these ideas in the more general context of *immersion*.

REFERENCES

[1] G. Baggio, V. Katewa, and F. Pasqualetti, "Data-driven minimum-energy controls for linear systems," *IEEE Control Systems Letters*, vol. 3, no. 3, pp. 589–594, 2019.

[2] S. Dean, H. Mania, N. Matni, B. Recht, and S. Tu, "On the sample complexity of the linear quadratic regulator," *Foundations of Computational Mathematics*, vol. 20, pp. 633–679, 2020.

[3] C. De Persis and P. Tesi, "Formulas for data-driven control: Stabilization, optimality, and robustness," *IEEE Transactions on Automatic Control*, vol. 65, no. 3, pp. 909–924, 2020.

[4] F. A. Pellegrino, F. Blanchini, G. Fenu, and E. Salvato, "Closed-loop control from data-driven open-loop optimal control trajectories," in *2022 European Control Conference*, 2022.

[5] J. Berberich, A. Koch, C. W. Scherer, and F. Allgöwer, "Robust data-driven state-feedback design," in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 1532–1538.

[6] H. van Waarde, K. Camlibel, and M. Mesbahi, "From noisy data to feedback controllers: Nonconservative design via a matrix S-lemma," *IEEE Transactions on Automatic Control*, vol. 67, no. 1, pp. 162–175, 2022.

[7] A. Bisoffi, C. De Persis, and P. Tesi, "Data-driven control via Petersen's lemma," *Automatica*, vol. 145, p. 110537, 2022.

[8] J. Coulson, J. Lygeros, and F. Dörfler, "Data-enabled predictive control: In the shallows of the DeePC," in *18th European Control Conference*, 2019, pp. 307–312.

[9] J. Berberich, J. Köhler, M. Müller, and F. Allgöwer, "Data-driven model predictive control with stability and robustness guarantees," *IEEE Transactions on Automatic Control*, vol. 66, no. 4, pp. 1702–1717, 2020.

[10] Y. Lian and C. N. Jones, "Nonlinear data-enabled prediction and control," in *Learning for Dynamics and Control*. PMLR, 2021, pp. 523–534.

[11] J. Coulson, J. Lygeros, and F. Dörfler, "Distributionally robust chance constrained data-enabled predictive control," *IEEE Transactions on Automatic Control*, vol. 67, no. 7, pp. 3289–3304, 2022.

[12] G. Baggio, D. Bassett, and F. Pasqualetti, "Data-driven control of complex networks," *Nature Communications*, vol. 12, no. 1429, 2020.

[13] F. Celi, G. Baggio, and F. Pasqualetti, "Distributed learning of optimal controls for linear systems," in *2021 60th IEEE Conference on Decision and Control*. IEEE, 2021.

[14] A. Allibhoy and J. Cortés, "Data-based receding horizon control of linear network systems," *IEEE Control Systems Letters*, vol. 5, no. 4, pp. 1207–1212, 2020.

[15] L. Sforni, A. Camisa, and G. Notarstefano, "Structured-policy Q-learning: an LMI-based design strategy for distributed reinforcement learning," in *2022 61st IEEE Conference on Decision and Control*. IEEE, 2022.

[16] A. Bisoffi, C. De Persis, and P. Tesi, "Data-based stabilization of unknown bilinear systems with guaranteed basin of attraction," *Systems & Control Letters*, vol. 145, p. 104788, 2020.

[17] Z. Yuan and J. Cortés, "Data-driven optimal control of bilinear systems," *IEEE Control Systems Letters*, vol. 6, pp. 2479–2484, 2022.

[18] M. Guo, C. De Persis, and P. Tesi, "Data-driven stabilization of nonlinear polynomial systems with noisy data," *IEEE Transactions on Automatic Control*, vol. 67, no. 8, pp. 4210–4217, 2022.

[19] T. Dai and M. Sznaier, "A semi-algebraic optimization approach to data-driven control of continuous-time nonlinear systems," *IEEE Control Systems Letters*, vol. 5, no. 2, pp. 487–492, 2020.

[20] A. Nejati, B. Zhong, M. Caccamo, and M. Zamani, "Data-driven controller synthesis of unknown nonlinear polynomial systems via control barrier certificates," in *4th Annual Learning for Dynamics and Control Conference, PMLR*, 2022.

