OPTIMAL KERNEL-BASED DYNAMIC MODE DECOMPOSITION

P. HÉAS* AND C. HERZET*

Abstract. The state-of-the-art algorithm known as kernel-based dynamic mode decomposition (K-DMD) provides a sub-optimal solution to the problem of reduced modeling of a dynamical system based on a finite approximation of the Koopman operator. It relies on crude approximations and on restrictive assumptions. The purpose of this work is to propose a kernel-based algorithm solving exactly this low-rank approximation problem in a general setting.

1. Introduction.

Consider a high-dimensional system of the form:

$$\begin{cases} x_t = f_t(x_{t-1}), \\ x_1 = \theta, \end{cases} \tag{1.1}$$

where the x_t's belonging to $\mathbb{R}^p$ constitute a state trajectory, $f_t : \mathbb{R}^p \to \mathbb{R}^p$, and $\theta \in \mathbb{R}^p$ denotes an initial condition. Computing trajectories for many values of the initial condition may be an intractable task for large p. Among reduced modeling methods, dynamic mode decomposition (DMD) [4, 7, 9, 10, 11] is a popular framework for the linear approximation of the trajectories of (1.1). It can be extended to the approximation of non-linear behaviors by finite approximation of the *Koopman operator* using a decomposition known as extended DMD (EDMD) [3, 8, 12, 13]. In this paper, we will focus on the latter approximation.

1.1. Koopman Operator Approximation.

We present in this section reduced models based on the finite approximation of the Koopman operator.

This class of reduced models presuppose the knowledge of a set of $n \geq p$ functions $\psi_i \in \mathcal{L}^2(\mathbb{R}^p)$ forming $\Psi = (\psi_1 \cdots \psi_n)^\intercal \in (\mathcal{L}^2(\mathbb{R}^p))^n$, an n-dimensional vector-valued function.[1] Given this knowledge, reduced models based on an approximation of the Koopman operator determine (according to some optimality criterion specified in Section 1.2) a matrix $\hat{A}_k \in \mathbb{R}^{n\times n}$ of rank at most $k \leq p$, yielding a set of approximations $\{\tilde{x}_t(\theta)\}_t$ of the states $\{x_t(\theta)\}_t$ of (1.1) according to

$$\begin{cases} \tilde{x}_t(\theta) = \Psi^{-1}(\Psi(\tilde{x}_t(\theta))), \\ \Psi(\tilde{x}_t(\theta)) = \hat{A}_k \Psi(\tilde{x}_{t-1}(\theta)), \\ \Psi(\tilde{x}_1(\theta)) = \Psi(\theta). \end{cases} \tag{1.2}$$

We have assumed in (1.2) that there exists an inverse function $\Psi^{-1} \in (\mathcal{L}^2(\mathbb{R}^n))^p$ such that $\Psi^{-1}(\Psi(\tilde{x}_t(\theta))) = \tilde{x}_t(\theta)$. We justify in Appendix A the fact that matrix $\hat{A}_k$ in (1.2) can be seen under a certain condition[2] as a finite approximation of the Koopman operator. The computational gain brought by (1.2) can be seen by considering the eigen-decomposition

$$\hat{A}_k = \Xi \Lambda \Xi^{-1}, \tag{1.3}$$

*INRIA Centre Rennes - Bretagne Atlantique, campus universitaire de Beaulieu, 35042 Rennes, France (patrick.heas@inria.fr, cedric.herzet@inria.fr)

[1] The number of components of this vector-valued function is possibly infinite. However, in order to simplify the presentation, we will all along the paper define formally infinite dimensional vectors using notations for vector spaces of finite dimension n with $n \to \infty$.

[2] The condition is that function Ψ should be chosen so that its image is an invariant vector space of the Koopman operator.

Columns of matrices $(\Xi^{-1})^\intercal = (\xi_1 \cdots \xi_n) \in \mathbb{C}^{n\times n}$ and $\Xi = (\zeta_1 \cdots \zeta_n) \in \mathbb{C}^{n\times n}$ are left and right eigen-vectors associated to the Jordan matrix $\Lambda \in \mathbb{C}^{n\times n}$ of rank at most k [5]. Indeed, using this decomposition, recursion (1.2) can be rewritten as

$$\tilde{x}_t(\theta) = \Psi^{-1}(\Xi\Lambda^{t-1}\Xi^{-1}\Psi(\theta)). \tag{1.4}$$

Assuming that $\hat{A}_k$ is diagonalisable, then $\Lambda = \mathrm{diag}(\lambda_1 \cdots \lambda_n)$ and it is easy to see that (1.4) becomes

$$\begin{cases} \tilde{x}_t(\theta) = \Psi^{-1}\left(\displaystyle\sum_{i=1}^{\mathrm{rank}(\hat{A}_k)} \zeta_i \nu_{i,t}\right), \\ \nu_{i,t} = \lambda_i^{t-1}\tilde{\varphi}_i(\theta), \end{cases} \tag{1.5}$$

with $\tilde{\varphi}_i = \xi_i^\intercal \Psi \in \mathcal{L}^2(\mathbb{R}^p)$, see details on this derivation in Appendix A. Note that if Ψ^{-1} is linear, then the first equation of (1.5) simplifies to

$$\tilde{x}_t(\theta) = \sum_{i=1}^{\mathrm{rank}(\hat{A}_k)} \nu_{i,t}\mu_i, \quad \text{with} \quad \mu_i = \Psi^{-1}\zeta_i \in \mathbb{C}^p. \tag{1.6}$$

The $\tilde{\varphi}_i$'s, the μ_i's and the λ_i's are known respectively as approximations of the i-th Koopman *eigen-function*, *eigen-mode* and *eigen-value* [12]. Note that by bartering the original system (1.1) by the reduced model (1.6), the computational cost necessary to compute a trajectory for a given initial condition θ, which we will refer to as the "on-line complexity", has been lowered to at most $\mathcal{O}(pk)$, as long as we can compute the $\tilde{\varphi}_i(\theta)$'s within a complexity scaling linearly in p. It is also independent of the trajectory length T. If we further assume that the evaluation of Ψ^{-1} does not increase the on-line complexity, the linear scaling with respect to the ambient dimension p is preserved in the reduced model (1.5).

1.2. Data-Driven Construction. We consider a data-driven approach to build "off-line" reduced model (1.5). It presupposes a data set from which we can estimate matrix $\hat{A}_k$. Assume a data set of representative trajectories $\{x_t(\theta_i)\}_{t=1,i=1}^{T,N}$ so-called *snapshots* of the high-dimensional system corresponding to N initial conditions $\{\theta_i\}_{i=1}^N$. Inspired by the low-rank approximations introduced in [4, 7], we consider a generalisation of the problem in [12, 13]: matrix $\hat{A}_k$ targets the solution of the *low-rank EDMD approximation* problem

$$A_k^\star \in \underset{A:\mathrm{rank}(A)\leq k}{\arg\min} \|\boldsymbol{\Psi}_\mathbf{Y} - A\boldsymbol{\Psi}_\mathbf{X}\|_F^2, \tag{1.7}$$

where matrices in $\mathbb{R}^{n\times m}$ with $m = N(T-1)$

$$\boldsymbol{\Psi}_\mathbf{X} = (\Psi_{1:T-1}^1 \cdots \Psi_{1:T-1}^N) \quad \text{and} \quad \boldsymbol{\Psi}_\mathbf{Y} = (\Psi_{2:T}^1 \cdots \Psi_{2:T}^N),$$

are defined as $\Psi_{t_1:t_2}^i = (\Psi(x_{t_1}(\theta_i)) \cdots \Psi(x_{t_2}(\theta_i)))$ and $\|\cdot\|_F$ refers to the Frobenius norm. Let us remark that in its original formulation, matrix $\hat{A}_k$ targeted the solution of the following unconstrained problem[3]

$$\underset{A\in\mathbb{R}^{n\times n}}{\arg\min} \|\boldsymbol{\Psi}_\mathbf{Y} - A\boldsymbol{\Psi}_\mathbf{X}\|_F^2. \tag{1.8}$$

[3]Note that the standard DMD approximation problem is a particularisation of (1.8) in the case Ψ is chosen so that $\Psi(x_t(\theta_i)) = x_t(\theta_i)$ for $i = 1, \cdots, N$ and $t = 1, \cdots, T$ [12].

This is an unconstrained least-square problem. The solution $\mathbf{\Psi_Y}\mathbf{\Psi_X}^\dagger$ of rank m is in this case simply obtained by singular value decomposition (SVD) [11]. The solution $\mathbf{\Psi_Y}\mathbf{\Psi_X}^\dagger$ is also a solution $A_k^\star$ of problem (1.7) in the case $k \geq m$, see [6]. The algorithm proposed [6] solves problem (1.7) for an arbitrary value of k: considering the vectors $\Psi(x_t(\theta_i))$ in place of the vectors $x_t(\theta_i)$, it is straightforward to obtain a closed-form expression of $A_k^\star$. Moreover, the solution is computed with an off-line complexity linear in n.

1.3. Contribution. But the algorithm proposed in [6] to solve (1.7) and to build (1.5) becomes in general intractable in the case $n \gg p$. This configuration is recurrent for many definitions of function Ψ. To circumvent this intractability for large values of n, authors in [13] propose to use the *kernel trick* [2] and derive an algorithm baptised kernel-based DMD (K-DMD), featuring an advantageous off-line complexity linear in p and independent of n. However, as proposed in [13], the K-DMD algorithm computes a sub-optimal solution of (1.7) and builds a crude approximation of (1.5) in most cases. The lack of optimality of the algorithm originates from :

i) the assumption that Ψ^{-1} is linear;
ii) the assumption that matrix $\mathbf{\Psi_X}$ is full-rank;
iii) the assumption that the eigen-mode μ_i belongs to the span of the $x_t(\theta_i)$'s;
iv) the ignorance of the low-rank constraint in problem (1.7).

