Koopman-based lifting techniques for nonlinear systems identification

A. Mauroy and J. Goncalves

Abstract—**We develop a novel lifting technique for nonlinear system identification based on the framework of the Koopman operator. The key idea is to identify the linear (infinite-dimensional) Koopman operator in the lifted space of observables, instead of identifying the nonlinear system in the state space, a process which results in a linear method for nonlinear systems identification. The proposed lifting technique is an indirect method that does not require to compute time derivatives and is therefore well-suited to low-sampling rate data sets.**

Considering different finite-dimensional subspaces to approximate and identify the Koopman operator, we propose two numerical schemes: a main method and a dual method. The main method is a parametric identification technique that can accurately reconstruct the vector field of a broad class of systems. The dual method provides estimates of the vector field at the data points and is well-suited to identify high-dimensional systems with small data sets. The present paper describes the two methods, provides theoretical convergence results, and illustrates the lifting techniques with several examples.

I. INTRODUCTION

The problem of identifying governing equations of continuous-time dynamical systems from time-series data has attracted considerable interest in many fields such as biology, finance, and engineering. It is also closely related to network inference, which aims at reconstructing the interactions between the different states of a system, a problem of paramount importance in systems biology. In many cases, the identification problem is challenging due to the nonlinear nature of the systems and must be tackled with black-box methods (e.g. Wiener and Volterra series models [1], nonlinear auto-regressive models [2], neural network models [3], see also [4], [5] for a survey). These methods are related to the classical approach to system identification [6]: they typically deal with long, highly-sampled time-series and provide a relationship between the system inputs and outputs.

In the related context of nonlinear parameter estimation, a large body of methods have been developed to identify the state dynamics of autonomous systems with a known structure (see e.g. [7], [8] and references therein). Typical methods seek the best linear combination of time

A. Mauroy is with the Department of Mathematics and the Namur Institute for Complex Systems (naXys), University of Namur, Belgium (email: alexandre.mauroy@unamur.be). J. Goncalves is with the Luxembourg Centre for Systems Biomedicine, University of Luxembourg, L-4367 Belvaux, Luxembourg (e-mail: jorge.goncalves@uni.lu), and with the Department of Plant Sciences, University of Cambridge, Cambridge CB2 3EA.

derivatives of the state over a set of library functions (similar to the basis functions used in black-box models) [9]. Similar approaches have also been proposed recently, partly motivated by the network identification problem (e.g. Bayesian approach [10], SINDy algorithm [11]). The above-mentioned methods are *direct methods*, and their main advantage is that they rely on static linear regression techniques. However, they assume that time derivatives of the state can be accurately estimated (e.g. by using collocation techniques), a requirement that becomes prohibitive when the sampling time is too low, the measurements too noisy, or the time-series too short (e.g. biology). Instead, *indirect* methods solve an initial value problem and do not require the estimation of time derivatives [12]. Hence, they offer a good alternative to direct methods, but at the expense of solving a (nonconvex) nonlinear least squares problem. The goal of this paper is to propose a new indirect method for estimating the state dynamics (i.e. governing equations) of nonlinear dynamical systems. In the context of nonlinear system identification/parameter estimation, this method not only circumvents the estimation of time derivatives but also relies on linear least squares optimization.

The approach proposed in this paper is based on the framework of the so-called Koopman operator [13], [14]. The Koopman operator is a linear infinite-dimensional operator that describes the evolution of observable-functions along the trajectories of the system. Starting with the seminal work of [15], several studies have investigated the interplay between the spectral properties of the operator and the properties of the associated system, a body of work that has led to new methods for the analysis of nonlinear systems (e.g. global stability analysis [16], global linearization [17], monotone systems [18], delayed systems [19]). While the above-mentioned studies focus on systems described by a known vector field, the Koopman operator approach is also conducive to data analysis and directly connected to numerical schemes such as Dynamic Mode Decomposition (DMD) [20], [21], [22], [23]. This yielded another set of techniques for data-driven analysis and control of nonlinear systems (observer synthesis [24], model predictive control [25], optimal control [26], power systems stability analysis [27], to list a few). In this context, this paper aims at connecting data to vector field, thereby bridging these two sets of methods.

The Koopman operator provides a linear representation of the nonlinear system in a lifted (infinite-dimensional) space of observable-functions. Through this lifting approach, one can therefore *identify the linear Koopman*

operator in the space of observables (see e.g. [28]), instead of identifying the nonlinear system in the state space. Our numerical scheme exploits this idea and proceeds in three steps: (1) lifting of the data, (2) identification of the Koopman operator, and (3) identification of the vector field. In the first step, snapshot data are lifted to the space of observables. In the second step, we derive two distinct methods: (a) a main method which identifies a representation of the Koopman operator in a basis of functions; (b) a dual method which identifies the representation of the operator in the "sample space". In the third step, we connect the vector field to the infinitesimal generator of the identified operator and obtain the linear combination of the vector field in a basis of library functions. The two methods are complemented with convergence results showing that they identify the vector field exactly in optimal conditions. The main method has been initially proposed in [29] and a similar approach developed in a stochastic framework and based on non-convex optimization can also be found in the more recent work [30]. It should be noted that the lifting technique is not new. More precisely, the first steps of our methods (i.e. lifting and identification of the operator) are directly related to a component of the Extended Dynamic Mode Decomposition (EDMD) technique [28] (main method) or inspired from kernel-based EDMD technique [31] (dual method). Although EDMD techniques focus on the spectral properties of the operator, their lifting approach could also be used for prediction [25]. In contrast, the main goal of the two methods proposed in this paper is not to predict trajectories, but to provide a functional representation of the vector field. This representation can be further used for system analysis (e.g. existence of equilibria, stability) and model-based control, and is also directly related to the network identification problem. Note however that the main method has recently been used with success in the context of robot motion prediction [32].

The proposed lifting technique has several advantages. First of all, it relies only on linear methods which are easy and efficient to implement. It is also well-suited to data acquired from short time-series with low sampling rates (e.g. several experiments in biology, with a few costly measurements). Although initially limited to polynomial vector fields, the main method works efficiently with a broad class of behaviors, including unstable and chaotic systems. In addition, the dual method is well-suited to identify large-dimensional systems and to reconstruct network topologies, in particular when the number of sample points is smaller than the unknown system parameters. Finally, lifting techniques can be extended to identify non-polynomial vector fields and open systems (with input or process noise). In contrast to these advantages, a main limitation of the methods is that they require full state measurements and therefore cannot provide an input-output representation of the system.

The rest of the paper is organized as follows. In Section II, we present the problem and introduce the general lifting technique used for system identification. Section III describes the main method and provides theoretical convergence results, while Section IV discusses some extensions of the methods to non-polynomial vector fields and open systems. In Section V, we propose the dual method to identify high-dimensional systems with small data sets and give convergence proofs. The two methods are illustrated with several examples in Section VI, where the network reconstruction problem is also considered. Concluding remarks and perspectives are given in Section VII.

II. Identification in the Koopman operator framework

A. Problem statement

We address the problem of identifying the vector field of a nonlinear system from time series generated by its dynamics. We consider the system

$$\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^n \tag{1}$$

where the vector field $\mathbf{F}(\mathbf{x})$ is of the form

$$\mathbf{F}(\mathbf{x}) = \sum_{k=1}^{N_F} \mathbf{w}_k \, h_k(x). \tag{2}$$

The vectors $\mathbf{w}_k = (w_k^1 \cdots w_k^n)^T \in \mathbb{R}^n$ are unknown coefficients (to be identified) and the library functions h_k are assumed to be known. Note that some coefficients might be equal zero. Unless stated otherwise, we will consider that the vector field is polynomial, so that h_k are monomials: $h_k = p_k$ with

$$p_k(\mathbf{x}) \in \{x_1^{s_1} \cdots x_n^{s_n} | (s_1, \ldots, s_n) \in \mathbb{N}^n, s_1 + \cdots + s_n \leq m_F\} \tag{3}$$

where m_F is the total degree of the polynomial vector field. The number of monomials in the sum (2) is given by $N_F = (m_F + n)!/(m_F! n!)$. As shown in Section IV-C, the proposed method can also be generalized to other types of vector fields in a straightforward way.

Our goal is to identify the vector field $\mathbf{F}$ (i.e. the N_F coefficients $\mathbf{w}_k$) from snapshot measurements of the system trajectories. We consider K snapshot pairs $(\mathbf{x}_k, \mathbf{y}_k)$ obtained from noisy measurements (proportional to the exact state value): we have

$$\mathbf{x}_k = \bar{\mathbf{x}}_k + \epsilon(\mathbf{x}_k) \qquad \mathbf{y}_k = \bar{\mathbf{y}}_k + \epsilon(\mathbf{y}_k) \tag{4}$$

where ϵ is the state-dependent measurement noise, and

$$\bar{\mathbf{y}}_k = \varphi^{T_s}(\bar{\mathbf{x}}_k) \tag{5}$$

where $t \mapsto \varphi^t(\mathbf{x_0})$ is the solution to (1) associated with the initial condition $\mathbf{x_0}$. We assume that the measurement noise is Gaussian and proportional to the state value, i.e. $\epsilon(\mathbf{x}) = \mathbf{x} \odot \mathbf{v}$ where $\odot$ denotes the element-wise product and $\mathbf{v}$ is a Gaussian random variable with zero mean and standard deviation σ_{meas}. We also assume that all pairs $(\mathbf{x}_k, \mathbf{y}_k)$ lie in a compact set $X \subset \mathbb{R}^n$ and are obtained with the same sampling period T_s. They can belong to a single trajectory or to multiple trajectories. Stochastic

systems with process noise and systems with inputs will also be considered (see Section IV).

Remark 1. For numerical reasons, we will assume in general that the data points lie in a set $X \subset [-1,1]^n$. If original data do not satisfy this assumption, then they can be rescaled to yield new data pairs $(\mathbf{x}'_k, \mathbf{y}'_k) = (\mathbf{x}'_k/\alpha, \mathbf{y}'_k/\alpha) \in [-1,1]^{2n}$. These new pairs enable to identify a vector field $\mathbf{F}'(\mathbf{x})$ with coefficients $\mathbf{w}'_k = \alpha^{m_k-1}\mathbf{w}_k$, where m_k is the total degree of the monomial p_k. $\diamond$

B. Koopman operator

System (1) represents the state dynamics in $\mathbb{R}^n$. Alternatively, the system can be described in a lifted space $\mathcal{F}$ of observable-functions $f : \mathbb{R}^n \to \mathbb{R}$. Provided that the observable functions are continuously differentiable, their dynamics in the lifted space are given by

$$\dot{f} = (\mathbf{F} \cdot \nabla)f, \quad f \in \mathcal{F}, \tag{6}$$

where $\dot{f}$ denotes $\partial(f \circ \varphi^t)/\partial t$ (with a slight abuse of notation) and ∇ denotes the gradient (see e.g. [33]). In contrast to (1), the dynamics (6) are infinite-dimensional but linear.

While the flow induced by (1) in the state space is given by the nonlinear flow map φ, the flow induced by (6) in the lifted space is given by the linear semigroup of Koopman operators $U^t : \mathcal{F} \to \mathcal{F}$, $t \geq 0$. This semigroup governs the evolution of the observables along the trajectories, i.e.

$$U^t f = f \circ \varphi^t.$$

Under appropriate conditions (see Section III-C), the semigroup of Koopman operators is strongly continuous and generated by the operator

$$L = \mathbf{F} \cdot \nabla \tag{7}$$

appearing in (6). In this case, we use the notation

$$U^t = e^{Lt}. \tag{8}$$

The operator L is called the infinitesimal generator of the Koopman operator and we denote its domain by $\mathcal{D}(L)$.

C. Linear identification in the lifted space

There is a one-to-one correspondence between systems of the form (1) and lifted systems (6), or equivalently between the flow φ^t and the semigroup of Koopman operators U^t. Exploiting this equivalence, we propose to solve the identification problem in the lifted space instead of the state space. This can be done in three steps (see Figure 1).

1) **Lifting of the data.** Snapshots pairs $(\mathbf{x}_k, \mathbf{y}_k)$ are lifted to the space of observable by constructing new pairs of the form $(g(\mathbf{x}_k), g(\mathbf{y}_k))$ for some $g \in \mathcal{F}$. The functions g are assumed to be continuously differentiable and we call them *basis functions*. It follows from (4) and (5) that

$$\begin{aligned} g(\mathbf{y}_k) &= g(\varphi^{T_s}(\mathbf{x}_k - \epsilon(\mathbf{x}_k)) + \epsilon(\mathbf{y}_k)) \\ &\approx U^{T_s}g(\mathbf{x}_k) + \mathcal{O}(\|\epsilon\|). \end{aligned} \tag{9}$$

2) **Identification of the Koopman operator.** A finite-dimensional projection of the Koopman operator is obtained through a classical linear identification method that is similar to a component of the Extended Dynamic Mode Decomposition (EDMD) algorithm [28]. This yields (an approximation of) the infinitesimal generator L of the Koopman operator.