[21] R. Strässer, J. Berberich, and F. Allgöwer, "Data-driven control of nonlinear systems: Beyond polynomial dynamics," in *IEEE Conference on Decision and Control (CDC)*, 2021, pp. 4344–4351.

[22] T. Dai and M. Sznaier, "Nonlinear data-driven control via state-dependent representations," in *IEEE Conference on Decision and Control*, 2021, pp. 5765–5770.

[23] C.-Y. Chang and A. Bernstein, "Robust data-driven control for systems with noisy data," *arXiv preprint arXiv:2207.09587*, 2022.

[24] C. De Persis, M. Rotulo, and P. Tesi, "Learning controllers from data via approximate nonlinearity cancellation," *IEEE Transactions on Automatic Control (Early Access)*, pp. 1–16, 2023.

[25] J. Umlauft, L. Pöhler, and S. Hirche, "An uncertainty-based control Lyapunov approach for control-affine systems modeled by Gaussian process," *IEEE Control Systems Letters*, vol. 2, no. 3, pp. 483–488, 2018.

[26] J. Umlauft and S. Hirche, "Feedback linearization based on Gaussian processes with event-triggered online learning," *IEEE Transactions on Automatic Control*, vol. 65, no. 10, pp. 4154–4169, 2020.

[27] L. Fraile, M. Marchi, and P. Tabuada, "Data-driven stabilization of SISO feedback linearizable systems," *arXiv:2003.14240*, 2020.

[28] A. Luppi, C. De Persis, and P. Tesi, "On data-driven stabilization of systems with nonlinearities satisfying quadratic constraints," *Systems & Control Letters*, vol. 163, p. 105206, 2022.

[29] S. K. Cheah, D. Bhattacharjee, M. S. Hemati, and R. J. Caverly, "Robust local stabilization of nonlinear systems with controller-dependent norm bounds: A convex approach with input-output sampling," *IEEE Control Systems Letters*, vol. 7, pp. 931–936, 2022.

[30] M. Guo, C. De Persis, and P. Tesi, "Data-driven stabilizer design and closed-loop analysis of general nonlinear systems via Taylor's expansion," *arXiv:2209.01071*, 2022.

[31] T. Martin, D. Schön, and F. Allgöwer, "Gaussian inference for data-driven state-feedback design of nonlinear systems," *arXiv:2212.03225*, 2022.

[32] T. Westenbroek, D. Fridovich-Keil, E. Mazumdar, S. Arora, V. Prabhu, S. Sastry, and C. Tomlin, "Feedback linearization for uncertain systems via reinforcement learning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[33] D. Gadginmath, V. Krishnan, and F. Pasqualetti, "Data-driven feedback linearization using the Koopman generator," *arXiv:2210.05046*, 2022.

[34] M. Alsalti, J. Berberich, V. G. Lopez, F. Allgöwer, and M. A. Müller, "Data-based system analysis and control of flat nonlinear systems," in *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021.

[35] Z. Wang and R. Jungers, "A data-driven immersion technique for linearization of discrete-time systems," in *IFAC PapersOnLine*, 2020, p. 869–874.

[36] E. Kaiser, N. Kutz, and S. Brunton, "Data-driven discovery of Koopman eigenfunctions for control," *Mach. Learn.: Sci. Technol.*, vol. 2, 2021.

[37] M. Haseli and J. Cortés, "Learning Koopman eigenfunctions and invariant subspaces from data: Symmetric subspace decomposition," *IEEE Transactions on Automatic Control*, vol. 67, pp. 3442–3457, 2022.

[38] S. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *PNAS*, vol. 113, no. 15, p. 3932–3937, 2016.

[39] C. De Persis, R. Postoyan, and P. Tesi, "Event-triggered control from data," *arXiv preprint arXiv:2208.11634*, 2022.

[40] A. Isidori, *Nonlinear control systems*, 3rd ed. Springer-Verlag, 1995.

[41] F. Dörfler, J. Coulson, and I. Markovsky, "Bridging direct & indirect data-driven control formulations via regularizations and relaxations," *IEEE Transactions on Automatic Control*, vol. 68, no. 2, pp. 883 – 897, 2023.

[42] C. De Persis and P. Tesi, "Designing experiments for data-driven control of nonlinear systems," in *IFAC-PapersOnLine*, vol. 54 (9), 2021, pp. 285–290.