This work shows how to solve exactly problem (1.7) and how to build an optimal Koopman operator approximation (1.5) using kernel-based computation and eliminating these crude assumptions.

The paper is organised as follows. In the next section, we describe the optimal solution of the low-rank EDMD problem (1.7) introduced in [6]. We also detail the kernel-based algorithm proposed in [13] and point out the assumptions depriving the computed solution of optimality. In Section 3, we then present our exact kernel-based solution. In particular, we show that the eigen-vectors of the problem solution admit a low-dimensional representation and design an algorithm accordingly. We finally evaluate in Section 4 the performance of the proposed algorithm in terms of the reduced-model approximation error and compare to state-of-the-art methods. Conclusions are given in Section 5.

2. Preliminaries. In this section, we present state-of-the-art methods in order to *i)* build exactly reduced model (1.5) with standard techniques manipulating vectors in $\mathbb{R}^n$; *ii)* provide an approximation of this reduced model using the K-DMD algorithm, which manipulates vectors in $\mathbb{R}^p$. We begin by introducing some notations.

2.1. Some Notations. We wil adopt the following notations. I_k will denote the k-dimensional identity matrix. The SVD of a matrix $M \in \mathbb{R}^{p\times q}$ with $p \geq q$ will be noted as $M = U_M \Sigma_M V_M^\intercal$ with $U_M \in \mathbb{R}^{p\times q}$, $V_M \in \mathbb{R}^{q\times q}$ and $\Sigma_M \in \mathbb{R}^{q\times q}$ so that $U_M^\intercal U_M = V_M^\intercal V_M = I_q$ and Σ_M is diagonal. The columns of matrices U_M and V_M will be denoted $U_M = (u_M^1 \cdots u_M^q)$ and $V_M = (v_M^1 \cdots v_M^q)$ while diagonal components of matrix Σ_M will be $\Sigma_M = \mathrm{diag}(\sigma_{M,1}, \cdots, \sigma_{M,q})$ with $\sigma_{M,i} \geq \sigma_{M,i+1}$ for $i = 1, \cdots, q-1$. To keep the presentation as simple as possible, the SVD definition is given for matrices in $\mathbb{R}^{p\times q}$ with $p \geq q$. However, results presented in this work can be extended without any difficulty to the case $p < q$ by using an alternative definition of the SVD. The Moore-Penrose pseudo-inverse of matrix M is then defined as $M^\dagger = V_M \Sigma_M^\dagger U_M^\intercal$, where

$\Sigma_M^\dagger = \mathrm{diag}(\sigma_{M,1}^\dagger, \cdots, \sigma_{M,q}^\dagger)$ with

$$\sigma_{M,i}^\dagger = \begin{cases} \sigma_{M,i}^{-1} & \text{if} \quad \sigma_{M,i} > 0 \\ 0 & \text{otherwise} \end{cases}.$$

The orthogonal projector onto the span of the columns (resp. of the rows) of matrix M will be denoted by $\mathbb{P}_M = MM^\dagger = U_M \Sigma_M \Sigma_M^\dagger U_M^\intercal$ (resp. $\mathbb{P}_{M^\intercal} = M^\dagger M = V_M \Sigma_M^\dagger \Sigma_M V_M^\intercal$) [5]. The element at the i-th row and j-th column of a matrix M will be denoted as $M_{(i,j)}$. We define matrix $\mathbf{X}$ (resp. $\mathbf{Y}$) in $\mathbb{R}^{p\times m}$, whose i-th column is defined as the application of Ψ^{-1} on the i-th column of $\mathbf{\Psi_X}$ (resp. of $\mathbf{\Psi_Y}$). Using short-hand notations, we will write $\mathbf{X} = \Psi^{-1}\mathbf{\Psi_X}$ and $\mathbf{Y} = \Psi^{-1}\mathbf{\Psi_Y}$.

2.2. An Optimal Solution to Low-Rank EDMD [6]. Let the columns of matrix $\hat{P} \in \mathbb{R}^{n\times k}$ be the left singular vectors $\{u_{\mathbf{Z}}^i\}_{i=1}^m$ associated to the k largest singular values of matrix

$$\mathbf{Z} = \mathbf{\Psi_Y} \mathbb{P}_{\mathbf{\Psi_X}^\intercal} \in \mathbb{R}^{n\times m}. \tag{2.1}$$

THEOREM 2.1 ([6]). *Problem* (1.7) *admits the solution*

$$A_k^\star = \hat{P}\hat{P}^\intercal \mathbf{\Psi_Y} \mathbf{\Psi_X}^\dagger, \tag{2.2}$$

and the approximation error is

$$\|\mathbf{\Psi_Y} - A_k^\star \mathbf{\Psi_X}\|_F^2 = \sum_{i=k+1}^{m} \sigma_{\mathbf{Z},i}^2 + \sum_{i=i^*}^{m}\sum_{j=1}^{m} \sigma_{\mathbf{\Psi_Y},j}^2 \left((v_{\mathbf{\Psi_X}}^i)^\intercal v_{\mathbf{\Psi_Y}}^j \right)^2, \tag{2.3}$$

with $i^* = \mathrm{rank}(\mathbf{\Psi_X}) + 1$.

As shown by this therorem, the solution of (1.7) is in fact the orthogonal projection of the unconstrained problem solution $\mathbf{\Psi_Y}\mathbf{\Psi_X}^\dagger$ onto the span of the first k left singular vectors of a matrix $\mathbf{Z}$. This matrix is defined as the multiplication of $\mathbf{\Psi_Y}$ by the projector onto the span of the rows of $\mathbf{\Psi_X}$.

From these theoretical results, it is straightforward to adapt [6, Algorithm 3] in order to compute the parameters of the reduced model (1.5) and the low-rank approximation $\tilde{x}_t(\theta)$. The adaptation simply consists in setting $\mathbf{\Psi_X}$ and $\mathbf{\Psi_Y}$ in place of $\mathbf{X}$ and $\mathbf{Y}$ in the proposed algorithm. We remark that the methods proposed for EDMD in [12] or [13], which do not rely on kernel-based computation, also provide the optimal solution (2.2), but only in the case $k \geq m$ and matrix $\mathbf{\Psi_X}$ is full-rank. The adapted algorithm will be characterised by an off-line and on-line complexity scaling as $\mathcal{O}(m^2(m+n))$ and $\mathcal{O}(kn)$, *i.e.*, linearly with respect to n. Nevertheless, the complexity of any of the algorithms mentionned above is prohibitive in the case $n \gg p$.

2.3. The K-DMD Algorithm [13]. The idea followed in [13] to face the case $n \gg p$ is to derive a kernel-based method to approximate the eigen-decomposition of the solution of (1.7) for $k \geq m$. This method known as K-DMD is detailed in Algorithm 1. It relies on the kernel-trick presented in Appendix B. It therefore only applies for functions $\Psi \in (\mathcal{L}^2(\mathbb{R}^p))^n$, such that there exists a kernel function $h : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ computing the scalar product $h(y,z) = \Psi(y)^\intercal \Psi(z)$ for any $y, z \in \mathbb{R}^p$.

Algorithm 1 : K-DMD [13]

inputs: The $x_t(\theta_i)$'s and some θ

1) Compute $\mathbf{\Psi_X^\intercal \Psi_X}$, $\mathbf{\Psi_Y^\intercal \Psi_Y}$, $\mathbf{\Psi_Y^\intercal \Psi_X}$ and $\mathbf{\Psi_X^\intercal}\Psi(\theta)$ with the kernel trick [2].

2) Get $(V_{\mathbf{\Psi_X}}, \Sigma_{\mathbf{\Psi_X}})$ by eigen-decomposition of $\mathbf{\Psi_X^\intercal \Psi_X}$.

3) Get $(V_{\mathbf{\Psi_Y}}, \Sigma_{\mathbf{\Psi_Y}})$ by eigen-decomposition of $\mathbf{\Psi_Y^\intercal \Psi_Y}$.

4) Compute the eigen-vectors $\{\tilde{\xi}_i\}_{i=1}^m$ and eigen-values $\{\tilde{\lambda}_i\}_{i=1}^m$ of $R\,\mathbf{\Psi_Y^\intercal \Psi_X}\,R^\intercal$, with $R = \Sigma_{\mathbf{\Psi_X}}^\dagger V_{\mathbf{\Psi_X}}^\intercal$.

5) Estimate eigen-functions $\{\tilde{\varphi}_i(\theta)\}_{i=1}^m$ using (2.4).

6) Compute the pseudo-inverse of $\tilde{\Xi}_m^{-1} = (\tilde{\xi}_1 \cdots \tilde{\xi}_m)^\intercal$.

7) Estimate eigen-modes $\{\mu_i\}_{i=1}^m$ using (2.8).

output: $\tilde{x}_t(\theta)$'s using (1.6) with $\nu_{i,t} = \tilde{\lambda}_i^{t-1}\tilde{\varphi}_i(\theta)$.