3) **Identification of the vector field.** Using (7), we can finally obtain the vector field $\mathbf{F}$.
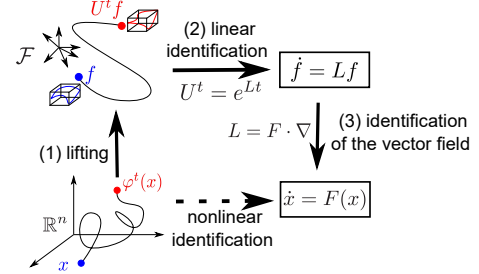

Figure 1. Classical nonlinear system identification is performed directly in the state space. In contrast, the proposed lifting technique consists of three steps: (1) lifting of the data; (2) linear identification of the Koopman operator in the lifted space; (3) identification of the vector field.

III. THE MAIN LIFTING METHOD

A. Description of the method

This section describes in detail the three steps of our main method. The first step and the first part of the second step are related to a component of the EDMD algorithm (see [28] for more details).

1) First step - lifting of the data: The data must be lifted to the infinite-dimensional space $\mathcal{F}$ of observables. However, the method has to be numerically tractable and is developed in a finite-dimensional linear subspace $\mathcal{F}_N \subset \mathcal{F}$ spanned by a basis of N linearly independent functions. The choice of basis functions $\{g_k\}_{k=1}^N$ can be arbitrary (e.g. Fourier basis, radial basis functions), but might affect the method performances. Since the vector field is assumed to be polynomial, we naturally choose the basis of monomials $\{g_k\}_{k=1}^N = \{p_k\}_{k=1}^N$ with total degree less or equal to m to facilitate the representation of the Koopman operator. The number of basis functions is equal to $N = (n+m)!/(n!\,m!)$. We impose $m \geq m_F$.

For each snapshot pair $(\mathbf{x}_k, \mathbf{y}_k) \in \mathbb{R}^{n\times 2}$, $k \in \{1, \ldots, K\}$, we construct a new pair $(\mathbf{p}(\mathbf{x}_k), \mathbf{p}(\mathbf{y}_k)) \in \mathbb{R}^{N\times 2}$, where $\mathbf{p}(\mathbf{x}) = (p_1(\mathbf{x}), \ldots, p_N(\mathbf{x}))^T$ denotes the vector of basis monomials. In the following, we will also use the $K \times N$ matrices

$$\mathbf{P_x} = \begin{pmatrix} \mathbf{p}(\mathbf{x}_1)^T \\ \vdots \\ \mathbf{p}(\mathbf{x}_K)^T \end{pmatrix} \qquad \mathbf{P_y} = \begin{pmatrix} \mathbf{p}(\mathbf{y}_1)^T \\ \vdots \\ \mathbf{p}(\mathbf{y}_K)^T \end{pmatrix}. \tag{10}$$

2) Second step - identification of the Koopman operator: Now we proceed to the identification of the Koopman operator U^t, for $t = T_s$. More precisely, we will identify the finite-rank operator $U_N : \mathcal{F}_N \to \mathcal{F}_N$ of the form $U_N =$

$P_N U^{T_s}|_{\mathcal{F}_N}$, where $P_N : \mathcal{F} \to \mathcal{F}_N$ is a projection operator onto the subspace $\mathcal{F}_N$ and where $U^t|_{\mathcal{F}_N} : \mathcal{F}_N \to \mathcal{F}$ is the restriction of the Koopman operator to $\mathcal{F}_N$. Considering

$$f = \mathbf{a}^T \mathbf{p}, \qquad U_N f = \mathbf{b}^T \mathbf{p}, \tag{11}$$

we can define a matrix $\overline{\mathbf{U}}_N \in \mathbb{R}^{N \times N}$ such that

$$\overline{\mathbf{U}}_N \mathbf{a} = \mathbf{b}. \tag{12}$$

The matrix $\overline{\mathbf{U}}_N$ is a representation of the projected Koopman operator U_N. It also provides an approximate finite-dimensional linear description of the nonlinear system. This description is not obtained through local linearization techniques and is valid globally.

It follows from (11) and (12) that

$$U_N f = U_N(\mathbf{a}^T \mathbf{p}) = (\overline{\mathbf{U}}_N \mathbf{a})^T \mathbf{p} \tag{13}$$

and, since (13) holds for all $\mathbf{a}$, we have

$$[U_N p_1 \cdots U_N p_N]^T = \mathbf{p}^T \overline{\mathbf{U}}_N, \tag{14}$$

where the operator U_N acts on each component of the vector $\mathbf{p}$. By considering each column separately, we obtain $P_N U^{T_s} p_j = U_N p_j = \mathbf{c}_j^T \mathbf{p}$, where $\mathbf{c}_j$ is the jth column of $\overline{\mathbf{U}}_N$. This shows that each column of $\overline{\mathbf{U}}_N$ is related to the projection onto $\mathcal{F}_N$ of the image of a basis function p_j through the Koopman operator U^{T_s}.

We consider here a discrete orthogonal projection yielding the least squares fit at the points $\mathbf{x}_k$, $k = 1, \ldots, K$, with $K \geq N$:

$$P_N g = \underset{\tilde{g} \in \text{span}\{p_1, \ldots, p_N\}}{\text{argmin}} \sum_{k=1}^{K} |\tilde{g}(\mathbf{x}_k) - g(\mathbf{x}_k)|^2. \tag{15}$$

This corresponds to the least squares solution

$$P_N g = \mathbf{p}^T \mathbf{P}_{\mathbf{x}}^\dagger \begin{pmatrix} g(\mathbf{x}_1) \\ \vdots \\ g(\mathbf{x}_K) \end{pmatrix}$$

where $\mathbf{P}^\dagger$ denotes the pseudoinverse of $\mathbf{P}$. For $g = U^{T_s} p_j$, we obtain

$$P_N(U^{T_s} p_j) = \mathbf{p}^T \mathbf{P}_{\mathbf{x}}^\dagger \begin{pmatrix} U^{T_s} p_j(\mathbf{x}_1) \\ \vdots \\ U^{T_s} p_j(\mathbf{x}_K) \end{pmatrix} \approx \mathbf{p}^T \mathbf{P}_{\mathbf{x}}^\dagger \begin{pmatrix} p_j(\mathbf{y}_1) \\ \vdots \\ p_j(\mathbf{y}_K) \end{pmatrix}$$

where we used (9) evaluated at the states $\mathbf{x}_k$ and assumed that measurement noise $\|\epsilon\|$ is small. Equivalently, we have $U_N \mathbf{p}^T \approx \mathbf{p}^T \mathbf{P}_{\mathbf{x}}^\dagger \mathbf{P}_{\mathbf{y}}$ so that (14) yields

$$\overline{\mathbf{U}}_N \approx \mathbf{P}_{\mathbf{x}}^\dagger \mathbf{P}_{\mathbf{y}}. \tag{16}$$

Inspired by (8), we finally compute

$$\overline{\mathbf{L}}_{\mathbf{data}} = \frac{1}{T_s} \log(\mathbf{P}_{\mathbf{x}}^\dagger \mathbf{P}_{\mathbf{y}}), \tag{17}$$

where the function log denotes the (principal) matrix logarithm. The matrix $\overline{\mathbf{L}}_{\mathbf{data}}$ is an approximation of the matrix representation $\overline{\mathbf{L}}_N$ of $L_N = P_N L|_{\mathcal{F}_N}$, where $L_N f = \mathbf{p}^T(\overline{\mathbf{L}}_N \mathbf{a})$ for all $f = \mathbf{p}^T \mathbf{a}$. A rigorous justification is given in Section III-C.

Remark 2. Even with no measure noise, $\overline{\mathbf{L}}_{\mathbf{data}}$ is only an approximation of $\overline{\mathbf{L}}_N$. Indeed, $\overline{\mathbf{L}}_{\mathbf{data}}$ is the matrix representation of the finite-rank operator $\frac{1}{T_s} \log(P_N U^{T_s}|_{\mathcal{F}_N}) = \frac{1}{T_s} \log(P_N e^{LT_s}|_{\mathcal{F}_N}) \neq P_N L|_{\mathcal{F}_N}$. The two matrices $\overline{\mathbf{L}}_{\mathbf{data}}$ and $\overline{\mathbf{L}}_N$ are identical only in the limit $N \to \infty$ and under some additional conditions related to the non-uniqueness of the matrix logarithm (see Section III-C for the details). ◇

3) Third step - identification of the vector field: We are now in position to identify the coefficients $\mathbf{w}_k = (w_k^1 \cdots w_k^n)$ of the vector field. With the basis function $p_l(\mathbf{x}) = x_j$ where x_j is the jth component of $\mathbf{x}$, we have

$$L_N p_l = P_N (\mathbf{F} \cdot \nabla p_l) = P_N F_j = F_j.$$

Since $L_N p_l = \mathbf{p}^T(\overline{\mathbf{L}}_N \mathbf{e}_l)$, it follows that

$$F_j = \mathbf{p}^T(\overline{\mathbf{L}}_N \mathbf{e}_l) \approx \mathbf{p}^T(\overline{\mathbf{L}}_{\mathbf{data}} \mathbf{e}_l), \tag{18}$$

i.e. the lth column of $\overline{\mathbf{L}}_{\mathbf{data}}$ contains the estimates $\hat{w}_k^j$. Equivalently, we have

$$\hat{w}_k^j = \left[\overline{\mathbf{L}}_{\mathbf{data}}\right]_{kl}. \tag{19}$$

Remark 3 (Nonlinear least squares problem)**.** The identification problem could also be performed at the level of the Koopman semigroup. However solving the equality $\overline{\mathbf{U}} = e^{\overline{\mathbf{L}} T_s}$ (with a square matrix $\overline{\mathbf{L}}$) amounts to solving a (nonconvex) nonlinear least squares problem (as done in [30]). This might also be equivalent to solving the direct identification problem with an exact Taylor discretization of time-derivatives [34]. ◇

Remark 4. The vector field coefficients are obtained with n columns of $\overline{\mathbf{L}}_{\mathbf{data}}$ related to the monomials of degree 1. Instead, we could use all columns $\overline{\mathbf{L}}_{\mathbf{data}}$. In this case, the coefficients are the solutions to an overdetermined set of equations, which could be solved by promoting sparsity (e.g. Lasso). More details can be found in [29]. However, numerical experiments suggest that this does not improve the results. ◇

Remark 5 (Estimation of the vector field values)**.** If needed, the method can directly provide the values $\mathbf{F}(\mathbf{x}_k)$ of the vector field. Evaluating (18) at $\mathbf{x}_k$ for all $k = 1, \ldots, K$, we obtain an approximation $\hat{F}_j$ of the vector field given by

$$\begin{pmatrix} \hat{F}_j(\mathbf{x}_1) \\ \vdots \\ \hat{F}_j(\mathbf{x}_K) \end{pmatrix} = \mathbf{P}_{\mathbf{x}} (\overline{\mathbf{L}}_{\mathbf{data}} \mathbf{e}_l) \tag{20}$$

with $p_l(\mathbf{x}) = x_j$. This is quite similar to the approach developed with the dual method presented in Section V. ◇

B. Algorithm

Our main lifting method for system identification is summarized in Algorithm 1.

Algorithm 1 Main lifting method for nonlinear system identification

Input: Snapshot pairs $\{(\mathbf{x}_k, \mathbf{y}_k)\}_{k=1}^K$, $\mathbf{x}_k \in \mathbb{R}^n$; sampling period T_s; integers $m \geq 1$ and $m_F \geq 0$ (with $m \geq m_F$).
Output: Estimates $\hat{w}_k^j$ (and $\hat{F}_j$).
1: $N := (m+n)!/(m!n!)$; $N_F := (m_F+n)!/(m_F!n!)$
2: **while** $N > K$ **do**
3:     Increase K (add snapshot pairs) or decrease m
4: **end while**
5: Construct the $K \times N$ matrices $\mathbf{P_x}$ and $\mathbf{P_y}$ defined in (10)
6: Compute the $N \times N$ matrix $\overline{\mathbf{L}_{\mathbf{data}}}$ defined in (17)
7: $\hat{w}_k^j := \left[\overline{\mathbf{L}_{\mathbf{data}}}\right]_{kl}$, with l such that $p_l(\mathbf{x}) = x_j$
    Alternatively, $\hat{F}_j$ is obtained with (20)

C. Theoretical results

In this section, we prove the convergence of Algorithm 1 in optimal conditions, i.e. with an infinite number of data points and basis functions, and an arbitrarily high sampling rate.