Analysing the K-DMD algorithm, we see that left eigen-vectors of the solution $A_m^\star = \mathbf{\Psi_Y \Psi_X^\dagger}$, are determined by the eigen-vectors $\{\tilde{\xi}_i\}_{i=1}^m$ and eigen-values $\{\tilde{\lambda}_{\ell,i}\}_{i=1}^m$ of the smaller matrix $R\,\mathbf{\Psi_Y^\intercal \Psi_X}\,R^\intercal$ with $R = \Sigma_{\mathbf{\Psi_X}}^\dagger V_{\mathbf{\Psi_X}}^\intercal$. Indeed, as we will show in Proposition 3.1, the left eigen-vectors of $A_m^\star$ are $\{U_{\mathbf{\Psi_X}}\tilde{\xi}_i\}_{i=1}^m$ in the case $\mathbf{\Psi_X}$ is full rank. Using the fact that $U_{\mathbf{\Psi_X}} = \mathbf{\Psi_X} R^\intercal$, a consequence is that the i-th Koopman eigen-function approximation $\tilde{\varphi}_i(\theta)$ evaluated at any point $\theta \in \mathbb{R}^p$ is

$$\tilde{\varphi}_i(\theta) = \xi_i^\intercal \Psi(\theta) = \tilde{\xi}_i^\intercal R \mathbf{\Psi_X^\intercal}\Psi(\theta). \tag{2.4}$$

Gathering approximations of the form (2.4), where $\Psi(\theta)$ is identified to the different columns of $\mathbf{\Psi_X}$, we obtain

$$\Xi_m^{-1}\mathbf{\Psi_X} = \tilde{\Xi}_m^{-1}\Sigma_{\mathbf{\Psi_X}} V_{\mathbf{\Psi_X}}^\intercal, \tag{2.5}$$

with $(\Xi_m^{-1})^\intercal = (\xi_1 \cdots \xi_m) \in \mathbb{C}^{n\times m}$, $(\tilde{\Xi}_m^{-1})^\intercal = (\tilde{\xi}_1 \cdots \tilde{\xi}_m) \in \mathbb{C}^{m\times m}$. K-DMD features advantageous off-line and on-line complexities scaling as $\mathcal{O}(m^2(m+p))$ and $\mathcal{O}(mp)$.

However, the estimated eigen-modes constitute potentially poor approximations. Indeed, the μ_j's in (1.6) are set as minimisers in $\mathbb{C}^p$ of the square of the ℓ_2-norm of the reconstruction error

$$\|x_{t+1}^i - \sum_{j=1}^m \lambda_j \xi_j^\intercal \Psi(x_t^i)\mu_j\|_2^2, \tag{2.6}$$

for any data pair (x_t^i, x_{t+1}^i) satisfying (1.1). Rewritten in matricial form using (2.5), minimising this error norm leads to a least-square problem,

$$\underset{\mu_1,\cdots,\mu_m}{\arg\min} \|\mathbf{Y} - \begin{pmatrix}\mu_1 & \cdots & \mu_m\end{pmatrix} \mathrm{diag}(\tilde{\lambda}_{\ell,1}\cdots\tilde{\lambda}_{\ell,m})\Xi_m^{-1}\mathbf{\Psi_X}\|_F^2. \tag{2.7}$$

whose solution is the approximation

$$\begin{pmatrix}\mu_1 & \cdots & \mu_m\end{pmatrix} \approx \mathbf{Y}R^\intercal((\tilde{\Xi}_m)^{-1})^\dagger \mathrm{diag}(\tilde{\lambda}_{\ell,1}^\dagger, \cdots, \tilde{\lambda}_{\ell,m}^\dagger). \tag{2.8}$$

Therefore we see that this approximation becomes exact only in the case the μ_j's belong to the span of $\mathbf{Y}$. Furthermore, we note that even in this case, the K-DMD algorithm relies on two important assumptions: *i)* Ψ^{-1} is linear so that the minimisation problem (2.6) is consistent, *ii)* matrix $\mathbf{\Psi_X}$ needs to be full-rank.

3. The Proposed Upgrade. We detail in this section our optimal kernel-based DMD (OK-DMD) algorithm building exactly reduced model (1.5). It relies on the optimal solution $A_k^\star$ of (1.7) given by Theorem 2.1. The proposed algorithm, which presents a complexity linear in p and independent of n, can be applied in a general setting and is relieved from all state-of-the-art limitations. To achieve the design of such an algorithm, we first need to introduce some theoretical material. We begin by showing that the eigen-vectors of the solution $A_k^\star$ admit a low-dimensional representation.

3.1. Eigen-Decomposition of $A_k^\star$. Our kernel-based method relies on the following proposition. Let $\{\xi_i\}_{i=1}^k$ and $\{\zeta_i\}_{i=1}^k$ denote the left and right eigen-vectors of $A_k^\star$ associated to its at most k non-zero eigen-values $\{\lambda_i\}_{i=1}^k$. We assume $A_k^\star$ is diagonalisable. We redefine $\{(\tilde{\xi}_i, \tilde{\lambda}_{\ell,i})\}_{i=1}^k$ and $\{(\tilde{\zeta}_i, \tilde{\lambda}_{r,i})\}_{i=1}^k$ respectively as the k eigen-vectors and eigen-values of the smaller matrices

$$(\tilde{A}_{\ell,k})^\intercal = R\,\mathbf{\Psi}_\mathbf{Y}^\intercal\mathbf{\Psi}_\mathbf{Y}\,S\,\mathbf{\Psi}_\mathbf{Y}^\intercal\mathbf{\Psi}_\mathbf{X}\,R^\intercal \in \mathbb{R}^{m\times m}, \tag{3.1}$$

$$\tilde{A}_{r,k} = C\,\mathbf{\Psi}_\mathbf{Y}^\intercal\mathbf{\Psi}_\mathbf{Y}\,R^\intercal\,R\,\mathbf{\Psi}_\mathbf{X}^\intercal\mathbf{\Psi}_\mathbf{Y}\,C^\intercal \in \mathbb{R}^{m\times m}, \tag{3.2}$$

with matrices S, $C \in \mathbb{R}^{m\times m}$ given by

$$S = \mathbb{P}_{\mathbf{\Psi}_\mathbf{X}^\intercal} V_\mathbf{Z} \mathrm{diag}(\sigma_{\mathbf{Z},1}^{-2}, \cdots, \sigma_{\mathbf{Z},k}^{-2}, 0, \cdots, 0) V_\mathbf{Z}^\intercal \mathbb{P}_{\mathbf{\Psi}_\mathbf{X}^\intercal},$$
$$C = \mathrm{diag}(\sigma_{\mathbf{Z},1}^{-1}, \cdots, \sigma_{\mathbf{Z},k}^{-1}, 0, \cdots, 0) V_\mathbf{Z}^\intercal \mathbb{P}_{\mathbf{\Psi}_\mathbf{X}^\intercal}.$$

Moreover, let the columns of matrix $\hat{P} \in \mathbb{R}^{n\times k}$ be the left singular vectors associated to the k largest singular values of $\mathbf{Z}$ given in (2.1).

PROPOSITION 3.1. *For $i = 1, \cdots, k$, the left and right eigen-vectors of $A_k^\star$ and its eigen-values satisfy*

$$\xi_i = U_{\mathbf{\Psi}_\mathbf{X}} \tilde{\xi}_i \quad \textit{and} \quad \zeta_i = \hat{P}\tilde{\zeta}_i \quad \textit{with} \quad \lambda_i = \tilde{\lambda}_{\ell,i} = \tilde{\lambda}_{r,i}.$$

This proposition gives a closed-form expression to the left eigen-vectors of $A_k^\star$, the solution of (1.7). Its proof detailed in Appendix C. It exploits the factorisation of the closed-form solution $A_k^\star$ given in [6]. Using (2.4), we deduce from Proposition 3.1 a closed-form i-th Koopman eigen-function approximation $\tilde{\varphi}_i(\theta)$ for $i = 1, \cdots, k$ at any point $\theta \in \mathbb{R}^p$. Moreover, this proposition provides an analytical expression for the ζ_i's, the right eigen-vectors of $A_k^\star$ and supply the related eigen-values. Nevertheless, we must be cautious with the normalisation of the eigen-vectors. We verify after some simple algebraic calculus that the condition $\xi_i^\intercal A_k^\star \zeta_i = \lambda_i$ (*i.e.*, the normalisation $\zeta_i^\intercal \xi_i = 1$) is ensured if $\tilde{\zeta}_i$ is rescaled to satisfy $\tilde{\zeta}_i^\intercal E \tilde{\xi}_i = 1$, with

$$E = \Sigma_\mathbf{Z}^\dagger V_\mathbf{Z}^\intercal \mathbb{P}_{\mathbf{\Psi}_\mathbf{X}^\intercal} \mathbb{P}_{\mathbf{\Psi}_\mathbf{Y}^\intercal} \mathbf{\Psi}_\mathbf{Y}^\intercal \mathbf{\Psi}_\mathbf{X} R. \tag{3.3}$$

The elements of the set $\{(\xi_i, \zeta_i, \lambda_i)\}_{i=1}^k$ issued from the eigen-decomposition of $A_k^\star$ correspond to the parameters of the reduced model in (1.5). Relying on Proposition 3.1, the computation of these parameters can be obtained through the eigen-decomposition $\{(\tilde{\xi}_i, \tilde{\zeta}_i, \lambda_i)\}_{i=1}^k$. The latter is tractable (in the sense that the complexity is independent of the dimension n) and can be done within a complexity linear in p by using the kernel-trick.

3.2. Kernel-Based Inversion. Nevertheless, to achieve the design of the sought algorithm of complexity independent of n, it remains to provide a manner to compute $\Psi^{-1}(\zeta_i \nu_{i,t})$ in (1.5) without resorting explicitly to vectors in $\mathbb{R}^n$. A way to achieve this goal is to exploit the structure of Ψ and of its inverse Ψ^{-1}.