We consider the space $\mathcal{F} = L^2(X)$ (where $\|\cdot\|$ is the L^2 norm) and the subspace $\mathcal{F}_N$ spanned by the monomials $\{p_k\}_{k=1}^N$. We will further assume that the flow induced by (1) is invertible and nonsingular [1], and that X is forward-invariant (i.e. $\varphi^t(X) \subseteq X$ for all $t > 0$) or backward-invariant [2] (i.e. $\varphi^{-t}(X) \subseteq X$ for all $t > 0$). Under these conditions, we can check that the semigroup U^t is strongly continuous. For continuous functions $g : X \to \mathbb{R}$, which are dense in $L^2(X)$, we have $\lim_{t\to 0} \|g - U^t g\| = 0$. Moreover, we have

$$\|U^t f\|^2 = \int_X |U^t f(x)|^2 dx = \int_{\varphi^t(X)} |f(x)|^2 |J_{\varphi^{-t}}(x)| dx$$
$$\leq \max_{x \in X} |J_{\varphi^{-t}}(x)| \|f\|^2$$

or equivalently

$$\frac{\|U^t f\|^2}{\|f\|^2} \leq \max_{x \in X} |J_{\varphi^t}(x)|^{-1}$$

where $|J_{\varphi^t}(x)|$ is the determinant of the Jacobian matrix of $\varphi^t(x)$. Since the flow is nonsingular, $|J_{\varphi^t}(x)| \neq 0$ implies that U^t is bounded. It follows that the semigroup of Koopman operators U^t is strongly continuous (see e.g. [35, Proposition I.5.3(c)]).

We are now in position to show that Algorithm 1 yields exact estimates $\hat{w}_k^j$ and $\hat{F}_j = \sum_{k=1}^N \hat{w}_k^j p_k$ in optimal conditions.

Theorem 1. *Assume that the sample points $\mathbf{x}_k$ are uniformly randomly distributed in a compact forward or backward invariant set X, and consider $\mathbf{y}_k = \varphi^{T_s}(\mathbf{x}_k)$*

[1] The flow is nonsingular if $\mu(A) \neq 0$ implies $\mu(\varphi^t(A)) \neq 0$ for all $A \in \mathbb{R}^n$ and all $t > 0$, where μ is the Lebesgue measure. This is a generic condition that is satisfied when the vector field F is Lipschitz continuous, for instance.

[2] When X is backward-invariant, we assume that $f(x) = 0$ for all $x \notin X$ and all $f \in \mathcal{F}$, so that U^t is a well-defined semigroup.

(no measurement noise) where φ^t is an invertible and nonsingular flow generated by the dynamics (1). If the Algorithm 1 is used with the data pairs $\{\mathbf{x}_k, \mathbf{y}_k\}_{k=1}^K$ (with $K \geq N$) and with a set of basis functions $\{p_k\}_{k=1}^\infty$ whose span is dense in $L^2(X)$ and which contains the identity function $id_j(\mathbf{x}) = x_j$, then the estimated vector field $\hat{F}_j = \sum_{k=1}^N \hat{w}_k^j p_k$ satisfies

$$\lim_{N\to\infty} \lim_{K\to\infty} \lim_{T_s\to 0} \left\|\hat{F}_j - F_j\right\| = 0$$

with probability one. Moreover, if the vector field is of the form (2) with $h_k = p_k$ (monomials), then

$$\lim_{T_s\to 0} \hat{w}_k^j = w_k^j, \qquad k = 1, \ldots, N_F$$

with probability one for all $N \geq N_F$.

Proof. Since $\mathbf{x}_k \in X$, the discrete orthogonal projection (15) P_N is a well-defined projection (with probability one) from $L^2(X)$ to $\mathcal{F}_N \subset L^2(X)$. For a finite integer N, consider the finite-dimensional operators $A_N^{T_s} = e^{P_N L P_N T_s} : \mathcal{F}_N \to \mathcal{F}_N$ and $U_N^{T_s} = P_N U^{T_s}|_{\mathcal{F}_N}$. Since $\|A_N^{T_s} - I\| \to 0$ and $\|U_N^{T_s} - I\| \to 0$ as $T_s \to 0$, it follows that

$$\lim_{T_s\to 0} \frac{\|\log A_N^{T_s} - (A_N^{T_s} - I)\|}{T_s} = 0$$
$$\lim_{T_s\to 0} \frac{\|\log U_N^{T_s} - (U_N^{T_s} - I)\|}{T_s} = 0$$

(The eigenvalues $\lambda(T_s)$ of $A_N^{T_s}$ and $U_N^{T_s}$ satisfy $|\lambda(T_s) - 1| \to 0$, which implies by L'Hôpital's rule that $(\log \lambda(T_s) - (\lambda(T_s) - 1))/T_s \to 0$.) We also have, for all $f \in \mathcal{F}_N$,

$$\left\|(A_N^{T_s} - U_N^{T_s})f\right\| = \left\|\int_0^{T_s} \frac{d}{d\tau}\left(A_N^\tau U_N^{T_s-\tau}\right) f d\tau\right\|$$
$$= \int_0^{T_s} \left\|A_N^\tau (P_N L P_N - P_N L) U^{T_s-\tau} f\right\| d\tau$$
$$= \int_0^{T_s} \left\|A_N^\tau P_N L P_N (U^{T_s-\tau} f - f)\right.$$
$$+ A_N^\tau (P_N L P_N f - P_N L f)$$
$$\left. + A_N^\tau P_N (Lf - L U^{T_s-\tau} f)\right\| d\tau$$
$$\leq \int_0^{T_s} \|A_N^\tau P_N L P_N\| \left\|U^{T_s-\tau} f - f\right\|$$
$$+ \|A_N^\tau (P_N L P_N - P_N L)f\|$$
$$+ \|A_N^\tau P_N\| \left\|Lf - U^{T_s-\tau} Lf\right\| d\tau$$

where we used the fact that L and $U^{T_s-\tau}$ commute. Since U^{T_s} is strongly continuous and $P_n f = f$, it follows from the mean value theorem that

$$\lim_{T_s\to 0} \frac{1}{T_s} \left\|(A_N^{T_s} - U_N^{T_s})f\right\| = 0 \qquad \forall f \in \mathcal{F}_N.$$

Then we get

$$\lim_{T_s\to 0} \frac{1}{T_s} \|(\log A_N^{T_s} - \log U_N^{T_s})f\|$$
$$\leq \lim_{T_s\to 0} \left(\frac{1}{T_s} \|(\log A_N^{T_s} - (A_N^{T_s} - I))f\|\right.$$
$$\left. + \frac{1}{T_s}\|(\log U_N^{T_s} - (U_N^{T_s} - I))f\| + \frac{1}{T_s}\|(A_N^{T_s} - U_N^{T_s})f\|\right) = 0.$$

Since there is no measurement noise, (17) is the (exact) matrix representation of $U_N^{T_s}$ and (20) is equivalent to $\hat{F}_j = \log U_N^{T_s} \mathrm{id}_j / T_s$. This implies that

$$\lim_{T_s \to 0} \left\| \hat{F}_j - P_N F_j \right\| = \lim_{T_s \to 0} \left\| \frac{\log U_N^{T_s} \mathrm{id}_j}{T_s} - P_N L P_N \mathrm{id}_j \right\|$$
$$= \lim_{T_s \to 0} \frac{1}{T_s} \left\| \log U_N^{T_s} \mathrm{id}_j - \log A_N^{T_s} \mathrm{id}_j \right\| = 0$$
(21)

with the identity function $\mathrm{id}_j \in \mathcal{F}_N$.

The discrete orthogonal projection converges in the strong operator topology with probability one to the L^2 projection as $K \to \infty$ (see e.g. [36] for a proof). Since the basis is complete in $L^2(X)$, the orthogonal projection converges in the strong operator topology to the identity operator as $N \to \infty$, with probability one. It follows that

$$\lim_{N \to \infty} \lim_{K \to \infty} \lim_{T_s \to 0} \left\| \hat{F}_j - F_j \right\|$$
$$\leq \lim_{N \to \infty} \lim_{K \to \infty} \left(\lim_{T_s \to 0} \left\| \hat{F}_j - P_N F_j \right\| + \| P_N F_j - F_j \| \right) = 0 \,.$$

Finally, if the vector field is polynomial with $N_F \leq N$, we have $P_N F_j = F_j$ (with probability one) and it follows from (21) that $\lim_{T_s \to 0} \hat{w}_k^j - w_k^j = 0$ for all k. $\square$

According to Theorem 1, Algorithm 1 identifies exactly the vector field, even if the data are collected in a small region of the state space (this is made possible by the a priori knowledge that the vector field is polynomial). Note that the requirement to collect the data points on an invariant set might not always be satisfied in practice. This is however a technical condition that ensures that U^t is a well-defined semigroup of operators on $[0, T_s]$. However, the result shows that an infinite sampling frequency ($T_s \to 0$) is required by the use of the matrix logarithm. This issue is related to the so-called system aliasing and is discussed with more details in [37]. Intuitively, an infinite sampling rate is needed to capture the infinity of frequencies that characterize a nonlinear system. This condition ensures in particular that the eigenvalues of $T_s P_N L P_N$ lie in the strip $\{ z \in \mathbb{C} : |\Im\{z\}| < \pi \}$ so that the properties of the principal branch of the logarithm imply that $\frac{1}{T_s} \log e^{T_s P_N L P_N} = P_N L P_N$ in (21). In the case of polynomial vector fields, it is noticeable that the number of basis functions does not need to tend to infinity. In fact, when T_s tends to zero, $\log U_N^{T_s} \mathrm{id}_j / T_s \approx (I - U_N^{T_s}) \mathrm{id}_j / T_s$ corresponds to the first order approximation of the time derivative $\dot{x}_j = F_j$ and we recover a direct method based on the computation of time derivatives. In practice, with a possibly large sampling time, it can be useful to increase the number of basis functions N. The Trotter-Kato approximation theorem (see e.g. [35, Theorem 4.8]) implies that $\|(A_N^{T_s} - U_N^{T_s})f\| \to 0$ as $N \to \infty$ for all $f \in \mathcal{F}_N$, so that one can expect that the error $\|(\log A_N^{T_s} - \log U_N^{T_s})\mathrm{id}_j\|$ decreases for larger values N. This is confirmed with numerical simulations suggesting that small estimation errors can be obtained with a sampling period $T_s = \mathcal{O}(1)$ provided that N is large enough.

The above theoretical results are valid only when there is no measurement noise. In presence of noise, the estimator is biased and not consistent, because of the lifting of the data. However, the algorithm performs well for small measurement noise levels and is also shown to be robust to process noise in Section VI.

IV. EXTENSIONS

We now consider several extensions of the proposed method, which allow to identify open systems driven by a known input or a white noise (i.e. process noise) and to identify systems with non-polynomial vector fields.

A. Systems with inputs

Consider an open dynamical system of the form

$$\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}, \mathbf{u}(t)) \tag{22}$$

with $\mathbf{x} \in \mathbb{R}^n$ and with the input $u \in \mathcal{U} : \mathbb{R}^+ \to \mathbb{R}^p$. We assume that the vector field consists of monomials in $\mathbf{x}$ and $\mathbf{u}$. We define the associated flow $\varphi : \mathbb{R}^+ \times \mathbb{R}^n \times \mathcal{U}$ so that $t \mapsto \varphi(t, \mathbf{x}, \mathbf{u}(\cdot))$ is a solution of (22) with the initial condition $\mathbf{x}$ and the input $\mathbf{u}(\cdot)$. Following the generalization proposed in [38], [39], [40], we consider observables $f : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}$ and define the semigroup of Koopman operators

$$U^t f(\mathbf{x}, \mathbf{u}) = f(\varphi^t(\mathbf{x}, \mathbf{u}(\cdot) = \mathbf{u}), \mathbf{u})$$

where $\mathbf{u}(\cdot) = \mathbf{u}$ is a constant input. In this case, $\mathbf{u}$ can be considered as additional state variables and the above operator is the classical Koopman operator for the augmented system $\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}, \mathbf{u})$, $\dot{\mathbf{u}} = \mathbf{0}$. In particular, the infinitesimal generator is still given by (7).

It follows that the method proposed in Sections III-A1 and III-A2 can be used if

$$\varphi^{T_s}(\mathbf{x}, \mathbf{u}(\cdot)) \approx \varphi^{T_s}(\mathbf{x}, \mathbf{u}(\cdot) = \mathbf{u}(0)) \,.$$

This condition holds when the input can be considered as constant between two snapshots (zero-order hold assumption), or equivalently if the sampling rate is high enough. The matrix $\overline{\mathbf{U}}_N$ is now obtained with snapshot pairs $([\mathbf{x}_k, \mathbf{u}_k], [\mathbf{y}_k, \mathbf{u}_k]) \in \mathbb{R}^{(n+p) \times 2}$ and the rest of the procedure follows on similar lines with the augmented state space $\mathbb{R}^{n+p}$. In this case, the identification method not only provides the vector field coefficients associated with the state $\mathbf{x}$, but also those associated with the input $\mathbf{u}$. The efficiency of the method is illustrated in Section VI-B.