We remark that

$$\hat{P} = \mathbf{\Psi_Y} \mathbb{P}_{\mathbf{\Psi_X^\intercal}} V_{\mathbf{Z}} (\mathrm{diag}\begin{pmatrix} \sigma_{\mathbf{Z},1} & \cdots & \sigma_{\mathbf{Z},k} & 0 & \cdots \end{pmatrix})^\dagger,$$

which implies that (1.4) can be rewritten as

$$\begin{aligned} \tilde{x}_t(\theta) &= \Psi^{-1}(\sum_i \zeta_i \lambda_i^{t-1} \varphi_i(\theta)), \\ &= \Psi^{-1}(\sum_i \hat{P} \tilde{\zeta}_i \lambda_i^{t-1} \varphi_i(\theta)), \\ &= \Psi^{-1}(\Psi_{\mathbf{Y}} G), \end{aligned} \tag{3.4}$$

with matrix $G \in \mathbb{R}^{m \times k}$ defined as

$$G = C^\intercal (\tilde{\zeta}_1 \cdots \tilde{\zeta}_k) \, \mathrm{diag}(\tilde{\lambda}_{\ell,1}^{t-1} \cdots \tilde{\lambda}_{\ell,k}^{t-1})(\tilde{\varphi}_1(\theta) \cdots \tilde{\varphi}_k(\theta)). \tag{3.5}$$

The computation of $\Psi^{-1}(\Psi_{\mathbf{Y}} G)$ can often be done within a linear complexity in p, independently of n, as illustrated by the following particular cases. We recall that we have denoted the kernel computing the scalar product of vectors in $(\mathcal{L}^2(\mathbb{R}^p))^n$ by $h : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$; $(y, z) \to h(y, z) = \Psi(y)^\intercal \Psi(z)$. We detail in the following the computation of $\Psi^{-1}(\Psi_{\mathbf{Y}} G)$ in $\mathcal{O}(p)$ for two standard class of kernel. We also present the computation for a logarithmic kernel, which will serve in our numerical evaluation.

i) **Polynomial kernels.** The most simple situation occurs when Ψ^{-1} is linear. In this case we simply get that

$$\Psi^{-1}(\Psi_{\mathbf{Y}} G) = \mathbf{Y} G. \tag{3.6}$$

This simplification occurs for the particular case of polynomial kernels of the form

$$h(y, z) = (1 + y^\intercal z)^\gamma, \tag{3.7}$$

where $y, z \in \mathbb{R}^p$ and γ is a positive scalar. It is easy to see that in this case $y \in \mathbb{R}^p$ is up to a multiplicative factor a component of $\Psi(y) \in \mathbb{R}^n$, implying that Ψ^{-1} is linear, see Appendix B. Therefore in this case (3.6) holds.

ii) **Gaussian kernels.**
Using a Taylor expansion of the exponential function, it is straightforward to show that (unormalized) Gaussian kernels of the form

$$h(y, z) = \exp\left(-\frac{\|y - z\|_2^2}{2\sigma^2}\right), \tag{3.8}$$

correspond to scalar products $\Psi(y)^\intercal \Psi(z)$ of infinite vectors in $(\mathcal{L}^2(\mathbb{R}^p))^n$ with $n = \infty$ but countable, written formally as

$$\Psi(y) = \left(\exp\left(-\frac{\|y\|_2^2}{2\sigma^2}\right), \exp\left(-\frac{\|y\|_2^2}{2\sigma^2}\right)\frac{y^\intercal}{\sigma 1!}, \exp\left(-\frac{\|y\|_2^2}{2\sigma^2}\right)\frac{(y^2)^\intercal}{\sigma^2 2!}, \cdots\right)^\intercal.$$

We have used the short-hand notation $y^\gamma = (y_{(1)}^\gamma \cdots y_{(p)}^\gamma)^\intercal \in \mathbb{R}^p$ for $\gamma \in \mathbb{R}$. We notice that the inverse function $\Psi^{-1}(z)$ evaluated at any point $z = (z_{(1)}, z_{(2)}, z_{(3)}, \cdots)^\intercal \in \mathbb{R}^n$ can be identified as the vector $\frac{\sigma}{z_{(1)}}(z_{(2)}, \cdots, z_{(p+1)})^\intercal$. We deduce after some calculation that we have

$$(\Psi^{-1}(\Psi_{\mathbf{Y}} G))_{(j,\ell)} = \frac{\sum_{i=1}^m w_i \mathbf{Y}_{(j,i)} G_{(i,\ell)}}{\sum_{i=1}^m w_i G_{(i,\ell)}}, \tag{3.9}$$

with the weights

$$w_i = \exp\left(-\frac{\sum_{r=1}^p (\mathbf{Y}_{(r,i)})^2}{2\sigma^2}\right).$$

iii) **Logarithmic kernel.**
Consider a logarithmic kernel defined for vectors $y = (y_{(1)} \cdots y_{(p)})^\intercal \in \mathbb{R}^p$ and $z \in \mathbb{R}^p$ of components greater than -1 as

$$h(y, z) = (\log(y+1))^\intercal \log(z+1), \tag{3.10}$$

where we have used the short-hand notation

$$\log(y+1) = (\log(y_{(1)}+1) \cdots \log(y_{(p)}+1))^\intercal.$$

The kernel corresponds in this case to vectors of functions $\Psi(y) = \log(y+1) \in \mathbb{R}^n$ with $n = p$. Using an notation analogous to the logarithm for the exponential of a vector, we notice that we simply have $\Psi^{-1}(z) = \exp(z) - 1$ at a point $z \in \mathbb{R}^n$. We easily deduce that for entries of $\mathbf{Y}$ greater than -1 we have

$$(\Psi^{-1}(\Psi_{\mathbf{Y}} G))_{(j,\ell)} = \prod_{i=1}^m \left(\mathbf{Y}_{(j,i)} + 1\right)^{G_{(i,\ell)}} - 1. \tag{3.11}$$

We see that for these three particular cases, computing $\Psi^{-1}(\Psi_{\mathbf{Y}} G)$ only implies the evaluation of a function of the entries of matrix $\mathbf{Y}$. Therefore the computation is done within a linear complexity in p, independently of n.

3.3. Proposed Algorithms.

Algorithm 2 : OK-DMD

inputs: The $x_t(\theta_i)$'s and some θ

1) Compute steps 1) and 2) of Algorithm 1

2) Compute $(V_{\mathbf{Z}}, \Sigma_{\mathbf{Z}})$ by eigen-decomposition of $\mathbf{Z}^\intercal\mathbf{Z}$ with $\mathbf{Z}$ given by (2.1).

3) Compute matrix $(\tilde{A}_{\ell,k})^\intercal$ given in (3.1) and compute eigen-vector / eigen-value couples $\{(\tilde{\xi}_i, \tilde{\lambda}_{\ell,i})\}_{i=1}^k$.

4) Compute matrix $\tilde{A}_{r,k}$ given in (3.2) and compute eigen-vector / eigen-value couples $\{(\tilde{\zeta}_i, \tilde{\lambda}_{r,i})\}_{i=1}^k$.

5) Rescale $\tilde{\zeta}_i$'s so that $\tilde{\zeta}_i\, E\, \tilde{\xi}_i = 1$, with E given in (3.3).

6) Compute eigen-functions $\{\tilde{\varphi}_i(\theta)\}_{i=1}^k$ using (2.4).

7) Compute $\tilde{x}_t(\theta) = \Psi^{-1}(\Psi(\mathbf{Y})G)$ with G given in (3.5), using either (3.6), (3.9) or (3.11).

output: $\tilde{x}_t(\theta)$'s.

3.3.1. The OK-DMD Algorithm. We have now gathered all the ingredients necessary to design Algorithm 2. It computes the approximation (1.5) using the solution of the low-rank EDMD problem (1.7), based only on computation with vectors in $\mathbb{R}^p$. Its off-line and on-line complexities scale as $\mathcal{O}(m^2(m+p))$ and $\mathcal{O}(p(k+m^2))$, which are comparable to the state-of-the-art K-DMD algorithm.

We finally remark that, if we consider instead the solution of the unconstrained EDMD problem (1.8), the approximation (1.5) is simply obtained using the OK-DMD algorithm in the particular case where $k \geq m$. The algorithm can be further simplified if we assume that the matrices $\mathbf{\Psi_X}$ and $\mathbf{\Psi_Y}$ are full rank. Indeed, in this case we have that $\mathbb{P}_{\mathbf{\Psi_X}^\intercal} = \mathbb{P}_{\mathbf{\Psi_Y}^\intercal} = I_m$ and that $\mathbf{Z} = \mathbf{\Psi_Y}$. The assumptions lighten the expressions of several matrices as presented in Algorithm 3. In particular, we recover in this case the matrix $(\tilde{A}_m^\ell)^\intercal = R\,\mathbf{\Psi_Y^\intercal}\mathbf{\Psi_X}\,R^\intercal$ used for eigen-decomposition at the core of the K-DMD algorithm.

3.3.2. The Low-Rank K-DMD Algorithm. We propose now a low-rank extension of the K-DMD algorithm [13]. The idea is to design an algorithm exploiting the optimal solution (2.2) of the low-rank EDMD problem, but relying on the eigen-mode approximation of K-DMD (steps 6) -7) in Algorithm 1). Therefore, as for K-DMD, this extension detailed in Algorithm 4 will produce a sub-optimal solution, on the contrary to OK-DMD. Its off-line complexity will scale as $\mathcal{O}(m^2(m+p))$ as for K-DMD or OK-DMD. However, its on-line complexities will scale as $\mathcal{O}(pk)$, that is slightly lower than for these two algorithms. Besides, it will be useful in our numerical evaluation to isolate the eigen-mode approximation error from the error due to ignoring constraints in problem (1.7).

4. Numerical Evaluation.

We describe in the following the benchmark used for the evaluation of the proposed algorithms.

4.1. Experimental Setup.

To build the reduced models, we use snapshots produced by a a mixture of a linear and a non-linear dynamical model. The data is composed of trajectories of the dynamics (1.1) satisfying the model

$$f(x_{t-1}) = (x_{t-1}+1)^\beta - 1, \quad \beta \in \mathbb{R}. \tag{4.1}$$

Algorithm 3 : OK-DMD, case $k \geq m = \text{rank}(\mathbf{\Psi_X}) = \text{rank}(\mathbf{\Psi_Y})$

inputs: The $x_t(\theta_i)$'s and some θ

1) Compute step 1) and 2) of Algorithm 2.