B. Process noise

We have considered so far only measurement noise. We show that the proposed method is also robust to process noise. Consider a system described by the stochastic differential equation

$$dx = F(\mathbf{x})dt + \sigma \, dw(t) \tag{23}$$

where $w(t)$ is the Wiener process. We define the flow $\varphi : \mathbb{R}^+ \times \mathbb{R}^n \times \Omega \to \mathbb{R}^n$, where Ω is the probability space, such

that $t \mapsto \varphi(t, \mathbf{x}, \omega)$ is a solution to (23). In this case, the semigroup of Koopman operators is defined by (see e.g. [15])

$$U^t f(\mathbf{x}) = \mathbb{E}[f(\varphi(t, \mathbf{x}, \omega))]$$

and its infinitesimal generator is given by

$$Lf = \mathbf{F} \cdot \nabla f + \frac{\sigma_{proc}^2}{2} \Delta f \qquad (24)$$

where $\Delta = \sum_k \partial^2 / \partial x_k^2$ denotes the Laplacian operator that accounts for diffusion. The infinitesimal generator is related to the so-called Kolmogorov backward equation.

The numerical scheme of the proposed identification method does not need to be adapted to take process noise into account. As explained in [28], the first step of the method (Section III-A1) is still valid for identifying the matrix $\overline{\mathbf{U}}$. In the second step (Section III-A2), the procedure is the same, except that one has to consider the infinitesimal generator whose matrix representation is given by $\overline{\mathbf{L}} + \sigma^2/2\overline{\mathbf{D}}$, where $\overline{\mathbf{D}}$ is the matrix representation of the Laplacian operator. For all l such that $p_l(\mathbf{x}) = x_j$ for some j, $\Delta p_l = 0$ so that the lth column of $\overline{\mathbf{D}}$ contains only zeros. It follows that we can still use (19) to compute the vector field coefficients. In Section VI-B, an example illustrates the robustness of the method against process noise.

Similar methods are also considered with manifold learning techniques (e.g. diffusion maps) for state estimation [41] and embedded vector field estimation [42]. Although developed in another context to solve different problems, these techniques are similar to our method in the sense that they rely on the backward Kolmogorov equation, which is directly connected to the infinitesimal generator (24).

C. Non polynomial vector fields

The method can be adapted to identify non polynomial vector fields of the form (2), where the library functions h_k are not monomials. In this context, the vector field and the library functions do not need to be analytic. In this case, one could consider the equality

$$P_N L = \sum_{j=1}^{n} \sum_{k=1}^{N_F} w_k^j P_N L_k^j$$

with the operators $L_k^j = h_k \partial / \partial x_j$. The final-dimensional representation of this equality yields a matrix equation that should be solved to compute the coefficients w_k^j.

However, using the projection P_N adds an additional error to the finite-dimensional approximation of the operator. Instead, we prefer to consider an "augmented" subspace that contains the library functions:

$$\mathcal{F}'_N = \mathcal{F}_N \times \text{span}\left(\{h_k\}_{k=1}^{N_F}\right)$$

where $\mathcal{F}$ is a subspace spanned by monomials. In this case, we can still use Algorithm 1 with the projection $P'_N : \mathcal{F} \rightarrow \mathcal{F}'_N$. The result of Theorem 1 could also be

extended to this case, provided that the set of basis functions is complete in L^2.

We finally note that a more straightforward method is to perform a least squares regression on the values of the vector field at the sample points, values which can be obtained according to Remark 5. However, numerical experiments suggest that this method is less efficient than the above-mentioned method.

V. A DUAL LIFTING METHOD FOR LARGE SYSTEMS

A major limitation of the main method presented in Section III (Algorithm 1) is that it might require a large number of data points. Indeed, the number of data points must be larger than the number of basis functions ($K \geq N$) to ensure that the discrete orthogonal projection (15) is well-defined. In the case of high-dimensional systems in particular, the number of basis functions is huge and is likely to exceed the number of available data points, an issue which might be critical in fields such as biology. Moreover, the algorithm might also be computationally intractable (e.g. computation of the matrix logarithm in (17)). In this section, we circumvent the above limitations by proposing a dual approach, which is developed in a K-dimensional "sample space" instead of the N-dimensional functional space. This method can be used when the number of basis functions is larger than the number of data points, i.e. $N \geq K$.

A. Description of the method

Similarly to the main lifting method, the dual method consists of three steps: lifting of the data, identification of the Koopman operator, and identification of the vector field. In the last step, the algorithm provides the value of the vector field at each data point, so that the dual method can be seen as an indirect method for time derivatives estimation. This is similar in essence to the vector field estimation detailed in Remark 5. The identification is achieved in a distributed way, a feature which makes the algorithm computationally tractable in the case of high-dimensional systems and well-suited to parallel computing.

1) First step - lifting of the data: This step is similar to the first step of the main method (Section III-A1). But in this case, choosing the basis functions equal to the library functions of the vector field is not more convenient for the next steps. Even if the vector field is polynomial, we can therefore consider other basis than monomials, such as Gaussian radial basis functions $g_k(\mathbf{x}) = e^{-\gamma \|\mathbf{x} - \mathbf{x}_k\|^2}$ with $k = 1, \ldots, K$ and where $\gamma > 0$ is a parameter. We construct the data $K \times N$ matrices

$$\mathbf{P_x} = \begin{pmatrix} \mathbf{g}(\mathbf{x}_1)^T \\ \vdots \\ \mathbf{g}(\mathbf{x}_K)^T \end{pmatrix} \qquad \mathbf{P_y} = \begin{pmatrix} \mathbf{g}(\mathbf{y}_1)^T \\ \vdots \\ \mathbf{g}(\mathbf{y}_K)^T \end{pmatrix} \qquad (25)$$

where $\mathbf{g}$ is the vector of basis functions g_k. When using Gaussian radial basis functions, the number of basis functions is equal to the number of samples (i.e. $N = K$) and therefore does not depend on the dimension n. This is

particularly useful in the case of high-dimensional systems, where the matrices (25) should be of reasonable size. In Section VI, we will only use Gaussian radial basis functions.

2) Second step - identification of the Koopman operator: We use a dual matrix representation of the Koopman operator, which is inspired (but slighted different, see Remark 6) from a kernel-based approach developed in [31] (kernel EDMD).

In the main method, we constructed the $N \times N$ matrix $\overline{\mathbf{U}}_N \approx \mathbf{P}_\mathbf{x}^\dagger \mathbf{P}_\mathbf{y}$ which represents the operator U_N. Instead, we can consider the $K \times K$ matrix representation

$$\widetilde{\mathbf{U}}_K \approx \mathbf{P}_\mathbf{y} \mathbf{P}_\mathbf{x}^\dagger = \mathbf{P}_\mathbf{x} \overline{\mathbf{U}}_N \mathbf{P}_\mathbf{x}^\dagger, \qquad (26)$$

a construction which is similar to the original formulation of the Dynamic Mode Decomposition (DMD) algorithm[3] [23]. The matrix $\mathbf{P}_\mathbf{x}$ can be interpreted as a change of coordinates, and $\widehat{\mathbf{U}}_K$ appears to be the matrix representation of U^{T_s} in the "sample space": for all $f \in \mathcal{F}_N$, we have

$$\begin{pmatrix} U^{T_s} f(\mathbf{x}_1) \\ \vdots \\ U^{T_s} f(\mathbf{x}_K) \end{pmatrix} \approx \widetilde{\mathbf{U}}_K \begin{pmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_K) \end{pmatrix}. \qquad (27)$$

We have seen that the jth column $\mathbf{c}_j$ of $\overline{\mathbf{U}}_N$ satisfies $\mathbf{P}_\mathbf{x} \mathbf{c}_j \approx (p_j(\mathbf{y}_1) \cdots p_j(\mathbf{y}_K))^T$ and corresponds to the projection (15) of $U^{T_s} p_j$ on $\mathcal{F}_N$ (expressed in the basis of functions). Each of the K data points yields a constraint and there are N unknowns, so that $K \geq N$ is required. In contrast, it follows from (27) that the ith row $\mathbf{r}_i$ of $\widetilde{\mathbf{U}}_K$ can be seen, for all f, as the coefficients of the linear combination of the values $f(\mathbf{x}_1), \ldots, f(\mathbf{x}_K)$ that is equal to $U^{T_s} f(\mathbf{x}_i)$. The row $\mathbf{r}_i$ satisfies $r_i \mathbf{P}_\mathbf{x} \approx (g_1(\mathbf{y}_i) \cdots g_N(\mathbf{y}_i))$, i.e. r_i is obtained by considering the N "test" functions g_j. In this case, each of the N functions yields a constraint and there are K unknowns, so that $K \leq N$ is required.

Remark 6. Following similar lines as in [31], we note that we have

$$\widetilde{\mathbf{U}}_K \approx \mathbf{P}_\mathbf{y} \mathbf{P}_\mathbf{x}^\dagger = \mathbf{P}_\mathbf{y} \mathbf{P}_\mathbf{x}^\mathbf{T} (\mathbf{P}_\mathbf{x} \mathbf{P}_\mathbf{x}^\mathbf{T})^\dagger \triangleq \mathbf{A}\, \mathbf{G}^\dagger$$

where the entries of $\mathbf{A}$ and $\mathbf{G}$ can be interpreted as the inner products

$$[\mathbf{A}]_{ij} = \mathbf{p}(\mathbf{x}_j)^T \mathbf{p}(\mathbf{y}_i), \quad [\mathbf{G}]_{ij} = \mathbf{p}(\mathbf{x}_j)^T \mathbf{p}(\mathbf{x}_i)$$

(Here, we consider without loss of generality that the matrices $\mathbf{P}_\mathbf{x}$ and $\mathbf{P}_\mathbf{y}$ are constructed with monomials.) The inner products can be approximated by a Gaussian kernel function $g(\mathbf{x}_i, \mathbf{x}_j) = g_j(\mathbf{x}_i)$, so that

$$[\mathbf{A}]_{ij} = g(\mathbf{x}_i, \mathbf{x}_j), \quad [\mathbf{G}]_{ij} = g(\mathbf{y}_i, \mathbf{x}_j).$$

In this context, constructing $\mathbf{P}_\mathbf{x}$ and $\mathbf{P}_\mathbf{y}$ with Gaussian radial basis functions is equivalent to constructing the inner-product matrices $\mathbf{A}$ and $\mathbf{G}$.
At this point, we can note that our matrix representation

$\widetilde{\mathbf{U}}_K$ is slightly different from the representation used for kernel EDMD in [31], which is given by

$$\mathbf{G}^\dagger \mathbf{A} = (\mathbf{P}_\mathbf{x}^\mathbf{T})^\dagger \mathbf{P}_\mathbf{x}^\dagger \mathbf{P}_\mathbf{y} \mathbf{P}_\mathbf{x}^\mathbf{T} \neq \widetilde{\mathbf{U}}_K.$$

$\diamond$

Finally, similarly to (17), we compute the $K \times K$ matrix

$$\widetilde{\mathbf{L}}_\mathbf{data} = \frac{1}{T_s} \log(\mathbf{P}_\mathbf{y}\, \mathbf{P}_\mathbf{x}^\dagger). \qquad (28)$$

3) Third step - identification of the vector field: Using a similar idea as the one explained in Remark 5, we can directly identify the vector field at the different values $\mathbf{x}_k$ and the coefficients w_j^k are then obtained by solving n separate regression problems.

Computation of the vector field $\mathbf{F}(\mathbf{x}_k)$: We assume that $\widetilde{\mathbf{L}}_\mathbf{data}$ is an approximation of the matrix representation of L in the sample space and we have

$$\begin{pmatrix} \mathbf{F}(\mathbf{x}_1) \cdot \nabla f(\mathbf{x}_1) \\ \vdots \\ \mathbf{F}(\mathbf{x}_K) \cdot \nabla f(\mathbf{x}_K) \end{pmatrix} = \begin{pmatrix} Lf(\mathbf{x}_1) \\ \vdots \\ Lf(\mathbf{x}_K) \end{pmatrix} \approx \widetilde{\mathbf{L}}_\mathbf{data} \begin{pmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_K) \end{pmatrix}.$$

Considering the above equality with the identity function $\mathbf{f}(\mathbf{x}) = \mathbf{x}$, we obtain an approximation $\hat{\mathbf{F}}$ of the vector field that is given by

$$\begin{pmatrix} \hat{\mathbf{F}}(\mathbf{x}_1)^T \\ \vdots \\ \hat{\mathbf{F}}(\mathbf{x}_K)^T \end{pmatrix} \approx \widetilde{\mathbf{L}}_\mathbf{data} \begin{pmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_K^T \end{pmatrix}. \qquad (29)$$

The choice of the functions f used to obtain (29) is arbitrary. However, considering monomials of degree one is natural and choosing more functions would yield an over-constrained problem which does not necessarily improve the accuracy of the result. Note also that an approach more similar to the main method is to compute (an approximation of) the matrix representation of $L = \mathbf{F} \cdot \nabla$ in the sampling space and compare it with $\widetilde{\mathbf{L}}_\mathbf{data}$. However, this does not yield better results.