2) Compute $(V_{\mathbf{\Psi_Y}}, \Sigma_{\mathbf{\Psi_Y}})$ through the eigen decomposition of $\mathbf{\Psi_Y^\intercal \Psi_Y}$.

3) Compute matrices $(\tilde{A}_{\ell,k})^\intercal = R\,\mathbf{\Psi_Y^\intercal \Psi_X}\,R^\intercal$ and $\tilde{A}_{r,k} = (C^\dagger)^\intercal R^\intercal\, R\,\mathbf{\Psi_X^\intercal \Psi_Y}\,C^\intercal$ with $C = \Sigma^\dagger_{\mathbf{\Psi_Y}} V^\intercal_{\mathbf{\Psi_Y}}$.

4) Compute the eigen-vectors and eigen-values $\{(\tilde{\xi}_i, \tilde{\lambda}_{\ell,i})\}_{i=1}^k$ and $\{(\tilde{\zeta}_i, \tilde{\lambda}_{r,i})\}_{i=1}^k$.

5) Rescale the $\tilde{\zeta}_i$'s so that $\tilde{\zeta}_i\, E\, \tilde{\xi}_i = 1$, where $E = C\mathbf{\Psi_Y^\intercal \Psi_X} R$.

6) Compute steps 6) and 7) of Algorithm 2.

output: $\tilde{x}_t(\theta)$'s.

Algorithm 4 : Low-rank K-DMD

inputs: The $x_t(\theta_i)$'s and some θ

1) Compute step 1) to 3) of Algorithm 2

2) Compute step 5) to 7) of Algorithm 1

output: $\tilde{x}_t(\theta)$'s using (1.6) with $\nu_{i,t} = \tilde{\lambda}_i^{t-1}\tilde{\varphi}_i(\theta)$.

More precisely, $x_t(\theta_i)$'s for $i = 1, \cdots, N/2$ (resp. for $i = N/2+1, \cdots, N$) correspond to the particular cases $\beta = \tau$ (resp. $\beta = \alpha$) with $\tau = 1$ (resp. $\alpha \in \{-1, 1, 2, 3\}$). For instance, in the case $\alpha = 2$, the $x_t(\theta_i)$'s will be a mixture of trajectories of a linear model and a quadratic model. The components of the initial conditions, *i.e.*, the θ_i's for $i = 1, \cdots, N/2$, are drawn independently according to the standard normal distribution and we set $\theta_{i+N/2} = \theta_i$ for $i = 1, \cdots, N/2$. We finally modify r out of the m trajectories so that we have either $\text{rank}(\mathbf{X}) = m - r$ or $\text{rank}(\mathbf{\Psi_X}) = m - r$.

We assess the gain in performance brought by learning the reduced model with OK-DMD by a comparison with state-of-the-art methods. Our benchmark is composed of the following algorithms:

- the original K-DMD proposed in [13] (Algorithm 1)
- the low-rank DMD algorithm proposed in [6]
- the low-rank extension of K-DMD (Algorithm 4)
- the OK-DMD (Algorithm 2)

Let $\tilde{\mathbf{Y}} \in \mathbb{R}^{p\times m}$ be a matrix whose i-th column is $\tilde{x}_2(\theta)$ given by (1.5) where θ is identified to the i-th column of $\mathbf{X}$. We will compare the approximations produced by the different algorithms in terms of the normalised error norm

$$\|\mathbf{Y} - \tilde{\mathbf{Y}}\|_F / \|\mathbf{Y}\|_F. \tag{4.2}$$

In particular, for reduced models based on low-rank DMD, this criterion simplifies into $\|\mathbf{Y} - A_k^\star \mathbf{X}\|_F / \|\mathbf{Y}\|_F$, where $A_k^\star$ given in (2.2) is the optimal solution for Ψ identified to the identity operator.

This criterion of performances measures the capabilities of the methods to reproduce the training data using a low-rank approximation. We remark that the low-rank DMD algorithm provides the optimal linear solution minimising the criterion of interest. On the contrary, for other choice of Ψ, this criterion does not corresponds necessarily to the objective function to be minimised. It results that low-rank DMD will be generally favoured according to this criterion, in comparison to kernel-based methods. But, as we will see in the results, the error norm of kernel-based methods

can be lowered for well-chosen function Ψ. This function should be chosen so that it generates a a sufficiently *sparse* representation of the data. On the other hand, this criterion does not directly measure the quality of estimation of certain features of the dynamics. In particular, it does not necessarily reveals the capabilities of the methods to estimate accurately the first k Koopman eigen-values, as it is done in [13]. Criterion (4.2) seems nevertheless appropriated to our study, since the focus of the present work is reduced modeling rather than data analysis.

4.2. Results. All figures are plots displaying for the different algorithms the evolution of the error norm (4.2) committed by the reduced model as a function of the rank k of the constraint in the approximation problem (1.7).

In particular, Figure 4.1 illustrates the impact of the kernel choice and shows the influence of the ratio p/m. Each plot of this figure corresponds either to the polynomial kernel (3.7) with $\gamma = 10$, the Gaussian kernel (3.8) with $\sigma = 50$ or the logarithmic kernel (3.10). Each plot also corresponds either to the ratio $p/m = 1$ or 1000. We set $\alpha = 2$ (quadratic model) and full-rank matrices $\mathbf{X}$ and $\Psi_{\mathbf{X}}$ (*i.e.*, $r = 0$).

As expected, we see that OK-DMD is nearly everywhere more accurate than K-DMD and its low-rank extension. We also often remark a gain in accuracy brought by using the low-rank K-DMD algorithm instead of its original version. The enhancement illustrates the importance of considering low-rank constraints in problem (1.7). The extra gain in accuracy between low-rank K-DMD and OK-DMD can be interpreted as the fact that the OK-DMD algorithm computes exactly the reduced model (3.4) instead of making the least-square approximation (2.8). We mention that, in theory, all these kernel-based algorithms yield identical results in the case Ψ^{-1} is a matrix in $\mathbb{R}^{p\times n}$, eigen-modes belong to the span of $\mathbf{Y}$ and $k = m$. This setting occurs at the abscise $k = m = 20$ in the top left plot of Figure 4.1. This plot corresponds to a polynomial kernel (satisfying $\Psi^{-1} \in \mathbb{R}^{p\times n}$) and to a ratio of $p/m = 1$ (the span of $\mathbf{Y}$ fills in this case the entire state space). We check that at this particular abscise the difference between error norms of the different algorithms can be neglected. Moreover, in agreement with (1.2) and (1.7), we observe that an exact reconstruction is guaranteed for $k = m$ with OK-DMD (or low-rank DMD) but not with K-DMD if $\mathbf{\Psi_X}$ (or $\mathbf{X} = \Psi^{-1}(\mathbf{\Psi_X})$) is full-rank.

Increasing the dimensionality p of the state space does not influence too much the OK-DMD performances. We notice however a moderated gain in accuracy. On the other hand, the performance of OK-DMD is very sensitive to the choice of kernel. A polynomial kernel yields poor results, the Gaussian kernel brings some improvement while the logarithmic kernel induces a vanishing error norm for $k \geq m/2 = 10$. This behaviour follows from the properties of the logarithmic kernel implying that $\text{rank}(\mathbf{\Psi_Y}) = m/2$ and an exact reconstruction with only $m/2$ components. Indeed, it is easy to see that the first 10 columns of $\mathbf{\Psi_Y}$ are proportional by a factor $\alpha = 2$ to the last 10 columns).

Although OK-DMD performs better than the other kernel-based methods, low-rank DMD is significantly more accurate in the case of polynomial and Gaussian kernels. As pointed out previously, this is likely to be due to the fact that the objective function optimised by the DMD algorithm is exactly the criterion used for evaluation and that for these kernels, function Ψ does not generate a sparse data representation (on the contrary to the logarithmic kernel).

Figure 4.2 displays the influence of parameter α in the non-linear model (4.1) on

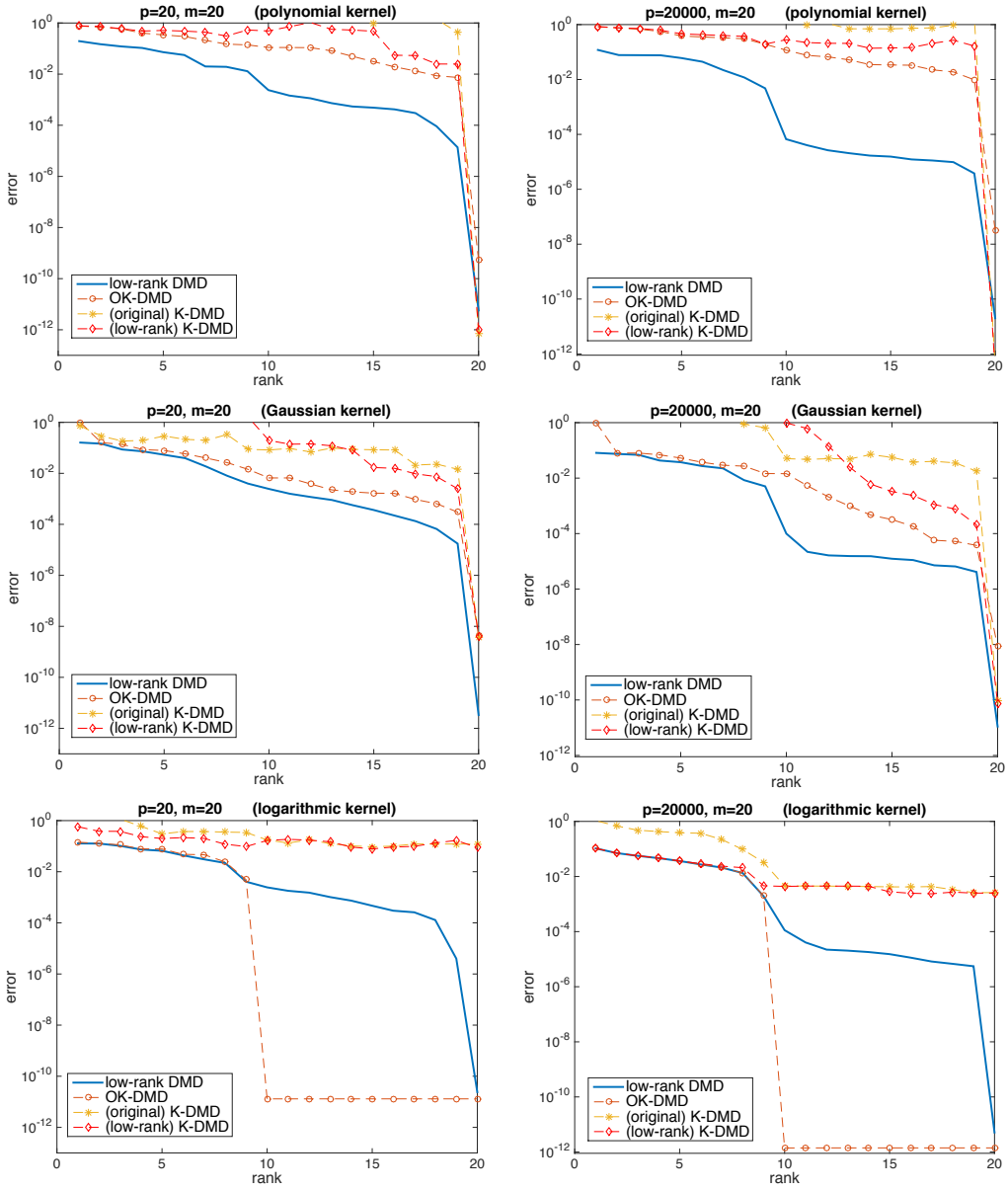

FIG. 4.1. *Influence of the kernel and the ratio p/m. Evaluation of the error norm $\|\mathbf{Y} - \tilde{\mathbf{Y}}\|_F / \|\mathbf{Y}\|_F$ as a function of the rank k of the reduced-model approximation. See details in Section 4.*

the algorithms performances for a logarithmic kernel and the ratio $p/m = 1$. The evolution of the error norm with respect to the approximation rank is plotted for $\alpha = -1, 1, 2$ and 3. We consider also here full-rank matrices $\mathbf{X}$ and $\Psi_{\mathbf{X}}$ (*i.e.*, $r = 0$). In order to lighten the presentation we will omit from now the evaluation of low-rank extension of K-DMD.

The case $\alpha = 1$ corresponds to a purely linear evolution with the first $m/2$ first

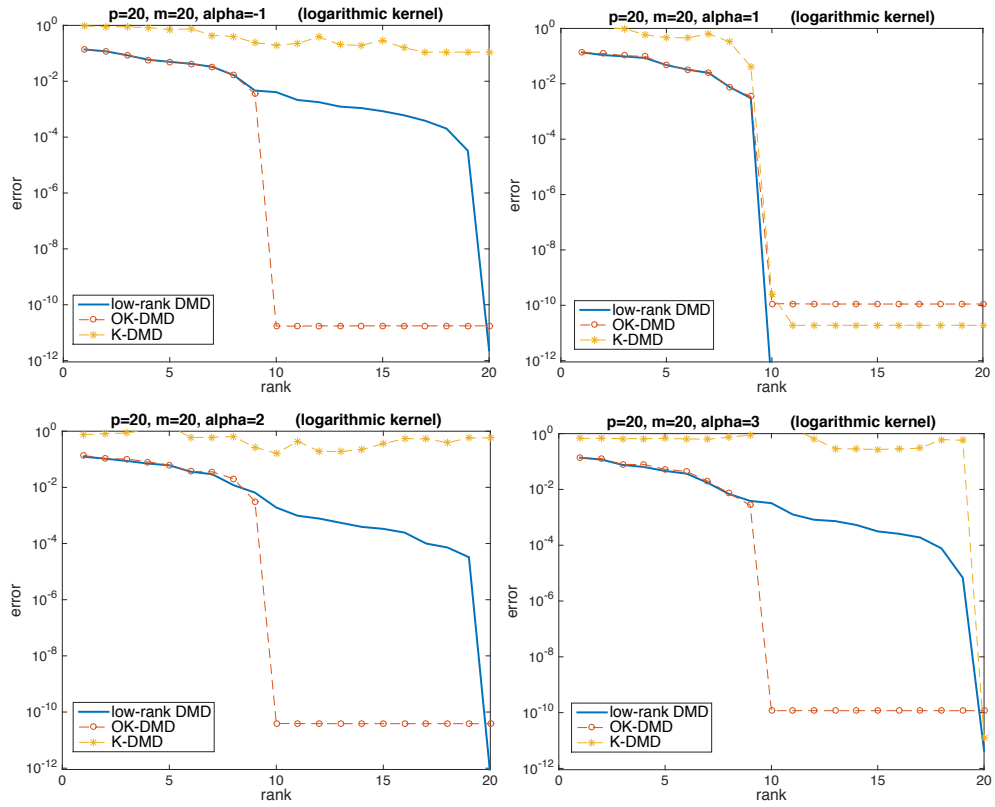

FIG. 4.2. *Influence of the degree of non-linearity α. Evaluation of the error norm $\|\mathbf{Y} - \tilde{\mathbf{Y}}\|_F / \|\mathbf{Y}\|_F$ as a function of the rank k of the reduced-model approximation for increasing values of α (from left to right and top to down). See details in Section 4.*

columns of $\mathbf{\Psi_Y}$ equal to the $m/2$ last ones. Therefore in this situation we have that $\text{rank}(\mathbf{\Psi_Y}) = m/2$ and we obtain an exact reconstruction with only $k = m/2$ components for any algorithm. For other values of α, we recover behaviours for the different algorithms which are in agreement with those described in Fig 4.1. We note that the degree α impacts only slightly the performances of the algorithms making the analysis and the determination of a global tendency not obvious.

Figure 4.3 illustrates the influence of the rank deficiencies $\text{rank}(\mathbf{X}) = m - r$ or $\text{rank}(\mathbf{\Psi_X}) = m - r$ with $r = 1$ for a logarithmic kernel in the case $\alpha = 2$ and a ratio $p/m = 10$. More precisely, the plots correspond respectively to the settings:

a) $\text{rank}(\mathbf{X}) = m - 1$, $\text{rank}(\mathbf{\Psi_X}) = m$, $\text{rank}(\mathbf{Y}) = m$ and $\text{rank}(\mathbf{\Psi_Y}) = m$,

b) $\text{rank}(\mathbf{X}) = m$, $\text{rank}(\mathbf{\Psi_X}) = m - 1$, $\text{rank}(\mathbf{Y}) = m$ and $\text{rank}(\mathbf{\Psi_Y}) = m/2$.

Unsurprisingly, for setting *a)*,, the error norm for the low-rank DMD decays very slowly (and in fact the decrease almost vanishes around $k \geq 3$). This non-zero asymptotic behaviour originates from the fact that above a certain value of k, $\|\mathbf{Y} - A_k^\star \mathbf{X}\|_F$ is dominated by the second term of (2.3). According to Theorem 2.1, this second term is the sum of the squared singular values of $\mathbf{Y}$, weighted by the scalar product of the right singular vectors of $\mathbf{Y}$ with the right singular vector of $\mathbf{X}$ associated to its zero singular value. Or in simpler words, if $\mathbf{Y}$ is full rank, the error

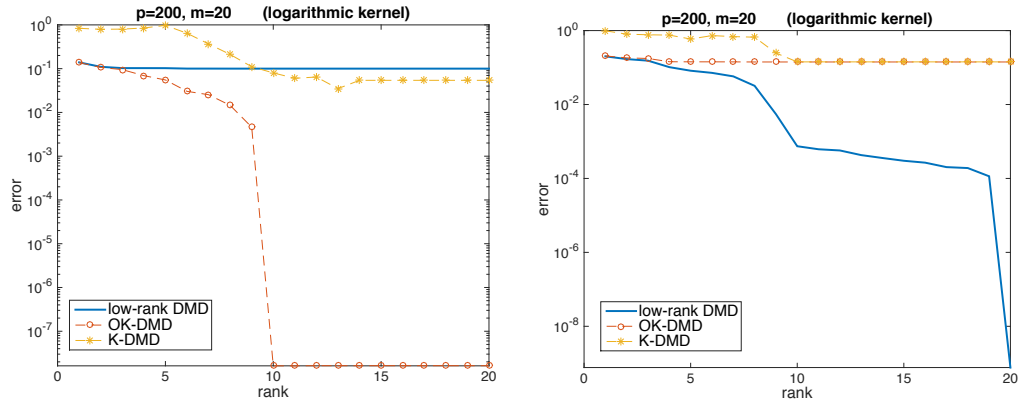

FIG. 4.3. *Influence of rank deficiency. Evaluation of the error norm* $\|\mathbf{Y} - \tilde{\mathbf{Y}}\|_F / \|\mathbf{Y}\|_F$ *as a function of the rank* k *of the reduced-model approximation for setting a) where* $\mathrm{rank}(\mathbf{X}) < m$ *(left) and setting b) where* $\mathrm{rank}(\mathbf{\Psi_X}) < m$ *(right). See details in Section 4.*

norm has an non-zero asymptotic value because the right singular vector of $\mathbf{X}$, which is associated to its zero singular value, is in the span of the rows of $\mathbf{Y}$. Because $\mathbf{\Psi_X}$ is full rank, the OK-DMD algorithm does not suffer from this deficiency and the error norm behaviour is consistent with our previous analysis. Besides, we observe again the sub-optimality of the K-DMD algorithm.