Computation of the coefficients w_k^j: When the value of the vector field is known at every data points, we can find an estimation $\hat{w}_k^j$ of the coefficients w_k^j by solving a regression problem. This problem is decoupled: for each $j = 1, \ldots, n$, we have to solve

$$\hat{F}_j(\mathbf{x}_k) = \sum_{l=1}^{N_F} \hat{w}_j^l\, h_l(\mathbf{x}_k) \qquad k = 1, \ldots, K,$$

which takes the form

$$\begin{pmatrix} \hat{F}_j(\mathbf{x}_1) \\ \vdots \\ \hat{F}_j(\mathbf{x}_K) \end{pmatrix} = \mathbf{H}_\mathbf{x} \begin{pmatrix} \hat{w}_1^j \\ \vdots \\ \hat{w}_{N_F}^j \end{pmatrix} \qquad (30)$$

with

$$\mathbf{H}_\mathbf{x} = \begin{pmatrix} \mathbf{h}(\mathbf{x}_1)^T \\ \vdots \\ \mathbf{h}(\mathbf{x}_K)^T \end{pmatrix} \qquad (31)$$

---

[3]This would correspond exactly to DMD if the basis functions g_j were replaced by functions $g_j(x) = \varphi^{(j-1)t_s}(x)$.

and where $\mathbf{h}$ is the vector of library functions h_k of the vector field. Since we do not make any assumption on the vector field, which might not be polynomial or even analytic, the library functions are not necessarily monomials.

Since we can reasonably assume that most coefficients are zero, we can promote sparsity of the vector of coefficients $\hat{w}_j^l$ by adding a penalty term, which yields the Lasso optimization problem [43]

$$\min_{\mathbf{w}\in\mathbb{R}^{N_F}} \left\| \mathbf{H_x w} - \begin{pmatrix} \hat{F}_j(\mathbf{x}_1) \\ \vdots \\ \hat{F}_j(\mathbf{x}_K) \end{pmatrix} \right\|_2^2 + \rho\|\mathbf{w}\|_1 \qquad (32)$$

where ρ is a positive regularization parameter. Other techniques could also be used to infer $\mathbf{w}$ from the values of the vector field (see e.g. [11], [10]). More generally, machine learning techniques could also be used to solve the regression problem (30).

B. Algorithm

The dual method is summarized in Algorithm 2.

Algorithm 2 Dual lifting method for nonlinear system identification

---

Input: Snapshot pairs $\{(\mathbf{x}_k, \mathbf{y}_k)\}_{k=1}^K$, $\mathbf{x}_k \in \mathbb{R}^n$; basis functions $\{g_k\}_{k=1}^N$ (with $N \geq K$) ; library functions $\{h_k\}_{k=1}^{N_F}$.
Output: Estimates $\hat{\mathbf{F}}(\mathbf{x}_k)$ and $\hat{w}_k^j$.
1: Construct the $K \times N$ matrices $\mathbf{P_x}$ and $\mathbf{P_y}$ defined in (25)
2: Compute the $K \times K$ matrix $\widetilde{\mathbf{L}}_{\mathbf{data}}$ defined in (28)
3: Obtain $\hat{\mathbf{F}}(\mathbf{x}_k)$ with (29)
4: Construct the $K \times N_F$ matrices $\mathbf{H_x}$ defined in (31)
5: For each j, solve the regression problem (30), e.g. solve the Lasso problem (32), to obtain $\hat{w}_k^j$

C. Theoretical results

We now show the convergence of Algorithm 2 in optimal conditions. Let $X \subset \mathbb{R}^n$ be a compact set and $\mathcal{F} = C(X)$ (where $\|\cdot\|$ is the L^∞ norm). It is easy to verify that the Koopman semigroup $U^t : \mathcal{F} \to \mathcal{F}$ is strongly continuous. Assuming that the data points $\mathbf{x}_k \in X$ are uniformly randomly distributed, we consider the linear functionals $\xi_k : f \mapsto \xi_k(f) = f(\mathbf{x}_k)$ which span a subspace $\widetilde{\mathcal{F}}_K^*$ of the dual space $\mathcal{F}^*$ of $\mathcal{F}$. We can define the discrete projection operator $\widetilde{P}_K^* : \mathcal{F}^* \to \widetilde{\mathcal{F}}_K^*$ by

$$\widetilde{P}_K^* \xi = \operatorname*{argmin}_{\tilde{\xi} \in \mathrm{span}\{\xi_1, \ldots, \xi_K\}} \sum_{l=1}^N |\xi(g_l) - \tilde{\xi}(g_l)|^2. \qquad (33)$$

The adjoint (projection) operator $\widetilde{P}_K : \mathcal{F} \to \mathcal{F}$ of $\widetilde{P}_K^*$ (i.e. $\widetilde{P}_K^* \xi(f) = \xi(\widetilde{P}_K f)$ for all $f \in \mathcal{F}$, $\xi \in \mathcal{F}^*$) satisfies

$$(\widetilde{P}_K f)(\mathbf{x}_k) = \xi_k(\widetilde{P}_K f) = \widetilde{P}_K^* \xi_k(f) = \xi_k(f) = f(\mathbf{x}_k).$$

Moreover, for all $\xi \neq 0$ such that $\xi(g_l) = 0$ $\forall l$, we have $\widetilde{P}_K^* \xi = 0$ and equivalently $\xi(\widetilde{P}_K f) = 0$ for all $f \in \mathcal{F}$ so that $\widetilde{P}_K f \in \mathrm{span}\{g_l\}_{l=1}^N$. It follows that $\widetilde{P}_K : \mathcal{F} \to \widetilde{\mathcal{F}}_K$ is a projection onto a subspace $\widetilde{\mathcal{F}}_K \subseteq \mathrm{span}\{g_l\}_{l=1}^N$ that is obtained through interpolation with collocation points $\mathbf{x}_k$. Finally the matrix $\widetilde{\mathbf{U}}_K$ can be interpreted as the matrix representation of the finite-rank operator $\widetilde{U}_K^* = \widetilde{P}_K^* (U^{T_s})^* |_{\widetilde{\mathcal{F}}_K^*} : \widetilde{\mathcal{F}}_K^* \to \widetilde{\mathcal{F}}_K^*$ in the sense that $\widetilde{U}_K^* \xi = \mathbf{c}^T \widetilde{\mathbf{U}}_K \Xi$ for all $\xi = \mathbf{c}^T \Xi \in \widetilde{\mathcal{F}}_K^*$ (with $\Xi = (\xi_1 \cdots \xi_K)^T$). It follows that we have

$$\begin{pmatrix} U^{T_s} f(\mathbf{x}_1) \\ \vdots \\ U^{T_s} f(\mathbf{x}_K) \end{pmatrix} \approx \begin{pmatrix} \widetilde{U}_K^* \xi_1(f) \\ \vdots \\ \widetilde{U}_K^* \xi_K(f) \end{pmatrix} = \widetilde{\mathbf{U}}_K \begin{pmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_K) \end{pmatrix}$$

and we recover (27). Equivalently, the matrix $\widetilde{\mathbf{U}}_K$ can be seen as the representation of $\widetilde{P}_K U^{T_s} |_{\widetilde{\mathcal{F}}_K} : \widetilde{\mathcal{F}}_K \to \widetilde{\mathcal{F}}_K$ with the basis functions $f_k \in \widetilde{\mathcal{F}}_K$ such that $f_k(\mathbf{x}_j) = \delta_{kj}$.

We will assume that

$$\mu\left\{ \mathbf{x} \in X \mid \sum_{l=1}^\infty c_l g_l(\mathbf{x}) = 0 \right\} = 0 \quad \forall(c_1, c_2, \ldots) \neq \mathbf{0}, \tag{34}$$

where μ denotes the Lebesgue measure. This independence condition (see also [36]) ensures that the projections $\widetilde{P}_K^*$ and $\widetilde{P}_K$ are well-defined with probability one as $K \to \infty$.

The following result shows that Algorithm 2 provides exact estimates $\hat{\mathbf{F}}(\mathbf{x}_k)$ of the vector field in optimal conditions.

Theorem 2. *Assume that the sample points $\mathbf{x}_k$ are uniformly randomly distributed in a compact forward invariant set X, and consider $\mathbf{y}_k = \varphi^{T_s}(\mathbf{x}_k)$ (no measurement noise) where φ^t is a (invertible and nonsingular) flow generated by the dynamics $\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x})$.*

If Algorithm 2 is used with $N = K$ basis functions $g_l \in C^1(X)$ such that (34) holds and the identity function $id_j(\mathbf{x}) = x_j$ is in the span of $\{g_l\}_{l=1}^N$, then

$$\lim_{T_s \to 0} \left| F_j(\mathbf{x}_k) - \hat{F}_j(\mathbf{x}_k) \right| = 0 \quad \forall k \in \{1, \ldots, K\}$$

with probability one.

If Algorithm 2 is used with $N \geq K$ basis functions $g_l \in C^1(X)$ such that (34) holds and $\lim_{K\to\infty} \lim_{N\to\infty} \|\widetilde{P}_K id_j - id_j\|_L = 0$ for all $j \in \{1, \ldots, n\}$ (with the graph norm $\|f\|_L = \|f\| + \|Lf\|$), then

$$\lim_{K\to\infty} \lim_{N\to\infty} \lim_{T_s\to 0} \left| F_j(\mathbf{x}_k) - \hat{F}_j(\mathbf{x}_k) \right| = 0 \quad \forall k \in \{1, \ldots, \infty\}$$

with probability one.

Proof. It follows from (34) that the discrete projection $\widetilde{P}_K$ is well-defined (with probability one). For a finite integer K, consider the finite-dimensional operators $\widetilde{A}_K^{T_s} = e^{\widetilde{P}_K L \widetilde{P}_K T_s} : \widetilde{\mathcal{F}}_K \to \widetilde{\mathcal{F}}_K$ and $\widetilde{U}_K^{T_s} = \widetilde{P}_K U^{T_s} |_{\widetilde{\mathcal{F}}_K}$. We have

$$\lim_{T_s\to 0} \frac{\|\log \widetilde{A}_K^{T_s} - (\widetilde{A}_K^{T_s} - I)\|}{T_s} = 0$$

$$\lim_{T_s\to 0} \frac{\|\log \widetilde{U}_K^{T_s} - (\widetilde{U}_K^{T_s} - I)\|}{T_s} = 0$$

and, for all $f \in \widetilde{\mathcal{F}}_K$,

$$\lim_{T_s \to 0} \frac{1}{T_s} \left\| (\widetilde{A}_K^{T_s} - \widetilde{U}_K^{T_s}) f \right\|$$

$$= \lim_{T_s \to 0} \frac{1}{T_s} \int_0^{T_s} \left\| \frac{d}{d\tau} \left(\widetilde{A}_K^\tau \widetilde{U}_K^{T_s - \tau} \right) f \right\| d\tau$$

$$= \lim_{T_s \to 0} \frac{1}{T_s} \int_0^{T_s} \left\| \widetilde{A}_K^\tau (\widetilde{P}_K L \widetilde{P}_K - \widetilde{P}_K L) U^{T_s - \tau} f \right\| d\tau = 0$$

for all $f \in \mathcal{F}$ since U^{T_s} is strongly continuous (see the details in the proof of Theorem 1). This implies that

$$\lim_{T_s \to 0} \frac{1}{T_s} \| (\log \widetilde{A}_K^{T_s} - \log \widetilde{U}_K^{T_s}) f \| = 0 \qquad \forall f \in \widetilde{\mathcal{F}}_K$$

and it follows that

$$\lim_{T_s \to 0} \left| \hat{F}_j(\mathbf{x}_k) - L\widetilde{P}_K \mathrm{id}_j(\mathbf{x}_k) \right|$$

$$\leq \lim_{T_s \to 0} \left\| \frac{\log \widetilde{U}_K^{T_s} \widetilde{P}_K \mathrm{id}_j}{T_s} - \widetilde{P}_K L \widetilde{P}_K \mathrm{id}_j \right\| \qquad (35)$$

$$= \lim_{T_s \to 0} \frac{1}{T_s} \left\| \log \widetilde{U}_K^{T_s} \widetilde{P}_K \mathrm{id}_j - \log \widetilde{A}_K^{T_s} \widetilde{P}_K \mathrm{id}_j \right\| = 0$$

where id_j is the identity function (with $\mathrm{id}_j(\mathbf{x}) = x_j$) and where we used $\widetilde{P}_K f(\mathbf{x}_k) = f(\mathbf{x}_k)$. If $K = N$ and since the vectors $(\xi_1(g_l) \cdots \xi_K(g_l))$ are linearly independent with probability one (this follows from (34), see also [36]), (33) implies that $\widetilde{P}_K^* \xi(g_l) = \xi(g_l)$ for all l, or equivalently $\xi(P_K g_l) = \xi(g_l)$ for all $\xi \in \mathcal{F}^*$, so that $\widetilde{P}_K g_l = g_l$ by the Hahn-Banach theorem. If $\mathrm{id}_j \in \mathrm{span}\{g_l\}_{l=1}^N$, then $\widetilde{P}_K \mathrm{id}_j = \mathrm{id}_j$ and it follows from (35) that

$$\lim_{T_s \to 0} \left| \hat{F}_j(\mathbf{x}_k) - F_j(\mathbf{x}_k) \right| = \lim_{T_s \to 0} \left| \hat{F}_j(\mathbf{x}_k) - L\widetilde{P}_K \mathrm{id}_j(\mathbf{x}_k) \right| = 0 .$$