The situation is different in the case of setting *b)*: the error norm related to OK-DMD saturates, while for low-rank DMD it keeps decaying until the error vanishes. The analysis can be transposed: the error norm has an non-zero asymptotic value because the right singular vector of $\mathbf{\Psi_X}$, which is associated to its zero singular value, belongs to the span of the right singular vectors of $\mathbf{\Psi_Y}$, which are associated to its $m/2$ non-zero singular values. As expected, we observe for $k < 10$ that OK-DMD performs significantly better than K-DMD. However, for greater values of k, we note that the error norm obtained with the K-DMD algorithm converges towards the optimal value obtained with OK-DMD. On the contrary, the good performance of low-rank DMD is explained by the fact that $\mathbf{X}$ is full rank.

5. Conclusion. This work provides an optimal algorithm to solve the low-rank EDMD problem using kernel-based computation. The idea underlying the method is to rely on a low-dimensional representation of the left and right eigen-vectors of the solution $A_k^\star$ of this problem. The representation enables to obtain: *i)* closed-form eigen-functions of the Koopman operator approximation, expressed in terms of the left eigen-vectors of $A_k^\star$; *ii)* the reduced model approximation $\tilde{x}_t(\theta)$ for a given initial condition $x_1 = \theta$, given in terms of the right eigen-vectors of $A_k^\star$ and exploiting the particular structure of the different kernels.

On the contrary to the original K-DMD algorithm, the proposed algorithm called OK-DMD is optimal in the sense it solves exactly the low-rank EDMD problem and is relieved from K-DMD restrictive assumptions.

Particularisations of the OK-DMD algorithm are presented respectively for polynomial, Gaussian and logarithmic kernels. The algorithm presents an off-line and on-line complexity of $\mathcal{O}(m(p+m^2))$ and $\mathcal{O}(p(k+m^2))$ which is comparable to the complexity of the state-of-the-art K-DMD algorithm. In particular, it is adequate for the case $p \ll n$, because of the linearity with respect to the ambient dimension p and

independence of the functional space dimension n.

A numerical evaluation shows the gain brought by the proposed OK-DMD algorithm in comparison to the state-of-the-art. In particular, we show that considering explicitly the low-rank constraints and computing exactly the parameters of the reduced model yield significative improvements. Finally, we also evaluate the influence of several features on the algorithm performances. In particular we highlight the importance of the kernel, which should be carefully chosen according to the data.

Acknowledgements. This work was supported by the "Agence Nationale de la Recherche" through the GERONIMO project (ANR-13-JS03-0002).

Appendix A. Obtention of Reduced Models (1.2) **and** (1.5)**.** We present hereafter the derivation of (1.2) and (1.5). It relies partially on a reformulation of the material presented in [12].

The sought reduced models rely on the finite approximation of the Koopman operator (related to the high-dimensional system (1.1)). We introduce hereafter this approximation. We will use the dictionary $\mathcal{D} = \{\psi_1, \cdots, \psi_n\}$ and its span $\mathcal{F}_\mathcal{D} \subset \mathcal{L}^2(\mathbb{R}^p)$. Any function $\phi \in \mathcal{F}_\mathcal{D}$ can be decomposed as

$$\phi(x) = \sum_{i=1}^{n} c_i^\phi \psi_i(x) = c^\phi \Psi(x), \tag{A.1}$$

with the vector-valued function $\Psi = (\psi_1 \cdots \psi_n)^\intercal \in (\mathcal{F}_\mathcal{D})^n$ and the row-vector of coefficients $c^\phi \in \mathbb{R}^{1\times n}$. For a vector-valued function $\Phi \in (\mathcal{F}_\mathcal{D})^n$, this decomposition writes in matrix notation as

$$\Phi(x) = \mathbf{c}^\Phi \Psi(x), \tag{A.2}$$

with a matrix of coefficients

$$\mathbf{c}^\Phi = \begin{pmatrix} \mathbf{c}_1^\phi \\ \vdots \\ \mathbf{c}_n^\phi \end{pmatrix} \in \mathbb{R}^{n\times n}$$

We can now justify that $\hat{A}_k$ in (1.2) can be seen as a finite approximation of the Koopman operator. The Koopman operator denoted by $\mathcal{K}$ is defined for any function $\phi \in \mathcal{L}^2(\mathbb{R}^p)$ by the composition

$$\mathcal{K}\phi(x_t) = \phi(f_t(x_t)) = \phi(x_{t+1}),$$

or equivalently using the matrix notation (A.2)

$$\mathcal{K}\Phi(x_t) = \Phi(x_{t+1}), \tag{A.3}$$

for any function $\Phi \in (\mathcal{L}^2(\mathbb{R}^p))^n$. By decomposing the function $\Phi(x_t)$ and $\Phi(x_{t+1})$ of $(\mathcal{L}^2(\mathbb{R}^p))^n$ in $(\mathcal{F}_\mathcal{D})^n$ with (A.2), we get on the one hand

$$\begin{aligned} \mathcal{K}\Phi(x_t) &= \mathcal{K}(\mathbf{c}^\Phi \Psi(x_t)), \\ &= \mathbf{c}^\Phi \mathcal{K}\Psi(x_t), \end{aligned} \tag{A.4}$$

where we have exploited the linearity of the Koopman operator, and on the other hand

$$\Phi(x_{t+1}) = \mathbf{c}^{\Phi}\Psi(x_{t+1}). \tag{A.5}$$

We note that equalities (A.4) and (A.5) rely on the fact that the vector space $\mathcal{F}_{\mathcal{D}}$ is an invariant subspace[4] of $\mathcal{K}$, *i.e.*, $\Phi(x_t) \in (\mathcal{F}_{\mathcal{D}})^n$ necessarily implies that $\Phi(x_{t+1}) \in (\mathcal{F}_{\mathcal{D}})^n$. Equality (A.3) then shows that vectors $\Psi(x_t), \Psi(x_{t+1}) \in \mathbb{R}^n$ are linear dependent, so that there exists a matrix $A \in \mathbb{R}^{n\times n}$ such that

$$A\Psi(x_t) = \Psi(x_{t+1}). \tag{A.6}$$

It follows that we can also write

$$\Phi(x_{t+1}) = \mathbf{c}^{\Phi}A\Psi(x_t). \tag{A.7}$$

Using (A.4) and (A.7), equality (A.3) thus yields

$$\mathbf{c}^{\Phi}\mathcal{K}\Psi(x_t) = \mathbf{c}^{\Phi}A\Psi(x_t).$$

Since Φ can be any function of $(\mathcal{L}^2(\mathbb{R}^p))^n$, this implies the representation of the Koopman operator in $(\mathcal{F}_{\mathcal{D}})^n$

$$\mathcal{K}\Psi(x_t) = A\Psi(x_t). \tag{A.8}$$

Then imposing that matrix A is of rank at most k is equivalent to impose that the Koopman operator approximation belongs to a k-dimensional sub-space of $(\mathcal{F}_{\mathcal{D}})^n$. In particular, by making the approximation

$$A \approx \hat{A}_k,$$

with $\hat{A}_k$ of rank at most k, we obtain the second equation of recursion (1.2). Then, assuming there exists Ψ^{-1} such that

$$x = \Psi^{-1}\Psi(x), \tag{A.9}$$

we obtain the sought reduced-model (1.2).

Now, to obtain reduced model (1.5), we assume that $\hat{A}_k$ is diagonalisable. Recall that $\{\xi_i\}_{i=1}^n$ are left eigen-vectors of $\hat{A}_k$ related to the eigen-values $\{\lambda_i\}_{i=1}^n$, where at most k eigen-values are non-zero. Using the linearity of $\mathcal{K}$ and relation (A.8), we obtain

$$\mathcal{K}\xi_i^{\intercal}\Psi(x) = \xi_i^{\intercal}\mathcal{K}\Psi(x) \approx \xi_i^{\intercal}\hat{A}_k\Psi(x) = \lambda_i\xi_i^{\intercal}\Psi(x), \quad \forall i = 1, \cdots, n,$$

[4] Note that in the case vector space $\mathcal{F}_{\mathcal{D}}$ is not an invariant subspace of $\mathcal{K}$, we need to introduce a residual function $r^{\Phi} \in (\mathcal{L}^2(\mathbb{R}^p))^n \backslash (\mathcal{F}_{\mathcal{D}})^n$. By decomposing the function $\Phi(x_t)$ and $\Phi(x_{t+1})$ of $(\mathcal{L}^2(\mathbb{R}^p))^n$ in $(\mathcal{F}_{\mathcal{D}})^n$ with (A.2), we get on the one hand

$$\begin{aligned}\mathcal{K}\Phi(x_t) &= \mathcal{K}(\mathbf{c}^{\Phi}\Psi(x_t) + r^{\Phi}(x_t)),\\ &= \mathbf{c}^{\Phi}\mathcal{K}\Psi(x_t) + \mathcal{K}r^{\Phi}(x_t),\end{aligned}$$

where we have exploited the linearity of the Koopman operator, and on the other hand

$$\Phi(x_{t+1}) = \mathbf{c}^{\Phi}\Psi(x_{t+1}) + r^{\Phi}(x_{t+1}).$$

Searching an approximation in $(\mathcal{F}_{\mathcal{D}})^n$, is then equivalent to neglect the additive residuals. We recover in this case (A.4) and (A.5).

which shows that $\varphi_i \in \mathcal{F}_\mathcal{D}$ given by

$$\varphi_i(x) = \xi_i^\intercal \Psi(x), \quad \forall i = 1, \cdots, n,$$

is an approximation of an eigen-function of $\mathcal{K}$. Equivalently, the latter can be rewritten in matrix form as

$$\Upsilon(x) = \Xi^{-1}\Psi(x), \tag{A.10}$$

with

$$\Upsilon(x) = \begin{pmatrix} \varphi_1(x) \\ \vdots \\ \varphi_n(x) \end{pmatrix},$$

and where we recall that $(\Xi^{-1})^\intercal = (\xi_1 \cdots \xi_n) \in \mathbb{C}^{n\times n}$ is the conjugate transpose of the inverse of $\Xi = (\zeta_1 \cdots \zeta_n) \in \mathbb{C}^{n\times n}$. Using (A.9) and (A.10), we get

$$\begin{aligned} x_t &= \Psi^{-1}(\mathcal{K}\Psi(x_{t-1})) \\ &\approx \Psi^{-1}(\mathcal{K}\Xi\Upsilon(x_{t-1})), \\ &= \Psi^{-1}(\Xi\mathcal{K}\Upsilon(x_{t-1})), \\ &= \Psi^{-1}(\Xi\Lambda\Upsilon(x_{t-1})), \end{aligned}$$

where we have exploited the linearity of operator $\mathcal{K}$ and the fact that the components of Υ are eigen-functions. Iterating this recursive decomposition we obtain that

$$x_t \approx \Psi^{-1}(\Xi\Lambda^{t-1}\Upsilon(x_1)),$$

rewritten as

$$x_t \approx \Psi^{-1}\left(\sum_{i=1}^{\mathrm{rank}(\hat{A}_k)} \lambda_i \zeta_i \tilde{\varphi}_i(x_1)\right),$$

which is the sought reduced model (1.5).

Appendix B. The Kernel Trick. Assume there exists a kernel function $h : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}; (y, z) \to h(y, z)$ that computes inner products of vectors of function $\Psi(y), \Psi(z) \in \mathbb{R}^n$ for any pair of points $y, z \in \mathbb{R}^p$, that is, $h(y, z) = \Psi(y)^\intercal\Psi(z)$. This asumption implies that inner products of elements in the functional space are elements of a reproducing kernel Hilbert space [1]. The kernel trick is a common technique exploiting this property: inner products of functions in $\mathbb{R}^n$ can implicitly computed within a complexity scaling as $\mathcal{O}(p)$ rather than $\mathcal{O}(n)$[2]. In the context of our EDMD problem, the choice of the kernel h defines Ψ, and therefore determines the dictionary $\mathcal{D} = \{\psi_1, \cdots, \psi_n\}$. Since this work presuppose the knowledge of $\mathcal{D}$, we consider the kernel h as a known ingredient of our problem. However, beyond the focus of the present work, we mention that the optimal choice of kernel, or the optimal choice of the dictionary, is an open question.

We illustrate the inner product computation with the kernel trick using the polynomial kernel defined in (3.7). In the case $\gamma = 2$ and $p = 2$, the kernel can be evaluated by a simple inner product in $\mathbb{R}^2$. Expanding the kernel, we obtain

$$h(y, z) = 1 + 2y_1z_1 + 2y_2z_2 + 2y_1y_2z_1z_2 + y_1^2z_1^2 + y_2^2z_2^2,$$

where $y = (y_1\, y_2)^\intercal$ and $z = (z_1\, z_2)^\intercal$ so that we have $h(y,z) = \Psi(y)^\intercal\Psi(z)$, for

$$\Psi(y) = \begin{pmatrix} 1 & \sqrt{2}y_1 & \sqrt{2}y_2 & \sqrt{2}y_1y_2 & y_1^2 & y_2^2 \end{pmatrix}^\intercal.$$

Therefore the kernel trick avoids here to evaluate explicitly $\Psi(y)$ and $\Psi(z)$ followed by an inner product in $\mathbb{R}^6$.

Appendix C. Proof of Proposition 3.1. Exploiting the SVD of $\mathbf{\Psi_X}$, the transpose of the solution (2.2) is

$$(A_k^\star)^\intercal = U_{\mathbf{\Psi_X}}\Sigma_{\mathbf{\Psi_X}}^\dagger V_{\mathbf{\Psi_X}}^\intercal \mathbf{\Psi_Y}^\intercal \hat{P}\hat{P}^\intercal.$$

We begin by proving that the $U_{\mathbf{\Psi_X}}\tilde{\xi}_i$'s are right eigen-vectors of $(A_k^\star)^\intercal$. Using the definition (3.1), we verify after some algebraic manipulations that

$$U_{\mathbf{\Psi_X}}^\intercal (A_k^\star)^\intercal U_{\mathbf{\Psi_X}} = (\tilde{A}_{\ell,k})^\intercal, \tag{C.1}$$

so that we have the following equality

$$(A_k^\star)^\intercal U_{\mathbf{\Psi_X}}\alpha_i = U_{\mathbf{\Psi_X}}(\tilde{A}_{\ell,k})^\intercal\alpha_i.$$

By setting $\alpha_i = \tilde{\xi}_i$ and exploiting the fact that $(\tilde{\xi}_i, \tilde{\lambda}_i)$ are couples of eigen-vectors and eigen-values of $(\tilde{A}_{\ell,k})^\intercal$, we get that for $i = 1, \cdots, k$

$$(A_k^\star)^\intercal U_{\mathbf{\Psi_X}}\tilde{\xi}_i = U_{\mathbf{\Psi_X}}(\tilde{A}_{\ell,k})^\intercal\tilde{\xi}_i = \tilde{\lambda}_i U_{\mathbf{\Psi_X}}\tilde{\xi}_i,$$

which proves that $U_{\mathbf{\Psi_X}}\tilde{\xi}_i$ and $\tilde{\lambda}_i$ are eigen-vectors and eigen-values of $(A_k^\star)^\intercal$.

We continue by showing that the $\hat{P}\tilde{\zeta}_i$'s are right eigen-vectors of $A_k^\star$. It easy to verify that

$$\hat{P}^\intercal A_k^\star \hat{P} = \tilde{A}_{r,k}, \tag{C.2}$$

with $\tilde{A}_{r,k} = \hat{P}\mathbf{\Psi_Y}\mathbf{\Psi}_\mathbf{X}^\dagger\hat{P}$ so that

$$A_k^\star\hat{P}\beta = \hat{P}\tilde{A}_{r,k}\beta,$$

for any $\beta \in \mathbb{R}^m$. Because $\tilde{\zeta}_i$ are eigen-vectors of A_k^r, we deduce from the previous relation that

$$A_k^\star\hat{P}\tilde{\zeta}_i = \hat{P}A_k^r\tilde{\zeta}_i = \tilde{\lambda}_i^r\hat{P}\tilde{\zeta}_i.$$

This proves that the $\hat{P}\tilde{\zeta}_i$'s and the $\tilde{\lambda}_i^r$'s are right eigen-vectors and eigen-values of $A_k^\star$.

Now, because $U_{\mathbf{\Psi_X}}$ is unitary, it follows from (C.1) that $\lambda_i = \tilde{\lambda}_{\ell,i}$ for $i = 1, \cdots, n$. Moreover, it follows from the definition of $A_k^\star$ that the ζ_i's are in the span of the columns of $\hat{P}$, *i.e.*, that every eigen-vector writes $\hat{P}\alpha$, with some $\alpha_i \in \mathbb{C}^m$. It follows from (C.2) that $\lambda_i = \tilde{\lambda}_i^r$ for $i = 1, \cdots, n$. We finally obtain that $\{\tilde{\lambda}_{\ell,i}, i = 1, \cdots, k\} = \{\tilde{\lambda}_{r,i} = 1, \cdots, k\} = \{\lambda_i = 1, \cdots, k\}$. □

REFERENCES

[1] Berlinet, A., Thomas-Agnan, C.: Reproducing Kernel Hilbert Spaces in Probability and Statistics. Springer US (2003)

[2] Bishop, C.M.: Pattern recognition (2006)

[3] Budišić, M., Mohr, R., Mezić, I.: Applied koopmanism a. Chaos: An Interdisciplinary Journal of Nonlinear Science **22**(4), 047,510 (2012)
[4] Chen, K.K., Tu, J.H., Rowley, C.W.: Variants of dynamic mode decomposition: boundary condition, koopman, and fourier analyses. Journal of nonlinear science **22**(6), 887–915 (2012)
[5] Golub, G., Van Loan, C.: Matrix Computations. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press (2013)
[6] Héas, P., Herzet, C.: Low rank dynamic mode decomposition: Optimal solution and algorithm. arXiv e-prints (2017)
[7] Jovanovic, M., Schmid, P., Nichols, J.: Low-rank and sparse dynamic mode decomposition. Center for Turbulence Research Annual Research Briefs pp. 139–152 (2012)
[8] Klus, S., Koltai, P., Schütte, C.: On the numerical approximation of the perron-frobenius and koopman operator. arXiv preprint arXiv:1512.05997 (2015)
[9] Kutz, J.N., Brunton, S.L., Brunton, B.W., Proctor, J.L.: Dynamic mode decomposition: Data-driven modeling of complex systems (2016)
[10] Schmid, P.J.: Dynamic mode decomposition of numerical and experimental data. Journal of Fluid Mechanics **656**, 5–28 (2010)
[11] Tu, J.H., Rowley, C.W., Luchtenburg, D.M., Brunton, S.L., Kutz, J.N.: On dynamic mode decomposition: Theory and applications. Journal of Computational Dynamics **1**(2), 391–421 (2014)
[12] Williams, M.O., Kevrekidis, I., Rowley, C.: A data–driven approximation of the koopman operator: Extending dynamic mode decomposition. Journal of Nonlinear Science **25**(6), 1307–1346 (2015)
[13] Williams, M.O., Rowley, C.W., Kevrekidis, I.G.: A kernel-based approach to data-driven koopman spectral analysis. arXiv preprint arXiv:1411.2260 (2014)