If $\lim_{K \to \infty} \lim_{N \to \infty} \| \widetilde{P}_K \mathrm{id}_j - \mathrm{id}_j \|_L = 0$, we have

$$\lim_{K \to \infty} \lim_{N \to \infty} \lim_{T_s \to 0} \left| \hat{F}_j(\mathbf{x}_k) - F_j(\mathbf{x}_k) \right|$$

$$= \lim_{K \to \infty} \lim_{N \to \infty} \lim_{T_s \to 0} \left| \left(\hat{F}_j(\mathbf{x}_k) - L\widetilde{P}_K \mathrm{id}_j(\mathbf{x}_k) \right) \right.$$

$$\left. + \left(L\widetilde{P}_K \mathrm{id}_j(\mathbf{x}_k) - L\mathrm{id}_j(\mathbf{x}_k) \right) \right|$$

$$\leq \lim_{K \to \infty} \lim_{N \to \infty} \lim_{T_s \to 0} \left| \hat{F}_j(\mathbf{x}_k) - L\widetilde{P}_K \mathrm{id}_j(\mathbf{x}_k) \right|$$

$$+ \lim_{K \to \infty} \lim_{N \to \infty} \lim_{T_s \to 0} \left\| L\widetilde{P}_K \mathrm{id}_j - L\mathrm{id}_j \right\| = 0$$

since $\| \cdot \|$ denotes the L^∞ norm. This concludes the proof. $\square$

For finite values K and N, Theorem 2 proves the convergence of Algorithm 2 as the sampling time goes to zero, provided that the identity function is contained in the span of test functions g_l (case 1). In practice, we will use Gaussian radial basis functions with $K = N$. In this case, $\widetilde{P}_K \mathrm{id}_j$ is obtained through the interpolation of id_j on a set of Gaussian radial basis functions and it can be shown that the convergence of $\widetilde{P}_K \mathrm{id}_j$ to id_j is uniform with all the derivatives (case 2) (see e.g. [44]). This basis also satisfies the independence condition (34) (see e.g. [36]).

Table I summarizes the main differences between the two frameworks (main and dual methods).

VI. Illustrative examples

The goal of this section is to provide several examples to illustrate the two methods, including some extensions of the main method. We do not provide here an extensive study of the performance with respect to the choice of basis functions and parameters, considering that this is out of the scope of the present paper.

We consider simulated data and, unless otherwise stated, we add a Gaussian state-dependent measurement noise with zero mean and standard deviation $\sigma_{meas} = 0.01$ (see (5)).

A. Main method

We use the lifting method described in Section III, with the parameters $m = m_F = 3$. We consider three systems that exhibit different types of behaviors.

1) Van der Pol oscillator: the dynamics are given by

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= (1 - x_1^2)x_2 - x_1 \end{aligned}$$

and possess a stable limit cycle.

2) Unstable equilibrium: the dynamics are given by

$$\begin{aligned} \dot{x}_1 &= 3\,x_1 + 0.5\,x_2 - x_1 x_2 + x_2^2 + 2\,x_1^3 \\ \dot{x}_2 &= 0.5\,x_1 + 4\,x_2 \end{aligned}$$

and are characterized by an unstable equilibrium at the origin.

3) Chaotic Lorenz system: the dynamics are given by

$$\begin{aligned} \dot{x}_1 &= 10(x_2 - x_1) \\ \dot{x}_2 &= x_1(28 - x_3) - x_2 \\ \dot{x}_3 &= x_1 x_2 - 8/3\,x_3 \end{aligned}$$

and exhibit a chaotic behavior.

A set of K data pairs is generated by taking snapshots at times $\{0, T_s, \ldots, K/rT_s\}$ from r trajectories of these systems. For the first two systems, we consider a setting that is not well-suited to a direct estimation of the derivatives: the sampling period T_s is (reasonably) large and only two or three data points are taken on each trajectory. The identification of the third system, however, requires a smaller sampling period and a larger number of samples. Parameters used to generate the data sets are summarized in the left part of Table II.

For each model, Algorithm 1 yields the estimates $\hat{w}_k^j$ of the coefficients w_k^j (Figure 2). We compute the root mean square error

$$\mathrm{RMSE} = \sqrt{\frac{1}{nN_F} \sum_{j=1}^{n} \sum_{k=1}^{N_F} \left((w_k^j) - (\hat{w}_k^j) \right)^2} \qquad (36)$$

and the normalized root mean square error $\mathrm{NRMSE} = \mathrm{RMSE}/\overline{w}$, where $\overline{w}$ is the average value of the nonzero coefficients $|w_k^j|$. The RMSE and NRMSE values averaged over 50 experiments are small (Table II) and show that the lifting method achieves good performance to identify

Table I
COMPARISON BETWEEN THE MAIN METHOD AND THE DUAL METHOD.

	Main method	Dual method
Constraints on data	$K \geq N$	$K \leq N$
Subspace of functions	Space of observables $\mathcal{F}_N$	"Sample space" $\widetilde{\mathcal{F}}_K^*$
Basis functions	Monomials (i.e. $g_k = p_k$)	Preferably Gaussian RBF g_k
Projected Koopman operator	$U_N f = P_N U^{T_s} f \ \forall f \in \mathcal{F}_N$	$\widetilde{U}_K^* f = \widetilde{P}_K^* (U^{T_s})^* \xi \ \forall \xi \in \widetilde{\mathcal{F}}_K^*$
Matrix representation	$\overline{\mathbf{U}}_N = \mathbf{P}_\mathbf{x}^\dagger \mathbf{P}_\mathbf{y} \in \mathbb{R}^{N \times N}$	$\widetilde{\mathbf{U}}_K = \mathbf{P}_\mathbf{y} \mathbf{P}_\mathbf{x}^\dagger \in \mathbb{R}^{K \times K}$

each system with a fairly low number of samples. Figure 3 shows predictions obtained with the identified vector field. These predictions are good, but some errors (in particular for the chaotic system) are due to measurement noise and finite-dimensional approximations. These results could be improved by increasing the number of basis functions and reducing the sampling period (not shown here). Note that this is for illustrative purposes only, since predictions could also be obtained directly through the lifted dynamics (see e.g. [25]).
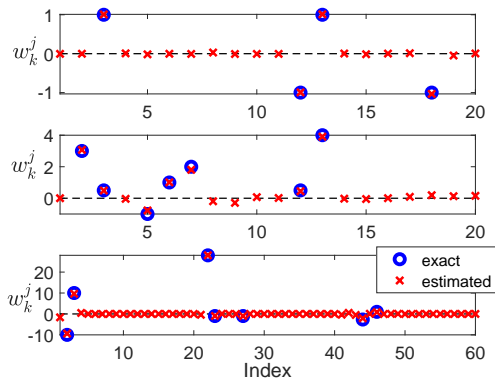

Figure 2. Vector field coefficients obtained for the Van der Pol oscillator (top), the unstable system (middle), and the chaotic Lorenz system (bottom).
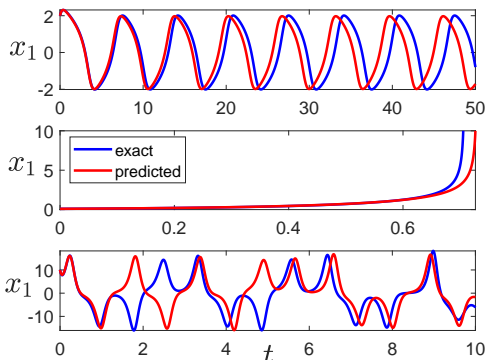

Figure 3. Prediction of trajectories using the estimated vector field (Van der Pol oscillator (top), unstable system (middle), and chaotic Lorenz system (bottom).

For the three systems described above, we also consider the effect of the sampling period T_s on the performance of the method (Figure 4). In the noiseless case, the

NRMSE decreases (exponentially) as the sampling period decreases. This is in agreement with the fact that the NRMSE tends to zero as $T_s \to 0$ (Theorem 1). With measurement noise, this is not the case since the method is biased. In this case, small values of the sampling period make the method more sensitive to noise, so that the minimal (nonzero) value of the NRMSE is obtained with an intermediate value of the sampling period.
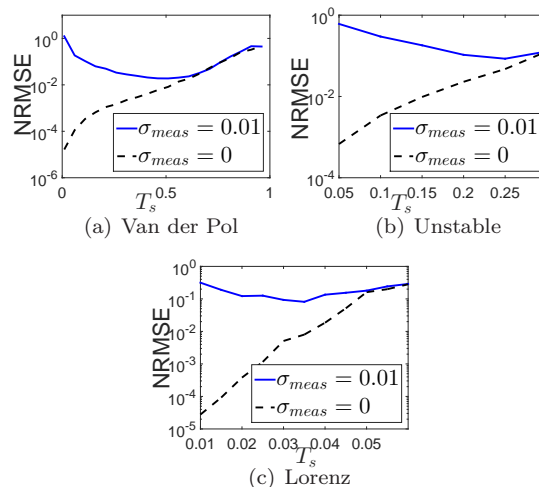

Figure 4. Effect of the sampling period on the normalized root mean square error (averaged over 50 experiments). Parameters are given in Table II.

Next, the approximation of the vector field obtained with (20) is compared with the approximation obtained directly from data through (central) finite differences, i.e.

$$\hat{\mathbf{F}}(\mathbf{x}_k) = \frac{\mathbf{x}_{k+1} - \mathbf{x}_{k-1}}{2T_s} .$$

We consider the three systems and compute the normalized root mean square error on the vector field

$$\mathrm{NRMSE}_F = \frac{\sqrt{\frac{1}{K-1} \sum_{j=1}^n \sum_{k=2}^K \left\| \hat{\mathbf{F}}(\mathbf{x}_k) - \mathbf{F}(\mathbf{x}_k) \right\|^2}}{\frac{1}{K-1} \sum_{k=2}^K \left\| \mathbf{F}(\mathbf{x}_k) \right\|}$$

averaged over 10 experiments, for different values of the sampling period. The results are shown in Figure 5. For each system, we observe that the approximation obtained with the lifting method provides an estimate with an acceptable error (e.g. $NRMSE_F < 0.1$) for larger values of the sampling period than the direct finite difference method. This approximation is also characterized by a clear transition at a critical value of the sampling period,

Table II
FEATURES OF THE DATA SETS AND (NORMALIZED) ROOT MEAN SQUARE ERROR AVERAGED OVER 50 SIMULATIONS.

	Sampling period (T_s)	Total number of data pairs (K)	Number of trajectories (r)	Initial conditions	RMSE	NRMSE
1. Van der Pol	0.5	30	15	$[-1, 1]^2$	0.023	0.023
2. Unstable	0.2	20	20	$[-0.5, 0.5]^2$	0.150	0.087
3. Lorenz	0.033	300	20	$[-20, 20]^3$	0.451	0.059

above which the NRMSE sharply increases (not observed with the unstable system, for which the critical value is beyond the maximal integration time). These results demonstrate the need of considering an indirect method to estimate the vector field (and therefore identify the system) when the sampling period is large.
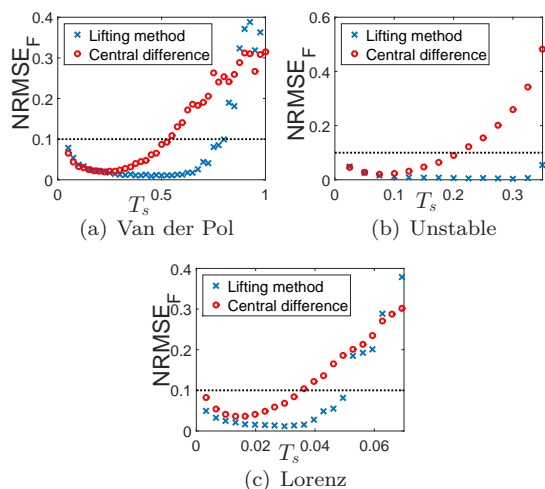
Figure 5. Comparison of the normalized root mean square error on the vector field estimated with the lifting method and with a finite difference method (averaged over 10 experiments). The parameters are the same as in Table II, except for the unstable system where $K = 40$ (2 data pairs on each trajectory) and the initial conditions are in the set $[-0.1, 0.1]^2$.

B. Extensions

We now illustrate several extensions of the lifting method mentioned in Section IV: systems with inputs, process noise, and non-polynomial vector fields.

1) Input and process noise: We consider the forced Duffing system

$$\dot{x}_1 = x_2 \tag{37}$$
$$\dot{x}_2 = x_1 - x_1^3 - 0.2\,x_2 + 0.2\,x_1^2\cos(t) \tag{38}$$

and generate $K = 250$ snapshot data pairs from 5 trajectories (50 on each), with initial conditions on $[-1, 1]^2$. The lifting method provides a good estimation of the vector field (including the forcing term $0.2\,x_1^2\cos(t)$). The RMSE (see Equation (36)) and NRMSE computed over all coefficients (including those related to the forcing term) are given in Table III for different values of the sampling period. Note that we use again the parameters $m = m_F = 3$.

Table III
(NORMALIZED) ROOT MEAN SQUARE ERROR (AVERAGED OVER 50 EXPERIMENTS) RELATED TO THE IDENTIFICATION OF THE FORCED DUFFING SYSTEM.

Sampling period (T_s)	RMSE	NRMSE
0.2	0.032	0.046
0.4	0.031	0.045
0.6	0.057	0.084

Now, we replace the forcing term in (38) by the white noise $\eta(t)$ with different values of the standard deviation σ_{proc} (note that we still add measurement noise with $\sigma_{meas} = 0.01$). We generate $K = 500$ snapshot data pairs from 10 trajectories computed with the Euler-Maruyama scheme, with initial conditions on $[-1, 1]^2$. The sampling period is equal to $T_s = 0.2$. As shown in Table IV, the error is small even with strong process noise, suggesting that the method is robust against process noise.

Table IV
(NORMALIZED) ROOT MEAN SQUARE ERROR (AVERAGED OVER 10 EXPERIMENTS) RELATED TO THE IDENTIFICATION OF THE DUFFING SYSTEM WITH PROCESS NOISE.

Noise strength (σ_{proc})	RMSE	NRMSE
0.2	0.063	0.079
0.4	0.065	0.082
0.6	0.074	0.092
0.8	0.067	0.084
0.1	0.094	0.117

2) Non polynomial vector fields: In this example, we consider a genetic toggle switch (see e.g. [45])

$$\dot{x}_1 = -x_1 + 2\,x_2$$
$$\dot{x}_2 = -x_2 + \frac{2}{1 + x_3^2}$$
$$\dot{x}_3 = -2\,x_3 + 2\,x_4$$
$$\dot{x}_4 = -2\,x_4 + \frac{1}{1 + x_1^3}$$

and we generate $K = 50$ snapshot data pairs from 50 trajectories, with initial conditions on $[0, 1]^4$. The sampling period is $T_s = 0.1$. Since the vector field is not polynomial, we use the extension presented in Section IV-C. The basis functions are the 5 monomials of total degree 0 and 1 (i.e. $m = 1$), to which we add 12 Hill functions

$$\frac{1}{1 + x_k^l} \quad k = \{1, 2, 3, 4\}, \quad l = \{1, 2, 3\}. \tag{39}$$

When there is no measurement noise, all coefficients (including those related to non-polynomial terms) are inferred correctly and we obtain a NRMSE equal to 0.008 (averaged over 50 experiments). However, the results are sensitive to noise in this case. With a measurement noise with $\sigma_{meas} = 0.001$, the NRMSE increases to 0.494. As shown below, the dual method is more robust to noise in this case.

C. Dual method

We illustrate the dual method in the case of a non-polynomial vector field. The main interest of the method, however, is its use with high-dimensional data sets, where the number of basis functions N is (much) larger than the number of sample points K. This will be illustrated in the next section.

The dual method requires to solve a regression problem. When $K < N_F$, we solve the (underconstrained) Lasso problem (32) with the MATLAB toolbox "yall1" [46], [47] (L1-L2 problem, with the parameter $\rho = 0.01$). When $K \geq N_F$, we solve the (overconstrained) problem (32) with the MATLAB function "lasso" (with the parameter $\lambda = 1/K$). Note that the value of the regularization parameter might not be optimal in all cases, and we did not extensively study its effect on the performance of the algorithm. In the following, we only use Gaussian radial basis functions with $\gamma = 0.1$ or $\gamma = 0.01$. Numerical simulations performed with monomial bases (not shown here) yield similar results for small dimensions, but are less accurate and more computationally expensive for large dimensions.

We consider the toggle switch system introduced in Section VI-B. Sample points are generated in the same conditions (i.e. $K = 50$, $T_s = 0.1$). We consider Gaussian radial basis functions with $\gamma = 0.1$ and 17 library functions (5 monomials of total degree 0 and 1, and 12 Hill functions (39)). With no noise, the NRMSE (averaged over 50 experiments) is equal to 0.064, which is worse than with the main method (Section VI-B). However, we obtain a NRMSE equal 0.117 with $\sigma_{meas} = 0.001$ and equal to 0.637 with $\sigma_{meas} = 0.01$. This shows that, in this case, the dual method is more robust to measurement noise than the main method. This might be due to the sparsity constraint that we impose in the dual method.

D. Application to network identification

In the context of dynamical systems, each state can be seen as the node of a network. Moreover, a link can be drawn from node i to node j if the dynamics of the state x_j depends on the state x_i. Under the assumption that the vector field is of the form (2), there is a link from node i to node j if there is at least one nonzero coefficient w_k^j such that the corresponding library function h_k depends on x_i.

Network reconstruction aims at predicting links between states from data, a goal which is equivalent to finding nonzero coefficients w_k^j in our setting. We will consider

that estimated coefficients $\hat{w}_k^j$ with a small absolute value are mainly due to measurement noise and have an exact value w_k^j equal to zero. Hence, we decide that a link is present in the network only if the related value $|w_k^j|$ is above a given threshold. To evaluate the performance of the method, one can compute the true positive rate (i.e. number of correctly identified links divided by the actual number of links) and the false positive rate (i.e. number of incorrectly identified links divided by the actual number of missing links). Varying the threshold value, we can plot the true positive rate against the false positive rate, which corresponds to the receiver operating characteristic (ROC) curve. If the area under the ROC curve (AUROC) is close to one, the network inference method provides good results (bad result correspond to a value close to 0.5).

1) Kuramoto oscillators: We consider a network of n Kuramoto phase oscillators

$$\theta_i = \omega_i + \frac{C}{n} \sum_{j=1}^{n} a_{ij} \sin(\theta_j - \theta_i) \qquad i = 1, \ldots, n$$

with $\theta_i \in [0, 2\pi)$. The coupling strength is set to $C = 10$ and the natural frequencies ω_i are uniformly randomly distributed on $[0, 0.1]$. The values a_{ij} are the entries of the weighted adjacency matrix of a random Erdős-Rényi graph (with a probability $p_{link} = 0.3$ for any two nodes to be connected). The link weights are uniformly randomly distributed on $[0, 1]$.

For two networks ($n = 20$ and $n = 100$), we generate K sample pairs from $K/5$ trajectories (5 data pairs on each trajectory), with $T_s = 0.2$. Initial conditions are uniformly distributed on $[0, 2\pi)^n$. Note that we do not consider data points on $[0, 2\pi)$ but on the real line $\mathbb{R}$ (i.e. without the modulo operation) where there is no discontinuity between 0 and 2π. We use the dual method with Gaussian radial basis functions (with $\gamma = 0.1$) and with $N_F = n$ library functions

$$\{1, \sin(\theta_1 - \theta_i), \ldots, \sin(\theta_{i-1} - \theta_i),$$
$$\sin(\theta_{i+1} - \theta_i), \ldots, \sin(\theta_n - \theta_i)\}$$

for the ith component of the vector field. ROC curves are shown in Figure 6 and the results are summarized in Table V, for different values of K and σ_{meas}. They show that the dual method achieves good performance to reconstruct the whole network. In particular, with high threshold values, one can infer many true positive links with no false positive links.

Table V
RESULTS OBTAINED WITH THE DUAL METHOD FOR THE
RECONSTRUCTION OF A NETWORK OF KURAMOTO OSCILLATORS.

n	K	σ_{meas}	AUROC
20	250	0.01	0.95
100	500	0.001	0.96
100	2000	0.01	0.83

2) Network with nonlinear couplings: We consider a network where each state is directly influenced by other states through n_{inter} quadratic and cubic nonlinearities.
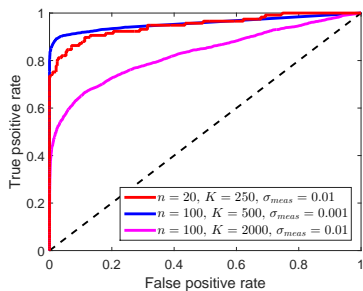
Figure 6. ROC curves obtained with the dual method for the reconstruction of a network of Kuramoto oscillators.

Each nonlinear interaction depends on at most two states. The dynamics of the system are given by

$$\dot{x}_j = w_1^j x_j + \sum_{k=2}^{N_F} w_k^j h_k \qquad j = 1, \dots, n \qquad (40)$$

where the functions h_k are monomials of total degree less or equal to 3. For each j, only n_{inter} coefficients w_k^j are nonzero and associated with monomials of the form $x_k^p x_l^q$, with total degree $p + q \in \{2, 3\}$. The coefficients w_1^j are chosen according to a uniform distribution on $[-1, 0]$ and the coefficients w_k^j, with $k > 1$, are distributed according to a Gaussian distribution of zero mean and standard deviation equal to one. The first term in (40) is a linear term that ensures local stability of the origin. For several network sizes ($n \in \{20, 50, 100\}$), we generate K samples from $K/2$ trajectories (2 data pairs on each trajectory), with $T_s = 0.5$. Initial conditions are uniformly randomly distributed on $[-0.5, 0.5]^n$.

Although we could also consider the main method for small networks (typically $n \leq 20$), we use only the dual method with Gaussian radial basis functions (with $\gamma = 0.01$). The library functions are monomials of total degree less or equal to 3. The method provides an accurate estimation of the vector field and a good reconstruction of the network (Table VI). The ROC curves depicted in Figure 7(a) show that most of half of the links can be inferred with no false positive link (with high threshold values). As shown in Figure 7(b-d), the method is also efficient to infer the nature of the interactions (e.g. quadratic, cubic). Taking advantage of sparsity, it uses not more than 1000 sample points to identify up to 17.10^6 coefficients (most of which are zero). We finally note that, for larger networks, the use of monomials as library functions becomes too demanding in terms of memory. In this case, the dual method can still be used to estimate the value of the vector field at the sample points, but should be combined with other (regression) methods to infer the network.

VII. Conclusion

We have proposed a novel method for nonlinear systems identification. This method relies on a lifting technique developed in an operator-theoretic framework: it aims at identifying the linear Koopman operator in the space

Table VI
Results obtained with the dual method for the reconstruction of a network with quadratic and cubic interactions.

n	n_{inter}	K	AUROC	NRMSE
20	5	200	0.94	0.019
50	15	600	0.87	0.015
100	10	1000	0.91	0.004

(a) ROC curves

(b) $N = 20$, $K = 200$

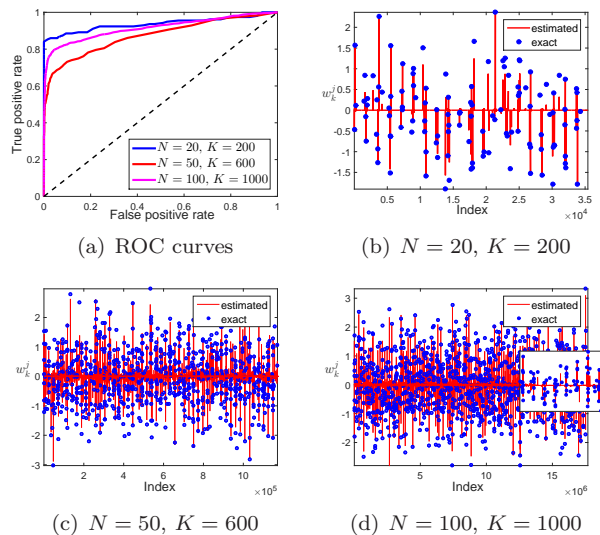(c) $N = 50$, $K = 600$

(d) $N = 100$, $K = 1000$

Figure 7. ROC curves and vector field coefficients obtained with the dual method for the reconstruction of a network with quadratic and cubic interactions. In (d), the inset shows a close-up of some estimated coefficients.

of observables. Key advantages of the method are that numerical schemes rely only on linear techniques and do not require the estimation of state time derivatives. For these reasons, this is a promising alternative to direct identification methods. As shown with several examples, the method is efficient to recover the vector field of several classes of systems, even from small time series with low sampling rate. Moreover, a dual method is also proposed to identify high-dimensional systems and is successfully applied to network reconstruction. Theoretical results also prove the convergence of the two methods in optimal conditions.

The results presented in this paper open the door to further developments and improvements of lifting techniques for nonlinear systems identification, some of which are related to recent advances in Koopman operator theory. For instance, identification lifting techniques with dictionary learning could be developed [48]. Extensions to general vector fields might also be considered, possibly without using library functions. Toward this end, lifting techniques could be combined with other methods: identify unknown parameters with Kalman filtering [49], consider rational functions in the vector field with alternating directions method [50], apply machine learning regression techniques on time derivatives estimated with the dual method, etc. Moreover, we might improve the method robustness to (measurement) noise and provide numerical schemes that

are unbiased and consistent. In this context, Bayesian inference could be considered as a relevant approach. A careful study of the matrix logarithm used in the lifting method could also help to select the good branch (instead of the principal one), a strategy which might improve the performances when the sampling rate is low. Theoretical results could also be obtained to provide bounds on the estimation error. Finally, a potential extension of the proposed approach is to consider the case of unobserved states (e.g. hidden nodes in the context of network identification). In this context, classical linear identification methods could be exploited (e.g. subspace identification methods [51]). In the same line, connections with (nonlinear) system identification methods such as the modulating function approach [52] could be investigated.

Acknowledgments

References

[1] N. Wiener, "Nonlinear problems in random theory," *Nonlinear Problems in Random Theory, by Norbert Wiener, pp. 142. ISBN 0-262-73012-X. Cambridge, Massachusetts, USA: The MIT Press, August 1966.(Paper)*, p. 142, 1966.

[2] I. Leontaritis and S. A. Billings, "Input-output parametric models for non-linear systems part I: deterministic non-linear systems," *International journal of control*, vol. 41, no. 2, pp. 303–328, 1985.

[3] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on neural networks*, vol. 1, no. 1, pp. 4–27, 1990.

[4] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P.-Y. Glorennec, H. Hjalmarsson, and A. Juditsky, "Nonlinear black-box modeling in system identification: a unified overview," *Automatica*, vol. 31, no. 12, pp. 1691–1724, 1995.

[5] R. Haber and H. Unbehauen, "Structure identification of nonlinear dynamic systems—a survey on input/output approaches," *Automatica*, vol. 26, no. 4, pp. 651–677, 1990.

[6] L. Ljung, "System identification," in *Signal analysis and prediction*. Springer, 1998, pp. 163–173.

[7] Y. Bard, *Nonlinear parameter estimation*. Academic press, 1974.

[8] K. J. Åström and P. Eykhoff, "System identification—a survey," *Automatica*, vol. 7, no. 2, pp. 123–162, 1971.

[9] J. M. Varah, "A spline least squares method for numerical parameter estimation in differential equations," *SIAM Journal on Scientific and Statistical Computing*, vol. 3, no. 1, pp. 28–46, 1982.

[10] W. Pan, Y. Yuan, J. Goncalves, and G.-B. Stan, "A sparse Bayesian approach to the identification of nonlinear state-space systems," *IEEE Transactions On Automatic Control*, vol. 61, no. 1, pp. 182–187, 2016.

[11] S. L. Brunton, L. P. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proceedings of the National Academy of Sciences*, vol. 113, no. 15, pp. 3932–3937, 2016.

[12] H. G. Bock, "Recent advances in parameter identification techniques for ode," in *Numerical treatment of inverse problems in differential and integral equations*. Springer, 1983, pp. 95–121.

[13] M. Budišić, R. Mohr, and I. Mezić, "Applied Koopmanism," *Chaos*, vol. 22, no. 4, pp. 047 510–047 510, 2012.

[14] B. O. Koopman, "Hamiltonian systems and transformation in Hilbert space," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 17, no. 5, p. 315, 1931.

[15] I. Mezić, "Spectral properties of dynamical systems, model reduction and decompositions," *Nonlinear Dynamics*, vol. 41, no. 1-3, pp. 309–325, 2005.

[16] A. Mauroy and I. Mezić, "Global stability analysis using the eigenfunctions of the Koopman operator," *IEEE Transactions On Automatic Control*, vol. 61, no. 3, pp. 3356–3369, 2016.

[17] Y. Lan and I. Mezić, "Linearization in the large of nonlinear systems and Koopman operator spectrum," *Physica D*, vol. 242, pp. 42–53, 2013.

[18] A. Mauroy and A. Sootla, "Geometric properties of isostables and basins of attraction of monotone systems," 2017, to appear in IEEE Transactions on Automatic Control.

[19] D. Müller, A. Otto, and G. Radons, "From dynamical systems with time-varying delay via circle maps to Koopmanism," 2017, arXiv preprint arXiv:1701.05136.

[20] H. Arbabi and I. Mezić, "Ergodic theory, Dynamic Mode Decomposition and computation of spectral properties of the Koopman operator," *SIAM Journal on Applied Dynamical Systems*, vol. 16, no. 4, pp. 2096–2126, 2017.

[21] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson, "Spectral analysis of nonlinear flows," *Journal of Fluid Mechanics*, vol. 641, pp. 115–127, 2009.

[22] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *Journal of Fluid Mechanics*, vol. 656, pp. 5–28, 2010.

[23] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz, "On dynamic mode decomposition: Theory and applications," *Journal of Computational Dynamics*, vol. 1, no. 2, pp. 391 – 421, December 2014.

[24] A. Surana and A. Banaszuk, "Linear observer synthesis for nonlinear systems using Koopman operator framework," in *Proceedings of the IFAC conference*, vol. 49, no. 18. Elsevier, 2016, pp. 716–723.

[25] M. Korda and I. Mezić, "Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control," *Automatica*, vol. 93, pp. 149–160, 2018.

[26] E. Kaiser, J. N. Kutz, and S. L. Brunton, "Data-driven discovery of Koopman eigenfunctions for control," *arXiv preprint arXiv:1707.01146*, 2017, arXiv preprint arXiv:1707.01146.

[27] Y. Susuki and I. Mezić, "Nonlinear Koopman modes and power system stability assessment without models," *IEEE Transactions On Power Systems*, vol. 29, no. 2, pp. 899–907, March 2014.

[28] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, "A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition," *Journal of Nonlinear Science*, vol. 25, no. 6, pp. 1307–1346, 2015.

[29] A. Mauroy and J. Goncalves, "Linear identification of nonlinear systems: A lifting technique based on the Koopman operator," in *Proceedings of the 55th IEEE Conference on Decision and Control*, 2016, pp. 6500–6505.

[30] A. N. Riseth and J. P. Taylor-King, "Operator fitting for parameter estimation of stochastic differential equations," *arXiv preprint arXiv:1709.05153*, 2017.

[31] M. O. Williams, C. W. Rowley, and I. G. Kevrekidis, "A kernel-based approach to data-driven Koopman spectral analysis," *Journal of Computational Dynamics*, vol. 2, no. 2, pp. 247–265, 2015.

[32] D. Bruder, C. D. Remy, and R. Vasudevan, "Nonlinear system identification of soft robot dynamics using Koopman operator theory," *arXiv preprint arXiv:1810.06637*, 2018.

[33] A. Lasota and M. C. Mackey, *Chaos, Fractals, and Noise: Stochastic aspects of dynamics*. Springer-Verlag, 1994.

[34] N. Kazantzis and C. Kravaris, "Time-discretization of nonlinear control systems via Taylor methods," *Computers & chemical engineering*, vol. 23, no. 6, pp. 763–784, 1999.

[35] K.-J. Engel and R. Nagel, *One-parameter semigroups for linear evolution equations.* Springer Science & Business Media, 1999, vol. 194.

[36] M. Korda and I. Mezić, "On convergence of extended dynamic mode decomposition to the Koopman operator," *Journal of Nonlinear Science*, vol. 28, no. 2, pp. 687–710, 2018.

[37] Z. Yue, J. Thunberg, L. Ljung, and J. Goncalves, "Identification of Sparse Continuous-Time Linear Systems with Low Sampling Rate: Exploring Matrix Logarithms," https://arxiv.org/abs/1605.08590.

[38] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Sparse identification of nonlinear dynamics with control (SINDYc)," in *Proceedings of the IFAC Conference*, vol. 49, no. 18, 2016, pp. 710–715.

[39] L. P. Proctor, S. L. Brunton, and J. N. Kutz, "Generalizing Koopman operator theory to allow for inputs and control," *SIAM Journal on Applied Dynamical Systems*, vol. 17, no. 1, pp. 909–930, 2018.

[40] J. L. Proctor, S. L. Brunton, and J. N. Kutz, "Dynamic mode decomposition with control," *SIAM Journal on Applied Dynamical Systems*, vol. 15, no. 1, pp. 142–161, 2016.

[41] T. Shnitzer, R. Talmon, and J.-J. Slotine, "Manifold learning with contracting observers for data-driven time-series analysis," *IEEE Transactions on Signal Processing*, vol. 65, no. 4, pp. 904–918, 2016.

[42] D. Perrault-Joncas and M. Meila, "Estimating vector fields on manifolds and the embedding of directed graphs," *arXiv preprint arXiv:1406.0013*, 2014.

[43] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[44] G. Ferrari-Trecate and R. Rovatti, "Fuzzy systems with overlapping Gaussian concepts: Approximation properties in Sobolev norms," *Fuzzy Sets and Systems*, vol. 130, no. 2, pp. 137–145, 2002.

[45] T. Gardner, C. R. Cantor, and J. J. Collins, "Construction of a genetic toggle switch in Escherichia coli," *Nature*, vol. 403, pp. 339–342, 2000.

[46] J. Yang and Y. Zhang, "Alternating direction algorithms for L1-problems in compressive sensing," *SIAM Journal on Scientific Computing*, vol. 33, pp. 250–278, 2011.

[47] Y. Zhang, J. Yang, and Y. W., *YALL1: Your ALgorithms for L1*, yall1.blogs.rice.edu, 2011.

[48] Q. Li, F. Dietrich, E. M. Bollt, and I. 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*, vol. 27, p. 103111, 2017.

[49] W. Pan, F. Menolascina, and G.-B. Stan, "Online model selection for synthetic gene networks," in *Proceedings of the 55th IEEE Conference on Decision and Control.* IEEE, 2016, pp. 776–782.

[50] N. M. Mangan, S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Inferring biological networks by sparse identification of nonlinear dynamics," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 2, no. 1, pp. 52–63, 2016.

[51] P. Van Overschee and B. De Moor, *Subspace identification for linear systems: Theory—Implementation—Applications.* Springer Science & Business Media, 2012.

[52] M. Shinbrot, "On the analysis of linear and nonlinear systems," *Trans. ASME*, vol. 79, no. 3, pp. 547–552, 1957.
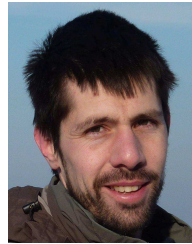
Alexandre Mauroy is an Assistant Professor at the University of Namur, Belgium (Department of Mathematics and Namur Institute for Complex Systems). He received the M.S. degree in aerospace engineering and the Ph.D. degree in applied mathematics from the University of Liège, Belgium, in 2007 and 2011, respectively. Prior to joining UNamur, he was a postdoctoral researcher at the University of California Santa Barbara from 2011 to 2013, at the University of Liège from 2013 to 2015, and at the University of Luxembourg in 2016.

His research interests include synchronization in complex networks, network identification, and applications of operator theoretic methods to dynamical systems and control.

Jorge Goncalves is a Professor at the Luxembourg Centre for Systems Biomedicine, University of Luxembourg and a Principal Research Associate at the Department of Plant Sciences, University of Cambridge. He received his Licenciatura (5-year S.B.) degree from the University of Porto, Portugal, and the M.S. and Ph.D. degrees from the Massachusetts Institute of Technology, Cambridge, MA, all in Electrical Engineering and Computer Science, in 1993, 1995, and 2000, respectively. He then held two postdoctoral positions, first at the Massachusetts Institute of Technology for seven months, and from 2001 to 2004 at the California Institute of Technology with the Control and Dynamical Systems Division. At the Information Engineering Division of the Department of Engineering, University of Cambridge he was a Lecturer from 2004 until 2012, a Reader from 2012 until 2014, and from 2014 until 2019 he was a Principal Research Associate. From 2005 until 2014 he was a Fellow of Pembroke College, University of Cambridge. From June to December 2010 and January to September 2011 he was a visiting Professor at the University of Luxembourg and California Institute of Technology, respectively. Since 2019 he is a Principal Research Associate at the Department of Plant Sciences, University of Cambridge, and since 2013 he is a Professor at the Luxembourg Centre for Systems Biomedicine, University of Luxembourg.